Protection and Guarantee for Voice and Video Traffic in IEEE 802.11e Wireless LANs

Yang Xiao
Computer Science Division
The University of Memphis
Memphis, TN 38152
E-mail: *yangxiao@ieee.org*

Haizhon Li
Computer Science Division
The University of Memphis
Memphis, TN 38152
E-mail: *hli1@memphis.edu*

Sunghyun Choi
School of Electrical Engineering
Seoul National University
Seoul, 151-744, Korea
E-mail: *schoi@snu.ac.kr*

***Abstract*—In order to support multimedia applications such as voice and video over the wireless medium, a contention-based channel access function, called Enhanced Distributed Coordination Function (EDCF), is being developed in the emerging standard IEEE 802.11e. In EDCF, differentiated services are provided for different traffic classes. In this paper, we propose a *two-level* protection and guarantee mechanism for voice and video traffic in IEEE 802.11e Wireless LANs. In the first-level protection, the existing voice and video flows are protected from the new and other existing voice and video flows. In the second-level protection, the voice and video flows are protected from the best-effort data traffic. For each protection level, a couple of protection mechanisms are proposed. Extensive simulation results show that the proposed two-level protection and guarantee mechanism is very effective in terms of protecting and guaranteeing existing voice and video flows as well as fully utilizing the channel capacity.**

***Index Terms*—IEEE 802.11, wireless LAN, admission control, medium access control (MAC), Quality of Service (QoS).**

I. Introduction

IEEE 802.11 medium access control (MAC) employs a mandatory contention-based channel access function called Distributed Coordination Function (DCF), and an optional centrally controlled channel access function called Point Coordination Function (PCF) [1]. The DCF adopts a carrier sense multiple access with collision avoidance (CSMA/CA) with binary exponential backoff. It is considered a wireless version of the most successful local area network (LAN), IEEE 802.3 (Ethernet), which adopts a CSMA with collision detection (CSMA/CD) with binary exponential backoff. Both IEEE 802.11 DCF and IEEE 802.3 enable fast installation with minimal management and maintenance costs, and are very robust protocols for the best–effort service. The popularity of the IEEE 802.11 wireless LAN (WLAN) is due mainly to the DCF, whereas the PCF is barely implemented in today's products due to its complexity and inefficiency for the normal data transmissions, even though it has some limited QoS support. Furthermore, the PCF may cause unpredictable beacon delays and unknown transmission durations of the polled stations [5].

However, the current DCF is unsuitable for multimedia applications with QoS requirements [6]–[8]. Under the DCF, a station might have to wait an arbitrarily long time to send a frame so that real-time applications such as voice and video may suffer [14]. One possible solution is to provide a good priority scheme for DCF. Simple DCF priority schemes can be easily designed with minor changes in DCF, and they are quite effective [7], [8]. Many studies have been reported in the literature for the priority supporting capability of the DCF [7], [8], [14]–[17]. Prioritized QoS will be useful for those multimedia applications that can live without rigid QoS. One advantage of prioritized QoS is that it is simple to implement and looks like DiffServ model in the IP networks [6], [11]. Note that QoS control is necessary at the MAC layer, and it should be designed in a similar way designed for IP.

To support the MAC-level QoS, the IEEE 802.11 Working Group is currently working on the standardization of IEEE 802.11e [2], which is in the final stage. The emerging IEEE 802.11e standard provides QoS features and multimedia support to the existing 802.11b [3] and 802.11a [4] WLANs, while maintaining full backward compatibility with these standards. The IEEE 802.11e MAC employs a channel access function, called Hybrid Coordination Function (HCF), which includes both contention-based channel access and centrally-controlled channel access mechanisms. The contention-based channel access mechanism is also referred to as Enhanced Distributed Coordination Function (EDCF). The EDCF provides a priority scheme by differentiating the inter-frame space, the initial and the maximum contention window sizes for backoff procedures.

In the previous work in [5]–[13], which the authors were involved, the main focus was on studying the EDCF mechanisms and differentiated services. However, without a good admission control mechanism and a good protection mechanism, the existing multimedia traffic cannot be protected and QoS requirements cannot be met, and that fact motivated our current work.

In this paper, we propose a *two-level* protection and guarantee mechanism for voice and video traffic. At the first-level protection, the existing voice and video flows are protected from the new and other existing voice and video flows. We first introduce a distributed admission control, which is a revised version based on the IEEE 802.11e draft [2], for differentiation services of EDCF, in which channel utilization measurements are conducted during each beacon interval, and available/residual budgets are calculated. When one class' budget becomes zero, new traffic streams (or sessions) belonging to this class cannot gain transmission time anymore, and existing nodes will not be allowed to increase the transmission time that they are already using. Therefore, the existing traffic streams are protected and the channel capacity is fully utilized. It is challenging to design a good admission control at MAC layer to protect existing multimedia traffic and to fully utilize the system capacity due to the contention-based nature.

According to our simulations, the above admission control can protect existing voice/video flows only when the traffic load is not very heavy. Therefore, we propose two enhancements: a *tried-and-known* method and an *early-protection* method. In the tried-and-known method, a new voice/video flow is first accepted tentatively, and then tries to measure throughput and delay performances for some beacon intervals. If the average throughput and/or delay do not meat reasonable requirements, the flow will kill/reject itself. In the early-protection method, when the budget is below a certain threshold, new flows are not allowed to enter. Through extensive simulations, we show that this first-level protection, i.e., the admission control coupled with two enhancements, protects and guarantees the existing voice and video flows from new and other existing voice and video flows quite well.

However, even though much of the channel capacity can be used by existing voice and video, too many unsuccessful best-effort data transmissions can degrade the existing voice and video flows since many data transmissions may cause many collisions. The existing voice and video flows become vulnerable to data traffic. Accordingly, we propose the second-level protection, in which the existing voice and video flows are protected from the best-effort data traffic. In this method, we attempt to dynamically control EDCF channel access parameters so that when the number of active stations is large, the number of collisions will be kept relatively small by increasing the initial contention window size and inter-frame space for the best-effort data traffic. Therefore, the second-level protection can be achieved.

The main contribution of this paper is to propose and study such a two-level protection and guarantee mechanism for IEEE 802.11e. To our best knowledge, such efforts have never been pursued in the literatures so far.

The rest of paper is organized as follows. We briefly introduce IEEE 802.11 DCF and 802.11e EDCF in Section II and Section III, respectively. The first-level protection and guarantee mechanisms are presented in Section IV. Section V presents the second-level protection and guarantee mechanisms. Performance studies are carried out in Section VI with extensive simulation results. We conclude our paper in Section VII.

II. IEEE 802.11 DCF

IEEE 802.11 MAC employs a mandatory DCF and an optional PCF. In a long run, time is divided into repetition intervals called *superframes*. Each superframe starts with a beacon frame, and the remaining time is further divided into a contention-free period (CFP) and a contention period (CP). The DCF works during the CP and the PCF works during the CFP. If the PCF is not active, superframes do not exist. However, the beacon frames are periodically transmitted irrespectively. The beacon frame is a management frame for synchronization, power management, and delivering network operation parameters. Beacon frames are generated in regular intervals called target beacon transmission times (TBTTs).

The DCF defines a basic access mechanism and an optional request-to-send/clear-to-send (RTS/CTS) mechanism. Under the DCF, a station with a frame to transmit monitors the channel activities until an idle period equal to a distributed inter-frame space (DIFS) is detected. After sensing an idle DIFS, the station waits for a random backoff interval before transmitting. The backoff time counter is decremented in terms of slot time as long as the channel is sensed idle. The counter is suspended when a transmission is detected on the channel, and resumed with the old remaining backoff interval when the channel is sensed idle again for a DIFS interval. The station transmits its frame when the backoff timer reaches zero. For each new transmission attempt, the backoff interval is uniformly chosen from the range [0, CW–1] in terms of timeslots, where CW is the current backoff window size. At the very first transmission attempt, CW equals the initial backoff window size CW_{min}. After each unsuccessful transmission, CW is doubled until a maximum backoff window size value CW_{max} is reached. After the destination station successfully receives the frame, it transmits an acknowledgment frame (ACK) following a short inter-frame space (SIFS) time. If the transmitter station does not receive an ACK within a specified ACK Timeout, it reschedules the frame transmission according to the backoff rules discussed above.

