ABSTRACT

PERFORMANCE ENHANCEMENT IN WIRELESS LOCAL AREA NETWORKS

by

Daji Qiao

Chair: Kang G. Shin

In recent years, the IEEE 802.11 WLANs (Wireless Local-Area Networks) have received significant attention due to their higher bandwidth than wide-area cellular systems and their use of unlicensed (free-to-use) operational bands. Most 802.11-compliant products available in the market only implement the mandatory DCF (Distributed Coordination Function) at the MAC (Medium Access Control) layer, and because of the inherent design flaws of the DCF protocol, the current 802.11 systems present poor fairness-performance/energy-efficiency/channel-utilization. Now, with the emergence of new high-speed 802.11 PHYs (physical layers), the problems become even worse, and it is essential to re-examine the original DCF and make necessary modifications. This dissertation addresses the problem of enhancing the performance of the 802.11 DCF systems from various related but distinct angles.

To achieve the weighted fairness and maximize the channel utilization for data communications in 802.11 DCF systems, we propose a simple enhancement to the 802.11 DCF, called the WB-DCF. It reduces the number of contending stations by introducing a new polling mode in addition to the contention mode used in the DCF, and achieves a low

frame collision probability with an advanced contention window selection scheme based on runtime load estimation. Besides, the relative weights of traffic flows are also taken into consideration in the polling scheme and the contention window selection to achieve the weighted fairness.

We investigate the problem of minimizing the energy consumption in the emerging 802.11a/h systems that will provide a structured means to support intelligent TPC (Transmit Power Control). We develop a novel scheme, called MiSer, as an optimal solution. The key idea is to combine TPC with PHY rate adaptation and compute offline an optimal rate-power combination table, and then at runtime, a wireless station determines the most energy-efficient transmission strategy for each data frame transmission by a simple table lookup. Using a similar table-driven idea, we also develop an intelligent link adaptation scheme, called ILA, for 802.11a DCF systems, which fully exploits the multiple transmission rates of the 802.11a PHY. ILA is able to react properly and quickly to the dynamically-changing network conditions and select the most appropriate transmission rate for the next transmission attempt.

Finally, we implement a new RT-WLAN device driver module, which extends the original Linux device driver for Agere ORiNOCO cards to support soft real-time communications. RT-WLAN uses separate queues for real-time and non-real-time traffic and the service preference is given to the real-time queue. By serving the real-time queue according to the EDF (Earliest-Deadline-First) policy and using an adaptive traffic smoother to regulate bursty non-real-time traffic, the desired real-time support and service differentiation among real-time sessions are achieved with RT-WLAN.

PERFORMANCE ENHANCEMENT IN WIRELESS LOCAL AREA NETWORKS

by

Daji Qiao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2004

Doctoral Committee:

Professor Kang G. Shin, Chair
Assistant Professor Achilleas Anastasopoulos
Assistant Professor Mingyan Liu
Assistant Professor Brian Noble

To my family

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my advisor, Professor Kang G. Shin, for his constant encouragement and support, invaluable guidance and advice, and enormous patience through my doctoral study. I have benefited tremendously from his vision, technical insights, and more importantly, strong pursuit of excellence. I would also like to thank Professors Achilleas Anastasopoulos, Mingyan Liu, and Brian Noble for serving on my advisory committee. Special thanks to Dr. Sunghyun Choi for giving me the opportunity to spend two summers with his group at Philips Research USA. Continuous collaborations with his group have helped me greatly in exploring and solving various challenging research problems during my study.

I am also grateful to the former and present members of the Real-Time Computing Laboratory, especially, Zhigang Chen, Chun-Ting Chou, Mohamad El-Gendy, Zonghua Gu, Hai Huang, Amit Jain, Hani Jamjoom, Songkuk Kim, Sharath Kodase, Jai-Jin Lim, Taejoon Park, Padmanabhan Pillai, John Reumann, Wei Sun, Haining Wang, Shige Wang, Jian Wu, and Xi Zhang, for their friendship, discussion, and suggestions. I am really proud to be part of all these talented colleagues. Finally, I would like to thank my family for their never-ending love, support, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER I

INTRODUCTION

1.1 IEEE 802.11 Wireless LANs

Portable computing and communication devices, such as laptops and PDAs, have become increasingly popular and more and more people rely on various wireless networks to communicate with each other. WLANs (Wireless Local-Area Networks) have received significant attention due mainly to their higher bandwidth than wide-area cellular systems as well as their use of unlicensed (free-to-use) operational bands. In recent years, the vision of pervasive ubiquitous computing where users have Internet access anytime and anywhere is being realized by the wide deployment of public WLANs, commonly known as *hotspots* [1], in public places of congregation such as conference venues, airport lounges, cafes, coffee shops, and so on. According to [13], at the end of year 2003, there were a total of 31,580 hotspots serving 1.53 million users around the world, and these numbers are expected to quadruple within the next three years.

The IEEE 802.11 [24] is the first international standard for WLANs, and almost all the WLAN products currently available in the U.S. market are designed according to this standard. The IEEE 802.11 specifies two different MAC (Medium Access Control) schemes in WLANs: the contention-based DCF (Distributed Coordination Function) and the polling-based PCF (Point Coordination Function). This dissertation research has been focusing on

the 802.11 DCF systems, which present various intriguing research challenges due to its contention nature. In fact, at present, most 802.11-compliant products only implement the mandatory DCF.

The current DCF was standardized in 1999 and was originally designed to provide simple medium access control and to work with low-speed PHYs (physical layers) operating at 1 or 2 Mbps. Now, with the emergence of new high-speed PHYs, such as 802.11b [26] supporting up to 11 Mbps and 802.11a [25] supporting up to 54 Mbps, it is essential to re-examine the original DCF and make necessary modifications to enhance the system performance. In this dissertation, we address the problem of enhancing the performance of the 802.11 DCF systems. More specifically, we first study the current systems and identify the factors that may constrain the system performance, then develop possible solutions to deal with heterogenous user locations and service requirements, to fully exploit multiple available transmit power levels and transmission rates, to effectively handle the dynamically-changing network conditions, and to support delay-sensitive and performance-intensive multimedia applications.

The main problems of the current 802.11 DCF systems can be summarized as follows, and some existing approaches to those problems will be discussed in the next section.

- The DCF uses a contention-based protocol. In a DCF system, any wireless station that wishes to transmit does so if the wireless medium is sensed free. The wireless stations are, in fact, contending for the shared wireless medium, and thus, collisions are inevitable. Due to this contention nature, the current DCF systems yield unpredictable delay characteristics and do not support prioritized transmission of real-time traffic, such as video streaming.

- The DCF resolves collisions using a slotted binary exponential backoff scheme. In a

DCF system, a wireless station is required to sense the wireless channel before any transmission attempt. If the wireless channel is busy, the wireless station has to wait until the wireless channel is cleared for a certain period, then backs off a random interval before its transmission attempt. However, in the current DCF systems, all the wireless stations use the same backoff scheme, which implies that the DCF assumes the same weight for all traffic flows. As a result, the DCF can not allocate different shares of bandwidth to different stations, i.e., it does not support the weighted fairness. Furthermore, due to the heuristic nature of the slotted binary exponential backoff scheme, the DCF system normally yields poor throughput performance, which becomes even worse as the number of contending stations increases.

- The current DCF systems do not support TPC (Transmit Power Control). Although the 802.11 PHY reports to the MAC layer the signal strength of each received frame, the current 802.11 MAC does not provide a transmit-power-reporting mechanism, which is essential to estimate the path loss condition between the transmitter and receiver stations and to apply TPC. Therefore, even when the network size is small, the transmitter station still has to use a higher power level than necessary to transmit over a relatively short distance, which results in low energy efficiency.

- The current DCF systems perform heuristic link adaptation. The 802.11 PHYs normally provide multiple transmission modes/rates by using different modulation schemes and different error correcting codes. The mechanism to select one out of multiple available transmission modes/rates at a given time is referred to as link adaptation and the effectiveness of a link adaptation scheme can affect the system performance significantly. Due to the heuristic and conservative nature of the link adaptation schemes implemented in most 802.11 devices, the current DCF systems

are likely to provide low bandwidth utilization, particularly when the wireless channel presents a high degree of variation.

1.2 Overview of Existing Approaches

1.2.1 Weighted Fairness and Efficient Bandwidth Utilization

In general, it is very difficult to maximize the channel utilization subject to the fairness constraint in an 802.11 DCF system because of the inherent tension between these two design goals. For example, maximum channel utilization may be achieved if there is only one station transmitting continuously with a zero backoff, while all the other stations are starved. Clearly, this is unfair.

Many scheduling algorithms have been proposed to achieve the weighted fairness among traffic flows that share the wireless medium [44, 48, 49, 67]. These algorithms are centralized by design, and therefore, can only be embedded into the polling-based MAC protocols. There have also been some work done on the contention-based MAC protocols. The mechanisms presented in [7, 51] attempt to provide equal shares of bandwidth to different stations and the traffic weights are implicitly assumed to be the same. The protocol in [14], although priorities have been taken into consideration when controlling the medium access, does not make fair allocation of bandwidth. The authors of [75] presented a fully-distributed scheduling algorithm that allocates bandwidth to different traffic flows in proportion to their weights. A novel mechanism was proposed in [47] to translate any pre-specified fairness model into a corresponding backoff-based contention resolution algorithm. However, the focus of both [47] and [75] was to ensure fairness through appropriate MAC-layer designs, and neither of them attempted to maximize the channel utilization. In [45], the authors dealt with both fairness and bandwidth efficiency simultaneously. They focused on maximizing the aggregate channel reuse in a multi-hop wireless network

subject to a minimum fairness guarantee, which is different from the fairness-constrained channel-utilization maximization problem addressed in this dissertation.

On the other hand, in order to improve the channel utilization in an 802.11 DCF system, many enhanced backoff schemes [8, 10] have been proposed to reduce the frame collision probability and the wasted idle backoff slots. In [8], a simple Markov chain model was presented to compute the saturation throughput of an 802.11 DCF system. From this paper, we have an important observation: *if the number of contending stations within an 802.11 DCF system is known, then by setting the probability that a station transmits in a randomly-chosen time slot to an optimal value — which is a function of the number of contending stations, the aggregate throughput can be maximized.* In other words, by simply changing the backoff rule, the aggregate throughput may not be improved. The critical factor of improving the aggregate throughput is not how to design a new backoff rule, but how to adjust the parameters of a backoff scheme so that each station can be tuned to its optimal operating point. For example, in the binary exponential backoff scheme used in the DCF, two parameters (cw_{min} and cw_{max}) need to be adjusted, while in the uniform backoff scheme proposed in [8], only one parameter (cw_{opt}) needs to be adjusted. The authors of [10] analyzed the throughput performance of an 802.11 DCF system from a different angle and proposed a similar idea to adjust the contention window size at each station based on the estimated number of active stations. Unfortunately, neither [8] nor [10] considered the weighted-fairness issue.

Notice that, due to the DCF's contention nature, there exists a theoretical limit on the aggregate throughput of an 802.11 DCF system. The authors of both [9] and [10] indicated such a theoretical upper bound of the 802.11 DCF capacity. Therefore, in order to improve the channel utilization further, a hybrid protocol of mixing contention with polling is a must, because polling utilizes the wireless channel much more efficiently when the traffic

load is heavy. The IEEE 802.11 WG (Working Group) recognized this problem and introduced a new HCF (Hybrid Coordination Function) [27, 46], which is not finalized yet and still open to amendment. The authors of [36] proposed a hybrid DQRUMA (Distributed-Queuing Request Update Multiple Access) protocol in which wireless stations need to send requests to the AP (Access Point) only for the frames that arrive at an empty buffer. For those frames that arrive at a non-empty buffer, transmission requests are piggybacked with the preceding frame transmissions, thus ensuring collision-freedom. However, this protocol simply uses a harmonic slotted ALOHA algorithm for the transmission requests. In addition, DQRUMA assumes that uplink and downlink communications are physically separate (i.e., using different frequency channels), which does not hold in an 802.11 DCF system.

1.2.2 Energy-Efficient Frame Transmissions

For wide-area cellular systems, such as IS-95 CDMA (Code-Division Multiple Access) and 3G W-CDMA (3^{rd} Generation Wide-band CDMA), TPC is critically important in order to (1) ameliorate the near-far problem, specifically, for CDMA uplink systems; (2) minimize the interference to/from other cells, i.e., co-channel interference; and (3) improve the system performance on fading channels by compensating fading dips [53]. For WLAN systems, which are mainly used in indoor home, office, and public access environments, TPC has not attracted enough attention as it was not considered as critical to success as in CDMA systems. However, since many WLAN stations such as laptops and palmtops are battery-powered, and extending the battery life and having a long station operating time are always desirable and important, applying TPC in WLAN systems to save battery energy is naturally an attractive idea. Moreover, in the multi-cell WLAN systems often found in office and public access environments, reducing the co-channel interference

via TPC could also be very beneficial since it results in better error performance in a given area.

In [41], the authors presented a scheme in which the most battery energy-efficient combination of FEC (Forward Error Correction) code and ARQ (Automatic Re-transmission reQuest) protocol is chosen and adapted over time for data transmissions, however, without considering TPC. A PARO (Power-Aware Routing Optimization) scheme was presented in [19] to achieve energy-efficient routing in multi-hop wireless networks. Before the actual data transmission, PARO exchanges RTS/CTS (Request-To-Send/Clear-To-Send) frames at the maximum power level. Then, the subsequent Data/Ack frames may be transmitted at lower power levels to save energy. Another similar power control scheme was proposed in [3], where the transmit power level of the data frame is dynamically adjusted with the help of an enhanced RTS/CTS mechanism that supports power control loop. In [16], the authors first showed the strong correlation among the packet size, the transmit power, and the energy consumption for 802.11 devices. Then, based on a theoretical analysis, they proposed a power control scheme to save energy by choosing the optimal transmit power levels for different packet sizes. One common problem of the above power control schemes is that none of them considered PHY rate adaptation, which is very effective in saving energy and should be considered in conjunction with TPC.

The authors of [18, 52] proposed a lazy scheduling algorithm and an iterative Move-Right algorithm, respectively, to minimize the energy used to transmit packets from a wireless station to a single receiver or to multiple receivers. The key idea is to transmit packets over longer periods with lower transmit power as long as the deadline constraint is met. However, they assumed that the wireless channel is time-invariant, and focused on devising optimal schedules for a wireless station to transmit multiple packets (sharing the same deadline constraint), which is different from the per-frame-based energy-efficient

transmission problem addressed in this dissertation.

1.2.3 Intelligent Link Adaptation

Commercial 802.11 products have their own proprietary link adaptation schemes. The ARF (AutoRate Fallback) protocol [34], which is used in Lucent Technologies' WaveLAN-II networking devices, is one of the few that are available to public. It alternates between 1 and 2 Mbps 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 next 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 [23] presented an RBAR (Receiver-Based Auto-Rate) protocol based on the RTS/CTS mechanism by modifying the 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 in the 802.11 standard, such as the Data, RTS, CTS frame formats, and the PHY header, it may not be practically useful. In [66], two link adaptation schemes were proposed for the GPRS (General Packet

Radio Services) development of GSM (Global System for Mobile communication). One is based on the estimate of the C/I (Carrier to Interference ratio), and the other is based on the observation of the block error rate. HIPERLAN/2 (HIgh PErformance Radio Local Area Network Type II) [37] is another wireless broadband access system that has been specified by ETSI (European Telecommunications Standards Institute) 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 [43] 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 [21] 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 PER (Packet Error Rate). 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, a 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 selection, 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 re-transmissions, even when the wireless channel condition changes.

1.2.4 Supporting Real-Time Traffic

A number of approaches have been proposed to support prioritized transmission of real-time traffic in 802.11 systems. The authors of [79] proposed a prioritized MAC scheme, by modifying the current 802.11 standard, which allows real-time control traffic to co-exist with multimedia and batch traffic. In [35], the authors presented a distributed priority scheduling technique that piggybacks the priority tag of a station's head-of-line packet onto handshake and data packets. By monitoring transmitted packets, each station maintains a scheduling table which is used to assess the station's priority level relative to other stations. The existing 802.11 backoff scheme is then modified to incorporate this scheduling table, so as to approximate the ideal schedule. However, both approaches require changes in the actual NIC (Network Interface Card) firmware, since the MAC functions are normally hard-coded in the NIC.

The IEEE 802.11 TGe (Task Group E) has been working on the new 802.11e standard [27, 46], which defines enhancements to the current 802.11 MAC to support applications with QoS (Quality of Service) requirements. One of the new mechanisms is called the EDCF (Enhanced Distributed Coordination Function), which realizes the QoS support by introducing the concept of TCs (Traffic Categories). A single station may implement up to eight transmission queues whose service priorities are determined by different queue management parameters. Each queue corresponds to a certain TC. Before the new 802.11e standard is finalized by the IEEE standardization committee and introduced to the market, the DCF-mode 802.11-compliant devices are expected to continue their dominance of the market. Actually, even after the new 802.11e devices are introduced to the market, there will still be many legacy 802.11 devices deployed in various sectors. In order to support real-time applications within current 802.11 systems, appropriate real-time extensions are essential.

The authors of [71] implemented a Wireless Rether (Real-Time Ethernet) protocol that provides QoS guarantees on current 802.11 systems. It adopts a centralized token passing architecture and uses sequential Token-Ack message exchanges between WRS (Wireless Rether Server) and WRCs (Wireless Rether Clients) to ensure collision-freedom on the wireless channel.

1.3 Main Contributions

The objective of this dissertation is to provide various schemes to enhance the performance of the 802.11 DCF systems. The mechanisms include a weighted-fair and bandwidth-efficient enhancement to the 802.11 DCF [62, 64, 65], an optimal low-energy transmission strategy for 802.11a/h [58, 59], an intelligent link adaptation scheme for 802.11a [56, 63], and a soft real-time extension to the ORiNOCO Linux device driver [32]. The main contributions of this dissertation are as follows.

- **A weighted-fair and bandwidth-efficient enhancement to the 802.11 DCF:** we address the problem of maximizing the channel utilization subject to the weighted fairness constraint in the 802.11 DCF systems, and propose a simple enhancement to the 802.11 DCF, called the WB-DCF, particularly for data communications. The key ideas of the WB-DCF are that (1) a new polling mode is introduced, in addition to the contention mode used in the DCF, to reduce the number of contending stations; (2) the contention window size for each wireless station is carefully determined according to runtime load estimation, so as to achieve a low frame collision probability and maximize the channel utilization; and (3) the relative weights of traffic flows are taken into consideration in the contention window selection and the polling scheme, so as to provide the weighted fairness.

- **An optimal low-energy transmission strategy for 802.11a/h:** we address the problem of energy-efficient frame transmission in the emerging 802.11a/h systems that will provide a structured means to support intelligent TPC. Based on a rigorous analysis of the relationship among different radio ranges and TPC's effects on the interference, we develop a novel scheme, called MiSer, that minimizes the communication energy consumption in 802.11a/h DCF systems by combining TPC with PHY rate adaptation. The key idea is to compute offline an optimal rate-power combination table, then at runtime, a wireless station determines the most energy-efficient transmission strategy for each data frame transmission by a simple table lookup.

- **An intelligent link adaptation scheme for 802.11a:** to fully exploit the multiple transmission rates of the 802.11a PHY, we present a generic method to analyze the goodput performance of an 802.11a DCF system, then based on the theoretical analysis, we develop a novel link adaptation scheme, called ILA, for 802.11a DCF systems. Similar to MiSer, ILA is a table-driven approach that requires each wireless station to establish a best PHY mode table before the communication starts and simplifies the runtime execution significantly to table lookups.

- **A soft real-time extension to the ORiNOCO Linux device driver:** to support soft real-time communications in the 802.11 systems using the popular Agere ORiNOCO cards, we implement a new RT-WLAN device driver module that extends the original ORiNOCO Linux device driver. RT-WLAN uses separate queues for real-time and non-real-time traffic, and the service preference is given to the real-time queue. Besides, an adaptive traffic smoother is implemented in RT-WLAN to regulate bursty non-real-time traffic before they are injected into the network, thus giving

higher priority to in-progress real-time transmissions.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter II introduces the 802.11 DCF, presents the fairness and throughput analyses of an 802.11 DCF system, and describes the details of the WB-DCF. The performance of the WB-DCF is evaluated and discussed using simulation.

In Chapter III, following the interference analysis, an enhanced RTS-CTS(strong)-Data(TPC)-Ack mechanism is proposed and justified to accommodate intelligent TPC in 802.11a/h DCF systems. A generic energy consumption model of the 802.11 device and the basic energy consumption computations in an 802.11a/h DCF system are also presented. Then, MiSer is detailed and the related implementation issues are discussed. Based on in-depth simulation for various scenarios, the performance of MiSer is evaluated and compared with that of other testing schemes to show its superiority.

Chapter IV presents the effective goodput analysis of an 802.11a DCF system and describes the details of ILA. Furthermore, a detailed example is given to show how to establish the best PHY mode table in ILA. Based on the simulation results, the performance of ILA is evaluated and discussed.

Chapter V presents the implementation details of the new RT-WLAN device driver module: the user interface, the real-time queue and its EDF (Earliest-Deadline-First) policy, the non-real-time queue and its adaptive traffic smoother, and the packet scheduler. Using experiments, the performance of RT-WLAN is evaluated.

Finally, Chapter VI concludes this dissertation with a summary of the main contributions, and describes possible future research directions.

CHAPTER II

A WEIGHTED-FAIR AND BANDWIDTH-EFFICIENT ENHANCEMENT TO THE 802.11 DCF

2.1 Introduction

In general, it is very difficult to maximize the channel utilization subject to the fairness constraint in an 802.11 DCF system because of the inherent tension between these two design goals. For example, maximum channel utilization may be achieved if there is only one station transmitting continuously with a zero backoff, while all the other stations are starved. Clearly, this is unfair. In this chapter, we study this problem and propose a simple weighted-fair and bandwidth-efficient enhancement to the DCF, particularly for data communications.

The WLAN architecture we are interested in is the most widely-deployed infrastructure DCF system, which can be often found in in-door office, home, or public access environments. It includes an AP (Access Point) to provide both the connection to the wired network, if any, and the local relaying function within the system, meaning that, if a wireless station wants to communicate with another station, the frames must be sent first to the AP, and then from the AP to the destination. Since all the downlink (AP-to-station) transmissions are centrally-controlled by the AP, we are more concerned about how to achieve the weighted fairness and maximize the channel utilization when the wireless sta-

tions contend for uplink (station-to-AP) transmissions.

The ideal weighted fairness is defined as follows. Assume that there are n ($\geqslant 1$) different traffic classes each characterized by a positive weight. Let ϕ_i denote the weight associated with class$-i$ traffic ($1 \leqslant i \leqslant n$), and without loss of generality, let's assume that $1 = \phi_1 > \phi_2 > \cdots > \phi_n > 0$. Further, assume that each wireless station carries only one traffic flow.[1] Let f_i denote the set of stations carrying class$-i$ traffic, and let $w_i^s(t_b, t_e)$ be the amount of class$-i$ traffic transmitted by station $s \in f_i$ during the time interval $[t_b, t_e]$. To be fair to all the traffic flows, it requires, regardless of where and how small the interval $[t_b, t_e]$ is,

$$\forall i, j \in \{1, \cdots, n\},\ \forall s \in f_i,\ \forall s' \in f_j,\ \ \frac{w_i^s(t_b, t_e)}{\phi_i} = \frac{w_j^{s'}(t_b, t_e)}{\phi_j}. \tag{2.1}$$

However, the ideal weighted fairness cannot be accurately achieved in practice, because data transmitted on a real network are packetized. Instead, we define a new weighted-fairness objective function [62] for data communications in 802.11 DCF systems.

As specified in the 802.11 standard [24], each data packet generated by the higher layer is fragmented further into smaller MAC frames for transmission. Therefore, it is reasonable to assume that *each data traffic flow has the same MAC frame size*. Let SU_s be the probability that a MAC frame transmission is successful and is from station s. If Eq. (2.2) holds, all the traffic flows would share the wireless medium fairly in a probabilistic sense, and we claim that the weighted fairness intended for data communications in an 802.11 DCF system is achieved:

$$\forall i, j \in \{1, \cdots, n\},\ \forall s \in f_i,\ \forall s' \in f_j,\ \ \frac{SU_s}{\phi_i} = \frac{SU_{s'}}{\phi_j}. \tag{2.2}$$

On the other hand, due to the contention nature of the DCF, there exists a theoretical limit on the aggregate throughput of an 802.11 DCF system. The authors of both [9]

[1] If a wireless station carries multiple traffic flows, the fairness among those traffic flows can be easily guaranteed by some in-station packet scheduling algorithms, e.g., weighted round-robin.

and [10] indicated such a theoretical upper bound of the 802.11 DCF capacity. Therefore, in order to improve the channel utilization over this limit, a hybrid protocol by mixing contention with polling is a must, because polling utilizes the wireless channel much more efficiently when the traffic load is heavy. The IEEE 802.11 WG (Working Group) recognized this problem and included the polling-based PCF as part of the standard. The authors of [11] showed that the PCF can indeed achieve a higher maximum throughput than the DCF when the system is overloaded. However, the original PCF has many open problems and the new HCF (Hybrid Coordination Function) [27, 46], which is being proposed to replace the PCF, is not finalized yet.

In this chapter, we propose a simple enhancement to the DCF, called the *WB-DCF* (Weighted-fair and Bandwidth-efficient DCF) [65], that achieves both the weighted fairness and the efficient channel utilization for data communications in an 802.11 DCF system. We are not claiming that the proposed enhancement can, or will, replace the HCF. It is just a practical solution that is simple, effective, and easy to deploy with the available DCF devices. The key ideas of the WB-DCF are:

- A new polling mode (P-mode) is introduced for frame transmissions in addition to the conventional contention mode (C-mode) used in the DCF. In the WB-DCF, only the head-of-line frame of each data frame burst — in reality, data frames are often generated in bursts and queued at the wireless station before they are transmitted — is transmitted in C-mode while all the follow-on frames are transmitted in P-mode. This way, the number of contending stations is reduced.

- The contention window size for each wireless station is carefully determined according to the runtime estimation of the number of contending stations, so as to achieve a low frame collision probability and maximize the channel utilization.

- The weighted fairness among P-mode and C-mode frame transmissions are achieved, respectively, by the weighted-round-robin scheduling policy (at the AP) and the weighted contention window selection scheme (at each wireless station).

The rest of this chapter is organized as follows. Section 2.2 describes the CSMA/CA protocol and the backoff behavior of the 802.11 DCF. Fairness and throughput analyses of an 802.11 DCF system are presented in Section 2.3. Section 2.4 describes the details of the WB-DCF and Section 2.5 presents and discusses the simulation results. Finally, this chapter concludes with Section 2.6.

2.2 System Overview

2.2.1 CSMA/CA of the 802.11 DCF

The DCF [24], as the basic access mechanism of the 802.11 MAC, achieves automatic medium sharing among contending stations via the use of CSMA/CA (Carrier-Sense Multiple Access with Collision Avoidance). 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. CSMA/CA requires a minimum specified gap/space between contiguous frame transmissions. A station will ensure that the medium has been idle for the specified inter-frame interval before attempting to transmit.

The DIFS (Distributed Inter-Frame Space) is used by stations operating under the DCF to transmit data frames. A DCF station is allowed to transmit only if its carrier-sense mechanism determines that the medium has been idle for at least DIFS time. Moreover, in order to reduce the collision probability among multiple stations accessing the medium, a DCF station is required to select a random backoff interval after deferral, or prior to attempting to transmit another frame after a successful transmission.

One important characteristic of the 802.11 MAC is that an Ack frame will be sent by the receiver upon successful reception of a data frame. It is only after receiving the Ack frame correctly that the transmitter assumes successful delivery of the corresponding data frame. The SIFS (Short Inter-Frame Space), which is smaller than the DIFS, is the time interval used between transmissions within a frame exchange sequence, i.e., a two-way Data-Ack handshake or a four-way RTS-CTS-Data-Ack handshake. Using this small gap prevents other stations — which are required to wait for the medium to be idle for a longer gap (i.e., at least DIFS time) — from attempting to use the medium, thus giving priority to completion of the in-progress frame exchange.

2.2.2 Backoff Behavior of the 802.11 DCF

The DCF adopts a *slotted binary exponential backoff* scheme to select the random backoff interval (in the unit of *tSlotTime*). This random number is drawn from a uniform distribution over the interval [0, cw], where cw is the contention window size and its initial value is cw_{min}. In the case of an unsuccessful RTS (Data) transmission, the backoff procedure will begin at the end of the CTS (Ack) timeout, and cw is updated to $[2 \times (cw + 1) - 1]$. Once cw reaches cw_{max}, it will remain at this value until it is reset to cw_{min}. In the case of a successful data transmission, the backoff procedure will begin at DIFS time after receiving the Ack frame, and the cw value is reset to cw_{min} before the random backoff interval is selected. Note, however, that the cw value is not reset after a successful RTS transmission. Each station decrements its backoff counter every *tSlotTime* interval after the wireless medium is sensed to be idle for DIFS time. If the counter has not reached zero and one of the other stations starts transmitting, the station freezes its counter. When the counter finally reaches zero, the station starts its transmission.

Fig. 2.1 illustrates such an operation of decrementing the backoff counter when the

four-way handshake is used. After the successful transmission and acknowledgment of frame A1, station A waits for DIFS time and selects a backoff interval equal to six, before attempting to transmit the next frame A2. Assume that station B selects a smaller backoff interval equal to three after it has sensed the medium to be idle for DIFS time. Since the backoff counter of station B reaches zero before that of station A, frame B1 is transmitted after exchanging the RTS/CTS frames. As a result of the medium sensed busy, the backoff counter of station A is frozen at three, and decrements again after the medium is sensed idle for DIFS time.

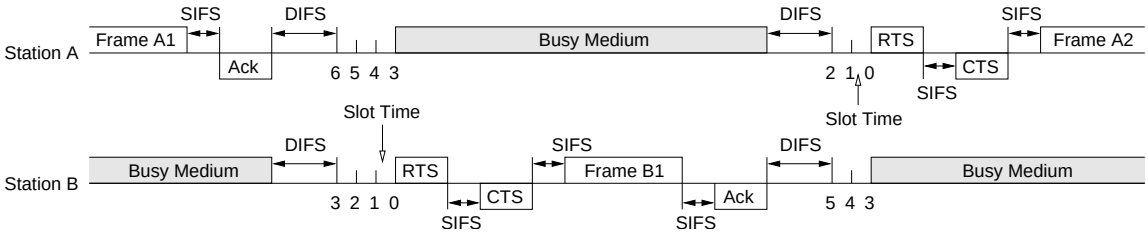


Figure 2.1: An example of backoff decrements with four-way handshake

2.3 Fairness and Throughput Analyses

Before delving into the the details of the WB-DCF, we will first analyze the fairness and throughput performances of an 802.11 DCF system. Since we are only interested in how the DCF should be enhanced to provide the weighted fairness and to alleviate the inevitable frame collisions caused by multiple stations contending for the shared wireless medium, we assume ideal channel conditions (i.e., no transmission errors and no hidden nodes) in the following analyses.

2.3.1 Fairness Analysis

Consider the scenario when there are n greedy contending stations — stations that always have frames to transmit — in the network. Assume that one of the contending

stations, $u \in f_i$, uses a contention window cw_u to access the wireless medium,[2] and initially, its backoff value $b_u(t)$ is uniformly selected from the range [0, cw_u]. As illustrated in Fig. 2.1, $b_u(t)$ is decremented at the end of each time slot, which could be either an idle period of length *tSlotTime*, or a busy period due to a collision, or a busy period due to a successful frame transmission. Note that t is a discrete time point corresponding to the end of a time slot. Then, as indicated in [9], the stochastic process $b_u(t)$ can be modeled by the following discrete-time Markov chain:

$$P\{b_u(t+1)=k\} = \begin{cases} P\{b_u(t)=k+1\} + \frac{P\{b_u(t)=0\}}{cw_u+1} & \text{for } k = 0, \cdots, cw_u - 1, \\ \frac{P\{b_u(t)=0\}}{cw_u+1} & \text{for } k = cw_u, \end{cases} \tag{2.3}$$

where the term $P\{b_u(t)=k+1\}$ corresponds to decrementing the backoff value at the end of each time slot. The term $\frac{P\{b_u(t)=0\}}{cw_u+1}$ accounts for the fact that, after a frame transmission attempt, the new backoff value is uniformly selected from the range [0, cw_u], regardless whether the frame transmission was successful or not. The steady state probabilities of this Markov chain are:

$$\lim_{t\to\infty} P\{b_u(t)=k\} = \frac{2\cdot(cw_u-k+1)}{(cw_u+1)\cdot(cw_u+2)}. \tag{2.4}$$

Refer to [9] for more details about the Markov chain.[3] Recall that, when the backoff counter reaches zero, the station starts its transmission. Therefore, the probability that station u transmits in a randomly-chosen time slot is

$$p_u = \lim_{t\to\infty} P\{b_u(t)=0\} = \frac{2}{cw_u+2}. \tag{2.5}$$

[2]The WB-DCF adopts a uniform backoff scheme, meaning that station u will use this contention window cw_u to select the backoff intervals for all of its frame transmission attempts.

[3]The following fairness and throughput analyses are novel, although they are based on the discrete-time Markov chain model described in [9].

The probability that at least one station attempts to transmit in a slot, or equivalently, the probability that a slot is not idle, is given by

$$P_{tr} = 1 - \prod_{\forall v}(1 - p_v), \tag{2.6}$$

and the probability that a transmission is successful and is from station u can be calculated as

$$SU_u = p_u \cdot \prod_{v \neq u}(1 - p_v). \tag{2.7}$$

To achieve the desired fairness among the stations carrying the same traffic class, say, class i, we must have

$$\begin{aligned} & \forall u, v \in f_i, \quad \frac{SU_u}{\phi_i} = \frac{SU_v}{\phi_i} \\ \Longleftrightarrow \quad & \forall u, v \in f_i, \quad p_u(1 - p_v) = p_v(1 - p_u) \\ \Longleftrightarrow \quad & \forall u, v \in f_i, \quad p_u = p_v \\ \Longleftrightarrow \quad & \forall u, v \in f_i, \quad cw_u = cw_v. \end{aligned} \tag{2.8}$$

The interpretation of Eq. (2.8) is trivial: in order to be fair to the traffic flows of the same priority, the source stations should use the same contention window size. Now, we use p_i and SU_i to denote the probability that a station carrying class$-i$ traffic transmits in a time slot, and the probability that a transmission is successful and is from a station carrying class$-i$ traffic, respectively. We can rewrite Eqs. (2.6) and (2.7) as

$$P_{tr} = 1 - \prod_{i=1}^{n}(1 - p_i)^{|f_i|}, \tag{2.9}$$

and

$$SU_i = p_i \cdot (1 - p_i)^{|f_i|-1} \cdot \prod_{j \neq i}(1 - p_j)^{|f_j|}. \tag{2.10}$$

Similarly, to achieve the desired fairness among the stations carrying different traffic classes, we must have

$$\forall i, j \in \{1, \cdots n\}, \quad \frac{SU_i}{\phi_i} = \frac{SU_j}{\phi_j}$$

$$\begin{aligned}
&\iff \quad \forall i,j \in \{1,\cdots n\}, \qquad \frac{p_i(1-p_j)}{\phi_i} = \frac{p_j(1-p_i)}{\phi_j} \\
&\iff \quad \forall j \in \{2,\cdots n\}, \qquad p_j = \frac{p_1}{\frac{1-p_1}{\phi_j}+p_1} \\
&\iff \quad \forall j \in \{2,\cdots n\}, \qquad cw_j = \frac{cw_1}{\phi_j}.
\end{aligned} \tag{2.11}$$

As expected, if a wireless station carries a lower-priority traffic flow, it should use a larger contention window to access the wireless medium, thus favoring higher-weight traffic flows.

