Results on Dynamic Adaptation of the Contention Window Max (CW_{max}) for Enhanced Service Differentiation in IEEE 802.11 Wireless Ad-Hoc Networks

Lassaâd Gannoune[1], Stephan Robert[1], Neha Tomar[2], and Tarun Agarwal[2]
EIVD[1] IIT[2]

Abstarct- The proposed IEEE 802.11e draft standard defines new MAC protocols for QoS in wireless networks, mainly EDCF and HCF. EDCF is a contention-based channel access scheme and is part of HCF for infrastructure networks and may be used as a separate coordination function for wireless ad-hoc networks. In this paper, we propose to extend EDCF with a dynamic adaptation algorithm of the maximum contention window (CWmax) for enhanced service differentiation in wireless ad-hoc networks. The purpose of our scheme is to reduce delay and jitter and increase the efficiency of the transmission channel. Priorities between access categories are provisionned by updating the size of the CWmax according to application requirements and channel conditions. The performances of IEEE 802.11e EDCF, enhanced with our adaptation algorithm, are extensively investigated by simulation. Results obtained indicate that the delay and jitter for high priority traffic, i.e., audio traffic, improve and delay decreases by up to 43%. Furthermore, delay for lower priority traffic, e.g., video traffic, remains stable. Throughput in CWmax adaptation, compared to EDCF, is stable in light and medium system load and slightly lower (less than 3%) especially in high loaded system.

Key-words: IEEE 802.11e, EDCF, ad-hoc WLANs, back-off algorithm, dynamic adaptation.

1. Introduction

Multimedia applications, including voice, require a certain quality of service (QoS) support such as guaranteed bandwidth, delay, jitter and error rate. Guaranteeing those QoS requirements is a challenging task with regard to 802.11 WLAN [9],[10],[13] protocols and Medium Access Control (MAC) functions.

In order to support QoS in 802.11 WLAN, several priority schemes has been developed [1], [3], [6], [8]. Currently, there are some priority schemes under discussion [10], [2]. The IEEE 802.11 Task Group E is currently defining enhancements to the 802.11 MAC access methods to support QoS, providing the classes of service, enhanced security and authentication mechanism. These enhancements are defined in 802.11e draft [2] which introduces two main access methods, the Hybrid Coordination Function (HCF) and the Enhanced Distributed Coordination Function (EDCF), renamed in latest 802.11e draft [2] to EDCA (enhanced distributed channel access). EDCF is a contention-based channel access scheme [4], [2], [5] and is part of HCF for infrastructure networks. However, it may be used as a separate coordination function for wireless ad-hoc networks. EDCF provides differentiated service, distributed access to the wireless medium for 4 delivery priorities or access categories [2]. A Traffic Categories TC in 802.11e is defined as the application traffic related to a special user priority UP specified in IEEE 802.1D [12]. The mapping between traffic categories TCs and access categories ACs is presented in 802.11e draft [2].

EDCF access channel on each QoS Station (QSTA) uses at most 4 prioritized output queues, one for each delivery priority, called Access Categories (ACs). Figure 2 illustrates the different queues for different priorities. As for a station, a QoS-supporting Access Point (QAP) should support at least 4 Access Categories (ACs). In EDCF, relative priorities are provisionned by configuring the time to access the channel [6], [4] once it is sensed idle defined as arbitrary interframe space (AIFS) as shown in figure 1. Differentiation is also provided by changing the size of the contention window (CW). EDCF uses the contention window to assign priority to each access category. Indeed, assigning a short contention window to a high priority AC ensures that in most cases, high priority AC is able to transmit ahead of low priority one. Thus, the CW_{min} and CW_{max} parameters can be set differently for different access categories, such as, a high priority AC with small values of CW_{min} and CW_{max}.

---

1. Ecole d'Ingénieurs du Canton de Vaud (EIVD), Rue de Galilée 15, CH-1400 Yverdon-les-Bains, Switzerland, {lassaad.gannoune, stephan.robert}@eivd.ch
2. Indian Institute of Technology (IIT) Delhi, Hauz Khas, New Delhi - 110016 India {mau99502,mau99517}@ccsun50.iitd.ernet.in

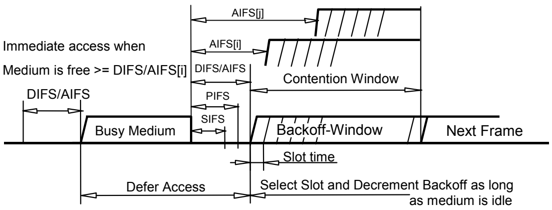

Fig. 1. Some IFS relationship

After any unsuccessful transmission a new contention window is calculated with the help of the persistence factor PF[AC_i] and another uniformly distributed backoff counter out of this new, enlarged CW, is drawn to reduce the probability of a new collision.

$$newCW[AC_i]=((oldCW[AC_i]+1)\times PF[AC_i])-1$$

Whereas in legacy 802.11 [9], CW is always doubled after any unsuccessful transmission (i.e., PF[AC_i]=2). EDCF, uses the PF to increase the CW different for each access category. In the latest 802.11e draft [2] PFs differentiation per access category are no longer considered, i.e., PFs equals to 2 for all access categories.