The DCF provides a channel access mechanism with equal probabilities to all stations contending for the same wireless medium. If an AP is present, stations are not allowed to transmit frames to another station directly, and hence the receiver is always its AP.

III. IEEE 802.11E EDCF

IEEE 802.11e provides a channel access function, called Hybrid Coordination Function (HCF) to support applications with QoS requirements. The HCF includes both contention-based channel access and centrally controlled channel access schemes. The contention-based channel access of the HCF is also referred to as Enhanced Distributed Coordination Function (EDCF). In this paper, we only consider EDCF since (1) it is simpler, and (2) it is expected to support many QoS applications, which do not require strict QoS provisioning [6].

A new concept, transmission opportunity (TXOP), is introduced in IEEE 802.11e. A TXOP is a time period when a station has the right to initiate transmissions onto the wireless medium. It is defined by a starting time and a maximum duration. A station cannot transmit a frame that extends beyond a TXOP. If a frame is too large to be transmitted in a TXOP, it should be fragmented into smaller frames.

The EDCF works with four Access Categories (ACs), which are virtual DCFs as shown in Fig. 1, where each AC achieves a differentiated channel access. This differentiation is achieved through varying the amount of time a station would sense the channel to be idle and the length of the contention window for a backoff. The EDCF supports eight different priorities, which are further mapped into four ACs, shown in Table I. Differentiated ACs are achieved by differentiating the arbitration inter-frame space (AIFS), the initial window size, and the maximum window size. That is, for AC i $(i = 0,...,3)$, the initial backoff window size is $CW_{\min}[i]$, the maximum backoff window size is $CW_{\max}[i]$, and the arbitration inter-frame space is $AIFS[i]$. For $0 \le i < j \le 3$, we have $CW_{\min}[i] \ge CW_{\min}[j]$, $CW_{\max}[i] \ge CW_{\max}[j]$, and $AIFS[i] \ge AIFS[j]$, and at least one of above inequalities must be "not equal to". In other words, the EDCF employs $AIFS[i]$, $CW_{\min}[i]$, and $CW_{\max}[i]$ (all for $i = 0,...,3$) instead of DIFS, $CW_{\min}$, and $CW_{\max}$, respectively. If one AC has a smaller $AIFS$ or $CW_{\min}$ or $CW_{\max}$, the AC's traffic has a better chance to access the wireless medium earlier. Fig. 2 shows the EDCF timing diagram, where 3 ACs are shown: *i*, *j* and *k*.

Fig. 1 shows four transmission queues implemented in a station, and each queue supports one AC, behaving roughly as a single DCF entity in the original IEEE 802.11 MAC. It is assumed that a payload from a higher layer is labeled with a priority value, and it is enqueued into the corresponding queue according to the mapping in Table I. Each queue acts as an independent MAC entity and performs the channel access with a different inter-frame space ($AIFS[i]$), a different initial window size ($CW_{\min}[i]$), and a different maximum window size ($CW_{\max}[i]$). Each queue has its own backoff counter ($BO[i]$), which acts independently in the same way as the original DCF backoff counter. If there is more than one queue finishing the backoff at the same time, the highest AC frame is chosen to transmit by the virtual collision handler. Other lower AC frames whose backoff counters also reach zero will increase their backoff counters with $CW_{\min}[i]$ ($i = 0,...,3$), accordingly. Furthermore, we have $AIFS[i] \ge PIFS$, where $PIFS$ is point (coordination function) inter-frame space.

The values of $AIFS[i]$, $CW_{min}[i]$, and $CW_{max}[i]$ (all for i=0,…,3), are referred to as the EDCF parameters, which will be announced by the QoS Access Point (QAP) via periodically transmitted beacon frames. The QAP can also adaptively adjust these EDCF parameters based on the network traffic conditions.

TABLE 1
PRIORITY TO ACCESS CATEGORY MAPPING

PRIORITY	AC	DESIGNATION
1	0	BEST EFFORT
2	0	BEST EFFORT
0	0	BEST EFFORT
3	1	VIDEO PROBE
4	2	VIDEO
5	2	VIDEO
6	3	VOICE
7	3	VOICE

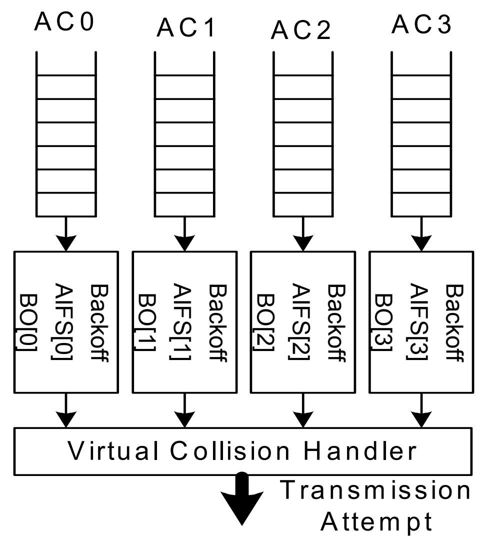


Fig. 1 Virtual transmission queues, where BO[i] stands for the backoff counter for AC *i*

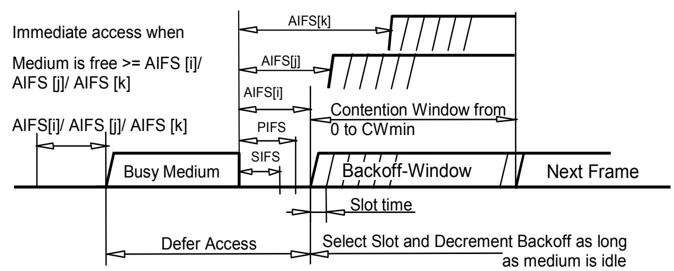


Fig. 2 EDCF timing diagram

IV. THE FIRST-LEVEL PROTECTION AND GUARANTEE

In this section, we propose the first-level protection and guarantee, i.e., protection and guarantee of the existing voice and video flows from the new and other existing voice and video flows. In Section IV.A, we introduce a distributed admission control scheme for voice and video, which is a revised version of that defined in [2]. We also propose two additional enhancements for the admission control algorithm in Section IV.B and Section IV.C, respectively.

A. Distributed Admission Control for EDCF

The distributed admission control (DAC) is developed to protect active QoS flows, such as voice and video. The QAP announces the transmission budget via beacons for each AC (except AC 0). Note that AC 0 supports the best-effort data traffic that will not be protected. The budget indicates the allowable transmission time per AC in addition to what is being utilized. QoS Stations (QSTAs) determine an internal transmission limit per AC for each beacon interval, based on the transmission count during the previous beacon period and the transmission budget announced from the QAP. The local voice/video transmission time per beacon interval shall not exceed the internal transmission limit per AC. When the transmission budget for an AC is depleted, new flows will not be able to gain transmission time, while existing flows will not be able to increase the transmission time per beacon interval, which they are already using. This mechanism protects existing flows.

1) Procedure at QAP

The QoS Parameter Set Element (QPSE) provides information needed by QSTAs for a proper operation of the QoS facility during the contention period. The QPSE includes $CW_{min}[i]$, $CW_{max}[i]$, $AIFS[i]$, for (i=0,…,3), and $TXOPBudget[i]$ and $SurplusFactor[i]$, for (i=1,2,3). These are global variables in the sense that they are maintained by QAP and transmitted to QSTAs via beacons. The first three variables/parameters were already discussed in the previous section. $TXOPBudget[i]$ specifies the additional amount of time available for AC i during the next beacon interval, and $SurplusFactor[i]$ (> 1) represents the ratio of over-the-air bandwidth reserved for AC i to bandwidth of the transported frames required for successful transmission. Note that bandwidth more than the minimum required is typically reserved to compensate potential transmission failures, e.g., due to collisions. The QPSE is calculated by the QAP for each beacon interval and embedded into the next beacon frame.