Based on the above analysis, we draw the following conclusion: *if the contention window sizes of all the stations are carefully selected to satisfy both Eqs. (2.8) and (2.11), the desired weighted fairness among traffic flows can be achieved.*

2.3.2 Throughput Analysis

Since the probability that a transmission is successful is given by

$$SU = \sum_{i=1}^{n} |f_i| \cdot p_i \cdot (1-p_i)^{|f_i|-1} \cdot \prod_{j \neq i} (1-p_j)^{|f_j|}, \tag{2.12}$$

and $(1-P_{tr})$ is the probability of an idle slot, the aggregate throughput can be calculated as

$$Th = \frac{SU \cdot \textit{aPayload}}{SU \cdot \ell_{succ} + (P_{tr} - SU) \cdot \ell_{coll} + (1-P_{tr}) \cdot \textit{tSlotTime}}, \tag{2.13}$$

where *aPayload* is the data payload length, ℓ_{succ} is the length of a successful frame transmission, ℓ_{coll} is the collision length, and *tSlotTime* is the length of an idle time slot. Eq. (2.13) can be rewritten as

$$Th = \frac{\textit{aPayload}}{\ell_{succ} - \ell_{coll} + \frac{P_{tr} \cdot \ell_{coll} + (1-P_{tr}) \cdot \textit{tSlotTime}}{SU}}. \tag{2.14}$$

Since *aPayload*, ℓ_{succ}, ℓ_{coll}, and *tSlotTime* are constant for all the stations, maximization of the aggregate throughput is equivalent to maximization of

$$Th' \quad = \quad \frac{SU}{P_{tr} \cdot \ell_{coll} + (1-P_{tr}) \cdot \textit{tSlotTime}}$$

$$= \frac{\sum_{i=1}^{n} |f_i| \cdot p_i \cdot (1-p_i)^{|f_i|-1} \cdot \prod_{j \neq i} (1-p_j)^{|f_j|}}{\ell_{coll} - \prod_{i=1}^{n} (1-p_i)^{|f_i|} \cdot (\ell_{coll} - \textit{tSlotTime})}. \quad (2.15)$$

Note that the collision lengths in the cases with and without RTS/CTS support can be calculated [56] by

$$\ell_{coll(basic)} = \textit{tPHYOverhead} + \textit{tPropDelay} + \textit{tDIFSTime} + \left\lceil \frac{\textit{aMACOverhead} + \textit{aPayload} + 2.75}{\textit{BpS}} \right\rceil \cdot \textit{tSymbol}, \quad (2.16)$$

and

$$\ell_{coll(rts/cts)} = \textit{tPHYOverhead} + \textit{tPropDelay} + \textit{tDIFSTime} + \left\lceil \frac{\textit{aRTSLength} + 2.75}{\textit{BpS}} \right\rceil \cdot \textit{tSymbol}, \quad (2.17)$$

respectively. It is clear from Eqs. (2.15) and (2.5) that Th' depends on cw_j via p_j ($j = 1, \cdots, n$). Since for each $j \in \{2, \cdots, n\}$, p_j is a function of p_1, we can first solve the following equation to get the optimal value of p_1 to maximize the aggregate throughput:

$$\frac{d\, Th'}{d\, p_1} = 0, \quad (2.18)$$

and then apply Eqs. (2.5) and (2.11) to obtain the optimal values of cw_1 and cw_j's. However, it is difficult to solve Eq. (2.18) directly to get a closed-form expression for the optimal p_1 when $n \geqslant 3$. Therefore, we instead give an approximate solution.

Let's first look at the simplest case where there are only two stations in the wireless network and each station carries a traffic flow with a different weight. Now, Eq. (2.15) is simplified to

$$Th'_{(simple)} = \frac{p_1(1-p_2) + p_2(1-p_1)}{\ell_{coll} - (1-p_1)(1-p_2) \cdot (\ell_{coll} - \textit{tSlotTime})}. \quad (2.19)$$

The optimal p_1 can then be derived as

$$p^*_{1(simple)} = \frac{1}{1 + \sqrt{\frac{\ell_{coll}}{\textit{tSlotTime}} \cdot \phi_2}}, \quad (2.20)$$

and the corresponding optimal contention window size is

$$cw^*_{1(simple)} = 2\sqrt{\frac{\ell_{coll}}{\textit{tSlotTime}} \cdot \phi_2}. \tag{2.21}$$

In particular, if $\phi_2 = 0$, we have $p_1^* = 1$ and $cw_1^* = 0$, which means that the only active station in the network transmits continuously without any backoff, thus maximizing the channel utilization.

Now, consider the general case with n ($\geqslant 3$) stations in the network. Notice that, under the assumption of

$$\forall j \in \{1, 2, \cdots, n\}, \quad cw_1 \gg |f_j|\phi_j, \tag{2.22}$$

we have

$$|f_j|p_j = \frac{2|f_j|}{cw_j + 2} \approx \frac{2|f_j|\phi_j}{cw_1} \ll 1, \tag{2.23}$$

so we can make the following approximations:

$$\frac{p_b(1-p_a)}{\sum_{j=2}^{n} \frac{|f_j|}{|f_1|}\phi_j} \approx \frac{p_a(1-p_b)}{\phi_1}, \tag{2.24}$$

and

$$Th' \approx \frac{p_a(1-p_b) + p_b(1-p_a)}{\ell_{coll} - (1-p_a)(1-p_b) \cdot (\ell_{coll} - \textit{tSlotTime})}, \tag{2.25}$$

where

$$\begin{cases} p_a \equiv |f_1|p_1, \\ \\ p_b \equiv \sum_{j=2}^{n} |f_j|p_j. \end{cases} \tag{2.26}$$

Comparing Eq. (2.25) with Eq. (2.19), we can get an approximate optimal value of p_1:

$$p_1^* \approx \frac{1}{|f_1| + \sqrt{\frac{\ell_{coll}}{\textit{tSlotTime}} \cdot |f_1| \cdot \left(\sum_{j=2}^{n} |f_j|\phi_j\right)}}. \tag{2.27}$$

As a result, the optimal contention window size for the stations carrying class-1 traffic can be approximated as:

$$cw^*_{1(I)} \approx 2|f_1| + 2\sqrt{\frac{\ell_{coll}}{\textit{tSlotTime}} \cdot |f_1| \cdot \left(\sum_{j=2}^{n} |f_j|\phi_j\right)} - 2. \tag{2.28}$$

We call Eq. (2.28) the *Approximation Function I (AF-I)*. Recall that this approximate solution is obtained under the assumption of "$cw_1 \gg |f_j|\phi_j$". Suppose that we define "$\gg$" as "10 times greater than", then for $cw^*_{1(I)}$ to be a good approximation, we require

$$cw^*_{1(I)} \gg |f_1|, \tag{2.29}$$

or equivalently,

$$\sum_{j=2}^{n} |f_j|\phi_j \geqslant \frac{16\ \textit{tSlotTime}}{\ell_{coll}} \cdot |f_1|. \tag{2.30}$$

In other words, when $|f_j|$ or ϕ_j is small, AF-I is expected to not perform well. Besides, by carefully examining the above approximation process, we can see that $cw^*_{1(I)}$ is actually obtained by replacing the contending stations with two virtual stations with probabilities $|f_1|p_1$ and $\sum_{j=2}^{n} |f_j|p_j$, respectively, to attempt frame transmissions in a randomly-chosen time slot. As a result, this approximation process always tends to select smaller contention window sizes than the actual optimal values. Therefore, any other approximate value that is less than $cw^*_{1(I)}$ is even farther away from the actual optimal value, and hence, is not acceptable.

On the other hand, when $|f_j| = 0$ for each $j \in \{2, \cdots, n\}$, a close approximation to the optimal value of cw_1 is given in [9] as:

$$cw^*_{1(II)} \approx |f_1| \cdot \sqrt{\frac{2\ \ell_{coll}}{\textit{tSlotTime}}}, \tag{2.31}$$

and we call it the *Approximation Function II (AF-II)*. Therefore, when the traffic on the wireless medium is dominated by class-1 traffic, we may use AF-II instead, but we require

$$cw^*_{1(II)} > cw^*_{1(I)}, \tag{2.32}$$

or equivalently,

$$\sum_{j=2}^{n} |f_j|\phi_j < \left[\frac{1}{2} + \frac{\mathit{tSlotTime}}{\ell_{coll}} - \sqrt{\frac{2\ \mathit{tSlotTime}}{\ell_{coll}}}\right] \cdot |f_1|. \tag{2.33}$$

Finally, we combine AF-I and AF-II and propose the following approximation to the optimal value of cw_1, which depends on the traffic scenario on the wireless medium:

$$cw_1^* \approx \begin{cases} 2|f_1| + 2\sqrt{\frac{\ell_{coll}}{\mathit{tSlotTime}} \cdot |f_1| \cdot \left(\sum_{j=2}^{n} |f_j|\phi_j\right)} - 2, & \text{if } \sum_{j=2}^{n} |f_j|\phi_j \geqslant \frac{|f_1|}{r}, \\ |f_1| \cdot \sqrt{\frac{2\ \ell_{coll}}{\mathit{tSlotTime}}}, & \text{otherwise,} \end{cases} \tag{2.34}$$

where

$$r = \max\left(\frac{\ell_{coll}}{16\ \mathit{tSlotTime}},\ \frac{1}{\frac{1}{2} + \frac{\mathit{tSlotTime}}{\ell_{coll}} - \sqrt{\frac{2\ \mathit{tSlotTime}}{\ell_{coll}}}}\right),$$

and we call it the *Final Approximation Function (FAF)*. Consequently, the optimal contention window sizes for other stations are

$$\forall j \in \{1, 2, \cdots, n\}, \quad cw_j^* = \frac{cw_1^*}{\phi_j}. \tag{2.35}$$

Based on the above analysis, we draw the following conclusion: *if the contention window sizes of all the stations are selected according to Eqs. (2.34) and (2.35), the desired weighted fairness among traffic flows is achieved while the channel utilization is maximized.*

2.3.3 Numerical Results

Our approximation scheme is evaluated numerically. Fig. 2.2 shows the throughput performance when there are only two traffic classes on the wireless medium without RTS/CTS support. Assume that there are a total of 50 stations in an 802.11 DCF system, and each station carries a traffic flow belonging to either of the two traffic classes. Let the values along the X-axis represent $|f_1|$, the number of stations carrying class-1

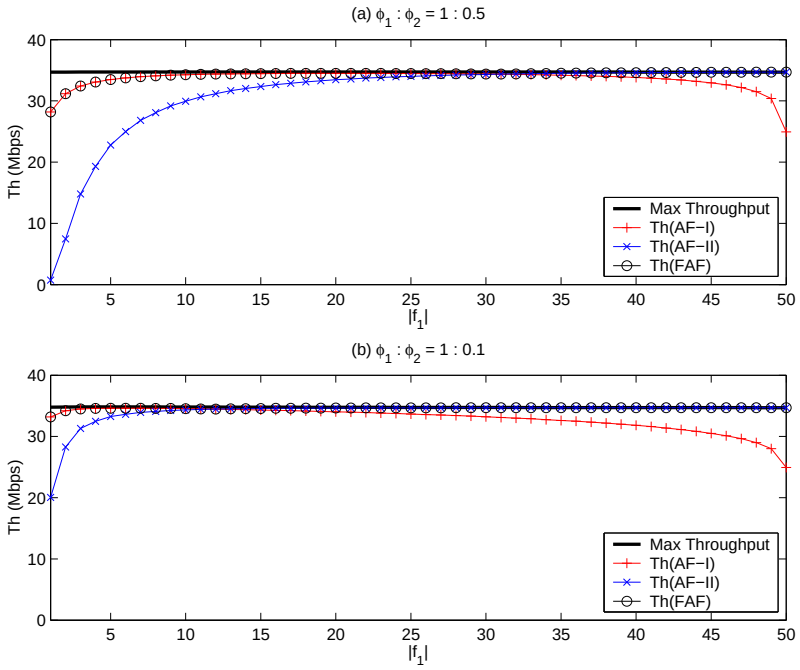


Figure 2.2: Throughput performance under the 2-traffic-class scenario w/o RTS/CTS

traffic. The thick solid lines in the figure represent the maximum achievable throughputs. As expected, AF-I (cross points in the figure) is not well-behaved when $|f_1|$ is large (hence $|f_2|$ is small), because the "$cw_1 \gg |f_1|$" assumption does not hold in this case. Furthermore, by comparing the cross points in the two sub-figures, we can see that the throughput performance of AF-I is even worse with a smaller ϕ_2 value. Fortunately, due to the capability of adaptively switching from AF-I to AF-II ($\times$ points in the figure), FAF (circle points in the figure) works well in this range. However, when $|f_1|$ is small (e.g., ≤ 5 in this figure), noticeable gaps can be observed between our approximated maximum aggregate throughputs and the actual values. The reason for this is that, when $|f_1|$ is small, the "$cw_1 \gg |f_2|\phi_2$" assumption does not hold. In general, if $|f_1|$ is small, then since cw_1 is not significantly larger than $\sum_{j=2}^{n} |f_j|\phi_j$ (based on Eq. (2.34)), the "$cw_1 \gg |f_j|\phi_j$" assumption must not hold for some $j \in \{2, \cdots, n\}$. Therefore, similar gaps can be observed under all the traffic scenarios. See Fig. 2.3, for example, where there are three traffic

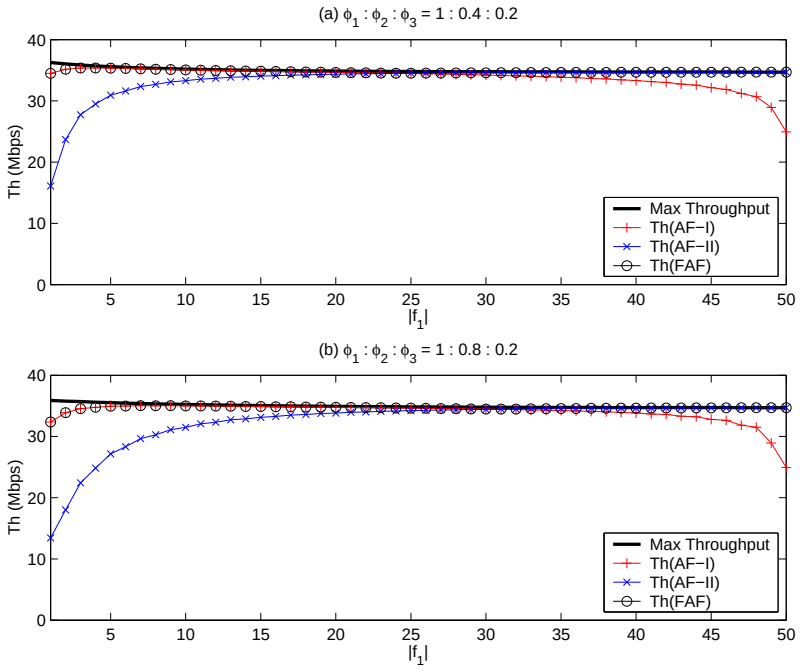


Figure 2.3: Throughput performance under the 3-traffic-class scenario w/o RTS/CTS

classes on the wireless medium and equal number of stations carry class-2 and class-3 traffic, respectively.

Figs. 2.4 and 2.5 show the results for both traffic scenarios, respectively, when the four-way handshake is used. Notice that, since the introduction of the RTS/CTS exchange significantly reduces the collision length, i.e., $\ell_{coll(rts/cts)} \ll \ell_{coll(basic)}$, the deviation of our approximate contention window sizes won't result in as big gaps as those without RTS/CTS support. We can see that even AF-I itself performs well for most network configurations.[4]

2.4 WB-DCF

In this section, we present the details of the WB-DCF, the proposed weighted-fair and bandwidth-efficient enhancement to the DCF. The key ideas are that (1) in order to re-

[4]In Figs. 2.4 and 2.5, most cross points (AF-I) and circle points (FAF) are overlapped with each other and are close to the solid lines (maximum achievable throughputs).

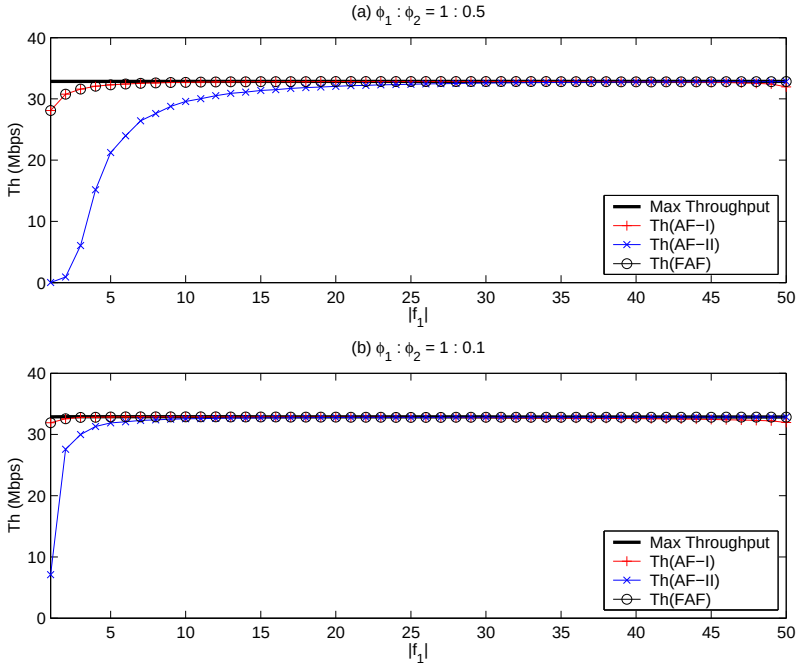


Figure 2.4: Throughput performance under the 2-traffic-class scenario with RTS/CTS

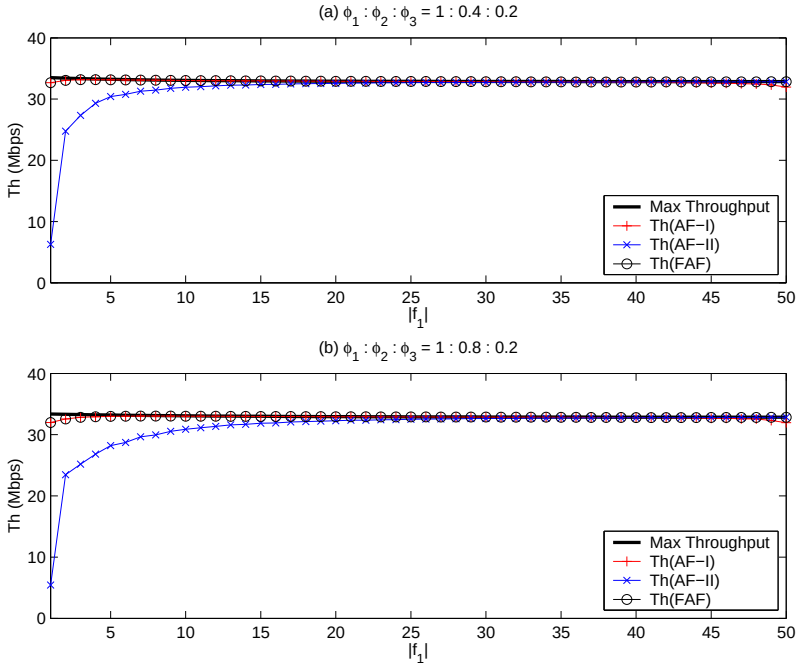


Figure 2.5: Throughput performance under the 3-traffic-class scenario with RTS/CTS

duce the number of contending stations, a new polling mode (P-mode) is introduced for frame transmissions in addition to the conventional contention mode (C-mode) used in the

DCF; (2) RTS/CTS frames are exchanged before each C-mode data transmission attempt to shorten the collision length; (3) the contention window size for each wireless station is carefully determined according to the runtime estimation of the number of contending stations, so as to achieve a low frame collision probability and maximize the channel utilization; and (4) the weighted fairness among P-mode and C-mode frame transmissions are guaranteed, respectively, by the weighted-round-robin scheduling policy and the weighted contention window selection scheme.

2.4.1 Part I: Introducing P-mode Frame Transmissions

Burstiness is one of the common features seen in data traffic flows, i.e., data frames are often generated in bursts by the source application and queued at the wireless station until they are transmitted. Hence, we classify the data frames into the following two categories: if the queue is empty when a data frame is generated, the frame is a *head-of-line* frame; otherwise, it is a *follow-on* frame.

In an 802.11 DCF system, a wireless station is required to contend for every frame transmission, regardless whether it is a head-of-line frame or a follow-on frame. As a result, when the system is heavily-loaded, the network suffers a high degree of contention for the wireless medium, and consequently, presents poor throughput performance. One natural idea to alleviate this problem is to reduce the number of contending stations. In order to do so, the WB-DCF allows a wireless station to piggyback its local queuing status in the preceding frame transmissions, and therefore, the follow-on frames may be transmitted using a poll-response-like scheme (P-mode) in comparison to the contention mode (C-mode) used in the DCF.

The WB-DCF defines new POLL frames to support P-mode frame transmissions. The POLL frames have the same frame format as the RTS frames except with value “0011” in

the *subtype* subfield[5] of the *frame control* field [50]. The AP maintains a polling list L, which consists of the wireless stations that have follow-on frames waiting to be transmitted in P-mode. The AP only sends the POLL frames to those stations on its polling list, and since the AP is also equipped with a DCF device, it has to contend for the wireless medium to transmit the POLL frames. As a result, collisions may occur to the RTS frames as well as to the POLL frames. Figs. 2.6 and 2.7 illustrate different timings of successful frame transmissions in C-mode and P-mode, respectively. Notice that a wireless station in P-mode can only transmit a data frame after receiving a POLL frame from the AP, and hence, does not contribute to the contention on the wireless medium.

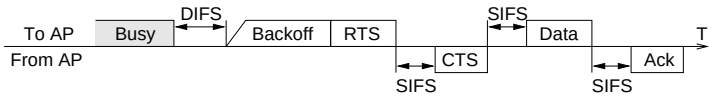


Figure 2.6: C-mode frame transmission

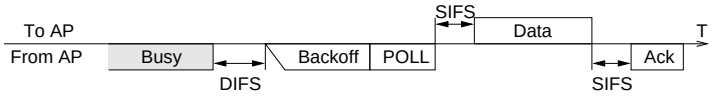


Figure 2.7: P-mode frame transmission

The WB-DCF requires each wireless station to maintain a local *xmit_mode* flag that is initially set to C-mode. Before a frame transmission attempt, a wireless station checks if there are more data frames waiting in the queue, and sets the *more data* bit in the current data frame accordingly. Then, the data frame is transmitted to the AP using the mode specified by the *xmit_mode* flag. After the frame transmission attempt, the *xmit_mode* flag is updated to P-mode if the queue is non-empty, or to C-mode otherwise.

Fig. 2.8 shows the pseudo-coded algorithm executed by the AP to update its polling list according to the information carried in the incoming data frames. The AP adds a station

[5] In the 802.11 standard, the RTS frames have value "1011" in the *subtype* subfield of the *frame control* field, while the value of "0011" is reserved for future use.

to L when it receives from the station a data frame with the *more data* bit set to one in the *frame control* field and removes a station from L upon reception of a data frame whose *more data* bit is zero. cw^*_{ap} is the contention window size used by the AP for its POLL frame transmission attempts, and how to determine its value will be discussed in the next section. As shown in the pseudo-code, after the backoff counter reaches zero, the AP sends a POLL frame to the next station on its polling list according to certain transmission policy. Since the WB-DCF is designed to achieve the weighted fairness among multiple traffic flows, the AP simply polls the stations in a weighted round-robin fashion.

```
L := ∅;  bkoff_value := 0;
while (the wireless network is alive) {
  if (L ≠ ∅) && (bkoff_value == 0)
    bkoff_value := cw*_ap × rand() / RAND_MAX;
  ⟨ monitor medium activity ⟩;
  ⟨ decrement bkoff_value and/or update cw*_ap if necessary ⟩;
  switch (one of the three monitored events happens) {
    case (an RTS frame R is received successfully):
        ⟨ transmit frame CTS(R) back to TA(R) ⟩
        break;
    case (a data frame D is received successfully):
        ⟨ transmit frame Ack(D) back to TA(D) ⟩
        if (more_data(D) == 1)
          L := L ∪ TA(D);
        if (more_data(D) == 0) && (TA(D) ∈ L)
          L := L − TA(D);
        break;
    case (bkoff_value == 0):
        if (L ≠ ∅)
          ⟨ transmit a POLL frame to next_sta(L) ⟩
        break;
  }
}
```

Figure 2.8: The polling-list-maintenance algorithm executed by the AP

In the WB-DCF, the mode selection for frame transmissions varies with the traffic load. When the system is lightly-loaded, most new frames see empty queues upon their generation and are transmitted in C-mode. When the system is heavily-loaded, most new frames

appear as follow-ons and are transmitted in P-mode. In particular, when the network is running under the saturation condition (i.e., the queues of wireless stations are never empty), all the frames will be transmitted in P-mode after the system gets stabilized. In this extreme case, the WB-DCF works exactly like a polling scheme. On the other hand, the worst-case traffic scenario happens if all the frames appear as head-of-lines. In this case, each data frame has to be transmitted in C-mode, and the WB-DCF works exactly like a contention-based scheme.

Fig. 2.9 gives a simple example to illustrate how the WB-DCF works. Assume there are two wireless stations, W1 and W2, in the network and each station has two data frames waiting for transmission. Initially, both stations attempt to transmit their head-of-line frames (F11 and F21) in C-mode. After the RTS collision, W2 backs off and W1 continues its transmission attempt. At Time 2, W1 successfully transmits its RTS frame to the AP, and as a result, the AP adds W1 to its polling list. The AP has to contend with W2 to send its POLL frames to W1. We can see that the POLL frames may also collide with the RTS frames. Eventually, at Time 4, an RTS frame from W2 gets through, and W2 is added to the AP's polling list. Both follow-on frames (F12 and F22) will then be transmitted in P-mode to avoid further collisions.

2.4.2 Part II: Weighted Contention Window Selection

Although introducing P-mode frame transmissions reduces the number of contending stations, it does not eliminate the collisions completely. Therefore, in order to achieve a low frame collision probability and maximize the channel utilization, the WB-DCF replaces the original binary exponential backoff scheme in the DCF with an enhanced backoff scheme that selects the contention window size for each wireless station according to Eqs. (2.34) and (2.35). In order to do so, a wireless station has to estimate $|f_i|$, the number

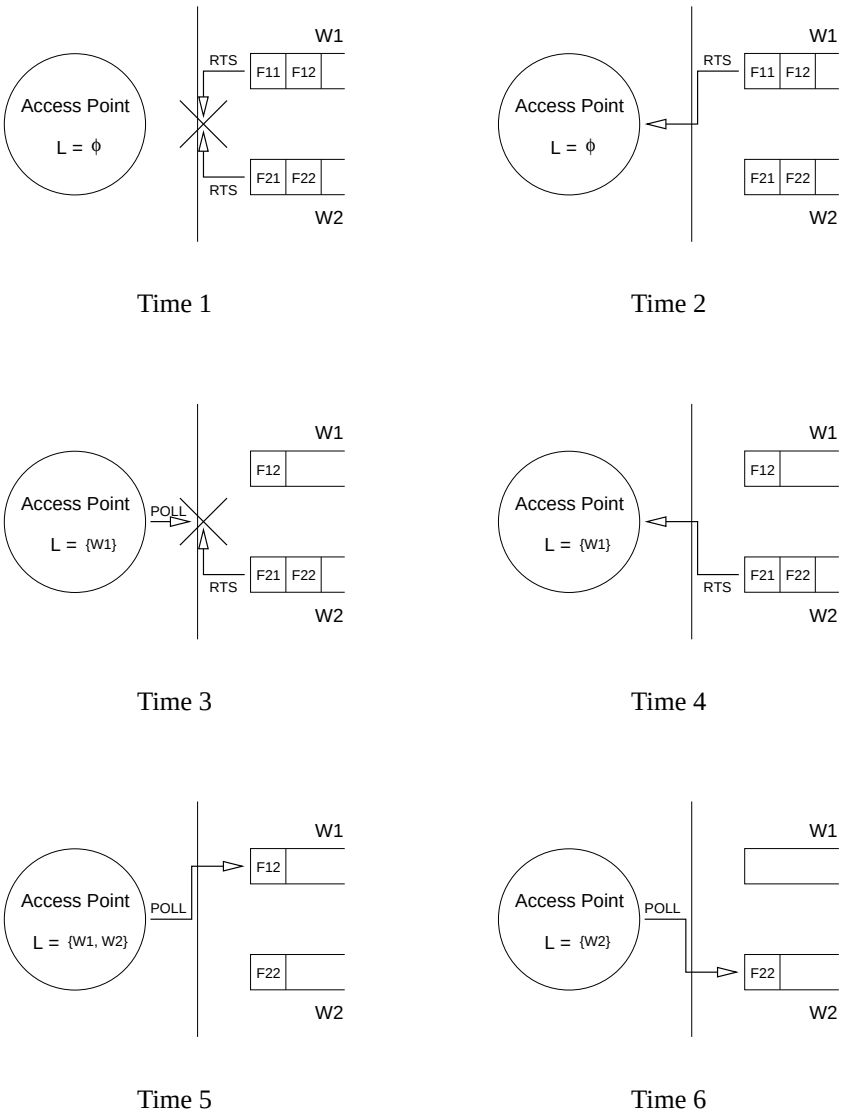


Figure 2.9: An example of frame transmissions under the WB-DCF

of contending stations carrying class$-i$ traffic ($i = 1, \cdots, n$), and we have developed a simple estimation scheme for this purpose as follows.

The WB-DCF requires each wireless station to keep sensing the channel and monitoring the activities on the wireless medium when it is not transmitting. Besides, the AP is given the highest priority to access the wireless medium in the WB-DCF, i.e., a P-mode frame transmission is treated at the same priority level as a C-mode frame transmission of

class-1:

$$\phi_{ap} = \phi_1 = 1 \quad \text{and} \quad cw^*_{ap} = cw^*_1. \tag{2.36}$$

Therefore, each station knows whether at each time slot the wireless medium is busy or idle, whether a busy period corresponds to a collision or not, and which traffic class a successful frame transmission belongs to. Let *avg_idle* and *avg_wait_i* ($i = 1, \cdots, n$) denote the average number of consecutive idle slots on the wireless medium and the average number of time slots between two consecutive successful class$-i$ frame transmissions, respectively, and they can be calculated as

$$\textit{avg\_idle} = \frac{1}{P_{tr}} - 1, \tag{2.37}$$

and

$$\textit{avg\_wait\_i} = \frac{1}{P_{tr(i)}} - 1. \tag{2.38}$$

P_{tr} was given by Eq. (2.9) and represents the probability that at least one station attempts to transmit in a slot. $P_{tr(i)}$ is the probability of a successful frame transmission from any station that carries class$-i$ traffic, and is given by

$$P_{tr(i)} = |f_i| \cdot p_i \cdot (1 - p_i)^{|f_i|-1} \cdot \prod_{j \neq i} (1 - p_j)^{|f_j|}. \tag{2.39}$$

Notice the following relation:

$$\frac{1 - P_{tr}}{P_{tr(i)}} = \frac{1 - p_i}{|f_i| p_i}, \tag{2.40}$$

and hence,

$$|f_i| = \frac{cw^*_i \cdot (\textit{avg\_idle} + 1)}{2 \cdot \textit{avg\_idle} \cdot (\textit{avg\_wait\_i} + 1)}. \tag{2.41}$$

So, based on the measurements of *avg_idle* and *avg_wait_i* by monitoring the medium activities, each station can estimate the values of $|f_i|$'s using Eq. (2.41).

Fig. 2.10 shows the pseudo-coded algorithm executed by each wireless station to monitor the medium activity and adjust its contention window size. The number of traffic

classes (n) and the associated weight for each class (ϕ_i) are assumed to be available *a priori* to each station. Each station maintains a set of random variables, "IDLE" and "WAIT(i)" for each $i \in \{1, 2, \cdots, n\}$. The contention window size for the stations carrying class-1 traffic, cw_1^*, is initialized to cw_start, a design parameter. Let an idle-busy-cycle be the time interval between the ends of two adjacent busy periods on the wireless medium. The observation window size w_{obs}, another design parameter, represents the number of idle-busy-cycles within which the measurements of *avg_idle* and *avg_wait_i* are taken, and the count w_{count} for monitored idle-busy-cycles is reset to zero. As shown in the pseudo-code, "IDLE" is updated after each idle-busy-cycle, while "WAIT(i)" is updated only if the busy period corresponds to a successful class$-i$ frame transmission. At the end of each observation window, the values of $|f_i|$'s are estimated, and the contention window sizes are adjusted according to these estimates. Finally, based on the traffic class a station is carrying, it can determine which contention window size to use for its next frame transmission attempt. Notice that t_{curr} and $t_{prev}(j)$ are the discrete time points measured in time slots, and α and β are both smoothing factors.

2.5 Performance Evaluation

We evaluate the effectiveness of the WB-DCF using the ns-2 simulator [2] after enhancing the original 802.11 DCF module of ns-2.

2.5.1 Simulation Setup

Although the results and conclusions presented in this chapter do not depend on the physical layer (PHY) technology, the PHY does determine some network parameters, such as SIFS and DIFS. We assume that each simulated wireless station operates at the 802.11a PHY [25] mode-8, and the related network parameters are listed in Table 2.1.

We evaluate three testing schemes: the proposed WB-DCF protocol, WB-DCF w/o

$cw_1^* := cw\_start$; $w_{count} := 0$;
$cw_{ap}^* := cw_1^*$;
for $(i = 2 : n)$ $cw_i^* := cw_1^*/\phi_i$;
while (the wireless network is alive) {
  if (a new idle-busy-cycle has been monitored on the wireless medium) **then** {
    $w_{count} := w_{count} + 1$;
    IDLE := $\alpha \cdot$ IDLE $+ (1 - \alpha) \cdot$ new_idle;
    if (the busy period is due to a successful frame transmission) **then** {
      $j :=$ the traffic class of the delivered frame;
      WAIT(j) := $\alpha \cdot$ WAIT(j) $+ (1 - \alpha) \cdot [t_{curr} - t_{prev}(j) - 1]$;
      $t_{prev}(j) := t_{curr}$;
    }
    if ($w_{count} = 0$ mod w_{obs}) **then** {
      for $(i = 1 : n)$
        $|f_i|_{est} = \beta \cdot |f_i|_{est} + (1 - \beta) \cdot \frac{cw_i^* \cdot (\text{IDLE}+1)}{2 \cdot \text{IDLE} \cdot (\text{WAIT}(i)+1)}$;
      $\langle$ calculate cw_1^* using Eq. (2.34) based on the new $|f_i|_{est}$ values $\rangle$
      $cw_{ap}^* := cw_1^*$;
      for $(i = 2 : n)$ $cw_i^* := cw_1^*/\phi_i$;
      $\langle$ reset IDLE, WAIT(i), and $t_{prev}(i)$ $\rangle$
    }
  }
}

Figure 2.10: The medium-monitoring algorithm executed by each wireless station

P-mode (in which all the data frames are transmitted in C-mode), and the original 802.11 DCF (with $cw_{min} = 15$ and $cw_{max} = 1023$, as specified in the 802.11a standard [25]). For the WB-DCF and WB-DCF w/o P-mode, the observation window w_{obs} is set to 50, while the smoothing factors α and β are chosen to be 0.9 and 0.7, respectively.

The testing schemes are compared with each other in terms of the aggregate system throughput and the average station access delay. The station access delay is defined as the time elapsed from when a data frame reaches the head of the queue to when the frame is successfully transmitted. For simplicity, the simulation results for the station access delay are measured in time slots. For example, in Fig. 2.1, the access delay of station A when transmitting frame A2 is six time slots. Note that such results illustrate the same information as those in absolute time units, or in seconds.

Parameters	Value	Comments
rTransmit	54 Mbps	transmission rate
BpS	27	bytes per OFDM symbol
tSlotTime	9 μs	Slot time
tSIFSTime	16 μs	SIFS time
tDIFSTime	34 μs	DIFS = SIFS + 2 $\times$ Slot
tPropDelay	1 μs	propagation delay
aRTSLength	20 octets	RTS frame length
aCTSLength	14 octets	CTS frame length
aAckLength	14 octets	Ack frame length
aMACOverhead	28 octets	MAC overhead
tPHYOverhead	20 μs	PHY overhead
tSymbol	4 μs	OFDM symbol interval

Table 2.1: Network parameters used in the simulation

We also compare the fairness of the testing schemes using a measure called the fairness index as follows. Let Th_f denote the throughput of traffic flow f and let ϕ_f be the associated weight. The fairness index is then defined as

$$\textit{fairness index} = \frac{\mu(\frac{Th_f}{\phi_f})}{\mu(\frac{Th_f}{\phi_f}) + \sigma(\frac{Th_f}{\phi_f})}, \tag{2.42}$$

where μ and σ are, respectively, the mean and the standard deviation of $\frac{Th_f}{\phi_f}$ over all the active traffic flows. When the perfect fairness is achieved, the ratio $\frac{Th_f}{\phi_f}$ is the same for all flows, and the fairness index is equal to one. In general, the fairness index is a real value between zero and one, and the closer to one the fairness index, the fairer.

The burst arrival process at each wireless station is assumed to be Bernoulli [36] with a deterministic number of data frames per burst. We conduct the simulation with three different burst models: 1 fpb (frames per burst), 2 fpb, and 5 fpb. Obviously, the 1-fpb model is the worst among the three for the WB-DCF, since the data frames generated by this model are most likely to become head-of-line frames. On the other hand, the 5-fpb model is the one that favors the WB-DCF the most. Each simulation run lasts six minutes in an 802.11 system with an AP and 10 wireless stations contending for the shared medium. The frame size is 2304 octets unless specified otherwise.

