International Journal of Soft Computing 1(3): 225-231, 2006

Comparative Study of MAC Protocols for Mobile Ad Hoc Networks

[1]Sanjeev Sharma, [2]R.C. Jain, [3]Sarita Bhadauria and [4]Ramesh Cherukuru
[1,3,4]School of Information Technology, Rajiv Gandhi Technological University, Bhopal M.P. India
[2]Department of Computer Application, Samrat Ashok Technological Institute, Vidisha M.P. India

Abstract: Studies of ad hoc wireless networks are a relatively new field gaining more popularity for various new applications. In these networks, the Medium Access Control (MAC) protocols are responsible for coordinating the access from active nodes. These protocols are of significant importance since the wireless communication channel is inherently prone to errors and unique problems such as the hidden-terminal problem, the exposed-terminal problem, and signal fading effects. Although a lot of research has been conducted on MAC protocols, the various issues involved have mostly been presented in isolation of each other. We therefore make an attempt to present a comparative study of major schemes, integrating various related issues and challenges. We present a classification of MAC protocols and their brief description based on their operating principles and underlying features.

Key words: Ad hoc networks, wireless networks, MAC, medium access control MANET

IEEE 802.11 DCF

IEEE 802.11 defines two mechanisms for medium access control: DCF (Distributed Coordinator Function) is used during contention period in both Ad-hoc and architecture network configurations, whereas PCF (Point Coordinator Function) is the access method during contention free period with the presence of a point coordinator (access point). In other words, PCF can be used only in architecture network.

The IEEE 802.11 legacy MAC[1] is based on the logical functions, called the coordination functions, which determine when a station operating within a Basic Service Set (BSS) is permitted to transmit and may be able to receive frames via the wireless medium. Two coordination functions are defined, namely, the mandatory DCF based on CSMA/CA and the optional Point Coordination Function (PCF) based on poll and- response mechanism. Most of today's 802.11 devices operate in the DCF mode only. We explain how the DCF works in this section as it is the basis for the Enhanced DCF (EDCF), which we discuss in this study.

The 802.11 MAC works with a single First-In-First-Out (FIFO) transmission queue. The CSMA/CA constitutes a distributed MAC based on a local assessment of the channel status, i.e., whether the channel is busy (i.e., a station is transmitting a frame) or idle (i.e., no transmission). Basically, the CSMA/CA of DCF works as follows:

When a frame (or an MSDU2) arrives at the head of the transmission queue, if the channel is busy, the MAC waits until the medium becomes idle, then defers for an extra time interval, called the DCF Interframe Space (DIFS). If the channel stays idle during the DIFS deference, the MAC then starts the backoff process by selecting a random Backoff Counter (or BC). For each slot time interval, during which the medium stays idle, the random BC is decremented. When the When a frame (or an MSDU2) arrives at the head of the transmission queue, if the channel is busy, the MAC waits until the medium becomes idle, then defers for an extra time interval, called the DCF Interframe Space (DIFS). If the channel stays idle during the DIFS deference, the MAC then starts the backoff process by selecting a random Backoff Counter (or BC). For each slot time interval, during which the medium stays idle, the random BC is decremented. When the BC reaches zero, the frame is transmitted. On the other hand, when a frame arrives at the head of the queue, if the MAC is in either the DIFS deference or the random backoff process3, the processes described above are applied again. That is, the frame is transmitted only when the random backoff has finished successfully. When a frame arrives at an empty queue with no on-going backoff process and the medium has been idle longer than the DIFS time interval, the frame is transmitted immediately.

Each station maintains a Contention Window (CW), which is used to select the random backoff counter. The BC is determined as a random integer drawn from a uniform distribution over the interval [0,CW]. How to determine the CW value is further detailed below. If the channel becomes busy during a backoff process, the backoff is suspended. When the channel becomes idle

Corresponding Author: Sanjeev Sharma, School of Information Technology, Rajiv Gandhi Technological University, Bhopal M.P. India

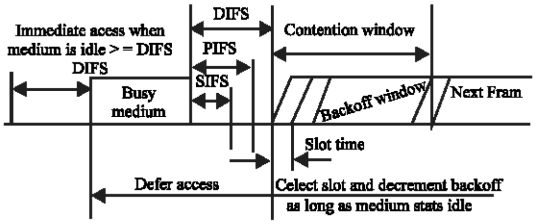

Fig. 1: IEEE 802.11 DCF channel access

Table 1: MAC parameters for 802.11B PHY

Parameters	SHIFT (usec)	DIFS (usec)	Slot time (usec)	Cwmin	Cwmax
802.11b PHY	10	50	20	31	1023

again and stays idle for an extra DIFS time interval, the backoff process resumes with the suspended BC value.

