國立交通大學

電機資訊學院 電信學程

碩士論文

應用於 802.11 無線區域網路的一個公平的遞延時間演算法

A Fair Backoff Algorithm for 802.11 WLAN

研 究 生：唐淑芬

指導教授：張仲儒 教授

中華民國九十三年六月

應用於 802.11 無線區域網路的一個公平的遞延時間演算法

A Fair Backoff Algorithm for 802.11 WLAN

研究生：唐淑芬　　Student：Athena Tang

指導教授：張仲儒　　Advisor：Chung-Ju Chang

國立交通大學

電機資訊學院 電信學程

碩士論文

A Thesis
Submitted to Degree Program of Electrical Engineering Computer Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Communication Engineering
June 2004
Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

應用於 802.11 無線區域網路的一個公平的遞延時間演算法

學生：唐淑芬　　　　　　　　　　　　指導教授：張仲儒 教授

國立交通大學電機資訊學院 電信學程（研究所）碩士班

摘要

由於今日無線區域網路的普及性, 有越來越多的人使用這種方便的傳輸媒介, 也因此導致無線區域網路中的資料量產生擁塞的現象. 最常見的情形是, 當資料量多的時候, 有很多封包會因為產生碰撞而必須長時間的等待可傳送的時間點, 這個現象使得無線網路中, 各種碰撞情況不同的封包的等待時間大不相同. 為了改善在現行的 IEEE 802.11 規範中, 採用的遞延時間演算法所導致的嚴重的封包延遲時間變異過大的問題, 本論文希望能夠研究出一個公平的遞延時間演算法, 可以將每個要傳送的封包的遞延時間的標準差控制在一定的容許時間內. 即使封包的碰撞次數增加, 也能夠在可預估的時間中傳送完成. 在文中提出的公平的遞延時間演算法, 先從網路的資料量估計存在於無線網路中的用戶端的多寡, 計算出在此種網路負載下相對應之 contention window 的大小, 以此當作 Gamma 函數的參數 a. 在每次的碰撞出現的時候, 依序更動 ß 的對應值, 期使該封包的遞延時間不會因為碰撞次數增加而呈現指數型的增加, 反而可以縮減下次遞延的時間, 藉此完成控制遞延時間標準差的目的. 在模擬的結果中, 可以發現無論網路負載的情況如何不同, 和傳統的遞延時間演算法相較, 公平的遞延時間演算法都可以將遞延時間標準差減少 70%, 這將有助於估計封包的可傳遞時間的準確度, 更同時達成每個封包傳遞的公平性. 在增加了封包的公平性之後, 通道的使用率及封包的平均遞延時間仍和傳統的遞延時間演算法下幾近相同.

A Fair Backoff Algorithm for 802.11 WLAN

Student：Athena Tang

Advisors：Prof. Chung-Ju Chang

Degree Program of Electrical Engineering Computer Science
National Chiao Tung University

ABSTRACT

When more and more people enjoy the convenience of the wireless LAN, we find the traffic loading will introduce the serious delay time of packets. Both the delay time and the standard deviation of delay time increase along with the traffic loading. This will induce the unfair packet transmission. According to the conventional backoff algorithm, a packet with more collisions will select a backoff timer from an exponential-expanded contention window based on the collision number, which means getting a larger backoff more possible if the packet is collided more. We wish to specify a fair backoff algorithm to balance all packet transmissions within a predictable delay time. The standard deviation of delay time should be as small as possible to achieve this goal. Instead of using the Uniform distribution, the Gamma distribution is the function we use in the fair backoff algorithm. The two parameters of Gamma function come from the measurement of the traffic loading and the collision number of that packet. From the simulation results, we can recognize that the standard deviation can be reduced 70% if comparing with the conventional backoff algorithm under any kind of traffic loading. We can enhance the precision of delay time without degrading the channel utilization or the average delay time.

誌謝

要感謝的人真的是太多了!

首先，真的很感激老師對我的幫助．因為工作的關係，常常無法很專心快速的完成每個階段的進度，而老師總是體諒我的情況，要我慢慢來好好做，不厭其煩的指導我把每個細節了解透徹．

也謝謝我的家人，朋友．在我課業工作兩頭奔波的時候，無怨尤的支持我，給予我最大的精神鼓勵，這個力量，撐持著我每每感到疲憊的心情．因為你們，我才能通過這些困難，希望這份喜悅能和你們共享 ！

目　錄

表目錄

圖　目　錄

Chapter 1

Introduction

Recently, as we know, the wireless LAN has been getting more popular in the world and actually a lot of people have enjoyed this convenient technology. The IEEE 802.11 standard [1] is the first international standard for wireless local area network, which defines a medium access control (MAC) sub-layer, MAC management protocols and services, and three low-speed physical layers (PHYs) which are known as IR (Infrared), DSSS (Direct Sequence Spread-Spectrum) and FHSS (Frequency Hopping Spread-Spectrum), operating at both 1Mbps and 2Mbps. Two new high-speed physical layers, the IEEE 802.11b [2] and IEEE 802.11a [3], have been developed to speed up the existing IEEE 802.11 standard to 11Mbps and 54Mbps, respectively. And, they operate at two different frequency bands, the 2.4GHz for the IEEE 802.11b and the 5GHz for the IEEE 802.11a. The 2.4GHz band is Industrial, Scientific, and Medical (ISM) band, and it’s a free air resource to use in most countries; however, the 5GHz is not so opened in everywhere. It depends on the public policy in that country. For the new coming standard IEEE 802.11g [4], it operates at the same frequency band of the IEEE 802.11b [2] but can support up to 54Mbps as the IEEE 802.11a [3] does.

Two different physical-layer coding specifications are provided for these three new physical (PHY) standards. The IEEE 802.11b [2] adopts Complementary Code Keying (CCK) as its physical layer specification because it easily provides a path for interoperability with the existing IEEE 802.11 1Mbps and 2Mbps systems, but the IEEE 802.11a [3] uses an alternative coding, Orthogonal Frequency Division Multiplexing (OFDM), which provides eight different PHY modes with data rates ranging from 6Mbps up to 54Mbps. As to the IEEE 802.11g [4], it uses both CCK and OFDM as its PHY layers. 12 data rates are supported in the IEEE 802.11g, which include ERP-DSSS with 1Mbps/2Mbps, ERP-CCK with 5.5Mbps/11Mbps, ERP-OFDM with 6Mbps/9Mbps/12Mbps/18Mbps/24Mbps/36Mbps/48Mbps/54Mbps, ERP-PBCC with 5.5Mbps/11Mbps/22Mbps/33Mbps, and DSSS-OFDM with the same rates as those of ERP-OFDM. It's easy to find that the IEEE 802.11g is backward compatible with the 2.4GHz 802.11b-CCK and the 802.11a-OFDM in the 5GHz band.

Furthermore, the IEEE 802.11e committee is currently working on the enhancement of the 802.11 MAC to expand the support for applications with QoS (*Quality of Service*) requirements [5]. WLANs have a potential to support the mobility of the clients and ubiquitous access to information by linking together heterogeneous wireless devices to the wired infrastructure network; they can also be used in a stand-alone ad hoc network setting as rapidly deployable networks for applications such as military operations, rescue mission, and law enforcement. Unlike the character of the well-known Ethernet MAC of the IEEE 802.3 [6], WLANs require a special MAC layer protocol due to the broadcast nature of radio medium. This nature provides a simple access medium without any installation or construction, also a hard method to negotiate the air channels between different devices. Basically, in case of no hidden terminals, all transmissions between the stations (STAs) in a Basic Service Set (BSS) are broadcasts, and should conceptually be heard by all other STAs in the same BSS.

Three different types of frames are defined in WLANs: management, control and data. The management frames are used for synchronization, authentication, deauthentication, association, reassociation, disassociation and timing. The control frames include polling, handshaking packets, positive acknowledgements and PCF/DCF period control. Data frames handed down from the upper LLC to the MAC may be fragmented if they exceed a threshold value set by *Fragmentation_Threshold*. That will make each fragment MAC Protocol Data Unit (MPDU) formed from the fragmented MAC Service Data Unit (MSDU), other than the last one, is of length *Fragmentation_Threshold* [1].

At present, the IEEE 802.11 standard MAC protocol supports two kinds of access methods: distributed coordination function (DCF) and point coordination function (PCF) [1]. In the mean time, the IEEE 802.11e draft specification introduces two new modes of operation: Enhanced DCF (EDCF) and Hybrid Coordination Function (HCF), which define mechanisms to enable QoS function, in corresponding to DCF and PCF [5]. The DCF supports best-effort delivery of packets and must be implemented in all stations, while the PCF is built on the top of the DCF and provides services for time-bounded traffic [7]. When both DCF and PCF are used, the IEEE 802.11 standard MAC is a hybrid protocol of random access and polling. In this case, a wireless channel is divided into a superframe format. Each superframe consists of a Contention Free Period (CFP) for the PCF stations and a Contention Period (CP) for the DCF stations [9].

The DCF is based on a multiple access spread spectrum scheme called Carrier

Sense Multiple Access with Collision Avoidance (CSMA/CA); the protocol desires to avoid collisions instead of detecting the collisions as used in the IEEE 802.3 LAN [6]. All stations in DCF mode have equal priorities and hence an equal chance of getting the idle medium when no station occupies that. Because in a wireless environment, a station (STA) may not be able to transmit and listen to the channel at the same time, this scenario increases the cost of a collision in the system since a STA cannot stop transmitting even the packet is collided by any interference from others, unless it has transmitted the entire frame out. The performance of the DCF has already been studied by many researches [7], [8] and [10]. In [10], the author proposes a different idea to support service differentiation in wireless packet network using a fully distributed approach that supports service differentiation, radio monitoring, and admission control. After analyzing the delay experienced by a mobile host implementing the IEEE 802.11 DCF and derives a closed-form formula, it can extend the DCF to provide service differentiation for delay-sensitive and best-effort traffic based on the results from the analysis.

Also, there is a greater vulnerability to collisions due to the presence of hidden terminals in the system as two stations hidden from each other may concurrently try to send data to another station which is not hidden from either of them. In order to avoid collision of packets due to hidden terminals, Request-To-Send (RTS) and Clear-To-Send (CTS) control packets are exchanged between the sender and the receiver to reserve the medium prior to the transmission of data packets when these packet sizes are longer than *RTS_Threshold*. The RTS/CTS mechanism is a four-way handshaking technique (RTS-CTS-DATA-ACK) where the sender probes whether the receiver is ready to receive a data packet by sending a RTS frame first to the aimed receiver and has to receive a CTS packet back from the addressed STA in the RTS packet before it can start the transmission [11]. The known destination will generate the CTS packet responsed to the request and heard by all hidden terminals after receiving a RTS packet if there is no collision. All other hidden terminals will recognize there exists a hidden terminal while hearing the CTS from the known station, and reduce the possibility of collision in advanced. However, the RTS control packets as well as data packets with packet length smaller than the *RTS_threshold* can still generate collisions [7]. Previous studies [12] have shown that the probability of these collisions occurring increases with the number of hidden terminal pairs. And, the throughput of the IEEE 802.11 DCF scheme with hidden node problems in multi-hop ad hoc networks based on the assumption that the carrier sense range is equal to the transmission range is analyzed in [13].

In PCF mode, the polling scheme occurs with a point coordinator (PC) determining which station has the right to seize the medium. On the other hand, the PCF is intended for the transmission of real-time traffics in a round-robin manner as well as that of asynchronous data traffic in each contention free period. This access method is optional and is based on the polling method controlled by an Access Point (AP). Many researches [9], [14] and [15] have pointed out that the use of centralized scheme in PCF constrains the operation of WLAN and results in a worse performance especially for the real-time multi-medium packets. In [9], the combined performance considering multimedia wireless terminals transmit voice, video and data with PCF operation of the IEEE 802.11 standard MAC protocol as best-effort traffics is evaluated. It focuses on the scheduling problem for multimedia transmission with the PCF of the MAC protocol and proposes two priority-based schemes to improve the performance of the multimedia packets. According to the simulation result of [14], it shows that if the CFP repetition interval is set too long, PCF performance deteriorates drastically. How channel transmission errors affect the performance of the protocol is also described detailed. A new decentralized control mechanism is proposed in [15]. It suppresses delay fluctuation in CSMA/CA networks, called DDFC (Decentralized Delay Fluctuation Control).

The draft IEEE 802.11e specifies EDCF and HCF modes to serve differentiated control of accessing to the medium with eight differing priorities. The EDCF still follows the same CSMA/CA architecture of the IEEE 802.11 and the exponential backoff mechanism to access the wireless medium, but is enhanced by the Traffic Categories (TCs) that map directly to the RSVP protocol and other protocol priority levels [16]. Up to eight TCs within one station is proposed in this draft specification, and each one presents a different priority defined by assigning different values of the EDCF access parameters, like the minimum contention window, the maximum contention window, and the arbitration interframe space [17]. As the PCF function in the IEEE 802.11, the HCF polls all authenticated STAs and gives one of them the permission to access the channel during the CFP, that starts with every beacon, of a superframe. During this period, the Hybrid Coordinator (HC) issues a QoS CF-Poll to a particular STA to give it a Transmission Opportunity (TXOP), specifying the start time and maximum duration. The CFP ends after the time announced by the beacon frame or by a CF-END Frame. The second section of a superframe is the CP. During the CP, the access is governed by the EDCF, though the Hybrid Coordinator can initiate a HCF access at any time [16].

We can easily predict that how heavy the wireless traffic loading will be in the near future, especially in a crowded area. Even with any quality of service guarantee embedded in every station, there will be equal to no quality assurance when everyone gains the highest priority at the same time. How to control the quality of each wireless connection in case every station has the same priority? How to maintain the least requirement of throughput in the wireless environment? Those can help all joined stations to foresee the available bandwidth they probably are able to get before they start their transmission. It also provides some kind of quality of service.

Many researches have been proposed a lot of new methods to differentiate the serviced levels [18] – [23]. More complicated logics and higher cost are required to implement those new architectures but only gain less efficiency improvement. Some analysis [24], [25] focus on reducing the interference and prove that are also helpful in the enhancement of the performance. The objective of this thesis is to investigate the throughput and the packet fairness of DCF mode in a BSS infrastructure with a simple modified backoff algorithm. The throughput is defined as the time of all successful transmitted data over the time used to transmit these data. Higher throughput means the station can transmit data out more efficient because the channel is utilized most. The packet fairness is presented by the characteristics of delay time. Traditionally, the delay time is longer when the packet is collided more. This proposed backoff mechanism should control the delay time of each packet as almost the same as possible, no matter collided or non-collided. Then, we define the fairness of each packet as a better fairness.

Before we analyze how to achieve a higher throughput and better fairness methodology, an overview of the current standard IEEE 802.11 is introduced in chapter 2. Carrier-Sense Multiple Access with Collision Avoidance mechanism is introduced first because it is the fundamental algorithm in a wireless environment. As mentioned a lot in other researches, we briefly describe what the RTS/CTS control packets work for, and how they can resolve the hidden terminals problem. We also analyze how the conventional exponential backoff algorithm operates. We are very interested in the performance of the recommended backoff algorithm in the IEEE 802.11 and discuss this procedure more detailed in this chapter. The throughput and packet fairness are shown in couple figures. Simulation results of average delay time distribution of different collision numbers and the delay time distributions are shown in this chapter too. We can figure out why the performance of the DCF mode in the standard IEEE 802.11 works so bad and how hard it is to control the delay time of each packet.

Refer to the modified backoff algorithm proposed in the chapter 3, we can compare the simulation results between the conventional exponential backoff algorithm and the new proposed algorithm, and see how the new one works to improve the packet fairness without affecting the throughput which is also can be viewed as the channel utilization improvement.

Finally, the concluding remarks are made in the chapter 4.

Chapter 2

Overview of IEEE 802.11

In recent years, much interest has been involved in the design of wireless networks for local area communications. The standard IEEE 802.11 [1] provides a detailed medium access control (MAC) and physical layer (PHY) specifications for WLAN. The standard has a hierarchical architecture with the fundamental building block known as the Basic Service Set (BSS). A set of stations residing in a certain geographical area, which is known as the Basic Service Area (BSA), communicating with each other using the same coordination function (DCF or PCF) form a BSS.

The prime access method of the IEEE 802.11 MAC [1] is called Distributed Coordination Function (DCF), which is known as a *Carrier-Sense Multiple Access with Collision Avoidance* (CSMA/CA) scheme with a random selection of exponential backoff algorithm. This protocol is designed to reduce the collision probability between multiple stations accessing the same medium, at the point where the collisions would most likely occur. Before a station initiates a transmission, it shall sense the wireless medium to determine if any station is transmitting that time. If the medium is not occupied by others, the STA can start its transmission from the

RTS/CTS handshaking procedure as known to avoid the hidden terminal issues; if the medium is determined as busy, the STA should follow the backoff procedure to wait for a period and sense the medium again. The DCF is mandatory, and shall be implemented in all STAs, for using within both ad hoc and infrastructure network configurations. In addition, all directed traffic uses an immediate positive acknowledgement (ACK Frame), which the retransmission is scheduled by the sender if no ACK is received during a specified duration.

In this chapter, we will study the fundamental mechanism of DCF in Section 2.1. After understanding how the carrier-sense operates, the detailed of collision avoidance of DCF is introduced in Section 2.2. Since in this thesis, the hidden terminal problems are viewed as ignorable, the RTS/CTS procedure is showed in the Section 2.3 and we can understand it’s reasonable to neglect this issue. As to the exponential backoff algorithm and the simulation results are presented in the Section 2.4 and Section 2.5. We can recognize how the performance will be along with the traffic loadings.

2.1 Carrier-Sense Mechanism

Both physical and virtual carrier-sense functions are used to determine the state of the medium. When either function indicates a busy medium, the medium shall be considered busy; otherwise, it shall be considered idle. The physical carrier-sense mechanism that is well known as *Clear Channel Assessment* (CCA) shall be provided by the PHY. And the virtual carrier-sense mechanism that is referred as the *Network Allocation Vector* (NAV) shall be provided by the MAC.

2.1.1 Physical Carrier-Sense Mechanism

The architecture of the entire frame transmitted out from the PHY is like the Figure 1, and it includes two main portions: the former is the Physical Layer Convergence Protocol (PLCP) and the latter is the MAC Protocol Data Unit (MPDU).