The QAP shall measure the amount of time occupied by transmissions from each AC during the beacon period, including associated SIFS and ACK times if applicable. The QAP shall maintain a set of counters $TxTime[i]$, which shall be set to zero immediately following transmission of a beacon. For each data frame transmission (either uplink or downlink), the QAP shall add the time, equal to the frame transmission time and all overhead involved such as SIFS and ACK, to the *TxTime* counter corresponding to the AC of that frame. The QAP determines $TXOPBudget[i]$ by:

$$TXOPBudget[i] = \max(ATL[i] - TxTime[i] \times SurplusFactor[i], 0) \quad (1)$$

where $ATL[i]$ is for the maximum amount of time that may be used for transmissions of AC i, per beacon interval.

2) Procedure at Each QSTA

When the transmission budget for an AC is depleted, new QSTAs cannot gain transmission time, while existing QSTAs cannot increase the transmission time per beacon interval, which they are already utilizing. Accordingly, this mechanism protects existing flows.

Each QSTA has to maintain the following local variables for each AC: $TxUsed[i]$, $TxSuccess[i]$, $TxLimit[i]$, $TxRemainder[i]$, and $TxMemory[i]$. These are local variables in the sense that each station locally updates these variables by counting only those related to itself. $TxUsed[i]$ counts the amount of time occupied on-air by transmissions, irrespective of success or not, from AC i of this station, including associated SIFS and ACK times if applicable. $TxSuccess[i]$ counts for the transmission time for successful transmissions. A station shall not transmit a data frame if doing so would result in the value in $TxUsed[i]$ exceeding the value in $TxLimit[i]$, where how to determine this value is presented below. If the QSTA is prevented from sending a frame for this reason, it may carry over the partial frame time remainder to the next beacon interval, by storing the remainder in $TxRemainder[i]$, where $TxRemainder[i] = TxLimit[i] - TxUsed[i]$; Otherwise, $TxRemainder[i] = 0$. $TxMemory[i]$ 'memorizes' the amount of resource that AC i of this station utilized during a beacon interval. Let f denote the damping factor whose function will be explained below. At each target beacon transmission time, the *TxMemory*, *TxLimit* and *TxSuccess* variables are updated according to the following procedure:

- If $TXOPBudget[i] = 0$,
  - Both $TxMemory[i]$ and $TxRemainder[i]$ shall be set to zero for new QSTAs which start transmission with this AC in the next beacon interval. All other QSTAs' $TxMemory[i]$ remains unchanged;
- Else if the $TXOPBudget[i] > 0$,
  - For new QSTAs, which start transmission with this AC in the next beacon interval, an initial value for $TxMemory[i]$ could be a number between 0 and $TXOPBudget[i]/SurplusFactor[i]$. All other QSTAs' $TxMemory[i]$ are updated according to the following procedure:
    - $TxMemory[i] = f \times TxMemory[i] + (1-f) \times (TxSuccess[i] \times SurplusFactor[i] + TXOPBudget[i])$;
- $TxSuccess[i] = 0$;
- $TxLimit[i] = TxMemory[i] + TxRemainder[i]$;

Note that in the above procedure, only $TXOPBudget[i]$ and $SurplusFactor[i]$ are global variables, and the others are local variables. From the above procedure, when the transmission budget for an AC becomes zero,

- Its $TxLimit[i]$ will become zero for new STAs, and hence AC i of any new QSTA will not be able to gain a transmission time in the next beacon interval.
- The existing QSTAs' $TxMemory[i]$ remains unchanged, and hence the existing QSTAs' $TxLimit[i]$ remains basically unchanged. In other words, existing stations will not be able to increase the transmission time above what they are already using. Note that this mechanism protects existing flows.

From the above procedure, as long as the transmission budget is larger than zero, both $TxMemory[i]$ and $TxLimit[i]$ need be adjusted periodically. The new $TxMemory[i]$ value is a weighted average of the old $TxMemory[i]$ value and the sum of the successful transmission time and the budget. The value $TxSuccess[i] \times SurplusFactor[i] + TXOPBudget[i]$ is the target to which *TxMemory* converges. The *TxLimit* is equal to *TxMemory* plus a possible capped remainder, where *TxMemory* 'memorizes' the amount of time, which a specific AC of the QSTA has been able to utilize per beacon interval. Once the budget is depleted (i.e., *TXOPBudget* hovers around 0), *TxMemory* converges to *TxSuccess*, which is the lower limit. This ensures that a QSTA can continue consuming the same amount of time in subsequent beacon intervals. The damping allows for some amount of fluctuation to occur. However, *TxMemory* cannot grow any further in the saturated state. This prevents new flows from entering a specific AC when it is saturated.

The damping factor does not affect the entrance of a new flow into the system when an enough budget is available, because the decreased *TXOPBudget* is offset by an increased

TxSuccess instantaneously, so *TxMemory* does not change a lot. The damping factor does affect *TxMemory* when a new flow starts up in a QSTA, which does not have an existing flow of the corresponding AC. In such a case, the decreased *TXOPBudget* is not offset by an increased *TxSuccess*, and the *TxMemory* converges to the lower target value consequently. QSTAs shall not increase their *TxLimit*[*i*] if they did not transmit traffic of AC *i* during the previous beacon interval.

For each video/voice flow, a Leaky-Bucket algorithm plus a Token-Bucket algorithm can be also implemented at the QSTA to control the flow rate.

B. Enhancement with Required Throughputs and/or Delays

In this subsection, we attempt to enhance the above distributed admission control considering the required throughput and/or delay performance. The basic idea behind this scheme is that by observing several beacon intervals, the information whether the currently-available capacity can accept a new flow can be determined. We refer to this as a *tried-and-known* method. In this scheme, we assume that the required minimum throughput T_{min} and/or the maximum tolerable delay D_{max}, both per AC, are passed from the higher layer. This algorithm is a superset of the algorithm presented in Section IV.A. That is, it includes all the features of the previously-presented algorithm, as well as some new features as follows. We denote the DAC plus an Enhancement with required Throughput and/or Delay (ETD) as DAC+ETD. Note that this enhancement is only applied to new flows during the starting phase. At each of the very first *k* beacon intervals for a newly-started flow, a new flow measures its *Throughput*[*j*] and *Delay*[*j*] of the *j*-th measurement. Then, the additional procedure works as follows:

If $\frac{\sum_{j=1}^{k} Throughout[j]}{k} \le \alpha T_{\min}$ and/or $\frac{\sum_{j=1}^{k} Delay[j]}{k} \ge \beta D_{\max}$ where $0 < \alpha < 1$ and $\beta \ge 1$, then this flow rejects itself. In other words, for those flows with throughput and/or delay requirements, if these requirements cannot be satisfied during the first *k* beacon intervals, the flows will kill/reject themselves. Note that this enhancement does not need to be applied to existing voice and video flows. Therefore, utilization can be maintained at a reasonable level, and the remaining bandwidth can be used for the best-effort data traffic so that there is nothing wasted.

C. Enhancement with a Non-Zero Budget Value

In this subsection, we present an enhancement of DAC with an *early-protection* method. In this scheme, when the budget is below some threshold, new flows are not allowed to enter. It is referred to as the Enhancement with a Non-zero Budget value for new flows (ENB). We refer to the new revision as DAC+ENB, and it works as follows.