2.5.2 Simulation Results

We first study the scenario under which all the wireless stations carry class-1 traffic ($\phi_1 = 1$). Fig. 2.11 shows the simulation results with the 1-fpb model. Along the X-axis are the generating probabilities (p_{ber}) of the Bernoulli burst arrival process. We have two observations. First, WB-DCF w/o P-mode only shows a marginal, if any, throughput improvement over the original DCF. This observation is surprising at the first sight, but rather reasonable. In WB-DCF w/o P-mode, the contention window size is carefully selected to achieve a low frame collision probability and maximize the aggregate system throughput. As shown in Fig. 2.11(a), when $p_{ber} > 0.04$, throughput is almost constant around 33 Mbps, which is the maximum possible with any purely contention-based access protocol for our simulated network. On the contrary, in the DCF, the slotted binary exponential backoff mechanism is heuristic and may result in more collisions. However, due to the introduction of the RTS/CTS exchange, the collision length is reduced drastically since collisions only occur to the RTS frames that are much shorter than the data frames. As a result, the more collisions resulted in the DCF do not have significant impact on the throughput performance. For the same reason, WB-DCF w/o P-mode also shows comparable delay performance to the DCF under heavy loads. However, it does show better delay performance under light loads, because a wireless station running WB-DCF w/o P-mode can select a very small contention window when the system is lightly-loaded, and transmit its data frames shortly after they are generated, instead of waiting for a longer backoff period as required in the DCF.

Second, the WB-DCF shows similar performance as WB-DCF w/o P-mode under light loads, because most new frames see empty queues upon their generation and are transmitted in C-mode. However, the WB-DCF achieves significantly higher throughput and shorter access delay under medium to heavy loads. The rationale behind these is that, when

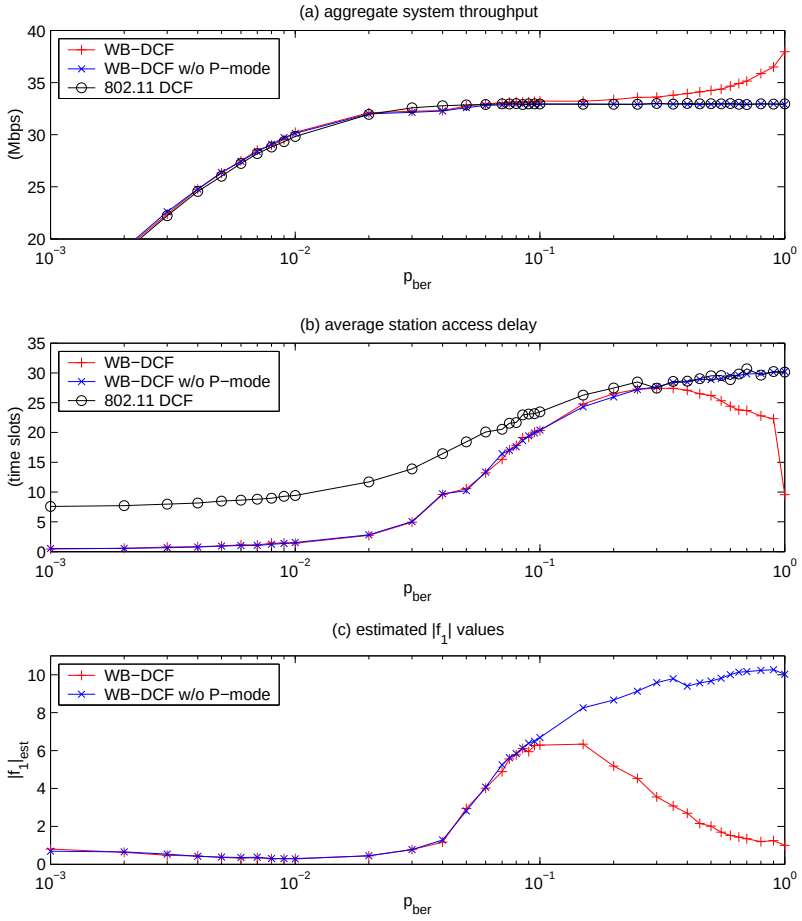


Figure 2.11: Comparison results with the 1-fpb model

the traffic load increases, more and more frames appear as follow-ons and are transmitted in P-mode, and the number of contending stations starts decreasing. Fig. 2.11(c) compares the estimated numbers of contending stations under different traffic load conditions for the WB-DCF and WB-DCF w/o P-mode. Clearly, the hump curve for the WB-DCF corroborates the facts explained above. In contrast, without P-mode, the number of contending stations increases monotonically. When p_{ber} reaches one, i.e., the system is saturated, the WB-DCF is reduced to polling and achieves the highest throughput ($\sim$38 Mbps) and a short access delay ($\sim$9 time slots). Note that nine time slots is the shortest possible station access delay under the saturation condition of our network configuration, where 10 wireless stations are contending for the share medium.

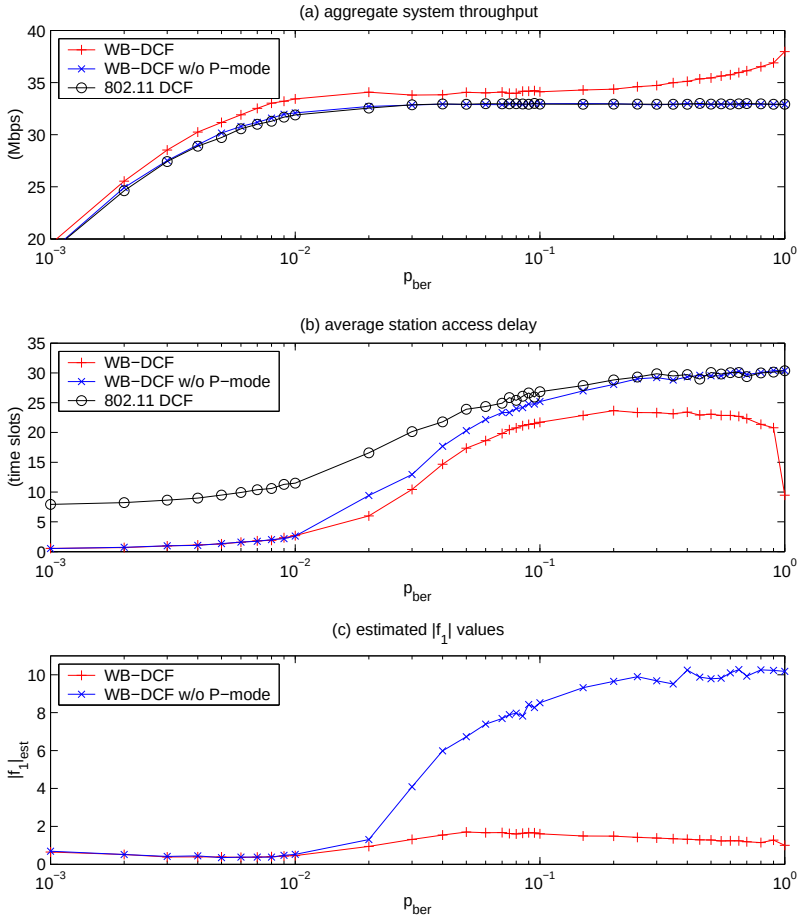


Figure 2.12: Comparison results with the 2-fpb model

Figs. 2.12 and 2.13 plot the simulation results for the 2-fpb model and the 5-fpb model, respectively. As expected, the performance improvements of the WB-DCF over WB-DCF w/o P-mode and the DCF are more pronounced due to the increasing percentage of the P-mode frame transmissions. Fig. 2.14 compares the testing schemes using the (throughput, delay) plots. Notice that the plots of both WB-DCF w/o P-mode and the DCF are quite similar for different burst models, and the throughput bound of $\sim$33 Mbps can be clearly observed in the figure. On the other hand, thanks to its mixture of contention with polling, the WB-DCF achieves higher throughput with shorter access delay. As shown in Fig. 2.14(c), under the 5-fpb model, the WB-DCF can even achieve the throughput of 36 Mbps with the access delay less than three time slots. The saturation points ($\sim$38 Mbps,

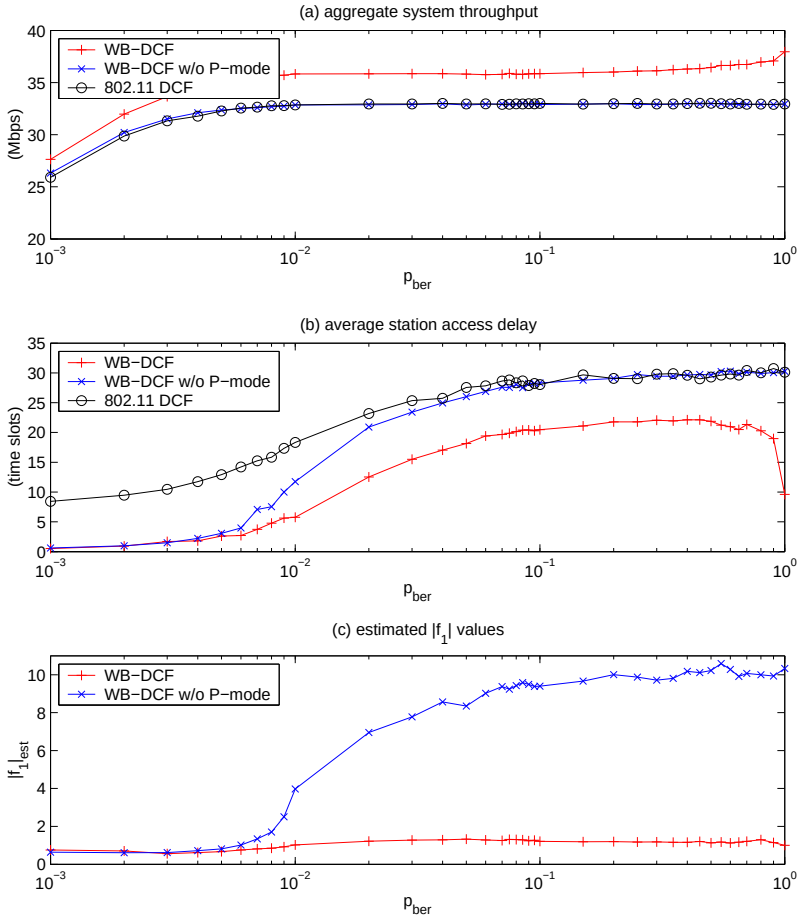


Figure 2.13: Comparison results with the 5-fpb model

$\sim$9 time slots) for the WB-DCF are singled out by the left arrows in the figure.

The fairness performances of the testing schemes are compared in Fig. 2.15. We observe that the WB-DCF yields the best fairness performance among the three, and in particular, it outperforms the DCF significantly under medium to heavy load conditions. Since we now study the scenario under which all the wireless stations carry class-1 traffic flows, a uniform optimal contention window (cw_1^*) is selected by all the stations for their C-mode frame transmissions in the WB-DCF. Besides, the round-robin transmission policy for P-mode frame transmissions results in fair medium sharing among the wireless stations on the AP's polling list, which also contributes to the excellent fairness performance of the WB-DCF. In contrast, the DCF allows each wireless station to adapt its contention window

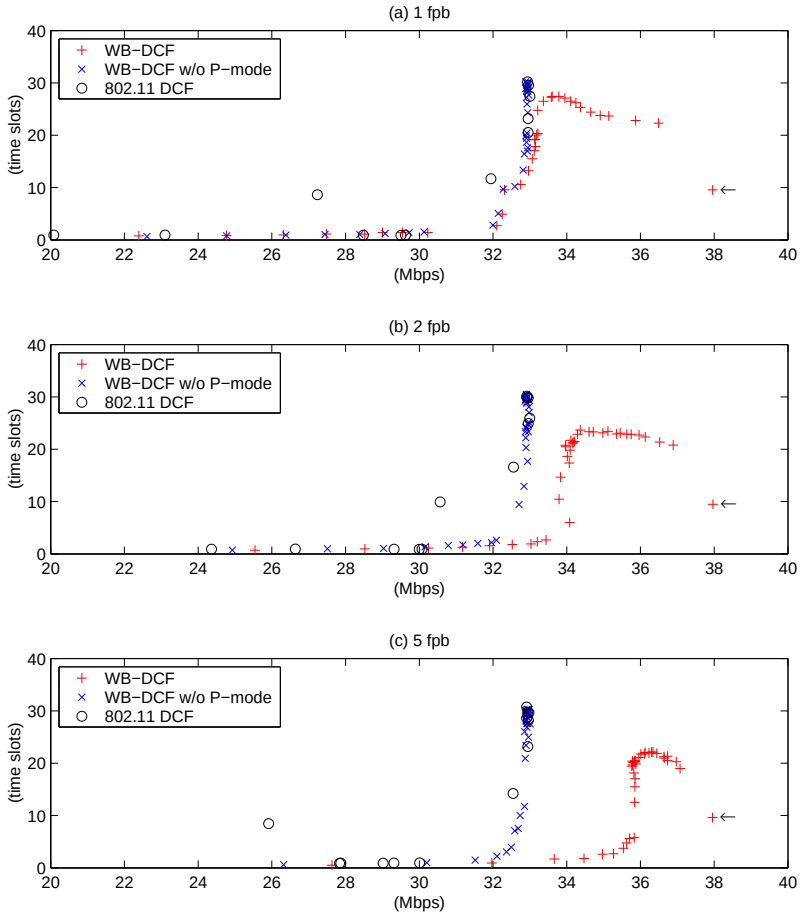


Figure 2.14: Aggregate system throughput vs. Average station access delay

size heuristically and independently between cw_{min} and cw_{max}.

Next, we study the behavior of the proposed load estimation scheme in the presence of traffic fluctuations. The burst model used in the simulation is 1 fpb. The variation pattern of p_{ber} is shown as the stair-like plot in Fig. 2.16(a), and the estimated numbers of contending stations in the WB-DCF and WB-DCF w/o P-mode are plotted in Figs. 2.16(b) and (c), respectively. Clearly, these estimates follow the load variation and react to the changes quickly. The figure also shows that the WB-DCF results in less contending stations under various load conditions, which is consistent with the previous observations in Fig. 2.11.

Finally, we study the scenario with multiple traffic classes in the network: five wireless stations carry class-1 traffic ($\phi_1 = 1$) and the other five carry class-2 traffic ($\phi_2 = 0.5$).

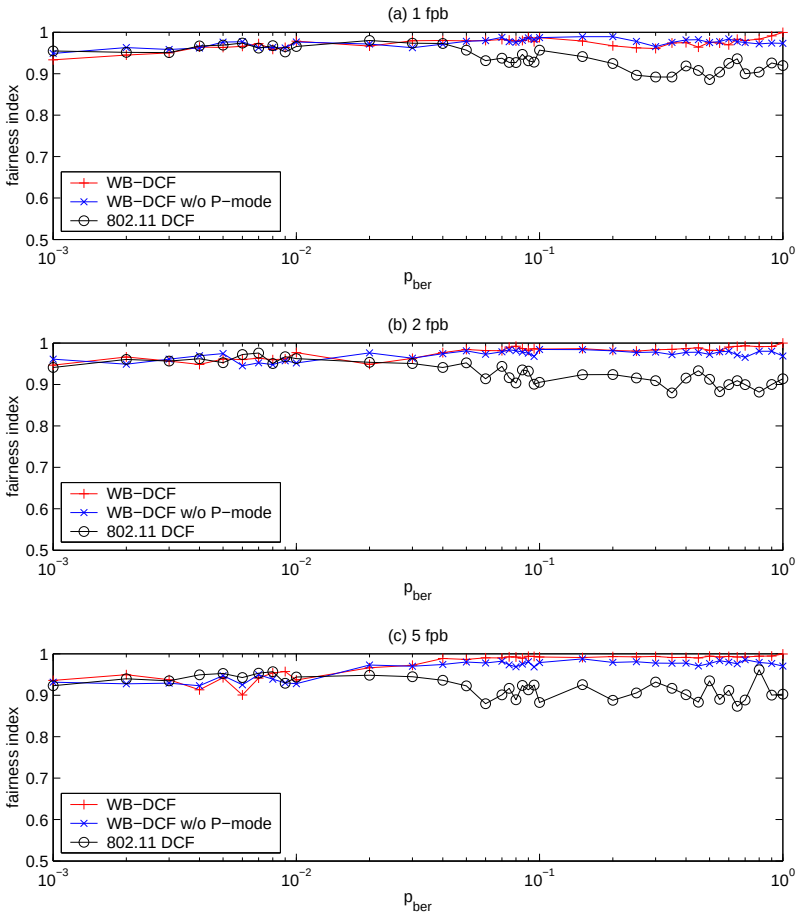


Figure 2.15: Fairness comparison (single traffic class: $\phi_1 = 1$)

The results for the aggregate system throughput and the average station access delay are omitted since they are very similar to those under the single-traffic-class scenario. We only show the fairness comparison results in Fig. 2.17. As expected, the DCF cannot support the weighted fairness at all, since it is designed to achieve equal medium sharing among contending stations by assuming implicitly the same weights of all traffic flows. On the other hand, the WB-DCF considers the relative weights among traffic flows in both the contention window selection scheme for C-mode frame transmissions and the polling scheme for P-mode frame transmissions, and hence, achieves excellent weighted fairness.

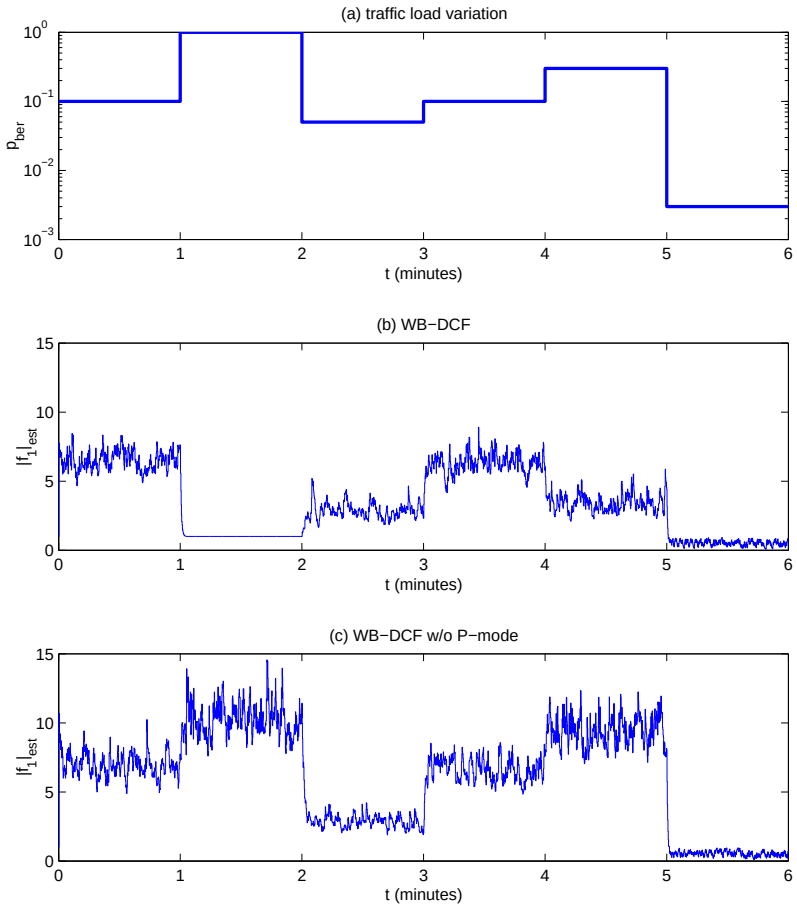


Figure 2.16: Performance of the proposed load estimation scheme

2.6 Conclusion

In this chapter, we propose a simple enhancement to the 802.11 DCF, called the WB-DCF, to achieve the weighted fairness and maximize the channel utilization for data communications in an 802.11 DCF system. The key ideas of the WB-DCF are that (1) the number of contending stations is reduced by introducing P-mode frame transmissions; (2) the contention window size for each wireless station is carefully determined according to the runtime load estimation, so as to achieve a low frame collision probability and maximize the channel utilization; and (3) the weighted fairness among traffic flows is guaranteed by the weighted contention window selection scheme at each wireless station and

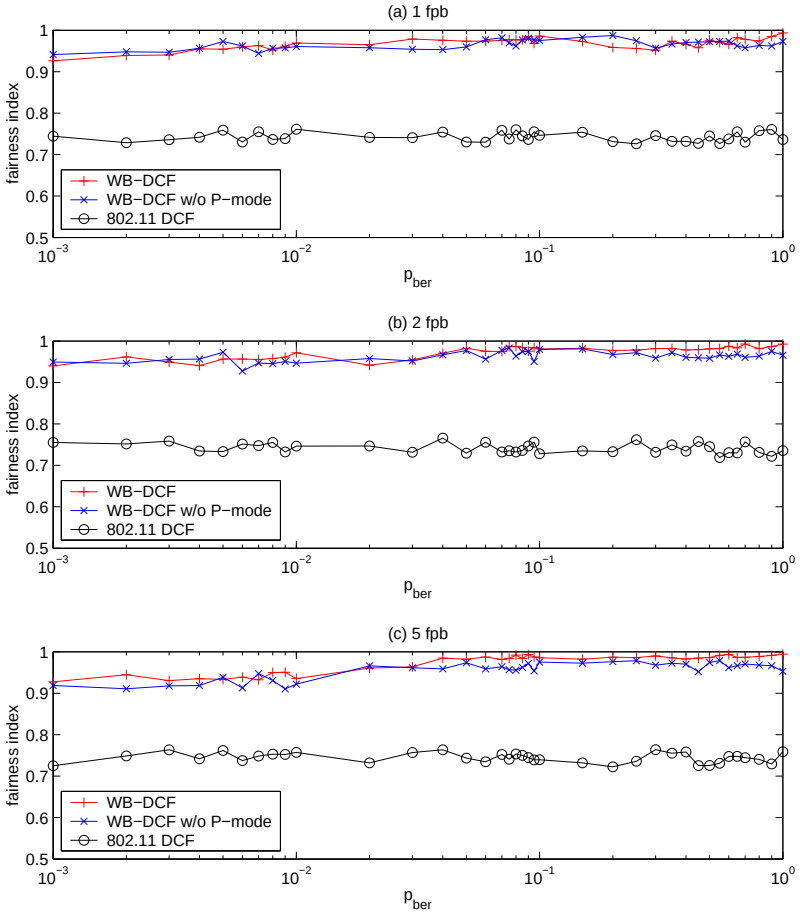


Figure 2.17: Fairness comparison (two traffic classes: $\phi_1/\phi_2 = 1/0.5$)

the weighted round-robin transmission policy at the AP. Simulation results show that the WB-DCF outperforms the DCF significantly in terms of throughput, delay, and fairness under various load conditions, and is able to react properly and quickly to the traffic load fluctuation.

CHAPTER III

AN OPTIMAL LOW-ENERGY TRANSMISSION STRATEGY FOR 802.11A/H

3.1 Introduction

Most wireless stations, such as laptops and palmtops, are battery-powered and hence have a limited amount of energy. It is, therefore, very important to reduce the energy consumption by wireless communication devices. In this chapter, we address the energy conservation issue in 802.11 systems, or more specifically, the emerging 802.11a/h systems.

An 802.11 device can operate in one of the following modes: *transmit mode*, *receive mode*, *idle mode*, or *doze mode*. It consumes the highest power in the transmit mode and very little energy in the doze mode. In the idle mode, an 802.11 device is required to sense the medium, and hence, consumes a similar amount of power as when it is in the receive mode [73]. Several power-management policies [5, 33, 38, 72, 73] have been proposed to force an 802.11 device to enter the power-saving doze mode adaptively at appropriate moments to save battery energy.

An alternative way to conserve energy is to apply TPC (Transmit Power Control) in 802.11 systems [3, 16, 19], which allows an 802.11 device to use the minimum required power level in the transmit mode and is complementary to the power-management poli-

cies. In this chapter, we first provide a thorough analysis of the interference in 802.11a DCF systems [59], then propose a novel intelligent TPC mechanism, called *MiSer* [58], to minimize the communication energy consumption.

3.1.1 Motivation and Key Contributions

The 802.11 PHYs provide multiple transmission modes/rates by employing different modulation and channel coding schemes. For example, the 802.11b PHY [26] provides four PHY rates from 1 to 11 Mbps at the 2.4 GHz ISM (Industrial, Scientific, and Medical) band and most 802.11 devices available today in the market are based on this PHY. Another higher-speed PHY, the 802.11a PHY [25], has also been developed to extend the 802.11 operation in the 5 GHz U-NII (Unlicensed National Information Infrastructure) band and provides eight PHY rates ranging from 6 to 54 Mbps. As the first-generation 802.11a products become available in the market, the 802.11a PHY receives increasing attention due mainly to its higher transmission rates as well as the cleaner 5 GHz operational band. Moreover, the emerging 802.11h standard [28], which is an extension to the current 802.11 MAC and the 802.11a PHY, provides a transmit-power reporting mechanism that makes intelligent TPC feasible at the MAC layer. So, it is important to have a well-designed TPC mechanism work with the 802.11a/h such that its TPC capability and multiple transmission rates can be fully exploited.

Note that, due to the contention nature of the DCF, the effectiveness of a TPC mechanism relies on the condition that applying TPC on data transmissions will not aggravate the “hidden nodes” problem or the interference in the network [20]. The first contribution of this chapter is to provide a rigorous analysis of the relationship among different radio ranges and TPC’s effects on the interference in 802.11a DCF systems, then based on the interference analysis, we propose to apply TPC in 802.11a DCF systems in the follow-

ing way: in addition to exchange RTS/CTS frames to reserve the wireless channel before each data transmission attempt, which has been used in many other proposed TPC mechanisms [3, 19], the CTS frames are transmitted at a stronger power level to ameliorate the TPC-caused interference.

The second contribution of this chapter is to propose a novel per-frame-based intelligent TPC mechanism, called *MiSer* (Minimum-energy transmission Strategy), for 802.11a/h DCF systems. MiSer is deployed as RTS-CTS(strong)-Data(MiSer)-Ack. Obviously, the lower the transmit power or the higher the PHY rate (hence, the shorter the transmission time), the less energy consumed in one single transmission attempt, but more likely the transmission will fail, thus causing re-transmissions and eventually consuming more energy. So, there are inherent tradeoffs, and the key idea behind MiSer is to combine TPC with PHY rate adaptation and pre-establish a rate-power combination table indexed by the data transmission status quadruplet that consists of the data payload length, the path loss condition, and the frame retry counts. Each entry of the table is the optimal rate-power combination in the sense of maximizing the energy efficiency — which is defined as the ratio of the expected delivered data payload to the expected total energy consumption — under the corresponding data transmission status. At runtime, a wireless station determines the best transmit power as well as the proper PHY rate for each data transmission attempt by a simple table lookup, using the most up-to-date data transmission status as the index.

3.1.2 The Evolution of MiSer

Initially, we researched energy-efficient frame transmission in 802.11a systems under the (optional) PCF. Since the access to the wireless medium is centrally-controlled by the AP (Access Point), there is no “hidden nodes” problem or medium contention in a PCF

system, which makes it easy to apply TPC to save energy. In [60], we derived the energy-consumption performance analytically for uplink data transmissions under the PCF and demonstrated the energy-efficient PCF operation via TPC and PHY rate adaptation.

It may seem reasonable to apply a similar idea under the (mandatory) DCF as well. However, as described in [20], if wireless stations are simply allowed to transmit at different power levels in a DCF system, the number of hidden nodes is likely to increase and the interference is aggravated, which, in turn, results in more transmission failures and re-transmissions, and hence, more energy will eventually be consumed. A natural way to deal with this problem is to exchange RTS/CTS frames before each data transmission attempt, which has been used in many proposed TPC mechanisms [3, 19].

Our preliminary study in [57] considered the simple infrastructure DCF system that includes an AP to provide both the connection to the wired network, if any, and the local relaying function within the system. Therefore, each wireless station must be able to hear the AP, and consequently, the hidden nodes are completely eliminated if RTS/CTS frames are exchanged before each data transmission attempt. Our simulation results showed the energy savings by TPC (with RTS/CTS support) in an infrastructure DCF system and confirmed the aggravated "hidden nodes" problem when TPC is applied directly to data transmissions without RTS/CTS support.

This problem becomes much more complicated in an ad hoc DCF system where the wireless stations, if within the communication range, communicate directly with each other. Since not every wireless station may be able to hear directly from all other stations, the RTS/CTS mechanism cannot guarantee elimination of the hidden nodes. Moreover, applying TPC to data transmissions, even with RTS/CTS support, aggravates the interference in an ad hoc DCF system. MiSer reflects our latest research results on this topic and can be used in both infrastructure and ad hoc DCF systems.

3.1.3 Organization

The rest of this chapter is organized as follows. Section 3.2 introduces the virtual sensing mechanism of the 802.11 DCF as well as the 802.11a PHY. In Section 3.3, following a theoretical analysis of the relationship among different radio ranges and TPC's effects on the interference in 802.11a DCF systems, an enhanced RTS-CTS(strong)-Data(TPC)-Ack mechanism is proposed and justified to accommodate intelligent TPC. A generic energy consumption model of the 802.11 device and the basic energy consumption computations in an 802.11a/h DCF system are presented in Section 3.4. Section 3.5 describes the details of MiSer and discusses the related implementation issues. Section 3.6 presents and evaluates the simulation results, and finally, the chapter concludes with Section 3.7.

3.2 System Overview

3.2.1 Virtual Sensing Mechanism of the 802.11 DCF

The timing of a successful four-way frame exchange in an 802.11 DCF system is shown in Fig. 3.1. If a CTS (Ack) frame is not received due possibly to an erroneous reception of the preceding RTS (Data) frame, as shown in Figs. 3.2 and 3.3, the transmitter will contend again for the medium to re-transmit the frame after a CTS (Ack) timeout. Note that, in these figures, a crossed block represents an erroneous reception of the corresponding frame.

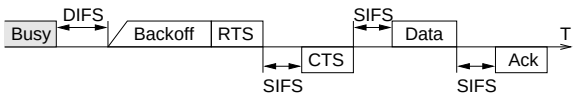


Figure 3.1: Timing of a successful four-way frame exchange under the DCF

Recall that, in an 802.11 DCF system, a wireless station is allowed to transmit only if its carrier-sense mechanism determines that the medium has been idle for DIFS time. The

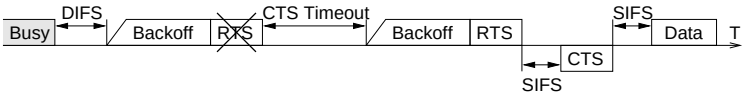


Figure 3.2: Re-transmission due to RTS transmission failure

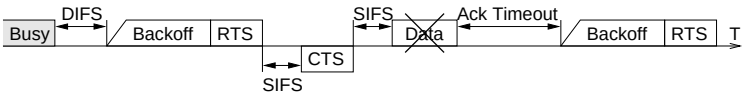


Figure 3.3: Re-transmission due to Data transmission failure

DCF includes a virtual sensing mechanism, called the NAV (Network Allocation Vector), in addition to physical sensing. The NAV is a value that indicates to a station the remaining time before the wireless medium becomes available, and it is updated upon each RTS/CTS frame reception using the Duration/ID value carried in the frame header. By examining the NAV, a station avoids transmitting a frame that may interfere with the subsequent Data/Ack frame exchange even when the wireless medium appears to be idle according to physical sensing. Fig. 3.4 illustrates how the wireless stations adjust their NAVs during a four-way frame exchange.

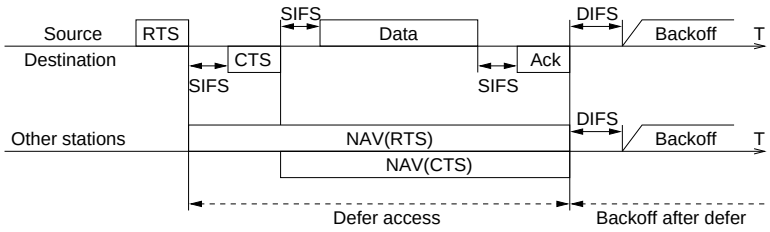


Figure 3.4: NAV setting during a four-way frame exchange

The 802.11 DCF requires a wireless station to maintain a short retry count (SRC) and a long retry count (LRC) for each data frame, and these counts are incremented and reset independently. When the length of a data frame is less than or equal to *dot11RTSThreshold*, the two-way Data-Ack handshake is used to transmit the frame, and only SRC is active and it is incremented every time a data transmission fails. The data frame is discarded when

SRC reaches *dot11ShortRetryLimit*. Else, the four-way RTS-CTS-Data-Ack handshake is used to transmit the frame, and SRC (LRC) is incremented every time an RTS (Data) transmission fails. The data frame is discarded when either SRC reaches *dot11ShortRetryLimit* or LRC reaches *dot11LongRetryLimit*. The default values of *dot11ShortRetryLimit* and *dot11LongRetryLimit* are seven and four, respectively. Note that, in both cases, SRC and/or LRC are reset to zero only after a successful data transmission or after a data frame is discarded.

3.2.2 The 802.11a PHY

The 802.11a PHY [25] is based on OFDM (Orthogonal Frequency Division Multiplexing) and provides 8 PHY modes/rates with different modulation schemes and convolutional codes at the 5 GHz U-NII band. As listed in Table 3.1, the OFDM system provides a WLAN with capabilities of communicating at 6 to 54 Mbps. The frame exchange between MAC and PHY is under the control of the PLCP (Physical Layer Convergence Procedure) sublayer.

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

* Bytes per OFDM Symbol

Table 3.1: Eight PHY Modes/Rates of the 802.11a PHY

3.3 Interference Analysis in 802.11a DCF Systems

Applying TPC, which allows an 802.11 device to use the minimum required power level in the transmit mode, is naturally an attractive way to save battery energy. However, due to the contention nature of the DCF, the effectiveness of a TPC mechanism relies on the condition that applying TPC on data transmissions will not aggravate the "hidden nodes" problem or the interference in the network. In this section, we first investigate the relationship among different radio ranges and TPC's effects on the interference in 802.11 DCF systems, then propose a novel way to apply TPC in 802.11a DCF systems while ameliorating the TPC-caused interference, and justify it based on a theoretical analysis.

3.3.1 Radio Ranges in 802.11 DCF Systems

In general, there are four different radio ranges in an 802.11 DCF system: *transmission range*, *NAV set range*, *CCA busy range*, and *interference range*.

- *Transmission range* is central to the transmitter and represents the range within which the receiver station can receive a frame successfully, assuming no interference from neighboring stations. It varies with the data payload length, the PHY rate, the transmit power, the radio propagation property that determines the path loss, and the receiver-side noise level.

- *NAV set range* is the range within which the wireless stations can set the NAVs correctly based on the Duration/ID information carried in the RTS/CTS frames and will not interfere with the subsequent Data/Ack frame exchange. Since the RTS/CTS frames are always transmitted at a fixed rate (e.g., 6 Mbps in 802.11a systems), the NAV set range is independent of the data rate.

- *CCA busy range* is central to the transmitter and represents the range within which the wireless stations can physically sense the channel busy during the data transmission (by the transmitter) and then defer their own transmission attempts. There are two methods for a wireless station to report CCA (Clear Channel Assessment) busy. One is based on *carrier detection*, and the other is based on *energy detection* by which a wireless station will report a busy medium upon detection of any signal power above the ED (Energy Detection) threshold.

- *Interference range* is central to the receiver and represents the range within which the wireless stations are able to interfere with the reception of data frames at the receiver.

3.3.2 TPC's Effects on the Interference in 802.11 DCF Systems

Figs. 3.5 and 3.6 sketch the relative positions of different radio ranges when the transmitter (*T*) transmits a data frame to the receiver (*R*) using the two-way Data-Ack handshake and the four-way RTS-CTS-Data-Ack handshake, respectively. NAV set range, CCA busy range, and interference range are shown as the light-, medium-, and dark-shaded areas, respectively. The NAV set range is actually the conjunction of the RTS transmission range and the CTS transmission range. Note that the sizes of radio ranges vary with 802.11 systems equipped with different PHYs.

A, *B*, *C*, *D*, *E*, *F*, and *G* are the seven neighboring stations. As shown in Fig. 3.5, when the two-way handshake is used, *A*, *B*, *C*, *D*, and *G* will not interfere with the Data/Ack frame exchange, since *A*, *B*, *C*, and *D* can physically sense the channel busy, and *G* is outside the interference range. On the other hand, *E* and *F* are unable to sense the data transmission, but are close enough to the receiver (within the interference range) to cause the interference. They are often referred to as the "hidden nodes" to *T*.

