國 立 交 通 大 學

資訊科學系

碩 士 論 文

IEEE 802.11e 無線區域網路之

進階分散式頻道存取機制的效能提升

Improving Performance of Enhanced Distributed Channel Access

in IEEE 802.11e Wireless LANs

研 究 生：康博翔

指導教授：王國禎　博士

中 華 民 國 九 十 三 年 六 月

IEEE 802.11e 無線區域網路之進階分散式頻道存取機制的效能提升

Improving Performance of Enhanced Distributed Channel Access in IEEE 802.11e Wireless LANs

研 究 生：康博翔　　Student：Po-Hsiang Kang

指導教授：王國禎　　Advisor：Kuochen Wang

國 立 交 通 大 學

資 訊 科 學 系

碩 士 論 文

A Thesis

Submitted to Department of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

IEEE 802.11e 無線區域網路之
進階分散式頻道存取機制的效能提升

學生：康博翔　　　指導教授：王國禎 博士

國立交通大學資訊科學系

摘 要

IEEE 802.11e進階分散式頻道存取機制具有能夠提供服務分類以及支援即時性應用的優勢。然而，它在高傳輸負荷下可能會大大地降低產量以及增加碰撞機率，這是由於它在每次封包傳輸成功之後會固定地重設競爭視窗的大小。於本論文中，我們提出了一個新的協定以提升進階分散式頻道存取機制的服務品質，這個協定稱為*改良型進階分散式頻道存取機制*，是為了要解決上述進階分散式頻道存取機制的缺失。改良型進階分散式頻道存取機制在每次封包傳輸成功之後會根據平均的碰撞機率來動態地調整競爭視窗的大小，而不是像進階分

散式頻道存取機制去重設競爭視窗的大小。此外，一旦發生虛擬碰撞，改良型進階分散式頻道存取機制在每次封包傳輸失敗之後會維持較低優先權存取類別的競爭視窗大小，而不是像進階分散式頻道存取機制去加倍競爭視窗的大小。它也提供向後相容於原本的802.11分散協調機制。模擬結果顯示我們的改良型進階分散式頻道存取機制就平均產量而言，比適應性進階分散協調機制、時間相關的後退機制、進階分散式頻道存取機制分別高出9%、11%、以及15%。另外，我們的改良型進階分散式頻道存取機制就高優先權資料的平均封包延遲時間而言，比時間相關的後退機制、適應性進階分散協調機制、進階分散式頻道存取機制分別縮短了22%、26%、以及40%。因此，我們所提之改良型進階分散式頻道存取機制對於在IEEE 802.11e無線區域網路下支援服務品質是很有效的。

關鍵詞：無線區域網路、IEEE 802.11e、進階分散式頻道存取機制、競爭視窗、產量、封包延遲、服務品質。

Improving Performance of Enhanced Distributed Channel Access in IEEE 802.11e Wireless LANs

Student：Po-Hsiang Kang Advisor：Dr. Kuochen Wang

Department of Computer and Information Science
National Chiao Tung University

Abstract

The EDCA in IEEE 802.11e has advantages of providing service differentiation and supporting real-time applications. However, under high traffic load, its throughput performance may degrade and the collision rate may increase significantly due to its statically resetting the contention window size after each successful transmission of packets. In this thesis, we propose a new protocol for quality of service (QoS) enhancement to the EDCA, called *Improved Enhanced Distributed Channel Access* (I-EDCA), to resolve the above shortcomings of EDCA. I-EDCA dynamically adjusts the contention window size according to the *average collision rate* instead of resetting the contention window like EDCA after each successful transmission of packets. Besides, once a virtual collision occurs, I-EDCA retains the contention window size of the Access Category (AC) with lower priority instead of doubling the contention window size like EDCA after each unsuccessful transmission of packets. It also provides backward compatibility to the legacy 802.11 DCF. Simulations results have shown that the average throughput of our I-EDCA is 9%, 11%, and 15% better than AEDCF, ADB, and EDCA, respectively. In addition, for high priority traffic, compared to ADB, AEDCF, and EDCA, I-EDCA shortens the average packet delay by

22%, 26%, and 40%, respectively. Therefore, I-EDCA is effective in supporting the QoS of IEEE 802.11e wireless LANs.

Keywords：WLANs, IEEE 802.11e, EDCA, contention window, throughput, packet delay, QoS.

Acknowledgements

Many people have helped me with this thesis. I am in debt of gratitude to my thesis advisor, Dr. Kuochen Wang, for his intensive advice and instruction. Also, I would like to thank my classmates in *Mobile Computing and Broadband Networking Laboratory* for their invaluable assistance and kind help both in the research and daily life during these two years. The support by the National Science Council under Grant NSC92-2213-E-009-078 is also grateful acknowledged.

Finally, my families deserve special mention. This thesis is dedicated to them for their support.

Contents

Abstract (in Chinese) **i**

Abstract (in English) **iii**

Acknowledgements **v**

Contents **vi**

List of Figures **viii**

List of Tables **x**

1. Introduction..1

2. Existing Approaches 3

2.1 Legacy 802.11 DCF 3

2.2 802.11e EDCA 5

2.3 Adaptive EDCF (AEDCF) 8

2.4 Age Dependent Backoff (ADB)...9

2.5 Comparison 9

3. Design Approach 12

3.1 New MAC Layer Architecture 12

3.2 Proposed I-EDCA Protocol 13

3.2.1 After Each Successful Transmission of Packets 13

3.2.2 After Each Unsuccessful Transmission of Packets 16

4. Evaluation and Discussion 21

4.1 Simulation Model 21

4.2 Performance of I-EDCA 22

5. Conclusions and Future Work 29

5.1 Concluding Remarks 29

5.2 Future Work 30

Bibliography 31

List of Figures

Fig. 1. The timing diagram of legacy 802.11 DCF channel access. 4

Fig. 2. The timing diagram of 802.11e EDCA channel access. 6

Fig. 3. New MAC layer architecture..13

Fig. 4. Virtual collision vs. real collision..17

Fig. 5. Flowchart of the proposed I-EDCA protocol. 20

Fig. 6. Throughput comparison among different approaches under high priority traffic..23

Fig. 7. Throughput comparison among different approaches under medium priority traffic..24

Fig. 8. Throughput comparison among different approaches under low priority traffic..25

Fig. 9. Aggregate throughput comparison among different approaches..................26

Fig. 10. Packet delay comparison among different approaches..............................27

1896

Fig. 11. Average throughput comparison among different approaches....................28

List of Tables

Table 1：The relationship between user priority and AC...5

Table 2：The default EDCA parameters. 7

Table 3：Comparison of I-EDCA with classical approaches. 11

Table 4：The MAC parameters used in the simulation...22

Chapter 1

Introduction

In recent years, Wireless Local Area Networks (WLANs) increasingly become very popular at home, office, and hot spot locations. As WLANs become more cost-effective and more ubiquitous, they will be used for multimedia applications. Subsequently, there must have high demand of Quality of Service (QoS) to support real-time applications such as voice over IP (VoIP), video on demand, and video conferencing. Currently, the most popular WLANs technology is IEEE 802.11. In the IEEE 802.11 Medium Access Control (MAC) layer [1][2][3], it defines two coordination functions： one is the Distributed Coordination Function (DCF) [4][5], the other is the Point Coordination Function (PCF) [4]. However, both functions can not guarantee the QoS. With the growing popularity and acceptance of IEEE 802.11 WLANs, it is important to focus on QoS support at the IEEE 802.11 MAC layer. In order to improve the current IEEE 802.11 MAC protocol for supporting multimedia applications with QoS requirements, IEEE 802.11e [6] has been proposed to improve this shortcoming. A new access method, Hybrid Coordination Function (HCF) [7][11][12], which contains Enhanced Distribution Channel Access (EDCA) [8][9][10] and HCF Control Channel Access (HCCA) [11][13], is included in the IEEE 802.11e.

Recent researches [15][16] have shown that the traditional DCF and the new coming EDCA will deteriorate in throughput and increase packet delay when the number of stations is large. The main problem is that the contention window is reset after each successful transmission of packets. Therefore, a lot of work based on IEEE 802.11 and IEEE 802.11e has been presented to improve throughput and decrease packet delay. Most researches [14][16][17][18] focused on optimizing the contention window instead of resetting contention window mechanism; however, they still did not effectively solve the shortcomings of the traditional DCF and the new coming EDCA. In this thesis, our proposed I-EDCA dynamically adjusts the contention window size according to the *average collision rate* instead of resetting the contention window like EDCA after each successful transmission of packets. Besides, once a virtual collision occurs, I-EDCA retains the contention window size of the Access Category (AC) with lower priority instead of doubling the contention window size like EDCA after each unsuccessful transmission of packets.

The remainder of this thesis is organized as follows. In Chapter 2, we briefly review several existing approaches that were based on adjusting the contention window. The design approach of our proposed protocol, I-EDCA, will be described in Chapter 3. In Chapter 4, we evaluate the performance of I-EDCA and compare it with three existing approaches：EDCA, AEDCF, and ADB. Finally, we give concluding remarks and future work in Chapter 5.

Chapter 2

Existing Approaches

We first briefly review the legacy 802.11 DCF and the 802.11e EDCA. We also review two classical approaches, Adaptive EDCF (AEDCF) [18] and Age Dependent Backoff (ADB) [19]. Finally, we qualitatively compare these existing approaches including legacy 802.11 DCF, 802.11e EDCA, AEDCF, and ADB with our proposed approach.

