Hybrid adaptation of the maximum contention window (CWmax) and minimum contention window (CWmin) for Enhanced Service Differentiation in IEEE 802.11 Wireless Ad-hoc Networks*

Hassan Takabi†, Ali H. Moghadam†, and Ahmad Khonsari††,†††,

†Computer Engineering Department, Sharif University of Technology, Tehran, Iran
†† Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
†††Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran

Summary
The IEEE 802.11e draft standard that proposed by IEEE Task Group E, defines new MAC protocols for QoS in wireless networks, mainly HCF that combines functions from DCF and PCF with enhanced QoS-specific mechanisms and frame types. HCF has two modes of operation EDCA and HCCA. EDCA is a contention-based mechanism and HCCA is a polling-based mechanism for channel access. EDCA, in fact, is an extension to DCF and is a contention-based channel access method that provides prioritized access to the medium. In this paper, we propose to extend EDCA using a hybrid adaptation algorithm of the maximum contention window(CWmax) and minimum contention window(CWmin) for Enhanced Service Differentiation in IEEE 802.11 Wireless Ad-hoc Networks, that enables each station to adjust size of CWmax and CWmin used in its back-off algorithm at run time. The purpose of suggested scheme is to reduce delay and jitter and increase the efficiency of the transmission channel. Priorities between access categories are prepared by updating the size of the CWmax and CWmin according to application requirements and network conditions. The performances of the IEEE 802.11e EDCA, enhanced with our hybrid adaptation algorithm, are inquired by simulations. Our results show that the hybrid adaptation algorithm outperforms the 802.11e EDCA standard in terms of channel throughput, packet delay and utilization, specially at high traffic load conditions. Indeed our proposed scheme increases the total throughput for high priority access category by up to 27% and reduces the delay for high priority traffic more than 49%. Furthermore, channel utilization ratio also increases at least 23%. Moreover, throughput for medium and low priority access categories remains stable.

Key words:
IEEE 802.11e, QoS, EDCA, back-off algorithm, wireless ad-hoc networks, hybrid adaptation algorithm, CWmax, CWmin

1. Introduction

One of the drawbacks of wireless networks in comparison to wired networks is that they are generally less efficient and unpredictable. Wireless has limited bandwidth, high packet overheads, and is more prone to environmental factors such as obstructions, interference, and weather and so on. The wireless medium (air) is much harder to control than a physical wire. The WLAN medium is also unlicensed and is therefore subject to interference from other devices. To further compound the problem, wireless devices are generally constrained by size, weight and battery size, limiting the processing power and the battery life. These factors further limit the capability of the network to provide an optimal solution. The main objective of WLAN QoS is to optimize use of limited bandwidth offered by a WLAN to address the issues noted above.

To optimize the best use of the resources and fulfill the resource requirements of different applications, QoS provides mechanisms to control access and usage of the medium based on the application. Each application has different needs in terms of bandwidth, delay, jitter, and packet-error rate and, therefore, QoS must cater to each of these needs. Applications requiring low delay (e.g., voice) may be given higher priority to use the medium, whereas applications requiring higher bandwidth may be assigned longer transmit times (e.g., video). Other traffic may require high reliability (e.g., email and data) and must be delivered with low packet-error rate.

Quality of Service (QoS) support is critical to multimedia applications [14]. These applications, including audio, video conferencing, voice, etc, require some specified bandwidth, delay, jitter and error rate guarantee to support a certain Quality of Service (QoS). Guaranteeing these QoS requirements is a challenging task with regard to 802.11 WLAN [3], [8], [9] protocols and Medium Access Control (MAC) functions [5].

In order to support QoS in 802.11 WLAN, several priority schemes has been developed [1], [4], [15], [16]. There are some priority schemes under discussion currently [7], [8]. The IEEE 802.11 Task Group E is working on the support of QoS in a new standard. It is defining enhancements to the 802.11 MAC access methods to support QoS, providing the classes of service, enhanced security and authentication mechanism [6]. These enhancements are defined in 802.11e draft [7] which introduces a main access method, HCF that combines functions from DCF and PCF with enhanced QoS-specific mechanisms. HCF

* This research was in part supported by a grant from I.P.M. (No. CS1385-2-02).
Manuscript received December 5, 2006.
Manuscript revised December 25, 2006.

has two modes of operation, Enhanced Distribution Coordinate Access (EDCA) and HCF Controlled Channel Access (HCCA). EDCA is a contention-based mechanism and HCCA is a polling-based mechanism for channel access. EDCA is an extension to DCF and is a contention-based channel access method that provides prioritized access to the medium [2], [7], [11]. EDCA is part of HCF for infrastructure networks. However, it may be used as a separate coordination function for wireless ad-hoc networks. The EDCA channel access method defines 4 access categories (ACs) based on the IEEE 802.11D standards and provides differentiated service, distributed access to the wireless medium for them [7]. A traffic category (TC) in 802.11 is defined as the application traffic related to a special user priority (UP) specified in IEEE 802.11D. The mapping between traffic categories (TCs) and access categories (ACs) is presented in IEEE 802.11e draft standard [2].