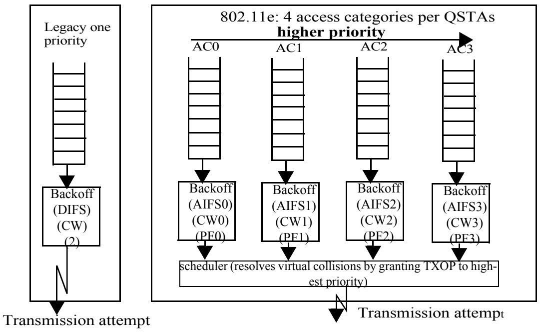

Fig. 2. Queue based EDCF vs legacy DCF

In addition, the CW never exceeds the parameter $CW_{max}[AC_i]$, which is the maximum possible value for contention windows associated with each access categories.

In this paper we focus on the dynamic adaptation of the maximum contention window (CW_{max}) with a special designed scheme. We compare the performance of our proposed scheme with basic EDCF related to 802.11e proposed standard which does only consider a fixed CW_{max}. In the following sections we present our proposed scheme and its performance results.

2. CW_{max} adaptation algorithm

We assume, in the rest of the paper, that AC_i is the *ith* access category, with *i* varies between 0 and 3 and that the high priority level is 0 and low priority is 3. The main idea behind CW_{max} adaptation is to adapt the $CW_{max}[i]$ value for a certain access category *i* to traffic load and conditions. The problem is that when setting $CW_{max}[i]$ too small the backoff growth stops too soon and delay an jitter increases (because of a higher number of collisions). Moreover, if the $CW_{max}[i]$ is too large, the backoff growth stops too late and results in greater delay and jitter. We believe, that by a continuous adaptation or tuning of $CW_{max}[i]$ to traffic load, delay and jitter can be reduced especially in high loaded environment.

2.1 Scheme description

In the basic EDCF scheme for ad-hoc networks [4] and [2], the CW_{min}[i] and CW_{max}[i] are statically set for each priority level. After each failed transmission (i.e., collision) the CW is doubled, i.e., with an exponential backoff, and if it reaches or higher than the CW_{max}[i], so CW[i] remains at this value. We propose that before each exponential backoff, we update the $CW_{max}[i]$ according to the traffic load. So that the doubled contention window will be compared to an updated value of the $CW_{max}[i]$.

We note that we use both an adaptive mechanism for CW_{max}[i] according to traffic load and also we differentiate between access categories while updating CW_{max}[i] for different priority levels. The update of CW_{max}[i] is the following principals:

- Keep CW_{max}[i] low for higher priority traffic. Increase it at a lower pace, compared to lower priority traffic.
- The update formula of CW_{max}[i] is a function of the current traffic load which is measured in a distributed manner and represents the collision average rate seen locally by a station.

The next sub-sections detail how the contention window of each priority i is set after each collision and after each successful transmission and the method used to estimate the collision rate at each wireless station p.

2.1.1 Setting CW after each collision

In our adaptation scheme we propose that each access category updates each CW_{max}[i] parameter in an adaptive way using the estimated collision rate at regular update period T_{update} in time slots. In our scheme we re-use the method defined in [5] to estimate the average collision rate as seen by a station p. Indeed, instantaneous collision rate f^j_{avg} at the j^{th} update period T_{update} is calculated using the number of collisions and the number of packets sent during that period. Collision rate is given by (1) [5]:

$$f^j_{curr} = \frac{E(collisions_j[p])}{E(data\_sent_j[p])} \tag{1}$$

Where $E(collisions_j[p])$ is the number of collisions at station p during the period (update period or step) j and $E(data\_sent_j[p])$ is the number of packets sent during the update period T_{update}. In order to get an estimation of the collision rate that minimizes random fluctuations, an Exponentially Weighted Moving Average (EWMA) is used to smooth the series of collision rates (i.e., f^j_{curr}). Equation (2) gives the corresponding value of collision rate at step j.