2.1 Legacy 802.11 DCF

Legacy 802.11 DCF is the mandatory method no matter where it is in an infrastructure network or it is in an ad hoc network, and it is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to support asynchronous data traffic [1]. Each station has to contend for the channel to transmit packets. When a packet arrives at the queue, it has two kinds of situations. First, if the channel has been idle for longer than the DCF Inter Frame Space (DIFS) to avoid the potential collision with other packets of stations, transmission can begin immediately. Second, if the channel is busy, the station must wait for the channel to become idle for the DIFS and activate the Binary Exponential Backoff (BEB) procedure. Each station maintains a contention window, which is used to select the random Backoff Counter (BC). The BC is a pseudo-random integer drawn from a uniform distribution over the interval from zero to the minimum contention window value. If the channel is busy, the BC is frozen. If the channel has been idle for longer than the DIFS, the BC will be decreased. When the BC reaches zero, transmission can begin immediately. The timing diagram of legacy 802.11 DCF channel access is illustrated in Fig. 1 [1].

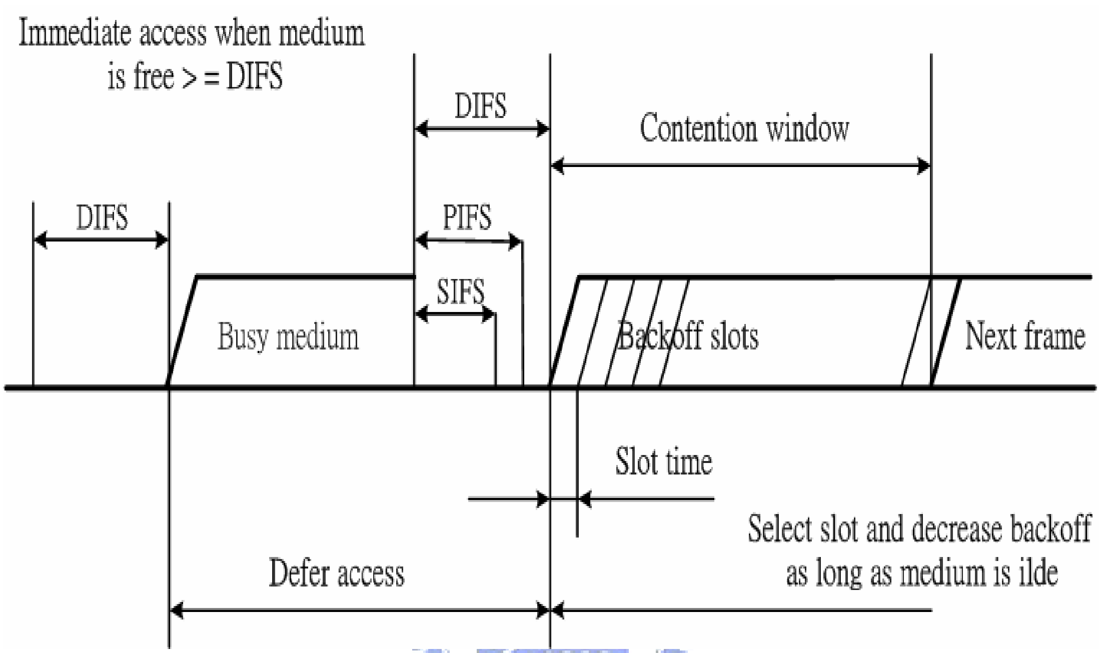

Fig. 1. The timing diagram of legacy 802.11 DCF channel access.

Legacy 802.11 DCF can provide a channel access with equal probabilities to all stations contending for the channel access in a distributed manner. Through this way, each station can contend for the channel access equally. However, legacy 802.11 DCF will deteriorate in throughput and increase packet delay due to its contention window resetting mechanism especially when the network load is very high. Besides, legacy 802.11 DCF can not provide differentiated, distributed channel access for packets with different priorities. Packets with different priorities are considered the same packets. Consequently, legacy 802.11 DCF can not support QoS trivially because it is supposed to provide a channel access with equal probabilities to all stations contending for the channel access in a distributed manner.

2.2 802.11e EDCA

802.11e EDCA is an extension of legacy 802.11 DCF. Their main difference is whether they can support QoS or not. 802.11e EDCA is designed to provide differentiated, distributed channel access for packets with eight different user priorities which are from zero to seven. One or more user priorities are assigned to one AC. The relationship between eight user priorities and four ACs is shown in Table 1 [6]. In Table 1, AC_BK is used for the background traffic. AC_BE is used for the best effort traffic. AC_VI is used for the video traffic. And, AC_VO is used for the voice traffic. In addition, the user priority of AC_VO is the highest. The second is AC_VI. The third is AC_BE. The last is AC_BK.

Table 1：The relationship between user priority and AC.

User Priority	AC	Designation (Informative)
1	AC_BK	Background
2	AC_BK	Background
0	AC_BE	Best Effort
3	AC_VI	Video
4	AC_VI	Video
5	AC_VI	Video
6	AC_VO	Voice
7	AC_VO	Voice

For further differentiation, 802.11e proposed the use of different IFSs according to different ACs. Instead of DIFS, an Arbitration IFS (AIFS) is used. The value of AIFS is determined by the following equation (1)：

$$AIFS = AIFSN \times aSlotTime + SIFS \quad (1)$$

where the value of AIFS Number (AIFSN) is an integer greater than zero and is dependent on each AC. Besides, the values of *a*SlotTime and Short IFS (SIFS) are defined in the physical layer.

The AC with the smallest AIFS will have the highest priority as shown in Fig. 2 [6]. It shows the timing diagram of 802.11e EDCA channel access. We can see that DIFS is equal to the smallest AIFS. However, it does not mean that the priority of the station in DCF is equal to that of the highest priority AC in EDCA. This is because the highest priority AC in EDCA has smaller minimum contention window and smaller maximum contention window. Consequently, the priority of the highest priority AC in EDCA is still higher than that of the station in DCF.

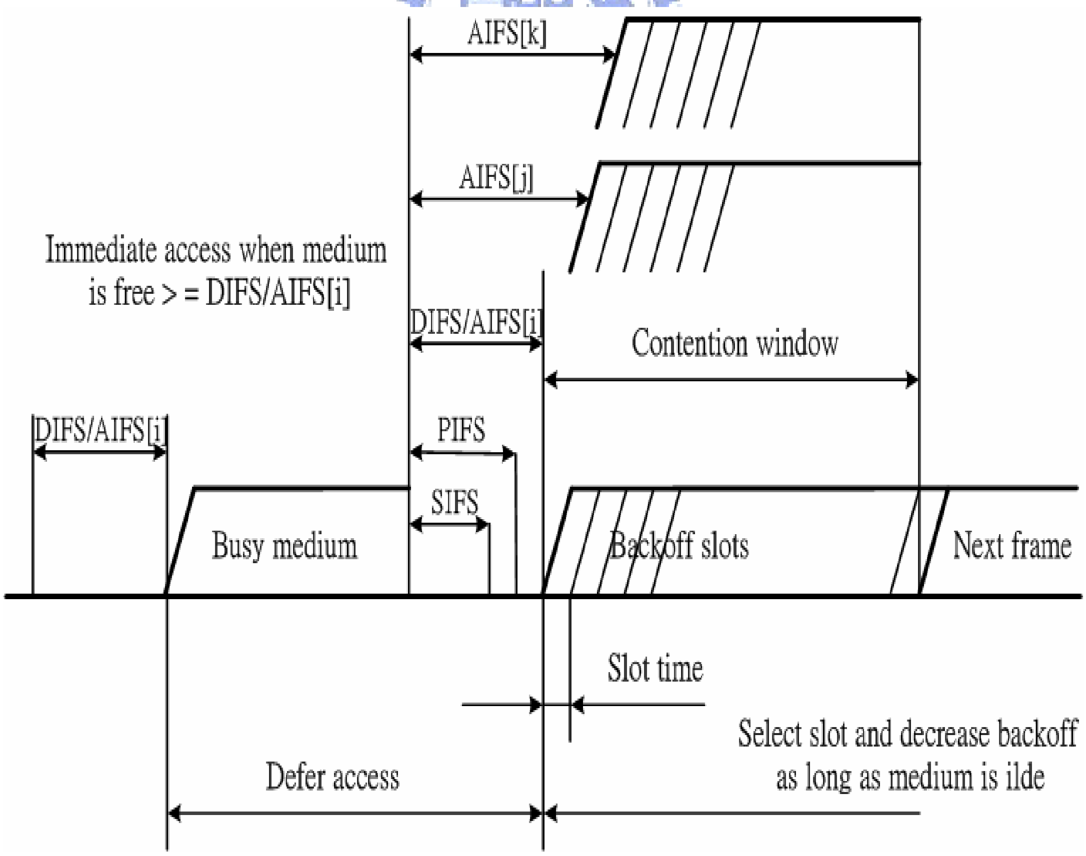

Fig. 2. The timing diagram of 802.11e EDCA channel access.

In 802.11e, the AIFS, the minimum contention window, and the maximum contention window, which are the main parameters in EDCA. Each AC of a station has its own queue and its own parameters such as the minimum contention window, the maximum contention window, and a different AIFS. Table 2 [6] shows the default EDCA parameters for the minimum contention window, CW_{min}, the maximum contention window, CW_{max}, and *AIFSN* for each AC. In addition, based on Table 2 and Eq. (1), the *AIFS* for each AC can be derived.

