Quality of Service Enhancements in IEEE 802.11 Wireless LANs

by

George Wai Wong

B.A.Sc., The University of British Columbia, Canada, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
September 2003

Department of Electrical and Computer Engineering

The University of British Columbia
Vancouver, Canada

Date: September 26th, 2003.

Abstract

One of the most important of the many developments in telecommunications is the convergence of voice, video, and data communications within the Internet Protocol (IP) suite. IP was originally designed to support best-effort data services. The development of IP-based multimedia networking applications has imposed Quality of Service (QoS) requirements on the IP network, especially with respect to real-time traffic. The Internet Engineering Task Force (IETF) is currently working on QoS differentiation at the IP layer; however the result is sub-optimal without lower layers' support. With the increasing use of wireless Internet services over 802.11 wireless LANs, it is essential to focus on QoS differentiation support at the 802.11 MAC layer.

To improve the current 802.11 MAC protocol, the IEEE Task Group E was formed and is defining QoS enhancements for the 802.11 MAC layer by introducing an Enhanced Distributed Coordination Function (EDCF). The EDCF provides prioritization enhancement of the 802.11 Distributed Coordination Function (DCF).

The objective of this thesis is to propose and evaluate a novel packet retransmission algorithm called Age-Dependent Backoff to improve the QoS performance of EDCF. The ADB algorithm dynamically varies the persistent factor associated with the contention window, based on real-time packet queue age and lifetime. ADB maintains the backward compatibility property of EDCF and involves relatively easy implementation at low cost.

Extensive simulation results obtained using OPNET software show that EDCF with the ADB retransmission algorithm provides low values for delay, jitter, and drop rate for real-time traffic without sacrificing the throughput of best-effort data traffic. ADB is a viable, novel and low cost means to improve the QoS performance of EDCF.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I wish to thank most sincerely my supervisor, Dr. Robert W. Donaldson, for his inexhaustible patience, encouragement and valuable assistance in connection with this thesis. I would like to express my thanks to my colleague, Victor Fong, for participating in technical and non-technical discussions with me. I would also like to thank OPNET technologies, Inc. for providing us a free academic version of OPNET Modeler.

Finally, I am most grateful to my parents not only for their support but also for their constant encouragement.

Chapter 1 Introduction

In this introductory chapter, background material on IEEE 802.11 wireless LANs is reviewed in section 1.1. The motivations for this thesis work are presented in section 1.2. The thesis contributions are summarized in section 1.3. Finally, the outline of the thesis is provided in section 1.4.

1.1 Background on IEEE 802.11 Wireless LANs

Wireless LAN products first appeared in the late 1980s, marketed as substitutes for traditional wireline LANs [1]. The idea was to use wireless LANs to avoid the cost of installing LAN cables and to ease the task of relocating computer stations. The history of wireless LAN development has been fraught with proprietary non-standard technologies which cause wireless LAN products being non-interoperable among vendors. With the ratification of the IEEE 802.11 wireless LAN standard, wireless LANs have emerged from being proprietary implementations to become open solutions for providing mobility as well as essential network services where wireline installations are impractical [2].

The original IEEE 802.11 standard [3], published in 1997, supports data rates up to 2 Mbps in the 2.4-GHz industrial, scientific, and medical (ISM) band. In 1999, The IEEE 802.11 Working Group published its enhanced versions, IEEE 802.11a [4] and 802.11b [5], that extend the data rate to 54 Mbps in the 5-GHz unlicensed national information infrastructure (UNII) band and 11 Mbps in the 2.4-GHz ISM band, respectively. Recently, the IEEE has approved the final specification for IEEE 802.11g [6], which is backwards compatible with 802.11b and which boosts the bandwidth capability to 54 Mbps in the 2.4-GHz ISM band. Although most of the growth in 802.11 wireless LANs has been around the 802.11b standard, pundits predict that the emergence of the 802.11g standard will dominate the 802.11 wireless LAN market because of the full backward compatibility with 802.11b and the high data rate of 54 Mbps [7]. Despite not being finalized until recently, many vendors have already been shipping 802.11g products based on the drafts of the 802.11g standard since late 2002. Following the approval of 802.11g, the formation of IEEE 802.11 Task Group N began. The data rate of 802.11n has not been determined yet but is expected to be at least 108 Mbps or possibly beyond up to 320 Mbps.

The IEEE 802.11 wireless LAN standard is based on the definition of the medium access control (MAC) protocol and the physical layer (PHY) specifications. The PHY defines frequency bands, data rates, and other details of the actual wireless transmission. Above the PHY is the MAC layer, which regulates access to the shared wireless channel so that station transmissions do not interfere with one another. The 802.11 MAC layer was designed to be common among different 802.11 PHYs such as 802.11a, 802.11b and 802.11g. The 802.11 MAC layer protocol is specified in terms of coordination functions that determine when a wireless station is allowed to transmit data over the wireless medium. The mandatory Distributed Coordination Function (DCF) uses a carrier-sense multiple access/collision avoidance (CSMA/CA) protocol for sharing the wireless medium. This CSMA/CA protocol reduces the probability of collisions among stations which share the medium by using a random backoff time when the medium is busy. Once the medium is idle, a random backoff time defers a station from transmitting a frame, minimizing the chance of inter-station collisions.

802.11 wireless LANs are today's most deployed wireless LANs and are expected to play a major role in next-generation wireless communications. The increasing use of wireless Internet services has created a strong demand for public Internet access over wireless LANs. It is envisioned that 802.11 will soon become one of the most common wireless Internet access technologies.

1.2 Motivations

The world of communications has undergone many changes over the last few years. One of the most important changes is the convergence of voice, video, and data communications within the Internet Protocol (IP) suite. IP was originally designed to support data services such as file transfer, e-mail and remote terminal, which are tolerant of delay and jitter. Voice and video services, as opposed to data services, require a minimum transmission rate and suffer significantly from high delay and jitter. The development of IP-based multimedia networking applications has imposed additional requirements on the IP network, creating a need for end-to-end Quality of Service (QoS) support. Although the Internet Engineering Task Force (IETF) is currently working on service differentiation at the IP layer, the result is sub-optimal without lower layers' support. In recent years, there has been a substantial increase in the deployment of 802.11 wireless LANs for wireless Internet services. Wireless Internet Service Providers (WISPs) are appearing everywhere, deploying 802.11 hotspots in coffee shops, hotels, and airports. With the growing popularity and acceptance of 802.11 wireless LANs, it is essential to focus on service differentiation support at the 802.11 MAC layer.

To improve the current 802.11 MAC protocol to support applications with QoS requirements, the IEEE 802.11 Task Group E was formed and is defining QoS enhancements for the 802.11 MAC protocol. The 802.11e draft introduces Enhanced Distributed Coordination Function (EDCF) and Hybrid Coordination Function (HCF), both of which are currently under discussion [8]. EDCF is a prioritization enhancement of the 802.11 DCF using the Virtual DCF mechanism [9]. EDCF is the contention-based channel access mechanism for HCF which is based on a polling mechanism similar to the 802.11 Point Coordination Function (PCF) [10]. HCF allows the Hybrid Coordinator (HC), typically located at the access point (AP), to initial contention-free periods at any given time during a contention period.

The success of 802.11 wireless LANs is based on DCF, which is a distributed protocol with minimum management and maintenance costs. The dynamic nature of ad-hoc networks makes it difficult to use a polling protocol relying on centralized control to maintain connection, reservation, and scheduling states. With these observations, we argue that a distributed control of wireless channel results in a more productive use of wireless resources than a centralized control. Therefore, we will focus on the improvement of EDCF in this thesis.

1.3 Thesis Contributions

The thesis extends the current work on providing QoS enhancements in 802.11 wireless LANs studied by IEEE 802.11 Task Group E. In particular, we address the improvement of the new 802.11e EDCF by proposing an efficient retransmission algorithm called Age-Dependent Backoff (ADB) [11], which dynamically adjusts the persistent factors based on the ages of the real-time packets in the transmission queues and the lifetimes of the real-time packets. The complexity of implementing the ADB algorithm is relatively low. ADB requires minor modifications in the computation of CW minimizing the migration effort from the 802.11e EDCF and provides backward compatibility to the 802.11 DCF.

The performance of EDCF with the ADB retransmission algorithm is evaluated using the OPNET simulation tools. We test the ADB algorithm in two typical environments, an ad-hoc network and a hotspot, in which 802.11 wireless LANs might most probably be applied. The ad-hoc network is used mostly in situations where users need to deploy a network to start communication quickly without the benefit of a fixed network infrastructure. The hotspot is usually deployed by a WISP, providing wireless Internet access services for public use. In both environments, voice, video and data services are active simultaneously. Two simulation scenarios are implemented using OPNET to model these two common environments. We study the improvements in service differentiation under the environments when the ADB retransmission algorithm is employed in the new 802.11e EDCF protocol.

The simulation results indicate that using ADB in EDCF provides low values for delay, jitter and drop rate for real-time traffic without sacrificing the throughput of the best-effort traffic in a wide range of traffic loads and network configurations.

1.4 Thesis Outline

The subsequent chapters of this thesis are organized as follows. In Chapter 2, we review the 802.11 standard and introduce the new 802.11e EDCF protocol proposed by the 802.11 Task Group E. In Chapter 3, we describe our proposed ADB retransmission algorithm for EDCF and discuss the complexity of ADB. The simulation model is presented in Chapter 4 and the performance analysis of 802.11e EDCF with ADB is studied in Chapter 5. Finally, a summary and conclusions are provided in Chapter 6.

Chapter 2 Overview of IEEE 802.11 Wireless LANs

Before studying the details of anything, it often helps to get a general "lay of the hand." A basic introduction is often necessary when studying networking topics because the number of acronyms can be overwhelming. The 802.11 standard uses a significant number of acronyms, which makes the introduction to be important. This chapter introduces the acronyms used throughout the entire thesis and provides an overview of the IEEE 802.11 wireless LAN standard. The introduction to the 802.11 wireless LAN standard appears in Section 2.1. The 802.11 DCF, a mandatory 802.11 MAC protocol, mentioned briefly in the previous chapter is presented thoroughly in Section 2.2. Finally, Section 2.3 describes the new 802.11e EDCF protocol proposed by the 802.11 Task Group E for supporting service differentiation at the MAC layer of 802.11 wireless LANs.

2.1 Introduction to the IEEE 802.11 Standard

The 802.11 wireless LAN standard, officially called "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," defines over-the-air protocols necessary to support networking in a local area. As with other IEEE 802-based standards (such as 802.3 Ethernet and 802.5 Token Ring), the primary service of the 802.11 standard is to deliver MAC Service Data Units (MSDUs) between peer Logical Link Controls (LLCs). As a refresher for readers, the IEEE Standards Committee subdivided the Data Link layer in the open systems interconnection (OSI) reference model developed by the International Organization for Standardization (ISO). The result of this subdivision depicted in Figure 2.1 split the Data Link layer into a MAC layer and a LLC layer [12]. LLC is the highest layer of the IEEE 802 reference model, providing addressing and data link control, and is independent of the network topology, the transmission medium, and the MAC protocol. The 802.11 standard specifies MAC protocols and PHY specifications for wireless connectivity of fixed, portable, and mobile stations moving at pedestrian or vehicular speed within a local area. Wireless cards and access points typically provide functions of the 802.11 standard.

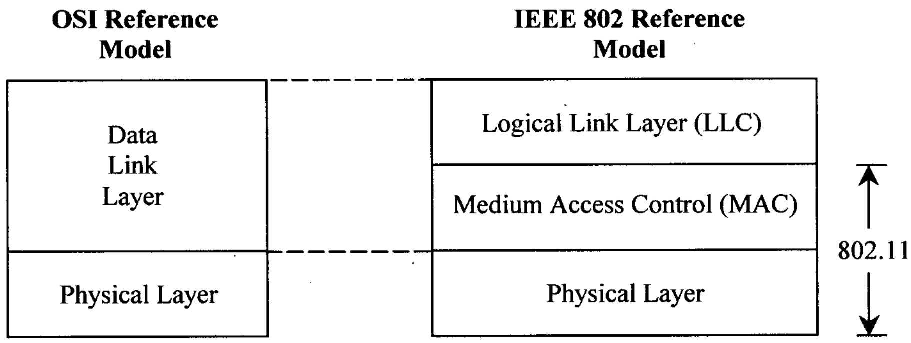


Figure 2.1. IEEE reference model comparing with the ISO reference model

2.1.1 802.11 Topology

The basic building block of a 802.11 wireless LAN is a basic service set (BSS), which consists of stations that execute the same MAC protocol and compete for access to the same shared wireless medium. An independent BSS (IBSS) is a standalone BSS that has no backbone infrastructure and consists of at least two wireless stations as shown in Figure 2.2. This type of network configuration is often referred to as an ad-hoc network. A BSS may be connected to a backbone distribution system through an access point (AP) to form an extended service set (ESS) as illustrated in Figure 2.3. The term infrastructure network is used informally to refer to the ESS. The BSSs are like cells in a cellular network and the APs are analogous to base stations in the cellular network [13]. This type of configuration satisfies the needs of large coverage networks of arbitrary size and complexity.

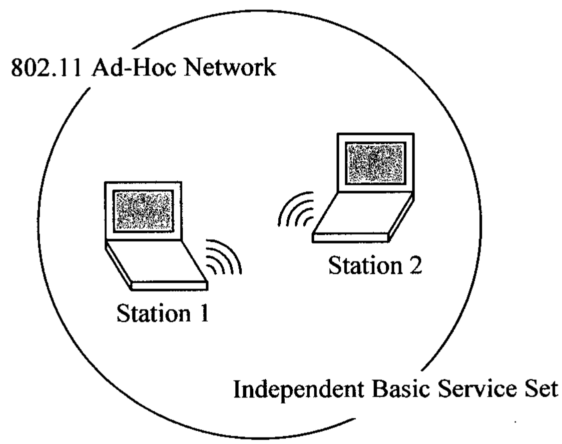


Figure 2.2. Independent basic service set

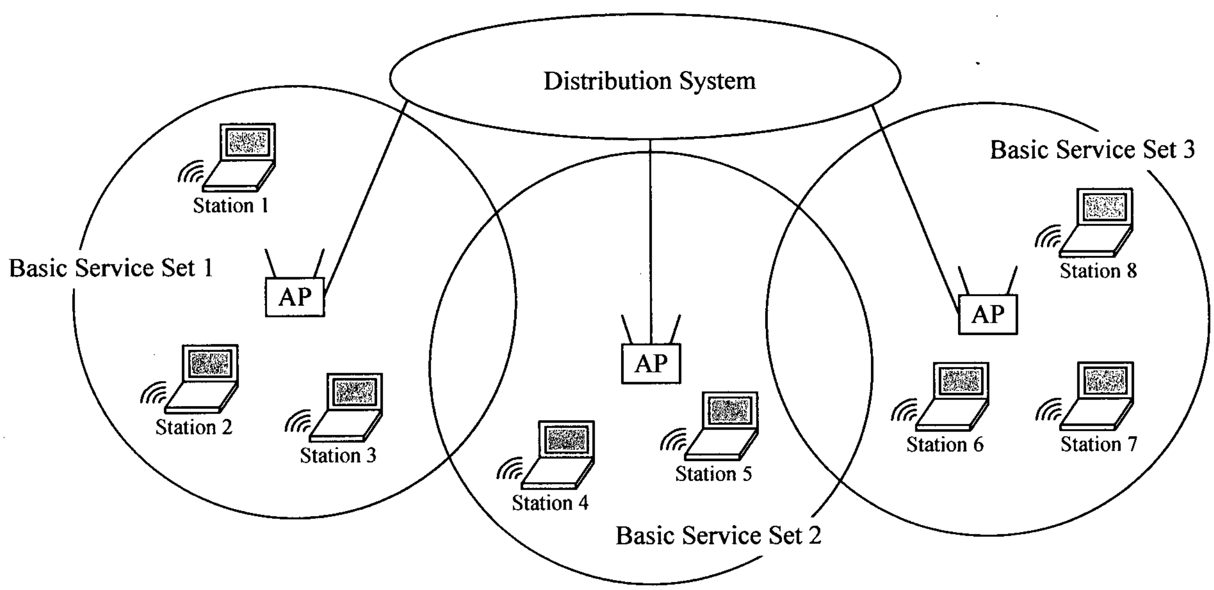


Figure 2.3. Extended service set

2.1.2 802.11 Protocol Architecture

Figure 2.4 depicts the protocol architecture of the 802.11 standard. The channel access for the wireless stations in a BSS is under the control of the MAC layer with two sublayers, namely Distributed Coordination Function (DCF) and Point Coordination Function (PCF). DCF is a contention-based MAC protocol providing asynchronous data transfer on a best-effort basis and is a mandatory MAC protocol supported by all stations. The access control in ad-hoc networks uses DCF only. Infrastructure networks can operate using just DCF or a coexistence of DCF and PCF. Details of the DCF operation will be described in section 2.2.