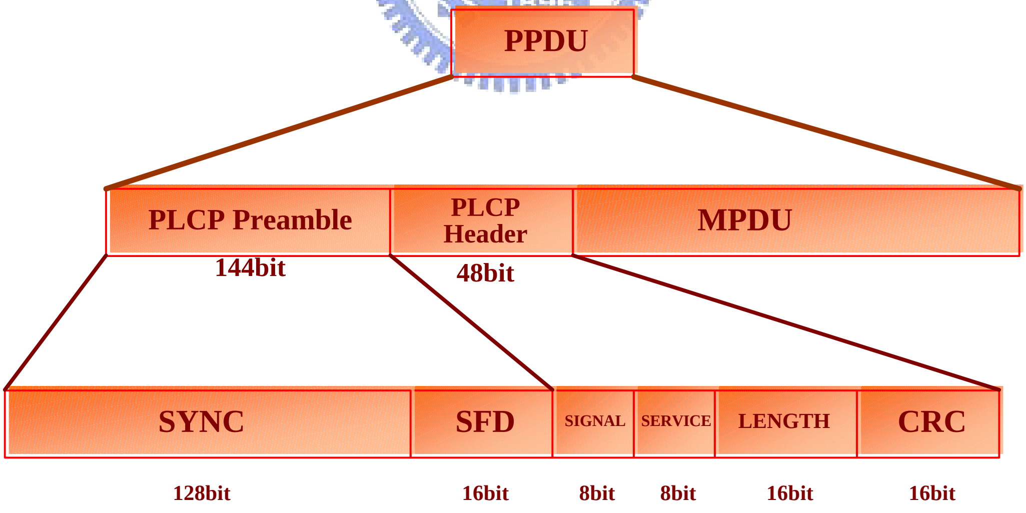

Figure 1. PLCP Frame Format

The DSSS PHY shall provide the capability to perform CCA mechanism according to at least one of these three methods, and generate a primitive as

PHYCCA.indicate(STATE) showing the state of the medium:

- CCA mode 1: Energy detect above a specified threshold;
- CCA mode 2: Carrier sense only;
- CCA mode 3: A combination with carrier sense and energy detect above a specified threshold.

The STATE parameter can be one of these two values: BUSY, IDLE. The parameter value shall be BUSY if the channel assessment controlled by the PHY sublayer results in the medium not being available.

2.1.2 Virtual Carrier-Sense Mechanism

The virtual carrier-sense mechanism is achieved by distributing the reservation information announcing the impending use of the medium. One means of the distribution of the medium reservation information is that the NAV maintains a prediction of future traffic loading on the medium based on the duration information that is announced in both RTS/CTS frames prior to the actual transmission of data. The Figure 2 and Figure 3 are the contents of RTS and CTS frames individually.

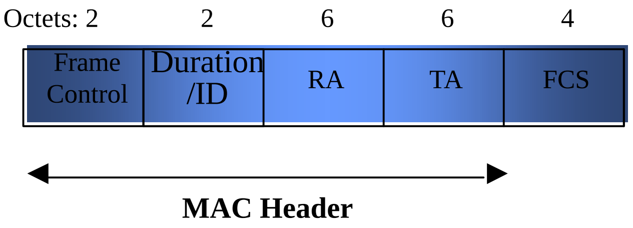

Figure 2. RTS Frame

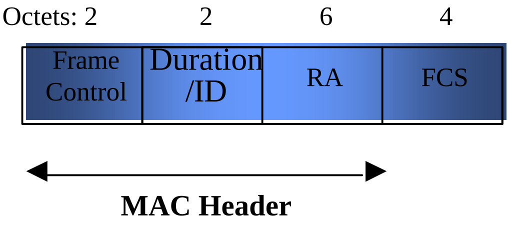

Figure 3. CTS Frame

Another means of the virtual carrier-sense mechanism is from the normal data frame if the STA just joins the BSS and cannot hear the RTS/CTS packets. The Figure 4 is the frame format of a MPDU.

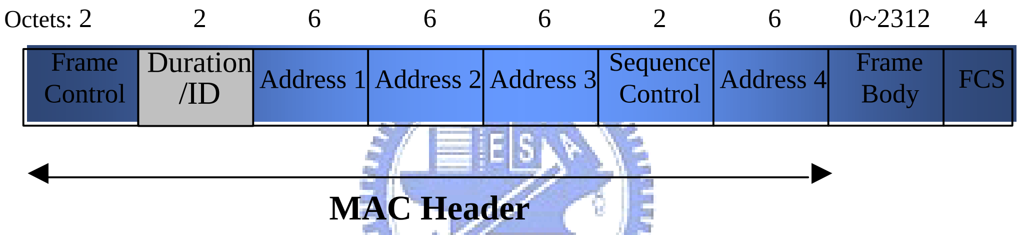

Figure 4. MPDU Frame Format

The Table I. describes the meanings of each subfield included in the Duration/ID field of the MAC header. The Duration/ID field is 16 bits in length and the contents of this field are as follows:

- In the control type frames of subtype Power Save (PS)-Poll, the Duration/ID field carries the Association IDentity (AID) of the station that transmitted the frame in the 14 Least Significant Bits (LSB), with the 2 Most Significant Bits (MSB) both set to 1. The value of the AID is in the range of 1 – 2007.
- In all other frames, the Duration/ID field contains a duration value as defined for each frame type used in management, control and data. For

frames transmitted during the Contention-Free Period (CFP), the duration field is set to 32768.

TABLE I. DURATION / ID FIELD ENCODING

Bit 15	Bit 14	Bit 13-0	Usage
0	0 ~ 32767		Duration
1	0	0	Fixed value within frames transmitted during the CFP
1	0	1 ~ 16383	Reserved
1	1	0	Reserved
1	1	1 ~ 2007	AID in PS-Poll frames
1	1	2008 ~ 16383	Reserved

Whenever the contents of the Duration/ID field are less than 32768, the duration value is used to update the network allocation vector (NAV) according to the procedures

Whenever the STAs receives a valid frame of any type listed as above, they shall update the NAV value with the information filled in the Duration/ID field, but only when the new NAV value is greater than the current NAV value of this STA and only when the frame is not addressed to the receiving STA. The Figure 5 indicates the virtual carrier-sense procedure. Once a transmitting STA initiates a RTS frame, the NAV value for other STAs that may receive the RTS frame from the transmitting STA will be updated to defer the access time; while other STAs may only receive the CTS frame responded by the addressed STA, results in the lower NAV bar as shown. (With the exception of the STA to which the RTS frame was addressed). If the STA already contains a non-zero NAV, it will update its NAV value only if the announcing NAV value is larger than the existing one; otherwise it will keep the original NAV value.

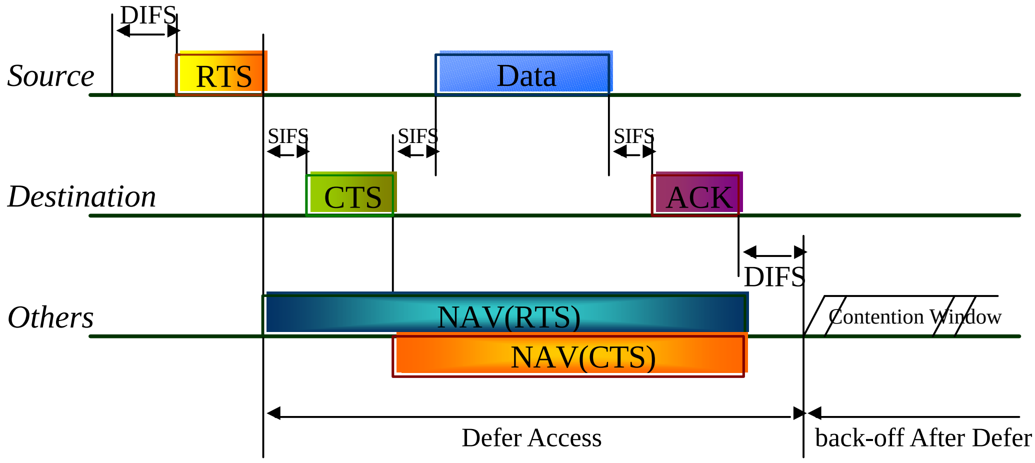

Figure5. RTS/CTS/Data/ACK and NAV setting

2.2 Collision-Avoidance Mechanism

The CSMA/CA distributed algorithm is mandated that a gap of a minimum specified duration exists between the contiguous frame sequences. A transmitting STA shall ensure that the medium is idle for this required duration before attempting to transmit the next frame. If the medium is determined to be busy by either physical or virtual carrier-sense mechanism, the STA shall defer a period and listen to the channel until the end of the current transmission. After the deferral, or prior to attempting to transmit packet again immediately after a successful transmission, the STA shall select a random backoff time interval and shall decrease the backoff time counter as long as the medium goes to idle state.

The time interval between the continuous two frames is called the Inter-Frame Space (IFS). Four different IFSs are defined to provide different priorities for accessing to the wireless medium in the IEEE 802.11 [1]. The four IFSs are listed in order here, from the shortest one to the longest one, that are Short IFS (SIFS), PCF IFS (PIFS), DCF IFS (DIFS), and Extended IFS (EIFS). These different IFSs shall be independent of the STA bit rate. Each IFS timing shall be defined as time gaps on the medium, and shall be fixed for each PHY (even in multi-rate-capable PHYs).

The basic time slot is the *aSlotTime*, and every counting step is based on the time slot. The *aSlotTime* for the DSSS PHY shall be the sum of the Rx-to-Tx turnaround time (5μs), measured at the MAC/PHY interface, and the energy detecting time (15μs). The propagation delay shall be regarded as being included in the energy detecting time.

The SIFS shall be used for an ACK frame, a CTS frame, the second or the subsequent MPDU frames of a fragment burst, and by a STA responding to any polling from the PCF mode. The SIFS is the time interval from the end of the last symbol of the previous frame to the beginning of the first symbol of the preamble of the subsequent frame as seen at the air interface. The IEEE 802.11 defines the SIFS time to be:

$$\begin{aligned} \boldsymbol{aSIFSTime} \; ? \; \boldsymbol{aRxRFDelay} \; ? \; \boldsymbol{aRxPLCPDelay} & \\ ? \; \boldsymbol{aMACProcessingDelay} \; ? \; \boldsymbol{aRxTxTurnaroundTime} & \qquad ?2.1? \end{aligned}$$

And its implementation shall not allow the tolerance between two contiguous frames that are defined to be separated by a SIFS time, as measured on the medium, to vary from the nominal SIFS value by more than ?10% of the *aSlotTime* for the PHY in use.

The PIFS shall be used only by the STAs operating under the PCF mode to gain the priority access to the medium at the start of the CFP in a superframe. According to the definition, the PIFS duration is:

$$\boldsymbol{aPIFSTime} \; ? \; \boldsymbol{aSIFSTime} \; ? \; \boldsymbol{aSlotTime} \qquad ?2.2?$$

A STA using the PCF shall be allowed to transmit the contention-free traffics after its carrier-sense mechanism determines that the medium is idle for at least *aPIFSTime*.

The DIFS shall be used by the STAs operating under the DCF mode to transmit the data frames (MPDUs) and the management frames (MMPDUs). The *aDIFSTime* is defined as:

$$\textbf{\textit{aDIFSTime}} ? \textbf{\textit{aSIFSTime}} ? 2 ? \textbf{\textit{aSlotTime}} \qquad ?2.3?$$

A STA using the DCF shall not transmit within an EIFS period after it determines that the medium is idle following the reception of a frame for which PHYRXEND.indication primitive contained an error or a frame for which the MAC FCS value was not correct. A STA may transmit after subsequent reception of an error-free frame, re-synchronizing the STA. This allows the STA to transmit using the DIFS following that frame.

The EIFS shall be used by the DCF whenever the PHY has indicated by the MAC that a frame transmission was begun that did not result in the correct reception of a complete MAC frame with a correct FCS value. The duration of an EIFS is:

$$\begin{aligned} \textbf{\textit{aEIFSTime}} &? \textbf{\textit{aSIFSTime}} ? \textbf{\textit{aDISFTime}} \\ &? 8 ? \textbf{\textit{ACKSize}} ? \textbf{\textit{aPreambleLength}} ? \textbf{\textit{aPLCPHeaderLength}} \end{aligned} \qquad ?2.4?$$

where, ACKSize is the length, in bytes, of an ACK frame; and

(8xACKSize + aPreambleLength + aPLCPHeaderLength) is expressed in microseconds required to transmit at the PHY’s lowest mandatory rate.

The EIFS is defined to provide enough time for another STA to acknowledge what was, to this STA, an incorrectly received frame before this STA commences transmission. Reception of an error-free frame during the EIFS resynchronizes the STA to the actual busy/idle state of the medium, so the EIFS is terminated and the normal medium access (using DIFS and, if necessary, backoff algorithm) continues following reception of that frame.

The basic access mechanism is illustrated in the Figure 6. A STA with a new packet to transmit shall monitor the channel activity through both physical carrier-sense mechanism CCA and virtual carrier-sense mechanism NAV. If the channel is determined to be idle for a period of time equal to the DIFS, the STA transmits the packet immediately. Otherwise, if the channel is sensed as busy (either immediately or during the DIFS), the STA persists to monitor the channel until it is measured to be idle for a DIFS period. At this point, the STA generates a random backoff time interval before transmitting to minimize the probability of collision with packets being transmitted by other stations. In addition, to avoid channel capture, a STA must wait a random backoff time between two consecutive new packet transmissions, even if the medium is sensed idle after a DIFS time.

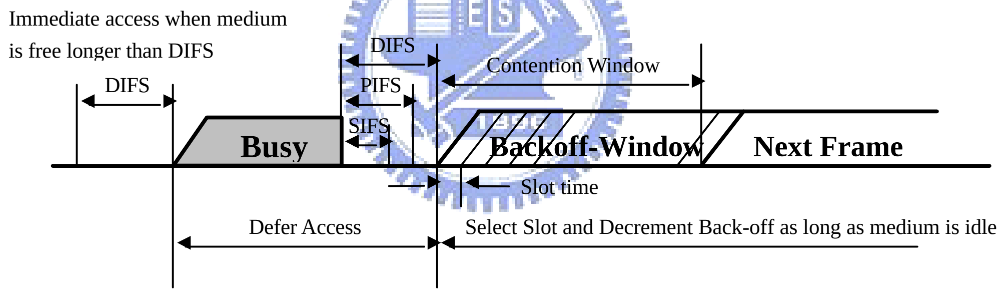

Figure 6. Basic Access Method in DCF

2.3 RTS/CTS Mechanism

There is a high penalty for collisions between long data frames. As a station can not hear the collisions, it continues the transmission of a frame even when there are simultaneous transmissions. This results in wastage of bandwidth. A refinement of the method may be used under various circumstances to further minimize collisions – here the transmitting and receiving STAs exchange two short control frames [Request To Send (RTS) and Clear To Send (CTS) frames] after determining that the medium is idle and after any deferral or backoff, prior to the real data transmission.

The virtual carrier-sense mechanism is achieved by distributing the reservation information in the MAC header to announce the impending use of the medium. All STAs within the reception range of either the originating STA (which transmits the RTS frame) or the destination STA (which transmits the CTS frame) shall learn of the medium reservation information. Thus a STA, which is even unable to receive any packet from the originating STA, yet still knows about the impending use of the medium and will not transmit any data frame to cause a collision when the destination STA sends the CTS frame.

The RTS/CTS mechanism may also improve operation in a typical situation where all STAs can receive from the AP, but cannot receive from all other STAs in the BSA. Take the Figure 7 as an example. When STA1 wants to transmit the data packets which are longer than the threshold value *RTS_Threshold* after the specified period, it should send the RTS packet to the AP priori to the real data transmission. With the returned CTS packet from the AP, both STA1 and STA2 will learn the medium reservation information they have to know. For STA1, it will know it can transmit data, but for STA2, it should know the medium is busy even it wants to transmit data at that time and defers the transmission according to the backoff procedure until the medium is idle later.

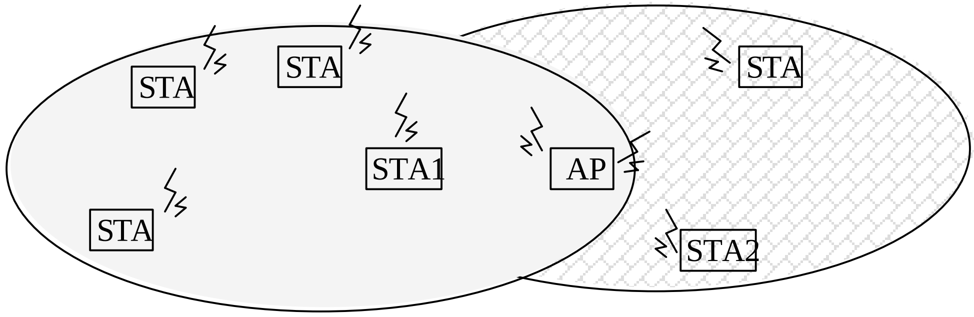

Figure 7. Hidden Terminal

The RTS/CTS mechanism cannot be used for MPDUs with either broadcast or multicast immediate address because there will be multiple destinations for the RTS, and thus potentially multiple concurrent senders of the CTS in response. The RTS/CTS mechanism need not be used for every data frame transmission. Because the additional RTS and CTS frames cause overhead and make the channel inefficiency. A STA configured not to initiate the RTS/CTS mechanism shall still update its virtual carrier-sense mechanism with the duration information contained in a received RTS or CTS frame, and shall always respond to an RTS addressed to it with a CTS frame.

2.4 Backoff Algorithm

A STA desiring to initiate a transfer of the data MPDUs and/or management packet MMPDUs shall invoke both the physical and virtual carrier-sense mechanisms to determine the busy or idle condition of the medium. If the medium is sensed as busy, the STA shall defer until the medium is determined to be idle without any interruption for one period of time equal to DIFS when the last frame detected on the medium was received correctly, or after the medium is determined to be idle without interruption for one period of time equal to EIFS when the last frame detected on the medium was not received correctly. After this required DIFS or EIFS medium idle time, the STA shall then excuse the backoff procedure and generate a random exponential period for an additional deferral time before the real packet transmission, unless the backoff timer already contains a nonzero value, in which case the selection of a new random number is not necessary and actually not performed. This process minimizes the possible collisions during a long contention between multiple STAs that have been deferring to the same event.

As recommended in the IEEE 802.11, the method to choose a backoff timer is:

$$\textbf{\textit{Backoff Time}} \; ? \; \textbf{\textit{Random}}?\,?? \, \textbf{\textit{aSlotTime}} \qquad ?2.5?$$

where

Random( ) = Pseudorandom integer drawn from a uniform distribution over the interval [0,CW], where CW is an integer within the range of values of the PHY characteristics *aCWmin* and *aCWmax*, *aCWmin*≦CW≦*aCWmax*. It is important that all designers should recognize the need for statistical independence among the random number streams among STAs.

aSlotTime = The value of the correspondingly named PHY characteristic.

The Contention Window (CW) parameter shall take an initial value from the minimum contention window size, *aCWmin*, according to the IEEE 802.11 recommendation. Every STA shall maintain a STA Short Retry Count (SSRC) as well as a STA Long Retry Count (SLRC), both of which shall take an initial value of zero. The SSRC shall be incremented whenever any short retry count associated with any MSDU frame is incremented. The SLRC shall be incremented whenever any long retry count associated with any MSDU frame is incremented. The CW shall take the next value in the series every time an unsuccessful attempt to deliver a MPDU packet. Once it reaches the maximum contention window size, *aCWmax*, the CW shall remain at the value of *aCWmax* until it is reset when the collided packet is sent out. This can improve the stability of the access protocol under high loading conditions.