Table 2：The default EDCA parameters.

AC	CW_{min}	CW_{max}	***AIFSN***
AC_BK	aCW_{min}	aCW_{max}	7
AC_BE	aCW_{min}	aCW_{max}	3
AC_VI	$\frac{aCW_{min}+1}{2}-1$	aCW_{min}	2
AC_VO	$\frac{aCW_{min}+1}{4}-1$	$\frac{aCW_{min}+1}{2}-1$	2

In EDCA, it also uses a similar BEB procedure like that in legacy DCF. Besides, because each AC can be viewed as a virtual station, virtual collisions may occur. If virtual collisions occur, they are resolved by allowing the packet with higher priority to transmit. However, the packet with lower priority is considered as encountering a collision and the corresponding AC needs to double its contention window. After each successful transmission of packets, the contention window is reset to the minimum contention window regardless of the network conditions. After each unsuccessful transmission of packets, the contention window is doubled.

IEEE 802.11e EDCA can support QoS and provide differentiated, distributed channel access for packets with different priorities. However, the contention window resetting mechanism causes a very large variation of the contention window size, and increase the probability of collisions, especially when the network is heavily loaded. Therefore, most researches [18][19] focused on how to decrease the probability of collisions and to improve the overall system throughput.

2.3 Adaptive EDCF (AEDCF)

In [18], the authors proposed a different protocol from 802.11e EDCA, called AEDCF. In brief, this protocol dynamically adjusts the contention window after each successful transmission of packets by taking into account both applications requirements and network conditions using a calculated collision rate. Besides, it adopts the Persistence Factor (PF) which is earlier version of 802.11e draft to adjust the contention window after each unsuccessful transmission of packets.

In this protocol, although its dynamically calculating contention window may decrease the collision rate, the delays of the lower priority packets may increase too much than before. The reason is that lower priority packets have bigger minimum contention window and maximum contention window than other priority packets. When the network is heavily loaded, the lower priority packets increase its contention window with a PF greater than 2 after each unsuccessful transmission of packets. This increases considerably the waiting time of lower priority packets and increases the overall packet delay when collisions occur. It may even lead to starvation.

2.4 Age Dependent Backoff (ADB)

ADB [19] is so called a new retransmission protocol which is different from 802.11e EDCA. This protocol dynamically adjusts the contention window after each unsuccessful transmission of packets. The basic idea of ADB is to dynamically adjust PF based on the ages of the real-time packets in the transmission queues and the lifetimes of real-time packets. Besides, by using the concepts of the age and the lifetime of packets, packets with the queuing delay longer than the lifetime, will eventually be discard by their applications and will not contend for the medium. Therefore, one can save the bandwidth and prevent causing additional delay to other packets. Finally, ADB only requires minor modifications in the computation of the contention window to minimize the migration effort from the new 802.11e EDCA and provides backward compatibility to the legacy 802.11 DCF.

Although ADB can alleviate the delay and jitter of real-time packets by adjusting the contention window dynamically after each unsuccessful transmission of packets, it did not solve the contention window resetting mechanism problem. Due to resetting the contention window after each successful transmission of packets, the collision rate is still high at high traffic load.

2.5 Comparison

We summarize the four existing approaches, 802.11 DCF, 802.11e EDCA, AEDCF, and ADB, which have been described above and compared them with our proposed scheme. The scheme we proposed is called I-EDCA which is an improved scheme of EDCA in IEEE 802.11e. The following five metrics are considered: priority classification, contention window adjustment, collision rate, throughput, and packet delay.

First, the priority classification metric checks if an approach can provide differentiated, distributed channel access for packets with different priorities. In these approaches, only 802.11 DCF can not provide differentiated, distributed channel access for packets with different priorities. Second, the contention window adjustment metric judges if an approach adopts static or dynamic contention window adjustment after each successful or unsuccessful transmission of packets. Among these approaches, 802.11 DCF and 802.11e EDCA adopted static contention window adjustment based on the contention window resetting mechanism; the other approaches adopted dynamic contention window adjustment based on the network load. Third, the collision rate metric is the percentage of the transmitted packets encountered collisions. Again, I-EDCA has the lowest collision rate among these approaches because of its flexible contention window adjustment. Fourth, the throughput metric means that the total data is actually delivered to destination. I-EDCA has the highest throughput among these approaches because of its flexible contention window adjustment. Fifth, the packet delay metric is the time that the head-of-line queue packet takes before being successfully transmitted out. I-EDCA has the lowest packet delay among these approaches because of its flexible contention window adjustment. Table 3 summarizes the comparison results in terms of the above five metrics. In Chapter 4, quantitative comparison and discussion will be made.

Table 3：Comparison of I-EDCA with classical approaches.

Approach	Priority classification	Contention window adjustment	Collision rate	Throughput	Packet delay
802.11 DCF [1]	No	Static	High	Low	High
802.11e EDCA [6]	Yes	Static	High	Medium(15%)	High(40%)
AEDCF [18]	Yes	Dynamic	Low	Medium(9%)	Medium(26%)
ADB [19]	Yes	Dynamic	Medium	Medium(11%)	Medium(22%)
I-EDCA (proposed)	Yes	Dynamic	Low	High	Low

Chapter 3
Design Approach

In order to efficiently support time-bounded multimedia applications, and to avoid the drawbacks of the IEEE 802.11e EDCA, we propose a dynamic contention window adjustment procedure that takes into account of the network condition to change the contention window size after each successful transmission of packets. Besides, we also statically adjust the contention window size after each unsuccessful transmission of packets. This scheme is called *Improved EDCA* (I-EDCA).

3.1 New MAC Layer Architecture

The new MAC layer consists of three sublayers, as shown in Fig. 3. First, the lowest sublayer of the MAC layer is called legacy DCF as in the IEEE 802.11 standard, which provides fundamental access used in the contention-based period of the superframe to support asynchronous data traffic. Next, another sublayer of the MAC layer is called legacy PCF as in the IEEE 802.11standard, which is built on top of legacy DCF and provides a polling mechanism in the contention-free period of the superframe to centrally control channel usage. Finally, the sublayer HCF is on top of legacy DCF and legacy PCF, which consists of HCCA and I-EDCA to support the

QoS of real-time traffic. Note that we did not modify the HCCA structure of the sublayer HCF. That is, the function of HCCA is the same as that in the IEEE 802.11e.

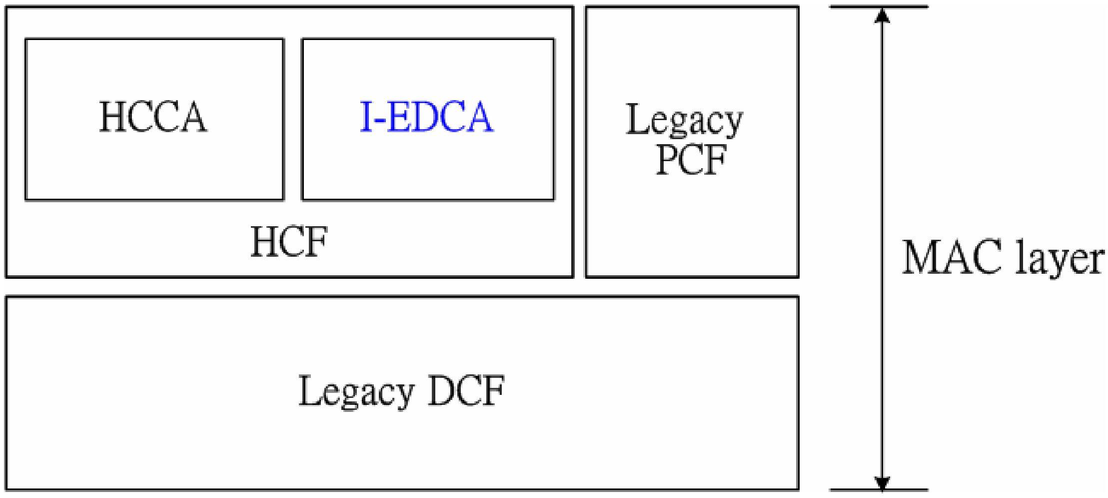

Fig. 3. New MAC layer architecture.

3.2 Proposed I-EDCA Protocol

The proposed I-EDCA protocol can operate in the infrastructure mode and ad hoc mode. In the infrastructure mode, I-EDCA can be used for association with an access point (AP) or contending the Transmission Opportunity (TXOP) in the contention-based period of the superframe. In the ad hoc mode, I-EDCA can be used for contending TXOP. As mentioned above, we use a dynamic procedure that takes into account the network condition to change the contention window size after each successful transmission of packets. Besides, we also statically adjust the contention window size after each unsuccessful transmission of packets. Thus, we divide the I-EDCA protocol into two parts to describe：

3.2.1 After Each Successful Transmission of Packets

At first, we discuss the situation of how to change the contention window size after each successful transmission of packets. In the current IEEE 802.11e draft [6], no matter how many active stations, EDCA will reset the contention window of a transmitting station to the minimum contention window after each successful transmission. However, the contention window resetting mechanism will cause a very high variation of the contention window size, and degrade the overall performance when the network is heavily loaded. Based on the above observation, we suggest that the contention window may be adjusted according to the collision rate and the priority of a station instead of resetting the contention window every time. The following three steps will explain how to proceed when the packet transmission is successful.