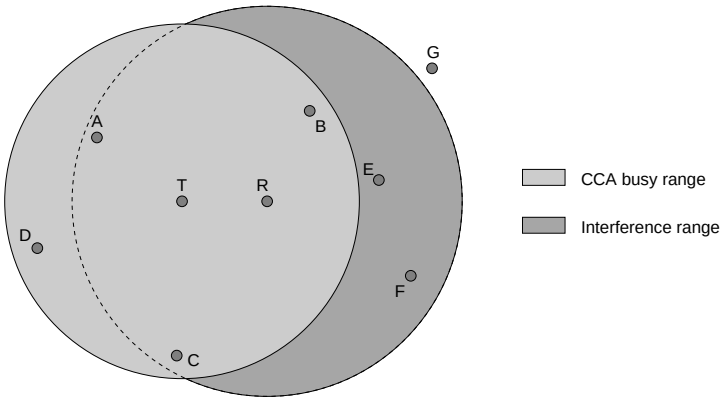


Figure 3.5: A sketch of the radio ranges during a two-way handshake

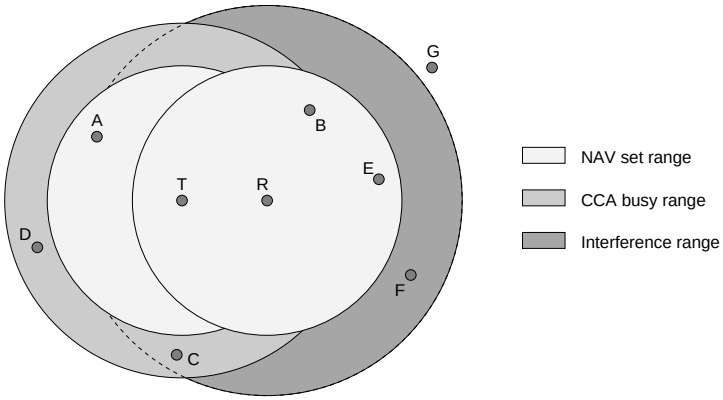


Figure 3.6: A sketch of the radio ranges during a four-way handshake

In order to alleviate such "hidden nodes" problem, *T* and *R* may exchange RTS/CTS frames to reserve the wireless channel before the actual data transmissions, as shown in Fig. 3.6. This way, *E* sets its NAV upon CTS reception and will not interfere with the subsequent Data/Ack frame exchange.

Now, let us see how the radio ranges are affected when TPC is applied on data transmissions. Since the kernel idea of TPC is to transmit a data frame at the minimum required power level, so when the two-way handshake is used, it may result in more hidden nodes in the network. For example, *B* becomes a hidden node when TPC is applied with the two-way handshake (see Fig. 3.5), while it will not interfere with the Data/Ack frame ex-

change when the four-way handshake is used as it is covered by the NAV set range (see Fig. 3.6). In fact, even with RTS/CTS support, applying TPC on data transmissions may still aggravate the interference in the following ways.

- *Scenario I:* The CCA busy range shrinks so that some neighboring stations that originally deferred their transmission attempts based on physical sensing may now interfere with the data frame reception (e.g., *C* in Fig. 3.6) or with the Ack frame reception (e.g., *C* and *D* in Fig. 3.6). This scenario may occur only when the original CCA busy range is larger than the NAV set range.

- *Scenario II:* The interference range is enlarged so that some neighboring stations (e.g., *G* in Fig. 3.6) that were originally outside the interference range, may now interfere with the data frame reception.

Having recognized the potential problem of aggravating the interference when applying TPC on data transmissions, we know that, in order for an intelligent TPC mechanism to be effective, it is critical to ameliorate the TPC-caused interference.

3.3.3 NAV Set Range vs. CCA Busy Range in 802.11a DCF Systems

According to the 802.11a standard [25], the *receiver minimum input level sensitivity* is defined as the received signal strength level at which the PER (Packet Error Rate) of a 1000-octet frame is less than 10%. It is rate-dependent and different sensitivity levels for different PHY rates are listed in Table 91 of [25]. For example, the receiver minimum sensitivity level for 6 Mbps is -82 dBm. Since the length of an RTS/CTS frame is much shorter than 1000 octets and they are transmitted at the most robust 6 Mbps, the PER of an RTS/CTS frame at the minimum 6 Mbps sensitivity level (-82 dBm) is almost zero. Therefore, it is safe to say that the RTS/CTS transmission range in an 802.11a DCF system

corresponds to the minimum 6 Mbps sensitivity level (-82 dBm). Recall that the NAV set range is the conjunction of the RTS transmission range and the CTS transmission range.

On the other hand, the *CCA sensitivity* is defined (in Clause 17.3.10.5 of [25]) as: "The start of a valid OFDM transmission at a receive level equal to or greater than the minimum 6 Mbps sensitivity (-82 dBm) shall cause CCA to indicate busy with a probability $> 90\%$ within 4 μs. If the preamble portion was missed, the receiver shall hold the carrier sense (CS) signal busy for any signal 20 dB above the minimum 6 Mbps sensitivity." Therefore, the CCA busy sensitivity levels based on carrier detection and energy detection are -82 dBm and -62 dBm, respectively, regardless of the data transmission rate.

We can make an important observation: *when the four-way handshake is used in an 802.11a DCF system to transmit a data frame, the CCA busy range is completely covered by the NAV set range.* This unique feature of 802.11a DCF systems is due to the fact that, the 802.11a PHY's ED threshold is set 20 dB higher than the carrier detection threshold, which is different from the 802.11b PHY that will be investigated in the next section for comparison purpose. Hence, the *Scenario I* described in Section 3.3.2 will never occur in an 802.11a DCF system, while it may cause serious interference problems in 802.11b DCF systems.

3.3.4 NAV Set Range vs. CCA Busy Range in 802.11b DCF Systems

The *receiver minimum input level sensitivity* of the 802.11b PHY [26] is defined as the received signal strength level at which the PER of a 1024-octet frame is less than 8%. Similarly, the RTS/CTS transmission range in an 802.11b DCF system corresponds to the minimum 1 Mbps sensitivity level since they are transmitted at the most robust 1 Mbps. However, the standards only specify the minimum 2 Mbps and 11 Mbps sensitivity levels to be -80 dBm (in Clause 15.4.8.1 of [24]) and -76 dBm (in Clause 18.4.8.1 of [26]),

respectively, but not for the PHY rates of 1 Mbps and 5.5 Mbps.

On the other hand, the value of the ED threshold varies with the actual 802.11b device as long as it obeys the following rule (in Clause 18.4.8.4 of [26]): "If a valid high-rate signal is detected during its preamble within the CCA assessment window, the energy detection threshold shall be less than or equal to -76 dBm for TX (transmit) power $>$ 100 mW; -73 dBm for 50 mW $<$ TX power $<$ 100 mW; and -70 dBm for TX power $<$ 50 mW." Besides, an 802.11b device is allowed to perform CCA using one of the following three methods: *energy above threshold*, *carrier sense with timer*, or *a combination of carrier sense and energy above threshold*. Note that, unlike the 802.11a, the 802.11b standard does not mandate either the ED threshold or the CCA method, and consequently, the relationship between the NAV set range and the CCA busy range in an 802.11b DCF system is implementation-dependent.

3.3.5 NAV Set Range vs. Interference Range in 802.11a DCF Systems

Since the signal power needed for interrupting a frame reception is much lower than that of delivering a frame successfully [78], under certain circumstances — especially, when TPC is used for data transmissions, as will be shown below — the interference range may be larger than the NAV set range. We now investigate the relationship between the transmit power and the interference range when three different four-way frame exchange mechanisms are used in an 802.11a DCF system.

RTS-CTS-Data-Ack

First, let us consider the conventional four-way handshake, where all the frames are transmitted at the same nominal power level ($\mathcal{P}_{nom}$).[1] As shown in Fig. 3.7, the distance between *T* and *R* is d. Let $d_{tx\_rc,6mbps}$ denote the radius of the RTS/CTS transmission

[1] In the following analysis, we let $\mathcal{P}_{nom}$ be 15 dBm, the nominal transmit power of the Agere ORiNOCO cards [4].

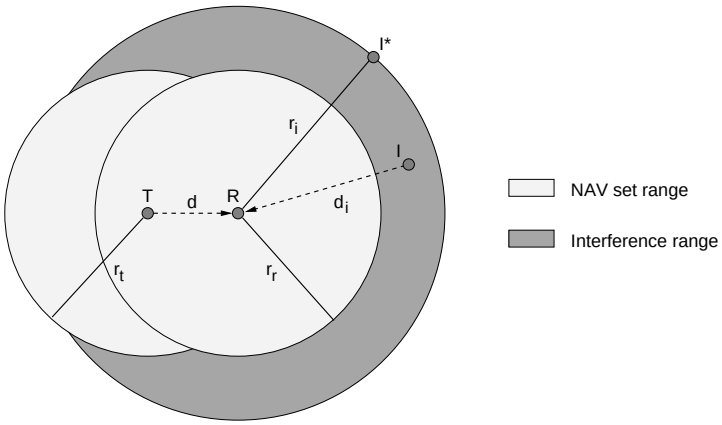


Figure 3.7: NAV set range vs. Interference range

range. So, we have

$$r_t = r_r = d_{tx\_rc,6mbps} \geqslant d. \tag{3.1}$$

Note that the CCA busy range is not shown in Fig. 3.7, as it is completely covered by the NAV set range in 802.11a systems.

Fig. 3.8 illustrates the radius of the interference range (r_i) when *T* transmits (at rate x) a data frame (with payload ℓ) to *R*. Let $SIR_{th\_\ell,x}$ be the SIR (Signal-to-Interference Ratio) threshold above which the data frame can be successfully received. Therefore, a neighboring station *I* interferes with the data frame reception if the following condition holds:

$$\begin{aligned} SIR_{th\_\ell,x} &\geqslant SIR \\ &= \mathcal{P}_{r,data} - \mathcal{P}_{r,int} \\ &= (\mathcal{P}_{t,data} - PL_d) - (\mathcal{P}_{t,int} - PL_{d_i}) \\ &= (\mathcal{P}_{nom} - PL_d) - (\mathcal{P}_{nom} - PL_{d_i}) \\ &= PL_{d_i} - PL_d \\ &= \left(\frac{d_i}{d}\right)^4 \qquad (3.2) \\ \iff \quad d_i &\leqslant \sqrt[4]{SIR_{th\_\ell,x}} \cdot d. \qquad (3.3) \end{aligned}$$

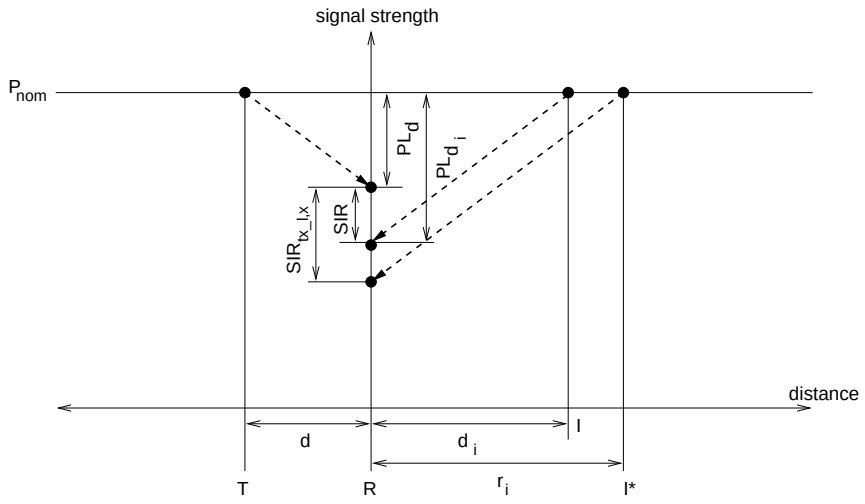


Figure 3.8: Without TPC, the size of the interference range varies with the data payload length, the transmission rate, and the distance between the transmitter and the receiver.

$\mathcal{P}_{t,data}$, $\mathcal{P}_{r,data}$, $\mathcal{P}_{t,int}$, and $\mathcal{P}_{r,int}$ are the transmit power of the data frame, the received data signal strength, the transmit power of the interference signal, and the received interference signal strength (all in dBm), respectively. PL_d and PL_{d_i} are the path losses (in dB) over distances d and d_i, respectively. Eq. (3.2) is obtained by assuming the log-distance path loss model with path loss exponent of four [68], which is suitable for indoor office environments. Eq. (3.3) implies that the radius of the interference range is

$$r_i = \sqrt[4]{SIR_{th\_\ell,x}} \cdot d. \tag{3.4}$$

We have two observations. First, when the conventional four-way handshake is used, the size of the interference range varies with the data payload length (ℓ), the transmission rate (x), and the distance (d) between the transmitter and the receiver. Only when d is larger than a certain value, the NAV set range will not be able to cover the interference range, i.e.,

$$d > \frac{d_{tx\_rc,6mbps}}{\sqrt[4]{SIR_{th\_\ell,x}}} \quad \Longrightarrow \quad r_i > r_r, \tag{3.5}$$

and then the neighboring stations that are inside the interference range but outside the NAV

set range can interfere with the data frame reception. Second, the interference signal could be RTS, CTS, Data, or Ack frames.

RTS-CTS-Data(TPC)-Ack

Now, let us examine how the interference range is affected when we only apply TPC on data transmissions while keeping the transmit power of RTS, CTS, and Ack frames at the nominal level, and consider the same configuration as shown in Fig. 3.7.

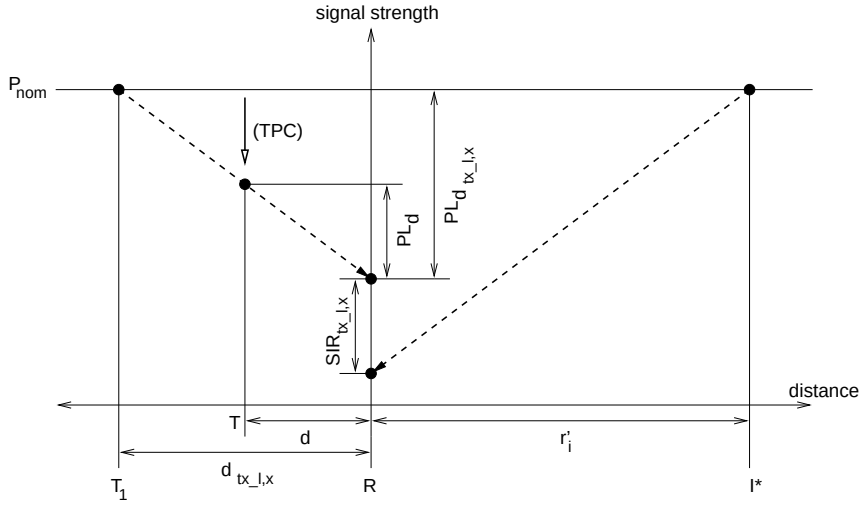


Figure 3.9: With TPC, the size of the interference range is independent of the distance from the transmitter to the receiver.

In this case, as illustrated in Fig. 3.9, the transmitter adapts its transmit power in such a way that the received data signal strength is always kept at the minimum required level, i.e.,

$$\mathcal{P}'_{t,data} \leqslant \mathcal{P}_{nom} \tag{3.6}$$

and

$$\mathcal{P}'_{t,data} - PL_d = \mathcal{P}_{nom} - PL_{d_{tx\_\ell,x}}, \tag{3.7}$$

where $d_{tx\_\ell,x}$ is the transmission range when a data frame with payload ℓ is transmitted at rate x using the nominal transmit power. Therefore, the condition for an interference to

occur becomes

$$\begin{aligned} SIR_{th\_\ell,x} &\geqslant SIR \\ &= \mathcal{P}'_{r,data} - \mathcal{P}'_{r,int} \\ &= \left(\mathcal{P}'_{t,data} - PL_d\right) - \left(\mathcal{P}'_{t,int} - PL_{d_i}\right) \\ &= \left(\mathcal{P}_{nom} - PL_{d_{tx\_\ell,x}}\right) - \left(\mathcal{P}'_{t,int} - PL_{d_i}\right) \\ &\geqslant \left(\mathcal{P}_{nom} - PL_{d_{tx\_\ell,x}}\right) - \left(\mathcal{P}_{nom} - PL_{d_i}\right) \\ &= PL_{d_i} - PL_{d_{tx\_\ell,x}}. \end{aligned} \tag{3.8}$$

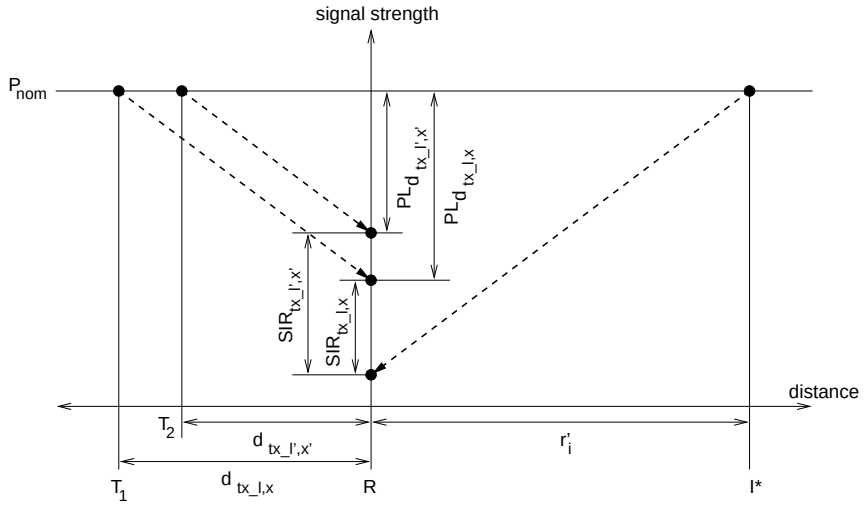


Figure 3.10: With TPC, the size of the interference range is also independent of the data payload length and the transmission rate.

Note that, when the data frame carries a larger payload ($\ell\uparrow$) or is transmitted at a higher rate ($x\uparrow$), a higher receiver-side SIR is required to have a successful frame reception ($SIR_{th\_\ell,x}\uparrow$), and consequently, the frame transmission range shrinks ($d_{tx\_\ell,x}\downarrow$), yielding a smaller $PL_{d_{tx\_\ell,x}}$ value. The changes in $SIR_{th\_\ell,x}$ and $PL_{d_{tx\_\ell,x}}$ are similar as shown in Fig. 3.10. Therefore, Eq. (3.8) is equivalent to

$$SIR_{th\_rc,6mbps} \geqslant PL_{d_i} - PL_{d_{tx\_rc,6mbps}}$$

$$\iff \quad SIR_{th\_rc,6mbps} \;\geqslant\; \left(\frac{d_i}{d_{tx\_rc,6mbps}}\right)^4$$
$$\iff \quad d_i \;\leqslant\; \sqrt[4]{SIR_{th\_rc,6mbps}} \cdot d_{tx\_rc,6mbps}. \tag{3.9}$$

Hence, the radius of the interference range becomes

$$r_i' = \sqrt[4]{SIR_{th\_rc,6mbps}} \cdot d_{tx\_rc,6mbps}. \tag{3.10}$$

It is interesting to see that the size of the interference range is now independent of the data payload length (ℓ), the transmission rate (x), and the distance (d) between the transmitter and the receiver, unlike when the conventional four-way handshake is used. Moreover, since $SIR_{th\_rc,6mbps}$ is larger than one, we have

$$r_i' > d_{tx\_rc,6mbps} = r_r, \tag{3.11}$$

which means that the interference range is always larger than the NAV set range. As a result, there are always some potential hidden nodes to interfere with the data frame reception, meaning that the interference is aggravated. This is actually the *Scenario II* described in Section 3.3.2. The interference signal could be RTS, CTS, Data, or Ack frames.

RTS-CTS(strong)-Data(TPC)-Ack

In order to deal with the aggravated interference problem caused by TPC, we propose to transmit the CTS frames at a stronger power level ($\mathcal{P}_{nom^+}$), which is 5 dB higher than, or equivalently, 3.16 times, the nominal transmit power. Since we let $\mathcal{P}_{nom}$ be 15 dBm, $\mathcal{P}_{nom^+}$ is 20 dBm and conforms to the 23 dBm transmit power limitation.[2] Now, with our enhanced four-way frame exchange mechanism, the NAV set range is enlarged to

$$r_r'' = \sqrt[4]{3.16} \cdot r_r = 1.33\; d_{tx\_rc,6mbps}. \tag{3.12}$$

[2] According to the 802.11 standard [50], the maximum transmit power is limited to 200 mW (i.e., 23 dBm) for the middle band of the 5 GHz U-NII band, which is suitable for indoor environments.

Consider the same configuration as shown in Fig. 3.7. When the interference signal is RTS, Data, or Ack frames, since these frames are transmitted at or lower than the nominal power level, the analysis presented for RTS-CTS-Data(TPC)-Ack still holds and we have

$$r_i'' \;=\; \sqrt[4]{SIR_{th\_rc,6mbps}} \cdot d_{tx\_rc,6mbps}. \tag{3.13}$$

Comparing Eq. (3.13) with Eq. (3.12), we can see that, as long as $SIR_{th\_rc,6mbps}$ is less than, or equal to, 5 dB — which is true according to [56][3] — the interference range is completely covered by the enlarged NAV set range, and hence, the data frame reception will never be interfered with by any RTS, Data, or Ack frames from neighboring stations.

On the other hand, when the interference signal is the stronger-power-transmitted CTS frames, the condition for an interference to occur becomes

$$\begin{aligned} SIR_{th\_\ell,x} &\geqslant SIR \\ &= \mathcal{P}''_{r,data} - \mathcal{P}''_{r,int(cts)} \\ &= \left(\mathcal{P}''_{t,data} - PL_d\right) - \left(\mathcal{P}''_{t,int(cts)} - PL_{d_i}\right) \\ &= \left(\mathcal{P}_{nom} - PL_{d_{tx\_\ell,x}}\right) - \left(\mathcal{P}_{nom^+} - PL_{d_i}\right) \\ &= PL_{d_i} - PL_{d_{tx\_\ell,x}} - 5 \text{ dB}. \end{aligned} \tag{3.14}$$

Similar to Eq. (3.8)'s equivalency to Eq. (3.9), Eq. (3.14) is equivalent to

$$\begin{aligned} & SIR_{th\_rc,6mbps} \;\geqslant\; PL_{d_i} - PL_{d_{tx\_rc,6mbps}} - 5 \text{ dB} \\ \iff \quad & SIR_{th\_rc,6mbps} \;\geqslant\; \left(\frac{d_i}{d_{tx\_rc,6mbps}}\right)^4 \cdot \frac{1}{3.16} \\ \iff \quad & d_i \;\leqslant\; \sqrt[4]{SIR_{th\_rc,6mbps}} \cdot 1.33\, d_{tx\_\ell,6mbps}, \end{aligned} \tag{3.15}$$

and the radius of the interference range, when the interference is caused by CTS frames,

[3] Our error probability analysis in [56] showed that, when a data frame with 1152-octet payload is transmitted at 6 Mbps and the receiver-side SIR is larger than 5 dB, the PER of the frame is extremely small and, hence, negligible.

is

$$r_i'' = \sqrt[4]{SIR_{th\_rc,6mbps}} \cdot 1.33\, d_{tx\_rc,6mbps} > r_r''. \tag{3.16}$$

Therefore, the data frame reception may still be interfered with by the CTS signals. However, considering the fact that the CTS frames are normally much shorter than the data frames, such interference is not as severe as that caused by the data signals, which may occur when the conventional four-way handshake is used.

3.3.6 Summary

Based on the above interference analysis in 802.11a DCF systems, we summarize the main analysis results as follows:

- Without RTS/CTS support, applying TPC on data transmissions may result in more hidden nodes and aggravate the interference;
- With RTS/CTS support, the "hidden nodes" problem is alleviated and the CCA busy range is completely covered by the NAV set range in 802.11a DCF systems. However, applying TPC on data transmissions may still aggravate the interference because of the enlarged interference range;
- Our enhanced RTS-CTS(strong)-Data(TPC)-Ack mechanism is suitable to accommodate intelligent TPC in 802.11a DCF systems, because it not only allows the data frames to be transmitted at lower power levels to save energy, but also ameliorates the potentially aggravated interference caused by TPC by transmitting the CTS frames at a stronger power level.

3.4 Energy Consumption Analysis of an 802.11a/h DCF System

Before delving into the details of MiSer, we will first analyze the average energy consumed by an 802.11a/h device when it is actively transmitting, receiving, or sensing the

channel, i.e., when it is not in the doze mode. We also list the related notations.

3.4.1 Energy Consumption Model

Since we do not have access to the energy-consumption characteristics of any 802.11a-compliant product currently available in the market, we present a generic energy consumption model for 802.11 devices for our analysis. It is based on the power characteristics of two popular 802.11b-compliant products, the Agere ORiNOCO card [4] and the Intersil Prism II card [31].

Fig. 3.11 shows the simplified block diagram of an 802.11 device. In general, the power consumption is different for the receive mode and the transmit mode, because different circuits are used in different modes. As shown in the figure, the RF power amplifier (PA) is active in the transmit mode only, while the receiving front end (e.g., the low noise amplifier in an Intersil Prism II card) is active only in the receive mode.

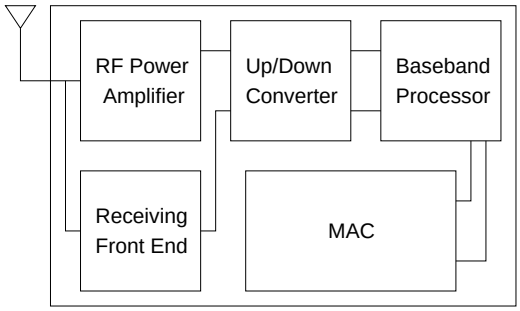


Figure 3.11: Simplified block diagram of an 802.11 device

The power conversion efficiency (η) of a PA is defined as the ratio of the actual signal power emitted from the antenna, or the transmit power ($\mathcal{P}_t$), to the total power consumed by the PA ($\mathcal{P}_{pa}$). Basically, η is a function of $\mathcal{P}_t$, and a PA presents the following non-linearity feature: it achieves high efficiency at high transmit power levels, but the efficiency drops flat at low power levels. The E-P (Efficiency vs. transmit Power) curve varies with the PA design. Based on the E-P curves given in [30, 69, 70], we assume an exponential

E-P curve for the 5 GHz power amplifiers used in the 802.11a/h-compliant devices. Since we are only interested in how MiSer saves energy by combining TPC with PHY rate adaptation, not the exact amount of energy savings, this assumption has little impact on the results and conclusions to be presented in Section 3.6.

Let $\mathcal{P}_{rec}$ denote the power consumption of the receiving front end. In general, $\mathcal{P}_{rec}$ is lower than $\mathcal{P}_{pa}$, and the difference becomes significant when the transmit power is high. Converter, baseband processor, and MAC are considered to be the common components of both receive and transmit circuits, and they are assumed to consume the same amount of power ($\mathcal{P}_{com}$) in both receive and transmit modes. Let $\mathcal{P}_{r\_mode}$ and $\mathcal{P}_{t\_mode}$ be the total power consumption in the receive and transmit modes, respectively. Then, we have

$$\begin{cases} \mathcal{P}_{r\_mode} = \mathcal{P}_{com} + \mathcal{P}_{rec}, \\ \\ \mathcal{P}_{t\_mode}(\mathcal{P}_t) = \mathcal{P}_{com} + \mathcal{P}_{pa} = \mathcal{P}_{com} + \frac{\mathcal{P}_t}{\eta(\mathcal{P}_t)}. \end{cases} \tag{3.17}$$

Furthermore, we assume that, when an 802.11 device is in the idle mode, it presents the same power consumption as when it is in the receive mode.

3.4.2 Energy Consumption Analysis

MAC/PHY Layer Overheads

As shown in Fig. 3.12, in an 802.11 system, each MAC data frame, or MPDU (MAC Protocol Data Unit), consists of the following components:[4] *MAC header*, *frame body* of variable length, and *FCS* (Frame Check Sequence). The MAC overheads due to the MAC header and the FCS are 28 octets in total. Besides, the frame sizes of an Ack frame, an RTS frame, and a CTS frame are 14, 20, and 14 octets, respectively.

[4] Actually, an additional field of "Address 4" appears in the WDS (Wireless Distribution System) data frames being distributed from one AP to another AP. However, since such WDS frames are rarely used, we do not consider the "Address 4" field. Besides, we do not consider the WEP (Wired Equivalent Privacy) option, which may introduce an extra 8-octet overhead.

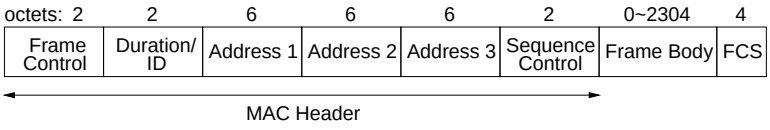


Figure 3.12: Frame format of a data frame MPDU

During the transmission, a PLCP preamble and a PLCP header are added to an MPDU to create a PPDU (PLCP Protocol Data Unit). The PPDU format of the 802.11a PHY is shown in Fig. 3.13, 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 μs) and two repetitions of a long training sequence (4 μs). 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 the 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 μs. 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 3.2 lists the related characteristics for the 802.11a PHY.

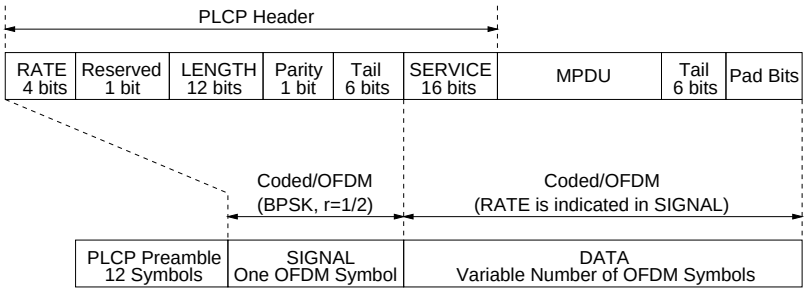


Figure 3.13: PPDU frame format of the 802.11a PHY

Note that, while the data frames can be transmitted at any supported PHY rate, the

Characteristics	Value	Comments
tSlotTime	9 μs	Slot time
tSIFSTime	16 μs	SIFS time
tDIFSTime	34 μs	DIFS = SIFS + 2 $\times$ Slot
cw_{min}	15	min contention window size
cw_{max}	1023	max contention window size
tPLCPPreamble	16 μs	PLCP preamble duration
tPLCP_SIG	4 μs	PLCP SIGNAL field duration
tSymbol	4 μs	OFDM symbol interval

Table 3.2: The 802.11a PHY Characteristics

RTS/CTS frames are always transmitted at 6 Mbps and the Ack frame is required to be transmitted at the highest rate in the BSS basic rate set[5] that is less than, or equal to, the rate of the data frame it is acknowledging. For example, if the BSS basic rate set is $\{$6 Mbps, 12 Mbps, 24 Mbps$\}$ and a data frame is transmitted at 18 Mbps, the corresponding Ack frame will be transmitted at 12 Mbps.

Based on the above analysis, to transmit a data frame with ℓ octets payload over the 802.11a PHY using PHY rate $\mathcal{R}$ and transmit power $\mathcal{P}_t$, the energy consumption is

$$\mathcal{E}_{data}(\ell, \mathcal{R}, \mathcal{P}_t) = \mathcal{T}_{data}(\ell, \mathcal{R}) \cdot \mathcal{P}_{t\_mode}(\mathcal{P}_t), \tag{3.18}$$

where the data transmission duration, $\mathcal{T}_{data}(\ell, \mathcal{R})$, is given by

$$\begin{aligned} \mathcal{T}_{data}(\ell, \mathcal{R}) &= \textit{tPLCPPreamble} + \textit{tPLCP\_SIG} + \left\lceil \frac{28 + (16+6)/8 + \ell}{\textit{BpS}(\mathcal{R})} \right\rceil \cdot \textit{tSymbol} \\ &= 20\mu s + \left\lceil \frac{30.75 + \ell}{\textit{BpS}(\mathcal{R})} \right\rceil \cdot 4\mu s. \end{aligned} \tag{3.19}$$

Note that *BpS*($\mathcal{R}$), the Bytes-per-Symbol information for PHY rate $\mathcal{R}$, is given in Table 3.1. The energy consumed to receive the corresponding Ack frame is

$$\mathcal{E}_{ack} = \mathcal{T}_{ack}(\mathcal{R}') \cdot \mathcal{P}_{r\_mode}, \tag{3.20}$$

[5]BSS (Basic Service Set) is the basic building block of an 802.11 system. It consists of a set of stations controlled by a single coordination function. BSS basic rate set is the set of rates that all the stations in a BSS are capable of using to receive/transmit frames from/to the wireless medium.

where the PHY rate used for the Ack transmission, $\mathcal{R}'$, is determined based on $\mathcal{R}$ according to the rule specified earlier, and the Ack transmission duration is

$$\mathcal{T}_{ack}(\mathcal{R}') = 20\mu s + \left\lceil \frac{16.75}{\mathit{BpS}(\mathcal{R}')} \right\rceil \cdot 4\mu s. \tag{3.21}$$

Similarly, the energy consumption for an RTS frame transmission ($\mathcal{E}_{rts}$) using transmit power $\mathcal{P}_t$ and a CTS frame reception ($\mathcal{E}_{cts}$) can be calculated by

$$\mathcal{E}_{rts} = 52\mu s \cdot \mathcal{P}_{t\_mode}(\mathcal{P}_t), \tag{3.22}$$

and

$$\mathcal{E}_{cts} = 44\mu s \cdot \mathcal{P}_{r\_mode}, \tag{3.23}$$

respectively. Besides, We use $\mathcal{E}_{sifs}$ and $\mathcal{E}_{difs}$ to denote the energy consumptions of an 802.11a device being idle for SIFS time and DIFS time, respectively, and they can be calculated by

$$\mathcal{E}_{sifs} = \mathit{tSIFSTime} \cdot \mathcal{P}_{r\_mode} = 16\mu s \cdot \mathcal{P}_{r\_mode}, \tag{3.24}$$

and

$$\mathcal{E}_{difs} = \mathit{tDIFSTime} \cdot \mathcal{P}_{r\_mode} = 34\mu s \cdot \mathcal{P}_{r\_mode}. \tag{3.25}$$

Backoff Period

The backoff behavior of the 802.11 DCF was described in Section 2.2.2. If the retry counts of a data frame are (SRC, LRC), which means that there have been SRC unsuccessful RTS transmission attempts and LRC unsuccessful data transmission attempts for this frame, then the average energy consumption of the 802.11 device during the next backoff period consists of the following two parts: the energy consumption while the backoff counter is decrementing, which can be calculated by

$$\overline{\mathcal{E}}_{bkoff}(\mathrm{SRC}, \mathrm{LRC}) = \mathcal{P}_{r\_mode} \cdot \mathit{tSlotTime} \cdot \frac{\min\left[2^{\mathrm{SRC+LRC}} \cdot (cw_{min}+1) - 1,\ cw_{max}\right]}{2}, \tag{3.26}$$

and the energy consumption while the backoff counter is frozen due to the busy medium, which is bounded by

$$\overline{\mathcal{E}}_{freeze} \leq \mathcal{P}_{r\_mode} \cdot (N_{sta} - 1) \cdot \\ [P_{c,rts} \cdot 86\mu s + (1 - P_{c,rts}) \cdot (178\mu s + \mathcal{T}_{data}(1500, 6) + \mathcal{T}_{ack}(6))], \quad (3.27)$$

where N_{sta} is the number of contending stations in the network and $P_{c,rts}$ is the RTS collision probability. 86 μs and 178 μs represent an RTS transmission time plus a DIFS time, and an RTS transmission time plus a CTS transmission time plus three SIFS times plus a DIFS time, respectively.

Eq. (3.27) is derived based on the assumption that all the wireless stations are evenly distributed in the network and contending for the wireless medium fairly. So on average, during the backoff period, each of the contending stations attempts its RTS transmission once and succeeds with probability $(1-P_{c,rts})$. We also assume that the worst-case (longest) data transmission time corresponds to a data frame with 1500 octets payload — the Ethernet standard MTU (Maximum Transmission Unit) — transmitted at 6 Mbps.

3.5 MiSer

MiSer [58] is our intelligent TPC mechanism for 802.11a/h DCF systems. In order to deal with the "hidden nodes" problem and the TPC-caused interference, MiSer is deployed in the format of RTS-CTS(strong)-Data(MiSer)-Ack, which was discussed in Section 3.3.5.

MiSer is motivated by [17] and is a simple table-driven approach. The basic idea is that the wireless station computes offline a rate-power combination table indexed by the data transmission status and each entry of the table is the optimal rate-power combination ($<\mathcal{R}^*, \mathcal{P}_t^*>$) in the sense of maximizing the energy efficiency ($\mathcal{J}$) under the corresponding

data transmission status. The *data transmission status* is characterized by a quadruplet (ℓ, s, SRC, LRC), where ℓ is the data payload length, s is the path loss from the transmitter to the receiver, and (SRC, LRC) are the frame retry counts. The *energy efficiency* ($\mathcal{J}$) is defined as the ratio of the expected delivered data payload ($\mathcal{L}$) to the expected total energy consumption ($\mathcal{E}$). This table is then used at runtime to determine the proper PHY rate and transmit power for each data transmission attempt.

3.5.1 Step I: Offline Establishment of the Rate-Power Combination Table

We assume that the transmission error (due to background noise) probabilities of the RTS, CTS, and Ack frames are negligible because of their small frame sizes and robust transmission rates (refer to Section 3.3.3). Then, the table entries of the rate-power combination table are computed as follows.