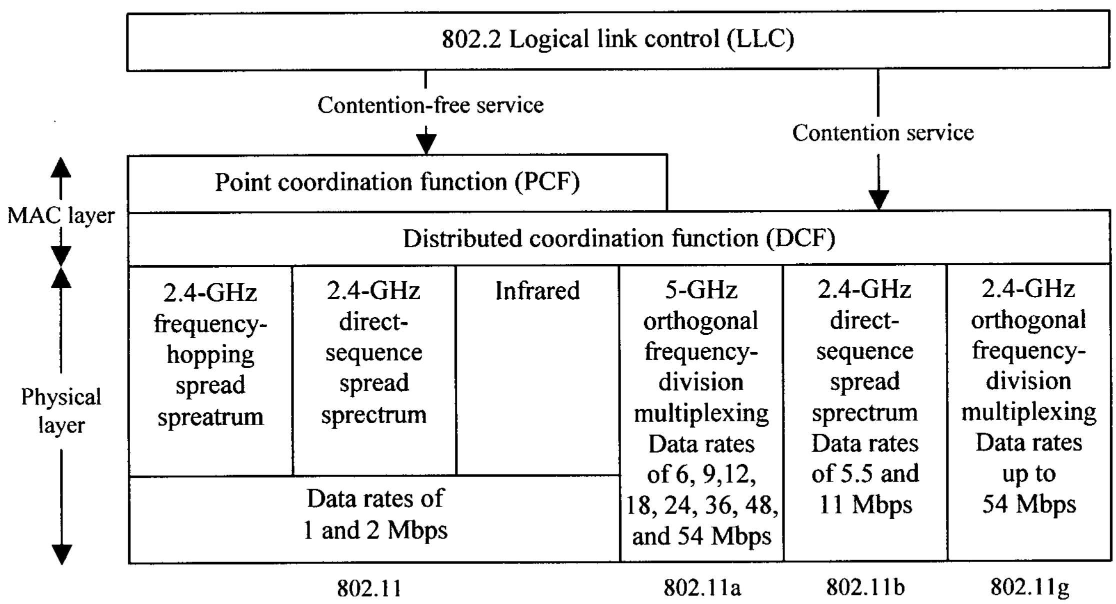


Figure 2.4. 802.11 protocol layers

PCF is a centralized MAC protocol providing contention-free multiple access service for the transmission of real-time traffic by polling stations in turn. Studies have shown that it is difficult for PCF to achieve high efficiency and to satisfy the time requirement for real-time traffic, [14], [15] and the cooperation between DCF and PCF leads to poor performance [16]. PCF is not supported by commercially available wireless cards and APs.

The PHY specifies the actual transmission details and has been the focus of much work in the past few years. The 802.11 standard specifies several PHYs. The initial standard approved in 1997 included frequency hopping spread spectrum (FHSS) and direct-sequence spread spectrum (DSSS) delivering data rates of 1 and 2 Mbps in the 2.4-GHz ISM band. The standard also defined an infrared option operating at 1 and 2 Mbps via passive ceiling reflection; one reason that the infrared option has never gained wide market support might be because infrared transmissions cannot penetrate walls. Most of the early 802.11 wireless LANs are FHSS due to the relative implementation simplicity.

In 1999, the IEEE issued the second and third PHYs, IEEE 802.11a and 802.11b, at roughly the same time. 802.11a operates in the 5-GHz UNII band at data rates up to 54 Mbps using orthogonal frequency division multiplexing (OFDM); 802.11b operates in the 2.4-GHz ISM band with 5.5 and 11 Mbps data rates using DSSS. The IEEE has recently approved the final specification for 802.11g which adopts 802.11a's OFDM but runs 54 Mbps in the 2.4-GHz ISM band and is backwards compatible with 802.11b. Most 802.11 wireless LANs deployed today comply with the 802.11b standard.

2.2 802.11 DCF Protocol

The fundamental access method of the 802.11 MAC protocol is DCF, which supports asynchronous data transfer on a best-effort basis and is the only possible MAC protocol in 802.11 ad-hoc networks. The 802.11 DCF MAC protocol operation is depicted in Figure 2.5.

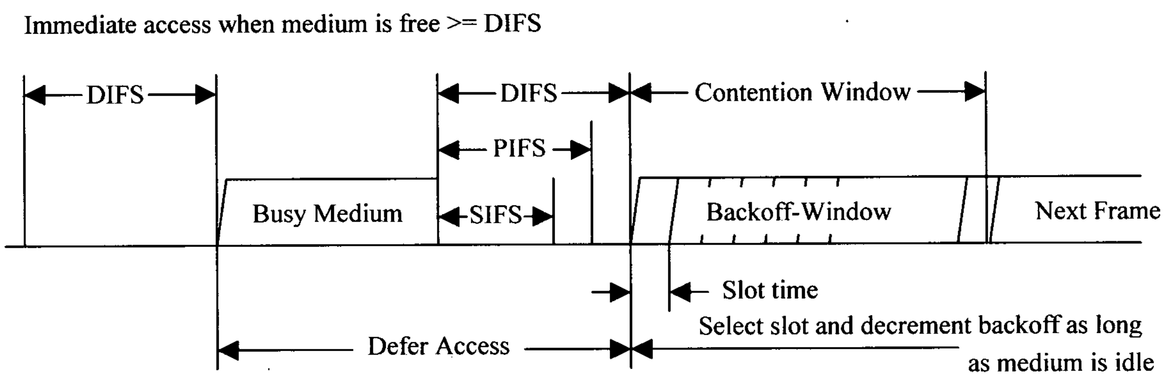


Figure 2.5. 802.11 DCF MAC protocol operation

For a station to transmit a MAC protocol data unit (MPDU), it must sense the medium to determine if another station is transmitting and must ensure that the medium is idle for the specified DCF Interframe Space (DIFS) duration. A station may transmit a pending MPDU when it determines that the medium is idle for a time interval greater than or equal to the DIFS period. If the medium is sensed busy, the station continues to sense the channel for an additional random time after detecting the channel as being idle for the DIFS duration. The additional random time period is selected from CW. The size of CW, bounded by the maximum value CW_{max}, is doubled after each unsuccessful transmission to reduce the collision probability. Following each successful transmission, CW is reset to the minimum value CW_{min}. This is the well-known binary exponential backoff (BEB) algorithm. The backoff time, *backoff_time*, can be expressed as follows [17]:

$$backoff\_time = randInt(0, \min(CW_{\min} \times 2^{retry}, CW_{\max})) \times slot\_time \quad (1)$$

In (1), *randInt*(*a*, *b*) generates a random integer in the range from *a* to *b* uniformly, *min*(*c*, *d*) gives the smaller value of *c* and *d*, *retry* is the number of retransmission attempts, and *slot_time* is a time duration specified by the physical layer parameters.

During backoff, the station decreases its backoff counter by one if the medium is idle for a *slot_time* period and freezes the backoff counter when the medium is busy. When the backoff counter reaches zero, the station will transmit its MPDU immediately. When a destination station receives the MPDU successfully, it returns an Acknowledgement (ACK) frame to the source station after a Short Interframe Space (SIFS) duration.

DCF offers an optional means of transmitting data frames that requires the transmission of Request To Send (RTS) and Clear To Send (CTS) frames prior to the transmission of the actual data frames. The RTS/CTS transmission scheme can alleviate the hidden terminal problem and can reduce the transmission time wasted as a result of a collision due to the longer frame size of the actual MPDU. The RTS and CTS frames include information regarding the transmission time of the next data frame and the corresponding ACK frame. The Network Allocation Vector (NAV) maintained by each station is an indicator of time periods when other stations close to the transmitting station and hidden stations close to the receiving station will not commence any transmissions. DCF with the RTS/CTS transmission scheme and the NAV settings of other stations are shown in Figure 2.6.

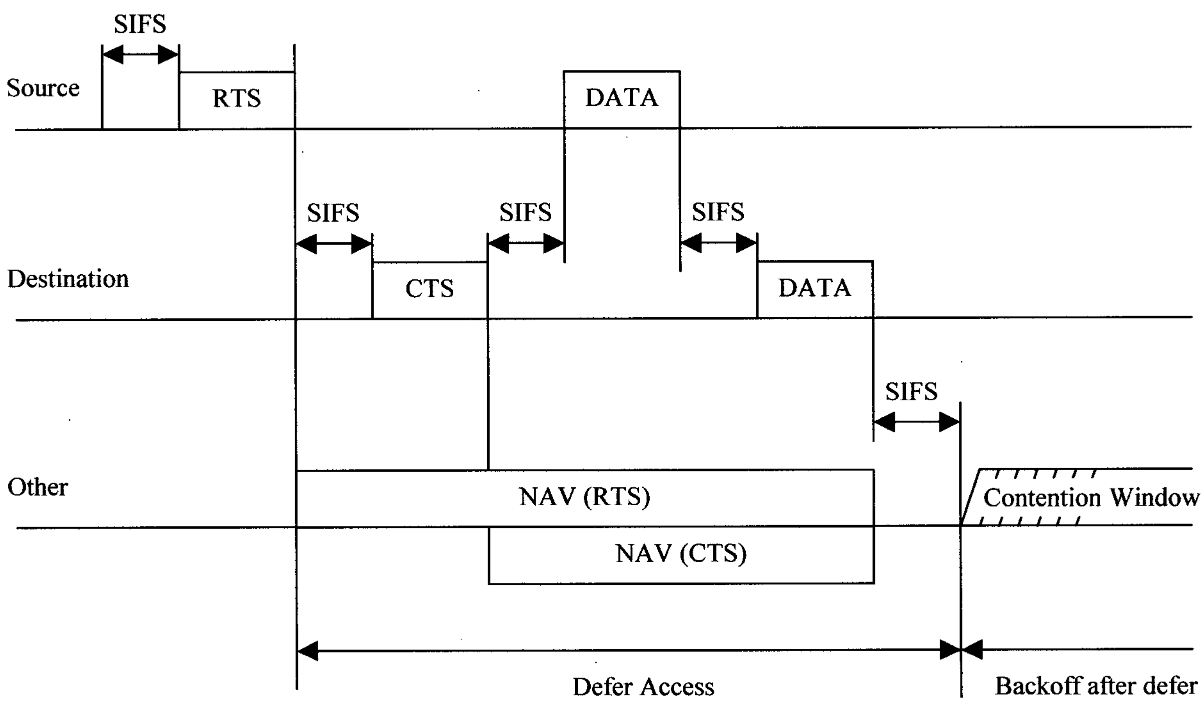


Figure 2.6. RTS/CTS/data/ACK and NAV settings

2.3 New 802.11e EDCF Protocol

The new 802.11e EDCF medium access scheme is governed by a distributed mechanism very similar to 802.11 DCF. Service differentiation is achieved through the introduction of Traffic Categories (TCs). Each TC has a different transmission queue and each transmission queue has a different interframe space (Arbitrary Interframe Space – AIFS[TC]), a different set of contention window limits (CW_{min}[TC] and CW_{max}[TC]), and a different persistent factor (PF[TC]).

Figure 2.7 illustrates the service differentiation accomplished by using different AIFS values. Each TC within a station starts a backoff independently after detecting the channel as idle for an AIFS[TC] duration. In the EDCF retransmission scheme, the size of the new CW[TC] after an unsuccessful transmission is determined by expanding/reducing the size of the previous CW[TC] by a factor of PF[TC], whereas in legacy 802.11 DCF, CW is always double after every unsuccessful transmission, i.e. PF=2. As in legacy DCF, the CW[TC] never exceeds its maximum bound CW_{max}[TC]. A random backoff counter is chosen from the interval [0, CW[TC]] in the case of AIFS[TC]≥DIFS and from [1, CW[TC]+1] in the case of AIFS[TC]<DIFS [18]. When the backoff counter of a TC reaches zero, the station transmits a pending MSDU from the corresponding transmission queue. A short AIFS, a small size of CW limits, and a low PF value are associated with high priority packets, enabling them to start contenting the channel earlier and to complete the backoff sooner, thus offering a high probability of winning the contention race.

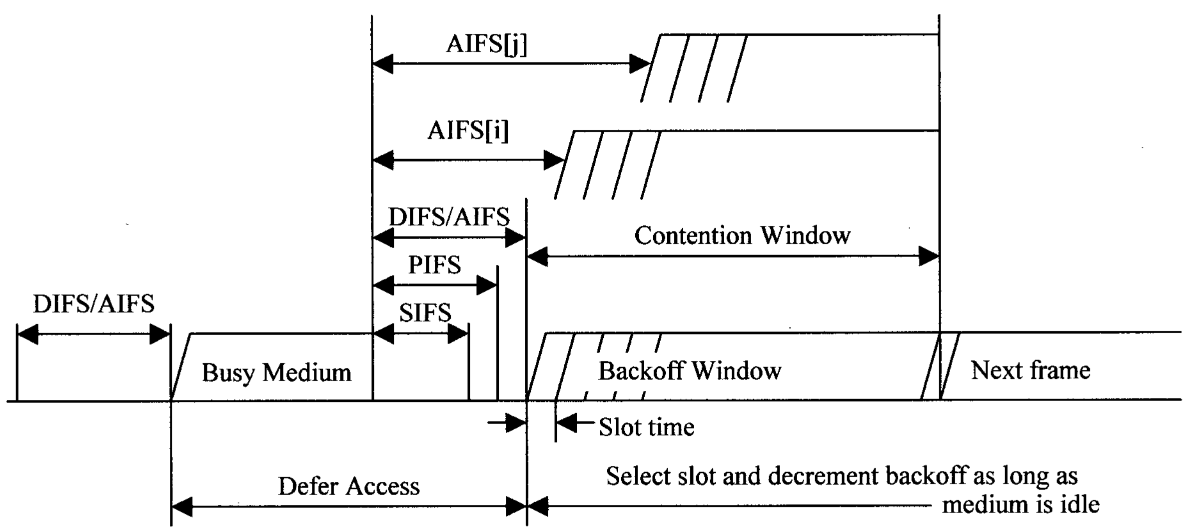


Figure 2.7. Service differentiation by different AIFS values

Within a station, up to eight transmission queues are realized as virtual stations. See Figure 2.8. Should the backoff counters of two or more parallel TCs in a single station count down to zero at the same time, a scheduler, which resides in the station, resolves the virtual collision by allowing the highest priority TC among the virtually collided TCs to transmit its MPDU [19]. The other virtually collided TCs execute the retransmission mechanism independently, as if a collision had occurred.