The backoff procedure shall be invoked for a STA to transfer a frame when the STA finding the medium busy as indicated by either the physical or virtual carrier-sense mechanism. The backoff procedure shall also be invoked when a transmitting STA infers a failed transmission from the non-response of the ACK frame. To begin the backoff procedure, the STA shall set its Backoff Timer to a random backoff time using the equation in (2.5).

A STA performing the backoff procedure shall use the carrier-sense mechanisms to sense the medium and to determine whether there is any channel activity during each backoff time slot. If no medium activity is indicated for the duration of a particular backoff time slot, then the backoff procedure which the STA operates shall decrease its backoff timer once an *aSlotTime*. If the medium is determined to be busy at any time during the backoff duration, then the STA will suspend the backoff procedure; that is the backoff timer shall not decrease for that time slot. The medium shall be determined to be idle for the duration of DIFS or EIFS before the backoff procedure is allowed to resume if it is suspended by the interruption of the busy medium. All transmissions shall commence again whenever the Backoff Timer reaches zero.

The Figure 8 intorduces the procedure when the backoff algorithm is performed by the STAs under the IEEE 802.11 [1].

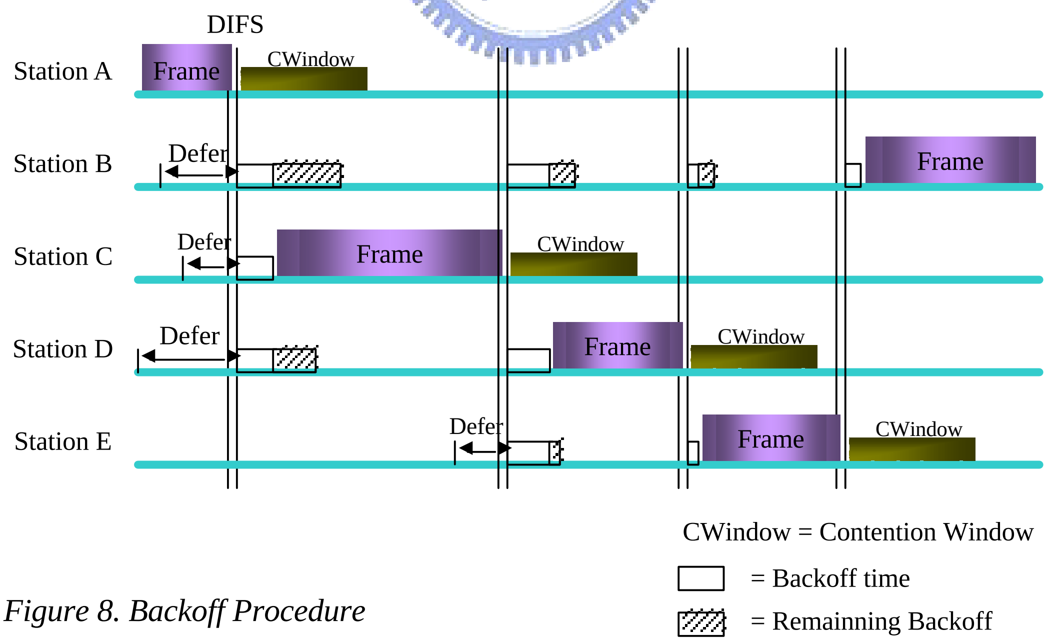

Figure 8. Backoff Procedure

While the STA A occupies the medium to transmit the data frames, the STA B, STAC and STA D desire to initiate a data transmission during this period and sense the condition of the medium as busy. All of them will defer until both the physical and virtual carrier-sense mechanisms indicate that the medium is determined as idle for the duration equal to one DIFS. At that point, every STA processes the backoff algorithm right away, and selects the backoff timer individually. The STA with the smallest backoff timer will count down its counter to zero first, and starts its transmission immediately, like the STA C in the Figure 8. All other STAs should suspend the decrease of the backoff timer and keep the remainder until the medium is determined as idle for one DIFS duration, as the STAB and STA C show. Following the idle time, all STAs can resume the count down procedure. Any STA with the backoff timer reaching to zero earlier will get the chance to launch the transmission.

In the case of the successful acknowledged transmissions, this backoff procedure shall begin at the end of the received ACK frame. In the case of an unsuccessful transmission requiring an acknowledgement, this backoff procedure shall begin at the end of the ACK timeout interval right away. If the transmission is successful, the CW value reverts to the initial value *aCWmin* before the next random backoff interval is chosen, and the STA short retry count and/or the STA long retry count are updated. This assures that the transmitted frames from a STA are always separated by at least one backoff time slot. The advantage of this procedure is that when multiple STAs are deferring and going into the random backoff state, the STA selecting the smallest backoff time value using the random function will win the contention and gain the right to access the free medium.

2.5 Simulation Results

According to the specification of backoff procedure recommended in the IEEE 802.11 [1], we can estimate the performance of each packet will be impacted seriously because of the expansion method of the backoff selection window especially when the environment has heavy traffics.

We concentrate on the performance of the DCF architecture. Assuming there exists some STAs along with one AP, and they construct a BSS environment. Using the RTS/CTS handshaking mechanism prior to the real data packets exchange to emulate the complete IEEE 802.11 standard [1]. The ACK frame is assumed to be received correctly by the sender without any exception after each data transmission is completed. Here, the hidden-terminal problems will be left for future discussion, and we don't put this issue into consideration because we already implement an ideal RTS/CTS algorithm to prevent this problem. The difference of air propagation delay, dependent on the characteristics of the PHY, can be ignored too, which means the real collisions will happen only when there are more than two STAs initiating to transmit packets at the same time and all of them don't figure out the medium is busy until they finish their transmission without any positive ACK frame receiving. Based on the CSMA/CA mechanism of the IEEE 802.11 MAC, we can assume that the probability of this condition is zero since the STA follows the collision avoidance scheme. Under these assumptions, we emulate an environment by every packet is with the same length of frame body, no hidden-terminal exists, and every possible collision will be sensed by each STA immediately and then get into backoff procedure without destroying the data neither wasting the channel utilization.

The Figure 9 shows the timing required for the normal data transmission. For a single packet transmission, the medium shall be sensed as idle again by all STAs after:

$$\boldsymbol{Time} : 2 ? \boldsymbol{aDIFStime} ? 3 ? \boldsymbol{aSIFStime} ? \boldsymbol{aRTStime} ? \boldsymbol{aCTStime} ? \boldsymbol{aACKtime} ? \boldsymbol{Data} \quad ?2.6?$$

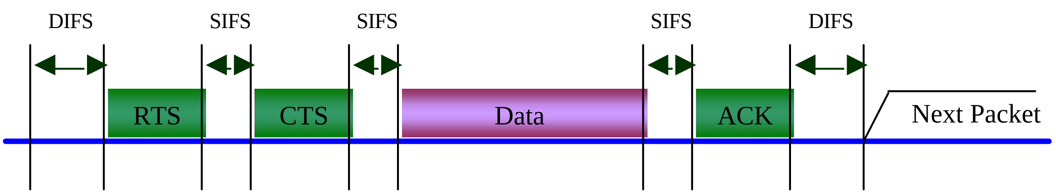

Figure 9. Data Transmission Timing Required

The slot time *aSlotTime* and the short inter-frame space *aSIFSTim*e are assumed to be fixed per PHY, and the value of *aSlotTime* is 20μs defined in the IEEE 802.11. (Refer to 15.4.6.8 Slot time). Based on the requirement defined in the IEEE 802.11, the *aSIFSTime*, as measured on the medium, is not allowed to vary from the nominal SIFS value by more than 10% of one *aSlotTime* for the PHY in use. 5μs is recommended for the *aSIFSTime* value in the standard and we can ignore the propagation delay compared with the *aSIFSTime*. The durations of PIFS and DIFS are derived by the following equations specified in the IEEE 802.11:

$$\boldsymbol{aPIFSTime} ? \boldsymbol{aSIFSTime} ? \boldsymbol{aSlotTime} \quad ?2.7?$$

$$\boldsymbol{aDIFSTime} ? \boldsymbol{aSIFSTime} ? 2 ? \boldsymbol{aSlotTime} \quad ?2.8?$$

In our simulation case, the retransmission is ignored. That means no collision will happen or degrade the throughput of the medium, and every transmission is successful with an ACK frame is received by the transmitting STA.

The throughput is defined as the average data transfer rate. It can be also taken for the utilization of the wireless channel, but different dealing with the dummy

RTS/CTS/ACK frames. The medium which is determined to be not idle and used for the real data exchange is the transmission time we put into our calculation. Those auxiliary packets, like RTS, CTS or ACK frames, used to complete the real wireless environment are not taken into consideration. But they do exist in the simulation model. We also have to put the inter-frame space (IFS) into consideration, like the *aDIFStime* and the *aSIFStime* in the DCF mode. Based on this assumption, we can formulate the throughput rule as:

$$\boldsymbol{Throughput} \; ? \; \frac{\boldsymbol{Data}}{\boldsymbol{aDIFStime} \; ? \; 3 ? \; \boldsymbol{aSIFStime} \; ? \; \boldsymbol{RTS} \; ? \; \boldsymbol{CTS} \; ? \; \boldsymbol{ACK} \; ? \; \boldsymbol{Data}} \; ? \, 100\% \qquad ?2.9?$$

As we know, the *aDIFStime* is 45µs comes from the *aSIFStime* 5µs plus twice the *aSlotTime* 20µs, and we can get the *aPIFStime* 25µs from the same time units. Obviously, the extra PLCP bits for the PHY required are viewed as the useful data, and we put those bits into the throughput consideration.

Some parameters are fixed, like the distribution of the time interval between continuous packets is Exponential with mean rate 50msec, which constructs a Poisson distribution for packet generating in one STA shown in the Figure 10.

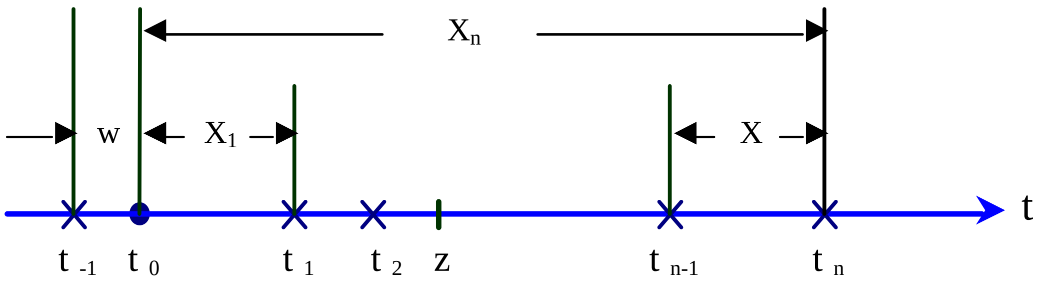

Figure 10. Poisson Distribution Density

And, the Poisson distribution function is [31], [32]:

$$f_n ?x?? \frac{1}{n!} ?? t?^n e^{??t} \qquad n ? 0,1,2\square \qquad ?2.10?$$

In the simulation cases, we have tested different traffic loadings with station numbers 10, 30 and 50 along with an AP in a BSS infrastructure to emulate as the light, meddle and heavy loadings. The packet length is fixed at 1000 bytes, including the MAC Header, Frame body field and the FCS field. Add the 192 bits PLCP Preamble and Header to form the data packet we used in the emulation. The contention window range is from 7 to 255 as recommended in the IEEE 802.11, and this is referred as the *aCWmin* is 7 and the *aCWmax* is 255. And definitely no hidden terminal exists when the RTS/CTS packets are exchanged prior to the real data transmission. The whole emulation is stopped at the 100,000th data transmission.

The Figure 11 shows the average delay time of each collision number when there are 10 stations in the case.

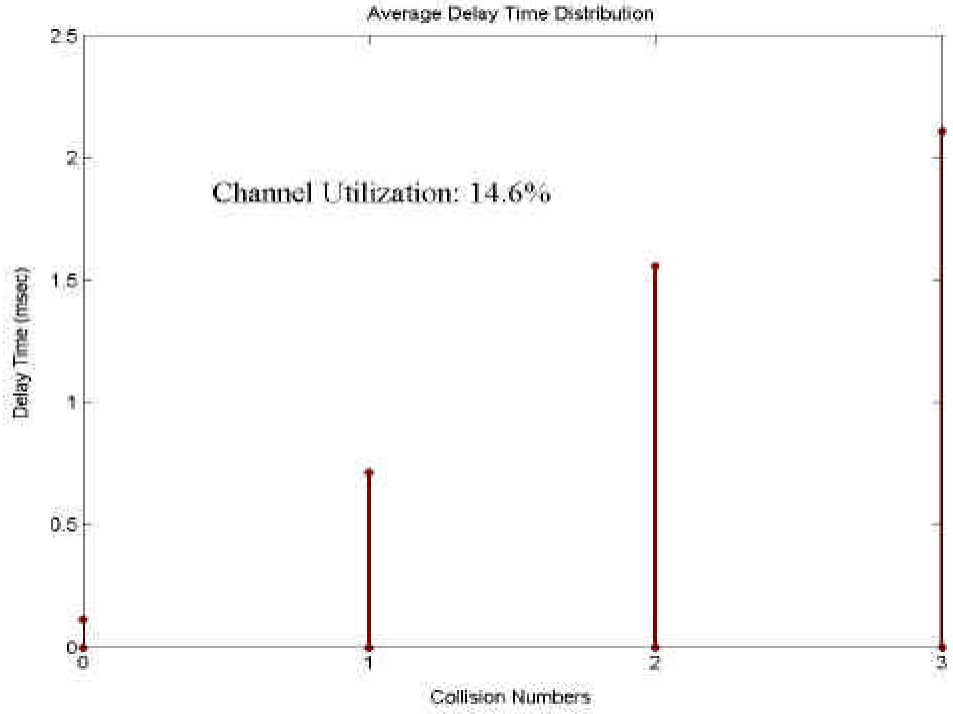

Figure 11. Average delay time of each collision number under light load with the MPDU 1000 bytes

We can tell the average delay time is increasing as the collision number increases. The channel utilization is quiet less. Since the loading is light, the medium is not too busy. If a packet doesn’t collide, its delay time always keeps at 115μs, including *aDIFStime*, *aSIFStime*, *aRTStime* and *aCTStime*.

The mean delay time and the standard deviation are shown in the Figure 12, which is for the frame size 1000 bytes. We can predict easily that the packet numbers of higher collision number is less than that of fewer collision number. Most packets can be transmitted out within 1msec as the Delay Time Distribution in Figure 12 shows. The meanings of delay time and standard deviation are the fairness of each packet. If the overall delay time is small, the delay of each packet can be viewed as small. Whenever the standard deviation is small, we can predict more exactly the timing this packet been transmitted out.

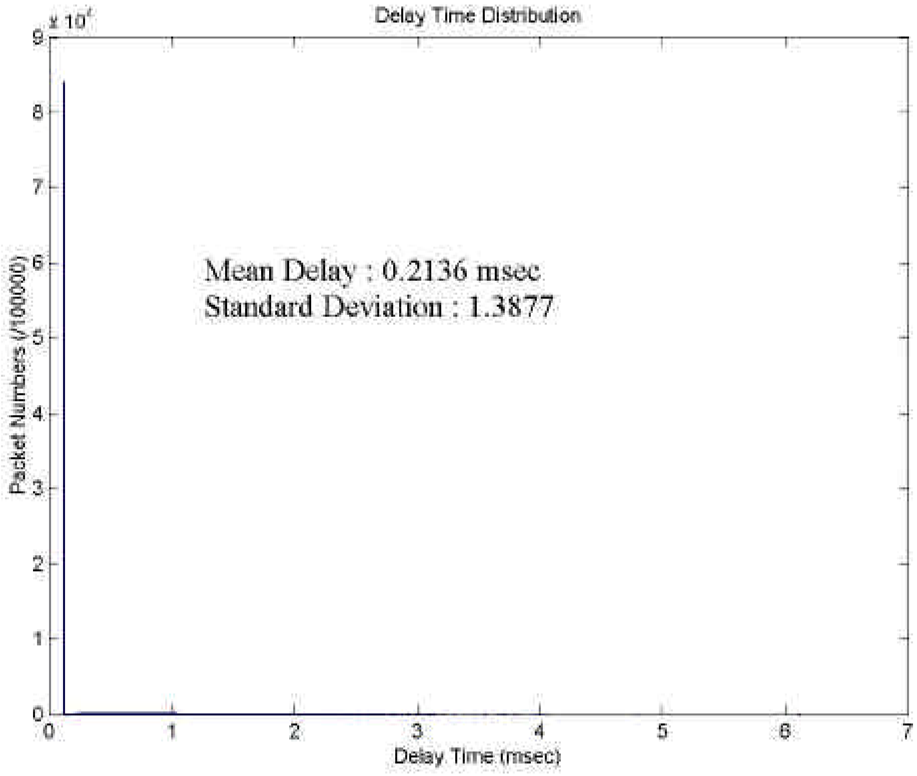

Figure 12. Delay time distribution under light load with the MPDU 1000 bytes

We are much interested at the packets with collisions. Narrow down from the Figure 12, we generate the Figure 13 to see what is the delay time distribution for those collided packets.

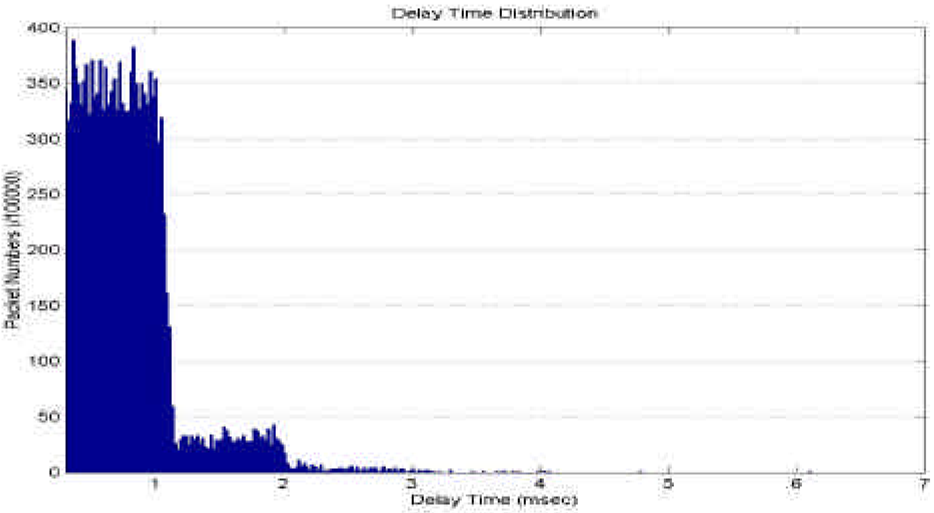

Figure 13. Delay time distribution under light load with the MPDU 1000 bytes, and without no-collision packets

There are more simulation results under middle and heavy loadings shown in the Figure 14 to the Figure 19.