The proposed EDCA channel access mechanism uses at most 4 prioritized output queues on each QoS station (QSTA), one for each Access Category (AC). Figure 1 illustrates the different queues for different priorities. Different from a station, a QoS-supporting Access Point (QAP) should support at least 4 Access Categories (ACs). Service differentiation is provided by changing the size of the contention window (CW). EDCA uses the contention window to assign priority to each access category. It assigns a short contention window (CW) to an access category (AC) that should have higher priority in order to ensure that in most cases, high priority ACs will be able to transmit before the low priority ones. Indeed, the CWmin and CWmax parameters can be set differently for different access categories, yielding higher priority ACs with smaller values of CWmin and CWmax.

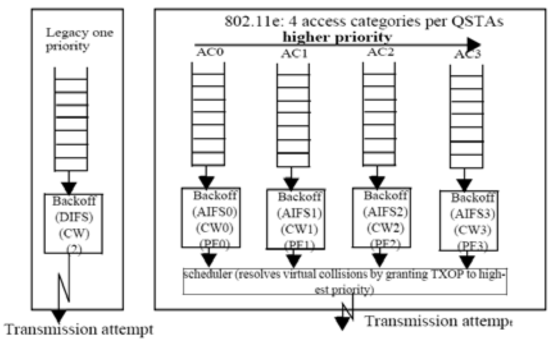

Figure 1. Queue-based EDCA fvs. basic DCF

For further differentiation, in 802.11e different IFS (Inter Frame Space) can be used according to traffic categories. Instead of DIFS, an Arbitration IFS (AIFS) is used. The AIFS for a given category should be a DIFS plus some (possibly zero) time slots. Categories with the smallest AIFS will have the highest priority as it is shown in figure 2.

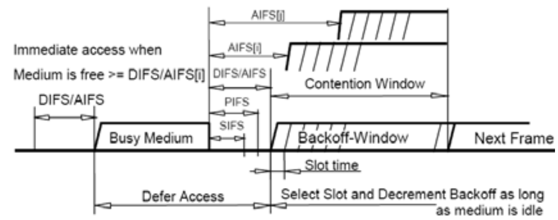

Figure 2. Some IFS Relationship

After any unsuccessful transmission (i.e. collision) to reduce the probability of a new collision, a new contention window (CW) is calculated with the help of the persistence factor PF[AC*i*] as shown in equation (1). Whereas in legacy 802.11 [9], CW is always doubled after any unsuccessful transmission (i.e., PF[AC*i*]=2). EDCA uses the PF to increase the CW different for each access category. Note that in the latest 802.11e draft [7] PFs differentiation per access category are no longer considered, i.e., for all access categories PFs equals to 2.

$$CW_{new}[AC_i] = ((CW_{old}[AC_i]) \times PF[AC_i]) - 1 \quad (1)$$

In addition, the CW never exceeds the maximum possible value for contention windows associated with each access categories, i.e., CWmax[AC*i*].

In this paper, we focus on the hybrid adaptation of the maximum contention window (CWmax) and minimum contention window (CWmin) for different channel conditions. We compare the performance of our proposed scheme with basic EDCA related to 802.11e proposed standard.

The rest of this paper is organized as fallows. In section 2, we present the proposed hybrid adaptation algorithm scheme. In section 3, the simulation topology and parameters are described. Simulations results and the performance of our proposed scheme are detailed in section 4. Finally in section 5, we conclude and describe our outlining future works.

2. HYBRID ADAPTATION ALGORITHM

In the rest of this paper, we assume that n stations are sending packets through the wireless media. The flows sent by each station may belong to different category of service with various priority levels. In each station and for each category i, the scheme maintains: the current contention window value (CW[*i*]), the minimum contention window value (CWmin[*i*]), and the maximum contention window value (CWmax[*i*]). Note that AC*i* is the *i*th access category, with *i* varies between 0 and 3 and that the high priority level is 0 and low priority is 3.

Both in the legacy DCF [9] and EDCA [7], after a successful transmission, the basic EDCA mechanism sets the contention window (CW) size of the corresponding category to its minimum contention window (CWmin) regardless the channel conditions. The problem is that when the transmission channel is high loaded or in a congested state, such aggressive reduction of the CW could cause more collisions. Also after an unsuccessful transmission (i.e. collision), the basic EDCA mechanism increases the contention window (CW) size of the corresponding category to its maximum contention window (CWmax) regardless the network condition. In this case the new contention window is calculated with the help of the persistence factor PF[ACi] whereas in legacy DCF [9], always we have PF[ACi]=2, but EDCA [7] uses the PF to increase the CW different for each access category. Note that in the latest 802.11e draft standard [7], like DCF [9], for all access categories PFs equals to 2. Furthermore, the CW never exceeds the maximum possible value for contention windows associated with each access categories, i.e., CWmax[AC*i*]. The problem is that when setting CWmax[i] too small the back-off growth stops too soon and delay an jitter increases (because of a higher number of collisions) and if the CWmax[i] is too large, the back-off growth stops too late and results in greater delay and jitter.