For the existing flows, the corresponding procedure at a QSTA is the same as the one in Section IV.A. Now, for a new flow, let *Required_Budget*[*i*] denote the required budget for a new flow, and let ϕ (< 1) denote a fraction. The corresponding procedure at a QSTA is,

- If *TXOPBudget*[*i*] < ϕ x *Required_Budget*[*i*],
  - Both *TxMemory*[*i*] and *TxRemainder*[*i*] shall be set to zero for a new QSTA which starts transmission with this AC in the next beacon interval;
- Else if *TXOPBudget*[*i*] ≥ ϕ x *Required_Budget*[*i*],
  - An initial value of *TxMemory*[*i*] is set to between 0 and *TXOPBudget*[*i*] / *SurplusFactor*[*i*].

We refer to the scheme with both ETD and ENB as DAC+ETD+ENB. A situation when DAC+ETD+ENB is better than DAC+ENB is exampled as follows: when the budget for one traffic type (either voice or video) is enough for only one new flow, two or more new flows may attempt to enter at the same beacon interval. DAC+ENB cannot prevent new flows to enter the system, but DAC+ETD+ENB can. Note that the remaining bandwidth will be used by the best-effort data traffic (AC 0). Therefore, we refer DAC+ETD+ENB as the first-level protection and guarantee mechanism.

V. The Second-Level Protection and Guarantee

According to our simulation results, which will be presented later, even thought much of the channel capacity can be used by existing voice and video, too many unsuccessful data transmissions could degrade the performance of existing voice and video flows since many data transmissions may cause many collisions. The existing voice and video flows become vulnerable to data traffic. The reason behind this is that priority supports are not strict but relative, and conducted stochastically. In this section, we propose the second-level protection and guarantee, i.e., protection and guarantee of the existing voice and video flows from data traffic. One may wonder the reasons why we cannot use a method similar to the first-level protection to control data traffic, i.e., using *TxLimit*[0] to control data traffic. However, first, data traffic does not typically involve flows with stationary traffic amount, and therefore we cannot use *TxLimit*[0]. Even though some data traffic has a form of flows, e.g., an FTP session, we can normally assume that data traffic does not have flows since there are not much delay constraints in consecutive data frames. Secondly, even if we assume that data traffic has flows, the admission control with *TxLimit*[0] for data traffic will cause unfairness among stations: new stations cannot transmit data traffic, thus suffering from starvation if all the budgets for data traffic (AC 0) are used up by the existing stations. Furthermore, the longer of the total successfully transmitted frames a station had previously, the better chance the station may have for transmitting more frames later.

To control data transmissions, we observe through simulations that the most effective way is to reduce the number of collisions or collision probability, caused by the data transmissions. However, we cannot control the number stations accessing to the wireless medium for data transmissions since otherwise it will cause unfairness among stations. Moreover, we cannot know the accurate number of active stations for data transmissions, as well as the associated data transmission rate. Our goal is to control the number of collisions or collision probability independent of the number of active stations for data transmissions. Our approach is to dynamically control data traffic's parameters (i.e., *AIFS*[0], CW_{min}[0], and CW_{max}[0]) based on data traffic load.

In the original MAC, when a collision occurs, the contention window size increases by the factor of 2, which is referred to as the window-increasing factor [7]. Similar to [7], we define the window-increasing factor σ to be any real number larger than 1. Different from [7], here we let σ change with the backoff stage. Let σ_i denote the window-increasing factor for the backoff stage-*i* ($i = 1,\ldots,L_{retry}$), where L_{retry} is the retry limit. In the proposed method, we define the following rules:

- *Fast-backoff***:** we let $2 \le \sigma_1 < \ldots < \sigma_{L_{retry}}$. Compared to the original binary exponential backoff, the proposed backoff method achieves a larger window size much quicker, and becomes faster when the backoff stage is large.
- *Dynamically adjusting parameters when fail:* when a fame reaches the retry limit and is dropped, we do following parameter adjustment until a limit is reached:
  - $CW_{\min}[0] = \theta \times CW_{\min}[0] \ (\theta > 1)$
  - $AIFS[0] = \psi \times AIFS[0] \ (\psi > 1)$
- *Dynamically adjusting parameters when consecutive successful:* when a station successfully transmits *m* consecutive frames, we perform the following parameter adjustment until the original low limit is reached:
  - $CW_{\min}[0] = CW_{\min}[0]/\theta \ \ (\theta > 1)$
  - $AIFS[0] = AIFS[0]/\psi \ \ (\psi > 1)$

We refer to this approach as the Fast-Backoff (BF) plus Dynamic Adjustment when Fail or Successful (DAFS): BF+DAFS.

VI. PERFORMANCE EVALUATIONS

In our simulations, we have three traffic types: voice (AC 3), video (AC 2), and data (AC 0). We have the following parameters unless stated otherwise: *AIFS*[3] = 25µs; *AIFS*[2] = 25µs; *AIFS*[0] = 34µs; *CWmin*[3]=16; *CWmin*[2] = 32; *CWmin*[0] = 256; *CWmax*[3] = 256; *CWmax*[2] = 2048; *CWmax*[0] = 51200; (except for section VI.A, where we use *CWmin*[3]=8; *CWmax*[3] = 1024); beacon interval is 100ms; damping factor is 0.9; for *i* = 2, 3, *SurplusFactor*[*i*] is 1.1, and the initial value of *TxMemory*[*i*] is *0.8* x *TXOPBudget*[*i*] / *SurplusFactor*[*i*]. The queue size for each AC per station is set to 30 frames in our simulations. Each voice flow is 0.0832Mbps, which is generated by a constant inter-arrival time 20ms with a fixed payload size of 208 bytes, corresponding to G.711-coded VoIP over RTP/UDP/IP/SNAP [18]. Each video flow is 4.86Mbps, which is generated by a constant inter-arrival time 2.5ms with a mean payload size of 1464 bytes [6]. It corresponds to a traffic-shaped CBR video flow. Each station generates data frames with an exponential distribution with a mean inter-arrival time 12ms and a fixed payload size of 1500 bytes. We adopt IEEE 802.11a, and both the data rate and control rate are 54Mbps. We assume that all the stations are within the transmission range. In subsection VI.A, the simulation time is 48*500ms, i.e., 24s, and in later subsections, the simulation time is 100*2s, i.e., 200s.

A. Video Traffic Only With and Without DAC

In this simulation, we simulate only video flows with and without DAC. Initially, there is only one video flow. For every 3s, a new video flow is added. *ATL*[2] is 0.7 x (beacon interval).

Fig. 3 shows the throughputs of video flows without and with DAC. We observe that without DAC [Fig. 3 (a)], the throughputs of video flows are messed up, especially, when more video flows are added. With DAC in Fig. 3 (b), we observe that the throughputs are improved in term of guaranteed throughput: only at very end of the simulations, there is a little messed up, but still much better than those without DAC. From Fig. 3 (b), we also observe that later flows do not get the required bandwidth (i.e., 4.86Mbps) and mess up the existing flows a little bit at the very end. We will show that throughputs can be further improved by the proposed enhancements in later figures.