First, similar to [18], the network condition must be measured periodically by calculating the *estimated collision rate*, R^j_{cur} in each station. The *estimated collision rate*, R^j_{cur}, is computed as follows [18]：

$$R^j_{cur} = \frac{n^j_{collisions}}{n^j_{packets\_sent}} \qquad (2)$$

where $n^j_{collisions}$ is the number of collisions in one station which occurred at period j, and $n^j_{packets\_sent}$ is the number of total packets sent, including the number of collisions and the number of packets successfully sent in one station which occurred at the same period j. Note that the above ratio R^j_{cur} is in the range of [0,1].

Besides, because R^j_{cur} can only represent the current network condition, it may cause the extreme bias between two continuous periods. In order to avoid the above mentioned condition, we apply the exponentially weighted average method [18] to alleviate the bias and to smooth the *estimated collision rate*, R^j_{cur}, in each period. We define R^j_{avg} as the *average collision rate* at period j. The calculation of the *average collision rate*, R^j_{avg}, is as follows [18]：

$$R^{j}{}_{avg} = \alpha \times R^{j-1}{}_{avg} + (1-\alpha) \times R^{j}{}_{cur} \quad (3)$$

where α is a *smoothing factor*, and its value can affect the degree of smoothing the *estimated collision rate*. If α is close to 1, it represents that $R^{j}{}_{avg}$ is almost decided by the previous *average collision rate*. If α is close to 0, it represents that $R^{j}{}_{avg}$ is almost decided by the current collision rate. In addition, the value of α will also indirectly influence the overall throughput and delay.

Note that we must update the *average collision rate*, $R^{j}{}_{avg}$, after a fixed number of time slots at each station. And the fixed number of time slots is called a period. The size of a period should be moderate. If it is too long, an appropriate collision rate can not be estimated in time. On the contrary, if it is too short, we probably spend extra overheads on calculating the *average collision rate*.

Second, after obtaining the *average collision rate*, $R^{j}{}_{avg}$, we can use it to update the contention window after each successful transmission of packets. But it is not enough to use $R^{j}{}_{avg}$ only to update the contention window because there are packets with eight different priorities which are from zero to seven. Consequently, we must ensure that a different user priority has a different contention window according to user priority *i*. We define a *decreasing factor* related to updating the contention window called β. β is based on the *average collision rate*, $R^{j}{}_{avg}$, and the user priority *i*, as follows：

$$\beta = max(1 - R^{j}{}_{avg} \times (7 - i + 0.1), 0) \quad (4)$$

Note that β is not less than zero. When the user priority *i* is equal to seven, β is independent from the *average collision rate*, $R^{j}{}_{avg}$. In order to avoid that, we add 0.1 to make $R^{j}{}_{avg}$ still exist. Since 0.1 is very small, it has little impact on β. In this way, we can ensure that a packet with higher user priority has a larger *decreasing factor* β than that with lower user priority.

Third, according to the above three equations, Eq. (2), (3) and (4), we can compute the CW^{i}_{new} of the AC with user priority *i* in a station after each successful transmission of packets as follows：

$$CW^{i}_{new} = CW^{i}_{old} - (CW^{i}_{old} - CW^{i}_{min}) \times \beta \qquad (5)$$

where CW^{i}_{new} is the updated contention window after each successful transmission of packets, CW^{i}_{old} is the current contention window, and CW^{i}_{min} is the minimum contention window.

Because the packet with higher user priority has a larger *decreasing factor* β than that with lower user priority, we can guarantee that the packet with higher user priority has a smaller new contention window CW^{i}_{new} than that with lower user priority by Eq. (5). That is, we will not reset the current contention window to the minimum contention window after each successful transmission of packets, unlike that suggested in the IEEE 802.11e draft [6]. Instead, we adjust the contention window by Eq. (5) in order to improve the overall performance when the network is heavily loaded.

3.2.2 After Each Unsuccessful Transmission of Packets

Next, we discuss the case of how to change the contention window value after each unsuccessful transmission of packets. Packet collisions of the IEEE 802.11e can be classified into two situations. One is virtual collisions between ACs in a station. The other is real collisions between stations. We will explain how to deal with each situation.

Fig. 4 is used to illustrate virtual collisions and real collisions. We use an instance to explain. If AC_BK of STA 1 has a packet to transmit and AC_BE of STA 1 also has a packet to transmit, they will separately use their AIFS and the minimum contention window to enter the backoff procedure. If both the backoff timers count down to zero at the same time, a virtual collision occurs. In addition, if AC_BK of STA 1 has a packet to transmit and AC_BK of STA 2 also has a packet to transmit, they will separately use their AIFS and the minimum contention window to enter the backoff procedure. If both the backoff timers count down to zero at the same time, a real collision occurs. In the following, how to handle virtual collisions and real collisions will be described.

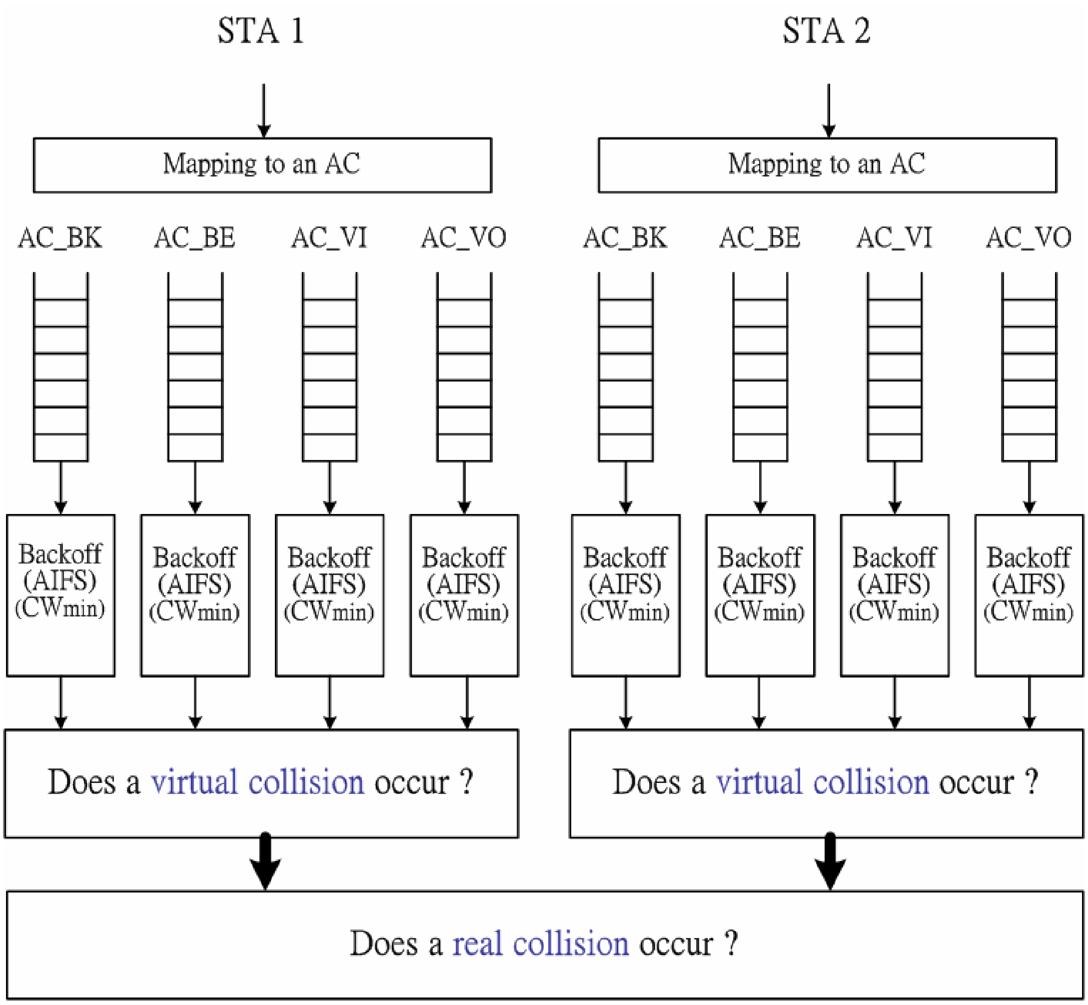

Fig. 4. Virtual collision vs. real collision.

Virtual collisions are resolved by allowing the packet with higher priority to transmit, while the packet with lower priority will not modify its contention window values after each unsuccessful transmission of packets. That is, we will not double the current contention window CW^{i}_{old} when the packets with lower priority encounter collisions. Since the collisions are not real collisions and they will not contribute to the *average collision rate*, we will not double the current contention window CW^{i}_{old}. If we double the current contention window CW^{i}_{old}, the packet with lower priority will have more delay. In this way, the system performance can be improved. That is, after encountering a virtual collision, the CW^{i}_{new} of the AC with lower user priority *i* in a station can be expressed as follows：

$$CW^{i}_{new} = CW^{i}_{old} \tag{6}$$

Next, collisions occur between stations are referred as real collisions. Real collisions increase packet delay and decrease system performance. We double the current contention window CW^{i}_{old} to avoid subsequent collisions; however, it can not exceed the CW^{i}_{max}. Therefore, after encountering a real collision, the CW^{i}_{new} of the AC with user priority *i* in a station can be expressed as follows [6]：

$$CW^{i}_{new} = min(CW^{i}_{old} \times 2, CW^{i}_{max}) \tag{7}$$

In sum, our I-EDCA dynamically changes the contention window value after each successful transmission of packets to reduce possible later collision. If collisions still occur, through our method we can reduce the impact to avoid the next collisions and decrease packet delay after each unsuccessful transmission of packets when virtual collisions occurred.