Previous researches focus on the adaptive enhanced distributed coordination function [14], the dynamic tuning of the minimum contention window (CWmin) value after successful transmissions [5] and the dynamic adaptation of the maximum contention window (CWmax) value after unsuccessful transmissions [6].

In this paper, we focus on hybrid adaptation of the CWmax and CWmin in both successful and unsuccessful transmissions. The main idea behind hybrid adaptation of the CWmax and CWmin is to adapt the CWmax[*i*] and CWmin[*i*] values for a certain access category *i* to traffic load and channel conditions. We believe, that by hybrid adaptation of CWmax and CWmin in both successful and unsuccessful transmissions for a certain access category *i* to the traffic load and channel conditions, we can improve parameters of QoS such as delay, jitter, throughput and channel utilization.2. Tables, Figures and Equations

2.1 Scheme Description

In the basic EDCA scheme for ad-hoc networks [7], [11], the CWmax[*i*] and CWmin[*i*] are statically set for each priority level. After each successful transmission the contention window is reduce to CWmin[*i*]. After each unsuccessful transmission (i.e. collision) the contention window is doubled, i.e. with an exponential back-off, and if it reaches or higher than the CWmax[*i*], so CW[*i*] remains at this value.

So, we propose that, after successful transmission update CWmax[*i*] and after collision update CWmin[*i*], both according to the traffic load and channel conditions.

We note that we use both a hybrid adaptation mechanism for CWmax[*i*] and CWmin[*i*] according to traffic load and also we differentiate between ACs while updating CWmax[*i*] and CWmin[*i*] for different priority levels.

In the next sub-sections detail how the contention window of each priority level *i* is set after each successful transmission and also after each collision.

2.2. Setting CW after Each Successful Transmission

After each successful transmission, the basic EDCA mechanism simply sets the contention window of the corresponding category to its minimum contention window regardless the network conditions, which probably lead to bursty collisions. In our hybrid adaptation scheme, we propose that each access category updates CWmin[*i*] and CWmax[*i*] parameters in an adaptive way using the estimated collision rate in each station at regular update period T_{update} expressed in time slots. We re-use the method defined in [14] to estimate the average collision rate as seen by a station *p*. Instantaneous collision rate f^{j}_{curr} at the j^{th} update period T_{update} is calculated using the number of collisions and the total number of packets sent during that period. The value of collision rate is given by equation (2).

$$f^{j}_{curr} = \frac{E(collisions_j[p])}{E(data\_sent_j[p])} \quad (2)$$

Where E(collisions*j*[*p*]) is the number of collisions of station *p* during period (update priod or step) *j*, and E(data_sent*j*[*p*]) is the total number of packets that have been sent during the update period T_{update}. Note that the above ratio f^{j}_{curr} is always in the range of [0, 1].

In order to minimize the bias against transient collisions, we use an estimator of Exponentially Weighted Moving Average (EWMA) to smoothen the estimated values. The value of collision rate at step *j* is given by equation (3).

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

Where j refers to the j^{th} update period T_{update} and f^{j}_{curr} stands for the instantaneous collision rate, α is

the *weight* (also called the *smoothing factor*) in the range [0, 1].
For adaptive calculation of the values of CWmax and CWmin, we re-use the methods that defined in [5]. Since, after each successful transmission, contention window sets to minimum contention window, in this case, we calculate the value of CWmin in an adaptive way by equations (4), (5).
We can calculate the value of CWmin in an adaptive way, using the equations (4).

$$DCW_{\min}[i] = (1 - f_{avg}^{j}) \times CW_{\min}[i] + f_{avg}^{j} \times (CW_{\max}[i] - CW_{\min}[i]) \times 2^{i-2} \quad (4)$$

Where DCWmin[i] shows the adaptive value of contention window minimum for an access category i, CWmin[i] is the minimum contention window (according to EDCA) assigned for the same access category i and f_{avg}^{j} represents the estimated collision rate at step j.
The dynamic contention window minimum for AC i obtained in equation (4) varies between a lower bound of CWmin[i], when the collision rate equals to 0, and an upper bound of $(CW_{\max}[i] - CW_{\min}[i]) \times 2^{i-2}$, when the collision rate equals to 1. Thus, this upper bound depends on the priority level of the access category. Indeed, this upper bound of DCWmin[i] is smaller for high priority traffic and greater otherwise. In order to ensure that the adaptive contention window minimum has an upper bound, the derived formula (in equation 4) uses the static value of CWmax according to EDCA along with the following formula:

$$DCW_{\min}[i] = \min(DCW_{\min}[i], CW_{\max}[i]) \quad (5)$$

2.3. Setting CW after Each Collision

The basic EDCA, after each unsuccessful transmission of packet of class i, the CW of this class is doubled, while remaining less than the maximum contention window CWmax[i]. In our hybrid adaptation scheme, we propose that each access category updates CWmin[i] and CWmax[i] parameters in an adaptive way using the estimated collision rate in each station at regular update period T_{update} expressed in time slots.