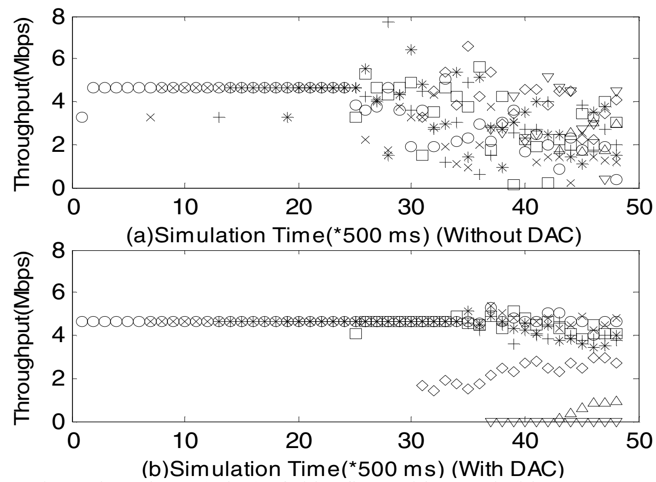


Fig. 3 Throughputs (Mbps) of video flows without and with DAC (different symbols represent different flows)

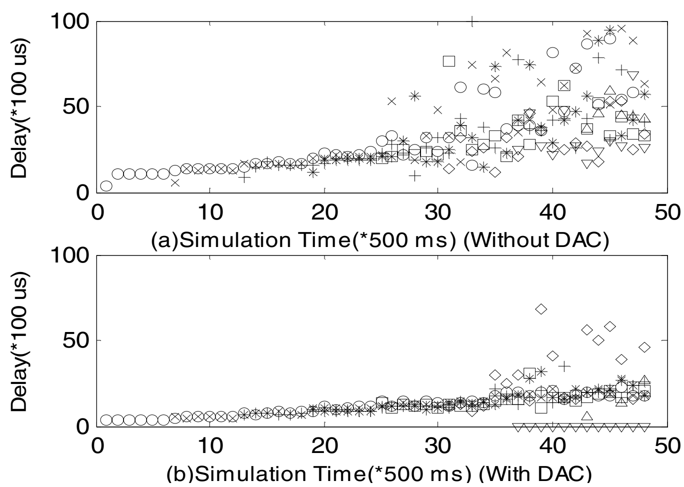


Fig. 4 Delays (in 100µs) of video flows without and with DAC

Fig. 4 shows the delays without and with DAC, where the delay is defined as the time interval from the time when a frame arrives at the front of the queue to the time when it is received by the receiver. The delay only considers channel access delay, transmission delay, and associated overhead. In other words, queuing delay is not included. The figures show that the delays with DAC are much better than delays without DAC in terms of protection and guarantee.

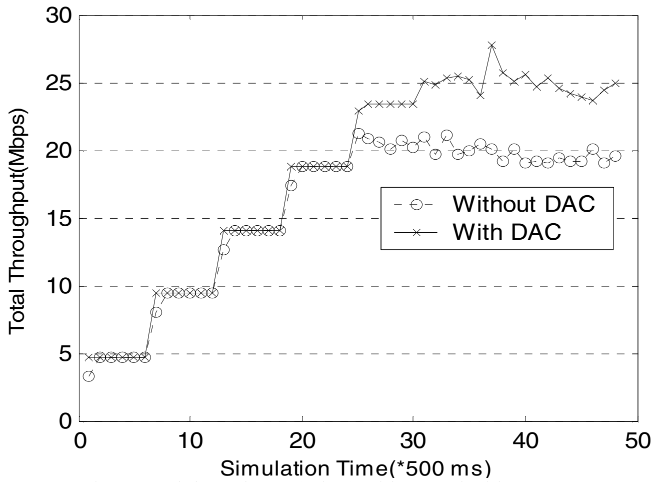


Fig. 5 Total throughputs (Mbps) without and with DAC

Fig. 5 shows the total throughput without and with DAC. The figures show the effects of total throughputs of adding one flow every 3s. Throughputs without DAC decrease a little at the end of the simulation, and achieve around 20Mbps. Throughputs with DAC becomes quite stable at the end of the simulation, and achieve 25Mbps, which is better than those without DAC.

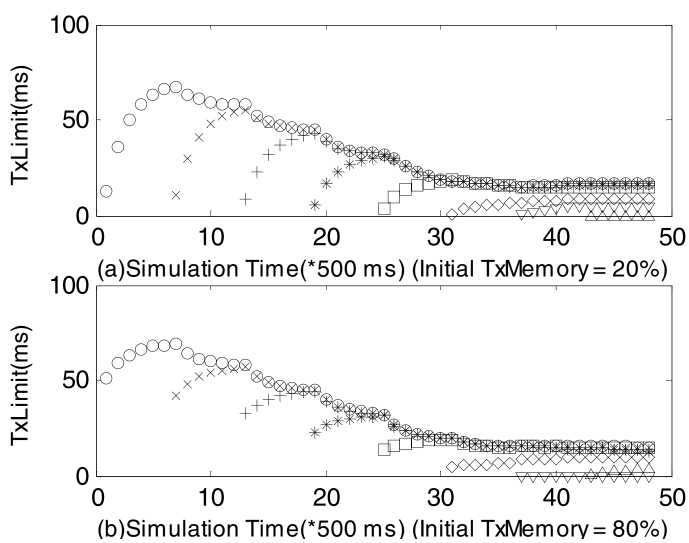


Fig. 6 Effects of *TxLimits* (ms) of video flows with DAC

Fig. 6 shows effects of *TxLimit* for different initial values with DAC. *TxMemory* in Fig. 6 (a) and (b) are initially 20% and 80% of *TXOPBudget/SurplusFactor*, respectively. Note that initial value of *TxMemory* is the same as the initial value of *TxLimit*. Fig. 6 shows the effects of convergence of *TxLimit* and *TxMemory*: after a couple of seconds, *TxLimits* of different video flows converge to almost the same value fairly. For an example, in Fig. 6 (b), at the time 7(*500ms), a new video flow is added, and at the time 13 (*500ms), all the flows almost have the same *TxLimit*. The reason behind this is that all video flows have the same rate, i.e., 4.86Mbps. If different video flows have different rates, *TxLimits* are expected to converge differently. Fig. 6 also shows that at the end of simulation when the system reaches saturation status, *TxLimits* of new flows cannot converge at reasonable values. We also observe that some new follows are rejected by DAC, i.e., *TxLimit* is zero. We will further improve these with enhancements of DAC in later figures. Fig. 6 (a) and (b) show *TxLimits* when initial values adopt 20% and 80 %, respectively. Accordingly, compared to Fig. 6 (b), Fig. 6 (a) shows that the initial values start at lower values. However, the effects of different initial values on these figures are similar.

B. Voice and Video Traffic Without Data Traffic

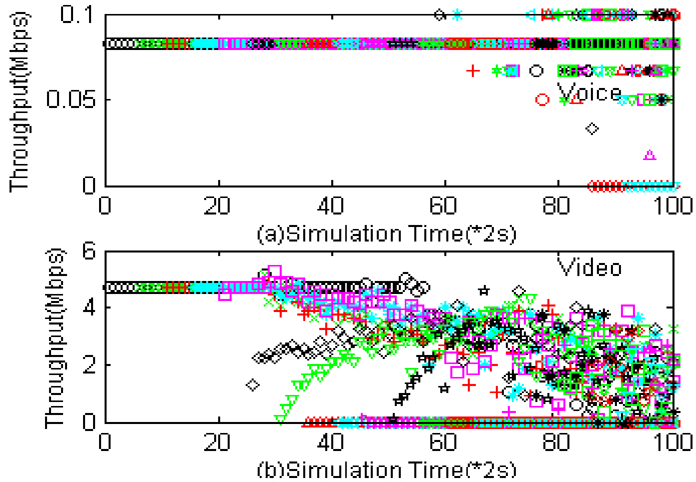
From Fig. 3 (throughputs with DAC) and Fig. 6 (*TxLimits* with DAC), we observe that DAC is not perfect when the system traffic reaches a saturated state. In other words, the existing video flows still cannot be protected well from the new video flows when the system is heavily loaded. In this subsection, we compare DAC and DAC+ETD+ENB. The simulations here include both voice and video traffic, but without any data traffic. In this simulation, for every 10 seconds, a new voice flow and a new video flow are added.

In ETD, the required delay is not considered, and α=80%. In ENB, ϕ x *Required_Budget*[3] = 1500µs (voice), and ϕ x *Required_Budget*[2] = 8500µs (video).