The timing of DCF channel access is illustrated in Fig. 1. For each successful reception of a frame, the receiving station immediately acknowledges by sending an acknowledgement (ACK) frame. The ACK frame is transmitted after a short IFS (SIFS), which is shorter than the DIFS. Other stations resume the backoff process after the DIFS idle time. Thanks to the SIFS interval between the data and ACK frames, the ACK frame transmission is protected from other stations' contention. If an ACK frame is not received after the data transmission, the frame is retransmitted after another random backoff.

The CW size is initially assigned CWmin, and increases when a transmission fails, i.e., the transmitted data frame has not been acknowledged. After any unsuccessful transmission attempt, another backoff is performed using a new CW value updated by 2 • (CW + 1)-1, with an upper bound of CWmax. This reduces the collision probability in case there are multiple stations attempting to access the channel. After each successful transmission, the CW value is reset to CWmin, and the station that completed the transmission performs another DIFS deference and a random backoff even if there is no other pending frame in the queue. This is often referred to as "post" backoff, as this backoff is done after, not before, a transmission. This post backoff ensures there is at least one backoff interval between two consecutive MSDU transmissions.

All of the MAC parameters including SIFS, DIFS, Slot Time, CWmin, and CWmax are dependent on the underlying Physical layer (PHY). Table 1 shows these values for the 802.11b PHY[2]. Irrespective of the PHY, DIFS is determined by SIFS+2•SlotTime and another important IFS, called PCF IFS (PIFS), is determined by SIFS+SlotTime. With 802.11b, the transmission rate is up to 11 Mbps. There are other PHYs with rates of up to 54 Mbps. As we are discussing MAC enhancements, our evaluation results in the following are valid, irrespective of the underlying PHY.

DCF with RTS/CTS: MACA: A station which needs to send data sends an RTS (Request to Send) frame in the normal CSMA/CA style. The receiver when it receives the RTS frame, sends a CTS frame after waiting for an SIFS. The sender sends its data frame after an SIFS after it gets the CTS. Likewise, on receiving a data frame, a station waits for an SIFS and sends an ACK. The RTS/CTS frames have two octets, which specify the time for which the medium is reserved. A machine does not transmit any DATA or CTS when its NAV is busy. This is the virtual carrier sensing. RTS/CTS reduces the number of collisions and solves the hidden and exposed terminal problems

As a summary, DCF is good for asynchronous data transmission, but it suffers significant performance degradation at high load conditions, because of the higher collision rate and wasted time on negotiations. Plus, DCF does not differentiate services, thus it s not suitable for time-bounded traffics.

802.11E mac enhanced dcf (EDCF): To handle service differentiation, IEEE 802.11 was extended to 802.11b, in which Enhanced DCF (EDCF) is deployed as the contention based media access mechanism. Its main modification on DCF is that eight levels of user priorities can be applied to stations. A station with higher priority is assigned shorter CWmin and CWmax, so that in most cases higher priority flows will have more chances to transmit before lower priority ones. Moreover, different IFS are introduced to different priority levels, which mean higher priority flows have longer IFS, while the lower priority ones have shorter IFS. The IFS here is called Arbitrary IFS (AIFS). EDCF achieves a good prioritization. However it still has the same problem as DCF that it performs poorly when the traffic load is high due to frequent collisions and wasted idle time.