First, consider the general case when

$$0 \leqslant \text{SRC} < \textit{dot11ShortRetryLimit} \tag{3.28}$$

and

$$0 \leqslant \text{LRC} < \textit{dot11LongRetryLimit}. \tag{3.29}$$

Assume that $<\mathcal{R}, \mathcal{P}_t>$ is selected for the data transmission attempt of status (ℓ, s, SRC, LRC). Also, assume that the future re-transmission attempts, if any, will be made with the most energy-efficient transmission strategies as well. Clearly, the frame delivery is successful only if the RTS transmission succeeds without collision and the data transmission is error-free or results in correctable errors. Otherwise, the station has to re-contend for the medium to re-transmit the frame. In particular, if the delivery failure was due to the RTS collision, the frame retry counts become (SRC+1, LRC); if the delivery failure was due to the erroneous reception of the data frame, the frame retry counts become (SRC, LRC+1).

Based on the above observations, the expected delivered data payload ($\mathcal{L}$) and the expected total energy consumption ($\mathcal{E}$) can be calculated recursively as follows:

$$\begin{aligned}
&\mathcal{L}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \text{LRC}) = \\
&(1 - P_{c,rts}) \cdot [1 - P_{e,data}(\mathcal{R}, \mathcal{P}_t, \ell, s)] \cdot \ell \\
&+ (1 - P_{c,rts}) \cdot P_{e,data}(\mathcal{R}, \mathcal{P}_t, \ell, s) \\
&\quad \cdot \mathcal{L}(\mathcal{R}^*(\ell, s, \text{SRC}, \text{LRC+1}), \mathcal{P}_t^*(\ell, s, \text{SRC}, \text{LRC+1}), \ell, s, \text{SRC}, \text{LRC+1}) \\
&+ P_{c,rts} \cdot \mathcal{L}(\mathcal{R}^*(\ell, s, \text{SRC+1}, \text{LRC}), \mathcal{P}_t^*(\ell, s, \text{SRC+1}, \text{LRC}), \ell, s, \text{SRC+1}, \text{LRC}),
\end{aligned} \tag{3.30}$$

and

$$\begin{aligned}
&\mathcal{E}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \text{LRC}) = \\
&\overline{\mathcal{E}}_{bkoff}(\text{SRC}, \text{LRC}) + \overline{\mathcal{E}}_{freeze} \\
&+ (1 - P_{c,rts}) \cdot [1 - P_{e,data}(\mathcal{R}, \mathcal{P}_t, \ell, s)] \\
&\quad \cdot [\mathcal{E}_{rts-sifs-cts-sifs} + \mathcal{E}_{data}(\mathcal{R}, \mathcal{P}_t, \ell) + \mathcal{E}_{sifs} + \mathcal{E}_{ack} + \mathcal{E}_{difs}] \\
&+ (1 - P_{c,rts}) \cdot P_{e,data}(\mathcal{R}, \mathcal{P}_t, \ell, s) \\
&\quad \cdot [\mathcal{E}_{rts-sifs-cts-sifs} + \mathcal{E}_{data}(\mathcal{R}, \mathcal{P}_t, \ell) + \mathcal{E}_{ack_tout} \\
&\qquad + \mathcal{E}(\mathcal{R}^*(\ell, s, \text{SRC}, \text{LRC+1}), \mathcal{P}_t^*(\ell, s, \text{SRC}, \text{LRC+1}), \ell, s, \text{SRC}, \text{LRC+1})] \\
&+ P_{c,rts} \cdot [\mathcal{E}_{rts} + \mathcal{E}_{cts_tout} \\
&\quad + \mathcal{E}(\mathcal{R}^*(\ell, s, \text{SRC+1}, \text{LRC}), \mathcal{P}_t^*(\ell, s, \text{SRC+1}, \text{LRC}), \ell, s, \text{SRC+1}, \text{LRC})].
\end{aligned} \tag{3.31}$$

$P_{c,rts}$ is the RTS collision probability and varies with the network configuration [8, 77]. $P_{e,data}$, the data transmission error probability, is a function of $\mathcal{R}$, $\mathcal{P}_t$, ℓ, and s, and varies with the wireless channel model [56]. Besides,

$$\mathcal{E}_{rts-sifs-cts-sifs} = \mathcal{E}_{rts} + 2 \cdot \mathcal{E}_{sifs} + \mathcal{E}_{cts}, \tag{3.32}$$

and $\overline{\mathcal{E}}_{bkoff}(\cdot)$, $\overline{\mathcal{E}}_{freeze}$, $\mathcal{E}_{data}(\cdot)$, $\mathcal{E}_{ack}$, $\mathcal{E}_{rts}$, $\mathcal{E}_{cts}$, $\mathcal{E}_{sifs}$, $\mathcal{E}_{difs}$ are given by Eqs. (3.26), (3.27), (3.18), (3.20), (3.22), (3.23), (3.24), and (3.25), respectively. Moreover, since an Ack (CTS) timeout is equal to a SIFS time, plus an Ack (CTS) transmission time, and plus a Slot time, we have

$$\mathcal{E}_{ack\_tout} = \mathcal{E}_{sifs} + \mathcal{E}_{ack} + \textit{tSlotTime} \cdot \mathcal{P}_{r\_mode}, \tag{3.33}$$

and

$$\mathcal{E}_{cts\_tout} = \mathcal{E}_{sifs} + \mathcal{E}_{cts} + \textit{tSlotTime} \cdot \mathcal{P}_{r\_mode}. \tag{3.34}$$

Hence, the energy efficiency ($\mathcal{J}$) is

$$\mathcal{J}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \text{LRC}) = \frac{\mathcal{L}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \text{LRC})}{\mathcal{E}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \text{LRC})}. \tag{3.35}$$

Since there are only finite choices for the PHY rate and the transmit power, we can calculate $\mathcal{J}$ for each rate-power combination, and the pair that maximizes $\mathcal{J}$ is then the most energy-efficient strategy for the data transmission attempt of status (ℓ, s, SRC, LRC):

$$\begin{aligned} &< \mathcal{R}^*(\ell, s, \text{SRC}, \text{LRC}),\ \mathcal{P}_t^*(\ell, s, \text{SRC}, \text{LRC}) > \\ &= \arg \max_{<\mathcal{R}, \mathcal{P}_t>} \mathcal{J}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \text{LRC}). \end{aligned} \tag{3.36}$$

Now, consider the special case when

$$\text{SRC} = \textit{dot11ShortRetryLimit} \tag{3.37}$$

and/or

$$\text{LRC} = \textit{dot11LongRetryLimit}. \tag{3.38}$$

Obviously, since at least one of the frame retry limits has been reached, the data frame will be discarded without any further transmission attempt. Hence, for any $<\mathcal{R}, \mathcal{P}_t>$, we

always have

$$\begin{cases} \mathcal{E}(\mathcal{R}, \mathcal{P}_t, \ell, s, \textit{dot11ShortRetryLimit}, \text{LRC}) = 0, \\ \\ \mathcal{L}(\mathcal{R}, \mathcal{P}_t, \ell, s, \textit{dot11ShortRetryLimit}, \text{LRC}) = 0, \end{cases} \tag{3.39}$$

and

$$\begin{cases} \mathcal{E}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \textit{dot11LongRetryLimit}) = 0, \\ \\ \mathcal{L}(\mathcal{R}, \mathcal{P}_t, \ell, s, \text{SRC}, \textit{dot11LongRetryLimit}) = 0. \end{cases} \tag{3.40}$$

So, by using this special case as the boundary condition, we have fully specified the computation of the rate-power combination table by Eqs. (3.30), (3.31), (3.35), (3.36), (3.39), and (3.40).

3.5.2 Step II: Runtime Execution

Fig. 3.14 shows the pseudo-coded algorithm for MiSer. Before running the program, the wireless station computes the optimal rate-power combination for each set of data payload length (ℓ), path loss (s), and frame retry counts (SRC, LRC). Thus, a rate-power combination table is pre-established and ready for runtime use. The retry counts SRC_{curr} and LRC_{curr} for the frame at the header of the data queue are both set to zero. At runtime, the wireless station estimates the path loss between itself and the receiver, and then selects the rate-power combination $<\mathcal{R}_{curr}, \mathcal{P}_{t,curr}>$ for the current data transmission attempt by a simple table lookup. Note that the rate-power selection is made before the RTS frame is transmitted, so that the Duration/ID information carried in the RTS frame can be properly set according to the PHY rate selection.

As shown in the pseudo-code, if an RTS/CTS frame exchange successfully reserves the wireless medium and an Ack frame is received correctly within the Ack timeout, the wireless station knows that the previous data transmission attempt was successful, and

```
⟨ compute <R*, P_t*> for each set of ℓ, s, SRC, LRC ⟩;
SRC_max := dot11ShortRetryLimit;
LRC_max := dot11LongRetryLimit;
F := the frame at the header of the data queue;
ℓ_curr := DataPayloadLength(F);
SRC_curr := 0;    LRC_curr := 0;

while (the data queue is non-empty) {
   s_curr := the up-to-date path loss estimation;
   R_curr := R*(ℓ_curr, s_curr, SRC_curr, LRC_curr) ;
   P_t,curr := P_t*(ℓ_curr, s_curr, SRC_curr, LRC_curr) ;
   ⟨ an RTS frame is sent to reserve the medium ⟩;
   if (a CTS frame is received correctly)   then {
      ⟨ F is transmitted using <R_curr, P_t,curr> ⟩;
      if (an Ack frame is received correctly)   then {
         SRC_curr := 0;    LRC_curr := 0;
      }
      else   LRC_curr := LRC_curr + 1;
   }
   else   SRC_curr := SRC_curr + 1;
   if (SRC_curr ⩾ SRC_max || LRC_curr ⩾ LRC_max)   then {
      SRC_curr := 0;    LRC_curr := 0;
   }
   if (SRC_curr == 0 && LRC_curr == 0)   then {
      ⟨ remove the header frame from the data queue ⟩;
      ⟨ refresh F and ℓ_curr ⟩;
   }
}
```

Figure 3.14: Pseudo-code of MiSer

resets both retry counts to zero; else, either SRC_{curr} or LRC_{curr} is increased and the wireless station will re-select the rate-power combination for the next transmission attempt of the data frame. If the data frame cannot be successfully delivered after SRC_{max} medium reservation attempts or LRC_{max} data transmission attempts, the frame will be dropped and both SRC_{curr} and LRC_{curr} are reset to zero for the next frame waiting in the data queue.

One important aspect of MiSer is that it shifts the computation burden offline, and hence, simplifies the runtime execution significantly. Therefore, embedding MiSer at the MAC layer has little effect on the performance of higher-layer applications, which is a

desirable feature for any MAC-layer enhancement.

3.5.3 Implementation Issues

Table Establishment

As described in Section 3.5.1, in order to establish the rate-power combination table, a wireless station needs the following information:

- *Network configuration* that indicates the number of contending stations (N_{sta}) and determines the RTS collision probability ($P_{c,rts}$);
- *Wireless channel model* that determines the error performances of the PHY rates ($P_{e,data}$).

There have been a number of papers dealing with the problems of estimating the network configuration [9, 10] or building accurate wireless channel models [6, 15, 76, 80], which are not the focus of this work. Instead, we propose MiSer as a simple and effective TPC mechanism by assuming that the wireless station either has the required knowledge *a priori* or can estimate them.

Path Loss Estimation

At runtime, in order to look up the pre-established rate-power combination table to determine the best transmission strategy for each data frame, a wireless station has to estimate the path loss between itself and the receiver. We have developed a simple path loss estimation scheme, based on the upcoming 802.11h standard [28], as a possible solution.

The 802.11h standard is an extension to the current 802.11 MAC and 802.11a PHY, and one of the key improvements in 802.11h is to enable a wireless station to report its transmit power information in the newly-defined TPC Report element, which includes a Transmit Power field and a Link Margin field. The Transmit Power field simply contains the transmit

power (in dBm) used to transmit the frame containing the TPC Report element, while the Link Margin field contains the link margin (in dB) calculated as the ratio of the received signal strength to the minimum desired by the station.

As specified in the 802.11h standard, the AP in an infrastructure network or a wireless station in an ad hoc network will autonomously include a TPC Report element with the Link Margin field set to zero and containing its transmit power information in the Transmit Power field in any Beacon or Probe Response frame it transmits. A wireless station keeps track of the path loss to the AP, if within an infrastructure network,[6] or the path loss to each neighboring station, if within an ad hoc network, and whenever it receives a Beacon or Probe Response frame, it updates the corresponding path loss value. That is, with the knowledge of the received signal strength (in dBm) via RSSI (Receive Signal Strength Indicator) as well as the transmit power (in dBm) via the TPC Report element found in the frame, the wireless station can calculate the path loss (in dB) from the sending station to itself by performing the simple subtraction. Note that RSSI is one of the RXVECTOR parameters, which is measured and passed to the MAC by the PHY and indicates the energy observed at the antenna used to receive the current frame. Basically, the path loss value(s) maintained in this manner can be used by the wireless station to determine its best transmission strategy.

This path loss estimation scheme is reasonable since with 802.11 systems, the same frequency channel is used for all transmissions in a time-division duplex manner, and hence, the channel characteristics in terms of path loss for both directions are likely to be similar. Moreover, since the Beacon frames are transmitted periodically and frequently, a wireless station is able to update the path loss value(s) in a timely manner.

[6] In an infrastructure network, if a wireless station wants to communicate with another station, the frames must be first sent to the AP, and then from the AP to the destination [50]. Therefore, a wireless station only needs to keep track of the path loss between itself and the AP.

Optimality of MiSer

As shown in Section 3.5.1, MiSer is designed to be the optimal low-energy transmission strategy for 802.11a/h. However, its optimality is based on the assumption of perfect knowledge on the network configuration and the wireless channel models and accurate estimation of the path loss. Therefore, MiSer can be viewed as a benchmark study on the energy-efficient frame transmissions in 802.11a/h DCF systems, which answers the important question of *what is the upper bound on energy conservation by applying TPC on data transmissions.* In reality, with less accurate knowledge on the required information, MiSer will be inevitably less effective.

3.6 Performance Evaluation

We evaluate the performance of MiSer using the ns-2 simulator [2] after enhancing the original 802.11 DCF module of ns-2 to support the 802.11a/h PHY, PHY rate adaptation, and TPC (Transmit Power Control).

3.6.1 Simulation Setup

In the simulation, we use 15 dBm as the nominal transmit power, and a TPC-enabled 802.11a/h device is allowed to choose any one of the 31 power levels (from -15 dBm to 15 dBm with 1 dBm gaps) to transmit a data frame. We assume an AWGN (Additive White Gaussian Noise) wireless channel model and the background noise level is set to -93 dBm. The exponential E-P curve for the 5 GHz PA of the simulated 802.11a/h devices is

$$\eta(\mathcal{P}_t) = 0.02 \cdot 5^{\frac{\mathcal{P}_t}{15}}, \tag{3.41}$$

where $\mathcal{P}_t$ is in dBm. Besides, we use a log-distance path loss model with path loss exponent of four to simulate the indoor office environment, and set the carrier sensing threshold

to -91 dBm, meaning that, when the distance between two stations is larger than 28.6 meters, the resulting path loss is larger than 106 ($= 15 + 91$) dB and these two stations are hidden to each other.

In the first part of the simulation, we compare MiSer against four testing schemes with RTS/CTS support: the PHY rate adaptation scheme without TPC (RA), and three single-rate TPC schemes using PHY rate 6 Mbps (Tpc/R6), 24 Mbps (Tpc/R24), and 54 Mbps (Tpc/R54), respectively. The comparison metrics are the aggregate goodput (in Mbps) and the delivered data per unit of energy consumption (in MBits/Joule), which is calculated as the ratio of the total amount of data delivered by the transmitter stations over their total energy consumption. Note that the larger this value, the more energy-efficient a scheme gets. We conduct the simulation with various network topologies and data payload lengths.

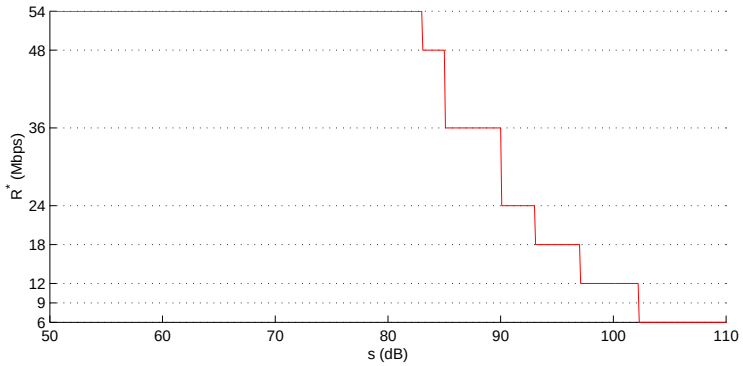
In the second part of the simulation, we compare MiSer against two schemes without RTS/CTS support: the rate-power adaptation scheme (DA-I) and the rate adaptation only scheme (DA-II). In addition to the aggregate goodput and the delivered data per unit of energy consumption, we also compare the frame collision probability for the testing schemes.

Each simulation run lasts 10 seconds in an 802.11a/h DCF system with eight transmitter stations contending for the shared medium. Each station transmits in a greedy mode, i.e., its data queue is never empty, and all the data frames are transmitted without fragmentation. The frame size is 1500 octets unless specified otherwise.

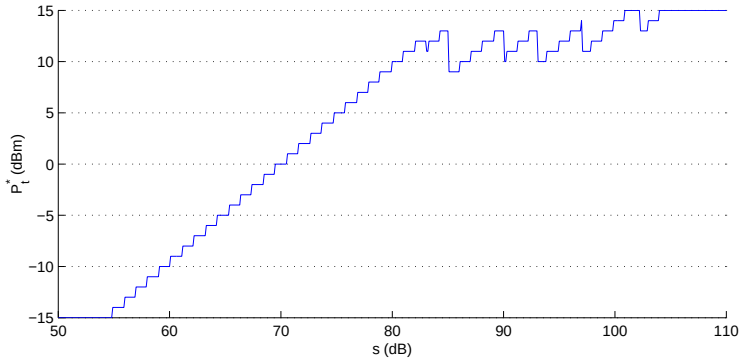
3.6.2 MiSer's Rate-Power Combination Table

Recall that MiSer's rate-power combination table is indexed by (ℓ, s, SRC, LRC), and hence, is four-dimensional. Fig. 3.15 shows a snapshot of this table when $\ell = 1500$ and (SRC, LRC) $=$ (0, 0). The optimal combinations of PHY rate and transmit power, which

achieve the most energy-efficient data transmissions, under different path loss (s) conditions are shown in Fig. 3.15(a) and (b), respectively. For example, when $s = 80$ dB, this figure reads that <54 Mbps, 9 dBm> is the most energy-efficient transmission strategy.



(a) PHY rate selection



(b) transmit power selection

Figure 3.15: A snap shot of MiSer's rate-power combination table when $\ell = 1500$ and (SRC, LRC) = (0, 0)

We make two observations from Fig. 3.15. First, when the path loss is large, the lower PHY rates are preferred as they are more robust and have better error performances. On the other hand, when the path loss is small, higher PHY rates are used to save energy since the duration of a single transmission attempt is shorter. Second, a low transmit power does not necessarily save energy. This is because, with the same PHY rate, using a lower transmit power may lead to less energy consumption in a single transmission attempt, but the resultant low SNR (Signal-to-Noise Ratio) at the receiver side may cause more

re-transmissions and, hence, more total energy consumption.

The snapshots for other ℓ and (SRC, LRC) values can be viewed approximately as shifted versions of Fig. 3.15. In general, when a data frame carried a larger payload ($\ell\uparrow$) or less transmission attempts remain for a data frame (SRC$\uparrow$ and/or LRC$\uparrow$), the figure shifts left and a more conservative combination (i.e., lower rate and/or higher power) is selected under the same path loss condition; otherwise, it shifts right.

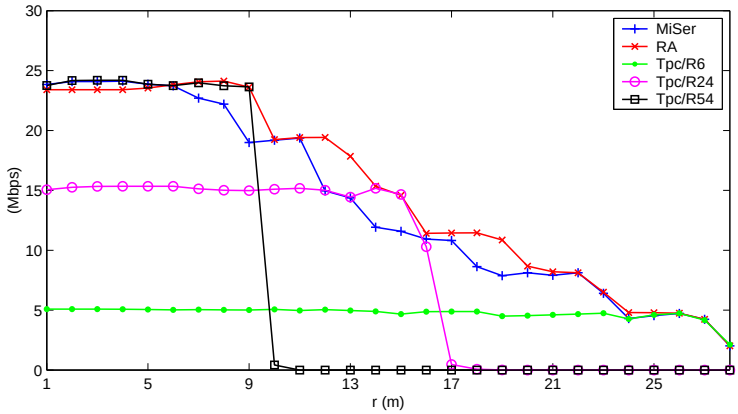
Note that RA's rate adaptation table or Tpc/Rx's ($x = 6$, 24, or 54) power adaptation table are computed in the same way as MiSer's rate-power combination table, except fixing the transmit power to 15 dBm or the transmission rate to x Mbps, respectively. Moreover, in order to have a fair evaluation on the effect of RTS/CTS support, we simply let DA-I and DA-II use MiSer's rate-power combination table to determine their transmission strategies.

3.6.3 MiSer vs. Schemes with RTS/CTS Support

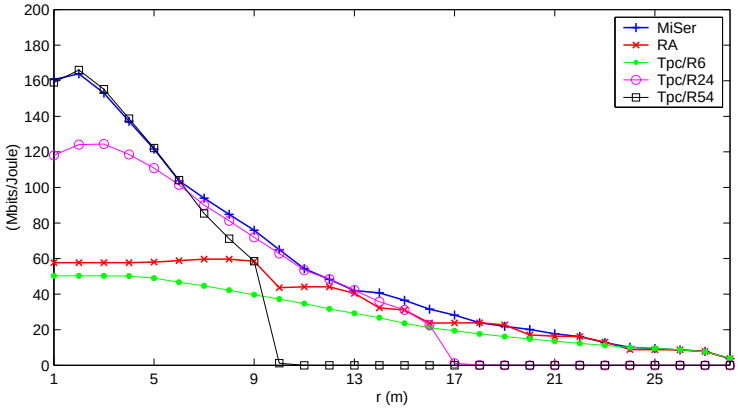
Star topologies with varying radius

We first compare the testing schemes in the star-topology networks, where eight transmitter stations are evenly spaced on a circle around one common receiver with the radius of r ($1 \leqslant r \leqslant 28$) meters, and all the stations are static. Although ideal star-topology networks are rarely found in a real world, the simulation results plotted in Fig. 3.16 help us understand better how TPC adapts to the path loss variation and why MiSer is superior to other simulated transmission strategies, thanks to the symmetric station deployment of star-topology networks, and hence, are valuable.

In general, as r increases, both the aggregate goodput and the delivered data per Joule decrease for all testing schemes. This is because more robust transmission strategies (i.e., lower rate and/or higher power) are used to deal with the increasing number of hidden nodes and a larger path loss between the transmitter and the receiver. However, different schemes show different decreasing curves determined by their respective design philoso-



(a) aggregate goodput



(b) delivered data per unit of energy consumption

Figure 3.16: Comparison for star-topology networks (various radius)

phies, which are discussed next. In order to have a better understanding of the figure, we list, in Table 3.3, the rate-power selections by each testing scheme, when $r = 5$, 9, 12, and 28, respectively.

RA achieves the highest aggregate goodput because its constant use of the strong 15 dBm transmit power allows it to choose the highest possible rate to transmit a data frame. On the other hand, since RA does not support TPC, so even within a small network, it still has to transmit a frame using a higher power than necessary over a short distance, hence consuming more energy. For example, as shown in Table 3.3, when $r = 5$, MiSer selects the same 54 Mbps rate as RA, but a much lower tranmit power level at

r (m)	rate-power selection (<Mbps, dBm>)			
	5	9	12	28
MiSer	<54, 5>	<36, 9>	<24, 11>	<6, 15>
RA	<54, 15>	<54, 15>	<36, 15>	<6, 15>
Tpc/R6	<6, -13>	<6, -3>	<6, 2>	<6, 15>
Tpc/R24	<24, -3>	<24, 6>	<24, 11>	<24, 15>
Tpc/R54	<54, 5>	<54, 15>	<54, 15>	<54, 15>

Table 3.3: Example Rate-Power Selections (ℓ = 1500 and (SRC, LRC) = (0, 0))

5 dBm. As a result, RA yields much lower delivered data per Joule than MiSer when r is small.

Tpc/R6 transmits all the data frames at the lowest 6 Mbps, and hence, results in the lowest aggregate goodput when r is small. As r increases, Tpc/R6 adjusts its transmit power adaptively such that the receiver-side SNR is maintained at a relatively stable level. For example, as shown in Table 3.3, when r increases from 5 to 9, 12, and 28, Tpc/R6 increases its transmit power from -13 dBm to -3 dBm, 2 dBm, and 15 dBm, respectively. Therefore, combined with rate 6 Mbps' strong error-correcting capability, Tpc/R6 shows an almost flat aggregate-goodput curve but a decreasing curve for the delivered data per Joule until $r = 28$, when even the most conservative combination of 6 Mbps and 15 dBm is still not robust enough to combat the resulting high path loss.

Tpc/R54 transmits all the data frames at the highest 54 Mbps. Similar to Tpc/R6, it also has a flat aggregate-goodput curve when r is small. However, due to rate 54 Mbps' poorest error-correcting capability, the aggregate-goodput curve starts dipping at a much smaller r value of 10. Actually, when $r > 10$, all the transmission attempts fail and the aggregate goodput drops to zero. Similar observations can be made for Tpc/R24 as well, which is a compromise between Tpc/R6 and Tpc/R54.

So we can see that, because of fixing the transmission rate, a single-rate TPC scheme either suffers a reduced transmission range (e.g., Tpc/R24 and Tpc/R54) or has to stick

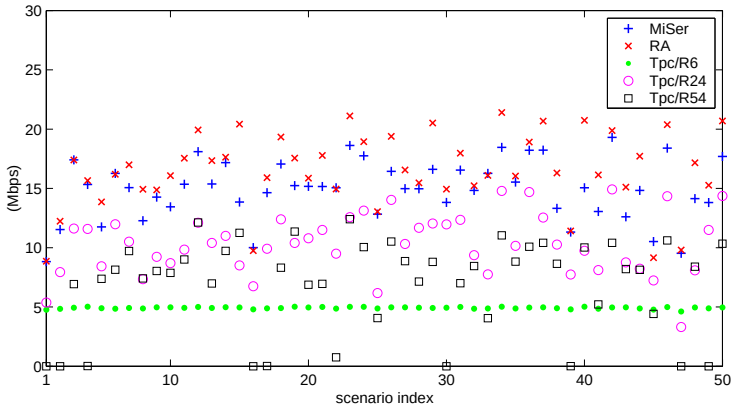
with a low transmission rate (e.g., Tpc/R6).

MiSer achieves the highest delivered data per Joule because of its adaptive use of (1) the energy-efficient combination of high rate and low power when r is small, and (2) the robust combination of low rate and high power when r is large. The key idea is to select the optimal rate-power combination, rather than the PHY rate or the transmit power alone, to minimize the energy consumption. Therefore, under certain path loss conditions (e.g., $r = 9$ in Table 3.3), MiSer may choose a lower rate than RA but with weaker transmit power. As a result, MiSer shows an aggregate goodput curve slightly lower than that of RA. Note that MiSer has the same transmission range as RA and Tpc/R6, since a transmitter station that supports MiSer can always lower the PHY rate and/or increase the transmit power, whenever necessary, to communicate with a far-away receiver station. Another observation in Fig. 3.16 is that, when 6 Mbps (or 24 Mbps, 54 Mbps) or 15 dBm is part of the optimal rate-power selections, MiSer is indeed equivalent to Tpc/R6 (or Tpc/R24, Tpc/R54) or RA, which is evidenced by the partial overlapping in both their aggregate-goodput curves and their curves for the delivered data per Joule.

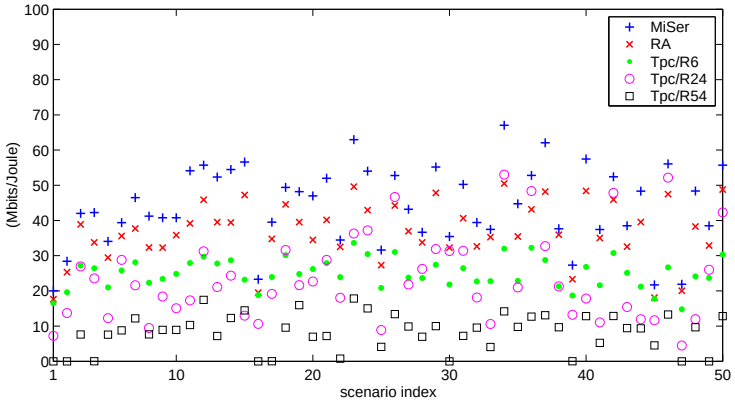
Random topologies with 50 different scenarios

We also evaluate and compare the performances of the testing schemes in randomly-generated network topologies: the eight transmitter stations and their (different) respective receivers are randomly placed within a (40 m $\times$ 40 m) flat area, and all the stations are static. We simulate 50 different scenarios and the results are plotted in Fig. 3.17.

We have three observations. First, MiSer and RA are significantly better than the single-rate TPC schemes, in terms of both the aggregate goodput and the delivered data per Joule, in each simulated random topology. This is because the inevitable low transmission rate or reduced transmission range of a single-rate TPC scheme, where the latter may



(a) aggregate goodput



(b) delivered data per unit of energy consumption

Figure 3.17: Comparison for random-topology networks (50 different scenarios)

cause more potential transmission failures, results in poor aggregate-goodput and energy-efficiency performances. On the other hand, both MiSer and RA are able to perform PHY rate adaptation, which adjusts the transmission rate dynamically to the path loss variation.

Second, MiSer achieves comparable aggregate goodput with RA while delivering about 20% (on average) more data per unit of energy consumption than RA. Actually, the energy saving by MiSer over RA could be more significant if the network size is smaller. This is because, in a smaller network, the transmitter and the receiver are, on average, closer to each other, which corresponds to a smaller path loss value. As a result, MiSer may choose a much lower transmit power (than 15 dBm) to transmit a frame, thus saving more energy.

On the other hand, when the network size gets larger, the energy-efficiency performances of MiSer and RA become comparable.

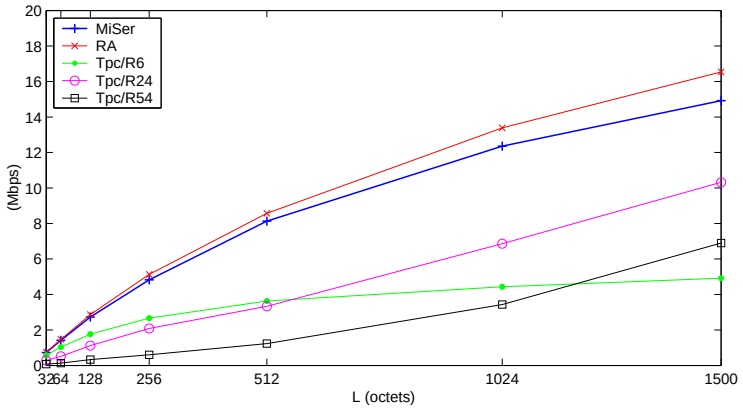
Third, Tpc/R6 produces near-constant aggregate goodput regardless of the network topology, which is consistent with a similar observation in Fig. 3.16. Besides, unlike in the small star-topology networks, where Tpc/R54 has the best energy performance, Tpc/R54 has the lowest delivered data per Joule in every scenario due to the arbitrary station locations in random-topology networks. Particularly, in 10 of the 50 simulated scenarios, Tpc/R54 results in almost zero aggregate goodput.

Random topologies with varying data payloads

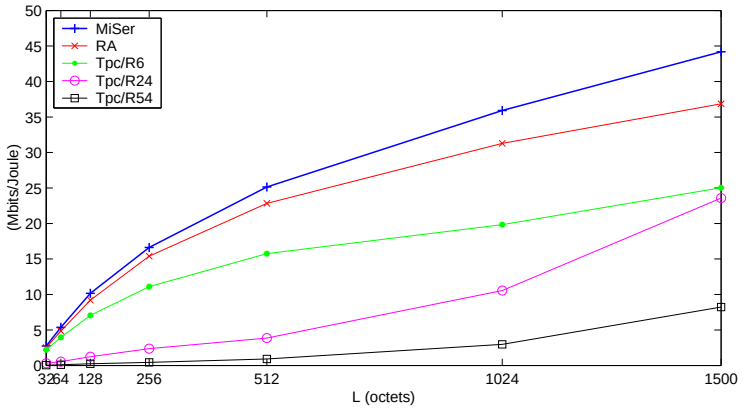
Fig. 3.18 shows the simulation results for random-topology networks with various data payloads. The simulated data payload lengths are 32, 64, 128, 256, 512, 1024, and 1500 octets. Again, each point in the figure is averaged over 50 simulation runs.

Since the RTS/CTS frames are always transmitted at 6 Mbps, the RTS/CTS overhead per data transmission attempt is independent of the payload length. Moreover, there are a number of fixed per-frame overheads such as the PLCP preamble/header, the MAC header, the FCS, and etc. Hence, both the aggregate goodput and the delivered data per Joule increase with the data payload length for all testing schemes. As expected, MiSer has the best energy-efficiency performance, and the gap between MiSer and RA becomes bigger as the data payload length increases. This is because, with the same PHY rate, a larger data payload results in a longer transmission time, during which MiSer may use low transmit power to save more energy. Moreover, RA outperforms single-rate TPC schemes in terms of both goodput and energy consumption due to PHY rate adaptation.

It is interesting to see that the aggregate goodput curves of three single-rate TPC schemes intersect with each other in Fig. 3.18(a). When the data payload is less than



(a) aggregate goodput



(b) delivered data per unit of energy consumption

Figure 3.18: Comparison for various data payloads (average over 50 random topologies)

600 octets, Tpc/R6 yields the best goodput performance, while, when the data payload increases to 1500 octets, both Tpc/R24 and Tpc/R54 achieve better aggregate goodput than Tpc/R6, however, with much higher transmit power. This is because the strong error-correcting capability of rate 6 Mbps allows Tpc/R6 to transmit a large-payload data frame at a very low power level. In Fig. 3.18(b), when $\ell = 1500$, both Tpc/R24 and Tpc/R54 have smaller values of the delivered data per Joule than Tpc/R6, which, in turn, supports the above explanation. Note that, under different network configurations, the relative positions of these three aggregate goodput curves may vary.

3.6.4 MiSer vs. Schemes without RTS/CTS Support

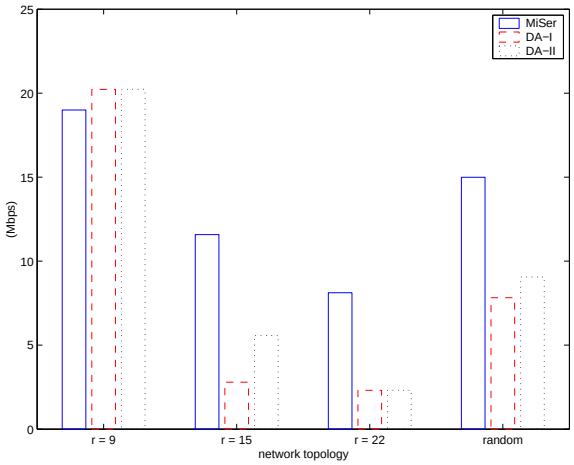
We now compare the performance of MiSer against schemes without RTS/CTS support. We introduce a new measure called the *hidden node ratio* (δ) of the network, which is defined as the ratio of the number of hidden nodes (average over all transmitter stations) to the total number of contending stations. Clearly, the δ value varies with the network topology, the network size, and the transmit power. We evaluate the performances of the testing schemes in star-topology networks of different sizes, and Table 3.4 lists the r values and the corresponding hidden node ratios when different testing schemes are used. We also compare the testing schemes in randomly-generated network topologies that were used in the previous section. The comparison results are plotted in Fig. 3.19.

r (m)	rate-power selection (<Mbps, dBm>) and δ					
	MiSer		DA-I		DA-II	
9	<36, 9>	0	<36, 9>	0	<36, 15>	0
15	<18, 12>	1/7	<18, 12>	3/7	<18, 15>	1/7
22	<12, 15>	5/7	<12, 15>	5/7	<12, 15>	5/7

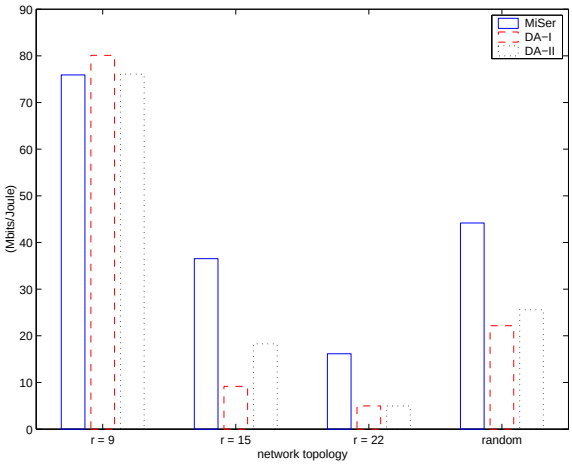
Table 3.4: Rate-Power Selections by Three Testing Schemes and Resultant Hidden Node Ratios in Star-Topology Networks

We have two observations. First, when there are no hidden nodes in the network ($r = 9$), all three schemes result in similar frame collision probabilities, and MiSer yields a lower aggregate goodput than DA-I/II due to the additional RTS/CTS overhead. However, since MiSer is able to select a lower power level (at 9 dBm) for its data transmissions, it shows comparable energy-efficiency performance with DA-II that always transmits at 15 dBm.