The flowchart of our I-EDCA protocol is illustrated in Fig. 5. When a new packet is in the head-of-line queue and is ready to be transmitted, the station must check if a channel is idle. If the channel is busy, the station must enter the backoff procedure. On the contrary, if the channel is idle, the station can start to transmit a packet. In the meantime, the station must check if a virtual collision occurs. If the virtual collision occurs, it is resolved by allowing the packet with higher priority to transmit, while the packets with lower priority update their contention window values according to Eq. (6) after each unsuccessful transmission of packets. That is, we will not double the current contention window CW^i_{old}. On the contrary, if a virtual collision did not occur, the station must further check if a real collision occurs. If the real collision occurs, the station updates the contention window value according to Eq. (7) after each unsuccessful transmission of packets. That is, we must double the current contention window CW^i_{old}. If a real collision did not occur, the station can successfully transmit the packet and the station must update the contention window value according to Eq. (5) after each successful transmission of packets. That is, we dynamically adjust the current contention window, CW^i_{old}, according to the *average collision rate*, R^j_{avg}.

Note that I-EDCA only requires small overhead in calculating the *average collision rate*, R^j_{avg}, and updating the contention window after each successful transmission of packets, and it also provides backward compatibility to the legacy 802.11 DCF.

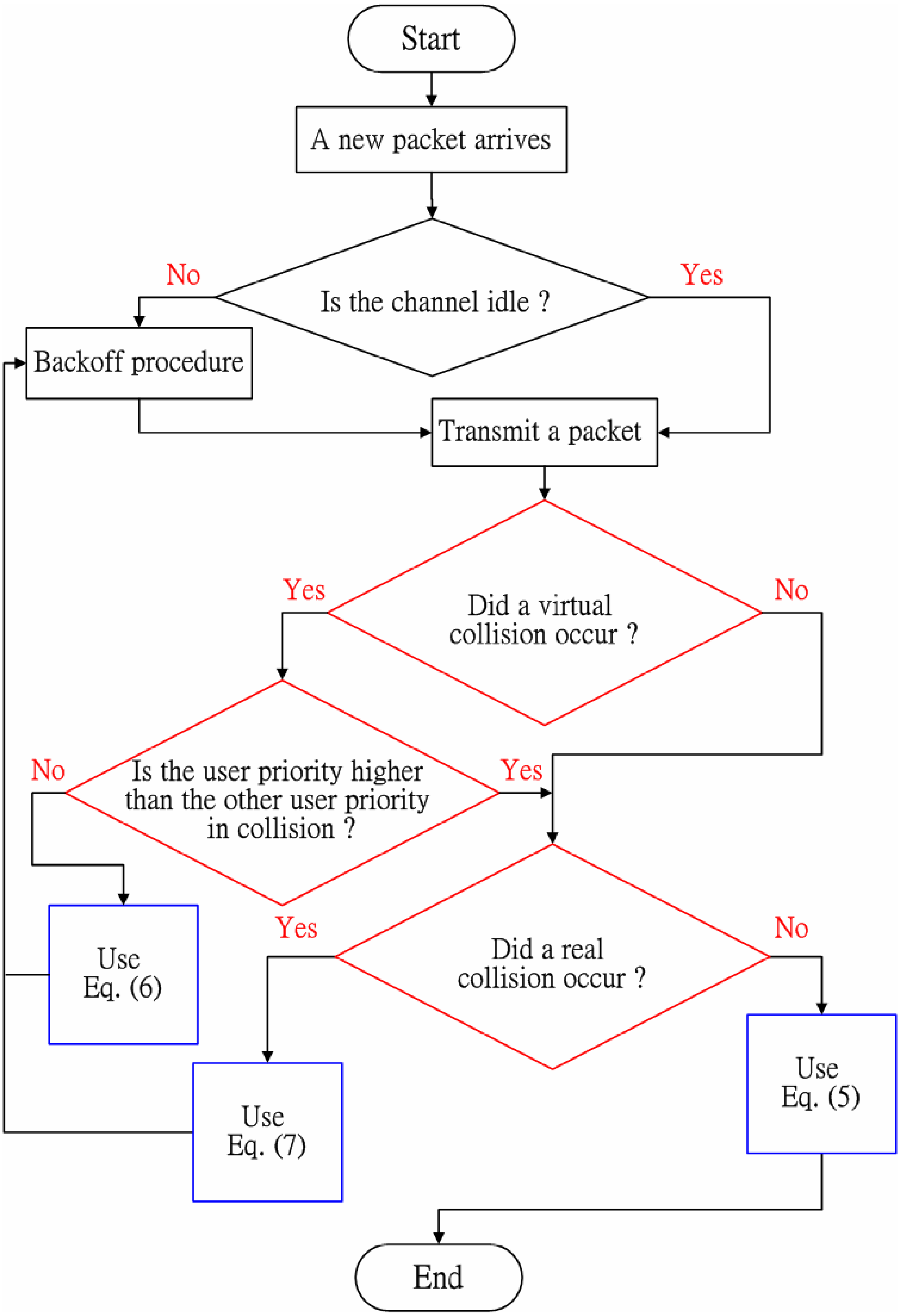

Fig. 5. Flowchart of the proposed I-EDCA protocol.

Chapter 4

Evaluation and Discussion

We use a popular network simulator, ns-2 [25], running on the Linux platform to simulate our proposed approach. Ns-2 is an open source software and it supports wired and wireless networking protocols. In order to demonstrate the superiority of I-EDCA, we compare it with the other three classical approaches, which have been reviewed in section 2.2, in terms of throughput and packet delay.

4.1 Simulation Model

All simulation results were obtained by running the ns-2 simulator. As referred from [19][20][23][24], assume that there are from 5 to 50 stations in an ad hoc network and each station generates three different types of traffic, namely, high, medium, and low priority traffic. The high, medium, and low priority traffic types are corresponding to AC_VO, AC_VI, and AC_BE, respectively. We use three Constant Bit Rate (CBR) [10][18] sources to simulate the three different types of traffic. The MAC parameters used in the simulation, referred from [1][6][18][21][22], are shown in Table 4. The parameters, CW_{min}, CW_{max}, and *AIFSN*, are the default EDCA parameters, referred from Table 2. Besides, the slot time is set to 9 μs, the SIFS is set to 16 μs, the *smoothing factor* α is set to 0.8, and the size of a period to update the *average collision rate*, R^{j}_{avg}, is set to 3000 time slots in this simulation [18].

Table 4 : The MAC parameters used in the simulation.

Parameters	Priority traffic		
	High	**Medium**	**Low**
CW_{min}	7	15	31
CW_{max}	15	31	1023
AIFSN	2	2	3
AIFS (μs)	34	34	43
PF	2	2	2
Frame size (bytes)	160	1280	200
Inter-frame interval (*ms*)	20	10	12.5
Sending rate (Kbytes/s)	8	128	16

4.2 Performance of I-EDCA

We first compare the throughput of I-EDCA with that of EDCA, AEDCF, and ADB. Simulation results under high, medium, and low priority traffic are shown in Fig. 6, Fig. 7, and Fig. 8, respectively. As shown in Fig. 6, we can see that the four approaches had almost equal throughput performance under high priority traffic. When the number of stations exceeds 35, AEDCF, ADB, and I-EDCA achieved better throughput than EDCA. The reason is that these three approaches all dynamically adjust the contention window size, which can reduce the collision rate at high load condition.

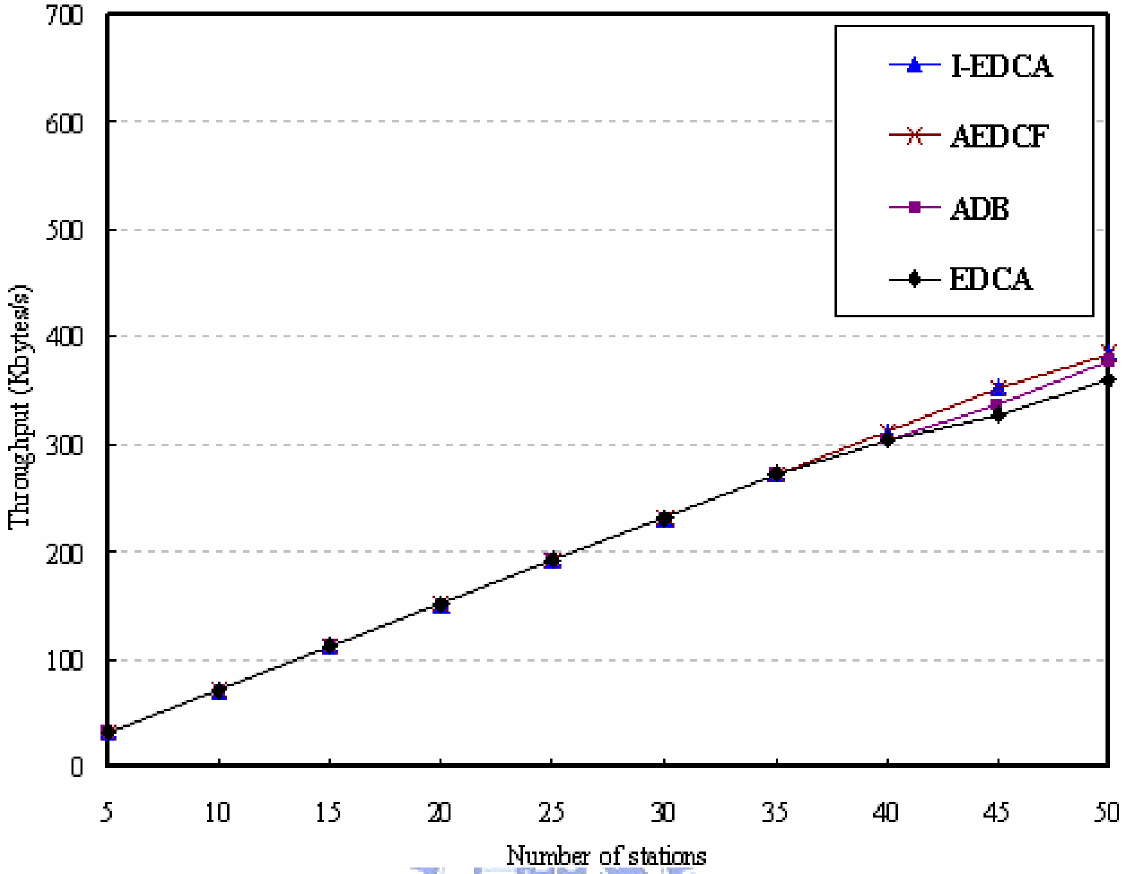

Fig. 6. Throughput comparison among different approaches under high priority traffic.