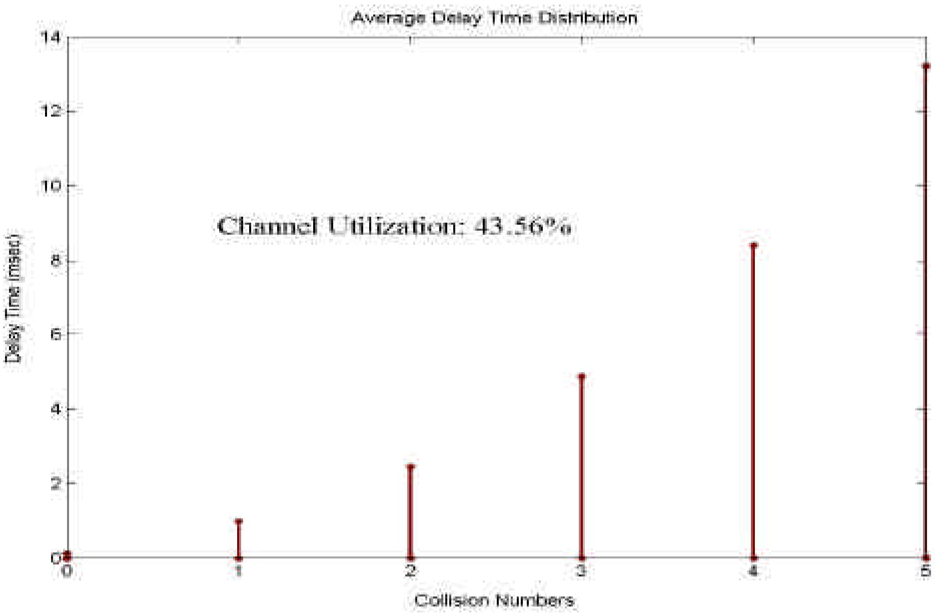

Figure 14. Average delay time of each collision number under middle load with the MPDU 1000 bytes

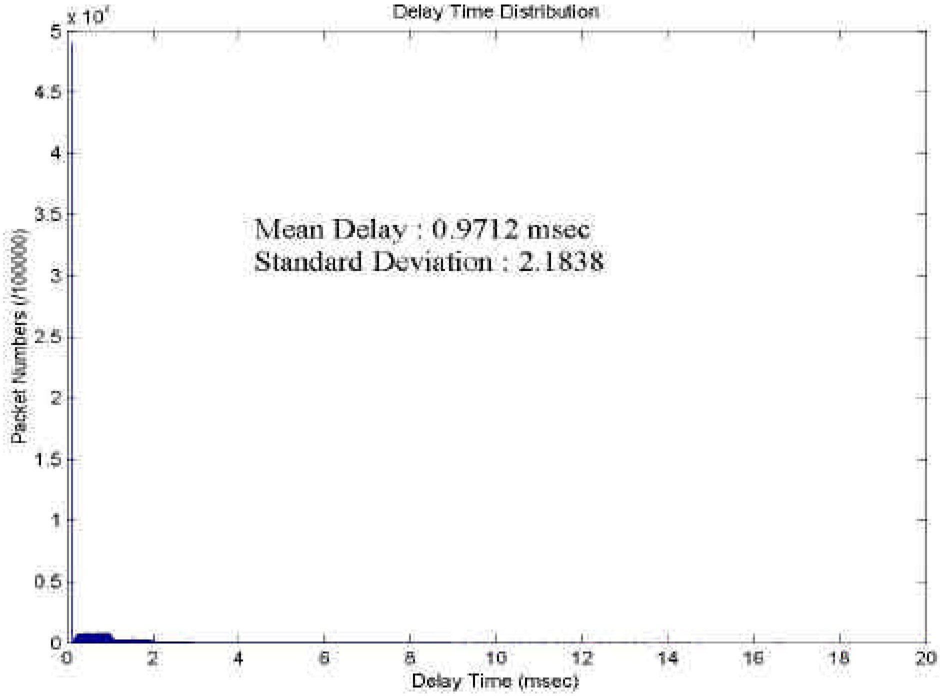

Figure 15. Delay time distribution under middle load with the MPDU 1000 bytes

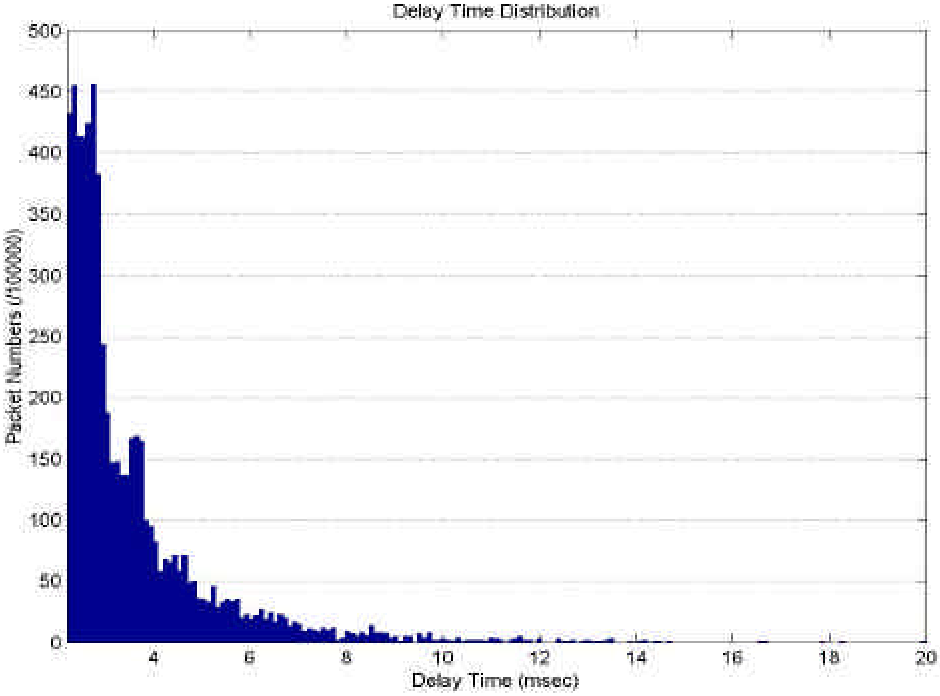

Figure 16. Delay time distribution under middle load with the MPDU 1000 bytes, and without no-collision packets

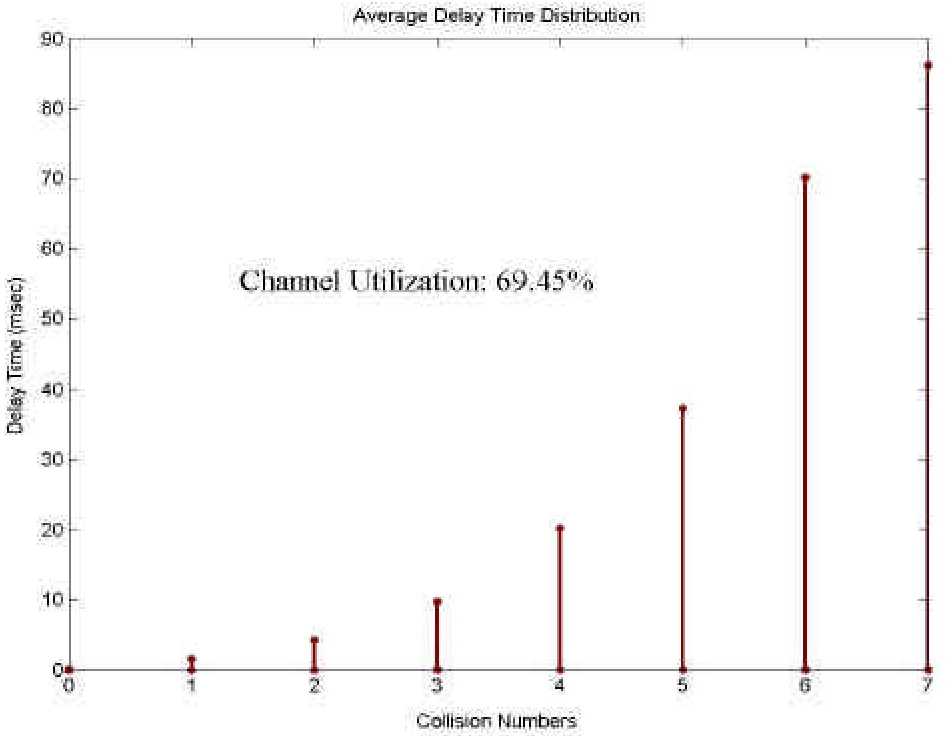

Figure 17. Average delay time of each collision number under heavy load with the MPDU 1000 bytes

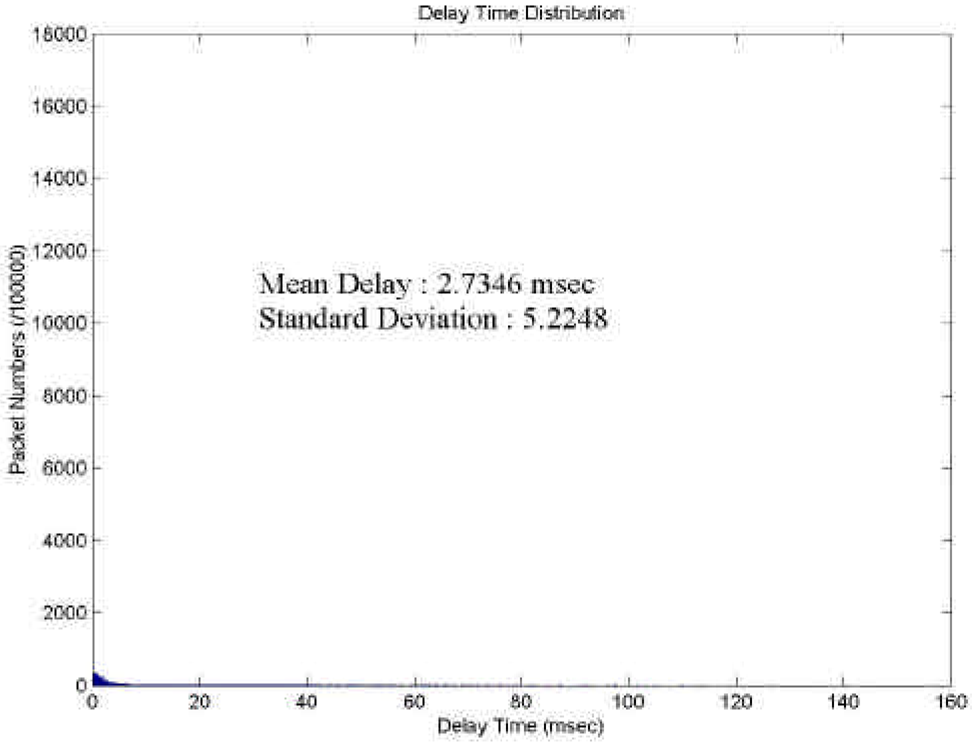

Figure 18. Delay time distribution under heavy load with the MPDU 1000 bytes

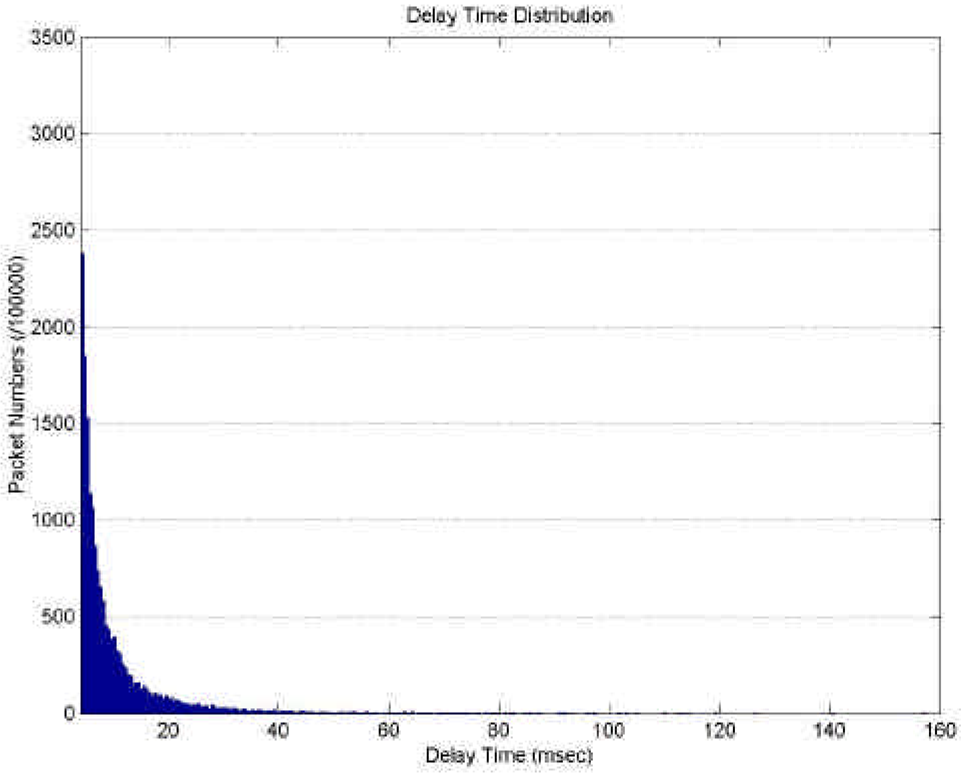

Figure 19. Delay time distribution under heavy load with the MPDU 1000 bytes, and without no-collision packets

The Figure 11, Figure 14 and Figure 17 show the obvious trend that the average delay time grows up along with the collision numbers and traffic loadings. According to the simulations, the maximum average delay is 2.1 msec in the light loading, but almost 90 msec in the heavy loading. This can explain why the performance of heavily collided packet is so bad. As recommended in the IEEE 802.11 [1], when the packet is collided, the contention window size is double unless it already reaches the maximum value and stands at the value, or if the collision number reaches the threshold, SLRC.

The contention window will not reset to the initial value until this packet is transmitted successfully. This property will result in that the heavily collided packet chooses a larger backoff timer more possible and so that the delay time is much more.

And, we can figure out the collision numbers grow also along with the traffic loadings. More heavy traffics will introduce more serious collisions. When there are more collided packets waiting for the chance to access the medium, the much more delay time and worse throughput will happen. It forms a negative loop.

We can understand how is the performance from the Figure 12, Figure 15 and Figure 18. The mean delay time is worse when the loading is getting heavy in the simulation environment. This time value stands for the loading status. There is another phenomenon we should pay more attention on that. The standard deviation varies fast when the traffic loading is heavy. In DCF mode, what we can not control well is the delay time of each collided packet. The exact transmitting time is so different for each packet because of the collisions. In other words, if we can estimate more precisely the possible delay time of most packets, we can make the packet to be transmitted more smoothly. In such case, we can roughly do some bandwidth allocation and data flow control, and can also sense the possible throughput before the real transmission starts. This information plays a very important role in the quality requirement.

The figure 20 shows the throughput under different station numbers.

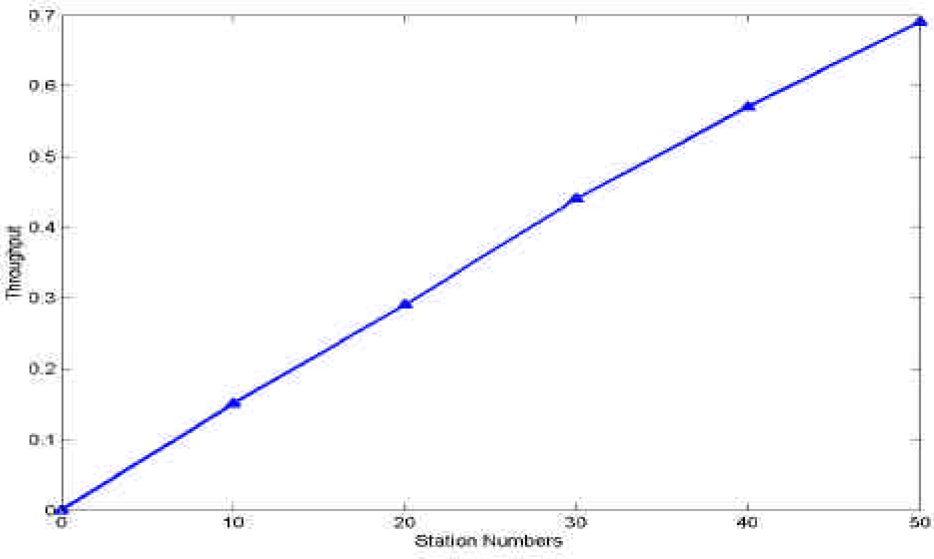

Figure 20. Throughput vs. Loadings with contention window 7-255

When the station number increases, the throughput increases too. Since there are many stations waiting for the chance to access the medium especially under heavy loading, the idle time of the medium can be reduced; which means the medium is quite busy and the deferring time of packets is longer. We also know some parameters will impact the throughput, like *aCWmin* and *aCWmax*. When *aCWmin* is set to be small, the packets will be collided more if there are many active stations in the BSS. Because the possibility of more than two stations choosing the same backoff timer will increase along with a smaller *aCWmin*. However, if the *aCWmin* is set to be large, the waiting time for accessing the medium will be longer. Also, a smaller *aCWmax* can limit the expansion rate of the contention window, but also induces more collisions when the loading is heavy.

Chapter 3
The Fair Backoff Algorithm

According to the simulation results in chapter 2, we found the performance is getting worse especially when the traffic is more heavy. The standard deviation of delay time is getting serious and the collision condition is completely out of the control. That is the root cause to make the transmission quality unstable.

We wish to resolve this condition and make every packet fairer no matter how many collisions happen on this packet. To achieve this goal, we should try to make the packet delay and the standard deviation of delay more limited and more predictable. That will let us be able to estimate the possible delay distribution, and therefore enhance the throughput of this channel even without PCF mode. When the delay time is predictable, we can do more research in DCF mode. Actually, we probably can leverage the channel bandwidth when all stations are in DCF mode and thus improve the performance when using the information of both traffic loading and delay time. With this, we can estimate the impact of the performance also. We wish to generate a pretty fair wireless environment even though the loading is heavy, we still can estimate the available bandwidth for each station.

The RTS/CTS algorithm should be used to avoid the hidden terminals problem, and it works as the standard IEEE 802.11 [1]. We ignore some conditions not happened always. For example, when one STA joins the BSS and not ready to hear the other STAs' packets to the AP or the third STA, it will not initiates a transmission immediately to reduce the collisions probability caused by the hidden terminals. Or, there are two or more parallel STAs desiring to transmit packets at the same time and thus collide the packets until figure it out without the ACK frame within a specified duration, *aSIFStime*. A four-way handshaking procedure is applied in our simulation cases, and we assume all packets, including RTS/CTS, Data, or ACK frame, are received correctly by the addressed STA.

The motivation and some other related works are included in the section 3.1. And, a modified backoff algorithm is shown in the section 3.2. Finally, we can compare the simulation results between the conventional exponential backoff algorithm and the modified one in the section 3.3.