The 802.11 legacy MAC does not support the concept of differentiating frames with different priorities. Basically, the DCF is supposed to provide a channel access with equal probabilities to all stations contending for the channel access in a distributed manner. However, equal access probabilities are not desirable among stations with different priority frames. The emerging EDCF is designed to provide differentiated, distributed channel accesses for frames with 8 different priorities (from 0 to 7) by enhancing the DCF. As distinct from the legacy DCF, the EDCF is not a separate coordination function. Rather, it is a part of a single coordination function, called the Hybrid Coordination Function (HCF), of the 802.11e MAC. The HCF combines the aspects of both DCF and PCF.

Table 2: Priority to access category mappings

Priority	Access category (AC)	Designation (Informative)
1	0	Best effort
2	0	Best effort
0	0	Best effort
3	1	Video probe
4	2	Video
5	2	Video
6	3	Video
7	3	Video

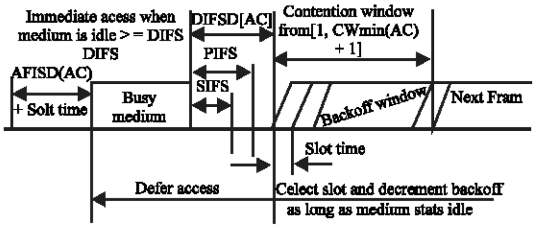

Fig. 2: IEEE 802. 11e EDCF channel access

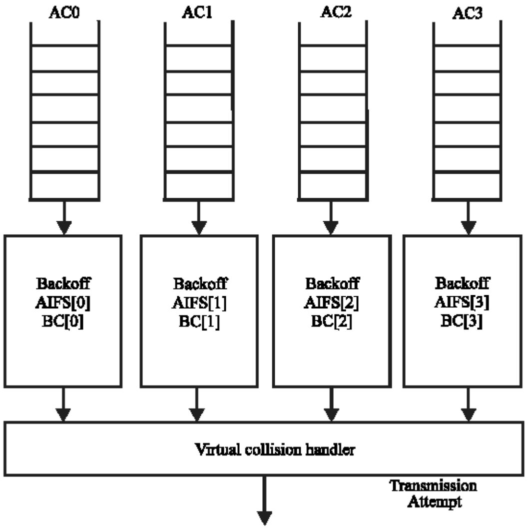

Fig. 3: Four access categories (Acs) for EDCF

Each frame from the higher layer arrives at the MAC along with a specific priority value. Then, each QoS data frame carries its priority value in the MAC frame header. An 802.11e STA shall implement four Access Categories (ACs), where an AC is an enhanced variant of the DCF 0. Each frame arriving at the MAC with a priority is mapped into an AC as shown in Table 2. Note the relative priority of 0 is placed between 2 and 3. This relative prioritization is rooted from IEEE 802.1d bridge specification[3].

Basically, an AC uses AIFSD[AC], Wmin[AC] and CWmax[AC] instead of DIFS, CWmin and CWmax, of the DCF, respectively, for the contention process to transmit a frame belonging to access category AC. AIFSD[AC] is determined by

$$\mathrm{AIFSD}[AC] = \mathrm{SIFS} + \mathrm{AIFS}[AC].\ \mathrm{Solt\ Time}$$

where AIFS[AC] is an integer greater than zero. Moreover, the backoff counter is selected from [1,1+CW[AC]], instead of [0,CW] as in the DCF. Figure 2 shows the timing diagram of the EDCF channel access.

The values of AIFS[AC], CWmin[AC], and CWmax[AC], which are referred to as the EDCF parameters, are announced by the AP via beacon frames. The AP can adapt these parameters dynamically depending on network conditions. Basically, the smaller AIFS[AC] and CWmin[AC], the shorter the channel access delay for the corresponding priority, and hence the more capacity share for a given traffic condition. However, the probability of collisions increases when operating with smaller CWmin[AC]. These parameters can be used in order to differentiate the channel access among different priority traffic.

Figure 3 shows the 802.11e MAC with four transmission queues, where each queue behaves as a single enhanced DCF contending entity, i.e., an AC, where each queue has its own AIFS and maintains its own Backoff Counter BC. When there is more than one AC finishing the backoff at the same time, the collision is handled in a virtual manner. That is, the highest priority frame among the colliding frames is chosen and transmitted, and the others perform a backoff with increased CW values.