The throughput comparison under the medium priority traffic is shown in Fig. 7. For the medium priority traffic, we can see that the throughput of all four approaches begins to drop when the number of stations exceeds 15. The reason is that at this point the number of stations is too many and there is not enough traffic for handling under the medium priority traffic. We can see that the throughput of I-EDCA is better than that of EDCA, AEDCF, and ADB when the number of stations is more than 10. Besides, although the throughput of I-EDCA begins to drop when the number of stations exceeds 15, the throughput of I-EDCA is still better than that of EDCA, AEDCF, and ADB even if the number of stations grows up to 50. The reason is that I-EDCA always adjusts the contention window whether each packet transmission is successful or not. In addition, the throughput of EDCA is the lowest due to its contention window resetting mechanism.

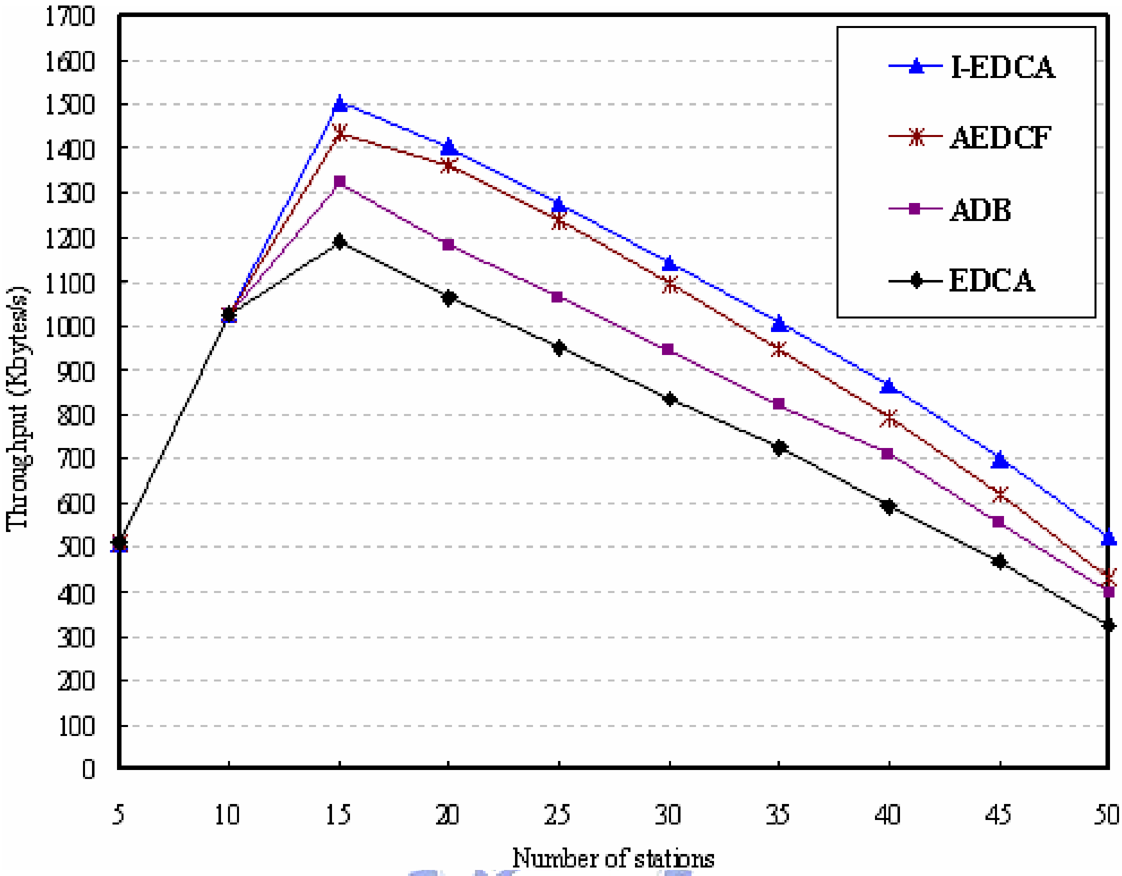

Fig. 7. Throughput comparison among different approaches under medium priority traffic.

In Fig. 8, for the low priority traffic, we can see that the throughput of EDCA, AEDCF, ADB, and I-EDCA all begins to drop when the number of stations exceeds 10. The reason is that at this point the number of stations is too many and there is not enough traffic for handling under the low priority traffic. We can see that the throughput of AEDCF is the lowest compared to that of EDCA, ADB, and I-EDCA when the number of stations exceeds 15. The reason is that AEDCF adopts a large PF, 5, for the low priority traffic, and it increases the waiting time of the low priority packets. Besides, in order to improve the throughput of high priority traffic, I-EDCA sacrifices a little throughput of low priority traffic. This is why the throughput of I-EDCA is lower than that of EDCA when the number of stations is between 25 and

50.

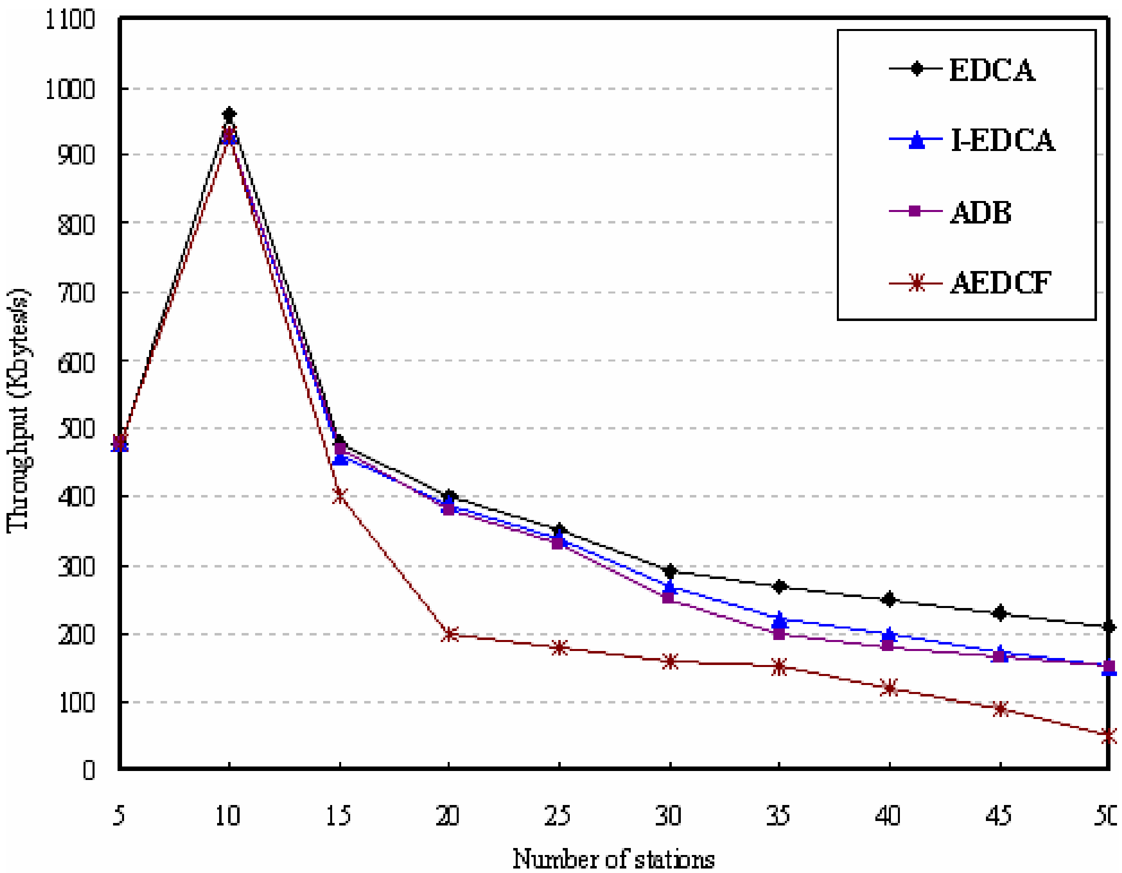

Fig. 8. Throughput comparison among different approaches under low priority traffic.