For adaptive calculation of the values of CWmax and CWmin, we re-use the methods that defined in [6]. Since, after each unsuccessful transmission, contention window sets to maximum contention window, in this case, we calculate the value of CWmax in an adaptive way, using the equations (6), (7).
The value of maximum contention window CWmax is calculated as bellows:

$$DCW_{\max}[i] = ()i+1) \times (f_{avg}^{j})^{5-2\times i} \times (CW_{\max}[i] - CW_{\min}[i]) \quad (6)$$

$$newCW_{\max}[i] = 2^{i+3} \times CW_{\min}[i] + DCW_{\max}[i] \quad (7)$$

Where newCWmax[i] shows the adaptive value of contention window maximum for an access category i (AC), CWmin[i] is the contention window minimum assigned for the same AC i and DCWmax[i] is a dynamic value of newCWmax.
In equation (6), CWmax[i] represents the static value of the contention window maximum according to EDCA and f_{avg}^{j} is the estimated collision rate. The new contention window value in equation (7) consists of two parts, a static part and a dynamic part given by DCWmax[i].
In order to ensure that the adaptive contention window maximum has an upper bound, the derived formulas (6) and (7) use the static value of CWmax according to EDCA along with the following formula:

$$newCW_{\max}[i] = \min(newCW_{\max}[i], \max PHYCW_{\lim}) \quad (8)$$

Where maxPHYCW*lim* is the maximum size of contention window limited by the physical layer, e.g., we use here a maximum value of the contention window of a 1023 slots.
In the adaptive contention window maximum defined in equation (6) and (7) there is a minimum backoff of at least $i+3$ times if there is no collision in the past (i.e., lower bound of the adaptive contention window maximum). With, the increase in the collision rate the value of newCWmax[i] increases and so with the value of i.
In order to ensure that the adaptive contention window maximum has an upper bound, the derived formulas (6) and (7) use the static value of CWmax according to EDCA along with the equation (8).
In the adaptive contention window maximum defined in equation (6) and (7) there is a minimum backoff of at least $i+3$ times if there is no collision in the past (i.e., lower bound of the adaptive contention window maximum). With, the increase in the collision rate the value of newCWmax[i] increases and so with the value of i.
After each unsuccessful transmission the contention window for an access category i is set as the following:

$$CW_{new}[i] = \max(newCW_{\max}[i], PF[i] \times CW_{old}[i]) \quad (9)$$

In our scheme we are using a PF[i]=2 for all access categories, so that, the contention window is doubled while remaining less than the maximum adaptive contention window, i.e., newCWmax[*i*].

3. SIMULATION METHODOLOGY

We have implemented our proposed scheme under the network simulator ns-2 [13] using the Atheros semi-package to support QoS enhancements features. We report in this section part of simulations we have done with different network topologies and source characteristics. Indeed this section presents the generic simulation topology used in order to evaluate the performance of the hybrid adaptation of the CWmin and CWmax scheme as well as a detailed analysis for a proper selection of α and T_{update} parameters used in the proposed scheme. An analysis of performance is presented in detail.

3.1. Simulation Topology

We use a generic topology (circular routing scenario) shown in figure 3, which consists of *n* stations indexed from *1* to *n*. Each station generates three types of data streams, labeled with high, medium and low according to their priorities. These data streams belong to the three traffic categories (TCs), respectively, audio (high), video (medium) and background traffic (low). Station *n* sends packets to station *1* and station *i* sends packets to station *i+1*. The highest priority access category in each station generates packets at sending rate of 64Kbps (PCM audio flow) which corresponds to a packet size of 160 bytes and an inter-packet arrival of 20ms. The medium priority access category generates packets at sending rate of 1024Kbps which corresponds to a packet size of 1280 bytes and inter-packet interval of 10ms. The low priority access category sending rate is 128Kbps which represents a packet size of 200 bytes and an inter-arrival packet of 12.5 ms. To increase the load of the system, we gradually increase the number of stations.

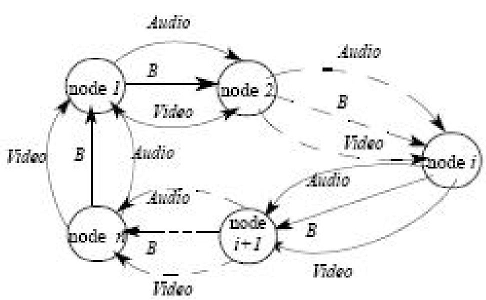

Figure 3. Simulation Topology

All the stations are located within an Independent Basic Service Set (IBSS), such that each station can detect the transmission from any other station. The different nodes are uniformly spread out of $500 \times 500m^2$ dimensions in 2D space. Table 1 shows the different MAC parameters for the three access categories (*0, 1, and 2*) used in the different simulation scenarios.