3.1 Motivation

Some works focus on how to differentiate the service level of each packet. In [20], the author designs a scheduling scheme that compensates for channel errors and balances transmission opportunities among all flows. This scheme equalizes the channel access between uplink and downlink flows, since the author observes a significant unfairness between these two directions. An Out-of-Band Signaling (OBS) is proposed in [22]. The author introduces a MAC layer scheduling scheme to improve the performance of a high speed WLAN with the use of a separate low speed channel for signaling which means the RTS/CTS handshaking mechanism. In [23], the Per-Packet Priorities (P3) function is present. The P3-DCF enhances the DCF for prioritized service in the IEEE 802.11. This integration establishes not only a per-flow differentiation but schedules the packets with an Earliest Deadline First discipline as well.

Fairness in wireless networks has been studied under various network scenarios, ranging from cellular networks and WLANs to ad hoc networks. With regards to WLANs, of particular interests are the works in [10], [15], [19], presenting new and distributed MAC algorithms aimed at providing the fair channel access. In [10], a quality of service aware MAC with two new algorithms of Virtual MAC (VMAC) and Virtual Source (VS) is implemented to support a distributed DiffServ capable and radio resource monitoring mechanisms. Similar to the localized management of [10], a decentralized control mechanism suppressing delay fluctuation in CSMA/CA networks, called DDCF, is proposed in [15]. Another Distributed Bandwidth Allocation/Sharing /Extension (DBASE) protocol is introduced in [19]. This protocol can support both asynchronous traffics and multimedia traffics with the characteristics of Variable Bit Rate (VBR) and Constant Bit Rate (CBR) over the IEEE 802.11 ad hoc WLAN. The designed DBASE protocol will reserve bandwidth for real-time stations based on a fair and efficient allocation.

There are some works focusing on the enhancement of the backoff algorithm in [26]-[30]. We can understand the conventional backoff algorithm performance in [26]. The author introduces the relationship between throughput, delay, contention window, and the offered load. The choice of the *aCWmin* and *aCWmax* arameters was analyzed in particularly. The simulation results show that the choice of aCWmin value is dependent on the number of transmitting stations. In [27], a mechanism named Asymptotically Optimal Backoff (AOB), dynamically adapts the backoff window size to the current load, is proposed. The AOB adapts the backoff to the network contention level by using two simple load estimates: the slot utilization and the average size of transmitted frames. A different point of view from the [27], the same author proposes an Adaptive backoff mechanism in [28]. The results obtained indicate that under stationary traffic and network configurations, the capacity of the enhanced protocol approaches the theoretical limits in all the configurations analyzed. More complicated algorithm is showed in [29]. Using exponential increase as the conventional backoff algorithm does, but different exponential decrease to enhance the performance of DCF. The contention window is not back to the *aCWmin* when the packet is transmitted successfully, but only reduce the value by $\sqrt{2}$. The simulations reveal the saturation throughput can be improved to 0.8. Another analysis of saturation throughput and saturation delay is in [30]. It includes the minimum backoff window size, the backoff window-increasing factor, and the maximum backoff stage as the parameters. A backoff-based priority schemes for IEEE 802.11 are achieved by differentiating these parameters.

In this thesis, we observe the delay time of each packet. Ideally, when the collision number of one packet happens more, the backoff time interval the collided STA chooses should be shorter if we want to balance the delay time of each packet no matter how many times it has been collided. In other words, the new algorithm should have a characteristic with lower increasing rate of the delay time of every packet in accordance to the collision numbers. In this case, every packet can have a fair opportunity to be transmitted out and is irrelevant to the collision numbers. The traffic loading is another issue we have to take care. If there exists too many STAs in the BSS, a smaller backoff timer will introduce more collisions and impact the delay time.

The conventional Exponential backoff algorithm cannot improve the fairness of each packet because of the expansion rate of the contention window size is much higher than the collided rate. One more collision will make the contention window size double according to the binary backoff procedure recommended in the IEEE

802.11 [1]. The contention window size plays an important role in choosing the backoff time interval. Since it operates according to the random number generating in the contention window, it’s so called uniform distribution.

The Figure 21 shows the growth trend of size of the contention window. When first collision happens, the backoff time interval should be chosen from the contention window, and its value should be *aCWmin*. In the case, 7 is selected. One more collision increases, the contention window increases by [$2^{(\text{collision number + n-1})}$-1]. The variable n shown in the formula is the power of 2 related with the *aCWmin*. If the minimum contention window size is 7, the variable n is 3. If an alternative is selected, like 15, the variable n is 4, which is followed the formula (3.1):

$$\boldsymbol{n} \; ? \; \sqrt{\boldsymbol{aCWmin} \; ? \; 1} \qquad ?3.1?$$

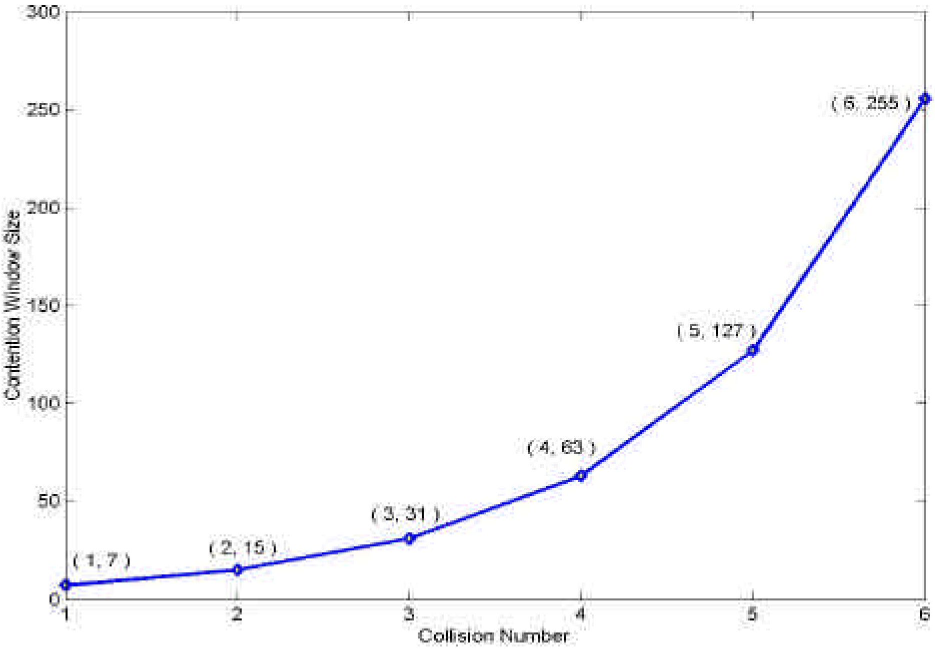

Figure 21. The Contention Window Size vs. the Collsion Numbers Double if One More Collision

The probability distribution function of uniform distribution is:

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a \le x \le b \\ 0, & others \end{cases} \qquad (3.2)$$

According to the IEEE 802.11, the contention window size distribution of backoff algorithm should be as the Figure 22. When the packet is collided more, the contention window is expanded more. If one time collision, the backoff timer is chosen from 0 ~ 7, and every number is with the probability 1/7. One more collision, the probability is down to 1/15 but the backoff timer is possible as large as 15. That is why the delay time is so worse in the Figure 18 when heavy loading.

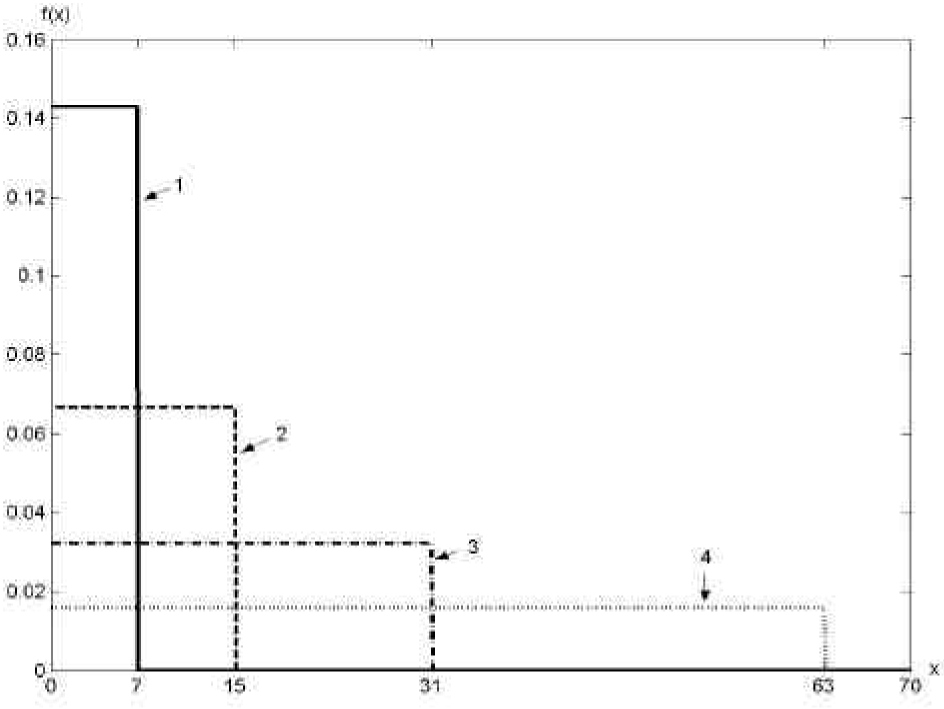

Figure 22. Uniform Distribution

Mean and standard deviation values of continuous probability distribution are defined as [31], [32]:

$$E(x) = \int x \cdot f(x)\,dx \qquad (3.3)$$

$$SD?x?? \sqrt{E(x^2) ? E^2(x)} \qquad ?3.4?$$

And, we can get the mean value and standard deviation of the uniform distribution from the definitions are:

$$E?x?? \frac{b ? a}{2} \qquad ?3.5?$$

$$SD?x?? \frac{?b ? a?}{\sqrt{12}} \qquad ?3.6?$$

We can exactly know why the collision delay time is so worse when the collision number only increases a little. According to the definition, we find the mean value, E(x), almost presents an double growth rate, and the standard deviation, SD(x) , even larger than that. The Figure 23 is the distribution of mean value based on the conventional exponential backoff algorithm. The *aCWmin* is set to 7 initially, which means when first collision happens, the contention window is 7.

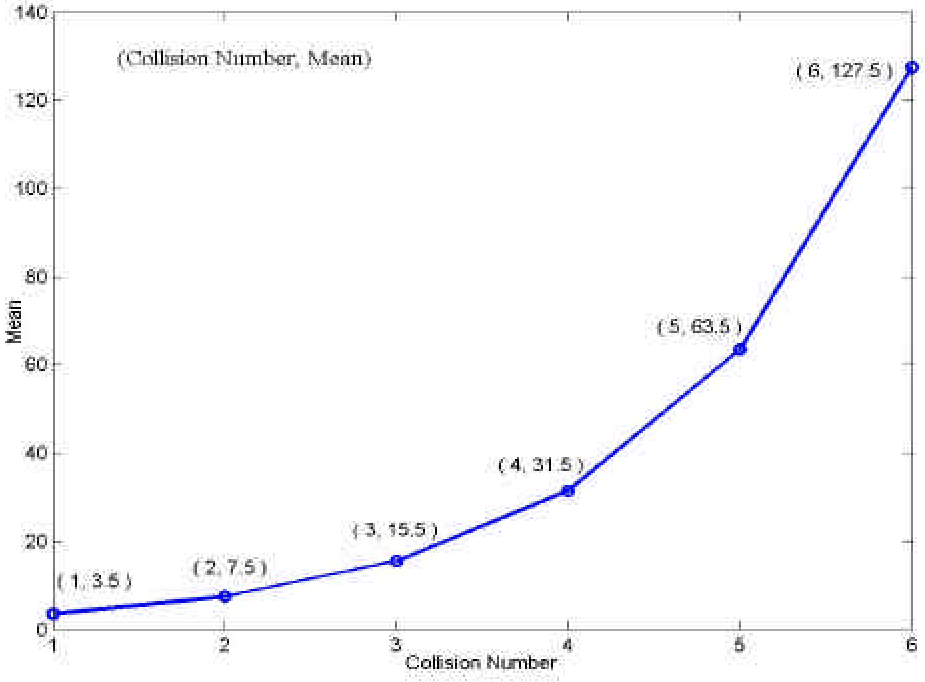

Figure 23. Mean of Conventional Backoff Algorithm under different Collision Number

The most impact of the delay time is not only the mean value of the backoff time interval, but also its standard deviation. As we know, when the standard deviation grows larger, the difference between two values becomes worse. In such condition, we cannot estimate the possible transmitting time, and the performance will be out of control. The Figure 24 only tells us how the possible standard deviation is when the collisions increases. As to the standard deviation time, we should put the *aSlotTime* into consideration according to the (2.5).

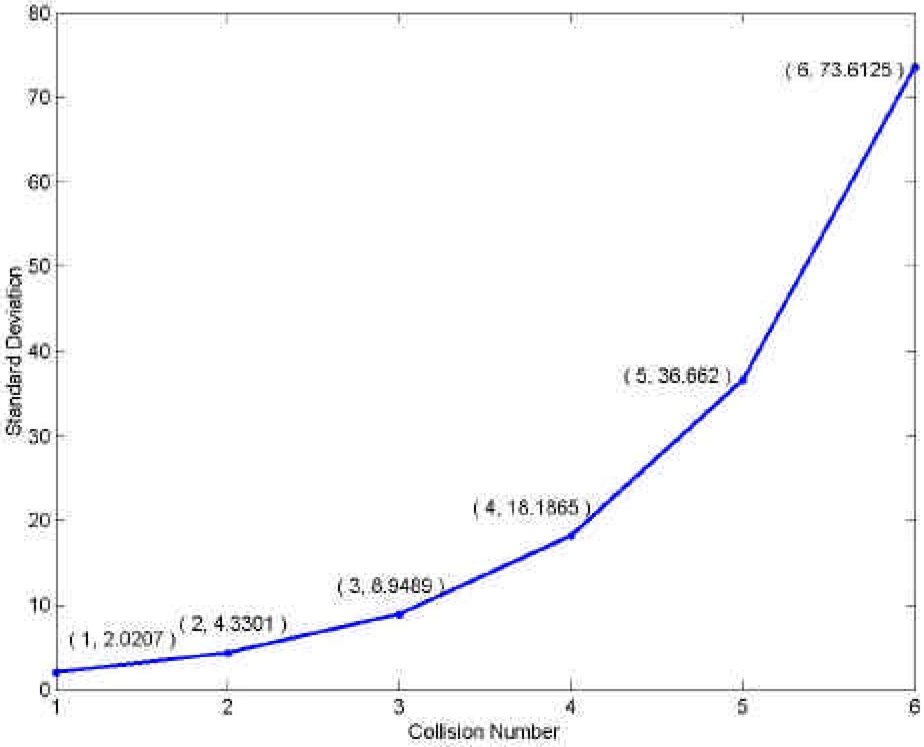

Figure 24. Standard deviation of Conventional Backoff Algorithm under different Collision Number

3.2 Fair Backoff Algorithm

Until now, we can make sure what we need for the fairness is a new backoff algorithm which can generate smaller mean and standard deviation values even when the collision happens more on that packet. Those characteristics can improve the overall performance of all packets with any kind of collisions, in other words, any kind of traffic loading. When most of the packets under these different loadings can have almost equal chance to be transmitted out, the fairness of each packet is better than that in the IEEE 802.11 [1]. In order to achieve the target, the Gamma distribution for the backoff algorithm is proposed here.

We should take a look at the probability distribution function of Gamma [31], [32] to understand the mathematical characteristics:

$$\boldsymbol{f} : ?\ \ ?\ \ ?_{?} : \boldsymbol{f}\,?\boldsymbol{x}??\ \begin{matrix} ? \\ ? \\ ? \\ ? \\ ? \end{matrix} \begin{matrix} \dfrac{1}{???\,??^{?}}\,\boldsymbol{x}^{??1}, \boldsymbol{if}\ \ \boldsymbol{x}\ ?\ 0, \\ \\ 0, \qquad\qquad \boldsymbol{if}\ \ \boldsymbol{x}\ ?\ 0, \end{matrix} \begin{matrix} ? \\ ? \\ ? \\ ? \\ ? \end{matrix} \qquad ?3.7?$$

where,

$$???\ ??\ ?_{0}^{??}\ \boldsymbol{y}^{??1}\boldsymbol{e}^{?\boldsymbol{y}}\boldsymbol{dy},\quad ??\ ?\ 0, \qquad ?3.8?$$

and,

(1) $??\ \ ?1, ???\ ??\ ??\ \ ?1???\ ??\ \ ?1?$

(2) $??\ ?\ ?\ ,???\ ??\ ??\ \ ?1?!$

(3) $??1/2??\ \sqrt{?}\,, ??3/2??\ \sqrt{?}\,/2$

Again, according to the definition of mean and variance, we can get:

$$\boldsymbol{E}\,?\boldsymbol{x}??\ ?\boldsymbol{x}\,?\boldsymbol{f}\,?\boldsymbol{x}??\boldsymbol{dx}\ ??\ ? \qquad ?3.9?$$

$$\boldsymbol{SD}?\boldsymbol{x}??\ \sqrt{\boldsymbol{E}(\boldsymbol{x}^{2})\,?\,\boldsymbol{E}^{2}(\boldsymbol{x})}\ ?\ \sqrt{?\ \ ?^{2}} \qquad ?3.10?$$

The Figure 25 is the probability distribution function of Gamma with a is fixed at 2, and ß varies from 0.5 to 3. We can see the standard deviation grows larger when ß varies from 0.5 to 3 in the figure. We can recognize the trend of the distribution with different combinations of a and ß.