The IEEE 802.11e defines a transmission opportunity (TXOP) as the interval of time when a particular STA has the right to initiate transmissions. Along with the EDCF parameters of AIFS[AC], CWmin[AC] and CWmax[AP], the AP also determines and announces the limit of an EDCF TXOP interval for each AC, i.e., TXOPLimit[AC], in beacon frames. During an EDCF TXOP, a STA is allowed to transmit multiple MPDUs from the same AC with a SIFS time gap between an ACK and the subsequent frame transmission[4,5]. We refer this multiple MPDU transmission to as "Contention-Free Burst (CFB)."

Figure 4 shows the transmission of two QoS data frames during an EDCF TXOP, where the whole transmission time for two data and ACK frames less than the EDCF TXOP limit is announced by the AP. As multiple MSDU transmission honors the TXOP limit, the worst-case delay performance is not be affected by allowing the CFB. We show below that CFB increases the system throughput without degrading other system performance measures unacceptably as long as the EDCF TXOP limit value is properly determined.

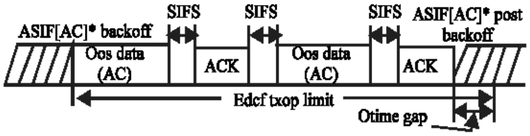

Fig. 4: CFB timing structure

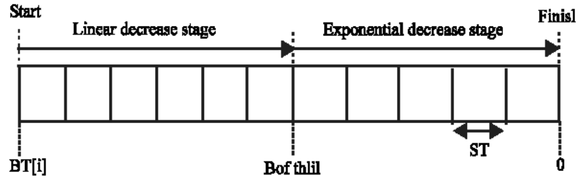

Fig. 5: Backoff time decrease phases

DCF and EDCF comparison: In this scenario, we simulate with four voice stations, two video stations, and four data stations for both the DCF and the EDCF. Figure 5 shows throughput, delay, and data dropping rate for the DCF and the EDCF. By comparing Fig. 5 (a) and (d), which plot the aggregated throughput of each traffic type, we observe that the throughputs of video and data are significantly different for the DCF and the EDCF. Knowing that the aggregate video rate from two stations is 2.8 Mbps, we can easily imagine that the video traffic is well served with the EDCF while many video frames are dropped with the DCF. This fact is confirmed in Fig. 5 (b) and (e), which show significant reduction in video frame losses with the EDCF. Note that a frame drop occurs when there is a buffer overflow. There is small voice frame loss with the DCF while there is none with the EDCF. On the other hand, we observe that with both the DCF and the EDCF, there is no data frame drop as an infinite size buffer is used for data stations. Instead, data frame delay goes to infinity with both the DCF and the EDCF. Note that the delay for data is not plotted in Fig. 5 (f) so as to clearly show the delay performances for voice and video with the EDCF. We observe in Fig. 5 (f) that voice performance is significantly improved via the EDCF. Note that with the DCF, the voice frame delay sometimes goes over 250 msec, which is not acceptable in most cases. The video delay performance is also improved remarkably with the EDCF. It should be noted that each delay curve is from a single station, e.g., one of four voice stations while the previous throughput and data dropping rate were aggregated from all the same types of stations. That is the main reason why the peaks in data dropping rate and delay curves look totally uncorrelated. One interesting observation is that even with the DCF, the voice frame delay is much smaller than those of video and data frames. That is because virtually every voice frame arrives at an empty queue thanks to its traffic pattern. That is each voice frame is transmitted after contention before the next frame arrives at the queue. Note that a voice frame arrives at a transmitting MAC every 20 msec while the voice delay with the DCF is less than 20 msec in most cases. From the results thus far, we conclude that the EDCF can provide differentiated channel accesses for different traffic types. With the observed delay and error performance, we expect that the EDCF can support real-time applications with voice and video traffic with a reasonable quality of service in certain environments.

Furthermore, EDCF suffers from low priority traffic starvation especially at high load conditions, which impairs its fairness.