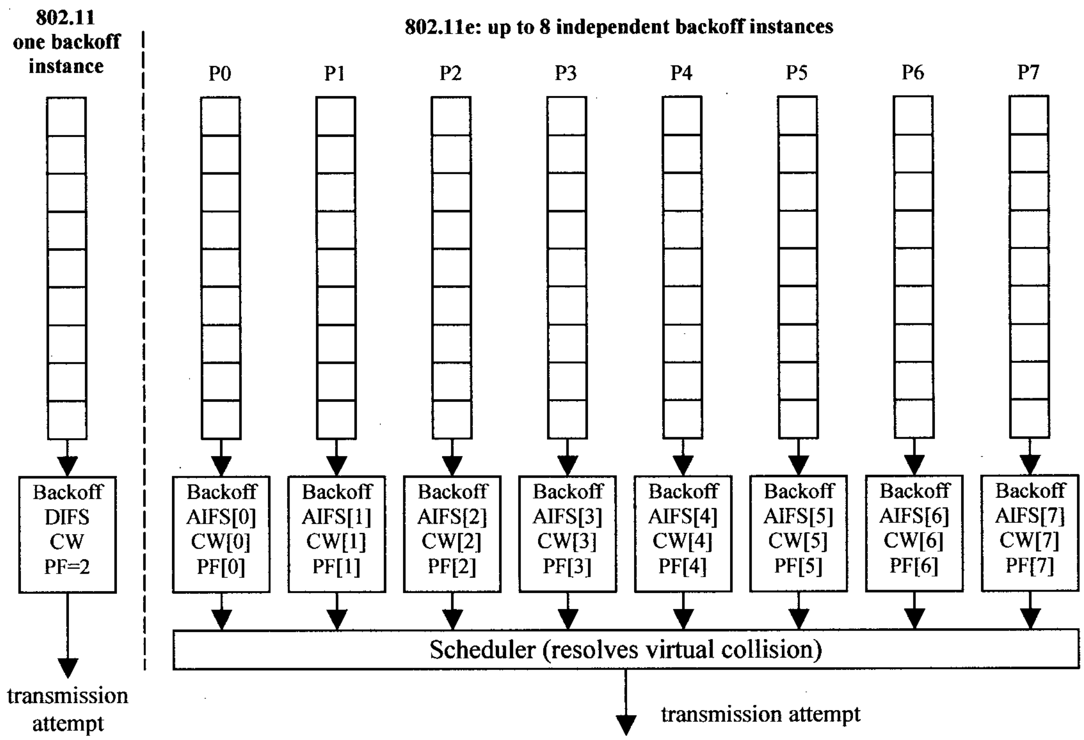


Figure 2.8. Transmission architecture of EDCF vs. DCF

Since 802.11e is a draft standard presently under review, many issues are still unsolved and are expected to change [8]. We assume that EDCF described here will not undergo any major modifications. The PF parameter mentioned in this section was proposed in a previous version of the drafts but has been removed from the current draft [20]. In the current version of EDCF, after each unsuccessful transmission, CW[TC] is doubled while remaining less than CW_{max}[TC], i.e. PF[TC] = 2.0 which is equivalent to the conventional BEB algorithm.

Chapter 3 Age-Dependent Backoff Algorithm for 802.11e EDCF

The previous chapter has reviewed the 802.11 standard and has introduced the new 802.11e EDCF protocol, proposed by the 802.11 Task Group E, to provide service differentiation at the MAC layer of 802.11 wireless LANs. The retransmission mechanism of the EDCF protocol causes large delay and jitter for real-time traffic. We now propose a retransmission algorithm called Age-Dependent Backoff (ADB) to alleviate packet delay, jitter and drop rate for real-time traffic. The proposed ADB algorithm for the new 802.11e EDCF is presented in this chapter. The backoff behavior of 802.11e EDCF is described in Section 3.1. Details of ADB are provided in Section 3.2. Finally, Section 3.3 discusses the complexity of implementing the ADB algorithm.

3.1 Backoff Behavior of 802.11e EDCF

In legacy 802.11 DCF, the BEB algorithm doubles CW with every transmission retry to reduce the collision probability in the next retransmission by providing greater transmission spacing among stations with pending MPDUs. CW becomes extremely large after successive retransmissions, which cause long delays and large jitter. To reduce delay and jitter, a smaller CW should be employed in the next retransmission thereby providing a better chance for retransmitted packets to access the medium. The wait for new arrivals would increase [21].

In a previous version of the 802.11e draft, EDCF utilizes a multiplier, PF, to govern the adjustment of CW after an unsuccessful transmission. PF should be less than 1 for time-sensitive applications. However, collisions are a result of congestion and a wider CW is desirable to alleviate congestion. Reducing the CW size for every retransmission causes heavy congestion leading to more collisions. Values of PF between one and two would be preferable. Use of different PFs for different TCs contributes to service differentiation.

The current version of EDCF employs the BEB algorithm to resolve collisions. Like DCF, BEB causes large delay and jitter values that are problematic for real-time traffic with time-bounded requirements. The situation worsens when channel conditions are bad or when the traffic load becomes heavy.

3.2 Age-Dependent Backoff Algorithm

To alleviate the problem of the retransmission mechanism employed in 802.11e EDCF protocol, we now present the proposed Age-Dependent Backoff (ADB) algorithm for high priority real-time traffic. The idea of ADB is to dynamically adjust PF based on two factors, namely the age of a real-time packet in the transmission queue and the lifetime of the real-time packet. The relationship between the new CW, *newCW*[*TC*], and the old CW, *oldCW*[*TC*], after a collision is shown in (2).

$$newCW[TC] = ((oldCW[TC]+1)\times PF[TC])-1 \tag{2}$$

where *PF*[*TC*] is given in (3).

$$PF[TC] = \frac{2}{LT[TC]}Age+2 \tag{3}$$

Age is the packet's age in the transmission queue and *LT*[*TC*] is the lifetime of the packet. Figure 3.1 shows the value of *PF*[*TC*] as the age of a real-time frame is varied. The *newCW*[*TC*] never exceeds the parameter CW_{max}[TC] but can be less than CW_{min}[TC] to provide differentiation between retransmitted packets and new arrivals.

Real-time packets are obsolete if they are not received by recipients within their lifetime. Packets with queuing delay longer than the lifetime will eventually be discarded by their applications and should not contend for the medium. Therefore, packets with $Age > LT$ are discarded before attempting transmission to save bandwidth and to prevent causing additional delay to other packets.

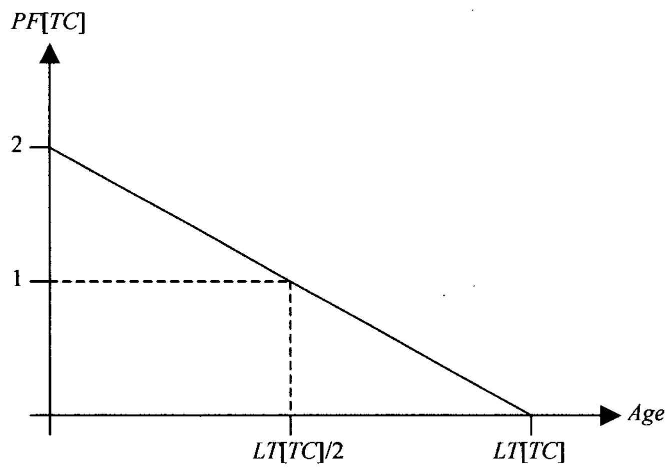


Figure 3.1. *PF*[*TC*] vs. *Age*

It can be seen in (2) and (3) that the new CW is expanded by the factor $1 \leq PF \leq 2$ in the first half of a packet's lifetime and compressed by the factor $0 \leq PF \leq 1$ in the second half. During the first half of the lifetime, ADB allows the backoff to increase gradually while avoiding high collision probability, but at the same time precluding a huge increase of delay and jitter. During the second half of the packet's lifetime, the backoff decreases slowly from the expanded CW to raise transmission probability, thereby preventing packets from being dropped.

It should be emphasized here that the ADB algorithm is used together with the BEB algorithm in 802.11 wireless LANs. The ADB algorithm is for the high priority real-time traffic with time-bounded requirements, while the BEB algorithm is for the best-effort traffic with tolerance of large delay and jitter. ADB provides higher access priority than BEB, since ADB expands CW by a factor between 1 and 2 or reduces the CW by a factor between 0 and 1, whereas BEB always expands CW by a factor of 2. ADB contributes to service differentiation by allowing a real-time frame with an age close to its lifetime to have higher priority for channel access than does a new arrival real-time frame.

3.3 Implementation Complexity

The ADB retransmission mechanism is easy to implement. It requires minor modifications in the computation of CW thereby minimizing the migration effort from the new 802.11e EDCF mode. Our ADB retransmission strategy provides backward compatibility to the 802.11 DCF protocol.

3.3.1 Implementation of the ADB Algorithm

Implementing the ADB algorithm is relatively simple. To keep track of the ages of the frames in each transmission queue, every pending frame in the queue requires a time stamp to record its arrival time from the upper layer. A temporary timestamp field can be appended to the 802.11 MAC frame to hold the arrival time. The timestamp field should be removed just before the actual transmission to keep the protocol backwards compatible with the 802.11 standard. The variable *Age* in (2) is calculated by the current time of the system minus the time stamp of the frame at the head of the queue. If $Age > LT[TC]$, the corresponding frame will be dropped and the next frame in the queue will be evaluated. The process continues until a frame with $Age < LT[TC]$ is encountered or until there are no frames in the queue. If the queue holds no frames, the corresponding queue will not contend for the access of the channel.

The *newCW*[*TC*] is determined using (2) and (3). The smaller value of *newCW*[*TC*] and CW_{max}[TC] is chosen to be the CW size. After the new CW size has been determined, the value of *oldCW*[*TC*] is updated and will be used for the calculation of the next CW size.

The flow chart shown in Figure 3.2 illustrates the implementation of the ADB algorithm.

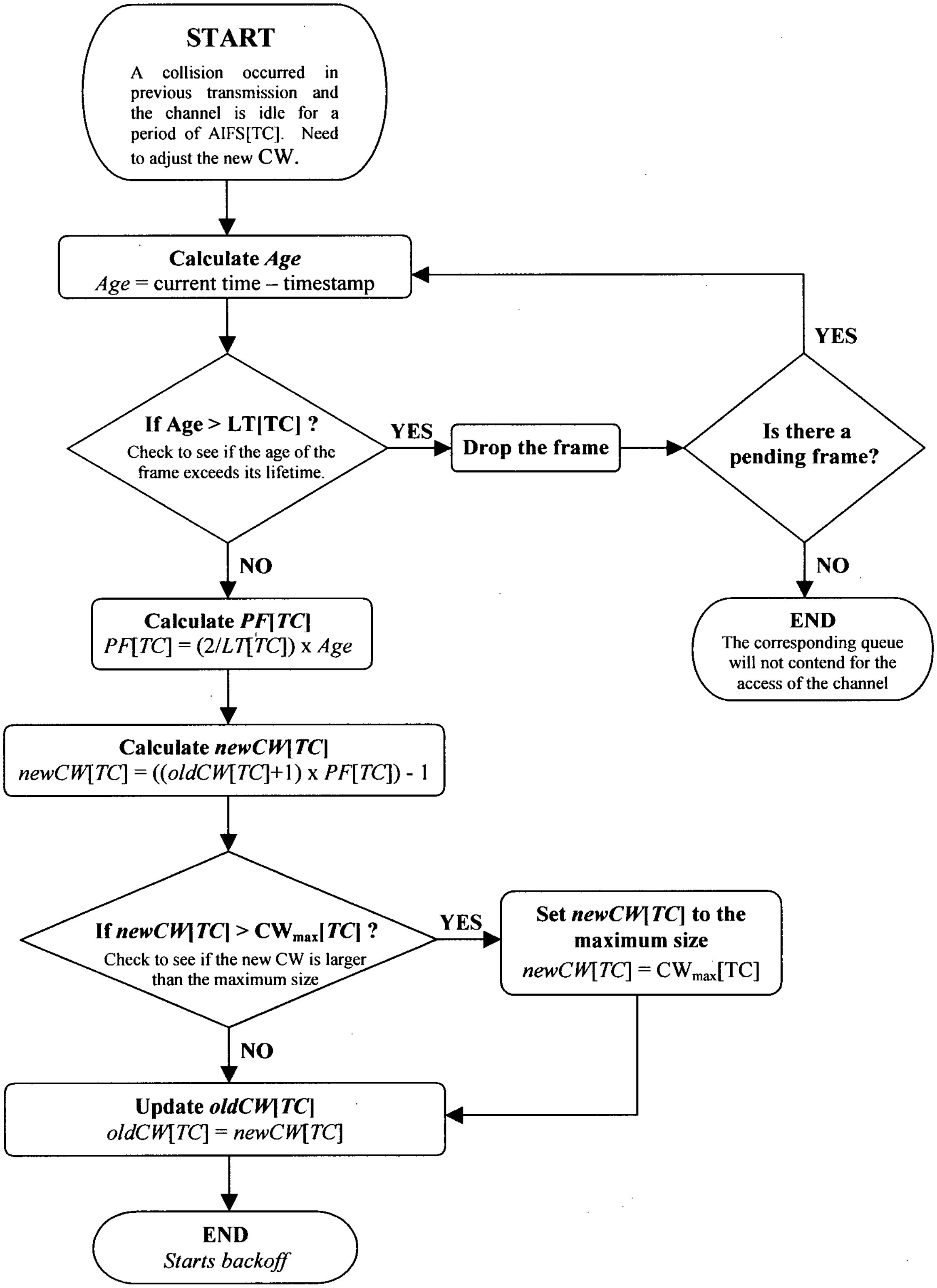


Figure 3.2. Flow chart for the implementation of the ADB retransmission algorithm

3.3.2 Discussion of the Implementation Complexity

The implementation of the ADB algorithm described in the previous section is practical with current software and hardware technologies and requires very few additional resources. Most of the modern microprocessors are equipped with real-time clocks offering timer functions; these can be used to complete the timestamp field and to provide system current time. The calculation of *Age* requires one subtraction. One comparison is needed to check if $Age > LT[TC]$ and one register is required to buffer the parameter $LT[TC]$ for each transmission queue. The calculation of $PF[TC]$ in (3) requires one division, one multiplication, and one addition and the calculation of $newCW[TC]$ in (2) introduces one addition, one subtraction, and one multiplication.

The additional resources mentioned above are necessary to implement the ADB algorithm. The addition, subtraction, multiplication, division, and comparison operations can be done using digital hardware circuits or software instruction codes. The ABD algorithm can be implemented with minor modification of the existing hardware or the embedded software of the wireless cards.

Chapter 4 Design of Simulation Models

The discussion in the last chapter has presented the problem of the retransmission mechanism employed in the new 802.11e EDCF protocol and has introduced our proposed ADB algorithm for real-time traffic in 802.11e EDCF. To evaluate the performance of ADB against BEB, several verification methods can be used. One method is to implement the ADB algorithm in wireless cards and to then measure the delay, jitter and drop rate of real-time packets from an actual network. This is definitely a high-cost and time-consuming approach. An alternative efficient method is to use computer simulation. We follow this widely used method, based on OPNET simulations, to evaluate the performance of ADB against BEB.

In this chapter, we review the OPNET simulation tools in section 4.1. Section 4.2 describes the simulation models for the 802.11 standard. Section 4.3 discusses the simulation models for the upper layer protocols. Section 4.4 presents the voice, the video, and the data traffic source models used in our work.

4.1 Overview of OPNET Simulation Tools

Optimized Network Engineering Tools (OPNET), licensed by Mil3, Inc., is a piece of engineering software capable of simulating large communication networks with detailed protocol modeling. It is widely accepted by researchers to model and simulate complex network scenarios, communication protocols, and traffic models. While OPNET has many pre-defined models, it allows users to build new models by using finite state machines together with the Proto-C language, a combination of general C/C++ facilities and an extensive library of built-in and high-level subroutines known as Kernel Procedures.

OPNET allows users to define network topologies, nodes, and links that describe a network. A simulation can be executed and the results are then analyzed for the network. OPNET has three main types of tools, namely the Model Development tool, the Simulation Execution tool and the Result Analysis tool. These three types of tools are used together to model, simulate and analyze the network.