We compare the *aggregate throughput* among different approaches in Fig. 9. The aggregate throughput represents the total throughput of high, medium, and low priority traffic. In Fig. 9, we can see that I-EDCA provides better aggregate throughput compared to EDCA, AEDCF, and ADB, especially when the number of stations exceeds 10. This is because for the medium priority traffic the throughput of I-EDCA is much better than that of EDCA, AEDCF, and ADB when the number of stations exceeds 10. In addition, we can see that the aggregate throughput of EDCA performs the worst among these approaches when the number of stations exceeds 10. This is because for the medium priority traffic the throughput of EDCA is the worst among that of I-EDCA, AEDCF, and ADB when the number of stations exceeds 10.

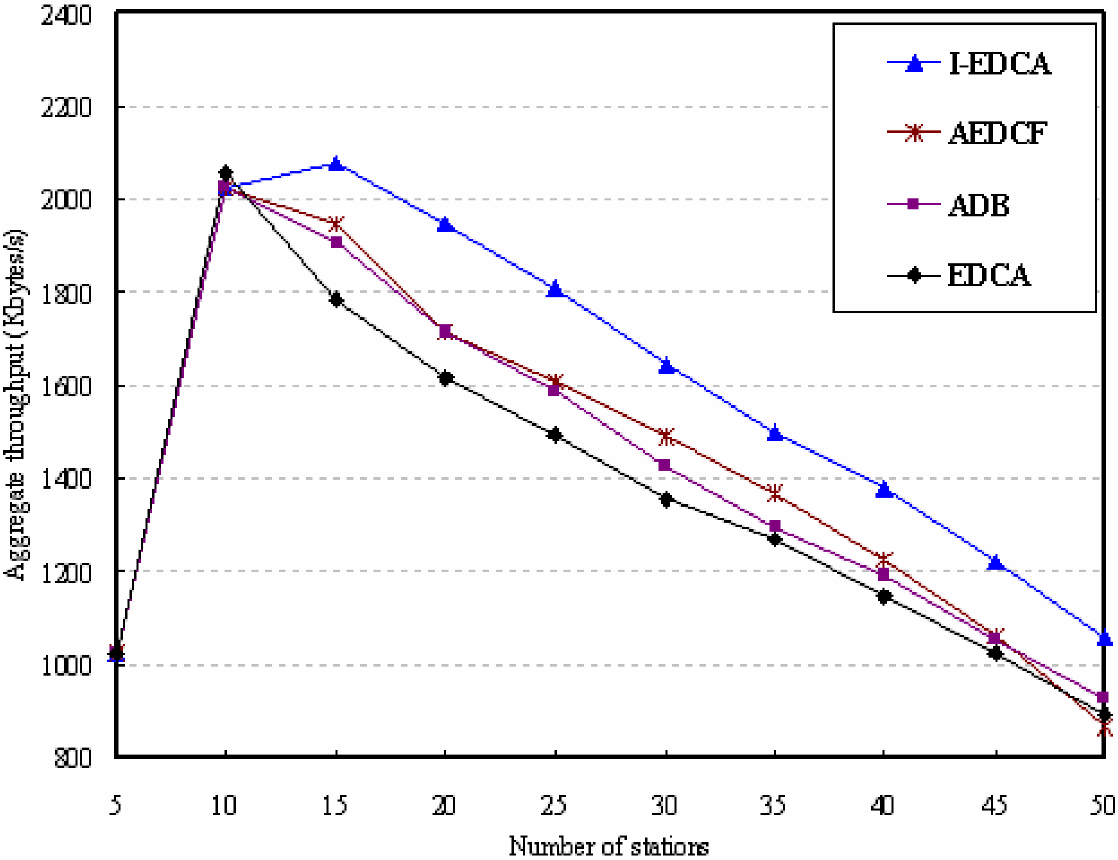

Fig. 9. Aggregate throughput comparison among different approaches.

In Fig. 10, when the number of stations exceeds 10, we can see that I-EDCA had the smallest packet delay for the high priority traffic among these approaches. This is because it adjusts the contention window whether each packet transmission is successful or not.

The packet delay of the medium priority traffic is also shown in Fig. 10. For the medium priority traffic, due to having the highest collision rate among these approaches, EDCA performs the worst. Besides, we can see that I-EDCA achieves the smallest packet delay among these approaches. The reason is that I-EDCA will not double its contention window when the packets with lower priority encounter internal collisions in a station. Although AEDCF also dynamically adjusts its contention window after each successful transmission of packets, it adopts a large PF, 4, for the low priority traffic. Consequently, AEDCF has longer packet delay than I-EDCA, especially when the network load is high. Note that we did not plot the packet delay of the low priority traffic. This is because the packet with low priority is delay tolerable.

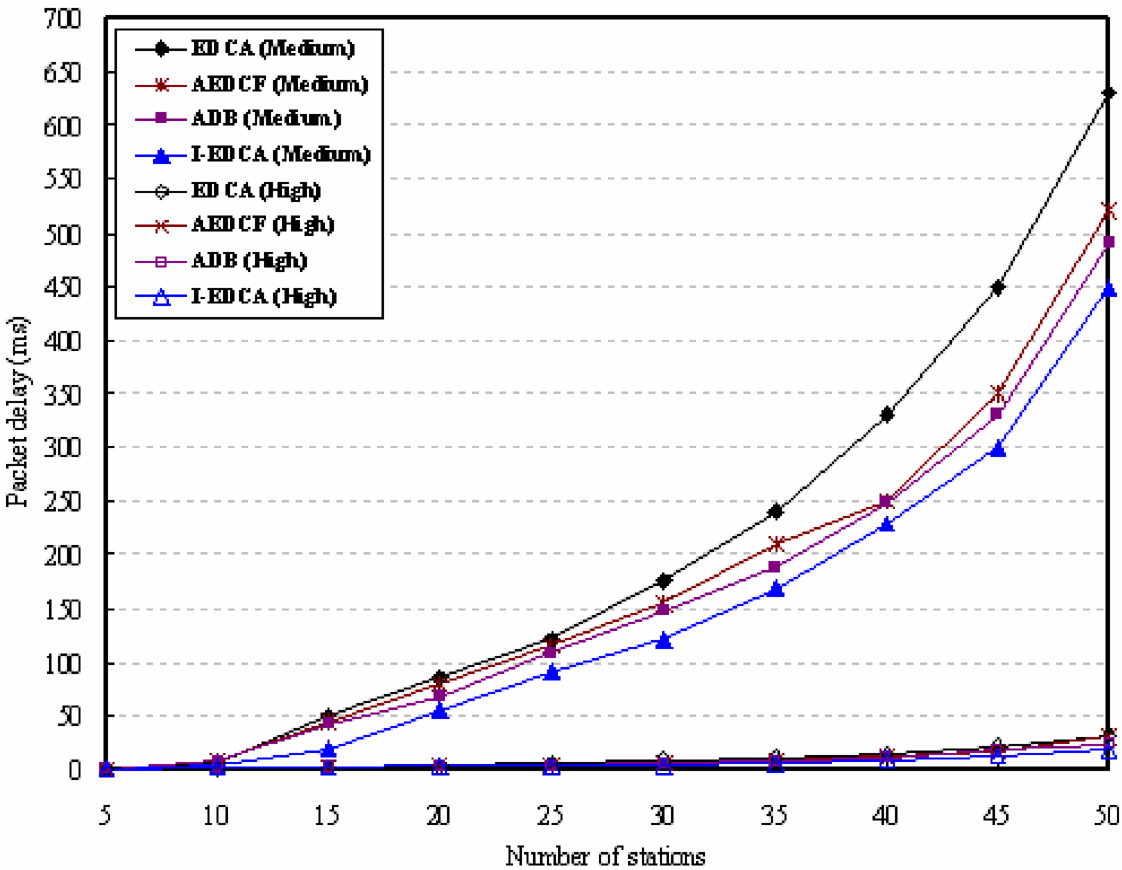

Fig. 10. Packet delay comparison among different approaches.

Besides, we compare the *average throughput* of I-EDCA with that of EDCA, AEDCF, and ADB in Fig. 11. The average throughput is the average of aggregate throughput for the number of stations from 5 to 50. From the simulation results, we can see that our proposed I-EDCA performs better than the other three approaches, EDCA, AEDCF, and ADB. The average throughput of our proposed I-EDCA is 9%, 11%, and 15% greater than that of AEDCF, ADB, and EDCA, respectively. In addition, for high priority traffic, compared to ADB, AEDCF, and EDCA, I-EDCA shortens the average packet delay by 22%, 26%, and 40%, respectively. In sum, the simulation results have shown that the proposed I-EDCA can indeed improve the throughput and packet delay of IEEE 802.11e.