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

Where α is a smoothing factor in the range [0, 1], j refers to the j^{th} update period T_{update}, f^j_{curr} stands for the instantaneous collision rate. Using the estimated average of collision rate the numerical expression of the proposed adaptive CWmax, $newCW_{max}$, is presented in Equations (3) and (4).

$$newCW_{max}[i] = 2^{i+3} \times CW_{min}[i] + DCW_{max}[i] \tag{3}$$

$$DCW_{max}[i] = (i+1) \times \left(f^j_{avg}\right)^{(5-2\times i)} \times (CW_{max}[i] - CW_{min}[i]) \tag{4}$$

Where $newCW_{max}$[i] stands for the adaptive value of contention window maximum for an access category i (AC), $CW_{min}[i]$ is the contention window minimum assigned for the same AC i and DCW_{max}[i] is a dynamic value of $newCW_{max}$ given by the equation (4).

In equation (4), f^{j}_{avg} represents the estimated collision rate, $\mathrm{CW}_{\mathrm{max}}$[i] is the static value of the contention window maximum according to EDCF. The new contention window value in equation (EQ 3) consists of two parts, a static part and a dynamic part given by $DCW_{max}[i]$. The basic criteria/ideas that motivate the proposed formula in (2) and (4) are:

- The adaptive $\mathrm{CW}_{\mathrm{max}}$ (i.e., $\mathrm{newCW}_{\mathrm{max}}$[i]) depends on the collision rate. Whenever there is a large traffic, the $\mathrm{newCW}_{\mathrm{max}}$ should be large and small otherwise.
- In order to ensure differentiation between access categories, the adaptive $\mathrm{CW}_{\mathrm{max}}$ (i.e., $\mathrm{newCW}_{\mathrm{max}}$) depends on the priority of the AC *i*. For higher priority we should have a small value. Therefore the $\mathrm{newCW}_{\mathrm{max}}$ vary directly with *i*.
- The adaptative $\mathrm{CW}_{\mathrm{max}}$ should also depend on the $\mathrm{CW}_{\mathrm{min}}$[i] value. Indeed, this is useful to set a minimum number of backoff's after which the contention window attains a lower bound of the adaptive CWmax.

In order to ensure that the adaptive contention window maximum has an upper bound, the derived formulas (3) and (4) use the static value of CWmax according to EDCF along with the following formula:

$$newCW_{max}[i] = min(newCW_{max}[i], maxPHYCW_{lim}) \qquad (5)$$

Where $\mathrm{maxPHYCW}_{\mathrm{lim}}$ 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 (3) and (4) 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 $\mathrm{newCW}_{\mathrm{max}}$[i] increases and so with the value of *i*.

After each unsuccessful transmission the contention window for a access category *i* is set as the following:

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

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, $\mathrm{newCW}_{\mathrm{max}}$[i].

2.1.2 Setting CW after each successful transmission

After each successful transmission, the basic EDCF mechanism sets the contention window related to a certain access category *i*, to the CWmin[*i*].

$$CW[i] = CW_{min}[i] \qquad (7)$$

So, in this case we re-use the basic EDCF scheme.

3. Simulation topology and results

We implemented our proposed scheme in ns-2 simulator [7] using the Atheros EDCF semi-package to support QoS enhancement feature. This section presents the generic simulation topology used in order to evaluate the performance of dynamic $\mathrm{CW}_{\mathrm{max}}$ adaptation as well as a detailed analysis of the results.

3.1 Generic simulation topology

We use a generic topology shown in figure 3, which consists of *n* stations indexed from *1* to *n*. Each station generates three type of data streams, labelled 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 queue (or 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 queue sending rate is 128Kbps which represents a packet size of 200 bytes and an inter-arrival packet of 12.5 ms.

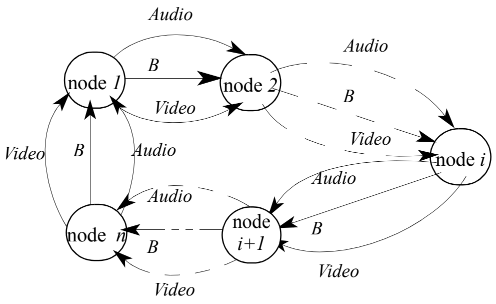

Fig. 3. Simulation topology

RTS/CTS mode is not used and all nodes are within the same 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 500X500m^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(µs)	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.

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

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

In the following simulations, we assume that each QSTA operates at IEEE 802.11a PHY mode 6 [11] (i.e., modulation 16-QAM, coding rate of 3/4, data rate of 36 Mbps).

3.2 Analysis of the impact of the smoothing factor and the update period

As described in section 2 our scheme uses two main parameters, a smoothing factor α and an update period T_{update}. We have done several simulations in order to evaluate the effect of both parameters. In order to analyse the effect of the smoothing factor 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. In order to have a confidence in obtained results, we run 20 simulations and results are averaged over these simulations. The goodput is defined as the total application layer received bytes divided by total simulation time. Figure 4 and 5 show resp. the effect of smoothing factor on total goodput and mean audio delay.