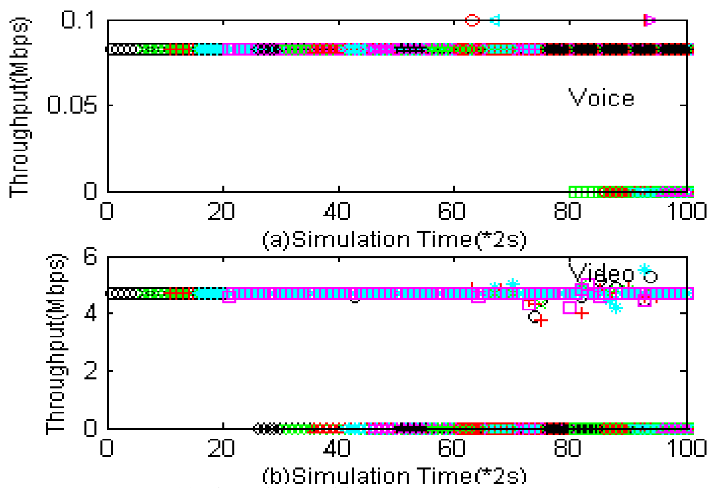
Fig. 7 (1) and (2) show throughputs for DAC and DAC+TED +ENB, respectively. Fig. 7 (1) shows that when the system traffic is heavy, the new video flows mess up the existing video flows. Therefore, the existing flows cannot be protected well from new flows. The effects have been also observed in Section VI.A. Compared to video flows, damage on voice flows is minors. This is probably because the voice flow rate is not high relatively. Fig. 7 (2) shows that with DAC+ETD +ENB, throughputs are improved significantly in terms of protection and guarantee. In Fig. 7 (2), there are more new flows being rejected to protect and guarantee the existing flows.

Fig. 8 shows *TxLimits* for DAC and DAC+ETD +ENB. Fig. 8 (1) shows that *TXLimits* for voice with DAC are pretty good, and 17 new voice flows are added. However, *TXLimits* for video with DAC are totally messed up in the middle of simulations when the system traffic is very heavy. Fig. 8 (2) shows that with DAC+ETD+ENB, new video flows are also rejected after 5 video flows are added when the system traffic is very heavy.

In summary, DAC+ETD+ENB is superior to DAC.

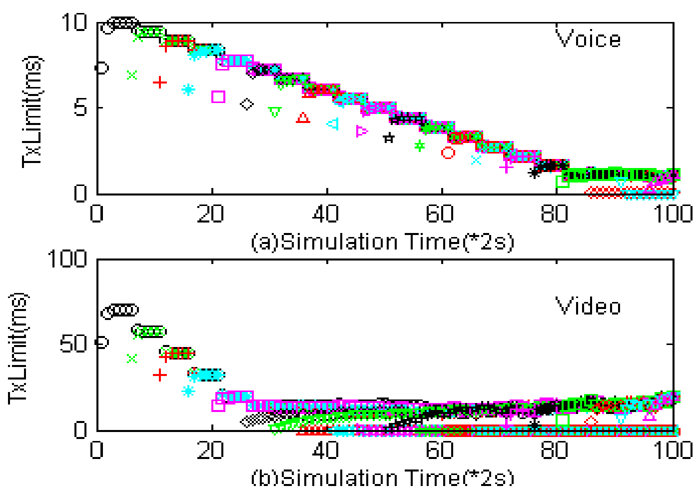


(1) Throughputs with DAC

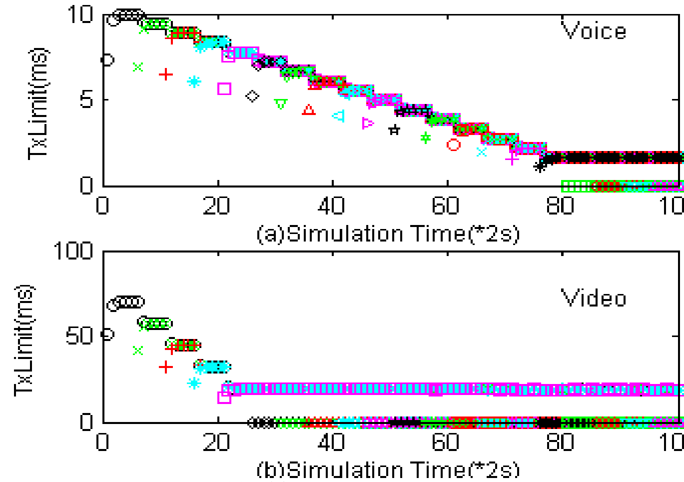


(2) Throughputs with DAC+ETD+ENB

Fig. 7 Throughputs (Mbps) for DAC and DAC+ETD +ENB



(1) with DAC



(2) with DAC+ETD +ENB

Fig. 8 *TXLimits* with DAC and DAC+ETD +ENB

C. Voice, Video and Data Traffic

In this subsection, we simulate with voice, video and data traffic for three schemes: DAC, DAC+ETD+ENB and DAC+ETD+ENB+BF+DAFS. We have θ= 1.5, ψ= 1, $m=1$, and $\sigma_{i+1}=2\sigma_i$, $(i=1,\ldots,L_{retry}-1)$, where m is the number of consecutive successful frames in the second-level protection. In this simulation, for every 10 seconds, a new voice flow, a new video flow, and a new data station are added.

Fig. 9 shows the number of collisions for DAC and DAC+ETD+ENB+BF+DAFS. As illustrated in the figures, DAC+ETD+ENB+BF+DAFS has a smaller number of collisions than DAC+ETD+ENB.

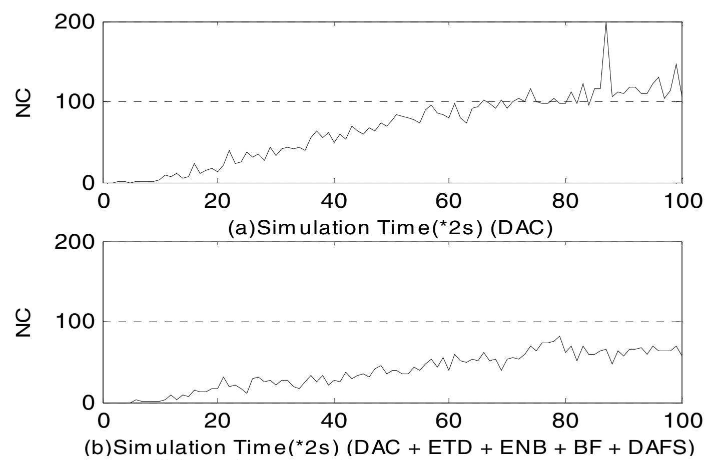
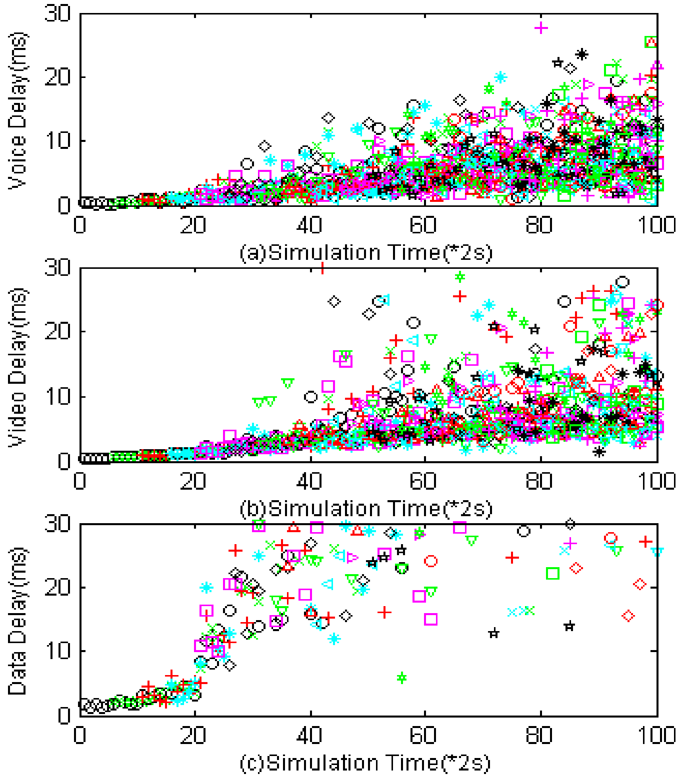
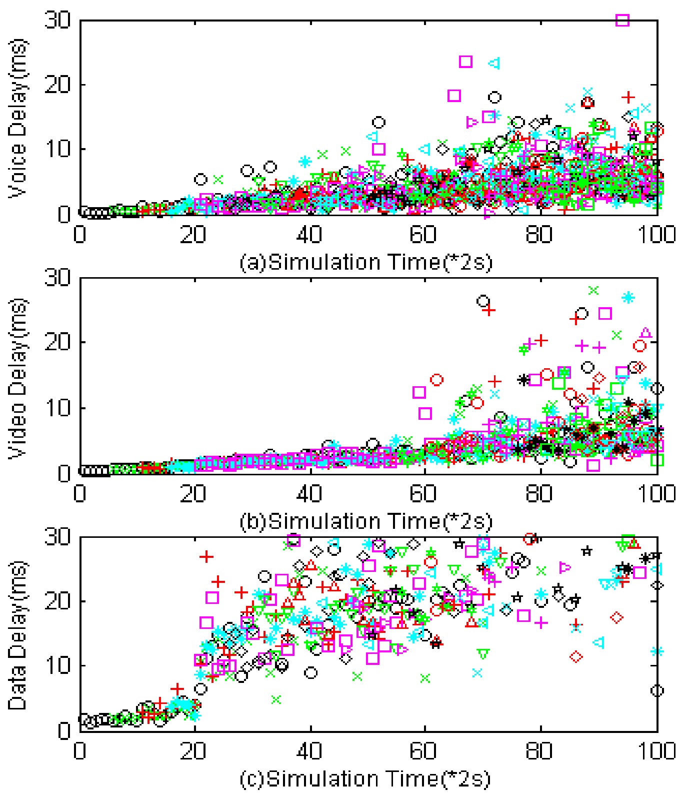