OPNET defines a network model using a hierarchical structure shown in Figure 4.1. The highest level is the network domain, which is constructed from the node domain and the link domain. The node domain is specified in terms of the process domain.

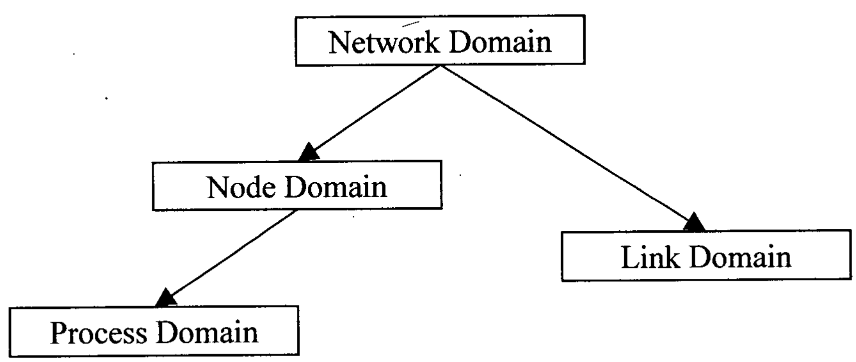


Figure 4.1. Hierarchical structure for OPNET models

The Network Domain defines a network topology described in terms of subnetworks, nodes, links, and geographical context. It consists of the nodes and the links which can be deployed within the geographical context. OPNET provides fixed nodes as well as point-to-point and bus links. In addition, it offers mobile and satellite nodes, and wireless links.

The Node Domain provides for the modeling of communication devices that can be deployed and interconnected at the Network Domain. The device models are called nodes and may correspond to various types of computing and communicating equipment in the real world such as routers, switches, bridges, workstations, servers, etc.

The Process Domain allows users to create processes using finite state machines and the Proto-C language. These processes are used to express the behavior of protocols, algorithms, and applications.

OPNET comes with built-in model libraries for some common protocols and enables users to focus on the modeling of user-defined protocols. The wireless LAN simulation model is shipped as part of the standard OPNET model and is based on the IEEE 802.11 standard. Specific parts of the wireless LAN model have been simplified or omitted in view of the fact that it is intended primarily for the performance estimation of the 802.11 DCF protocol. We implement the new 802.11e EDCF protocol and the proposed ADB algorithm by modifying the existing wireless LAN model. The detail description of the modification will be presented in the next section.

4.2 802.11 Model Design

Our 802.11 LAN model is implemented in OPNET by modifying the existing wireless LAN model to support the new 802.11e EDCF protocol and our proposed ADB retransmission algorithm. In this section, we describe the 802.11 model from the bottom to the top, i.e. from the physical layer to the MAC layer.

4.2.1 802.11 Physical Layer Model

The primary function of the 802.11 physical layer is to transmit a sequence of bits over the wireless medium. The IEEE 802.11 High Rate Direct Sequence Spread Sprectrum (HR-DSSS) physical layer, commonly referred to as 802.11b, is a rate extension to the 802.11 DSSS standard. It operates in the 2.4GHz ISM band and includes complementary code keying (CCK) to achieve additional data rates of 5.5 and 11 Mbps. 802.11b is the most common wireless LAN implementation today and is interoperable with 802.11 DSSS implementations. We design our 802.11 physical layer model to be similar to the 802.11b.

The existing wireless LAN model provides three choices for the physical layer configuration in the IEEE 802.11 standard: Frequency Hopping, Direct Sequence, and Infra Red. The wireless LAN model supports data transfer at 1, 2, 5.5 and 11 Mbps. All control packets are transmitted at a data rate of 1 Mbps as specified by the standard. These data rates are modeled as the speed of the transmitter and receiver connected to the 802.11 MAC layer. Although the wireless LAN model does not simulate the actual physical layer of the IEEE

802.11 specification, the physical layer configuration are needed by the MAC protocol to determine the parameters such as SIFS, DIFS and CW limits.

In our study, the modulation technique is relatively unimportant to the service differentiation in 802.11 wireless LANs. However, the effect of wireless channel noise must be considered due to its impact on bit and packet errors which cause retransmissions. Because of transmission impairments, such as noise and collisions, bit errors can disrupt the sequencing of frames. A station may send a data frame and never receive an ACK. Stations initiating the exchange of frames have the responsibility of error recovery. This recovery involves the retransmission of frames after SIFS sec., if no response is heard from the destination station. Therefore, a frame corrupted by noise or which has collided with another frame causes execution of the retransmission protocol.

The OPNET Wireless Module allows users to specify the quality of a wireless link based on a number of factors such as the distance between the transmitter and the receiver, the power of the transmitting signal, the transmitter and the receiver antenna gains, the background and the interference noises, etc. OPNET's simulation kernel relies on a 14-stage computational pipeline which uses these factors to emulate the characteristics of the wireless communication channel. The bit error rate can be measured statistically during the simulation.

The existing OPNET wireless LAN model uses the default wireless model settings with some minor modification to conform to the 802.11 standard. We specify the noise figure parameter, the transmitting power, and the distance between the transmitting and receiving stations to achieve a bit error rate of 10^{-9}. We set the error threshold to be zero. Hence, any errors in a packet are discarded by the MAC layer.

In order to model the 802.11b physical layer, we set the channel data rate at 11 Mbps with direct sequence spread spectrum physical characteristic. With this direct sequence physical characteristic, we specified SIFS = 10 μs, DIFS = 50 μs, CW_{min} = 31, and CW_{max} = 1023.

4.2.2 802.11 MAC Layer Model

Based on the existing wireless LAN model in OPNET, we modify the 802.11 DCF protocol at the MAC layer to develop the new 802.11e EDCF protocol with the proposed ADB algorithm. The existing wireless LAN model is intended primarily for DCF-based MAC performance estimation. Many important features of the 802.11 MAC are implemented. The following is a summary of the important features included in the wireless LAN model [22].

- Access Mechanism: Carrier sense multiple access and collision avoidance (CSMA/CA) DCF access scheme as defined in the standard.
- Frame exchange sequence: Data and ACK frame exchange to ensure the reliability of data transfer. Optional RTS/CTS frame exchange is available for media reservation.

- Backoff and Deference: The BEB algorithm and the interframe spacing: SIFS, DIFS, and EIFS for DCF implementation. The values for the interframe spacing are selected based on the selection of physical characteristics.
- Channel Data Rate: Date rates of 1, 2, 5.5 and 11 Mbps
- Recovery mechanism: Short and Long retry counters as defined in the standard.
- Fragmentation and Reassembly: Optional data frame fragmentation based on the size of the data packet received from the higher layer. The fragments are reassembled at the destination station.
- Duplicate Packet Detection: Any duplicated packets are discarded by the MAC layer.
- Access Point Functionality: Only a wireless LAN router can be configured as an access point in an infrastructure BSS network. In order to form an ESS such that stations within various BSSs can communicate with each other, an IP network needs to be configured.
- Buffer Size: Data arrived from a higher layer to the MAC layer is stored in a buffer. The buffer size is limited to the maximum value set by the user. Higher layer packets are dropped, once the maximum buffer size is reached.

Some additional features are necessarily added to make that the existing wireless LAN model supports EDCF and ADB. The wireless LAN model has only one transmission queue buffering frames waiting for transmission. All packets coming from the upper layer are forwarded directly to that transmission queue. The packets placed in the queue will be transmitted only if the backoff counter reaches zero and the wireless channel has been idle for

DIFS sec. The model has only one backoff counter and one interframe spacing period, DIFS, for all data frame transmissions.

Multiple transmission queues, backoff counters, CW limits and interframe spacing periods are necessary for implementing the EDCF protocol. Packets arriving from the upper layer are forwarded to the corresponding queues based on the traffic types. To simulate a wireless LAN carrying voice, video, and data traffic, we implement three transmission queues for three different traffic types. In order to simplify the implementation, each station is allowed to generate and receive only one type of bi-directional traffic and therefore, using destination MAC addresses is enough to determine the traffic types. A scheduler, which resides in an AP, is implemented to resolve the virtual collision when two or more TCs in the access point (AP) count down to zero at the same time.

To implement the ADB algorithm, each pending frame is given an arrival timestamp. All packets from the upper layer are sent to the corresponding transmission queues with their timestamps which are used to calculate *Age*, as explained in section 3.3.1. The implementation of ADB in the simulation model is as presented in section 3.3.1.

The finite state machine for modeling the 802.11 MAC layer with EDCF and ADB is shown in Figure 4.2. A transition between the BKOFF_NEED state and the IDLE state is added to the finite state machine. This transition is used when the ages of all the packets in the transmission queues are greater than their lifetimes. The expired real-time packets are then dropped from the queues since packets with ages greater than their lifetimes will eventually be discarded by the applications. The station should not contend for the access of the channel and the finite state machine should go back to the IDLE state.

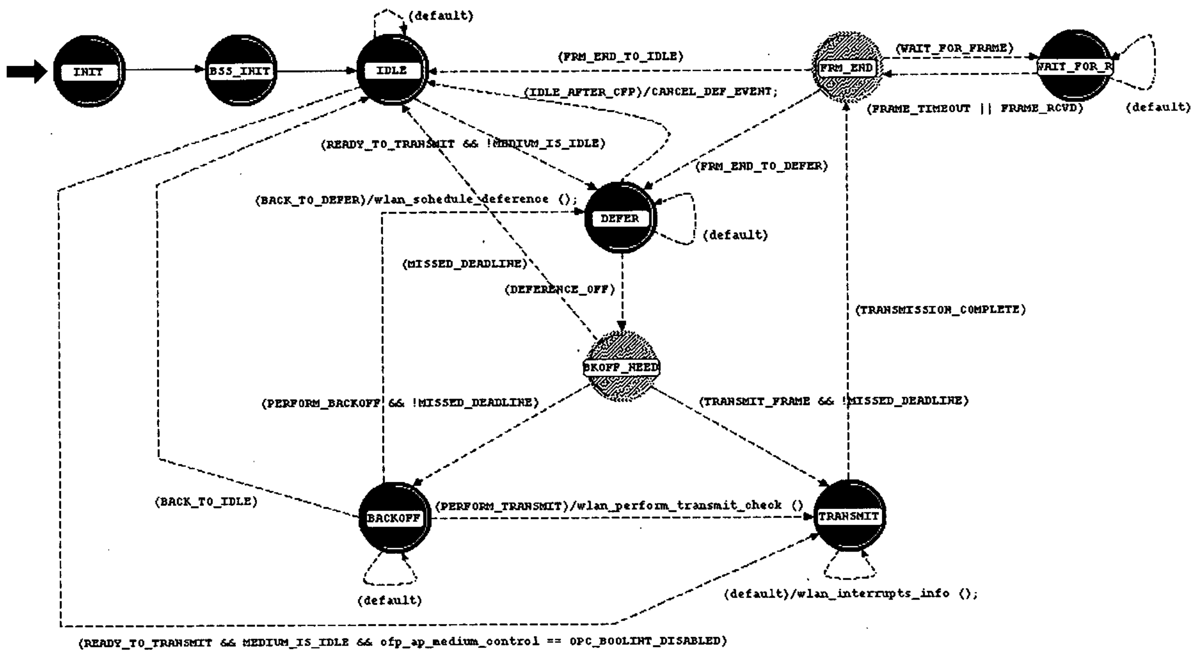


Figure 4.2. Finite state machine for 802.11 MAC layer model with EDCF and ADB

The INIT and the BSS_INIT states are used to initialize all the variables, statistics, memory allocations and other entities. When a packet from the upper layer arrives and the channel is idle, the state machine will go directly to the TRANSMIT state to transmit the packet. If a packet arrives but the channel is busy, the state machine will move to the DEFER state to defer for a period of DIFS/AIFS after the channel becomes idle.

After the deference described above, the state machine will jump to the BKOFF_NEED state. A CW size is then calculated using the BEB or the ADB algorithm.

Once the CW size is determined, the state machine will go to the BACKOFF state to udergo the backoff operation. When a backoff counter reaches zero, the state machine will move to the TRANSMIT state to transmit the packet. After finishing the transmission, the transmitting station will wait for an ACK frame from the receiving station to confirm the success of the frame reception. The state machine will jump to the FRM_END state and then to the WAIT_FOR_R state to wait for the ACK frame. If the ACK frame arrives successfully, it will go either to the IDLE state through the FRM_END state if there are no pending frames in the transmission queues or to the DEFER state through the FRM_END state to start the next transmission if the transmission queues are not empty.

Table 4.1 shows the simulation parameters for the MAC layer protocol. In our simulation scenarios, detailed in Chapter 5, we assume that the simulation network consists of one BSS with no hidden stations, i.e. stations can hear each other in the BSS. The RTS/CTS optional is disabled since it has been indicated that the RTS/CTS mechanism produces very limited advantages with no hidden stations [23]. Moreover, most of the default settings of commercially available wireless cards are without the RTS/CTS mechanism.

The short retry limit gives the maximum number of transmission attempts for frames whose size are less than or equal to RTS threshold, while the long retry limit specifies the maximum number of transmission attempts for frames whose size exceeds the RTS threshold. When the RTS/CTS is disabled, the short retry limit is used for all transmissions. Frames that could not be transmitted successfully after reaching the short retry limit are discarded by the MAC layer.

Parameters	Values
Slot time	20 μs
SIFS	10 μs
DIFS	10 μs + 2 × 20 μs = 50 μs
AIFS[1] (voice)	10 μs + 2 × 20 μs = 50 μs
AIFS[2] (video)	10 μs + 3 × 20 μs = 70 μs
AIFS[3] (data)	10 μs + 5 × 20 μs = 110 μs
[CWmin, CWmax] (DCF)	[31, 1023]
[CWmin[1], CWmax[1]] (voice)	[7, 31]
[CWmin[2], CWmax[2]] (video)	[15, 63]
[CWmin[3], CWmax[3]] (data)	[15, 255]
Short Retry Limit	255
Long Retry Limit	255

Table 4.1. Parameters for the MAC layer

The ADB algorithm is used for the retransmission mechanism for real-time traffic with time-bounded requirements. The lifetime of a real-time packet must be specified to allow ADB to calculate the new CW size after every unsuccessful transmission. We now discuss the lifetimes of voice and video real-time packets.

The main performance parameters for voice traffic are packet delay, jitter, and loss. The end-to-end delay for IP telephony ranges from 300 to 1000 ms. User tolerance of delay varies significantly. Demanding users require delay of 200 ms or less, while more patient users will accept a delay of 300 to 800 ms [24]. Since the wireless LAN represents only a single hop of an end-to-end connection, we consider 25 ms as the maximum acceptable value for the voice packet transfer delay over the wireless LAN and let *LT* for voice packets be 25 ms.

We assume that 75 ms is the maximum allowable video packet delay at a single wireless hop; thus $LT = 75$ ms for video packets.

4.3 Simulation Models for Upper Layer Protocols

The existing OPNET wireless model includes node models which correspond to various real world wireless LAN equipment types such as wireless stations and wireless routers. By replacing the original MAC layer with our modified MAC layer presented in the previous section, we developed wireless station and router node models supporting the new 802.11 EDCF protocol and the proposed ADB algorithm. The upper layer protocols such as the Internet Protocol (IP), the Transport Layer Protocol (TCP), the User Datagram Protocol (UDP) and the application layers are unmodified. In this section, we introduce the basic functions and parameters chosen for these upper layer protocols from bottom to top, i.e. from the network layer to the application layer.

4.3.1 Network Layer Model: Internet Protocol

IP is a connectionless network level protocol interconnecting multiple networks, possibly of different types. In the Internet Protocol Suite, IP provides best-effort delivery services to TCP and UDP and is situated beneath the transport layer of the OSI seven-layer model [13]. In turn, IP relies on the services of the data link layer, such as the 802.3 Ethernet and the 802.11 wireless LAN, to relay packets to other IP modules.