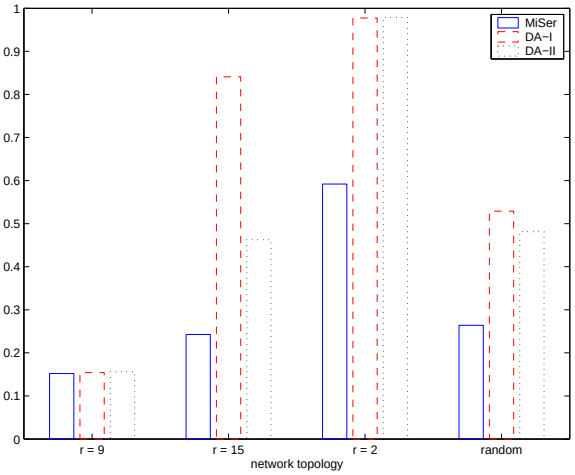
Second, when there are hidden nodes in the network ($r = 15$ or 22, or random), the performances of all three schemes degrade. With a larger hidden node ratio, more stations are hidden to each other in the network and the frame collision probability increases, thus the performance degrades even more. MiSer is less affected by the presence of hidden



(a) aggregate goodput



(b) delivered data per unit of energy consumption



(c) frame collision probability

Figure 3.19: Comparison of MiSer against schemes without RTS/CTS support

nodes than DA-I/II because, by exchanging the RTS/CTS frames to reserve the wireless channel before the actual data transmissions, the collisions can only occur to the RTS frames that are much shorter than the data frames. For this reason, MiSer outperforms DA-I/II significantly in terms of both the aggregate goodput and the delivered data per unit of energy consumption.

As discussed in Sections 3.1.2 and 3.3.2, one potential problem of applying TPC on data transmissions without RTS/CTS support is that it might result in more hidden nodes.

We observe such scenario in the simulated network of $r = 15$. In this case, DA-I selects the rate-power combination of 18 Mbps and 12 dBm for the data transmissions, while MiSer and DA-II contend for the wireless channel by transmitting its RTS or data frames at the 15 dBm power level. As a result, when DA-I is used, the hidden node ratio of the network becomes 3/7 instead of 1/7. The fact explained above is supported by the drastically higher frame collision probability for DA-I when $r = 15$.

3.6.5 Summary

Based on the observations from the simulation results, we summarize MiSer as follows:

- In order to save energy by applying TPC on data transmissions, RTS/CTS support is essential to alleviate the "hidden nodes" problem;
- MiSer is deployed in the format of RTS-CTS(strong)-Data(MiSer)-Ack;
- MiSer is significantly better than any other scheme (with RTS/CTS support) that simply adapts the PHY rate or adjusts the transmit power;
- PHY rate adaptation is very effective in saving energy and plays an important role in MiSer;
- Applying MiSer does not affect the transmission range;
- MiSer is most suitable for data communications with large data payloads.

3.7 Conclusion

In this chapter, we investigate the problem of minimizing the communication energy consumption in 802.11a/h DCF systems. Based on the analysis of the relationship among

different radio ranges and TPC's effects on the interference in 802.11a DCF systems, we propose MiSer, a novel intelligent TPC mechanism, as an optimal solution. The key idea of MiSer is to combine TPC with PHY rate adaptation, so that the most energy-efficient rate-power combination can be adaptively selected for each data transmission attempt. It establishes an optimal rate-power combination table before the communication starts, which shifts the computation burden offline, and hence, simplifies the runtime execution (to simple table lookups) significantly. MiSer is deployed in the format of RTS-CTS(strong)-Data(MiSer)-Ack to alleviate the "hidden nodes" problem and to ameliorate the TPC-caused interference in the network.

Our in-depth simulation shows that MiSer is significantly better than the schemes without RTS/CTS support in the presence of hidden nodes, which are often found in the real networks. Moreover, compared with other schemes with RTS/CTS support, MiSer clearly outperforms the single-rate TPC schemes and delivers about 20% more data per unit of energy consumption than the PHY rate adaptation scheme without TPC.

CHAPTER IV

AN INTELLIGENT LINK ADAPTATION SCHEME FOR 802.11A

4.1 Introduction

The 802.11 PHYs provide multiple transmission modes/rates by using different modulation schemes and different error correcting codes. The mechanism to select one out of multiple available transmission modes/rates at a given time is often 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 modes/rates, the 802.11a PHY introduces an even bigger challenge for the link adaptation algorithm design.

In this chapter, we present a generic method to analyze the goodout performance of an 802.11a DCF system, and derives 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 inter-frame intervals, the Ack transmission time, and the potential re-transmission times. Based on the goodput analysis, we propose a novel link adaptation scheme, called *ILA* (Intelligent Link Adaptation) [63], 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 re-transmissions. So, there is an inherent tradeoff, and the key idea of ILA is a table-driven approach similar to MiSer's. It pre-establishes a PHY mode table indexed by the system status triplet that 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. At run-time, 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 chapter, 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.

The authors of [42] analyzed the goodput performance of an 802.11 system using Lucent Technologies' WaveLAN devices. Since the 802.11a PHY was not available at that time, the authors assumed the fixed QPSK (Quadrature Phase Shift Keying) modulation and analyzed the goodput performance for different sizes of MSDUs (MAC Service Data Units), which are generated by the 802.2 LLC (Logical Link Control) sublayer. Actually,

as indicated in the 802.11 standard, an MSDU can be fragmented further into smaller MAC frames, i.e., MPDUs (MAC Protocol Data Units), for transmission. In [56], 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 MSDU 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 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 ILA, our new MPDU-based link adaptation scheme.

The rest of this chapter is organized as follows. Section 4.2 briefly recaps the 802.11 DCF. The effective goodput analysis of an 802.11a DCF system is shown in Section 4.3. Section 4.4 describes the details of ILA and discusses the implementation issues. Section 4.5 gives an example on how to establish the best PHY mode table in ILA. Section 4.6 presents and discusses the evaluation results. Finally, this chapter concludes with Section 4.7.

4.2 System Overview

The 802.11 DCF was detailed in Sections 2.2 and 3.2. For completeness, we briefly recap the DCF operation here. The kernel idea of the DCF is that all the wireless stations are contending for the shared wireless medium using a scheme called CSMA/CA (Carrier-Sense Multiple Access with Collision Avoidance). CSMA/CA requires a wireless station to sense the medium both physically and virtually before any transmission attempt. If the wireless medium appears to be idle for longer than DIFS (Distributed Inter-Frame Space)

time, the station may transmit immediately without any delay. Otherwise, the station has to wait until the medium is cleared for DIFS time, then backs off a random interval before its transmission attempt.

The SIFS (Short Inter-Frame Space), which is smaller than the DIFS, is the time interval used between transmissions within a frame exchange sequence, i.e., a two-way Data-Ack handshake or a four-way RTS-CTS-Data-Ack handshake. The timing of a successful two-way frame exchange in an 802.11 DCF system is shown in Fig. 4.1. If an Ack frame is received in error, i.e., received with an incorrect FCS (Frame Check Sequence), the transmitter will re-contend for the medium to re-transmit the frame after an EIFS (Extended Inter-Frame Space) interval, as shown in Fig. 4.2. On the other hand, if no Ack frame is received due possibly to an erroneous reception of the preceding data frame, as shown in Fig. 4.3, the transmitter will contend again for the medium to re-transmit the frame after an Ack timeout.

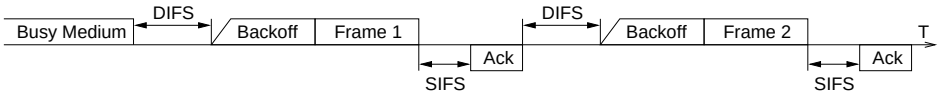


Figure 4.1: Timing of a successful two-way frame exchange under the DCF

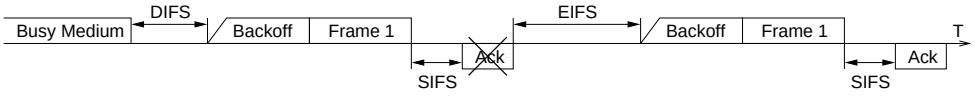


Figure 4.2: Re-transmission due to Ack transmission failure

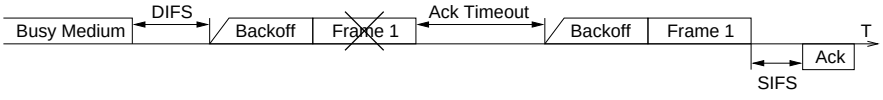


Figure 4.3: Re-transmission due to an erroneous Data frame reception

The 802.11 DCF 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*.

4.3 Goodput Analysis of an 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 the transmission strategy. 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 describing the details of ILA, 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}.

4.3.1 MAC/PHY Layer Overheads

The MAC/PHY layer overheads of an 802.11a system were analyzed in Section 3.4.2. As shown in Fig. 3.12, in an 802.11 system, each MPDU consists of the following com-

ponents: *MAC header*, *frame body* of variable length, and *FCS*. The MAC overheads due to the MAC header and the FCS are 28 octets in total. The frame size of an Ack frame is 14 octets. Besides, the PPDU format of the 802.11a PHY is shown in Fig. 3.13, and the 802.11a PHY characteristics are listed in Table 3.2.

Recall that, while the data frames can be transmitted at any supported PHY rate, all the control frames have to be transmitted at one of the rates in the BSS basic rate set so that they can be understood by all the stations in the same network. In particular, an Ack frame is required to 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 the BSS basic rate set is {6 Mbps, 12 Mbps, 24 Mbps}[1] and a data frame is transmitted at 18 Mbps, the corresponding Ack frame will be transmitted at 12 Mbps.

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

$$\begin{aligned} \mathcal{T}_{data}(\ell, m) &= \textit{tPLCPPreamble} + \textit{tPLCP\_SIG} + \left\lceil \frac{28 + (16+6)/8 + \ell}{\textit{BpS}(m)} \right\rceil \cdot \textit{tSymbol} \\ &= 20\mu s + \left\lceil \frac{30.75 + \ell}{\textit{BpS}(m)} \right\rceil \cdot 4\mu s. \end{aligned} \tag{4.1}$$

Note that *BpS*(m), the Bytes-per-Symbol information for PHY mode m, is given in Table 3.1. Similarly, the transmission duration for an Ack frame using PHY mode m' is

$$\begin{aligned} \mathcal{T}_{ack}(m') &= \textit{tPLCPPreamble} + \textit{tPLCP\_SIG} + \left\lceil \frac{14 + (16+6)/8}{\textit{BpS}(m')} \right\rceil \cdot \textit{tSymbol} \\ &= 20\mu s + \left\lceil \frac{16.75}{\textit{BpS}(m')} \right\rceil \cdot 4\mu s. \end{aligned} \tag{4.2}$$

4.3.2 Backoff Delay

According to the description given in Section 2.2.2 on the backoff behavior of the 802.11 DCF, the average backoff interval before the i^{th} transmission attempt, or equiva-

[1] {6 Mbps, 12 Mbps, 24 Mbps} is the set of the 802.11a mandatory data rates, and it will be assumed to be the BSS basic rate set in the example and simulation of this chapter.

lently, the $(i-1)^{th}$ re-transmission attempt, denoted by $\overline{\mathcal{T}}_{bkoff}(i)$, can be calculated by

$$\overline{\mathcal{T}}_{bkoff}(i) = \frac{\min\left[2^{i-1}\cdot(cw_{min}+1)-1,\ cw_{max}\right]}{2}\cdot \textit{tSlotTime}. \tag{4.3}$$

4.3.3 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 earlier. In the 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

$$P_{s,xmit}(\ell,s,m) = \left[1-P_{e,data}(\ell,s,m)\right]\cdot\left[1-P_{e,ack}(s,m')\right], \tag{4.4}$$

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 the wireless channel model.

Now, let's consider the entire delivery process of the data frame. Since the frame is allowed to be attempted transmission up to n_{max} times, 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

$$P_{succ}(\ell,\hat{s},\hat{m}) = 1 - \prod_{i=1}^{n_{max}}\left[1-P_{s,xmit}(\ell,s_i,m_i)\right]. \tag{4.5}$$

By referring to Fig. 4.1, 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, and then execute a backoff procedure before the re-transmission (see Figs. 4.2 and 4.3). According to the 802.11 standard [24], 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[2]

$$\mathcal{D}_{succ|\ell,\hat{s},\hat{m}} = \sum_{n=1}^{n_{max}} P[n|succ](\ell,\hat{s},\hat{m}) \cdot \Bigg\{ \sum_{i=2}^{n} \left[\overline{\mathcal{D}}_{wait}(i) + \overline{\mathcal{T}}_{bkoff}(i) + \mathcal{T}_{data}(\ell, m_i) \right] + \overline{\mathcal{T}}_{bkoff}(1) + \mathcal{T}_{data}(\ell, m_1) + \textit{tSIFSTime} + \mathcal{T}_{ack}(m'_n) + \textit{tDIFSTime} \Big\}, \quad (4.6)$$

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

$$P[n|succ](\ell,\hat{s},\hat{m}) = \frac{P_{s,xmit}(\ell, s_n, m_n) \cdot \prod_{i=1}^{n-1} [1 - P_{s,xmit}(\ell, s_i, m_i)]}{P_{succ}(\ell,\hat{s},\hat{m})} \quad (4.7)$$

and

$$\overline{\mathcal{D}}_{wait}(i) = \frac{P_{e,data}(\ell, s_{i-1}, m_{i-1})}{1 - P_{s,xmit}(\ell, s_{i-1}, m_{i-1})} \cdot \left[\textit{tSIFSTime} + \mathcal{T}_{ack}(m'_{i-1}) + \textit{tSlotTime} \right] + \frac{[1 - P_{e,data}(\ell, s_{i-1}, m_{i-1})] \cdot P_{e,ack}(s_{i-1}, m'_{i-1})}{1 - P_{s,xmit}(\ell, s_{i-1}, m_{i-1})} \cdot \left[\textit{tSIFSTime} + \mathcal{T}_{ack}(m'_{i-1}) + \textit{tSIFSTime} + \mathcal{T}_{ack}(1) + \textit{tDIFSTime} \right]. \quad (4.8)$$

$\mathcal{T}_{data}(\cdot)$, $\mathcal{T}_{ack}(\cdot)$, and $\overline{\mathcal{T}}_{bkoff}(\cdot)$ are given by Eqs. (4.1), (4.2), and (4.3), respectively. On the other hand, the average time wasted to attempt transmission of the data frame n_{max}

[2]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* (9μs), thus having almost no effect on the goodput analysis.

times in error is given by

$$\mathcal{D}_{fail|\ell,\hat{s},\hat{m}} = \sum_{i=1}^{n_{max}} \left[\overline{\mathcal{T}}_{bkoff}(i) + \mathcal{T}_{data}(\ell, m_i) + \overline{\mathcal{D}}_{wait}(i+1)\right]. \tag{4.9}$$

The expected effective goodput can then be calculated as:

$$\begin{aligned}
\mathcal{G}(\ell,\hat{s},\hat{m}) &= \frac{\ell}{\sum_{k=0}^{\infty}\left[(1-P_{succ}(\ell,\hat{s},\hat{m}))^k \cdot P_{succ}(\ell,\hat{s},\hat{m}) \cdot (k\cdot\mathcal{D}_{fail|\ell,\hat{s},\hat{m}} + \mathcal{D}_{succ|\ell,\hat{s},\hat{m}})\right]} \\
&= \frac{\ell}{\frac{1-P_{succ}(\ell,\hat{s},\hat{m})}{P_{succ}(\ell,\hat{s},\hat{m})}\cdot\mathcal{D}_{fail|\ell,\hat{s},\hat{m}} + \mathcal{D}_{succ|\ell,\hat{s},\hat{m}}} \\
&= \frac{P_{succ}(\ell,\hat{s},\hat{m})\cdot\ell}{(1-P_{succ}(\ell,\hat{s},\hat{m}))\cdot\mathcal{D}_{fail|\ell,\hat{s},\hat{m}} + P_{succ}(\ell,\hat{s},\hat{m})\cdot\mathcal{D}_{succ|\ell,\hat{s},\hat{m}}} \\
&= \frac{E[data](\ell,\hat{s},\hat{m})}{E[\mathcal{D}_{data}](\ell,\hat{s},\hat{m})}.
\end{aligned} \tag{4.10}$$

Eq. (4.10) is based on the fact that with probability $\left[(1-P_{succ}(\ell,\hat{s},\hat{m}))^k \cdot P_{succ}(\ell,\hat{s},\hat{m})\right]$, 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 zero 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.

4.4 ILA

From the above goodput analysis, we observe that, to deliver a data frame over a wireless channel, the higher the PHY modes are used ($\hat{m}\uparrow$), the shorter the expected transmission time will be ($E[\mathcal{D}_{data}]\downarrow$), but 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

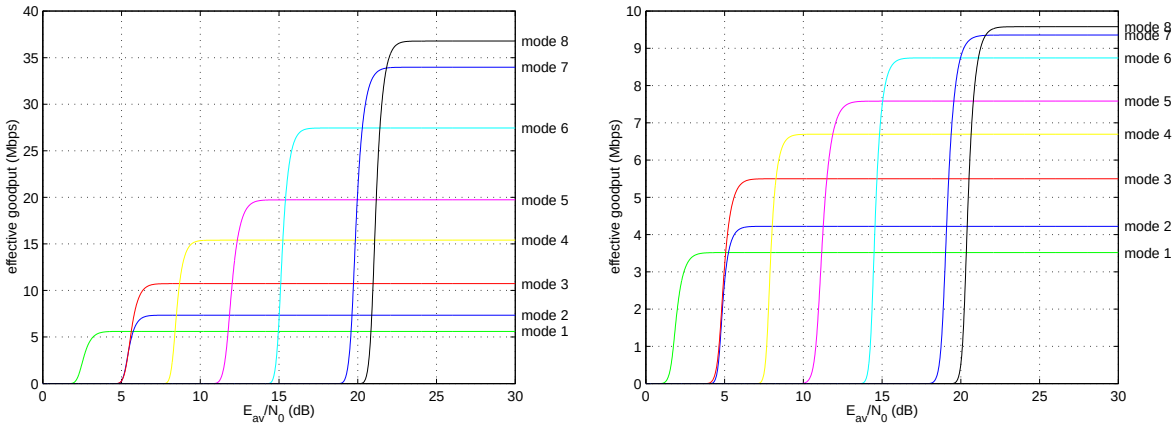
exists a corresponding set of PHY modes that maximize the expected effective goodput. Such set of PHY modes, denoted by $\hat{m}^*$, are called the best transmission strategy for the data frame delivery under the given wireless channel conditions.

ILA is our new MPDU-based link adaptation scheme for 802.11a DCF systems, and it is an enhancement to the simple MSDU-based adaptive PHY mode selection scheme that we originally proposed in [56]. For completeness, we briefly describe the MSDU-based adaptive PHY mode selection scheme, present some numerical results, and discuss its problems and limitations.

4.4.1 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 name indicates, one of the key features of this scheme is that, after a wireless station makes the PHY mode selection and starts transmitting, the selected PHY mode will be used for all the potential re-transmissions, i.e., $\hat{m}^* = \{m^*, m^*, \cdots, m^*\}$.

Figs. 4.4(a) and (b) 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 2000 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



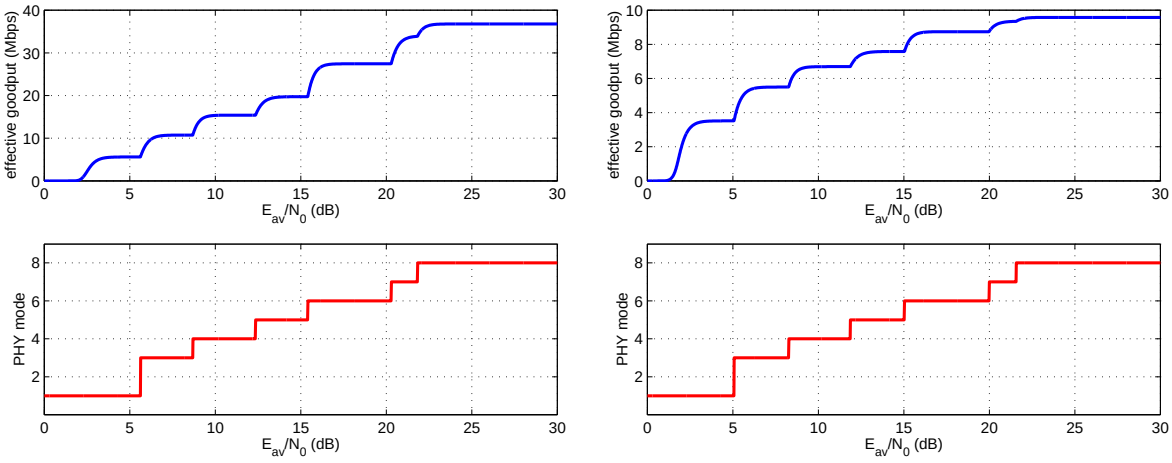
(a) MSDU size : 2000 octets

(b) MSDU size : 200 octets

Figure 4.4: Effective Goodput vs. SNR

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. 4.4 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. 4.5 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. 4.4.

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 re-transmissions. 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 re-transmissions, it can not adapt quickly to the fast-changing wireless channel.



(a) MSDU size : 2000 octets

(b) MSDU size : 200 octets

Figure 4.5: MSDU-based adaptive PHY mode selection to improve the effective goodput

4.4.2 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, called ILA, which features two major enhancements to 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 mid 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 run-time, based on the predictions 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 [17] and [61], we solve the problem from a different angle.

Pre-Established PHY Mode Table

The basic idea of ILA is that, the wireless station computes off-line 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 n^{th} transmission attempt, or equivalently, the n^{th} frame retry count. This table is used at run-time 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, similar to Eq. (4.10), the expected effective goodput, if PHY mode m is used, can be calculated by

$$\mathcal{G}(\ell, s, m, n_{max}) = \frac{E[data](\ell, s, m, n_{max})}{E[\mathcal{D}_{data}](\ell, s, m, n_{max})}, \tag{4.11}$$

where

$$E[data](\ell, s, m, n_{max}) = P_{s,xmit}(\ell, s, m) \cdot \ell, \tag{4.12}$$

and

$$E[\mathcal{D}_{data}](\ell, s, m, n_{max}) = \overline{\mathcal{T}}_{bkoff}(n_{max}) + \mathcal{T}_{data}(\ell, m) + \textit{tSIFSTime} + \mathcal{T}_{ack}(m') + P_{s,xmit}(\ell, s, m) \cdot \textit{tDIFSTime} + [1 - P_{s,xmit}(\ell, s, m)] \cdot \overline{\mathcal{D}}_{wait}(n_{max} + 1). \tag{4.13}$$

$P_{s,xmit}(\cdot)$, $\overline{\mathcal{T}}_{bkoff}(\cdot)$, $\mathcal{T}_{data}(\cdot)$, $\mathcal{T}_{ack}(\cdot)$, and $\overline{\mathcal{D}}_{wait}(\cdot)$ are given by Eqs. (4.4), (4.3), (4.1), (4.2), and (4.8), respectively. Consequently, the best PHY mode to use for the last transmission attempt and the corresponding maximum expected effective goodput are, respectively,

$$m^*(\ell, s, n_{max}) = \arg \max_{1 \leqslant m \leqslant 8} \mathcal{G}(\ell, s, m, n_{max}), \tag{4.14}$$

and

$$\mathcal{G}^*(\ell, s, n_{max}) = \mathcal{G}(\ell, s, m^*(\ell, s, n_{max}), n_{max}) = \max_{1 \leqslant m \leqslant 8} \mathcal{G}(\ell, s, m, n_{max}). \tag{4.15}$$

Now, let's consider the general case when $1 \leqslant n < n_{max}$. Assume that PHY mode m is selected for the n^{th} transmission attempt. The transmission is considered successful only upon receiving a positive acknowledgment; otherwise, the station has to re-contend for the medium to re-transmit 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:

$$\mathcal{G}(\ell, s, m, n) = \frac{E[data](\ell, s, m, n)}{E[\mathcal{D}_{data}](\ell, s, m, n)}, \tag{4.16}$$

where

$$\begin{aligned} E[data](\ell, s, m, n) = {} & P_{s,xmit}(\ell, s, m) \cdot \ell + [1 - P_{s,xmit}(\ell, s, m)] \\ & \cdot \int_{-\infty}^{\infty} f_{R|S}(r|s) \cdot E[data](\ell, r, m^*(\ell, r, n+1), n+1)\, dr, \end{aligned} \tag{4.17}$$

and

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

$$\overline{\mathcal{T}}_{bkoff}(n) + \mathcal{T}_{data}(\ell, m) + \textit{tSIFSTime} + \mathcal{T}_{ack}(m')$$
$$+ P_{s,xmit}(\ell, s, m) \cdot \textit{tDIFSTime} + [1 - P_{s,xmit}(\ell, s, m)] \cdot \overline{\mathcal{D}}_{wait}(n+1)$$
$$+ [1 - P_{s,xmit}(\ell, s, m)] \cdot \int_{-\infty}^{\infty} f_{R|S}(r|s) \cdot E[\mathcal{D}_{data}](\ell, r, m^*(\ell, r, n+1), n+1)\, dr. \quad (4.18)$$

Similarly, the best PHY mode to use for the n^{th} transmission attempt and the corresponding maximum expected effective goodput are

$$m^*(\ell, s, n) = \arg \max_{1 \leqslant m \leqslant 8} \mathcal{G}(\ell, s, m, n), \quad (4.19)$$

and

$$\mathcal{G}^*(\ell, s, n) = \mathcal{G}(\ell, s, m^*(\ell, s, n), n) = \max_{1 \leqslant m \leqslant 8} \mathcal{G}(\ell, s, m, n), \quad (4.20)$$

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 Eqs. (4.11), (4.14), (4.16), and (4.19).

Run-Time Execution

Fig. 4.6 shows the pseudo-coded algorithm for ILA. 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 pre-established and ready for run-time use. The counts *succ_count* for delivered frames and *fail_count* for dropped frames are both reset to zero, and the retry count (n_{curr}) for the frame at the header of the data queue is set to one. At run-time, 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 one and *succ_count* := *succ_count* + 1; else $n_{curr} := n_{curr} + 1$. As shown in the pseudo-code, if a frame can not be successfully delivered after n_{max} transmission

```
⟨ compute m*(ℓ, s, n) for each set of ℓ, s, and n ⟩;
succ_count := 0;    fail_count := 0;
F := the frame at the header of the data queue;
ℓ_curr := DataPayloadLength(F);    n_curr := 1;
while (the data queue is non-empty) {
   s_curr := the current wireless channel condition;
   m_{n_curr} := m*(ℓ_curr, s_curr, n_curr) ;
   ⟨ F is transmitted using PHY mode m_{n_curr} ⟩;
   if (an Ack frame is received correctly within the Ack timeout)    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 F and ℓ_curr ⟩;
   }
}
```

Figure 4.6: Pseudo-code of ILA

attempts, it will be dropped and *fail_count* := *fail_count* + 1, and n_{curr} is reset to one for the next frame waiting in the data queue. Notice that, since the best PHY mode table is computed off-line, there is no extra run-time computational cost for applying ILA.

Implementation Issues

Fig. 4.7 shows a system architecture to implement ILA. 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 the path loss estimation scheme we proposed for MiSer in Section 3.5.3 can be used for this purpose. Second, the link adaptor looks up the pre-built best PHY mode table to determine the best transmission

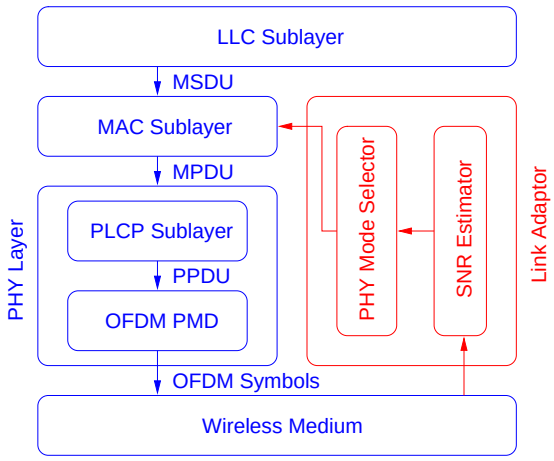


Figure 4.7: System architecture for link adaptation

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.

4.5 An Example of the Best PHY Mode Table

As described in the previous section, the key idea of the proposed ILA scheme 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 [6, 15, 68, 76, 80] dealing with the problem of

building accurate wireless channel models, which, however, is not the focus of this work. 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 on 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 may be more complicated.

4.5.1 Error Performances of PHY Modes over the AWGN Channel

Bit Error Probability

The symbol error probability for an M-ary (M = 4, 16, 64) QAM (Quadrature Amplitude Modulation) [54] with the average SNR per symbol, s, can be calculated by

$$P_M(s) = 1 - \left[1 - P_{\sqrt{M}}(s)\right]^2, \tag{4.21}$$

where

$$P_{\sqrt{M}}(s) = 2 \cdot \left(1 - \frac{1}{\sqrt{M}}\right) \cdot Q\left(\sqrt{\frac{3}{M-1} \cdot s}\right) \tag{4.22}$$

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

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2}\, dy. \tag{4.23}$$

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

$$P_b^{(M)}(s) \approx \frac{1}{\log_2 M} \cdot P_M(s). \tag{4.24}$$

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

same as the symbol error probability, which is given by

$$P_b^{(2)}(s) = P_2(s) = Q\left(\sqrt{2s}\right). \tag{4.25}$$

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

Packet Error Probability

In [55], 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

$$P_e^m(h,s) \leqslant 1 - [1 - P_u^m(s)]^{8h}, \tag{4.26}$$

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

$$P_u^m(s) = \sum_{d=d_{free}}^{\infty} a_d \cdot P_d(s), \tag{4.27}$$

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

$$P_d(s) = \begin{cases} \sum_{k=(d+1)/2}^{d} \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k}, & \text{if } d \text{ is odd,} \\ \frac{1}{2} \cdot \binom{d}{d/2} \cdot \rho^{d/2} \cdot (1-\rho)^{d/2} + \sum_{k=d/2+1}^{d} \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k}, & \text{if } d \text{ is even,} \end{cases} \tag{4.28}$$

where ρ is the bit error probability for the modulation scheme selected in PHY mode m, and is given by Eq. (4.24) or (4.25). 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 4.3.3 to compute $P_{s,xmit}(\ell, s, m)$, the probability of a successful frame transmission, can be calculated by

$$\begin{aligned} P_{e,data}(\ell, s, m) &= 1 - \left[1 - P_e^1(24/8, s)\right] \cdot \left[1 - P_e^m(28 + (16+6)/8 + \ell, s)\right], \\ &= 1 - \left[1 - P_e^1(3, s)\right] \cdot \left[1 - P_e^m(30.75 + \ell, s)\right], \end{aligned} \tag{4.29}$$

and

$$\begin{aligned} P_{e,ack}(s, m') &= 1 - \left[1 - P_e^1(24/8, s)\right] \cdot \left[1 - P_e^{m'}(14 + (16+6)/8, s)\right], \\ &= 1 - \left[1 - P_e^1(3, s)\right] \cdot \left[1 - P_e^{m'}(16.75, s)\right], \end{aligned} \tag{4.30}$$

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 Eq. (4.26).

4.5.2 Wireless Channel Variation Model

Fig. 4.8 shows the two-state discrete time Markov chain to model the wireless channel variation. The wireless channel could be in either *good* or *bad* state. When the wireless channel is in *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 *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.

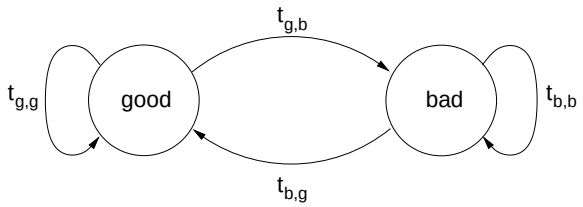


Figure 4.8: A two-state discrete time Markov chain to model the wireless channel variation

Therefore, the state transition probabilities $t_{g,b}$ and $t_{b,g}$ 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 zero (one), the wireless channel tends to stay in *bad* (*good*) state for most of the time.

4.5.3 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 4.4.2. Fig. 4.9 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 2000 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,

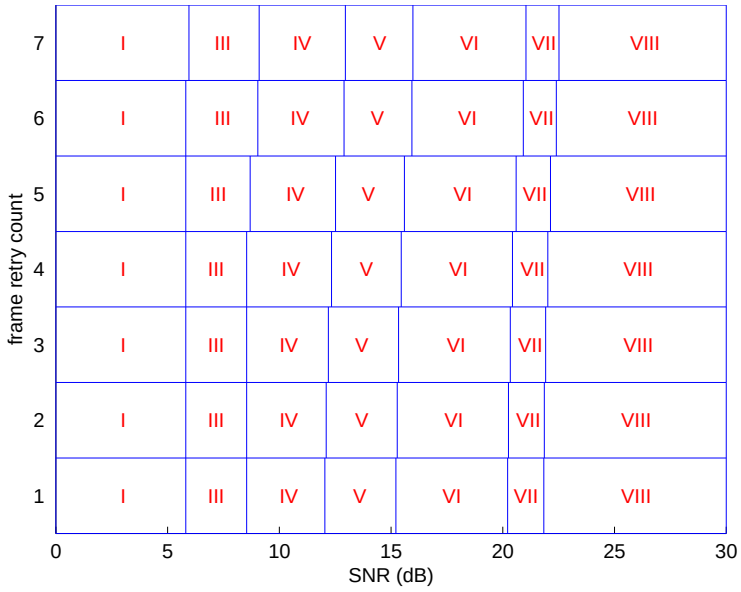


Figure 4.9: An example of the best PHY mode table

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.

4.6 Performance Evaluation

In this section, we evaluate the effectiveness of ILA using simulation. As specified in the 802.11 standard, the length of an MSDU must be less than or equal to 2304 octets (see Clause 6.2.1.1.2 in [24]), and an MSDU shall be broken into fragments if its size exceeds the fragmentation threshold after adding the 28-octet MAC overhead. Besides, if the length of an MSDU is less than or equal to *dot11RTSThreshold*, the number of transmission attempts is limited to *dot11ShortRetryLimit*. In the simulation, we set both the fragmentation threshold and *dot11RTSShreshold* to 2332 ($= 2304 + 28$) octets, therefore, each MSDU is transmitted without fragmentation or RTS/CTS support, and it can be attempted transmission up to seven (the default value of *dot11ShortRetryLimit*) times. The BSS basic rate set is the set of the 802.11a mandatory data rates, i.e., $\{$6 Mbps, 12 Mbps, 24 Mbps$\}$. The eight different PHY modes of the 802.11a PHY and the related parameters used in the simulation are listed in Tables 3.1 and 3.2, 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: the AutoRate Fallback (ARF) scheme used by Lucent Technologies, three Single-Mode schemes using PHY mode 1 (SM-1), mode 5 (SM-5), and mode 8 (SM-8), respectively, the simple MSDU-based adaptive PHY mode selection scheme (SLA), and ILA. 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 10000 MSDUs from the transmitter, and the length of each MSDU is 2000 octets. Testing schemes are compared with each other under different wireless channel variation patterns, i.e., different $t_{b,g}$ values.

4.6.1 "Timeout" Factor in ARF

In our simulation of ARF, we follow as closely as possible the protocol specifications in [34]. 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 [34]. So, similar to [23], 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. 4.10, 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

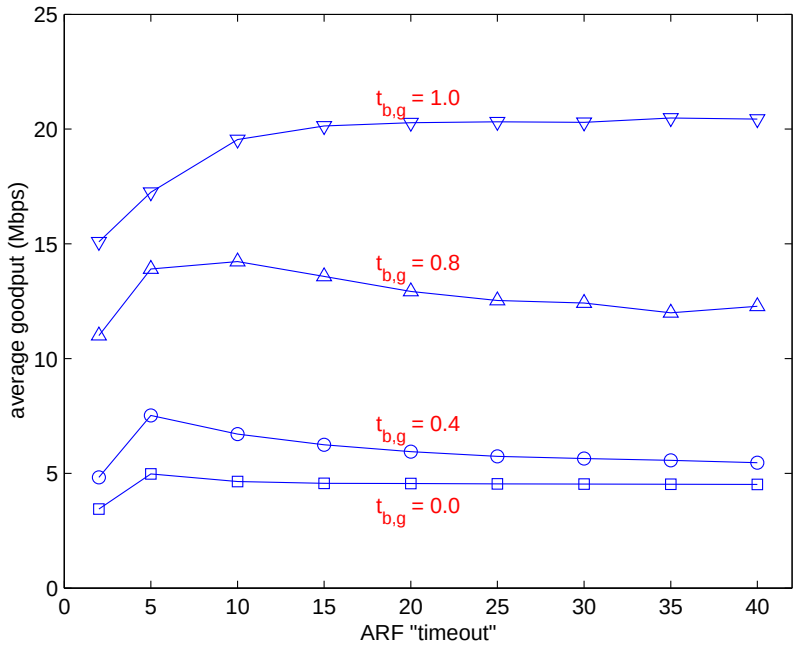


Figure 4.10: Goodput comparison of ARF for various “timeout” values

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 timeout is less than (equal to) 10, the PHY rate increase 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.