The adaptive edcf (AEDCF): Service differentiation used in EDCF provides better services to high priority class while offering a minimum service for low priority traffic. Although this mechanism improves the Quality-of-Service of real-time traffic, EDCF parameters cannot be adapted to the network conditions, such as the collision rate and the network load. Adaptive EDCF (AEDCF) was proposed to adapt the CW parameter according to the network conditions, so that better support for QOS is provided. The idea is as follows. After a successful transmission, the CW is updated slowly instead of being reset to CWmin, which is for the purpose of avoiding busty collisions. Similarly, after a collision, the new CW is not doubled but increased with a persistence factor, causing that the CW of high priority traffic increases slower than that of low priority traffic. The gradual update of the CW takes into account the average collision rate at each station, which is computed periodically.

The factor of CW update is calculated in such a way that flows with high collision rate will have a better chance to transmit next time. AEDCF successfully decreases the collision rate between stations with the same priority, and decreases the access latency as well.

This scheme extends the basic EDCF by making it more adaptive taking into account network conditions. We assume that n stations are sending packets through the wireless media. The flows sent by each station may belong to different classes of service with various priority levels. In each station and for each class 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 i varies from 0 (the highest priority class) to 7 (the lowest priority class).

Scheme description: In order to efficiently support time-bounded multimedia applications, we use a dynamic procedure to change the contention window value after each successful transmission or collision. We believe that this adaptation will increase the total goodput of the traffic which becomes limited when using the basic EDCF, mainly for high traffic load.

In the basic EDCF scheme for ad-hoc networks[6], the CWmin[i] and CWmax[i] values are statically set for each priority level. After each successful transmission, the CW[i] values are reset to CWmin[i]. We propose to reset the CW[i] values more slowly to adaptive values (different to CWmin[i]) taking into account their current sizes and the average collision rate while maintaining the priority-based discrimination. In other words, we ensure that at each instant, the highest priority class has the lowest contention window value so that it has the highest priority to access the media. The adaptive slow CW decrease is a tradeoff between wasting some backoff time and risking a collision followed by the whole packet transmission. After each collision, the source has to wait for a timeout to realize that the packet has collided, and then doubles its CW to reduce the number of collisions. We propose to change the mechanism and differentiate between classes using different factors to increase their CWs.

In the next sub-sections, we explain in detail how the contention window of each priority level is set after each successful transmission and after each collision.

Setting CW after each successful transmission: After each successful transmission, the basic EDCF mechanism simply sets the contention window of the corresponding class to its minimum contention window regardless the network conditions. Motivated by the fact that when a collision occurs, a new one is likely to occur in the near future, we propose to update the contention window slowly (not reset to CWmin) after successful transmission to avoid bursty collisions. The simplest scheme we can use to update the CW of each class i is to reduce it by a static factor such as 0.5 * CWold . In the remainder of this study, we denote this approach the Slow Decrease (SD) scheme. However, a static factor cannot be optimal in all network conditions. In our scheme, we propose that every class updates its CW in an adaptive way taking into account the estimated collision rate fjcurr in each station. Indeed, the collision rate can give an indication about contentions in a distributed network. The value of fj curr is calculated using the number of collisions and the total number of packets sent during a constant period (i.e., a fixed number of slot times) as follows:

$$f^{j}_{curr} = \frac{E(\text{collisions}_j)[p])}{E(\text{data\_sent}_j[p])} \tag{1}$$

where E(collisionsj [p]) is the number of collisions of station p which occurred at step j, and E(data-sentj [p]) is the total number of packets that have been sent in the same period j by flows belonging to the station p. Note that the above ratio fj curr is always in the range of [0, 1].

To minimize the bias against transient collisions, we use an estimator of Exponentially Weighted Moving Average (EWMA) to smoothen the estimated values. Let fjavg be the average collision rate at step j (for each update period) computed according to the following iterative relationship:

$$f^{j}_{avg} = (1-\alpha) * f^{j}_{curr} + \alpha * f^{j-1}_{avg} \tag{2}$$

where j refers to the jth update period and fj curr stands for the instantaneous collision rate, _ is the weight (also called the smoothing factor) and effectively determines the memory size used in the averaging process.

The average collision rate is computed dynamically in each period Tupdate expressed in time-slots. This period should not be too long in order to get good estimation and should not be too short in order to limit the complexity.