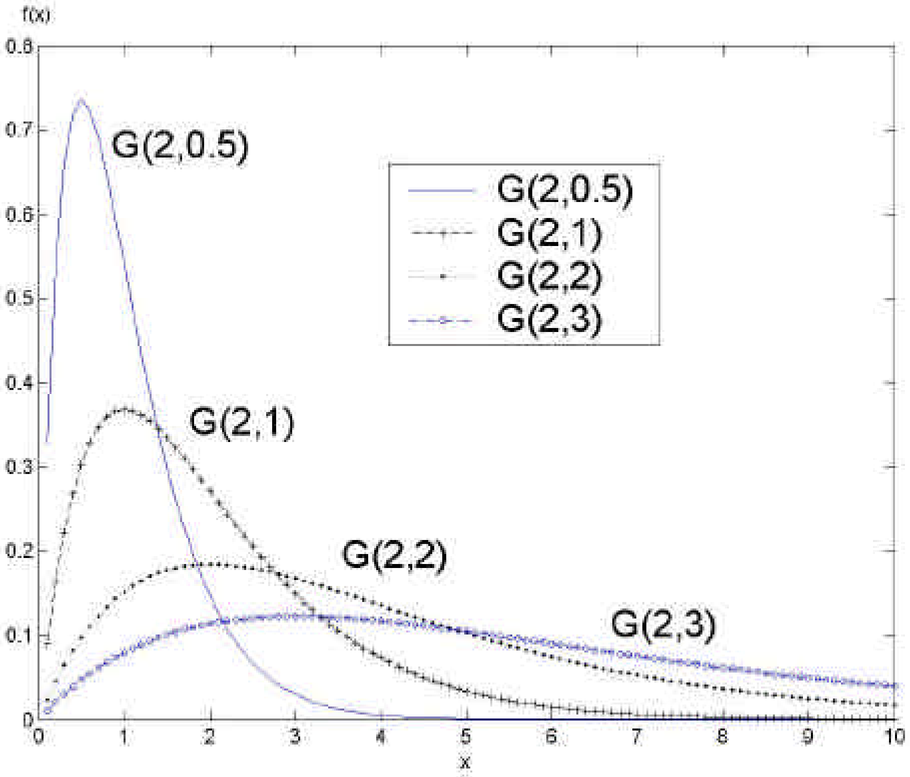

Figure 25. Gamma Distribution with different ß and the same a

With the same α, the distribution concentrates much more when β value goes small. For β < 1, since the function of Gamma Distribution has a vector with the inverse of β with the power of α, the value of f(x) goes smaller as β is getting larger based on the formula (3.7). The Figure 25 tells us how the value of β impacts the distribution under the same α. Based on the idea we want to implement, we define the β as the collision number. As to the α, it operates as a based parameter in the (3.7). If α increases, the value of Gamma(α) increases more, and f(x) has a smaller basement. We take the α as the contention window.

We don't want to change the backoff procedure too much, because we desire to achieve the target with the least modification and hardware requirement. In this case, the backoff timer is generated by the same way as the exponential backoff was, but with a different ‘Random Number’ generator.

The modified Gamma backoff procedure is described in detailed now. As recommended in the IEEE 802.11, we follow the same rule but modifying the method of random number as:

Backoff Time* ? *Gamrnd*? ?? *aSlotTime ?3.11?

where,

Gamrnd( ) = Pseudorandom integer drawn from a gamma distribution with the parameters α and β, where α is defined as the contention window CW and β is the collision numbers. The CW is an integer within the range of values of the PHY characteristics *aCWmin* and *aCWmax*, which is *aCWmin*≦CW≦*aCWmax*. Whenever collided one more time, the CW decreases one from *aCWmax* until it reaches *aCWmin*. The duration between *aCWmin* and *aCWmax* is defined as the Contention Window Size (CWS).

aSlotTime = The value of the correspondingly named PHY characteristic.

The contention window shall take the *aCWmax* as its initial value when the packet is first collided. As the same as the conventional backoff algorithm does, the CW will take the next value when the packet is collided again. The value of CW will decrease one by one whenever the packet is collided until the CW reaches the *aCWmin* and the CW shall remain at that value until it is reset when the collided packet is sent out or the SLRC reaches the threshold.

As we can tell from the discussion above, we know there are some reasons that will influence the collisions. The station number dominates the traffic loading. The probability of collision of every packet depends strongly on the station number within the BSS of an AP. The packet length and the arrival rate of each station also operate an important factor in collisions. We assume those as a fixed value in our simulation, and they result in a constant when we decide the *aCWmax* and the CWS values.

The *aCWmax* is the initial value of the contention window, and it should be derived from the station number, packet length and the arrival rate of packet. Take 6 as the basement of the rule because we set the packet length and the arrival rate of packet as fixed values, we follow the rule as below:

$$\boldsymbol{aCWmax} ? 6 ? 2^{[\boldsymbol{Station\ Number}/10]?1} \qquad ?3.12?$$

Then, we can get the values of the *aCWmax* for light loading, middle loading and heavy loading are 7, 10 and 22 respectively. We assume the STA can learn the loading before it starts to transmit packets, which means the STA will decide the initial value of contention window before the real collision happens. Similarly, the CWS is derived from the same parameters but different weighting. Even in the heavy loading, the number of packets with serious collision is definitely less than that of lightly collided. We can reduce the duration of contention window and this will help to restrict the diversity of the standard deviation. The way we choose CWS is:

$$\boldsymbol{CWS} ? ?\frac{\boldsymbol{Station\ Number}}{10}? ? ?\frac{\boldsymbol{Packet\ Length}}{1000}? ? ?\frac{1}{\boldsymbol{Arrival} ? \boldsymbol{rate}}? \qquad ?3.13?$$

According to the rule, we can get the CWS as 4, 6 and 8 for the different loadings in our simulation cases and result in the value of *aCWmin* for each case as 4, 5, and 15 individually, and again, the *aCWmin* can be learned from the loading before the STA initiates a transmission.

Then, we take the contention window of the light loading for example. The size ranges from 7 to 4, and those are the series for the values of α. All the criteria are the same as those of the conventional backoff algorithm except the method of decreasing the window size and the way to generate the random integer. The Figure 26 is the contention window distribution of the Gamma Backoff Algorithm with *aCWmax* 7.

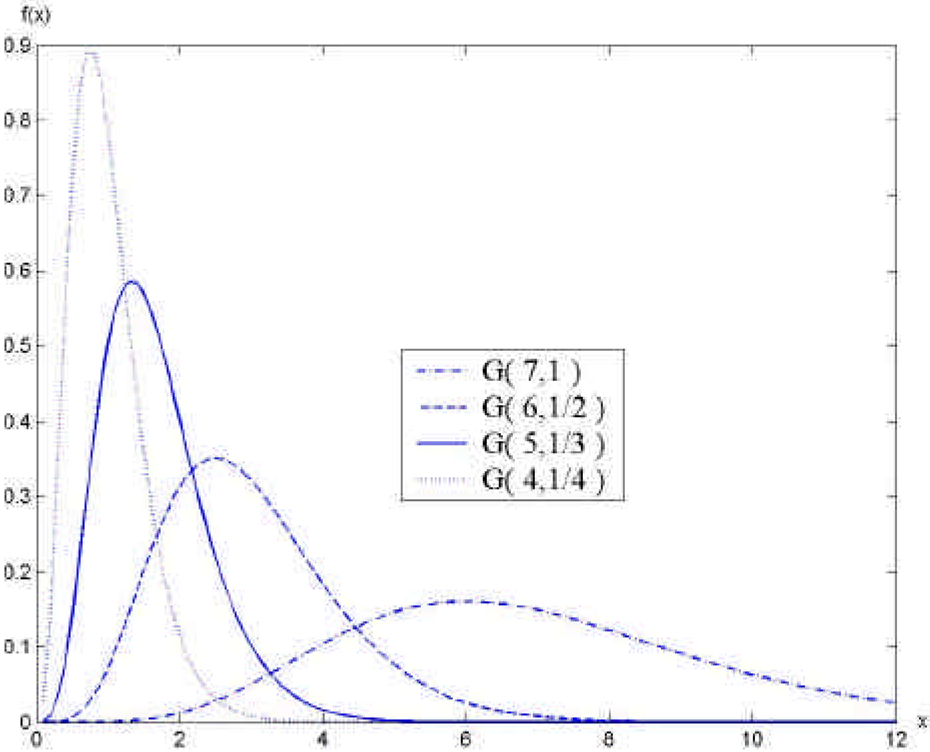

Figure 26. Gamma Distribution with different combinations of a and ß

The initial value *aCWmax* impacts the overall performance especially when the traffic is under the light loading according to the above description. The Figure 27 tells us the distribution when choosing different initial values, and it also means the distribution for the first collision. When the *aCWmax* is larger, the mean and standard deviation values are increasing, and that will make the delay time longer even the loading is light. But it do help when there are many stations in the BSS, since every station might choose more different backoff time interval and reduce the possibility of another collision.

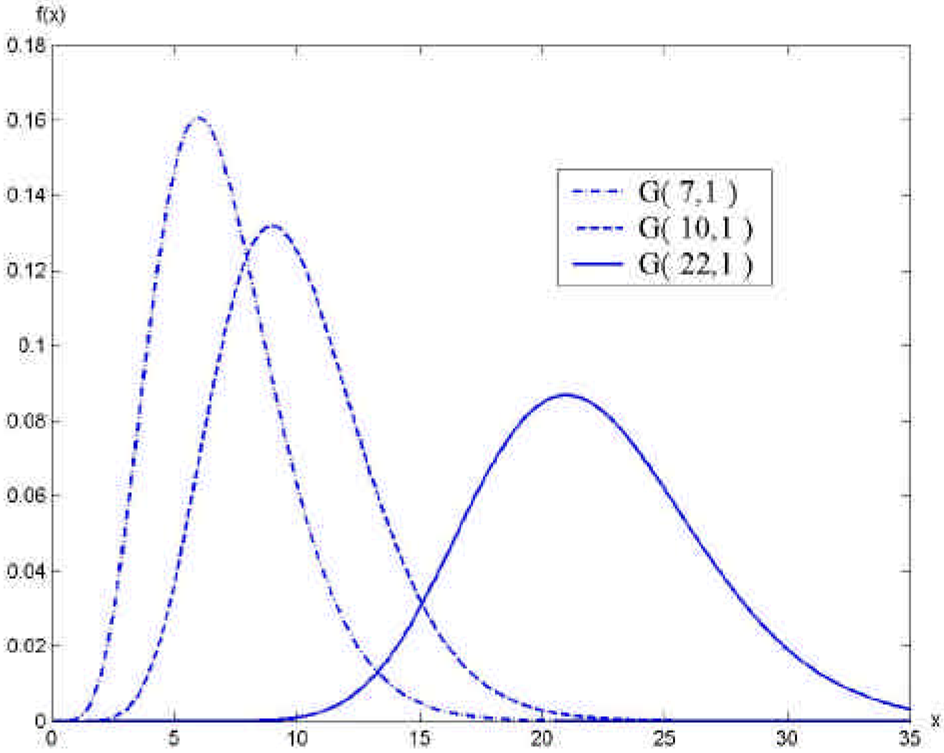

Figure 27. Gamma Distribution with different α (CW)

The collision number is reflected into the value of β, which means β equals to the inverse of collision number. If the packet is collided once, β is set to 1, and when this packet is collided one more time, β will become one over two, and so on. From the Figure 26, we can recognize that the mean and standard deviation values will decrease when the collision number increases, and also the value is less than that of the exponential backoff algorithm. In this case, we can estimate the collided worse packet will backoff less in probability, and this scenario can enhance the overall throughput of the medium.

The Figure 28 is the mean distribution of the modified backoff algorithm under different loadings. Compare with the Figure 23, the mean of the conventional exponential backoff algorithm, the average delay time of the modified backoff algorithm with the collision number 1 and 2 is worse than those of the exponential algorithms. But when the packet is collided more, which part we expect to balance, the modified algorithm shows up its smaller mean value and this value can decide the average delay time of those packets with such collided numbers.

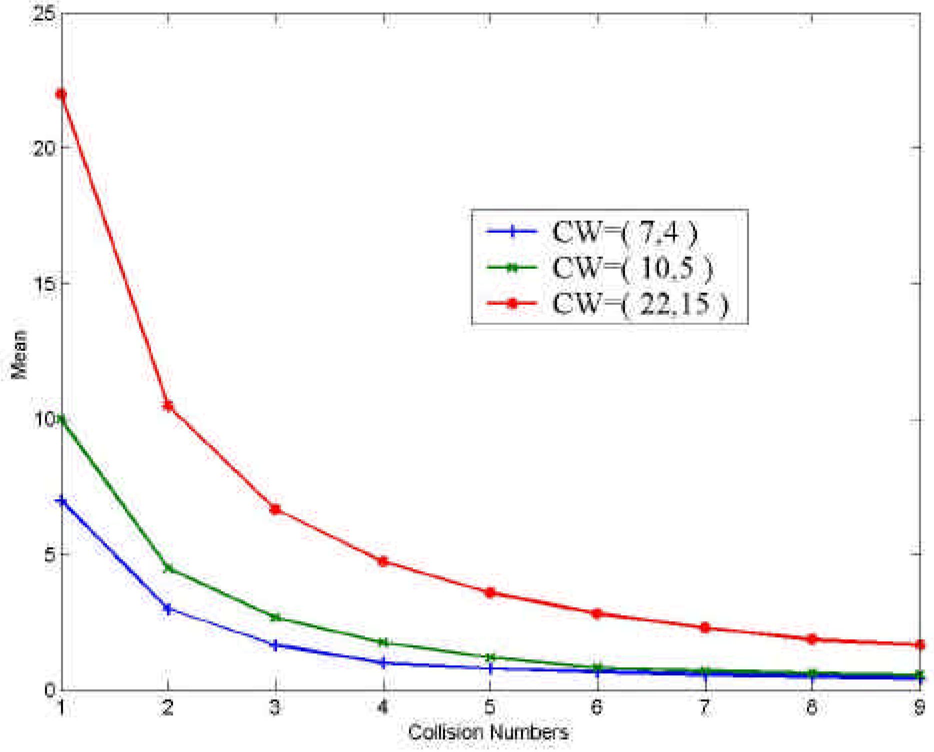

Figure 28. Mean of Gamma Backoff Algorithm under different Loadings and Collision numbers

According to the fair algorithm, when the loading is heavy, the STA shall choose a larger *aCWmax*, which will result in a larger mean value. We can find this characteristic in the Figure 28. In general, when the loading is heavy, the mean of this case will be always larger than other cases with any possible collision number. The decreasing rate slows down much after the collision number is over the contention window size. Take the light loading as the example. Since we get the CWS as 4 from the rule (3.13), we know the α value will keep at 4 even the packet is collided more than four times. But the value of β will present the real collision condition, it still decrease whenever any more collision happens to this packet because β equals to the inverse of the collision number. Following (3.9), we know the mean value equals to the multiply of α and β, and it is only be decided by the β now.

Another important parameter we should pay more attention is the standard deviation of the delay time. This variable can influence the range and precision of the delay time of each collided packet. When the standard deviation value is smaller, the higher precision of delay time can be reached. And we can have more accurately estimation to predict the possible transmitting time of each packet in different loading conditions. From the Figure 24 and the Figure 29, there is a great improvement in the standard deviation especially those seriously collided packets. For the modified Gamma backoff algorithm, we can completely control and predict the difference of delay time since the standard deviation is getting smaller when the collision happens more, comparing with the conventional exponential backoff algorithm.

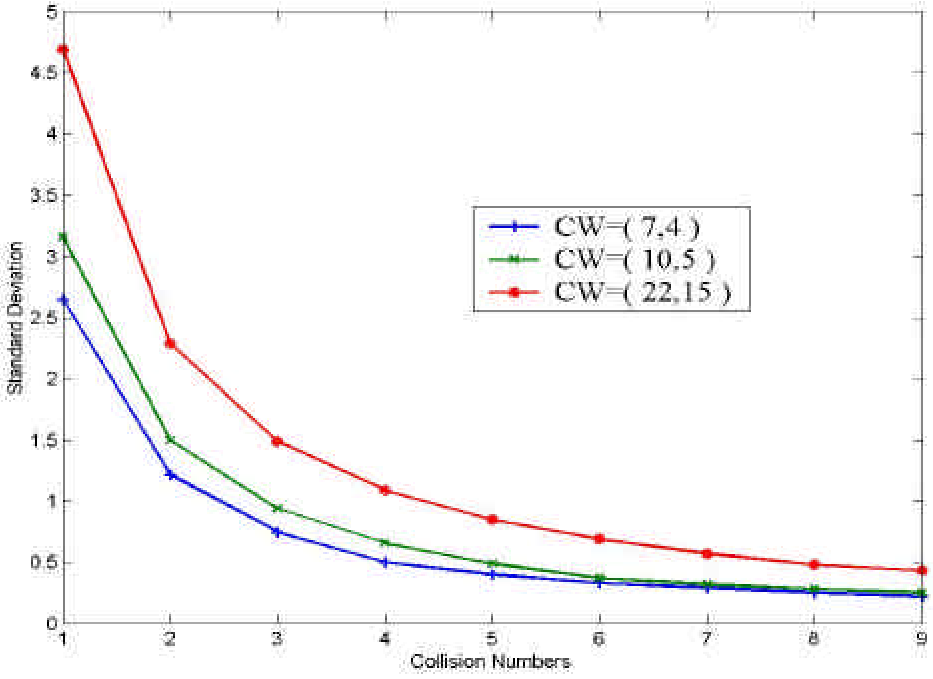

Figure 29. Standard deviation of Gamma Backoff Algorithm under different Loadings and Collision numbers

Based on the fair backoff algorithm, the standard deviation of heavy loading is a little larger than the value of light loading. The most difference happens when the packet is collided less. As we can understand, the collisions of the overall system will be more when the loading is heavy. From the Figure 29, the variation of standard deviation is getting smaller with more collisions under each loading. When the contention window reaches the lower bound *aCWmin*, only β, also refers to the inverse of collision number, can determine the characteristic of standard deviation from (3.10), and the effect is not much.

According to these estimations, we implement the Gamma distribution into the backoff procedure, and we will see the improvement in the next section.

3.3 Simulation Results

The simulation condition is the same as that of the conventional backoff algorithm. Every packet is set to 1000 bytes in length. The arrival rate of generating a new packet in the STA is 500 msec. The RTS/CTS handshaking procedure is implemented and we assume all packets, including the RTS/CTS, Data and ACK, are received correctly within the required duration. We use the new proposed fair backoff algorithm into the simulation case, we can get the simulation results of three different loadings, which are defined as the light, middle and heavy loadings, and we list them in the Figure 30 ~ Figure 40. As mentioned above, the contention windows are different from each case because the CWS depends on the traffic loading.

Compare the Figure 11 and the Figure 30, the characteristics under the light loading, the packets seem to be collided more when using the fair backoff algorithm. But refer to the delay time distribution and the standard deviation in the Figure 31 and Figure 32, we know the overall performance of the modified Gamma backoff algorithm is a little better.