The packets that are created and forwarded by IP modules are called datagrams. Because IP connects different types of networks that may support different maximum transfer units (MTUs), fragmentation is performed by the IP modules whenever they attempt to forward a datagram over an interface whose datagram length exceeds its MTU. Reassembly functions are only performed on datagrams when they reach their final destination.

The MTU of 802.11 wireless LANs is 2304 bytes. We configure the IP modules to perform fragmentation when forwarding a datagram with size larger than 2304 bytes over an 802.11 interface. Since the size of the data packets from the IP layer cannot be larger than the 802.11's MTU, the fragmentation and reassembly option at the 802.11 MAC layer can be disabled.

4.3.2 Transport Layer Models: TCP and UDP

Two transport layer protocols, TCP and UDP, are build on the best-effort service, provided by IP, to support a wide range of applications. In this section we discuss the properties of TCP and UDP and the parameters chosen in our simulations.

UDP is a very simple but unreliable, connectionless transport layer protocol that provides only two additional services beyond IP: demultiplexing and error checking on data. IP delivers packets to a host but does not know how to deliver them to the specific application in the host. UDP adds a mechanism that distinguishes among multiple applications within the host. IP only checks the integrity of its header. UDP can optionally check the integrity of the entire UDP datagram. If a datagram is found to be corrupted, it will be simply discarded and the source UDP entity will not be notified.

TCP is a transport layer protocol, providing a connection-oriented, reliable, in-sequence, byte-stream service. TCP offers flow control that allows receivers to control the rate at which the sender transmits information so that the receiver buffers do not overflow. TCP also provides congestion control which ensures that the sender does not waste resources by sending more traffic than the network can successfully carry to the receiver. Similar to UDP, TCP supports multiple applications in a host and error checking on TCP segments.

TCP and UDP both have the ability to establish a host-to-host communication channel for delivering packets between applications running in two different stations. The main difference is that TCP provides reliability, and flow and congestion control services, while UDP trades off those services to improve performance. Application protocols can choose TCP or UDP at the transport layer. The File Transfer Protocol (FTP) uses TCP to ensure that an exact copy of a file is delivered to the recipient. Real-time applications such as voice and video conferencing use UDP because they can tolerate some errors or data loss.

OPNET's TCP module is based on the TCP specified in RFC 793 and RFC 1122. The model incorporates some important features such as end-to-end reliability based on acknowledgment, retransmissions triggered by exponentially backed-off timers, flow control based on the availability of remote buffering resources, reordering of data that arrives out of sequence, connection establishment and closing through three-way handshake protocols, and "Slow-start" congestion avoidance and control [25]. The model allows users to control the TCP behavior by selecting the TCP parameters. We choose the default setting of the TCP parameters. Some of the important default parameter settings are shown in Table 4.2.

Maximum Segment Size	2272 bytes
Receive Buffer	8769 bytes
Maximum ACK Delay	0.200 sec
Slow-Start Initial Count	1 MSS
Fast Retransmit	Enabled
Fast Recover	Reno
Karn's Algorithm	Enabled
Initial RTO	3.0
Minimum RTO	1.0
Maximum RTO	64
RTT Gain	0.125
RTT Deviation Gain	0.25
RTT Deviation Coefficient	4.0
Timer Granularity	0.5 sec
Persistence Timeout	1.0 sec

Table 4.2. Default settings of the TCP parameters

The maximum segment size is set to 2272 bytes so that the underlying 802.11 network can carry traffic without any fragmentation. With the fast retransmit algorithm enabled, after receiving three duplicate ACKs, the TCP sender infers that a segment is lost and re-transmits the missing segment without waiting for its retransmit timer to expire [26]. This allows TCP to detect and retransmit a lost segment more quickly than for time-out retransmission. TCP Reno employs Karn's algorithm [27], which requires round-trip estimates to be updated only for the ACKs of those segments that have been transmitted only once. TCP uses frequent measurements of the round trip timer (RTT) to dynamically adjust the retransmission timeout (RTO). When the RTO expires before an ACK is received, retransmission occurs starting with the first byte of unacknowledged data. The RTT gain and RTT deviation gain are constants between 0 and 1 and control how fast the RTO adapts to changes in network traffic. The RTT gain and the RTT deviation gain are recommended to be 0.125 and 0.25 respectively

[28]. For further information regarding the default setting, please refer to the TCP Model Description of OPNET user's manual [25].

In OPNET's UDP model, there are no parameters to specify the behavior of UDP since UDP itself is a very simple protocol. If the application layer sends a packet with size larger than the maximum payload size of a UDP datagram, 65535 bytes - 8 bytes = 65527 bytes, the packet will be dropped.

4.4 Traffic Source Models

We use OPNET's standard network application model for the traffic source models. The standard network application model is a simple, general model of client and server or a peer-to-peer network application. Its behavior can be modified through parameter settings to make it emulate a wide variety of network applications. It does not, however, model the behavior of any particular application in detail.

The parameters of the standard network application model are selected by carefully emulating the environment in which 802.11 MAC would most probably be applied. With this in mind, we choose numerical parameters in such a way that the traffic sources reflect as closely as possible the network load conditions of an actual real-life situation.

The simulations consider three types of traffic sources: voice, video conferencing and FTP traffic. Each wireless station runs only one session and all sessions are bi-directional, i.e. each station is the source of an uplink flow and the sink of a downlink flow for the session it runs. Voice and video conferencing traffic have higher priority than FTP traffic. Section 4.4.1 and section 4.4.2 describe the voice and video conferencing models, respectively. The best-effort FTP file transfer model is presented in section 4.4.3.

4.4.1 Voice Model

Voice traffic is known to have a two-state ON/OFF behavior, where voice users are either in talkspurt or silence as shown in Figure 4.3. For efficient usage of wireless bandwidth, silence suppression is employed. Hence, voice packets are generated in talkspurt (ON), while no packets are generated during silence (OFF). Both the duration of talkspurt and of silence follow the exponential duration with the mean duration equal to 1 and 1.35 seconds respectively [29]; thus a voice source is in talkspurt mode approximately 42.6% of the time. Each voice station runs only one bi-direction voice session over UDP/IP.

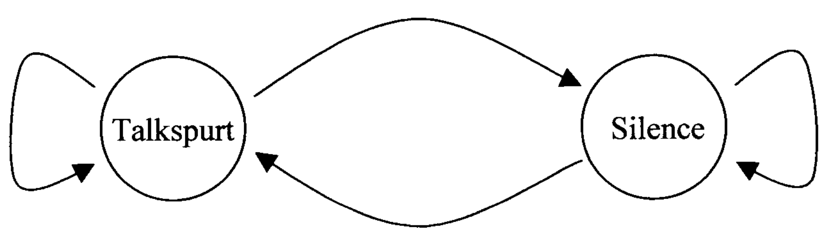


Figure 4.3. Two-state model of a voice user

The G.729 speech codec is selected to model good-quality voice calls, with 8-kbps coding rate, 10-byte packets generated in every 10 ms during talkspurt, 10 ms processing delay, and 5 ms lookahead delay. The processing delay is the delay required to run the encoding algorithm on a single frame and the lookahead delay is the amount of time needed to delay the speech for look ahead calculations for the encoder frame. The encoder buffers data for this extra time to enable prediction, while compressing data. The one-way latency of the encoder is the sum of the frame size, processing delay, and lookahead delay and therefore, the total codec delay for G.729 is 25 ms [30].

4.4.2 Video Conferencing Model

The video conferencing application in OPNET's standard network application model is used to generate streaming video frames across the wireless LAN. It allows users to control the properties of streaming video through the application parameters. Table 4.3 shows the parameters selected for modeling a low-quality video conferencing application characterized by a relatively low bit rate of 128 kbps for both the uplink and downlink

Parameters	Description	Values
Incoming Stream Interarrival Time	Time between frames generated within a video conferencing session from the destination	20 frames/sec
Outgoing Stream Interarrival Time	Time between frames generated within a video conferencing session from the source	20 frames/sec
Incoming Stream File Size	Average size of an incoming video frame	Exponentially distributed with a mean of 800 bytes
Outgoing Stream File Size	Average size of an outgoing video frame	Exponentially distributed with a mean of 800 bytes

Table 4.3. Parameters for video conferencing

UDP transport protocol is used for the video conferencing application. A video station runs one bi-directional video session over UDP/IP.

4.4.3 FTP File Transfer Model

An FTP application enables file transfers between a client and a server. The FTP application in OPNET's standard network application model is used to generate best-effort data traffic across the network. FTP has two basic commands for transferring a file: Get and Put. The Get command triggers the file transfer from the server and the Put command sends a file from the client to the server. FTP uses the connection-oriented TCP transport protocol. A new transport connection is opened for each file transfer. Table 4.4 shows the parameters chosen in the FTP application.

Parameters	Description	Values
Command Mix (get/total)	Ratio of the Get (download) commands to the total number of commands (sum of Get and Put commands)	50%
Inter-Request Time	Time between subsequent file requests	Exponentially distributed with a mean of 0.02048 second
File Size	Size of a file being transferred	1024 bytes

Table 4.4. Parameters for FTP file transfers

Using the above parameter values, we assume that the upload and the download streams are identical with 200 kbps of FTP traffic in each stream. The rate at which files are requested is independent of the response received; that is, the second request can initiate without the first response being received.

Chapter 5 Performance Analysis of the Proposed ADB Algorithm in an Ad-Hoc Network and a Hotspot

The previous chapter has introduced the OPNET simulation models. We will use these models to evaluate the performance of the new 802.11e EDCF protocol with our proposed ADB retransmission algorithm. The behavior of EDCF without ADB is presented for comparison. This chapter presents simulation parameters and scenarios, results from the simulations, and discussions of the results.

Two typical environments of 802.11 wireless LANs are considered, namely the ad-hoc network and the hotspot scenarios. In the ad-hoc network scenario, different services such as voice, video and data services may be active simultaneously in an ad-hoc network. The ad-hoc network is an independent BSS in a restricted space where stations can hear each other.

The hotspot scenario is conceived where voice, video, and data users are connected to wireline networks through an AP in a single BSS. A hotspot is a specific type of infrastructure mode installation in which a commercial entity provides wireless Internet access to the public. Any reference herein to "hotspot" installations would apply equally to other infrastructure mode installations.

Section 5.1 describes the ad-hoc scenario and section 5.2 presents the hotspot scenario. Both sections provide simulation results as well as discussions concerning the results. All results presented in this chapter are the average of results collected from simulating each of the scenarios for 180 seconds for each of 5 different seed numbers. Section 5.3 provides a conclusion regarding the simulation results.

5.1 Ad-Hoc Network Scenario

This section presents the ad-hoc network scenario. The overview of the ad-hoc network scenario is introduced in section 5.1.1. The simulation results are given in section 5.1.2. The discussions are provided in section 5.1.3.

5.1.1 Overview of the Ad-Hoc Network Scenario

One environment in which 802.11 wireless LANs is likely to be applied is in an ad-hoc network. Such situation is appropriate for, assemblies, outdoor activities, rescue operations, or major disaster areas, where users need to deploy networks to start communication immediately without the benefit of a fixed network infrastructure.

Figure 5.1 shows the topology of the ad-hoc network. For simplicity, we assume that there are no hidden stations and that the distance between every pair of transmitting and receiving stations is the same. The bit error rate of each wireless link is therefore assumed to be constant and equal. In the ad-hoc network, voice, video conferencing and best-effort FTP file transfer services are active simultaneously.

We conceive that the ad-hoc network is an independent BSS with 10 voice, 4 video and *n* FTP client and server stations where n is variable. Since 802.11e EDCF is designed to be backwards compatible to 802.11 DCF, we assume that half of the FTP client and server stations are using the 802.11 DCF while the other half are using the new 802.11e EDCF protocol with the BEB algorithm with PF value of 2.0. We simulate the ad-hoc network to compare, for real-time traffic, the performance of our ADB algorithm with that of the generalized BEB algorithm with PF = 1.5 and 2.0. Best-effort FTP data traffic always employs the conventional BEB algorithm where PF = 2.0.

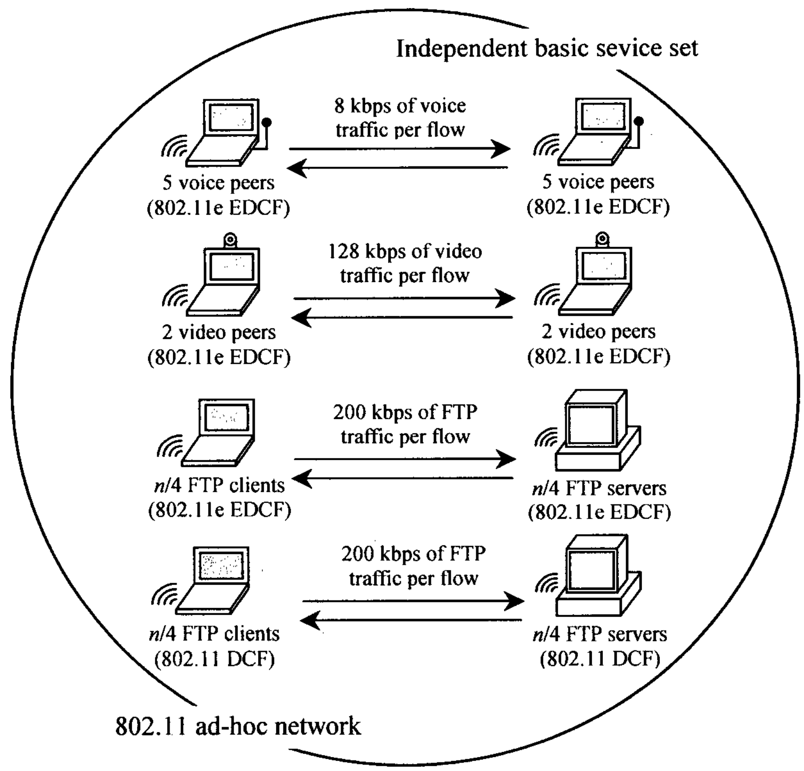


Figure 5.1. Network topology of an ad-hoc scenario

5.1.2 Simulation Results from the Ad-Hoc Network Scenario

This section presents the results of the ad-hoc network scenario. Voice packet delay, jitter, and drop rate are shown in Figures 5.2, 5.3, and 5.4 respectively. We define the jitter as the delay variance and the drop rate as the percentage of packets with delay longer than their lifetime. The lifetime of a voice packet is assumed to be 25 ms.