4.6.2 Simulation Results

First, the testing schemes are evaluated using the average goodput, and the results under different wireless channel variation patterns are plotted in Fig. 4.11. Clearly, the goodput varies with the channel variation pattern, and the link adaptation schemes outperform the single-mode schemes in most cases. We have several observations from this figure.

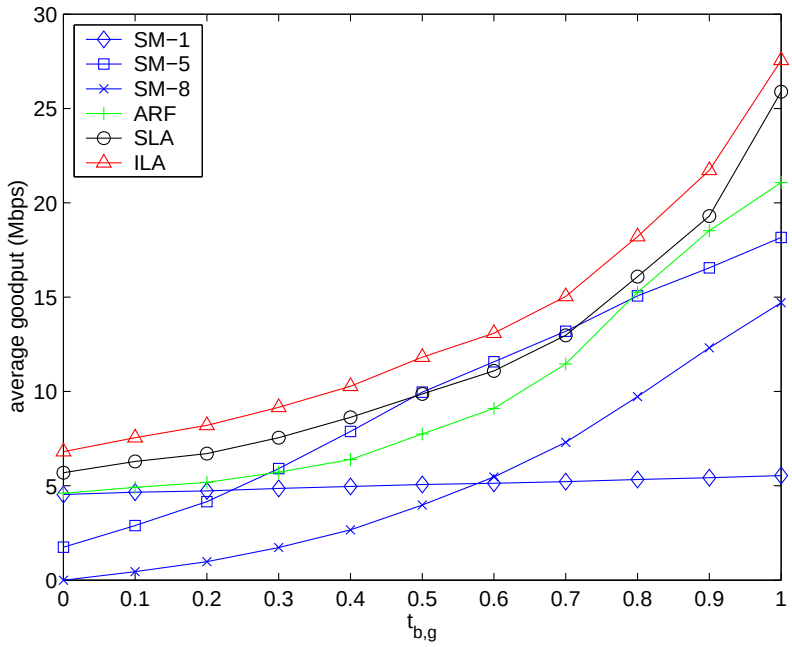


Figure 4.11: Goodput comparison of the testing schemes

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 *good* state for most of the time, i.e., when $t_{b,g}$ is close to one, 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 *bad* state for most of the time, i.e., when $t_{b,g}$ is close to zero. This is because the poorest error correcting capability of PHY mode 8 results in a large number of frame re-transmissions and delivery failures. For the extreme case of $t_{b,g} = 0$, 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 (SLA), and the triangle points (ILA) in Fig. 4.11. 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 *good* state and robust low PHY modes in *bad* state. ARF appears to be the most conservative link adaptation scheme of three. One interesting observation is that, if a link adaptation scheme is not designed carefully — e.g., ARF is purely heuristic and SLA can not adjust the PHY mode between frame re-transmissions, then it might even result in 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 SLA. On the other hand, since ILA 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 the figure, ILA is significantly better than ARF and outperforms SLA by about 10% 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 10000 frames waiting for delivery. The results are listed in Table 4.1. Notice that SM-1, ARF, and ILA 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 PHY rate, ILA 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 SLA. First, in general, the average number of dropped frames increases as the wireless channel gets worse for these three schemes. In particular, when $t_{b,g} = 0$, SM-8 results in 10000 dropped frames, which is consistent with the zero average goodput observed in Fig. 4.11. Second, SLA results in less dropped frames when $t_{b,g} = 0$ than

$t_{b,g}$	average number of dropped frames					
	SM-1	SM-5	SM-8	ARF	SLA	ILA
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

Table 4.1: Comparison for average number of dropped MSDUs

when $t_{b,g} = 0.1$ or 0.2. This counter-intuitive observation is surprising at the first sight, but rather reasonable. Recall that SLA is unable to adjust the PHY mode for re-transmissions when the wireless channel condition fluctuates. So the frame drop performance of SLA is affected by not only the wireless channel condition, but the wireless channel variation as well. When $t_{b,g} = 0$, since the wireless channel tends to stay in *bad* state for most of the time, there are relatively less SNR variations than when $t_{b,g} = 0.1$ or 0.2. As a result, SLA 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.

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 4.2. Notice that *avg_att* 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.

$t_{b,g}$	average number of attempts (*avg_att*)					
	SM-1	SM-5	SM-8	ARF	SLA	ILA
0.0	1.2139	4.0013	7.0000	1.3490	1.4299	1.2794
0.1	1.1829	3.2752	5.9086	1.3327	1.3866	1.2530
0.2	1.1664	2.7564	5.0749	1.3270	1.3811	1.2385
0.3	1.1374	2.3073	4.3263	1.3203	1.3386	1.2097
0.4	1.1126	1.9866	3.7411	1.3117	1.2997	1.1925
0.5	1.0899	1.6987	3.2283	1.3069	1.2739	1.1686
0.6	1.0777	1.5296	2.8354	1.3137	1.2467	1.1535
0.7	1.0589	1.3826	2.5230	1.3150	1.2128	1.1380
0.8	1.0374	1.2329	2.1960	1.3159	1.1783	1.1162
0.9	1.0199	1.1343	1.9838	1.2944	1.1627	1.1010
1.0	1.0000	1.0403	1.8176	1.2752	1.1344	1.0872

Table 4.2: Comparison for average number of transmission attempts per MSDU delivery

Finally, we use Fig. 4.12 to illustrate the behavior of the three testing link adaptation schemes, and the fast adaptability of ILA 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. 4.12(a) shows the SNR values observed during each of the transmission attempts from #3000 to #3100, and the PHY mode selections by ARF, SLA and ILA are shown in Figs. 4.12(b), (c), and (d), 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 dis-similarity between Figs. 4.12(a) and (b). It is clear that ARF is a conservative link adaptation scheme, where PHY modes 3 and 4 are the most popular choices. Second, SLA and ILA are better at reacting and adapting to the wireless channel variations, and their PHY mode selections basically follow the changes in SNR. However, in SLA, if an MSDU delivery starts right before the wireless channel turns *bad* (e.g., at #3011 and #3041), it has to stick with the original PHY mode selection for all the re-transmissions, thus resulting in more transmission failures than ILA.

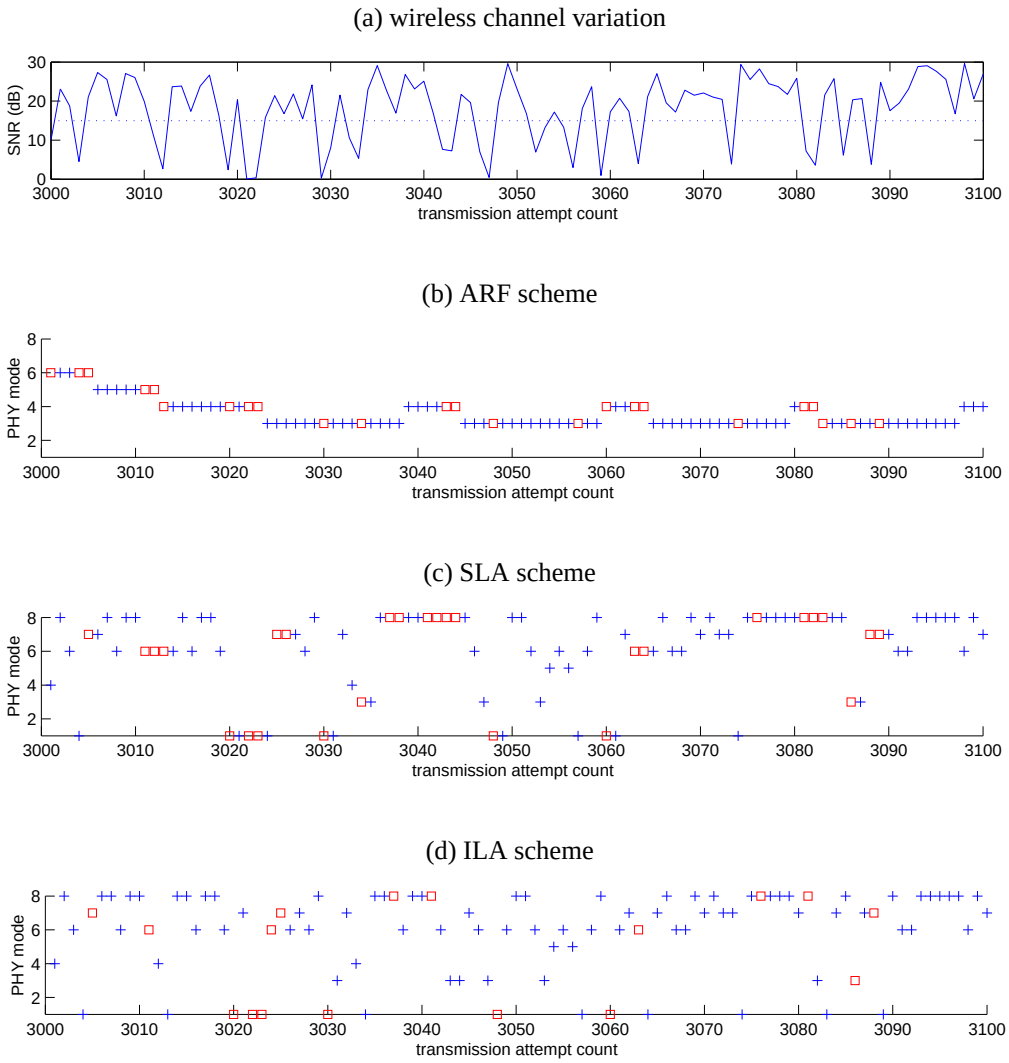


Figure 4.12: Adaptability comparison of three link adaptation schemes

4.7 Conclusion

In this chapter, we present a generic method to analyze the goodput performance of an 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, called ILA, assuming the availability of the wireless channel models. It is a table-driven approach similar to MiSer, and the key idea is

to pre-establish 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 using a recursive algorithm. At run-time, 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 off-line, there is no extra run-time computational cost for applying ILA.

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

CHAPTER V

A SOFT REAL-TIME EXTENSION TO THE ORINOCO LINUX DEVICE DRIVER

5.1 Introduction

At present, most 802.11-compliant products available in the market only implement the mandatory DCF, and due to the contention nature of the DCF, the current 802.11 systems yield unpredictable delay characteristics and do not support prioritized transmission of real-time traffic. The IEEE 802.11 TGe (Task Group E) has been working on the new 802.11e standard [27, 46], which defines enhancements to the current 802.11 MAC to support applications with QoS (Quality of Service) requirements. One of the new mechanisms is called the EDCF (Enhanced Distributed Coordination Function), which realizes the QoS support by introducing the concept of TCs (Traffic Categories). A single station may implement up to eight transmission queues whose service priorities are determined by different queue management parameters. Each queue corresponds to a certain TC. Before the new 802.11e standard is finalized by the IEEE standardization committee and introduced to the market, the DCF-mode 802.11-compliant devices are expected to continue their dominance of the market. Actually, even after the new 802.11e devices are introduced to the market, there will still be many legacy 802.11 devices deployed in various sectors. In order to support real-time applications within the current 802.11 systems,

appropriate real-time extensions are essential.

In this chapter, we study the 802.11 systems using the popular Agere ORiNOCO devices [4], and implement an enhanced Linux device driver for ORiNOCO devices, called RT-WLAN [32], which extends the original ORiNOCO Linux device driver to support soft real-time communications. To our best knowledge, this is the first effort in providing real-time support in the 802.11 systems at the device driver level. In contrast to the approaches of changing the OS kernel or modifying the NIC (Network Interface Card) firmware, our scheme has a significant advantage: it can be used along with the existing OS kernel and protocol stack as well as the off-the-shelf ORiNOCO devices, so users can simply replace the original ORiNOCO Linux device driver with RT-WLAN and enjoy the significantly better real-time support. In RT-WLAN, separate queues are used for real-time and non-real-time traffic with the service preference given to the real-time queue. The real-time queue is served according to the EDF (Earliest-Deadline-First) policy [39], while the non-real-time queue is served in a FIFO (First-In-First-Out) manner. Besides, we implement an adaptive traffic smoother [40] as part of the non-real-time queue, which regulates bursty non-real-time traffic before they are injected into the network, thus giving higher priority to in-progress real-time transmissions. In addition, being closer to the actual physical layer enables us to get more timely feedback about the transmission results, thus making our approach more responsive.

The rest of this chapter is organized as follows. Section 5.2 overviews the 802.11 systems using the Agere ORiNOCO devices. The implementation details of RT-WLAN are presented in Section 5.3. Section 5.4 presents and evaluates the experimental results, and finally, the chapter concludes with Section 5.5.

5.2 System Overview

The Agere ORiNOCO cards are designed according to the 802.11b PHY [26], which is the higher-speed extension to the DSSS (Direct Sequence Spread Spectrum) system specified in the 802.11 standard and, therefore, known as the HR/DSSS PHY. It provides the higher 5.5 Mbps and 11 Mbps data transmission rates in the 2.4 GHz ISM (Industrial, Scientific, and Medical) band, in addition to the 1 Mbps and 2 Mbps rates, using the advanced 8-chip CCK (Complimentary Code Keying) modulation scheme. Table 5.1 lists the related characteristics for the 802.11b PHY.

Characteristics	Value	Comments
tSlotTime	20 μs	Slot time
tSIFSTime	10 μs	SIFS time
tDIFSTime	50 μs	DIFS = SIFS + 2 $\times$ Slot
cw_{min}	31	min contention window size
cw_{max}	1023	max contention window size
tPLCPPreamble	144 μs	PLCP preamble duration
tPLCPHeader	48 μs	PLCP header duration

Table 5.1: The 802.11b PHY Characteristics

The Agere ORiNOCO cards only implement the mandatory DCF, which we have described in details in Sections 2.2 and 3.2. Recall that a DCF station is allowed to transmit only if its carrier-sense mechanism determines that the medium has been idle for at least DIFS time. The SIFS, which is smaller than the DIFS, is the time interval used between transmissions within a frame exchange sequence, i.e., a two-way Data-Ack handshake or a four-way RTS-CTS-Data-Ack handshake. If a CTS (Ack) frame is not received due possibly to an erroneous reception of the preceding RTS (Data) frame, the transmitter will contend again for the medium to re-transmit the frame after a CTS (Ack) timeout, which equals to a SIFS time plus the CTS (Ack) transmission time plus a Slot time.

Besides, in order to reduce the collision probability among multiple stations accessing

the medium, a DCF station is required to select a random backoff interval after deferral, or prior to attempting to transmit another frame after a successful frame transmission. According to the description given in Section 2.2.2 on the backoff behavior, the DCF adopts a *slotted binary exponential backoff* scheme to select the random backoff interval (in the unit of *tSlotTime*). This random number is drawn from a uniform distribution over the interval [0, cw], where cw is the contention window size and its initial value is cw_{min}. In the case of an unsuccessful transmission, cw is updated to $[2 \times (cw + 1) - 1]$. Once cw reaches cw_{max}, it will remain at this value until it is reset to cw_{min}. In the case of a successful data transmission, the cw value is reset to cw_{min} before the random backoff interval is selected. Therefore, the average backoff interval before the i^{th} transmission attempt, or equivalently, the $(i-1)^{th}$ re-transmission attempt, denoted by $\overline{\mathcal{T}}_{bkoff}(i)$, can be calculated by

$$\overline{\mathcal{T}}_{bkoff}(i) = \frac{\min\left[2^{i-1} \cdot (cw_{min} + 1) - 1,\ cw_{max}\right]}{2} \cdot \textit{tSlotTime}. \tag{5.1}$$

Each station decrements its backoff counter every *tSlotTime* interval after the wireless medium is sensed to be idle for DIFS time. If the counter has not reached zero and the medium becomes busy again, the station freezes its counter. When the counter finally reaches zero, the station starts its transmission. Fig. 5.1 illustrates such an operation of decrementing the backoff counter when the two-way handshake is used.

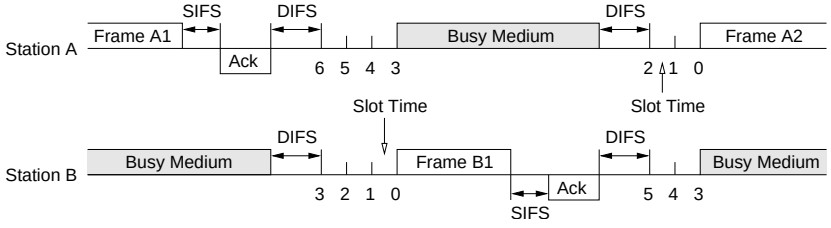


Figure 5.1: An example of backoff decrements with two-way handshake

As described in Section 3.4.2, in an 802.11 system, each MPDU consists of the fol-

lowing components: *MAC header, frame body* of variable length, and *FCS*. The MAC overheads due to the MAC header and the FCS are 28 octets in total. The frame size of an Ack frame is 14 octets. During the transmission, a PLCP preamble and a PLCP header are added to an MPDU to create a PPDU. The PPDU format of the 802.11b PHY is shown in Fig. 5.2. Both the PLCP preamble and the PLCP header are transmitted at 1 Mbps using DBPSK (Differential Binary Phase Shift Keying) modulation and Barker word spreading. The MPDU (conveyed from MAC) is transmitted at the data rate specified in the SIGNAL field.

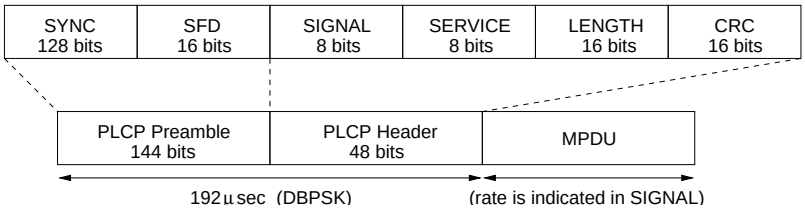


Figure 5.2: PPDU frame format of the 802.11b PHY

Based on the above analysis, the time for a data frame with ℓ octets payload to be transmitted over the 802.11b PHY and the Ack transmission time at rate r (Mbps) are, respectively,

$$\begin{aligned} \mathcal{T}_{data}(\ell, r) &= \textit{tPLCPPreamble} + \textit{tPLCPHeader} + \frac{(\ell + 28) \cdot 8}{r} \\ &= 192 + \frac{(\ell + 28) \cdot 8}{r} \ (\mu s), \end{aligned} \tag{5.2}$$

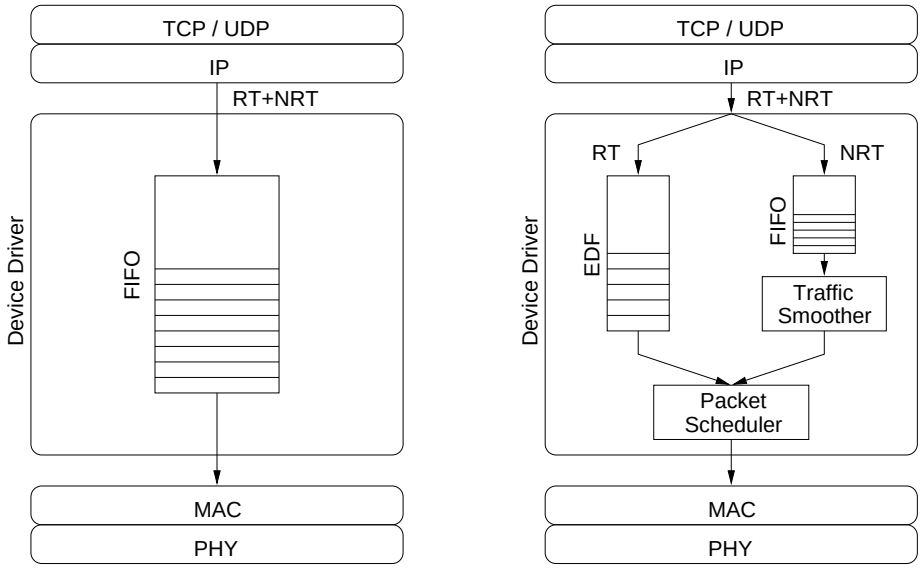
and

$$\begin{aligned} \mathcal{T}_{ack}(r) &= \textit{tPLCPPreamble} + \textit{tPLCPHeader} + \frac{14 \cdot 8}{r} \\ &= 192 + \frac{112}{r} \ (\mu s). \end{aligned} \tag{5.3}$$

5.3 RT-WLAN

RT-WLAN is implemented by modifying the original Linux device driver for Agere ORiNOCO cards (*orinoco.c* and *orinoco_cs.c*, version 0.08 [74]). The versions of the Linux kernel and the PCMCIA package, which RT-WLAN is based on, are 2.4.12 and 3.1.31, respectively. The key modification is to add soft real-time extensions to the original driver so that the deadline requirements of the real-time applications can be better guaranteed.

As shown in Fig. 5.3(a), the original ORiNOCO driver simply serves the packets in a FIFO (First-In-First-Out) manner without differentiation between RT (Real-Time) and NRT (Non-Real-Time) traffic. In contrast, RT-WLAN provides separate queues for RT and NRT traffic, and the service preference is given to the RT queue. Besides, in order to have most real-time packets meet their deadlines, we apply the EDF (Earliest-Deadline-First) policy to the RT queue in RT-WLAN, so that the real-time packets with the closest deadlines are served first in the RT queue.



(a) Original ORiNOCO Linux device driver (b) RT-WLAN

Figure 5.3: Comparison of two device driver architectures

Note that the above extension only provides the service differentiation between RT and NRT traffic, as well as among the RT packets, within the same station. Hence, RT traffic may still suffer high latency due to the potential collisions with other traffic on the shared wireless medium and the consequent backoff operations according to the 802.11 standard. A large burst of NRT traffic at one station makes it very hard to provide bounded transmission delays for the RT traffic at another station. To deal with this problem, we apply adaptive traffic smoothing [40] to NRT traffic in RT-WLAN. The key idea is to regulate bursty NRT traffic before they are injected into the network, thus giving higher priority to in-progress real-time transmissions. Since RT traffic (e.g., multimedia or real-time control applications) usually arrives pseudo-periodically, it need not be smoothed [40]. The RT-WLAN architecture is illustrated in Fig. 5.3(b).

5.3.1 User Interface

We have provided well-formulated APIs that are easily usable by application programmers. An application can indicate whether the packet it creates is a real-time packet, and specify the corresponding deadline information, if necessary, by using the function call: *set_priority(int packet_type, double relative_deadline)*. The *packet_type* parameter can take the value of zero (for non-real-time packets) or one (for real-time packets). The *relative_deadline* parameter specifies the relative deadline that each real-time packet should try to meet after it is generated. If a packet is specified as a non-real-time packet, the value of *relative_deadline* is simply ignored.

The *set_priority()* function call is implemented by using the *setsockopt()* system call. The real-time and non-real-time packets are differentiated by setting the TOS (Type-Of-Service) field in the IP header. The absolute deadline of each real-time packet is obtained by adding the relative deadline to the current time at the instant of packet generation, and

this deadline value is carried in the IP_OPTIONS field of the IP header. Besides, we extend the *ioctl()* system call by which the application programmer can revert back easily to the original ORiNOCO driver.

5.3.2 RT Queue and EDF Policy

In RT-WLAN, the real-time packets are served according to the EDF policy. A packet with a smaller absolute deadline receives priority over other packets with larger deadlines. Therefore, the RT queue is maintained by keeping the real-time packets in the increasing order of their absolute deadlines, and the packet with the earliest deadline is always positioned at the head of the RT queue. Whenever a new real-time packet arrives from the upper layer, an appropriate position will be found for this new packet so as to maintain the sorted order.

5.3.3 NRT Queue and Adaptive Traffic Smoother

In RT-WLAN, the NRT queue is maintained in a FIFO manner: all the non-real-time packets are served in the order that they were en-queued. Besides, RT-WLAN requires each packet to pass through an additional traffic smoother before it is actually de-queued. This traffic smoother decides whether a non-real-time packet should be sent directly to the NIC or returned to the NRT queue for a deferred transmission.

A traffic smoother regulates bursty NRT traffic to reduce the chance of packet collisions and keeps the network utilization under a certain limit. More specifically, a traffic smoother regulates the NRT packet stream using a credit bucket, which is the same as the well-known leaky-bucket regulator [12]. The credit bucket has two parameters: CBD (Credit Bucket Depth) and RP (Refresh Period). A credit of CBD bytes are replenished into the bucket every RP seconds, so the station input limit is given by CBD/RP. The traffic smoother used in RT-WLAN is adaptive in the sense that the station input limit may

vary according to the current network utilization. It uses a simple adaptation mechanism called the HIMD (Harmonic-Increase and Multiplicative-Decrease) adaptation as follows. HIMD decreases RP by a fixed constant δ every τ seconds when the network utilization is low, thus increasing the station input limit harmonically. The station input limit may be increased as long as the overall network utilization does not cause real-time packets to experience larger delays. On the other hand, whenever a non-real-time packet reaches the traffic smoother, the traffic smoother will check the time instant when the network utilization was last indicated high and compare it with the current time. If this time difference falls within a certain bound α, the traffic smoother assumes that another station is trying to transmit a real-time packet. In this case, it abstains from transmission by depleting the current credits and doubling the RP, thus decreasing the station input limit multiplicatively. The values of CBD, RP, δ, τ, and α may be modified through the extended *ioctl()* system call. The procedural description of the adaptive traffic smoother is shown in Fig. 5.4.

```
Adaptive_Traffic_Smoother() {
    if (Last_High_Network_Utilization_Indication.Time ≥ Current_Time − α) {
        send_packet_back_to_queue;
        Number_of_Credits = 0;
        RP = min(RP_max, 2×RP);
    }
    else if (Number_of_Credits > 0) {
        return NRT_packet;
    }
    else send_packet_back_to_queue;
    return NULL;
}
```

Figure 5.4: Procedural description of the adaptive traffic smoother

Note that, in order to implement such an adaptive traffic smoother, it is very important to detect a change in the network utilization. At the device driver level, the estimation of the network utilization can be indirectly obtained either from the collision status report by the NIC after it detects the packet collisions, or by measuring the clearing time of the

NIC buffer. The latter one is used in RT-WLAN. The rationale behind it and the related analysis will be presented next.

5.3.4 NIC Buffer Clearing Time: Network Utilization Indicator

An adaptive traffic smoother in the Ethernet environment — for example, the one presented in [40] — may use the collision status report as the network utilization indicator, since most Ethernet device drivers can easily collect the collision status information by querying the NIC. However, we are dealing with the WLAN environment, and the original ORiNOCO driver does not support the collision status report. Besides, the register details of the Hermes chip-set used in the ORiNOCO silver cards are not available to public. In RT-WLAN, we get around this problem by measuring the NIC buffer clearing time as the transmission delay of a packet, and also, as the indication of the current network utilization. The NIC buffer clearing time is measured as the time interval between when a packet is copied to the NIC buffer and when a successful packet delivery is reported by the NIC to the device driver. Clearly, our scheme works correctly only if the packets are served one at a time, i.e., the NIC buffer holds at most one packet at any time. This is also the way the NIC buffer is used by the original ORiNOCO driver.

Obviously, when a packet is successfully delivered without encountering any contention and/or collision on the wireless medium, the corresponding NIC buffer clearing time is small. Otherwise, the packet has to wait in the NIC buffer for a longer time until the wireless medium is cleared. To show how our scheme works, we ran two experiments, and the results are plotted in Fig. 5.5. The circle points represent the benchmark case when only one station is transmitting continuously. The cross points represent the case when two stations are contending for the wireless medium. In both cases, the packets are transmitted at 11 Mbps, the packet size is fixed at 1300 octets, the RTS/CTS option is turned off, and

fragmentation is disabled. We have two observations. First, the NIC buffer clearing time in the benchmark case varies in a small range and the average value is less than 1000 μs. Second, in the contention case, although some packets still show small transmission delays that are comparable to the benchmark case, most of them present much higher transmission delays than the benchmark case, and there are significant gaps in between. Reasons for such phenomenon can be explained as follows.

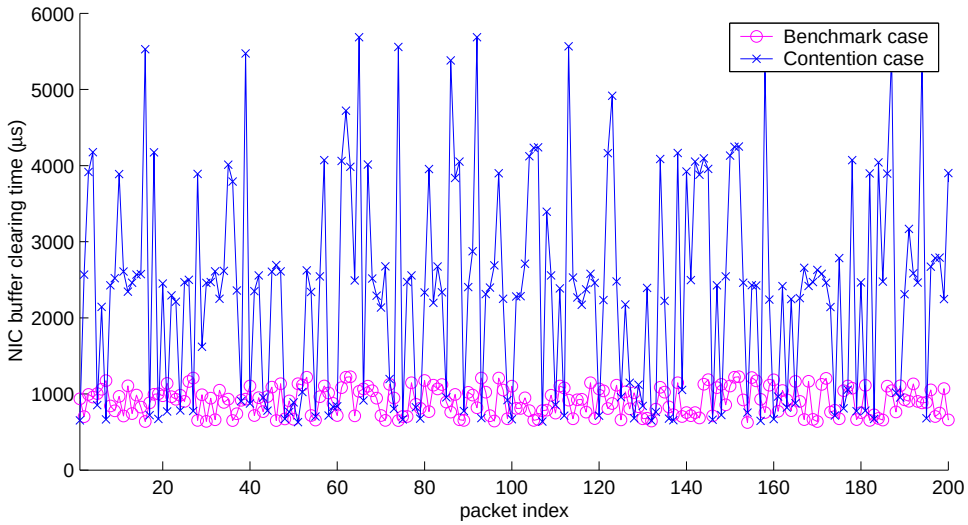


Figure 5.5: Comparison of NIC buffer clearing time

In the benchmark case, there are no contentions on the wireless medium, so all the packets are successfully transmitted in their first attempts. The random backoff interval before the transmission is in the unit of *tSlotTime* (20 μs) and this random number is uniformly selected from the minimum contention window [0, 31]. Therefore, the difference between the maximum transmission delay and the minimum transmission delay is 620 μs, which is exactly what we observed from Fig. 5.5. By referring to the timing of a successful two-way frame exchange in an 802.11 DCF system (shown in Fig. 4.1), the average packet

transmission delay in the benchmark case can be calculated by[1]

$$\overline{\mathcal{T}}_{benchmark} = \textit{tDIFSTime} + \overline{\mathcal{T}}_{bkoff}(1) + \mathcal{T}_{data}(1300, 11) + \textit{tSIFSTime} + \mathcal{T}_{ack}(1), \quad (5.4)$$

where $\overline{\mathcal{T}}_{bkoff}(\cdot)$, $\mathcal{T}_{data}(\cdot)$, and $\mathcal{T}_{ack}(\cdot)$ are given by Eqs. (5.1), (5.2), and (5.3), respectively.

In the contention case, there are three possible scenarios resulting in extra delay of a packet transmission. In the first scenario, the wireless channel is busy due to an in-progress transmission when the packet arrives the NIC buffer. The packet transmission delay and the extra waiting time are given by

$$\overline{\mathcal{T}}_{busy} = \mathcal{T}_{in-progress} + \textit{tDIFSTime} + \overline{\mathcal{T}}_{bkoff}(1) + \mathcal{T}_{data}(1300, 11) + \textit{tSIFSTime} + \mathcal{T}_{ack}(1), \quad (5.5)$$

and

$$\Delta_{busy} = \overline{\mathcal{T}}_{busy} - \overline{\mathcal{T}}_{benchmark} = \mathcal{T}_{in-progress}, \quad (5.6)$$

respectively. Since $\mathcal{T}_{in-progress}$ could be any value from zero to a full packet transmission time, so it is difficult to distinguish this scenario from the benchmark case. In the second scenario, the wireless station freezes its backoff counter since the other station starts transmitting first. By referring to Fig. 5.1, the packet transmission delay and the extra waiting time are given by

$$\begin{aligned} \overline{\mathcal{T}}_{freeze} &= \textit{tDIFSTime} + \overline{\mathcal{T}}_{bkoff}(1) \\ &\quad + [\mathcal{T}_{data}(1300, 11) + \textit{tSIFSTime} + \mathcal{T}_{ack}(1) + \textit{tDIFSTime}] \\ &\quad + \mathcal{T}_{data}(1300, 11) + \textit{tSIFSTime} + \mathcal{T}_{ack}(1), \end{aligned} \quad (5.7)$$

[1]Based on our calculation, the average transmission delay in the benchmark case should be around 1800 μs. However, it is quite different from our experimental results (less than 1000 μs). This may be due to our mis-interpretation of the HREG_EV_TX event [29], which we use as the indication of a successful packet delivery. Fortunately, the observed delay difference between the benchmark case and the contention case is still consistent with our analysis. Besides, we assume that the Ack frames are transmitted at the most conservative rate of 1 Mbps.

and

$$\begin{aligned}\Delta_{freeze} &= \overline{\mathcal{T}}_{freeze} - \overline{\mathcal{T}}_{benchmark} \\ &= \mathcal{T}_{data}(1300, 11) + \textit{tSIFSTime} + \mathcal{T}_{ack}(1) + \textit{tDIFSTime} \\ &= 1424\ \mu\text{s}, \end{aligned} \tag{5.8}$$

respectively, and the cross points around 2400 μs in Fig. 5.5 correspond to this scenario. In the third scenario, the transmitted packet collides on the wireless medium and the wireless station has to re-contend for the channel to re-transmit the packet. By referring to the timing of frame re-transmission due to an erroneous data frame reception in an 802.11 DCF system (shown in Fig. 4.3), the packet transmission delay and the extra waiting time are given by

$$\begin{aligned}\overline{\mathcal{T}}_{collision} &= \textit{tDIFSTime} + \overline{\mathcal{T}}_{bkoff}(1) + \mathcal{T}_{data}(1300, 11) + \textit{Ack\_timeout} \\ &\quad + \overline{\mathcal{T}}_{bkoff}(2) + \mathcal{T}_{data}(1300, 11) + \textit{tSIFSTime} + \mathcal{T}_{ack}(1), \end{aligned} \tag{5.9}$$

and

$$\begin{aligned}\Delta_{collision} &= \overline{\mathcal{T}}_{collision} - \overline{\mathcal{T}}_{benchmark} \\ &= \mathcal{T}_{data}(1300, 11) + \textit{Ack\_timeout} + \overline{\mathcal{T}}_{bkoff}(2) \\ &= \mathcal{T}_{data}(1300, 11) + [\textit{tSIFSTime} + \mathcal{T}_{ack}(1) + \textit{tSlotTime}] + \overline{\mathcal{T}}_{bkoff}(2) \\ &= 2024\ \mu\text{s}, \end{aligned} \tag{5.10}$$

respectively, and the cross points between 3000 μs and 5000 μs in Fig. 5.5 can be explained by this scenario. Note that a packet transmission may experience multiple backoff freezes and/or collisions, thus resulting in even larger extra delays — for example, the cross points above 5000 μs.

Based on the above analysis, in RT-WLAN, we select 2000 μs as the threshold: any NIC buffer clearing time larger than 2000 μs indicates that the current network utilization

```
Packet_Scheduler() {
    if (RT_Queue.size > 0) {
        remove_the_packet_from_head_of_RT_queue;
        send_packet_to_NIC;
        Number_of_Credits = Number_of_Credits − RT_Packet.size;
    }
    else if (NRT_Queue.size > 0) {
         NRT_packet = Adaptive_Traffic_Smoother();
        if (NRT_packet ≠ NULL) {
            send_packet_to_NIC;
            Number_of_Credits = Number_of_Credits − NRT_Packet.size;
        }
    }
}
```

Figure 5.6: Procedural description of the packet scheduler

is high. Actually, using the NIC buffer clearing time as the network utilization indicator is more accurate than using the collision status report, since packet collision is only one of the above three scenarios that may cause extra delay of a packet transmission.

5.3.5 Packet Scheduler

The procedural description of the packet scheduler is shown in Fig. 5.6. It monitors both the RT and NRT queues and gives priority to the RT queue over the NRT queue. Only NRT traffic is smoothed in order to keep the station traffic arrival rate — which includes both RT and NRT traffic — under the station input limit.

If the RT queue is not empty, the real-time packet at the head of the RT queue is immediately transferred to the NIC, regardless of the number of available credits, and as many credits as the size of the packet are removed from the credit bucket. So the balance of credits could be negative. On the other hand, for a non-real-time packet, the adaptive traffic smoother is called upon to decide whether it should be transferred to the NIC.

5.4 Performance Evaluation

In this section, we experimentally evaluate the effectiveness of our RT-WLAN device driver. The Agere ORiNOCO silver cards are used for wireless communications between laptops and are running in the IBSS (Independent Basic Service Set) ad hoc mode.