Table 1.MAC parameters for the three TCs

Parameters	High	Medium	Low
CW*min*	7	15	31
CW*max*	200	500	1023
AIFS(s)	34	43	52
PF	2	2	2
Packet size (bytes)	160	1280	200
Packet Interval (ms)	20	10	12.5
Sending rate(Kbps)	64	1024	128

This table presents as well as the parameters (e.g., packet size, sending rate and packet interval) of the three traffic categories associated with the defined three access categories. Table 2 presents the 802.11a PHY/MAC parameters.

In the following simulations, we assume that each QSTA operates at IEEE 802.11a PHY mode 6 [9] (i.e., modulation 16-QAM, coding rate of 3/4, data rate of 36 Mbps).

Table 2. 802.11a PHY/MAC parameters used in simulations

SIFS	16 s
DIFS	34 s
ACK size	14 bytes
Data rate	36 Mbps
Slot time	9 s
CCA time	3 s
MAC header	28 bytes
Modulation	16-QAM
Preamble Length	20 s
RxTxTurnaround time	1 s
PLCP header length	4 s

3.2. Impact of T_{update} (Update Period) and α (Smoothing Factor) Parameters

As described in Section 2.1, our proposed scheme uses an update period T_{update} (defined in number of time slots) and a smoothing factor α . In order to observe the effect of two mentioned parameters on the delay and on the goodput performances, we have done several set of simulations. First, to analyse the effect of the smoothing factor we vary the value of α in the range of [0, 1] and we set the update period to T_{update} =8000 time slots and we run simulations for a fixed number of stations, i.e. 25 stations. We run 20 simulations and results are averaged over these simulations. Goodput is defined as the total application layer received bytes divided by the total simulation time. Two performance criteria are used, the total goodput (or throughput) and the mean audio delay. The effect of smoothing factor on total goodput and mean audio delay shown in figure 4 and 5.

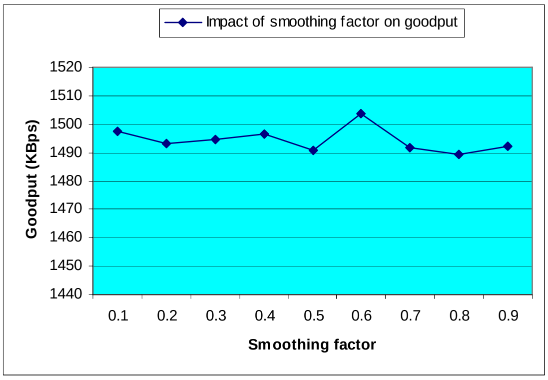

Figure 4. Impact of smoothing factor on goodput

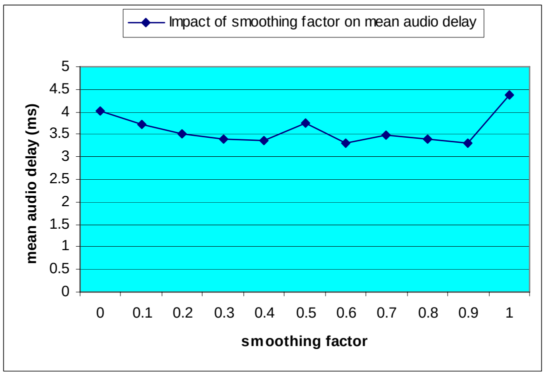

Figure 5. Impact of smoothing factor on mean audio delay

Figure 4 shows that a value of α in the range of [0.55, 0.65] achieves a higher goodput with a maximum of goodput for α =0.6. Also, we have higher goodput for values of α in the range [0, 0.2]. Since, small α values could contribute to random fluctuations we consider only values in the range of [0.55, 0.65].

From figure 5, we can see that, a value of α in the range [0.6, 0.9] achieves a lower delay and lowest delay is for α =0.9. Thus, we can see that values of α in the range of [0.6, 0.7] achieve a best trade-off between higher total goodput and lower mean audio delay. So, in the following simulations we set α to 0.6.

Figure 6 and 7 show the variations of total goodput and mean audio delay as a function of the update period values expressed in time-slots. The selection of the value of update period, T_{update} , should take into account that higher values make adaptations less useful and smaller ones could hurt the adaptation scheme since high frequent updates of CWmax and CWmin could be influenced by channel fluctuations.

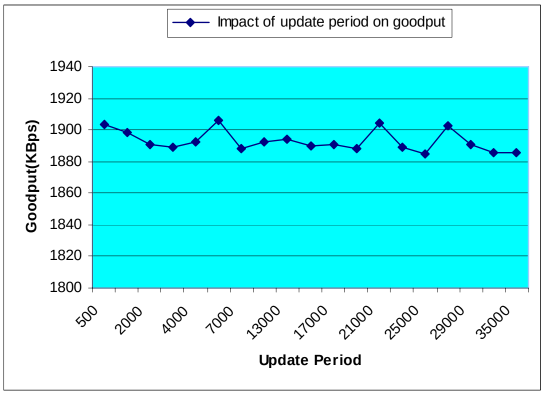

Figure 6. Impact of update period on goodput

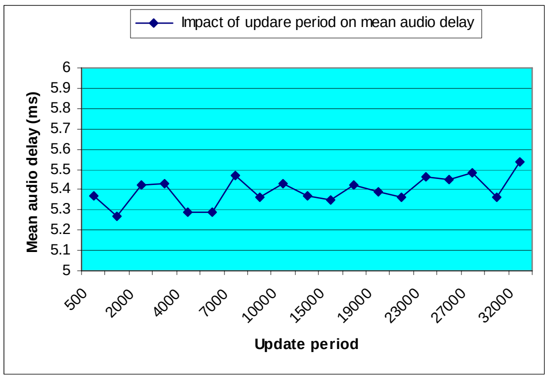

Figure 7. Impact of updare period on mean audio delay

We can see that trade-off between higher goodput and lower mean audio delay can achieve by selecting an update period T_{update} in the range of [5000, 8000].