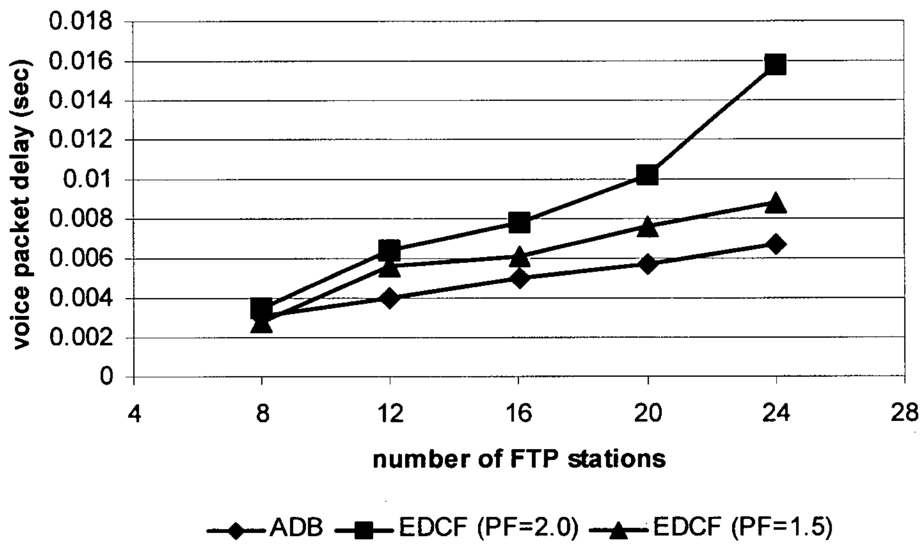


Figure 5.2. Voice packet delay in an ad-hoc network

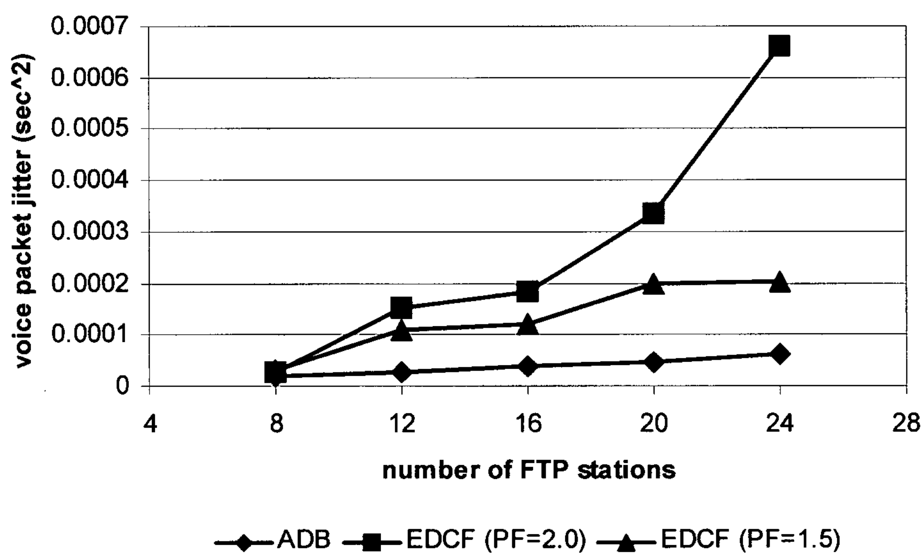


Figure 5.3. Voice packet jitter in an ad-hoc network

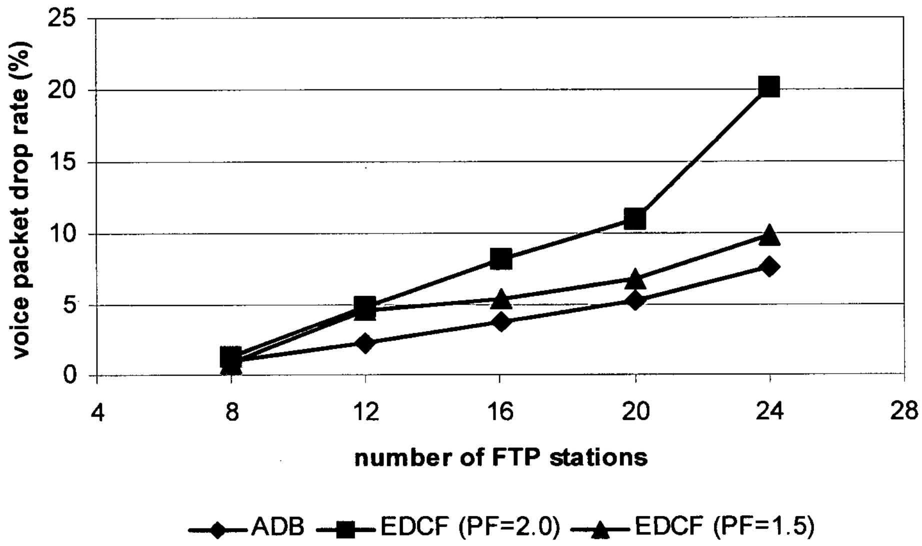


Figure 5.4. Voice packet drop rate in an ad-hoc network

Figures 5.5, 5.6, and 5.7 show delay, jitter and drop rate, respectively, for video packets. The lifetime of a video packet is assumed to be 75 ms and therefore, a video packet is dropped if its age exceeds 75 ms.

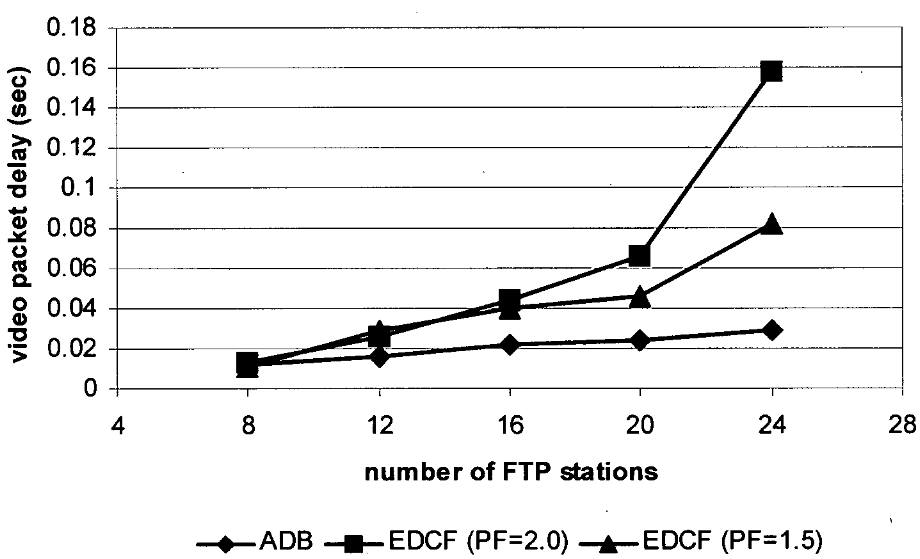


Figure 5.5. Video packet delay in an ad-hoc network

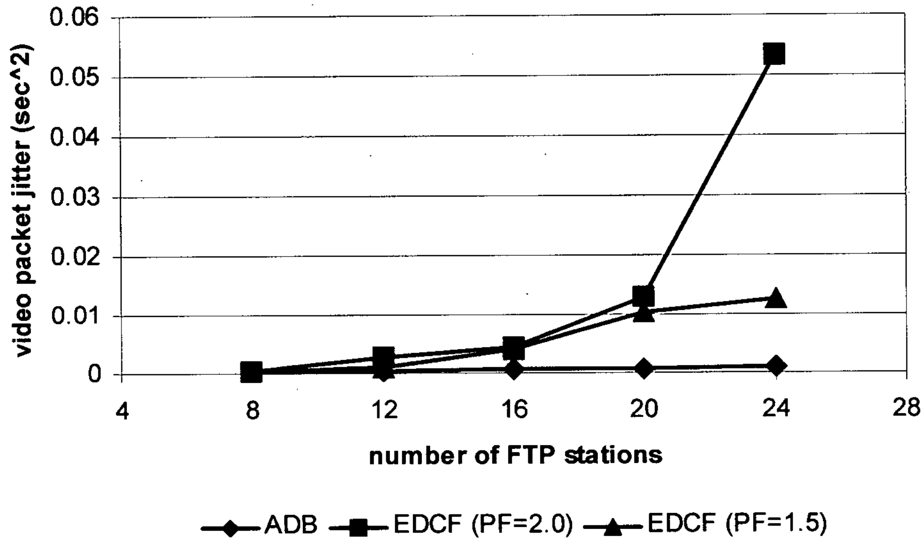


Figure 5.6. Video packet jitter in an ad-hoc network

The main performance characteristic of the best-effort FTP traffic is measured by throughput. The total FTP traffic throughput for the ad-hoc network is shown in Figure 5.8.

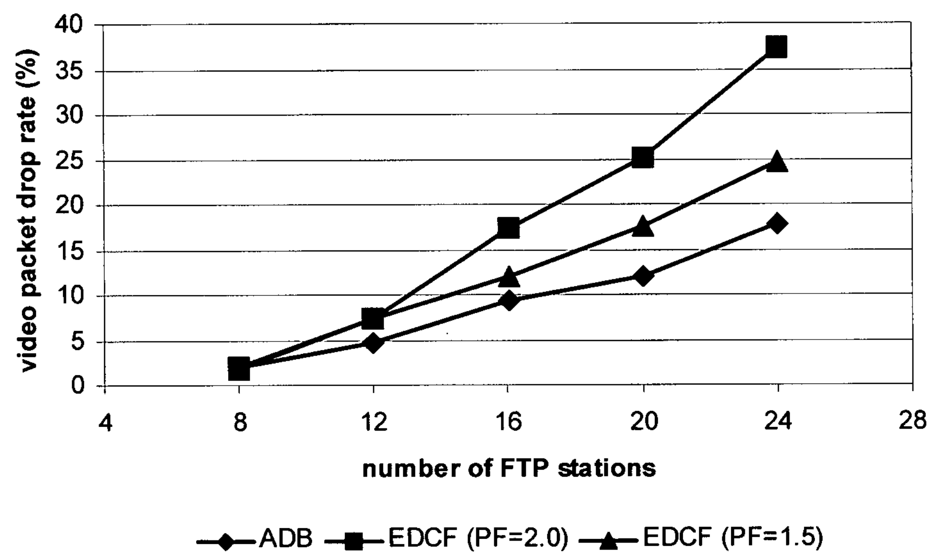


Figure 5.7. Video packet drop rate in an ad-hoc network

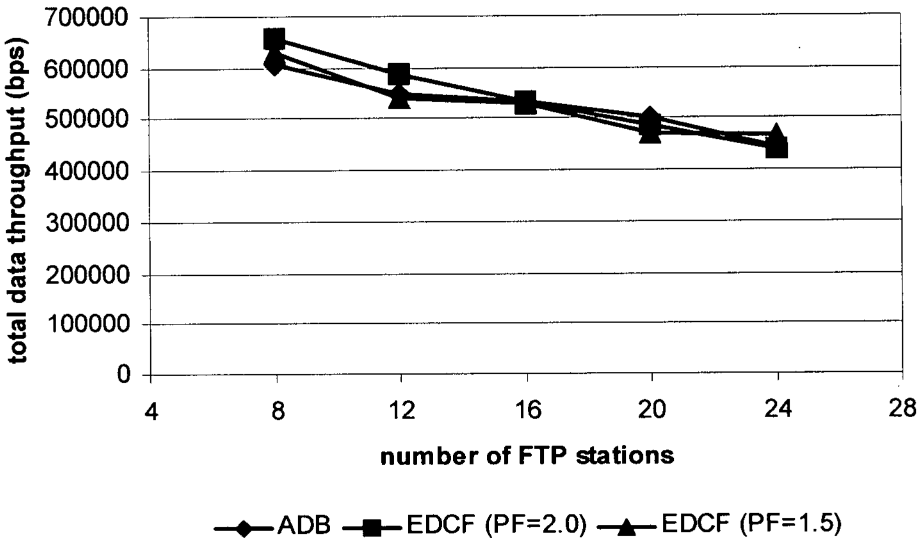


Figure 5.8. Total FTP data throughput in an ad-hoc network

5.1.3 Discussions of the Ad-Hoc Network Results

Our simulation results demonstrate that EDCF with the ADB algorithm outperforms EDCF with the generalized BEB algorithm for PF = 1.5 and 2.0 in all aspects of network performance that we have tested.

The voice packet delay, jitter and drop rate are significantly lower when using ADB as illustrated in Figures 5.2, 5.3, and 5.4. Relative to BEB, ADB also demonstrates considerable improvements in the video packet delay, jitter, and drop rate as depicted in Figures 5.5, 5.6 and 5.7.

The relative improvements in voice and video traffic become more noticeable and pronounced as the number of FTP stations increases. ADB enhances the performance of the voice and video traffic without sacrificing the throughput of the best-effort FTP traffic; as well, ADB prevents the FTP traffic from being starved as shown in Figure 5.8.

ADB dynamically adjusts the change of CW sizes based on the ages and lifetimes of voice and video packets to avoid long delay and to prevent excessive collisions under heavy traffic load. As a result, the delay, jitter and drop rate of both voice and video packets are improved considerably. These enhancements in voice and video packet performance do not adversely affect the FTP traffic throughput. Accordingly, we can conclude that ADB enables more efficient use of the wireless channel and improves the overall performance of the ad-hoc network.

5.2 Hotspot Scenario

The subject of this section is the hotspot scenario. An overview of this scenario appears in section 5.2.1. Simulation results are given in section 5.2.2, and discussions are provided in section 5.2.3.

5.2.1 Overview of the Hotspot Scenario

A hotspot is another environment where 802.11 wireless LANs would likely be deployed. A hotspot is a geographic area covered by wireless networks, with Internet access being available to those devices with wireless network cards.

Most of the hotspots today are covered by 802.11b wireless LANs to provide wireless Internet access services for public use. The network topology associated with the hotspot scenario is depicted in Figure 5.9.

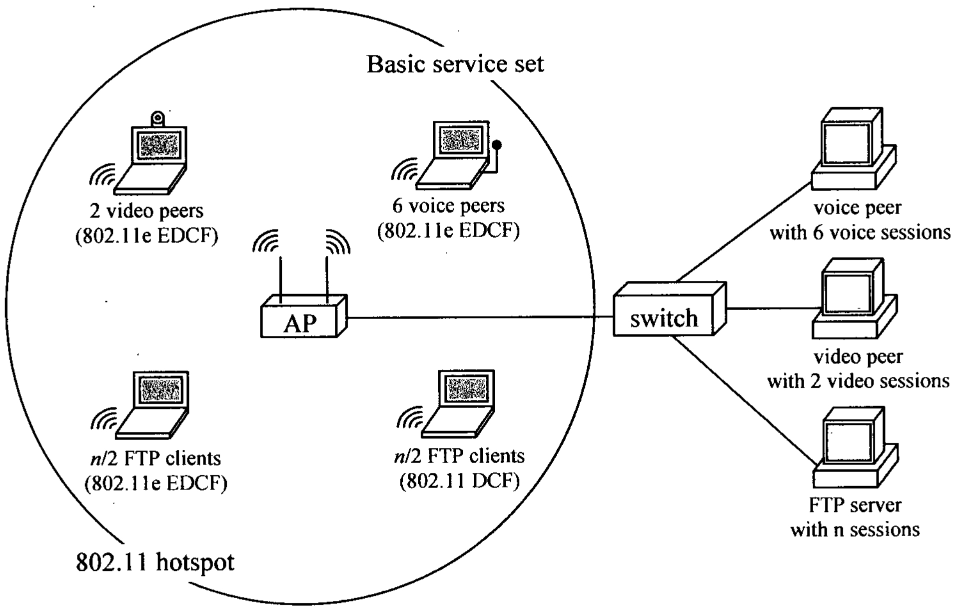


Figure 5.9. Network topology of a hotspot scenario

The hotspot scenario is developed to study the performance of the ADB algorithm in a hotspot environment. For simplicity, we assume that there is only one AP in the hotspot and that there are no hidden stations in a BSS. The distance between each wireless station and the AP is the same and constant, in which case the bit error rate of each wireless link is assumed to be identical.

Voice, video conferencing and best-effort FTP file transfer services are active simultaneously in the hotspot. Because users in the hotspot environment rarely make phone or video conferencing calls, or rarely transfer files to their neighbors, all types of traffic generated within the BSS are delivered to their peers or servers in the wireline network through the AP. As well, all types of traffic destined for wireless stations are also via the AP from the wireline network. The upstream and the downstream traffic loads for all types of services are assumed to be the same.

We assume that the hotspot consists of 6 voice, 2 video, and *n* FTP client stations. Since 802.11e EDCF is designed to be backwards compatible to 802.11 DCF, we assume that half of the FTP clients are using the 802.11 DCF while the other half are using the new 802.11e EDCF protocol with the BEB algorithm with PF value of 2.0. We simulate the hotspot scenario to compare the ADB algorithm for real-time traffic against the generalized BEB algorithm with PF values of 1.5 and 2.0 for real-time traffic. Again, best-effort FTP data traffic employs the conventional BEB algorithm with PF = 2.0.

5.2.2 Simulation Results from the Hotspot Scenario

This section presents results from simulations of the hotspot scenario. Voice packet delay is shown in Figure 5.10. Uplink and downlink voice packet jitter is shown in Figure 5.11a and 5.11b respectively, where jitter is defined as the delay variance. The uplink and downlink voice drop rate is shown in Figure 5.12a and 5.12b, respectively. The lifetime of a voice packet is assumed to be 25 ms.