Fig. 9 Collisions with DAC and DAC+ETD+ENB+BF+DAFS
(Legend: NC represents the number of collisions)

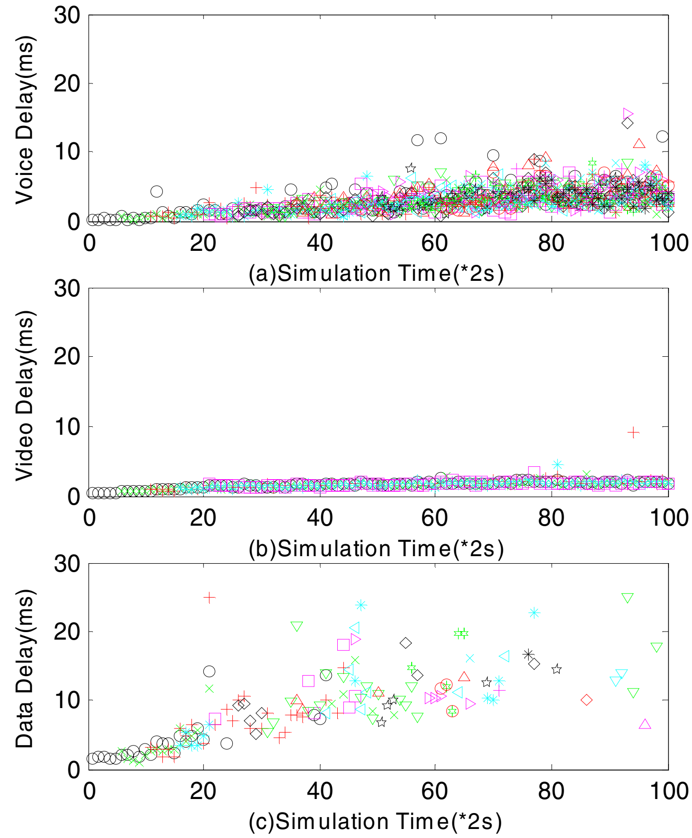
Fig. 10 shows delays for DAC, DAC+ETD+ENB, and DAC +ETD+ENB+BF+DAFS. As illustrated in the figures, with DAC, delays for both voice and video are not protected well; with DAC+ETD+ENB, voice and video are improved in terms of delay protection and guarantee; with DAC+ ETD+ENB+BF+DAFS, delays for both voice and video are the best in terms of delay protection and guarantee.



(1) Delays with DAC



(2) Delays with DAC+ETD+ENB

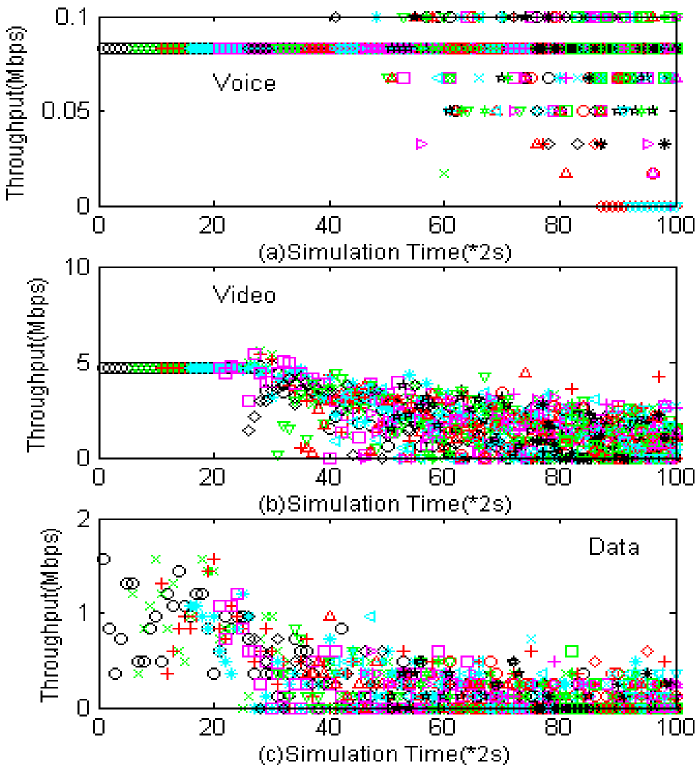


(3) Delays with DAC+ ETD+ENB+BF+DAFS

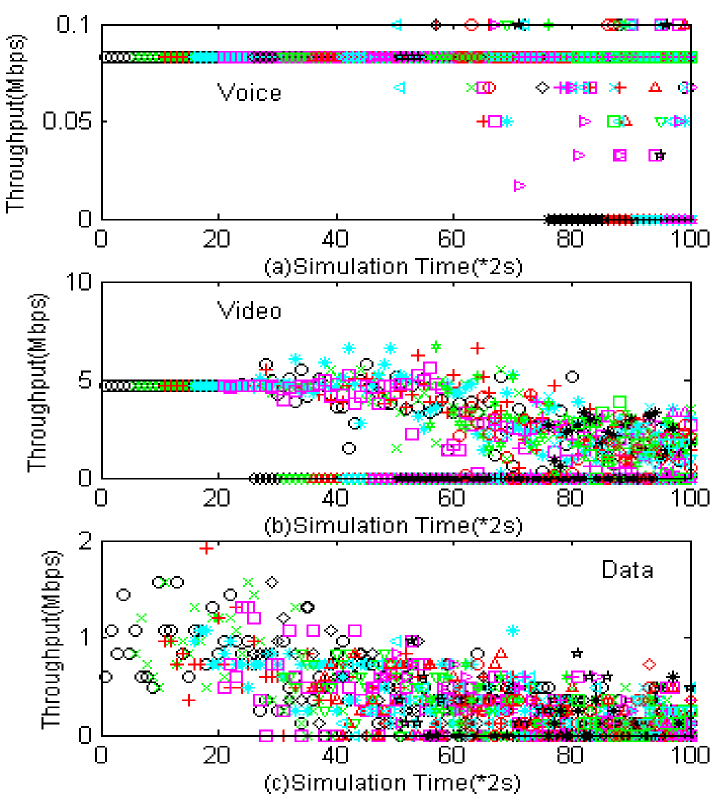
Fig. 10 Delays with DAC, DAC+ ETD+ENB and DAC+ETD+ENB+BF+DAFS

Fig. 11 shows throughputs for DAC, DAC+ETD+ENB, and DAC+ETD+ENB+BF+DAFS. Form Fig. 11 (1) with DAC, throughputs for voice are pretty good, but throughputs for video are very bad at the time 60s due to data transmissions. From Fig. 11 (2) with DAC+ETD+ENB, we observe that throughputs for video flows are improved since some new flows are rejected, but still are messed up with some degrees. From Fig. 11 (3) with DAC+ETD+ENB+BF+DAFS, we observe that throughputs for both voice and video are the best among three approaches in terms of protection and guarantee. We observe that the data traffic for DAC+ ETD+ENB+BF+DAFS becomes less than the other two approaches. These indicate that the data traffic gets controlled.