To ensure that the priority relationship between different classes is still fulfilled when a class updates its CW, each class should use different factor according to its priority level (we denote this factor by Multiplicator Factor or MF). Keeping in mind that the factor used to reset the CW should not exceed the previous CW, we limit the maximum value of MF to 0.8. We have fixed this limit according to an extensive set of simulations done with several scenarios. In AEDCF, the MF of class i is defined as follows:

$$MF[i]=\min((1+(i*2))*f^{j}_{avg},0.8) \tag{3}$$

This formula allows the highest priority class to reset the CW parameter with the smallest MF value (i.e., priority level 0, see P0 in Fig. 2). After each successful transmission of packet of class i, CW[i] is then updated as follows:

$$CW_{new}[i]=\min(CW_{max}[i],CW_{old}[i]*MF[i]) \tag{4}$$

The equation above guarantees that CW[i] is always greater than or equal to CWmin[i] and that the priority access to the wireless medium is always maintained.

Setting CW after each collision: In the current version of EDCF[15], 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]:

$$CW_{new}[i]=\min(CW_{max}[i], 2*CW_{old}[i]) \tag{5}$$

In AEDCF, after each unsuccessful transmission of packet of class i, the new CW of this class is increased with a Persistence Factor PF[i], which ensures that high priority traffic has a smaller value of PF[i] than low priority traffic:

$$CW_{new}[i]=\min(CW_{max}[i], CW_{old}[i]*PF[i]) \qquad (6)$$

In fact, this PF parameter has been proposed in a previous version of the draft, but it has been removed from draft. In this study we introduce PF in our AEDCF scheme because by this way we can reduce the probability of a new collision and consequently decrease delay.

The problem of AEDCF is that the performance of background low priority flows degrades at high load, because the background traffic will have much larger average CW size than high priority traffics, thus increasing waiting time and impairing channel utilization.

ADAPTIVE FAIR EDCF

From analysis of the above schemes, it can be seen that the main performance impairment of distributed contention-based approaches comes from packet collisions and wasted idle slots due to backoff in each contention cycle. The ideal case is reached when a successful packet transmission is followed by another successful packet transmission without any collisions or idle time loss. Aiming at decreasing collision and idle time, AFEDCF deploys the mechanism in which the CW is increased not only when there is a collision but also when the medium is sensed busy during deferring periods. The backoff timer is decreased when the medium is sensed idle in two different stages: linear decrease and fast decrease. In the linear decrease, the backoff timer goes down one by one, while in the fast decrease, it decreases exponentially. The boundary between those two stages is the varying backoff threshold. Considering the traffic load, it should increase during low contention periods but decrease during high contention periods. From another perspective, when a collision occurs or the station is in the deferring period waiting for the channel to be idle, it doubles the CW, randomly chooses a new backoff time and reduces the backoff threshold to make the fast decrease phase shorter. After a successful transmission, the station resets the CW to CWmin, randomly chooses the backoff time and increases the backoff threshold to make the fast decrease stage longer. AFEDCF doesn't use the adaptive CW update as AEDCF does.

Scheme algorithm: The algorithm of our adaptive fair EDCF scheme is described as follows.

Backoff timer decrease state: All priority queues in the different active stations monitor the medium. If a queue 6 senses the medium idle for a slot, then it will start decrementing its backoff timer by a slot time as in the IEEE 802.11e specification, i.e.., $BT_{new}[i]=BT_{old}[i]-ST$ If number of consecutive idle slots are detected and the remaining backoff timer value is less or equal than the Backoff Threshold Bof_Th[i] value, our algorithm will decrease faster exponentially) the backoff timer as proposed in[6] for DCF:

$$BT_{new}[i]=BT_{old}[i]/2$$

$$\text{if } BT_{new}[i]< ST, \text{then } BT_{new}[i]=0$$

___ 6 %$ _ _

When the backoff timer reaches zero, the station transmits a packet.