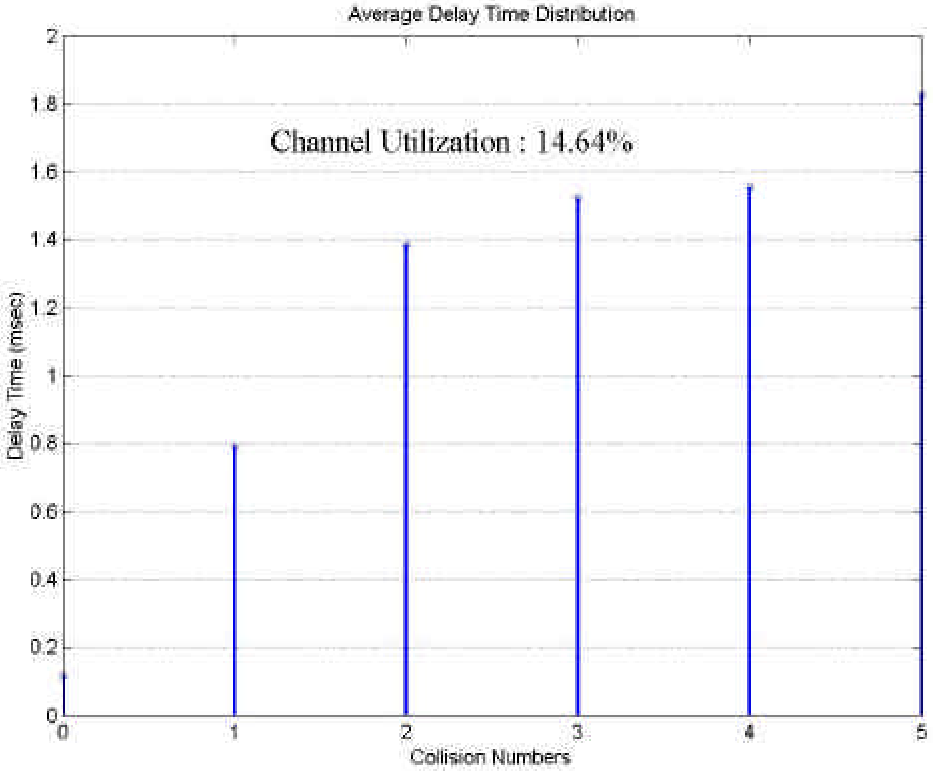

Figure 30. Average delay time of each collision number under light load with the MPDU 1000 bytes

The fairness is defined as the delay time of the packet, no matter collided or non-collided. When the delay time of every kind of packets is very close, we say the fairness of the packet is better. By using the fair backoff algorithm, under light loading, the mean delay of overall packets is not even improved, but the standard deviation is enhanced better. This result is followed the estimation mentioned in the section 3.2. In more detailed, we find the average delay time of those packets with fewer collisions is larger than that of the conventional backoff algorithm. According to the Figure 28, we know if only one collision happens to the packet, the mean delay time is pretty larger than the corresponding value in the Figure 23. Because we choose a wider distribution G(7,1) other than the conventional one, U(7,0) when the packet is collided once.

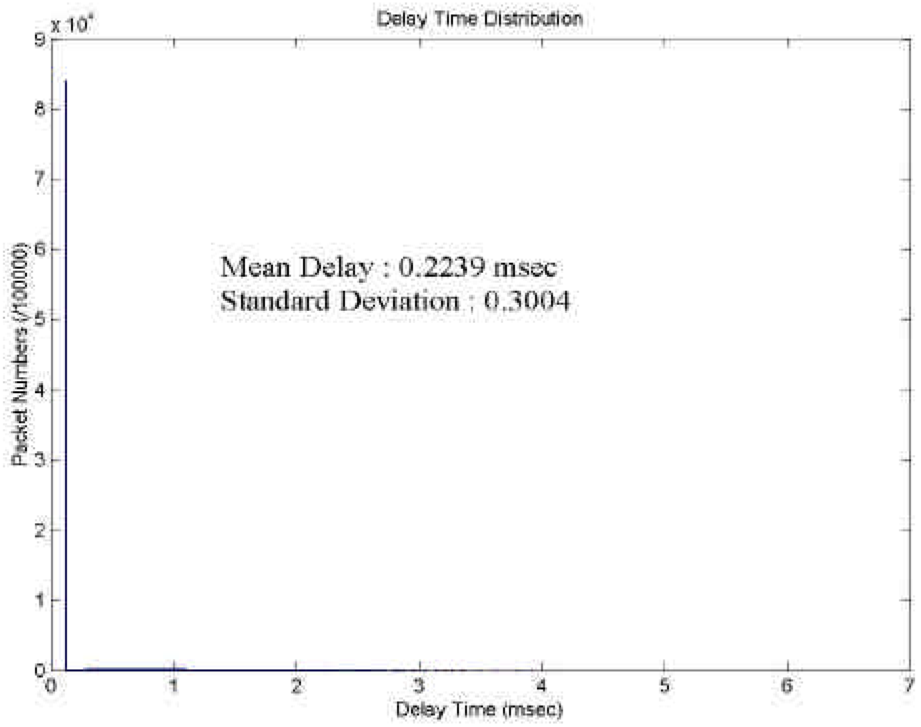

Figure 31. Delay time distribution under light load
with the MPDU 1000 bytes

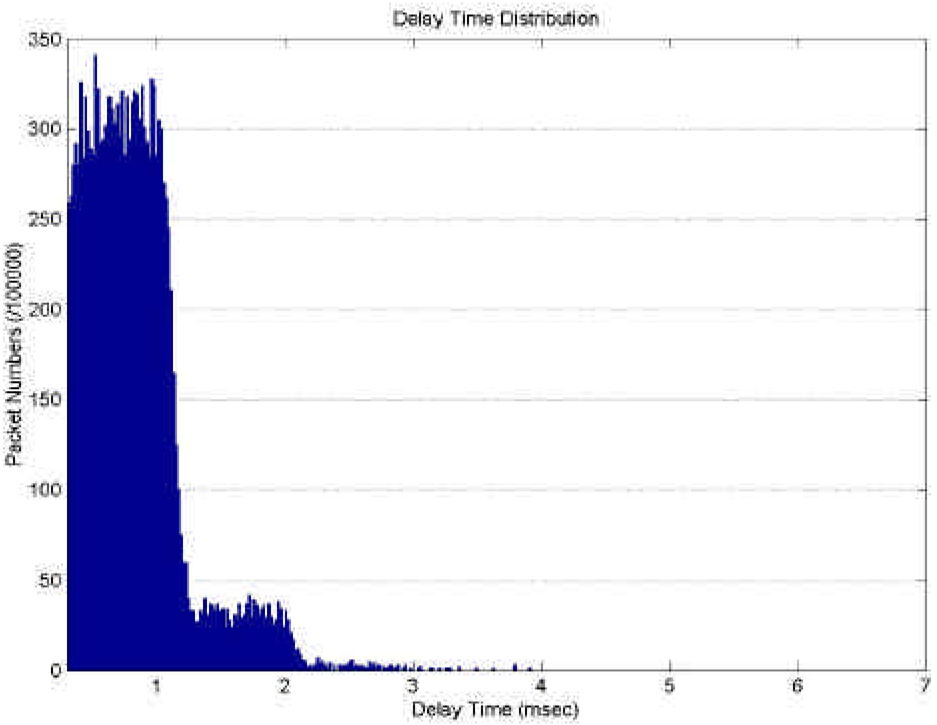

Figure 32. Delay time distribution under light load with the MPDU 1000 bytes,
and without no-collision packets

From the probability point of view, we have more chances to select the backoff time interval more than 7, the upper bound of uniform distribution U(7,0), when the packet is first collided and thus the mean value of those packets with only one collision is larger than the value of using the conventional backoff algorithm in statistically. And, the average delay time distribution is no longer be an exponential expansion, but more smooth when the collision number increases.

Under the middle loading, the simulation results are the Figure 33 ~ Figure 36. The improvement is much more if comparing with the conventional exponential backoff algorithm by the Figure 14, Figure 15 and Figure 16. And we can tell the average delay time and the standard deviation are much limited. Similarly, the collision number is more than that of the conventional backoff algorithm. But we get a great improvement in the average delay time. From the Figure 14, the range of the average delay time is up to 13 msec. As the Figure 33 shows, almost all average delay time can be restricted less than 2 msec.

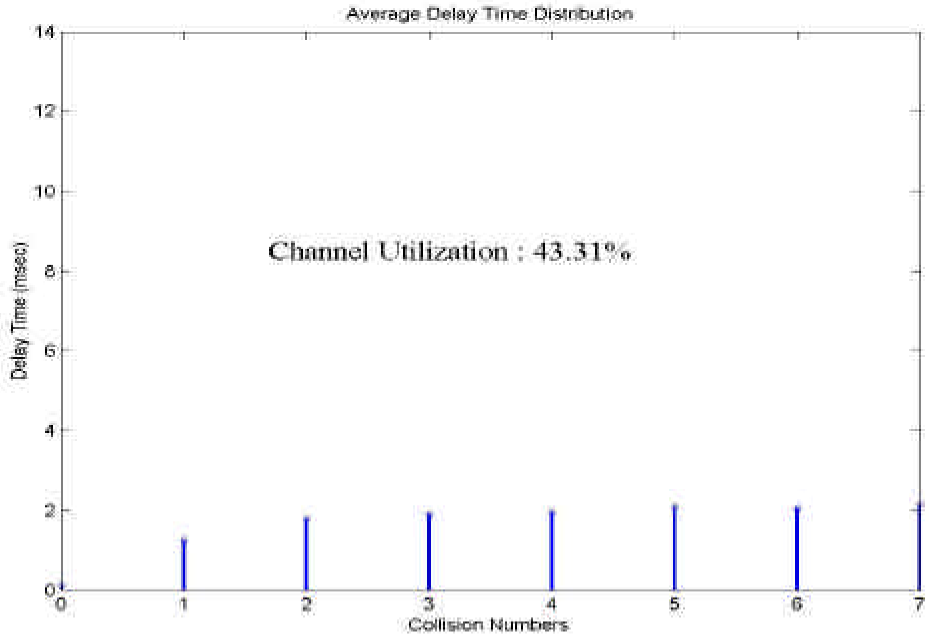

Figure 33. Average delay time of each collision number under middle load with the MPDU 1000 bytes

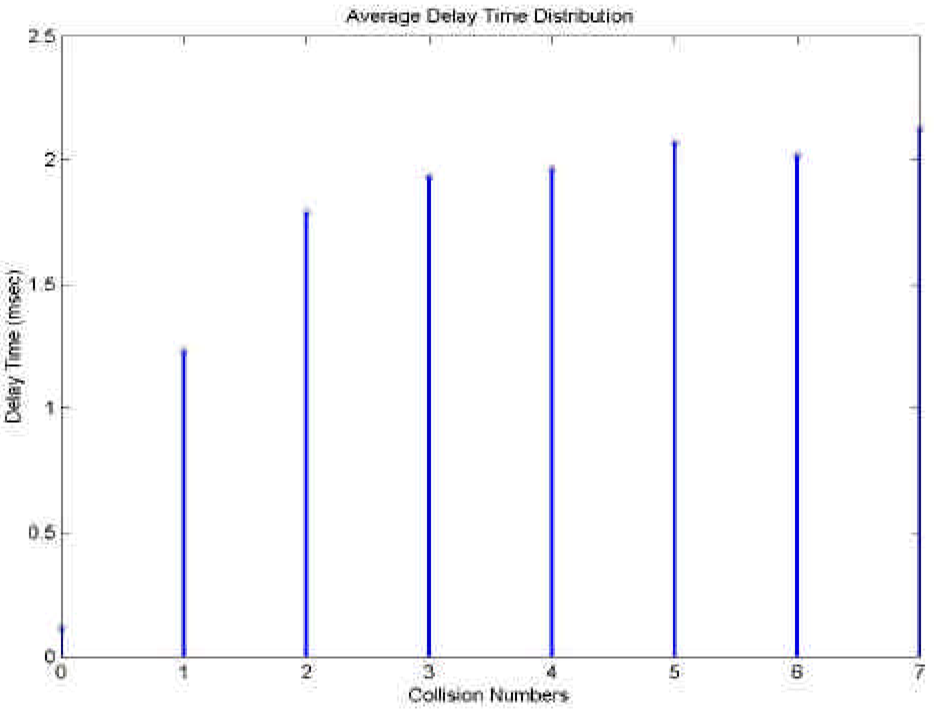

Figure 34. Detail of the average delay time of each collision number under middle loading with the MPDU 1000 bytes

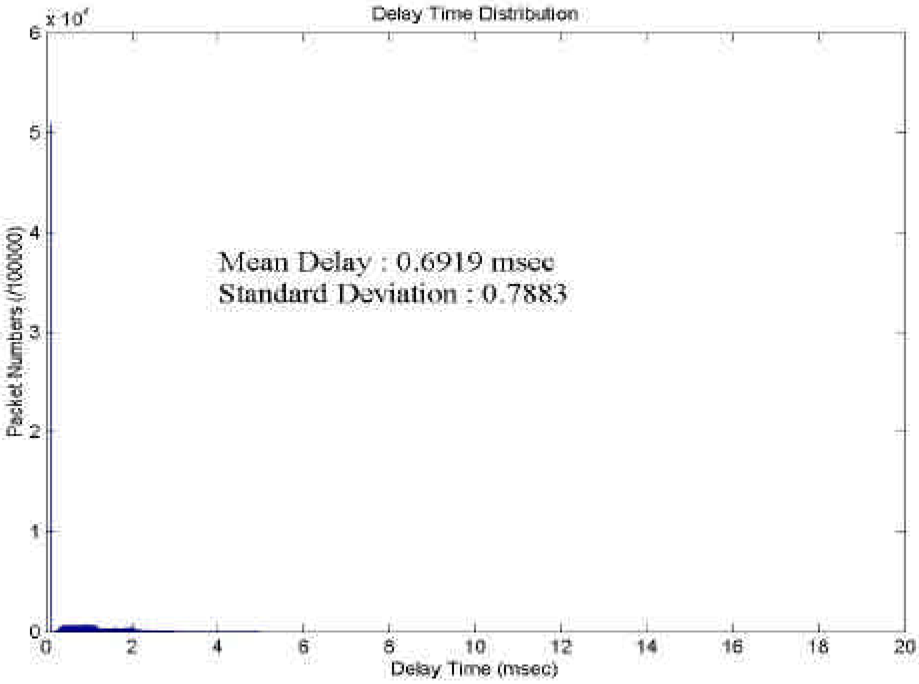

Figure 35. Delay time distribution under middle load with the MPDU 1000 bytes

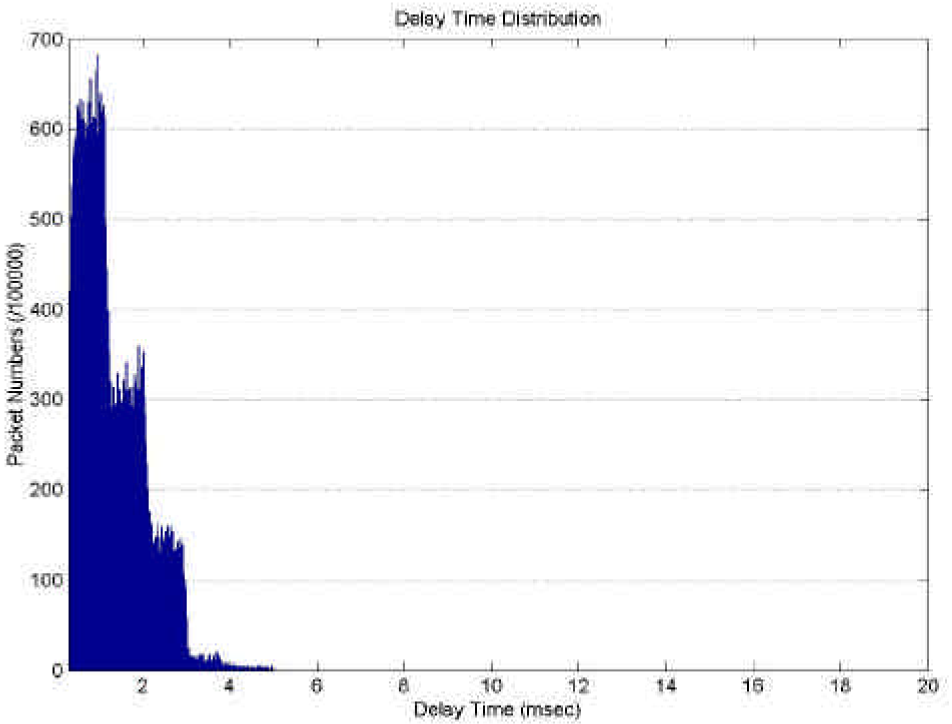

Figure 36. Delay time distribution under middle load with the MPDU 1000 bytes, and without no-collision packets

When we zoom in to get the Figure 34, there is a small ripple between the collision number 5 to the collision number 7. Because the packet number with more collisions is less than that of fewer collisions, we know the sample base is not enough to present the ideal characteristics of the fair algorithm.

The Figure 37 to Figure 40 are the results for the heavy loading. Again, when comparing with the same conditions show in the Figure 17, Figure 18 and Figure 19, we find the mean delay time reduces not much, only 4%, but the standard deviation can be reduced 60%. Of course, the delay time still increases if comparing with the light loading or the middle loading by using the same modified Gamma backoff algorithm. It has to be, because when there are many stations deferring for a busy medium, which is referred to a heavy loading, the probability to get the same backoff timer is higher if the contention window is too limited. When more than two stations select the same backoff integer, the packets will be collided one more time and backoff again. Such a case will reveal worse delay time as we can predict. Based on the algorithm described in the section 3.2, we generate different *aCWmax* and CWS according to the traffic loading to prevent this worse collision condition.

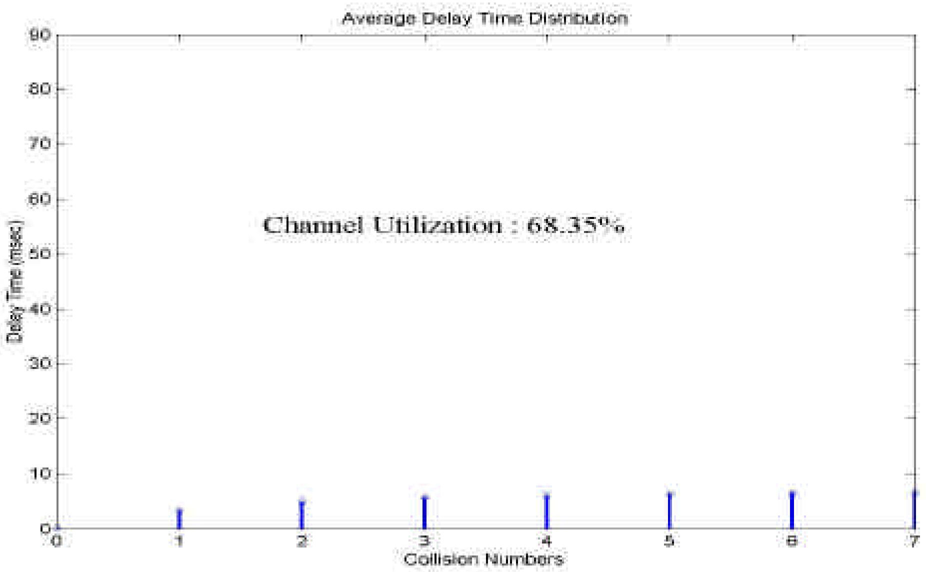

Figure 37. Average delay time of each collision number under heavy load with the MPDU 1000 bytes

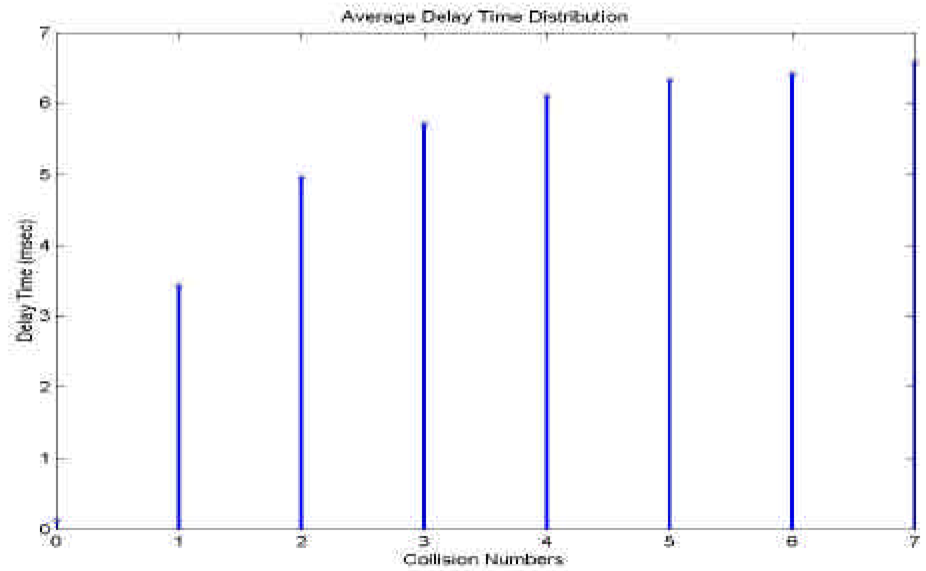

Figure 38. Detail of the average delay time of each collision number under heavy loading with the MPDU 1000 bytes.