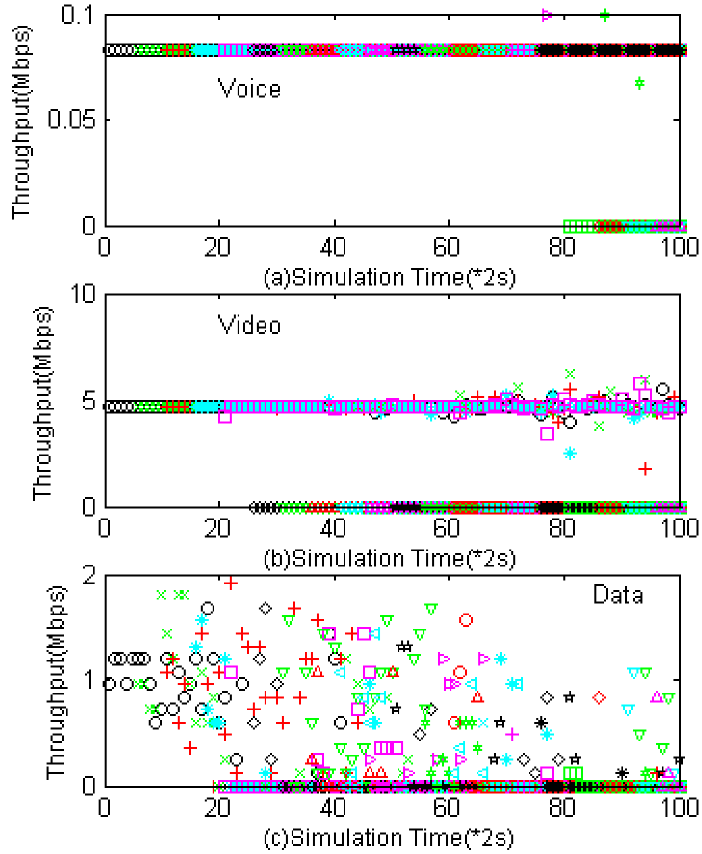
Fig. 12 shows total throughputs for DAC and DAC+ETD+ENB+BF+DAFS. Total throughputs for DAC decrease a lot when the traffic is heavy, whereas total throughput for DAC+ETD+ENB+BF+DAFS becomes relatively stable.



(1) Throughputs with DAC



(2) Throughputs with DAC+ETD+ENB



(3) Throughputs with DAC+ETD+ENB+BF+DAFS
Fig. 11 throughputs with DAC, DAC+ETD+ENB and DAC+ETD+ENB+BF+DAFS

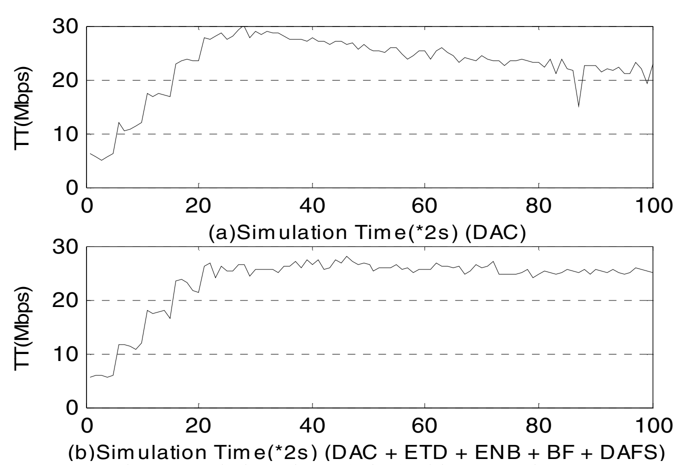


Fig. 12 Total Throughputs (Mbps) with DAC and DAC+ETD+ENB+BF+DAFS

VII. CONCLUSIONS

In this paper, we propose a two-level protection and guarantee mechanism for voice and video for EDCF of the IEEE 802.11e WLANs. Extensive simulations indicate that the proposed mechanisms are very effective in terms of protection and guarantee, and DAC+ETD+ENB+BF+DAFS is found to be the best approach. Due to limited space, many simulation results are skipped, and we will include them in the journal version of this paper.

REFERENCE

[1] IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification, Standard, IEEE, Aug. 1999.

[2] IEEE 802.11 WG, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE Std 802.11e/D4.3, May 2003.

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

[4] IEEE 802.11a WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: High-speed Physical Layer in the 5GHz Band, Sept. 1999.

[5] S. Mangold, S. Choi, P. May, O. Kein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," in *Proc. European Wireless'02*, Florence, Italy, Feb. 2002.

[6] S. Choi, J. Prado, S. Mangold, and S. Shankar, "IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation," in *Proc. IEEE ICC'03*, Anchorage, Alaska, USA, May 2003.

[7] Y. Xiao, "A Simple and Effective Priority Scheme for IEEE 802.11," *IEEE Communications Letters*, vol. 7, no. 2, Feb. 2003, pp. 70-72.

[8] Y. Xiao, "Performance Analysis of Priority Schemes for IEEE 802.11 Wireless LANs," submitted to *IEEE Trans. Wireless Communications*, Nov. 2002. The 2nd review in Apr. 2003.

[9] S. Mangold, S. Choi, P. May, and G. Hiertz, "IEEE 802.11e – Fair Resource Sharing between Overlapping Basic Service Sets," in *Proc. IEEE PIMRC'02*, Lisbon, Portugal, pp. 166-171, September 2002.

[10] S. Shankar and S. Choi, "QoS Signaling for Parameterized Traffic in IEEE 802.11e Wireless LANs," *Springer-Verlag Lecture Notes in Computer Science*, vol. 2402, 2002, pp. 67-84.

[11] S. Park, D. Kim, K. Kim, S. Choi, and S. Hong, "Collaborative QoS Architecture between DiffServ and 802.11e Wireless LAN," in *Proc. IEEE VTC'03-Spring*, Jeju, Korea, April 2003.

[12] Y. Xiao, "Enhanced DCF of IEEE 802.11e to Support QoS," in *Proc. IEEE WCNC'03*, New Orleans, March 2003.

[13] Y. Xiao, "Backoff-based Priority Schemes for IEEE 802.11," in *Proc. IEEE ICC'03*, Anchorage, AK, May 2003.

[14] D.-J. Deng and R.-S. Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method," *IEICE Trans. Communications*., vol. E82-B, no.1, Jan. 1999, pp.96-102.

[15] I. Aad and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," in *Proc. IEEE INFOCOM'01*, 2001.

[16] A. Veres, A. T. Campbell, M. Barry, and L.-H. Sun, "Supporting Differentiation in Wireless Packet Networks Using Distributed Control," *IEEE Journal of Selected Area in Communications (JSAC)*, vol. 19, no. 10, Oct. 2001, pp. 2081-2093.

[17] X. Pallot and L. E. Miller, "Implementing message priority policies over an 802.11 based mobile ad hoc network," in *Proc. IEEE MILCOM'01*, 2001.

[18] Daniel Collins, *Carrier Grade Voice over IP*, 2nd Ed., McGraw-Hill, September 2002.