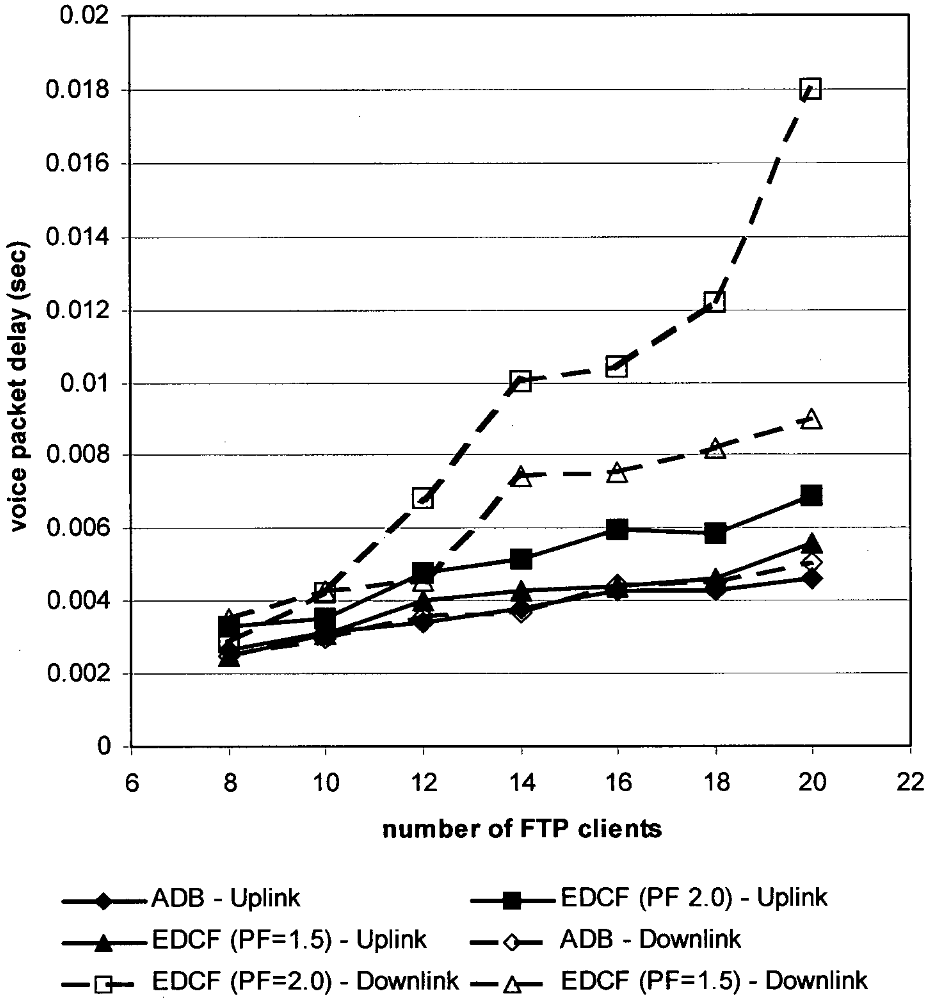
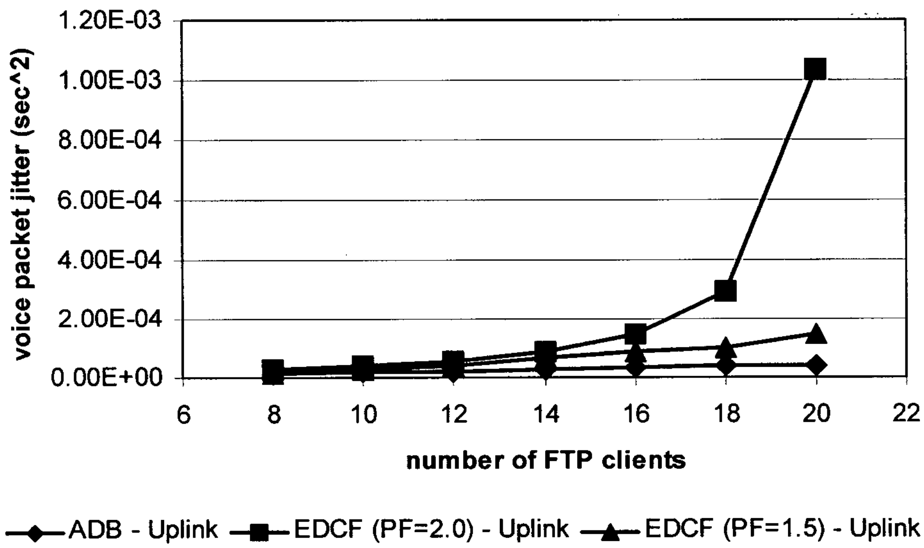
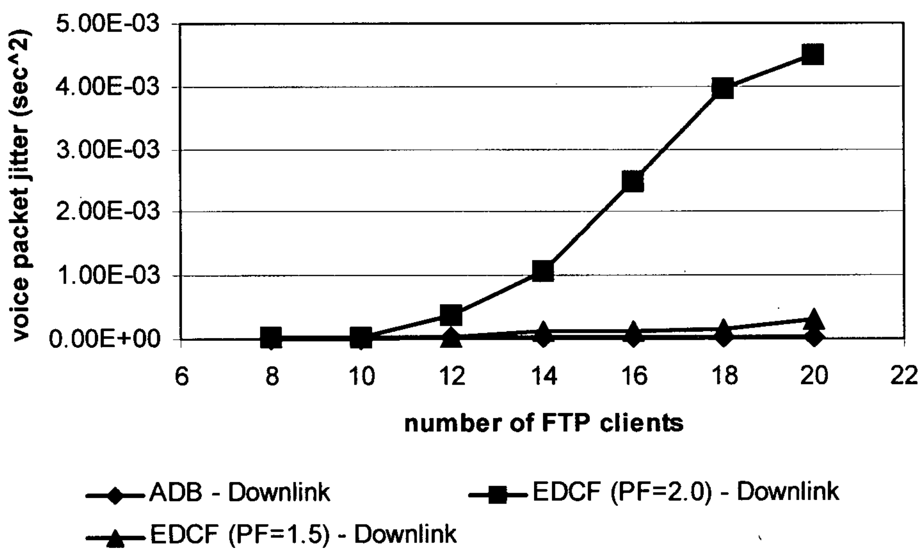


Figure 5.10. Uplink and Downlink voice packet delay in a hotspot

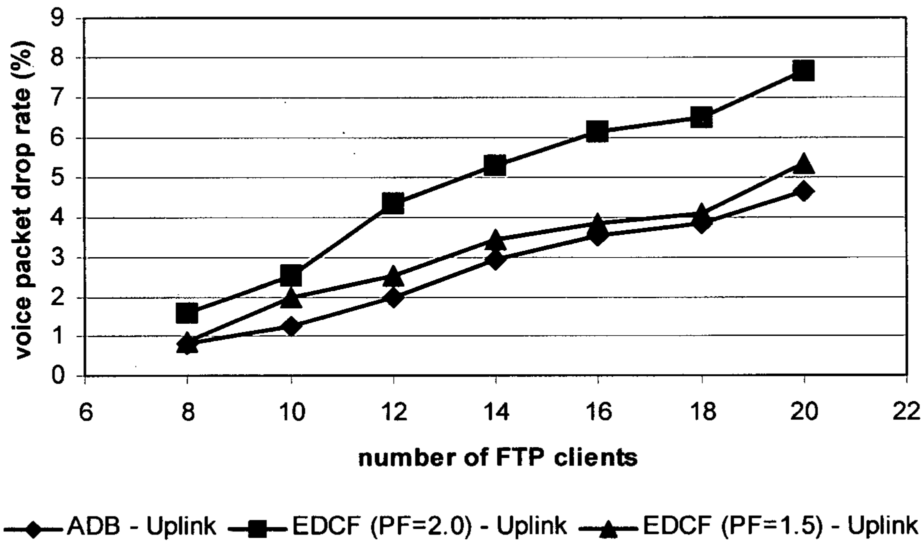


(a)

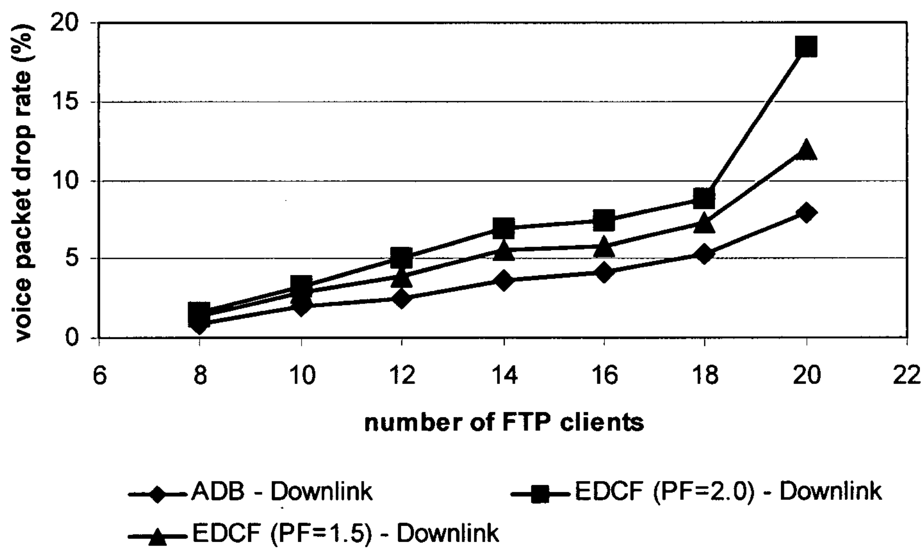


(b)

Figure 5.11. (a) Uplink voice packet jitter in a hotspot.
(b) Downlink voice packet jitter in a hotspot



(a)



(b)

Figure 5.12. (a) Uplink voice packet drop rate in a hotspot
(b) Downlink voice packet drop rate in a hotspot

Figure 5.13 shows video packet delay. Figure 5.14a and 5.14b show uplink and downlink video packet jitter. Figure 5.15a and 5.15b show uplink and downlink video packet drop rates.

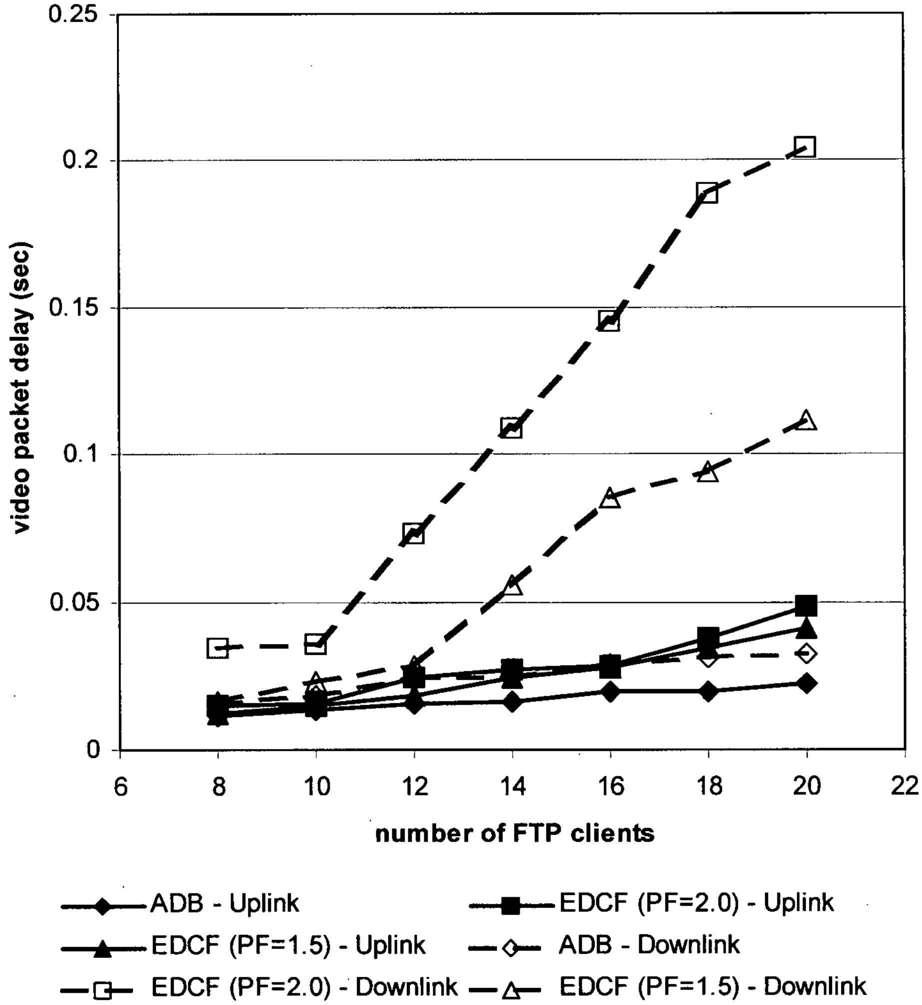
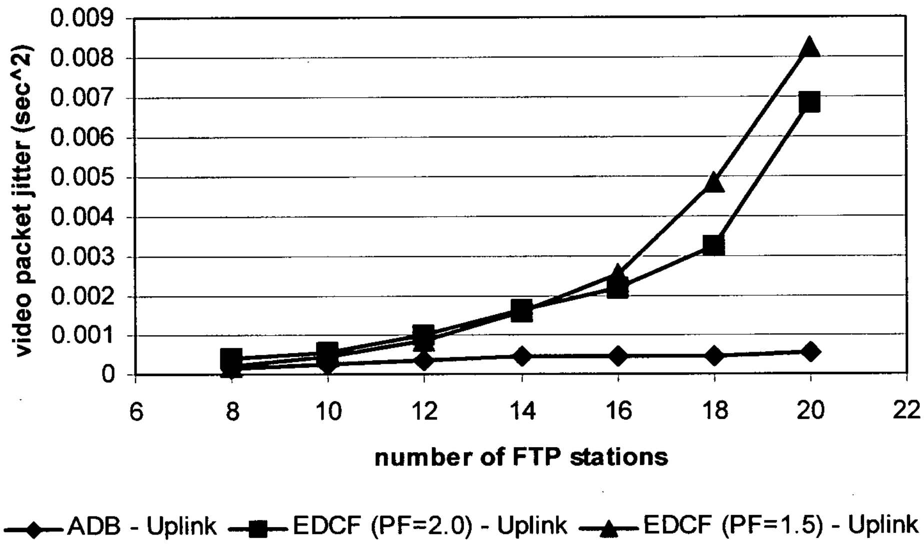
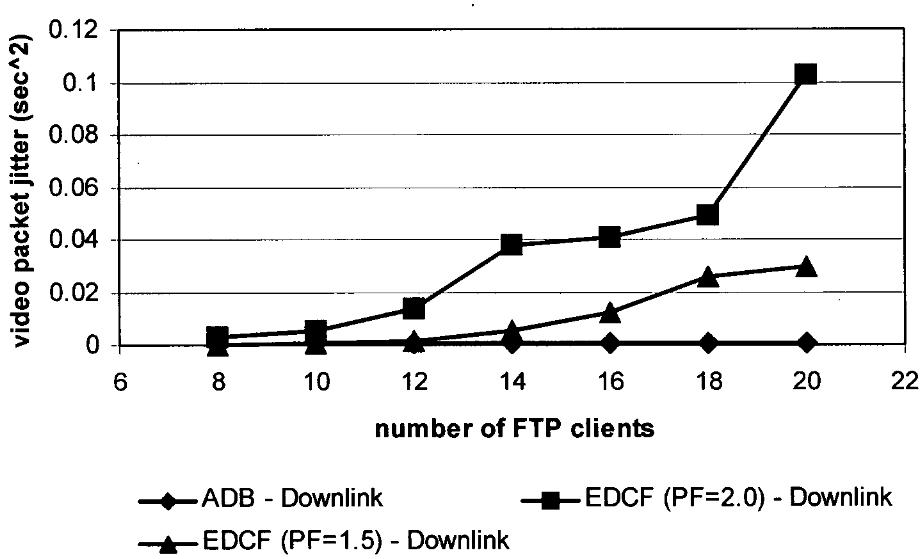