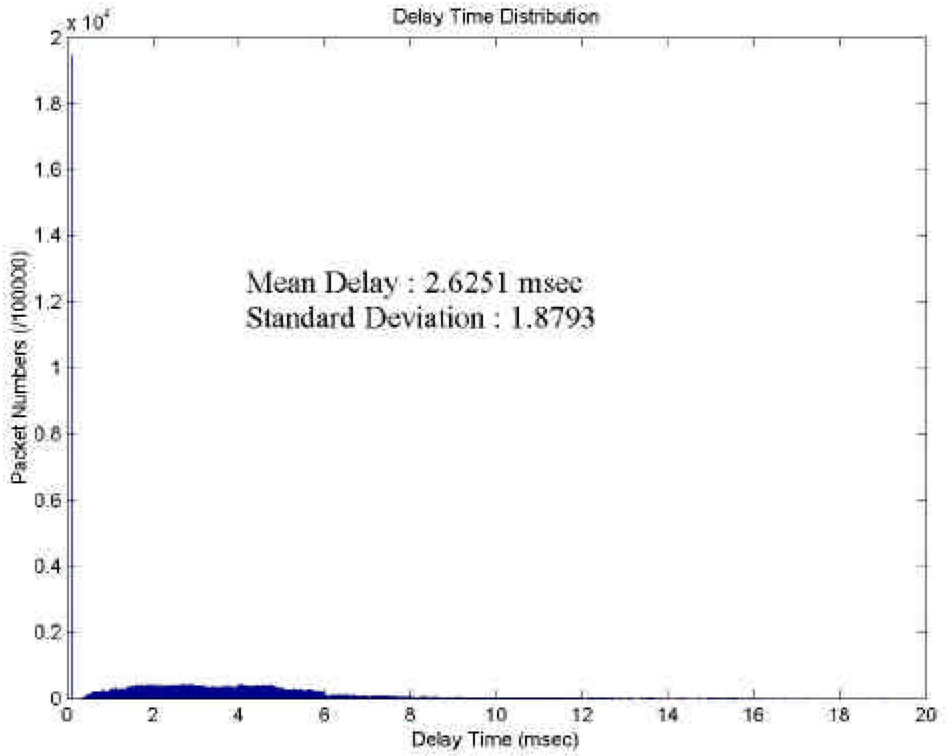

Figure 39. Delay time distribution under heavy load with the MPDU 1000 bytes

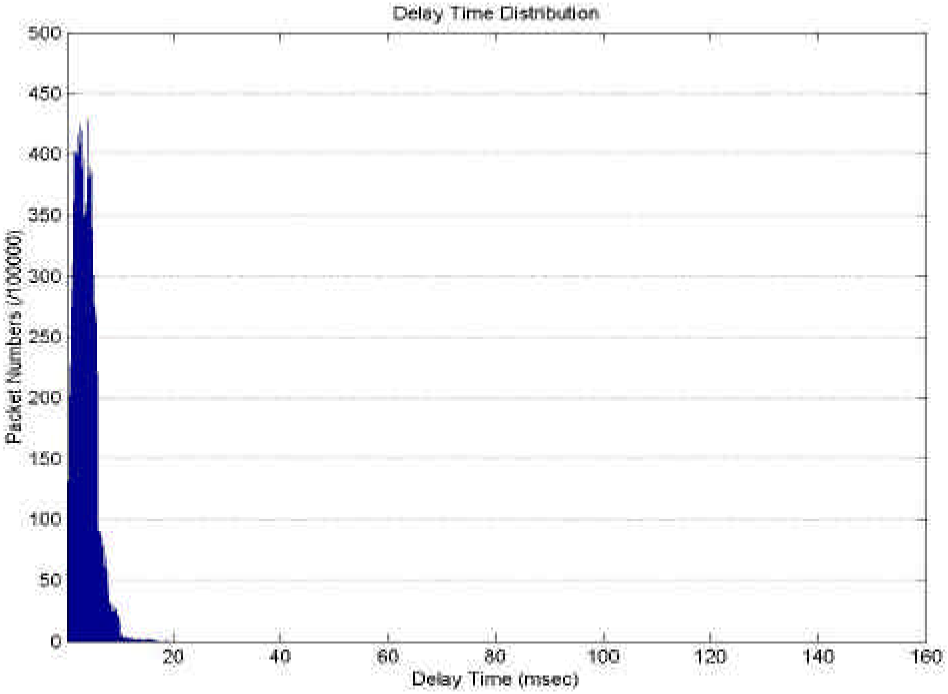

Figure 40. Delay time distribution under heavy load with the MPDU 1000 bytes, and without no-collision packets

The limitation of the throughput is about 70% in both two backoff algorithms according to the Figure 41. Following the (3.12), (3.13), we get (*aCWmax*, *CWS*) for 20 stations as (8, 5), and for 40 stations as (14, 7). Refer to the contention window bounds (*aCWmax*, *aCWmin*), (8, 4) is for 20-station, and (14, 8) is for 40-station. We can tell the throughput is kept close to the one of the conventional backoff algorithm. The fair backoff algorithm cannot enhance the channel utilization because we only try to reduce the backoff time for each collision or every deferment, but not focus on how to lower the handshaking procedure or dummy packets exchange. And, we do not improve the mean delay time of packets under these different loadings. We only restrict the standard deviation of the packet, which also can be referred as the precision time of each packet to achieve the fairness.

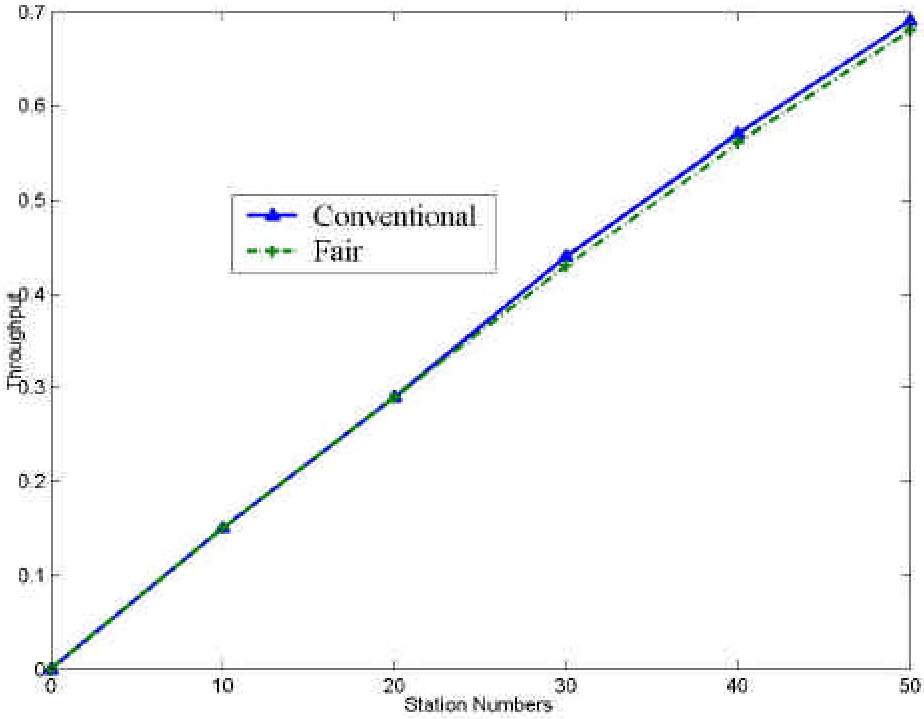

Figure 41. Throughput vs. Station Numbers under two backoff algorithms

We also like to know how is the collision probability when applying the fair backoff algorithm. Similarly, using the upper and lower bounds of contention window as mentioned above, and we take all collided packets into consideration no matter how many times it has been collided, we can get the collision probability distribution over the station numbers and show it in the Figure 42. It is formed that the overall collision is kept almost the same when the fair backoff algorithm is applied. The collision probability is not getting worse. Also, a tiny improvement will even happen when the traffic loading is heavy, because those packets with one collision will choose a larger backoff timer and reduce the possibility to collide another packet.

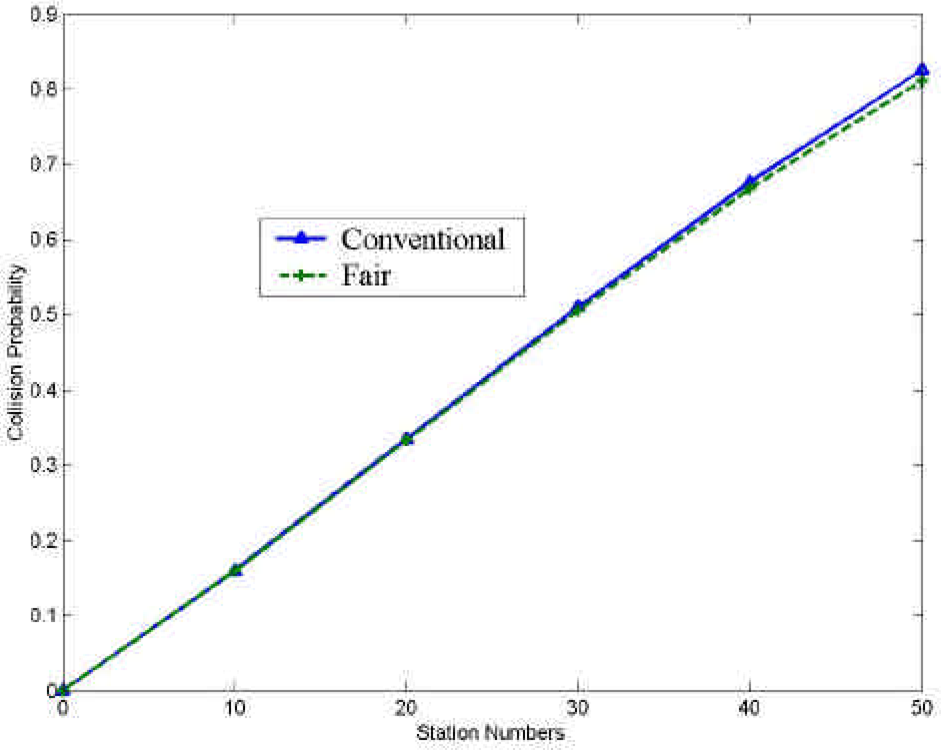

Figure 42. Collision Probability vs. Station Numbers under two backoff algorithms

Chapter 4

Conclusion

What we much concern in a wireless environment is the fairness of each packet. As we can understand, when the traffic is getting heavier, the packet will be collided more, and the delay time will be larger.

In this thesis, we have modeled and simulated the performance of the IEEE 802.11 MAC protocol with the conventional Exponential backoff algorithm. The CSMA/CA mechanism and the RTS/CTS handshaking mechanism are simulated to get the real performance and delay time distribution of each collided packet. One important character of this conventional backoff algorithm is the expansion rate of the delay time under different traffic loadings. This results in an unstable delay time of all kinds of packets, especially under a heavy loading.

We have proposed a modified Gamma backoff algorithm and studied the performance. Our simulations show that although the performance of the protocol may not enhanced much, but the fairness of each packet can be improved. The standard deviation of the delay time is more limited no matter how many collisions happened to this packet and this parameter can be presented as the fairness characteristic.

Reference

1. “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications”, IEEE 802.11, 1999
2. “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications : Higher-Speed Physical Layer Extension in the 2.4 GHz Band,” IEEE 802.11b, 1999
3. “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications : High-speed Physical Layer in the 5 GHZ Band”, IEEE 802.11a, 1999
4. “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 4 : Further Higher Data Rate Extension in the 2.4 GHz Band”, IEEE 802.11g, 2003
5. “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
6. “Information technology – Local and metropolitan area networks – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications”, IEEE 8802-3, 1993
7. S. Khurana, A. Kahol, S.K.S. Gupta and P.K. Srimani , “Performance Evaluation of Distributed Co-Ordination Function for IEEE 802.11 Wireless LAN Protocol in Presence of Mobile and Hidden Terminals”, *MASCOS 1999*, pp.40-47, 1999

8. Giuseppe Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal On Selected Areas In Communications*, vol. 18, no. 3, pp.535-547, Mar. 2000

9. T. Suzuki and S. Tasaka, "Performance Evaluation of Priority-Based Multimedia Transmission with the PCF in an IEEE 802.11 Standard Wireless LAN", *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol.2, pp.G70-G77, Oct. 2001

10. A. Veres, A. T. Campbell, M. Barry, and Li-Hsiang Sun, "Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control", *IEEE Journal On Selected Areas In Communications*, vol. 19, no. 10, pp.2081-2093, Oct. 2001

11. T-C Hou, L-F Tsao and H-C Liu, "Analyzing the Throughput of IEEE 802.11 DCF Scheme with Hidden Nodes", *VTC 2003-Fall*, vol. 5, pp.2870-2874, 6-9 Oct. 2003.

12. S. Khurana, A. Kahol and A. Jayasumana, "Effect Of Hidden Terminals On The Performance Of The IEEE 802.11 MAC Protocol", *In Proceedings of Local Computer Networks Conference*, pp.12-20, 1998

13. T-C Hou, L-F Tsao and H-C Liu, "Throughput Analysis Of The IEEE 802.11 DCF Scheme In Multi-Hop Ad Hoc Networks", *International Conference on Wireless Networks*, June 2003

14. T. Suzuki and S. Tasaka, "Performance Evaluation of Integrated Video and Data Transmission With The IEEE 802.11 Standard MAC Protocol", *Global Telecommunications Conference*, pp.580-586, 1999

15. H. Yamada, H. Morikawa and T. Ayoama, "Decentralized Control Mechanism Suppressing Delay Fluctuation in Wireless LANs", *VTC 2003-Fall*, vol. 2,

pp.801-852, 6-9 Oct.2003

16. P. Gart, R. Doshi, R. Greene, M. Baker, M. Malek, and X. Cheng, “Using IEEE 802.11e MAC for QoS over Wireless”, *Performance, Computing, and Communications Conference*, pp.537-542, 9-11, Apr. 2003.
17. Y-L Kuo, C-H Lu, Eric H-K Wu, G-H Chen, and Y-H Tseng, “Performance Analysis of the Enhanced Distributed Coordination Function in the IEEE 802.11e”, *VTC 2003-Fall*, vol. 5, pp.3488-3492, 6-9 Oct. 2003
18. S. Chakrabarti, and A. Mishra, “QoS Issues in Ad Hoc Wireless Networks”, *IEEE Communications Magazine*, pp.142-148, Feb. 2001
19. S-T Sheu, and T-F Sheu, “A Bandwidth/Allocation/Sharing/ Extension Protocol for Mkultimedia Over IEEE 802.11 Ad Hoc Wireless LANs”, *IEEE Journal On Selected Areas In Communications*, vol. 19, no. 10, pp. 2065-2080, Oct. 2001.
20. M. Bottigliengo, C. Casetti, C.-F. Chiasserini, and M. Meo, “Smart Traffic Scheduling in 802.11 WLANs with Access Point”, *VTC 2003-Fall*, vol. 4, pp.2227-2231, 6-9 Oct. 2003.
21. T. Hautala, T. Braysy, J. Makela, J. Lehtomaki, and T. Saarinen, “Scalability of Mobility Signaling in IEEE 802.11 WLAN”, *VTC 2003-Fall*, vol. 3, pp.2044-2047, 6-9 Oct. 2003
22. J. W. Tantra, C-H Foh, and B-S Lee, “An Efficient Scheduling Scheme for High Speed IEEE 802.11 WLANs”, *VTC 2003-Fall*, vol. 4, pp.2589-2593, 6-9 Oct. 2003
23. Y. M. Ghamri-Doudane, R. Naja, G. Pujolle, and S. Thome, “P3-DCF: Service Differentiation in IEEE 802.11 WLANs using Per-Packet Priorities”, *VTC 2003-Fall*, vol. 5, pp.3429-3433, 6-9 Oct. 2003
24. M. Cesana, D. Maniezzo, P. Bergamo, and M. Gerla, “Interference Aware (IA)

MAC: an Enhancement to IEEE 802.11b DCF", *UCLA Computer Sceience Department Technical Report 020037*

25. Y. Li, X. Wang, and S. A. Mujtaba, "Co-channel Interference Avoidance Algorithm in 802.11 Wireless LANs", *VTC 2003-Fall*, vol. 4, pp.2610-2614, 6-9 Oct. 2003
26. M. Natkaniec and A. R. Pach, "An Analysis of the Backoff Mechanism used in IEEE 802.11 Networks", *Computers and Communications 5th IEEE Symposium*, pp.444-449, 3-6 July 2000
27. L. Bononi, M. Conti, and E. Gregori, "Design and Performance Evaluation of an Asymptotically Optimal Backoff Algorithm for IEEE 802.11 Wireless LANs", *Proceedings of the 33rd Hawaii International Conference on System Sciences 2000*, vol. 2, pp.1-10, 4-7 Jan. 2000
28. F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism", *IEEE Journal On Selected Areas in Communications*, vol. 18, no. 9, pp.1774-1786, Sep. 2000.
29. N-O. Song, B-J Kwak, J. Song, and L. E. Miller, "Enhancement of IEEE 802.11 Distributed Coordination Function with Exponential Increase Exponential Decrease Backoff Algorithm", *VTC 2003-Spring*, vol. 4, pp.2775-2778, Apr. 2003
30. Y. Xiao, "Backoff-based Priority Schemes for IEEE 802.11", *IEEE International Conference 2003*, vol. 3, pp.1568-1572, 11-15 May 2003
31. R. L. Scheaffer, "Introduction to Probability and Its Applications", PWS-KENT, 1990
32. A. Papoulis, "Probability, Random Variables, and Stochastic Processes", McGraw-Hill, Third Edition, New York, 1991