In the following simulations, we set smoothing factor α and update period T_{update} values to 0.6 and 6000 time slots.

4. SIMULATION RESULTS

In order to evaluate the performance of the hybrid adaptation of the CWmax and CWmin scheme, in this section we inquiry the impact of traffic load and compare it to the basic EDCA scheme. The different type of traffic (associated with access categories) used for simulations are described in Table 1. We simulate various loads of the system by instantiating the simulation topology in figure 3 for different number of stations. All the stations are located within the same independent basic service (IBSS), so that, every station is able to detect the transmission from any other station.
The following QoS metrics are used to evaluate the performance of the proposed scheme:

- ***Gain of goodput:*** This metric stands for the gain (in %) on the average goodput (AG) of the proposed scheme compared with basic EDCA:

$$Gain\_of\_goodput = \frac{AG_{DCW\min\max} - AG_{EDCA}}{AG_{EDCA}}$$

- ***Mean delay:*** stands for the average delay of all the flows that have the same priority in the different stations. This metric is used to evaluate how well the scheme can accommodate real-time flows. However, real-time flows also require low average delay and boundd delay and jitter.

- ***Collision rate:*** represents the average number of collisions that occurs per second. In WLAN collisions increase the delay that station should wait, before initiating a new transmission attempt and it causes more delays.

- ***Medium utilization(Mu):*** the medium utilization represents the percentage of time used for the transmission of data frames and it is given by:

$$M_u = \frac{Totaltime - Collisiontime - Idletime}{Totaltime} \times 100\%$$

For the different simulation scenarios used in this section, all the traffic categories (associated with the access categories) are started at around 3.0 seconds with small individual offsets to have accurate CDFs (Cumulative Distribution function) of the delay. The simulation duration is 18 seconds. In order to have confidence in results obtained by simulations, we run 20 simulations and obtained results are averaged on theses simulations.

4.1. Throughput

Figure 8 shows the gain in goodput for hybrid adaptation scheme over EDCA.
The throughput improves in hybrid adaptation scheme and the gain of throughput for hybrid adaptation over EDCA is up to 27%. Furthermore, the gain increases when traffic load is greater than 10 stations as shown in figure 8. So, according to system throughput hybrid adaptation outperforms EDCA. The higher performance in throughput for hybrid adaptation is due to the increase in channel utilization because of the hybrid adaptation algorithm.

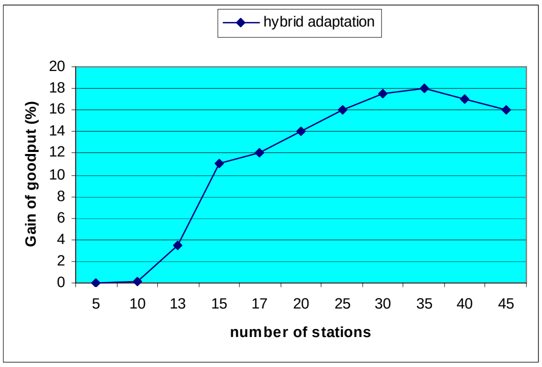

Figure 8. Throughput for hybrid adaptation scheme over EDCA

4.2. Packet delay

In this subsection, we compare the average packet delay under EDCA and our proposed scheme.
Figure 9 shows the mean audio delay as a function of traffic load for both hybrid adaptation scheme and EDCA. The mean audio delay improves significantly in hybrid adaptation compared to EDCA. Indeed, hybrid adaptation scheme maintains a lower audio delay than EDCA even in low traffic load environment, i.e., for a number of stations less than 10. As the load traffic increases, hybrid adaptation is able to maintain a lower delay than EDCA. The mean audio delay in hybrid adaptation scheme is up to 34% lower than in EDCA for a traffic load of 30 stations

and 49% lower for a traffic load of 45 stations and results in lower delay and jitter for high priority access categories.