Packet collision state: If a queue notices that its packet transmission has failed possibly due to a packet collision, the queue must react with these modifications: (i) it must double its current CW[i] value by using Equation (6) as in the IEEE 802.11e specification in order to avoid a new collision, (ii) it must update its BT[i] value by using Eq.7, and (iii)must reduce its Bof_Th[i] value by using Equation (8) in order to decrease the fast decrease phase (see Fig. 4). We explain later the logic behind (8).

$$CW[i]=\min(CW_{max}[i], 2.CW[i]) \qquad (6)$$

$$BT[i]=\text{uniform}(1,CW[i]+1).ST \qquad (7)$$

_ 4 6 _$____ 6 _ 1__ 7 _ _ 9 ___ 6 #(_ __, _'4 _

Successful packet transmission state: When a queue successfully transmits a packet, it must react with these modifications: (i) it reduce its current CW[i] size to $CW_{min}[i]$ as specified in the IEEE 802.11e specification in order to reduce the idle time, (ii) it must update its BT[i] value by using Eq.7 and (iii) it must increase its Bof_Th[i] value by using Eq. 8 in order to increase the exponential decrease stage (Fig. 4).

Deferring state: If a queue is in a deferring state (i.e., waiting the end of a busy period to continue decreasing its backoff timer), whenever it detects the start of a new busy period (which indicates either a collision or a packet transmission in the medium by another station), it will react as if it is in the packet collision state described in

Section IIB. 2 in contrast with the IEEE 802.11e specification. We propose such a behavior in order to protect multimedia flows transmission and to improve the fairness between the same priority applications especially when the medium is highly congested. Basically, the queues which are in the deferring state double CW[i] when they sense the medium is busy in order to: (i) penalize the low priority queue because it has the largest $CW_{max}[i]$ value while the highest priority queue will gain more transmission opportunities due to its small $CW_{max}[i]$ value, (ii) improve the fairness between the same priority queues by having almost, after the finish of a busy period, the same value of CW[i] equal to $CW_{max}[i]$ and consequently the same transmission opportunity. Our mechanism for adapting the function Bof_Th[i] ensures that the protection of multimedia flows performance is accompanied by a total throughput increasing since our scheme implements the fast backoff decrease mechanism. AFEDCF shows good performance in almost all respects. It achieves high throughput and fairness even at high load conditions. The fairness is due to the fact that if the traffic load is high, the CW of all stations reach rapidly their maximum values, thus they will be transmitting almost all the time at the same contention window. The deployment of varying backoff threshold contributes to less wasted idle time and adaptation to collision rate.

CONCLUSION

This study has presented a broad overview of the research work in the field of ad hoc wireless networks with respect to MAC protocols. We have discussed the characteristics and operating schemes of severel MAC schemes. We compared the MAC protocols DCF,EDCF,AEDCF and AFEDCF with their merits and demerits.

We extend the basic EDCF scheme by using an adaptive fast backoff mechanism to improve the total throughput along with a window doubling mechanism at busy periods to protect further high priority flows and improve the fairness between those of the same priority especially in the scenarios where the channel is highly congested. Our scheme adapts the stations aggressiveness during accessing the medium according to its load by using an adaptive fast backoff decrease mechanism. It protects further the transmission of multimedia flows and the fairness between those of the same priority by increasing the current contention window size whenever the queue detects the channel busy in both the transmission failure state and the deferring state. Adaptive fair EDCF provides good multimedia flow performance in all channel loads as well as a higher total throughput than EDCF. Besides, it provides a higher degree of fairness than EDCF and AEDCF between the different flows of the same priority.

REFERENCES

1. IEEE Std. 802.11-1999. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Reference number ISO/IEC 8802-11:1999(E), IEEE Std. 802.11, 1999 Edn.
2. IEEE Std. 802.11b, 1999. Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed Physical Layer Extension in the 2.4 GHz Band, IEEE Std. 802.11b-1999.
3. IEEE 802.1d-1998, 1998. Part 3: Media Access Control(MAC) bridges, ANSI/IEEE Std. 802.1D, 1998 Edn.
4. Sunghyun Choi, Javier del Prado, Atul Garg, Maarten Hoeben, Stefan Mangold, Sai Shankar and Menzo Wentink, 2002. Multiple Frame Exchanges during EDCF TXOP," IEEE 802.11-01/566r3.
5. IEEE 802.11e/D4.0, 2002. 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).
6. IEEE WG, 2002. 802.11e Draft 3.1.