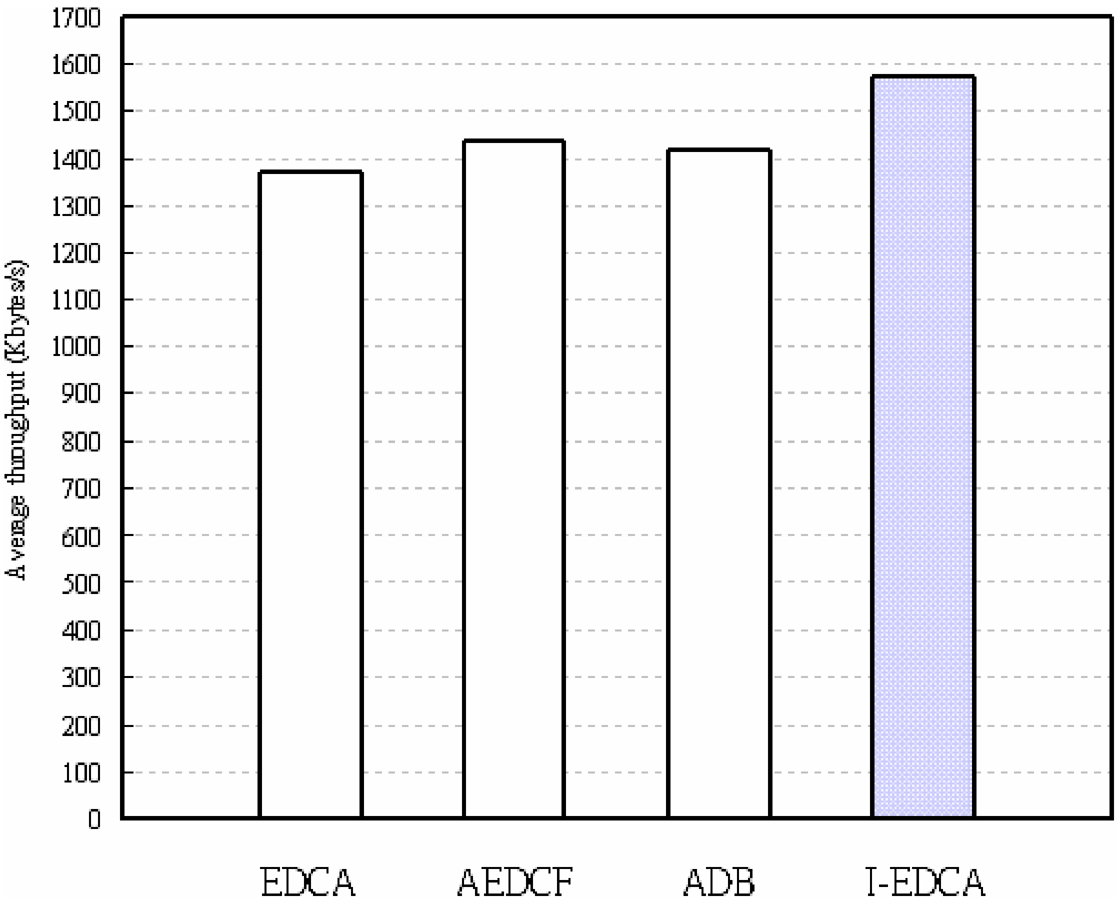

Fig. 11. Average throughput comparison among different approaches.

Chapter 5

Conclusions and Future Work

5.1 Concluding Remarks

In this thesis, we have presented a better service differentiation scheme, I-EDCA for QoS enhancement to IEEE 802.11e EDCA. The basic idea is that I-EDCA dynamically adjusts the contention window size according to the *average collision rate* after each successful transmission of packets. Besides, once a virtual collision occurs, I-EDCA retains its contention window size of the AC with lower priority after each unsuccessful transmission of packets. Although I-EDCA has the overheads of calculating the *average collision rate* and updating the contention window after each successful transmission of packets, it effectively reduces the contention window size according to different user priorities, and enhances the performance in terms of average throughput and packet delay. Simulations results have shown that the average throughput of our I-EDCA is 9%, 11%, and 15% better than AEDCF, ADB, and EDCA, respectively. In addition, for high priority traffic, compared to ADB, AEDCF, and EDCA, I-EDCA shortens the average packet delay by 22%, 26%, and 40%, respectively. Therefore, I-EDCA is effective in supporting the QoS of IEEE 802.11e wireless LANs.

5.2 Future Work

We will dynamically adjust *Persistence Factor* (PF) after each unsuccessful transmission of packets according to the previous calculated *average collision rate* to further improve the overall performance. Besides, we will use different MAC parameters, such as CW_{min}, CW_{max}, and frame size, to enhance the I-EDCA.

Bibliography

[1] IEEE, Part II 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Reference Number ISO/IEC 8802 - 11:1999 (E), IEEE Std. 802.11, 1999 Edition, 1999.

[2] IEEE, Supplement to Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band, IEEE Std. 802.11a – 1999, 1999.

[3] IEEE, Supplement to Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-speed Physical Layer Extension in the 2.4 GHz Band, IEEE Std. 802.11b – 1999, 1999.

[4] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, pp. 116-126, Sept. 1997.

[5] H. Wu, S. Cheng, Y. Peng, K. Long, and J. Ma, "IEEE 802.11 Distributed Coordination Function (DCF): Analysis and Enhancement," in *Proc. of IEEE International Conference on Communications*, April 2002, vol. 1, pp. 605-609.

[6] IEEE 802.11 WG, Draft Supplement to Standard for Telecommunications and Information Exchange between Systems – LAN/MAN Specific Requirements - Part II: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e/D5.0, July 2003.

[7] C. Dongyan, G. Daqing, and Z. Jinyun, "Supporting Real-time Traffic with QoS in IEEE 802.11e Based Home Networks," in *Proc. of Consumer Communications and Networking Conference*, Jan. 2004, pp 205-209.

[8] X. Kenan, W. Quanhong, and H. Hassanein, "Performance Analysis of Differentiated QoS Supported by IEEE 802.11e Enhanced Distributed Coordination Function (EDCF) in WLAN," in *Proc. of IEEE GLOBECOM*, Dec. 2003, vol. 2, pp. 1048-1053.

[9] D. Chie, J. C. Yang, and Y. W. Jia, "The Performance Study of Contention-based Differentiation Mechanisms in IEEE 802.11e MAC Layer," in *Proc. of Personal, Indoor and Mobile Radio Communications Conference*, Sept. 2003, vol. 2, pp 1415-1419.

[10] C. Sunghyun, J. D. Prado, S. N. Sai, and S. Mangold, "IEEE 802.11e Contention-based Channel Access (EDCF) Performance Evaluation," in *Proc. of IEEE International Conference on Communications*, May 2003, vol. 2, pp. 1151-1156.

[11] S. Mangold, C. Sunghyun, G. R. Hiertz, O. Klein, and B. Walke, "Analysis of IEEE 802.11e for QoS Support in Wireless LANs," *IEEE Wireless Communications*, vol. 10, issue 6, pp. 40-50, Dec. 2003.

[12] X. Yang, "Enhanced DCF of IEEE 802.11e to Support QoS," in *Proc. of IEEE Wireless Communications and Networking Conference*, Mar. 2003, vol. 2, pp 1291-1296.

[13] P. Garg, R. Doshi, R. Greene, M. Baker, M. Malek, and X. Cheng, "Using IEEE 802.11e MAC for QoS over Wireless," in *Proc. of IEEE International Conference on Computing and Communications*, April 2003, pp. 537-542.

[14] D. Qiao, and K. G. Shin, "UMAV: A Simple Enhancement to the IEEE 802.11 DCF," in *Proc. of 36th Annual Hawaii International Conference on System Sciences*, Jan. 2003, pp. 306-314.

[15] K. Kim, A. Ahmad, and K. Kim, "A Wireless Multimedia LAN Architecture Using DCF with Shortened Contention Window for QoS Provisioning," *IEEE Communications Letters*, vol. 7, issue 2, pp. 97-99, Feb. 2003.

[16] Y. Peng, H. Wu, S. Cheng, and K. Long, "A New Self-adapt DCF Algorithm," in *Proc. of IEEE GLOBECOM*, Nov. 2002, vol. 1, pp. 87-91.

[17] C. Li, and T. Lin, "Fixed Collision Rate Back-off Algorithm for Wireless Access Networks," in *Proc. of Vehicular Technology Conference*, Sept. 2002, pp. 1212-1215.

[18] L. Romdhani, Q. Ni and T. Turletti, "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-hoc Networks," in *Proc. of IEEE Wireless Communications and Networking Conference*, Mar. 2003, vol. 2, pp 1373-1378.

[19] G. W. Wong and R. W. Donaldson, "Improving the QoS Performance of EDCF in IEEE 802.11e Wireless LANs," in *Proc. of Pacific Rim Conference on Communications, Computers and Signal*, Aug. 2003, pp. 392-396.

[20] C. Yunli, A. Z. Qing, and D. P. Agrawal, "Performance Analysis of IEEE 802.11e Enhanced Distributed Coordination Function," in *Proc. of 11th IEEE International Conference on Networks*, Oct. 2003, pp. 573-578.

[21] H. Dajiang and C. Q. Shen, "Simulation Study of IEEE 802.11e EDCF," in *Proc. of 57th IEEE Semiannual Conference on Vehicular Technology*, April. 2003, vol. 1, pp. 685-689.

[22] A. Lindgren, A. Almquist, and O. Schelen, "Quality of Service Schemes for IEEE 802.11 Wireless LANs – An Evaluation," in *Mobile Networks and Applications*, vol. 8, pp. 223-235, 2003.

[23] A. Grilo and M. Nunes, "Performance Evaluation of IEEE 802.11e," in *Proc. of 13th IEEE International Symposium Conference on Personal, Indoor and Mobile Radio Communications*, Sept. 2002, vol. 1, pp. 511-517.

[24] A. Lindgren, A. Almquist, and O. Schelen, "Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs," in *Proc. of 26th Annual IEEE Conference on Local Computer Networks*, Nov. 2001, pp. 348-351.

[25] VINT Group, "UCB/LBNL/VINT Network Simulator ns (version 2)." [Online]. Available: http://www.isi.edu/nsnam/ns.