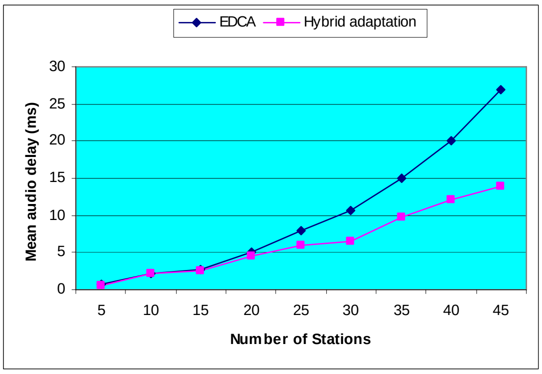

Figure 9. Mean audio delay for hybrid adaptation scheme and EDCA

This gain in delay for hybrid adaptation scheme can be explained by the hybrid adaptation algorithm that performs better than static CWmax and CWmin values and especially for medium and high loaded environment. As it can be seen in figure 10, there is an improvement of the mean video delay (medium priority traffic) in hybrid adaptation compared to EDCA. Both EDCA and hybrid adaptation have the same mean video delay when the traffic load is low, i.e., less than 13 stations. However, the delay improves in hybrid adaptation as the traffic load increases. The video delay is 75% lower in hybrid adaptation scheme than in EDCA for a system load of 35 stations. This can be explained by the hybrid adaptation algorithm used to adjust the size of CWmax[*i*] and CWmin[*i*] that performs better than a static ones in medium and high loaded channel system.

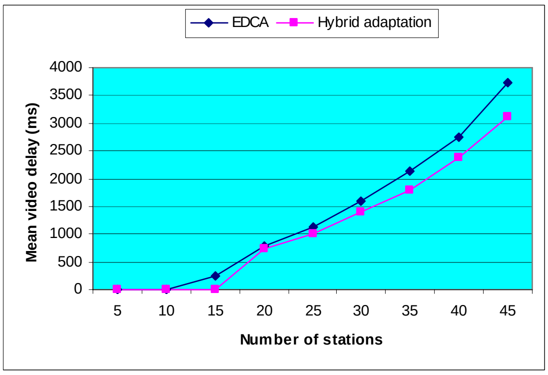

Figure 10. Mean video delay for hybrid adaptation scheme and EDCA

4.3. Medium utilization

Figure 11 shows the collision rate for hybrid adaptation scheme and EDCA. The collision rate is the same for hybrid adaptation and EDCA for a very low traffic load, i.e., 5 stations. As the traffic increases, the collision rate in hybrid adaptation, starting from a system load of 10 stations, maintains a lower increase than in EDCA. It can be seen that, for 25 stations, the collision rate in hybrid adaptation is 40% lower than in EDCA.

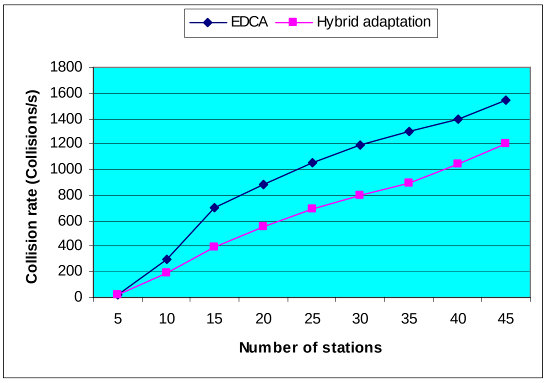

Figure 11. Collision rate for hybrid adaptation scheme and EDCA

We believe that the hybrid adaptation has contributed to reduce the number of collisions in the IBSS. Figure 12 shows the medium utilization for hybrid adaptation scheme and EDCA. It can be seen that under most system loads, hybrid adaptation scheme has much better channel utilization than EDCA. The capacity in hybrid adaptation is higher than EDCA (maximum channel utilization in EDCA is reached for 16 station while in hybrid adaptation corresponds to 27 stations). Therefore, the channel capacity is 23% higher than in EDCA.This is, because hybrid adaptation adjusts the size of CWmax [*i*] and CWmin[i] according to the network conditions.

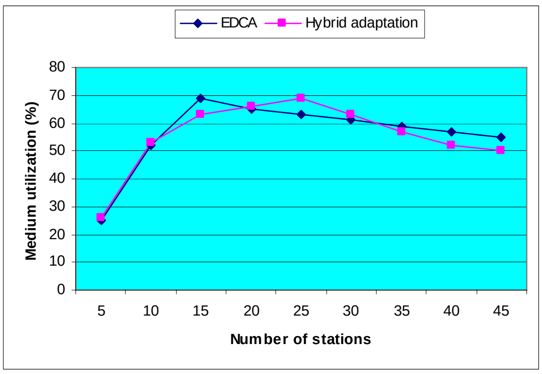

Figure 12. Medium utilization delay for hybrid adaptation scheme and EDCA

From the simulation results, we can conclude that hybrid adaptation scheme outperforms EDCA scheme in light, medium and high system load.
The total throughput increases by up to 27% for high priority traffic and remain stable for medium and low priority access categories. Furthermore, the delay for high priority access category reduces more than 49%. Moreover, the channel capacity improves and is 23% higher than in EDCA.