Figure 5.13. Video packet delay in a hotspot

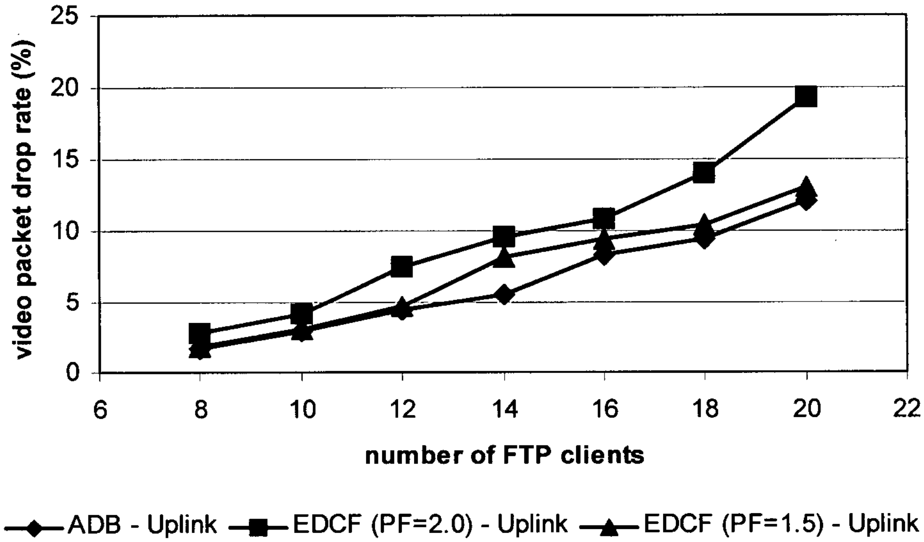


(a)

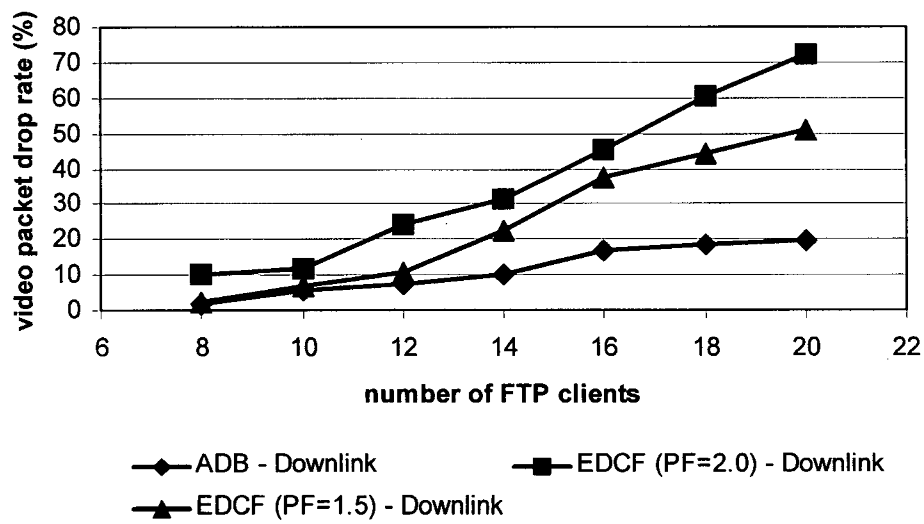


(b)

Figure 5.14. (a) Uplink video packet jitter in a hotspot
(b) Downlink video packet jitter in a hotspot



(a)



(b)

Figure 5.15. (a) Uplink video packet drop rate in a hotspot
(b) Downlink video packet drop rate in a hotspot

The main performance characteristic of data traffic is measured by throughput. The total throughput of the FTP data traffic in the hotspot network is shown in Figure 5.16.

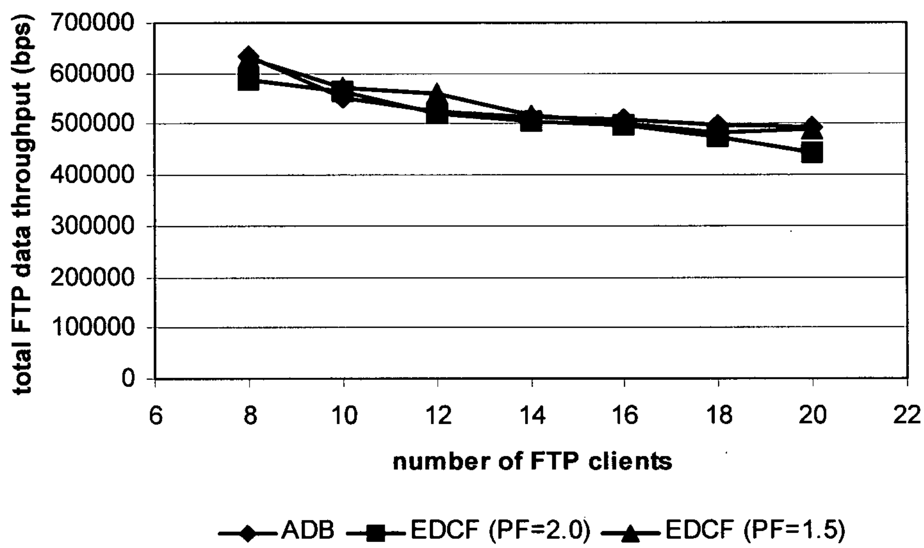


Figure 5.16. Total FTP data throughput in a hotspot

5.2.3 Discussions of the Hotspot Results

Figures 5.10, 5.11 and 5.12 show that, relative to EDCF without ADB, voice packet delay, jitter and drop rate improve significantly as the number of FTP stations increases when EDCF incorporates ADB. Figures 5.13, 5.14 and 5.15 demonstrate that with ADB, video packet delay, jitter and drop rate all improve considerably when many best-effort FTP stations are active.

Like the simulation results from the ad-hoc scenario, those from the hotspot scenario indicate that when the ADB retransmission algorithm is employed in the new 802.11e EDCF protocol, the delay, jitter, and packet drop rate of the voice and video traffic is reduced without sacrificing the throughput of best-effort FTP traffic (see Figure 5.16). The results suggest that ADB enhances QoS performance of EDCF in the hotspot.

The EDCF protocol provides service differentiation, which is an important improvement over 802.11 DCF. EDCF presents delay asymmetry, giving an advantage to uplink transmission as shown in Figures 5.10 and 5.13. This is because the aggregate downlink real-time traffic sent by the AP must complete for the channel in equal terms with all the stations that want to transmit in the uplink direction.

The ADB algorithm reduces delay asymmetry by giving priority to real-time packets with high age values. Downlink real-time traffic having long queuing delay will be assigned to use small PF value as adjustment to their CW sizes.

5.3 Conclusion from the Simulations Results

Results from both the ad-hoc and hotspot scenarios suggest that the ADB retransmission algorithm used in the new 802.11e EDCF protocol offers improvement, relative to that from use of the BEB algorithm (with real-time traffic) in QoS differentiation, under a wide range of traffic loads, in both ad-hoc networks and in hotspots. The performance of the real-time traffic is enhanced without reducing the throughput of the best-effort traffic.

Service differentiation becomes more noticeable and more pronounced as the number n of best-effort stations increases. ADB can also reduce delay asymmetry at the hotspot. The difference between the uplink and the downlink delay is reduced significantly when the ADB algorithm is employed.

There is a penalty for using ADB to reduce delay, jitter, and packet drop rate of real-time traffic while maintaining throughput levels of best-effort data traffic. The best-effort traffic delay and jitter increase, in some cases substantially, when the ADB algorithm is employed. However such delay and jitter increase, not presented here, is not regarded as problematic for best-effort data traffic.

Chapter 6 Summary and Conclusions

The increasing use of wireless and Internet communications has created a strong demand for public Internet access over 802.11 wireless LANs. IETF is currently working on service differentiation at the IP layer, but the result is sub-optimal without lower layers' support. Since 802.11 wireless LANs appear everywhere, it is essential to focus on service differentiation support at the 802.11 MAC layer.

To improve the current 802.11 MAC protocol to support applications with QoS requirements, the IEEE 802.11 Task Group E was formed and is defining QoS enhancements for 802.11 MAC. The 802.11e draft introduces the EDCF protocol, which is a prioritization enhancement of the 802.11 DCF protocol. In the current version of EDCF, the BEB algorithm is employed in EDCF to resolve collisions; however BEB causes long delay and large jitter that are unfavorable for real-time packets with time-bounded requirements.

This thesis is an extension to the current work on providing QoS enhancements in the 802.11 MAC layer studied by IEEE 802.11 Task Group E. In particular, we focus on the improvement of the 802.11e EDCF protocol by proposing our ADB retransmission algorithm.

6.1 Summary of the Work

The primary contribution of this thesis is our proposal, analysis, and evaluation of the ADB retransmission algorithm that can alleviate delay, jitter, and drop rate for real-time traffic with time-bounded requirements without reducing the throughput of best-effort data traffic. The ADB algorithm is used together with the BEB algorithm in 802.11 wireless LANs. The ADB algorithm is for high priority real-time traffic, while the BEB algorithm is for best-effort data traffic.

The ADB retransmission mechanism is easy to implement. It requires minor modifications in the computation of CW thereby minimizing the migration effort from the new 802.11e EDCF protocol. Our ADB retransmission strategy provides backward compatibility to the 802.11 DCF protocol. The implementation of the ADB algorithm is relatively simple and is practical with current software and hardware technologies.

OPNET simulation models are built to study the performance of the new 802.11 EDCF protocol with the proposed ADB retransmission algorithm in two typical environments which are modeled by two simulation scenarios, namely an ah-hoc network and a hotspot scenarios. The results from both scenarios indicate that using ADB in EDCF provides low delay, jitter and drop rate for real-time traffic. The delay asymmetry which exists without ADB is reduced significantly in the hotspot environment when ADB is employed. The improvements are more noticeable and more pronounced as the data traffic load increases.

In conclusion, ADB is a useful retransmission algorithm for the new 802.11e EDCF protocol. Since ADB can be implemented by changing the software without alternating the existing hardware, the cost of implementation is considered to be relatively low.

6.2 Future Work

To further extend the work of this thesis, the following possible directions for future research are suggested.

1. In our work, it was assumed that all packets are transmitted at the same rate. In reality, some stations may transmit faster than others depending on the wireless link quality. Simulation of the proposed ADB algorithm in a noisy environment is desirable for extending validation of the results presented here.

2. Admission control is a key aspect for the real-time mechanism to work well, since such control limits the amount of real-time traffic admitted to 802.11 wireless LANs. An exact design of an admission control scheme for the new 802.11 EDCF protocol with the proposed ADB algorithm requires further investigation.

3. The proposed ADB retransmission algorithm can be applied to the CSMA/CD protocol to support QoS differentiation in 802.3 Ethernet. It would be interesting to study the performance of the 802.3 CSMA/CD protocol with ADB.

References

[1] William Stalling, "IEEE 802.11: Moving closer to practical wireless LANs", IT Professional, vol. 3, issue 3, pp. 17-23, May-June 2001.

[2] Jim Geier, Wireless LANs, Second Edition, Indianapolis, Indiana: Sams Publishing, 2002.

[3] IEEE Std. 802.11-1997, Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.

[4] IEEE Std. 802.11a-1999, Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks - Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 1: High-speed Physical Layer in 5 GHz band, 1999.

[5] IEEE Std. 802.11b-1999, Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 2: High-speed Physical Layer (PHY) extension in the 2.4 GHz band – Corrigendum 1, 1999

[6] IEEE Std. 802.11g-2003, Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and

Physical Layer (PHY) specifications – Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band.

[7] "High on Wi-Fi," Laptop, pp. 36- 47, March 2003.

[8] H. L. Truong and G. Vannuccini, "The IEEE 802.11e MAC for quality of service in wireless LANs," IBM Zurich Research Laboratory, 2002.

[9] IEEE 802.11 TGe, "EDCF Proposed Draft Text," TF-01/131r1, March 2001.

[10] António Grilo and Mário Nunes, "Performance evaluation of IEEE 802.11e," Proc. IEEE PIMRC '02, Lisbon, Portugal, pp. 511-517, Sept 2002.

[11] George W. Wong and Robert W. Donaldson, "Improving the QoS Performance of EDCF in IEEE 802.11e Wireless LANs," Proc. IEEE PACRIM '03, pp. 392-396, Aug. 2003.

[12] Gil Held, Deploying Wireless LANs, New York: McGraw-Hill, 2002.

[13] Alberto Leon-Garcia and Indra Widjaja, Communication Networks, Singapore: McGraw-Hill, 2000.

[14] P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," IEEE Commun. Mag., pp 116-126, Sept 1997.

[15] C. Heide, "CODIAC protocol – Centrialized or distributed integrated access control," IEEE P802.11/93-54, May 1993.

[16] M. A. Visser and M. E. Zarki, "Voice and data transmission over an 802.11 wireless network," Proc. IEEE PIMRC '95, Toronto, Canada, Sept. 1995.

[17] Wen-Tsuen Chen, Bo-Bin Jian, and Shou-Chih Lo, "An adaptive retransmission scheme with QoS support for the IEEE 802.11 MAC enhancement," Proc. IEEE VTC Spring 2002, Birmingham, USA, pp. 70-74, May 2002.

[18] Stefan Mangold, Sunghyun Choi, Peter May and Guido Hietz, “IEEE 802.11e – Fair resource sharing between overlapping basic service sets,” Proc. IEEE PIMRC 2002, Lisbon, Portugal, pp. 166-171, Sept 2002.

[19] Stenfan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor, “IEEE 802.11e wireless LAN for quality of service,” Proc. European Wireless ‘02, Florence, Italy, Feb. 2002.

[20] Lamia Romdhani, Qiang Ni, and Thierry Turletti, “Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks,” see http://citeseer.nj.nec.com/romdhani02aedcf.html.

[21] Mathilde Benveniste, “Tiered Contention Multiple Access (TCMA), a QoS-based distributed MAC protocol,” Proc. IEEE PIMRC ’02, Lisbon, Portugal, pp. 166-171, Sept. 2002.

[22] OPNET Wireless LANs Model Description, OPNET Technologies Inc., 2000.

[23] Raffaele Bruno, Marco Conti, and Enrico Gregori, “IEEE 802.11 Optimal performances: RTS/CTS mechanism vs. basic access,” Proc. PIMRC ’02, Lisbon, Portugal, pp. 166-171, Sept. 2002.

[24] Thomas J. Kostas, Michael S. Borellla, Ikhlaq Sidhu, Guido M. Schuster, Jacek Gragiec, and Jerry Mahler, “Real-Time voice over packet-switching networks,” IEEE Network, pp. 18-27, Jan. 1998.

[25] OPNET TCP Model Description, OPNET Technologies Inc., 2000.

[26] Larry L. Peterson and Bruce S. Davie, Computer Networks: A System Approach, Second Edition, Morgan Kaufmann, 1999.

[27] P. Karn and C. Patridge, "Improving round-trip time estimates in reliable transport protocols," Proc. ACM SIGCOMM '87, Aug. 1987.

[28] V. Jacobson, "Congestion Avoidance and Control," Proc. ACM SIGCOMM, Aug. 1988.

[29] P. Brady, "A model for generating on-off speech patterns in two-way conversation," Bell Syst. Tech. Journal, vol. 48, no. 7, pp. 2245-2272, Sept. 1969.

[30] Richard V. Cox, "Three new speech coders from the ITU cover a range of applications," IEEE Commun. Mag., pp. 40-47, Sept. 1997.