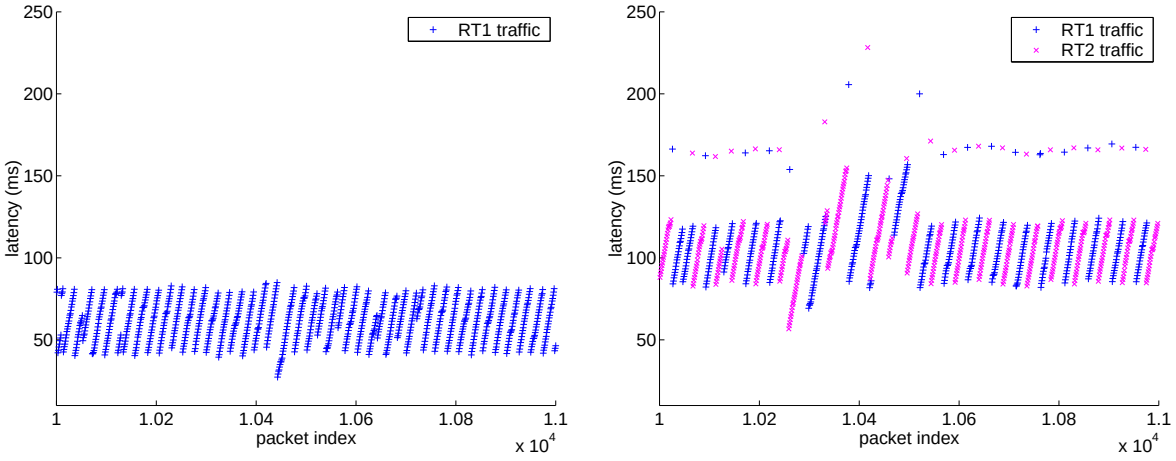
For all the traffic sources used in the experiments, packets are generated in succession and transmitted at 11 Mbps. The packet size is fixed at 1300 octets, the RTS/CTS option is turned off, and fragmentation is disabled.[2] Moreover, for a real-time packet, we measure the time interval between when it is generated and when it is successfully delivered by the NIC. This time interval is referred to as the latency the real-time packet experiences, which includes the queuing delay as well as the transmission delay. The duration of each experiment run is 45 seconds.

5.4.1 Peer-to-Peer Real-Time Streaming

In this experiment, only two laptops are communicating with each other. The transmitter has two real-time traffic sources, namely RT1 and RT2. The purpose is to show the benefit of applying the EDF policy to the RT queue.

First, we investigate the behavior of the original ORiNOCO driver. Figs. 5.7(a) and (b) represent the benchmark case when only RT1 is activated and the case when both sources are activated, respectively. We can see that RT1 latency in the benchmark case is always less than 100 ms, and when both traffic sources are activated, the latency performances of both RT1 and RT2 are equally affected and deviate significantly from the benchmark case. Based on this observation, in the following experiments, we set the relative deadline for RT1 traffic to 140 ms such that all the RT1 packets in the benchmark case will meet the deadline requirement, while a significant amount of RT1 packets will miss the deadline

[2] Experimental results, when the RTS/CTS option is turned on and/or fragmentation is enabled, yield very similar observations to what we will present in this section, and hence, are omitted.



(a) single RT traffic (benchmark)

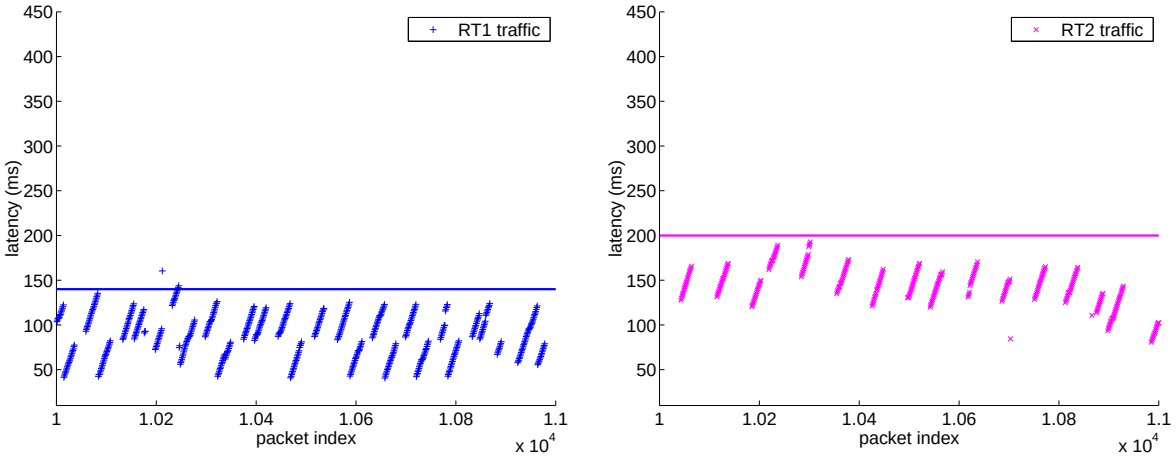
(b) two RT traffic

Figure 5.7: Latency comparison for RT traffic with a FIFO queue

when both traffic sources are activated. Then, we vary the relative deadline for RT2 traffic to see the benefit of applying EDF.

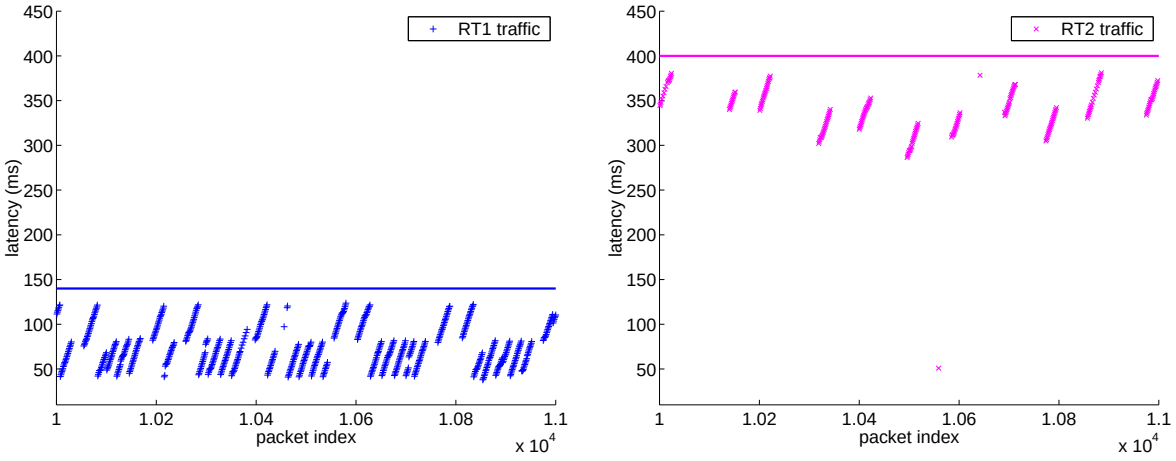
Now, we replace the original ORiNOCO driver with RT-WLAN. Fig. 5.8 shows the results when the relative deadline for RT2 traffic is set to 200 ms. The thick solid lines represent the relative deadlines for both traffic. Due to the less stringent deadline requirement of RT2 traffic, a higher transmission priority is given to RT1 traffic. As a result, less RT1 packets miss their deadlines at the expense of RT2 packets experiencing larger latencies. In Fig. 5.8, the integer number along the X-axis represents the order of the transmitted packets, which may belong to either RT1 or RT2. We can see that both sub-figures show certain degrees of data sparseness and the empty positions actually correspond to the packet transmissions from the other source. Clearly, more RT1 packets are transmitted. Similar observations can be found in Fig. 5.9, where the relative deadline for RT2 traffic is increased to 400 ms, and as expected, even less RT1 packets miss their deadlines and more transmission opportunities are offered to RT1.

In order to evaluate the benefit of using an EDF RT queue quantitatively, we calculate



(a) RT1 (*relative_deadline* = 140 ms) (b) RT2 (*relative_deadline* = 200 ms)

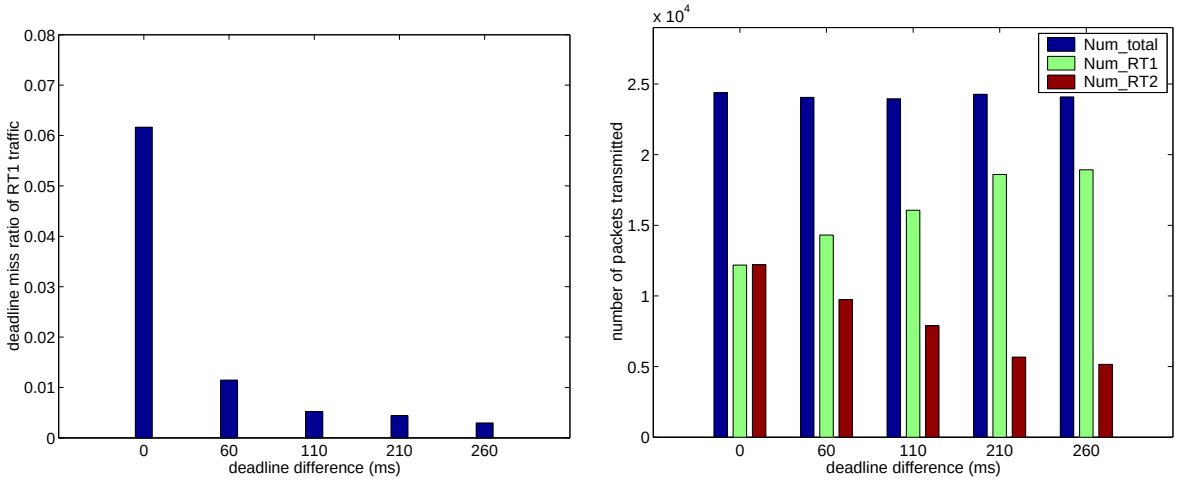
Figure 5.8: Latency comparison for RT traffic with an EDF queue



(a) RT1 (*relative_deadline* = 140 ms) (b) RT2 (*relative_deadline* = 400 ms)

Figure 5.9: Latency comparison for RT traffic with an EDF queue

the deadline miss ratio for RT1 traffic and show the results in Fig. 5.10(a). We also count the number of packets transmitted from either source, from RT1 only, and from RT2 only during the 45-second experiment run, and the results are shown in Fig. 5.10(b). The X-axis represents the difference of the relative deadlines of RT1 and RT2 traffic. Note that, when the deadline difference is zero, all the packets are actually served in a FIFO manner, so RT1 and RT2 are equally competing for the service. As a result, almost an equal number



(a) Deadline miss ratio for RT1

(b) Throughput comparison

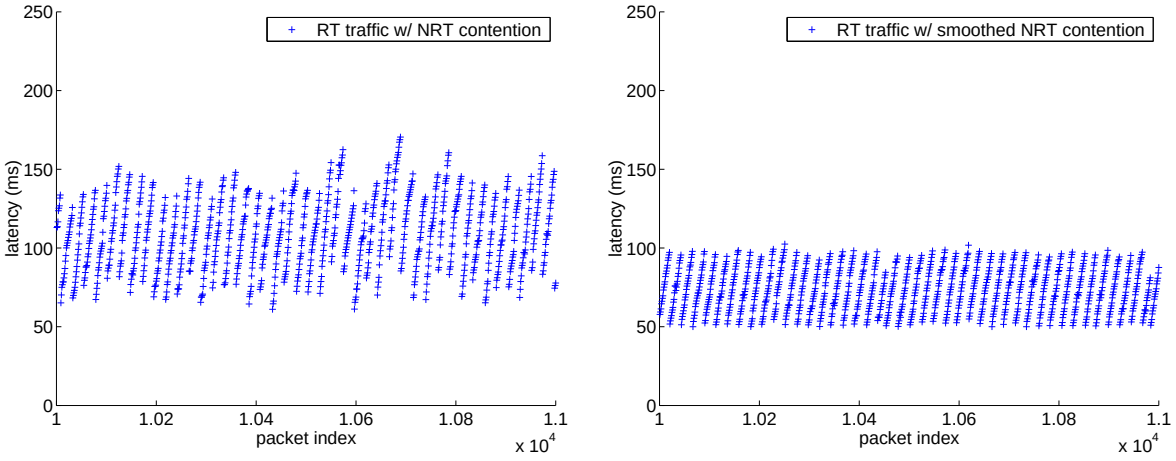
Figure 5.10: More experimental results for RT traffic with an EDF queue

of RT1 and RT2 packets are transmitted, and RT1 traffic presents a large deadline miss ratio (> 0.06). As the deadline difference increases, RT1 is assigned a higher transmission priority, thus resulting in a smaller deadline miss ratio and more shares of bandwidth. On the other hand, the total number of transmitted packets remains the same regardless of the deadline difference. Based on the above observations, we draw the following conclusion: by applying the simple EDF policy to the RT queue, we are able to achieve service differentiation among multiple real-time sessions with different deadline requirements without sacrificing the total throughput.

5.4.2 Real-Time Streaming in the Presence of Third-Party Non-Real-Time Traffic

In this experiment, three laptops are used. Two of them generate RT and NRT traffic, respectively, and the third laptop serves as the common receiver to both. RT and NRT traffic are contending for the shared wireless medium. The purpose is to show the benefit of applying adaptive traffic smoothing to NRT traffic.

We create two different scenarios in our experiment and compare their latency performances. First, NRT traffic is injected into the network through the original ORiNOCO



(a) RT (+ NRT contention) (b) RT (+ smoothed NRT contention)

Figure 5.11: Latency comparison for adaptive traffic smoothing

driver, and contends with RT traffic for the wireless medium without adaptive traffic smoothing. Second, the original ORiNOCO driver is replaced by RT-WLAN, and thus, NRT traffic is smoothed before contending for the wireless medium. The parameters of our adaptive traffic smoother are: $\alpha = 10$ ms, $\delta = 100$ μs, CBD $= 1500$ octets, $RP_{max} = 50$ ms, $RP_{min} = 3$ ms, and $\tau = 10$ ms. The corresponding results are plotted in Figs. 5.11(a) and (b), respectively. We can see that, without adaptive traffic smoothing, RT traffic experiences much higher latency due to the NRT contention. In contrast, with adaptive traffic smoothing, the latency performance of RT traffic is only slightly affected compared to the benchmark scenario, which is shown in Fig. 5.7(a). This is because the traffic smoother stops sending non-real-time packets and lowers its station input limit as soon as it finds out that its on-going packet transmission experiences contention and/or collision on the wireless medium.

We also compare the throughput performances for these two scenarios, and the results are shown in Fig. 5.12. We have three observations. First, without adaptive traffic smoothing, equal numbers of real-time and non-real-time packets are transmitted, because RT

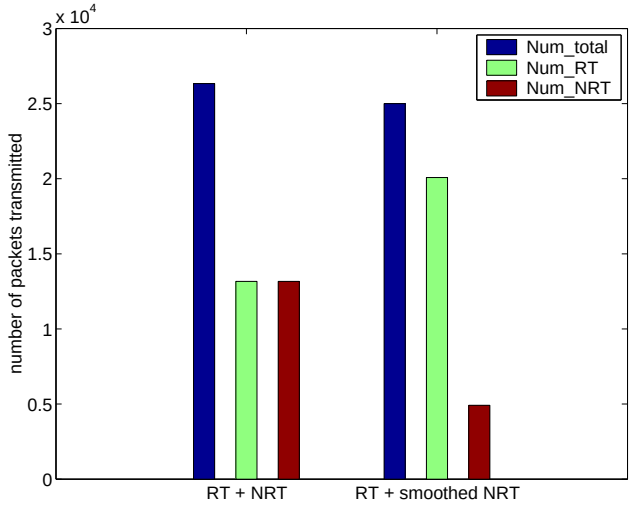


Figure 5.12: Throughput comparison for adaptive traffic smoothing

and NRT traffic are contending equally for the wireless medium. Second, with adaptive traffic smoothing, more real-time packets are transmitted, while still a reasonable number of non-real-time packets are served when the wireless medium is available. Third, there is about a 5% drop in the total throughput when adaptive traffic smoothing is applied. The rationale behind the drop is that the cautious nature of the adaptive traffic smoother results in a conservative transmission strategy for non-real-time packets. Therefore, the wireless medium may not be fully-utilized under our experimental setup.

5.5 Conclusion

In this chapter, we present the implementation details of RT-WLAN, our soft real-time extension to the original Linux device driver of the 802.11b-compliant Agere ORiNOCO cards. RT-WLAN is implemented as a loadable device driver module and is very easy to deploy. Users can simply replace the original ORiNOCO driver with RT-WLAN, and then realize soft real-time communications without having to re-compile the Linux kernel or change the NIC firmware.

RT-WLAN provides separate queues for RT and NRT traffic. The high-priority RT queue is served according to the EDF policy, while the low-priority NRT queue is served in a FIFO manner. Besides, an adaptive traffic smoother is implemented in RT-WLAN to regulate bursty NRT traffic before they are injected into the network, thus giving higher priority to in-progress real-time transmissions. Experimental results show that the latency of RT traffic is only slightly affected even when a significant amount of NRT network traffic is present, and the service differentiation among multiple real-time sessions is also achieved.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

6.1 Research Contributions

This dissertation addresses the problem of enhancing the performance of the 802.11 DCF systems from several related but distinct angles. Main contributions of the dissertation are as follows.

- **WB-DCF:**

  In Chapter II, we proposed a weighted-fair and bandwidth-efficient enhancement to the 802.11 DCF, called the WB-DCF, to achieve the weighted fairness and maximize the channel utilization for data communications in 802.11 DCF systems. The WB-DCF reduces the number of contending stations by introducing a new polling mode in addition to the contention mode used in the DCF, and achieves a low frame collision probability with an advanced contention window selection scheme based on runtime load estimation. Besides, the relative weights of traffic flows are also taken into consideration in the polling scheme and the contention window selection. This way, the weighted fairness is achieved. Simulation results show that, with few changes to the DCF, the WB-DCF performs significantly better in terms of throughput, delay, and fairness.

- **MiSer:**

  In Chapter III, we developed a novel intelligent TPC mechanism, called MiSer, as an optimal solution to the problem of minimizing the communication energy consumption in 802.11a/h DCF systems. The key idea of MiSer is to combine TPC with PHY rate adaptation and compute offline an optimal rate-power combination table, then at runtime, a wireless station determines the most energy-efficient transmission strategy for each data transmission attempt by a simple table lookup. MiSer is deployed in the format of RTS-CTS(strong)-Data(MiSer)-Ack to deal with the "hidden nodes" problem and to ameliorate the TPC-caused interference in the network. Simulation results show MiSer's clear superiority to the two-way frame exchange mechanisms in the presence of hidden nodes. Besides, compared with other four-way frame exchange mechanisms, MiSer delivers about 20% more data per unit of energy consumption than the PHY rate adaptation (alone without TPC) scheme, and outperforms single-rate TPC schemes significantly thanks to the excellent energy-saving capability of PHY rate adaptation.

- **ILA:**

  In Chapter IV, we presented a generic method to analyze the goodput performance of an 802.11a DCF system, then based on the theoretical analysis, we developed a novel MPDU-based link adaptation scheme, called ILA, for 802.11a DCF systems. Similar to MiSer, ILA is a table-driven approach and the basic idea is to pre-establish a best PHY mode table indexed by the system status before the communication starts. Then at runtime, a wireless station determines the most appropriate PHY mode for the next transmission attempt by a table lookup, using the most up-to-date system status as the index. Our in-depth simulation shows that ILA outperforms the single-mode schemes and the ARF scheme by Lucent Technologies significantly

in terms of the average goodput, the frame drop rate, and the average number of transmission attempts per data frame delivery.

- **RT-WLAN:**

  In Chapter V, we implemented a new RT-WLAN device driver module, which extends the original Linux device driver for the popular Agere ORiNOCO cards to support soft real-time communications. RT-WLAN uses separate queues for real-time and non-real-time traffic, and the service preference is given to the real-time queue. The real-time queue is served according to the EDF policy while the non-real-time queue is served in a FIFO manner. Besides, an adaptive traffic smoother is implemented in RT-WLAN to regulate bursty non-real-time traffic before they are injected into the network, thus giving higher priority to in-progress real-time transmissions. Experimental results show that the desired real-time support and service differentiation among multiple real-time sessions are achieved by using RT-WLAN.

6.2 Future Research Directions

This section describes several additional research issues that are related to performance enhancement of the 802.11 systems, which require further investigation.

- **QoS Provisioning in 802.11e Systems:**

  The upcoming 802.11e standard aims at providing QoS support in 802.11 systems. While the QoS mechanisms in the 802.11e, namely the EDCF (Enhanced Distributed Coordination Function) and the HCF (Hybrid Coordination Function), have been defined in the standard draft, the challenge lies in the configuration of these mechanisms to provide the desired services, such as data services with guaranteed throughputs or multimedia services with bounded delay/jitter.

- **Secure WLAN:**

  To enable widespread deployment of the public WLANs, it is crucial to protect the network against unknown and potentially malicious users, and to protect legitimate user communication. One interesting topic is countering DoS (Denial of Service) attacks in 802.11 systems. There have been significant research on countering DoS attacks in the Internet domain, while few considered it in the 802.11 systems. In fact, an 802.11 system is more vulnerable to DoS attacks because of the inherent design flaws in the 802.11 management and medium access protocols. Attackers can easily prevent legitimate users from accessing the network with a small number of bogus transmissions. We would like to develop application-level and middleware-level defense mechanisms to mitigate the underlying vulnerabilities of the 802.11 protocols and to effectively prevent malicious DoS attacks. Other related topics include distributed key management and secure user authentication/association.

- **Hybrid LAN-WAN System:**

  Although both 802.11-based WLANs and 3G cellular WANs (Wide-Area Networks) support data services and provide access to the Internet, the capabilities of these two types of wireless networks differ greatly. Most notably, the available WLAN bandwidth exceeds the WAN bandwidth by one to three degree of magnitude, while the WAN system offers a much broader geographical coverage. As an increasing number of multi-homed wireless/mobile computing and communication devices are being equipped with both 802.11 and 3G interfaces, it is naturally attractive to have a hybrid LAN-WAN system that integrates both networks and leverages the advantages of each other. Issues to address in this direction include resource management, bandwidth aggregation, voice and video over WLAN, and service discovery and routing protocols.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] http://www.wi-fihotspotlist.com/. Online Link.

[2] The Network Simulator – ns-2. http://www.isi.edu/nsnam/ns/. Online Link.

[3] Sharad Agarwal, Srikanth V. Krishnamurthy, Randy K. Katz, and Son K. Dao. Distributed Power Control in Ad-Hoc Wireless Networks. In *Proc. IEEE PIMRC'01*, pages 59–66, 2001.

[4] Agere Systems. *User's Guide for ORiNOCO PC Card*, Sep. 2000.

[5] Manish Anand, Edmund B. Nightingale, and Jason Flinn. Self-Tuning Wireless Network Power Management. In *Proc. ACM MobiCom'03*, pages 176–189, San Diego, CA, Sep. 2003.

[6] P. Bergamo, D. Maniezzo, A. Giovanardi, G. Mazzini, and M. Zorzi. An Improved Markov Chain Description for Fading Processes. In *Proc. IEEE ICC'02*, volume 3, pages 1347–1351, New York City, NY, Apr. 2002.

[7] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A Media Access Protocol for Wireless LAN's. In *Proc. ACM SIGCOMM'94*, pages 212–225, London, England UK, Aug. 1994.

[8] Giuseppe Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar. 2000.

[9] Giuseppe Bianchi, Luigi Fratta, and Matteo Oliveri. Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs. In *Proc. IEEE PIMRC'96*, pages 407–411, Taipei, Taiwan, Oct. 1996.

[10] Federico Cali, Marco Conti, and Enrico Gregori. IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism. *IEEE Journal on Selected Areas in Communications*, 18(9):1774–1786, Sep. 2000.

[11] Sunghyun Choi. PCF vs. DCF: Limitations and Trends. IEEE 802.11-01/154, Jan. 2001.

[12] R. L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation. *IEEE Trans. on Information Theory*, 37:114–131, Jan. 1991.

[13] DataMonitor.Com. Public Wireless LANs: Hotspots - Finally Heating Up? Report No. DMTC0921, Sep. 2003.

[14] Dr-Jiunn Deng and Ruay-Shiung Chang. A Nonpreemptive Priority-Based Access Control Scheme for Broadband Ad Hoc Wireless ATM Local Area Networks. *IEEE Journal on Selected Areas in Communications*, 18(9):1731–1739, Sep. 2000.

[15] Jean-Pierre Ebert and Andreas Willig. A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation. TKN Technical Report TKN-99-002, Technical University Berlin, Telecommunication Networks Group, Berlin, Germany, Mar. 1999.

[16] Jean-Pierre Ebert and Adam Wolisz. Combined Tuning of RF Power and Medium Access Control for WLANs. *Mobile Networks & Applications*, 5(6):417–426, Sep. 2001.

[17] Moncef Elaoud and Parameswaran Ramanathan. Adaptive Use of Error-Correcting Codes for Real-time Communication in Wireless Networks. In *Proc. IEEE INFOCOM'98*, volume 2, pages 548–555, San Francisco, CA, Mar. 1998.

[18] Abbas El Gamal, Chandra Nair, Balaji Prabhakar, Elif Uysal Biyikoglu, and Sina Zahedi. Energy-Efficient Scheduling of Packet Transmissions over Wireless Networks. In *Proc. IEEE INFOCOM'02*, volume 3, pages 1773–1782, New York City, NY, Jun. 2002.

[19] Javier Gomez, Andrew T. Campbell, Mahmoud Naghshineh, and Chatschik Bisdikian. Conserving Transmission Power in Wireless Ad Hoc Networks. In *Proc. IEEE ICNP'01*, pages 24–34, Nov. 2001.

[20] Steven D. Gray and Venkatesh Vadde. Throughput and Loss Packet Performance of DCF with Variable Transmit Power. IEEE 802.11-01/227, May 2001.

[21] Jorg Habetha and Diego Calvo de No. New Adaptive Modulation and Power Control Algorithms for HIPERLAN/2 Multihop Ad Hoc Networks. In *Proc. European Wireless (EW'2000)*, Dresden, Germany, Sep. 2000.

[22] David Haccoun and Guy Begin. High-Rate Punctured Convolutional Codes for Viterbi and Sequential Decoding. *IEEE Trans. on Communications*, 37(11):1113–1125, Nov. 1989.

[23] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. ACM MobiCom'01*, Rome, Italy, Jul. 2001.

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

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

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

[27] IEEE 802.11e. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*. Draft Amendment to IEEE 802.11 Standard, Draft 6.0, Nov. 2003.

[28] IEEE 802.11h. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Spectrum and Transmit Power Management extensions in the 5 GHz band in Europe*. Draft Supplement to IEEE 802.11 Standard, Draft 3.3.4, Feb. 2003.

[29] Lucent Technologies Inc. *Draft Software Interface Specification for Wireless Connection Interface for WaveLAN/IEEE (HCF-Light)*, Nov. 2001. Doc No. S0005, Rev. 11.

[30] Intersil Americas Inc. *2.4 GHz Power Amplifier and Detector*, Mar. 2000.

[31] Intersil Americas Inc. *Prism II 11Mbps Wireless Local Area Network PC Card*, Apr. 2001.

[32] Amit Jain, Daji Qiao, and Kang G. Shin. RT-WLAN: A Soft Real-Time Extension to the ORiNOCO Linux Device Driver. In *Proc. IEEE PIMRC'03*, Beijing, China, Sep. 2003.

[33] Eun-Sun Jung and Nitin H. Vaidya. An Energy Efficient MAC Protocol for Wireless LANs. In *Proc. IEEE INFOCOM'02*, volume 3, pages 1756–1764, New York City, NY, Jun. 2002.

[34] Ad Kamerman and Leo Monteban. WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal*, pages 118–133, Summer 1997.

[35] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints. In *Proc. ACM MobiCom'01*, Rome, Italy, Jul. 2001.

[36] Mark J. Karol, Zhao Liu, and Kai Y. Eng. Distributed-Queueing Request Update Multiple Access (DQRUMA) for Wireless Packet (ATM) Networks. In *Proc. IEEE ICC'95*, pages 1224–1231, Seattle, WA, 1995.

[37] Jamshid Khun-Jush, Peter Schramm, Udo Wachsmann, and Fabian Wenger. Structure and Performance of the HIPERLAN/2 Physical Layer. In *Proc. IEEE VTC'99 Fall*, Amsterdam, Netherlands, Sep. 1999.

[38] Ronny Krashinsky and Hari Balakrishnan. Minimizing Energy for Wireless Web Access with Bounded Slowdown. In *Proc. ACM MobiCom'02*, pages 119–130, Atlanta, GA, Sep. 2002.

[39] C. M. Krishna and Kang G. Shin. *Real-Time Systems*. McGraw Hill, 1997.

[40] Seok-Kyu Kweon, Kang G. Shin, and Gary Workman. Achieving Real-Time Communication over Ethernet with Adaptive Traffic Smoothing. In *Proc. IEEE Real-Time Technology and Applications Symposium*, pages 90–100, Jun. 2000.

[41] Paul Lettieri, Christina Fragouli, and Mani B. Srivastava. Low Power Error Control for Wireless Links. In *Proc. ACM MobiCom'97*, pages 139–150, Budapest, Hungary, 1997.

[42] Paul Lettieri and Mani B. Srivastava. Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency. In *Proc. IEEE INFOCOM'98*, Mar. 1998.

[43] Zihuai Lin, Goran Malmgren, and Johan Torsner. System Performance Analysis of Link Adaptation in HiperLAN Type 2. In *Proc. IEEE VTC'00*, pages 1719–1725, 2000.

[44] Songwu Lu, Thyagarajan Nandagopal, and Vaduvur Bharghavan. A Wireless Fair Service Algorithm for Packet Cellular Networks. In *Proc. ACM MobiCom'98*, pages 10–20, Dalla, TX, Oct. 1998.

[45] Haiyun Luo, Songwu Lu, and Vaduvur Bharghavan. A New Model for Packet Scheduling in Multihop Wireless Networks. In *Proc. ACM MobiCom'00*, pages 76–86, Boston, MA, Aug. 2000.

[46] Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *Proc. European Wireless (EW'2002)*, Florence, Italy, Feb. 2002.

[47] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *Proc. ACM MobiCom'00*, pages 87–98, Boston, MA, Aug. 2000.

[48] Thyagarajan Nandagopal, Songwu Lu, and Vaduvur Bharghavan. A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms. In *Proc. ACM MobiCom'99*, pages 132–142, Seattle, WA, Aug. 1999.

[49] T. S. Eugene Ng, Ion Stoica, and Hui Zhang. Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors. In *Proc. IEEE INFOCOM'98*, pages 1103–1111, San Francisco, CA, Mar. 1998.

[50] Bob O'Hara and Al Petrick. *The IEEE 802.11 Handbook: A Designer's Companion*. Standards Information Network, IEEE Press, 1999.

[51] Timucin Ozugur, Mahmoud Naghshineh, Parviz Kermani, C. Michael Olsen, Babak Rezvani, and John A. Copeland. Balanced Media Access Methods for Wireless Networks. In *Proc. ACM MobiCom'98*, pages 21–32, Dallas, TX, Oct. 1998.

[52] Balaji Prabhakar, Elif Uysil Biyikoglu, and Abbas El Gamal. Energy-Efficient Transmission over a Wireless Link via Lazy Packet Scheduling. In *Proc. IEEE INFOCOM'01*, volume 1, pages 386–394, Anchorage, AK, Apr. 2001.

[53] Ramjee Prasad and Tero Ojanpera. An Overview of CDMA Evolution toward Wideband CDMA. *IEEE Communications Surveys*, 1(1):2–29, Fourth Quarter 1998.

[54] J. G. Proakis. *Digital Communications*. McGraw Hill, 3rd edition, 1995.

[55] Michael B. Pursley and D. J. Taipale. Error Probabilities for Spread-Spectrum Packet Radio with Convolutional Codes and Viterbi Decoding. *IEEE Trans. on Communications*, COM-35(1):1–12, Jan. 1987.

[56] Daji Qiao and Sunghyun Choi. Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation. In *Proc. IEEE ICC'01*, Helsinki, Finland, Jun. 2001.

[57] Daji Qiao, Sunghyun Choi, Amit Jain, and Kang G. Shin. Adaptive Transmit Power Control in IEEE 802.11a Wireless LANs. In *Proc. IEEE VTC'03-Spring*, Jeju, Korea, Apr. 2003.

[58] Daji Qiao, Sunghyun Choi, Amit Jain, and Kang G. Shin. MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h. In *Proc. ACM MobiCom'03*, pages 161–175, San Diego, CA, Sep. 2003.

[59] Daji Qiao, Sunghyun Choi, and Kang G. Shin. Inteference Analysis and Transmit Power Control in IEEE 802.11a/h Wireless LANs. Submitted to IEEE/ACM Trans. on Networking (TON), Nov. 2003.

[60] Daji Qiao, Sunghyun Choi, Amjad Soomro, and Kang G. Shin. Energy-Efficient PCF Operation of IEEE 802.11a Wireless LAN. In *Proc. IEEE INFOCOM'02*, volume 2, pages 580–589, New York City, NY, Jun. 2002.

[61] Daji Qiao and Kang G. Shin. A Two-Step Adaptive Error Recovery Scheme for Video Transmission over Wireless Networks. In *Proc. IEEE INFOCOM'00*, volume 3, pages 1698–1704, Tel-Aviv, Israel, Mar. 2000.

[62] Daji Qiao and Kang G. Shin. Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF. In *Proc. IWQoS'02*, pages 227–236, Miami Beach, FL, May 2002.

[63] Daji Qiao and Kang G. Shin. Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. *IEEE Trans. on Mobile Computing (TMC)*, 1(4):278–292, Oct.-Dec. 2002.

[64] Daji Qiao and Kang G. Shin. UMAV: A Simple Enhancement to the IEEE 802.11 DCF. In *Proc. HICSS-36*, Hawaii, HI, Jan. 2003.

[65] Daji Qiao and Kang G. Shin. Weighted-Fair and Bandwidth-Efficient Enhancement to the IEEE 802.11 DCF. Submitted to IEEE Trans. on Mobile Computing (TMC), Sep. 2003.

[66] Olav Queseth, Fredrik Gessler, and Magnus Frodigh. Algorithms for Link Adaptation in GRPS. In *Proc. IEEE VTC'99*, pages 943–947, 1999.

[67] Parameswaran Ramanathan and Prathima Agrawal. Adapting Packet Fair Queueing Algorithms to Wireless Networks. In *Proc. ACM MobiCom'98*, pages 1–9, Dalla, TX, Oct. 1998.

[68] T. S. Rappaport. *Wireless Communications: Principle and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[69] M. Rofougaran, A. Rofougaran, C. Olgaard, and A. A. Abidi. A 900 MHz CMOS RF Power Amplifier with Programmable Output. In *1994 Symposium on VLSI Circuits Digest of Technical Papers*, pages 133–134, Jun. 1994.

[70] John F. Sevic. Statistical Characterization of RF Power Amplifier Efficiency for CDMA Wireless Communication Systems. In *Proc. Wireless Communications Conference*, pages 110–113, Boulder, CO, Aug. 1997.

[71] Srikant Sharma, Kartik Gopalan, Ningning Zhu, Gang Peng, Pradipta De, and Tzi cker Chiueh. Implementation Experiences of Bandwidth Guarantee on a Wireless LAN. In *Proc. ACM/SPIE Multimedia Computing and Networking (MMCN 2002)*, Jan. 2002.

[72] Tajana Simunic, Luca Benini, Peter Glynn, and Giovanni De Micheli. Dynamic Power Management for Portable Systems. In *Proc. ACM MobiCom'00*, pages 11–19, Boston, MA, Aug. 2000.

[73] Mark Stemm, Paul Gauthier, Daishi Harada, and Randy H. Katz. Reducing Power Consumption of Network Interfaces in Hand-Held Devices. In *Proc. 3rd International Workshop on Mobile Multimedia Communications*, Princeton, NJ, Sep. 1996.

[74] Jean Tourrilhes. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Orinoco. Online link.

[75] Nitin H. Vaidya, Paramvir Bahl, and Seema Gupta. Distributed Fair Scheduling in a Wireless LAN. In *Proc. ACM MobiCom'00*, pages 167–178, Boston, MA, Aug. 2000.

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

[77] Haitao Wu, Yong Peng, Keping Long, Shiduan Cheng, and Jian Ma. Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement. In *Proc. IEEE INFOCOM'02*, volume 2, pages 599–607, New York City, NY, Jun. 2002.

[78] Kaixin Xu, Mario Gerla, and Sang Bae. How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Network. In *Proc. IEEE GlobeCom'02*, Taipei, Taiwan, Nov. 2002.

[79] Hong Ye, Gregory C. Walsh, and Linda G. Bushnell. Real-Time Mixed-Traffic Wireless Network. *IEEE Trans. on Industrial Electronics*, 48(5):883–890, Oct. 2001.

[80] Michele Zorzi, Ramesh R. Rao, and Laurence B. Milstein. Error Statistics in Data Transmission over Fading Channels. *IEEE Trans. on Communications*, 46(11):1468–1477, Nov. 1998.