5. CONCLUSIONS

In this paper we have proposed a new dynamic scheme for the hybrid adaptation of the contention window maximum (CWmax) and the contention window minimum (CWmin) in order to enhance the service differentiation for 802.11 WLANs. We have extended the basic EDCA scheme by an algorithm that enables each station to adjust the size of the CWmax and CWmin used in its back-off algorithm at run time.
The adjusting is differentiated for each access category *i* and performed according to the channel traffic conditions.
Simulation results demonstrated that our scheme achieves better performance of throughput, delay and jitter than basic EDCA, specially for high priority traffic.
Results are validated by analyzing the impact of sources and network dynamics on the performance metrics and compared with the basic 802.11e EDCA. On one hand, results have shown that audio delay associated with high priority access category, improves greatly and decreases by up to 49%. Furthermore, the channel capacity improves and is 23% higher than in EDCA. Moreover, we have observed the total throughput increases by up to 27% for high priority traffic and remain stable for medium and low priority access categories.
Further work could include implementation of the proposed hybrid adaptation algorithm for infrastructure networks.

References

[1] Aad, Imad and Castellucia,C. "Differentiation mechanisms for IEEE 802.11", In Proceedings of IEEE Infocom 2001, A, Alaska, April 2001.
[2] ANSI/IEEE Standard 802.1D, 1998 ed., "IEEE standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Common specifications. Part 3: media access control (MAC) bridges", Institute of Electrical and Electronics Engineers, 1998.
[3] Cali, F., Conti, M. and Gregori, E. "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism", In IEEE JSAC, vol. 18, no. 9, September 2000.
[4] Deng, D-J. and Chang, R-S. "A priority scheme for IEEE 802.11 DCF access method", IEICE Transactions on Communications, E82-B(1), January 1999.
[5] Gannoune, L. and Robert, S. "Dynamic Tuning of the Contention Window Minimum (CWmin) for Enhanced Service Differentiation in IEEE 802.11 Wireless Ad-Hoc Networks", To appear in IEEE PIMRC'04
[6] Gannoune, L. and Robert, S. "Results on Dynamic Adaptation of the Contention Window Maximum (CWmax) for Enhanced Service Differentiation in IEEE 802.11 Ad-Hoc Networks", Accepted for publication in IEEE VTC Fall 2004
[7] IEEE WG, "Draft Supplement to Standard for Telecommunications and Information Exchange Between Systems-LAN/MAN specific requirements -Part 11: Wireless Medium Access Control (MAC) and Physical Layer specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)", IEEE 802.11e/Draft D5.0, July 2003.
[8] IEEE WG home page: http://grouper.ieee.org/groups/802/11
[9] IEEE WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11 Standard", 1999.
[10] Lindgren, A. "Evaluation of Quality of Service of Service Schemes for IEEE 802.11", In Proc. Of IEEELCN, Nonember 2001.
[11] Mangold, S., Choi, S., May, P., Klein, O., Hiertz, G. and Stibor, L. "IEEE 802.11e Wireless LAN for Quality of Service", in Proc. European Wireless '02, Florence, Italy, February 2002.
[12] Ni, Q. "QoS issues and enhancements for IEEE 802.11 Wireless LAN", INRIA Research Report No.4612, November 2002.
[13] Ns-2 network simulator homepage: http://www.isi.edu.edu/nsnam/ns
[14] Romdhani, L., Ni, Q., and Turletti, T., "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks", in the Proceedings of the IEEE WCNC'03.
[15] Vaidya, N.H., Bahl, P. and Gupta, S. "Distributed fair scheduling in wireless LAN. In Sixth Annual International Conference on Mobile Computing and Networking", Boston, August 2000.
[16] Veres, A., Campbell, A.T., Barry, M. and Sun, L-H. "Supporting service differentiation in wireless packet networks using distributed control", IEEE Journal of Selected Areas in Communications (JSAC), Special Issue on Mobility and Resource Management in Next-Generation Wireless Systems, vol. 19, No. 10, pp. 2094-2104, October 2001.

Hassan Takabi received his B.Sc. degree in Software Engineering from AmirKabir University of Technology (Tehran Polytechnic), Tehran, Iran in 2004. He is currently an M.Sc. student in the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. He also works as a research assistant in Sharif Network Security Center from 2005. His main research interests include Information and Network Security; Computer, and Communication Networks; and Wireless, Adhoc and Sensor Networks.

Ali H. Moghadam is currently an M.S. student in the Software Quality Research Lab. in Sharif University of Technology, Theran, Iran. He recieved his B.S. degree in Software Engineering from AmirKabir University of Technology (Tehran Polytechnic),, Tehran, Iran. His research interests are Software Reliability Modeling, Software Engineering, and Wireless Networks.

Ahmad Khonsari received the BSc degree in electrical and computer engineering from Shahid-Beheshti University, Iran, in 1991, and MSc degree in computer engineering from the Iran University of Science and Technology, Iran, in 1996 and PhD degree in computer science from the University of Glasgow, UK, in 2003. He is currently an assistant professor in the Department of Electrical and Computer Engineering, University of Tehran, Iran and a researcher in School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (I.P.M.), Iran. His research interests are performance modelling/evaluation, mobile and ubiquitous computing, communication networks and distributed systems, and high performance computer architecture.