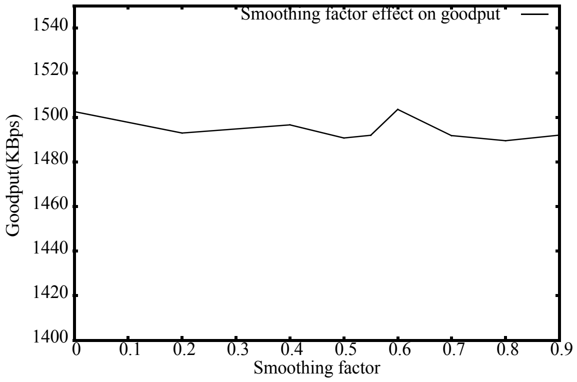

Fig. 4. Effect of smoothing factor on goodput

We can note that from the variation of goodput with the smoothing factor, a value of α in the range [0.55, 0.65] achieves a higher goodput. Also, we have higher goodput for values of α in the range of [0, 0.2]. Since, small α values could contribute to random fluctuations we consider only values in the range of [0.55, 0.65].

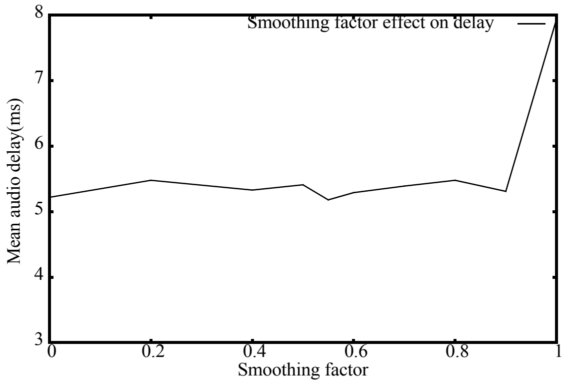

Fig. 5. Effect of smoothing factor on audio delay

Figure 5 shows that a smoothing factor value in the range [0.6, 0.9] achieves small mean audio delay as well as values in the range [0, 0.2]. Therefore, we can note that values in the range of [0.6, 0.7] achieves a best trade-off between higher total goodput and low mean audio delay. So in the following simulations we set α to 0.6.

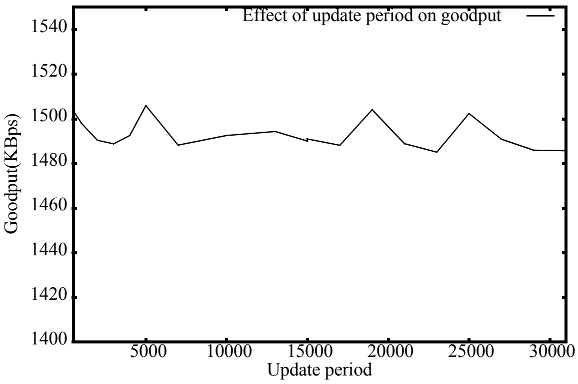

Fig. 6. Effect of update period on goodput

Figure 6 and 7 show the variations of resp. total goodput and mean audio delay with update period values expressed in time-slots. The choice of the value of update period, *Tupdate,* should take into account that higher values make adaptations less useful and small values could hurt the adaptation scheme since high frequent updates of CW_{max} could be influenced by channel fluctuations.

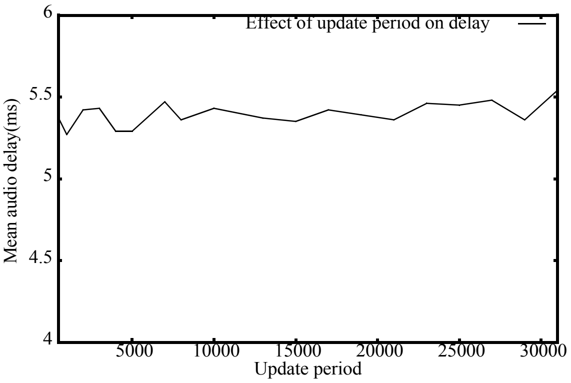

Fig. 7. Effect of update period on delay

We can note that a trade-off between higher goodput and low latency (mean audio delay) can be achieved by choosing an update period T_{update} in the range [5000, 8000]. In the following simulations, we set the value of the update period T_{update} to 5000.

3.3 Simulation results

In order to evaluate the performance of the dynamic CW_{max} adaptation scheme, we investigate in this section the impact of traffic load and compare it to the basic EDCF 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 can detect the transmission from any other station.

The following metrics are used to evaluate the performance of the proposed scheme:

- Gain of goodput: stands for the gain (in *%)* on the average goodput (AG) of the proposed scheme (CW_{max} adaptation) compared with basic EDCF:

$$Gain\_of\_goodput = \frac{AG_{DCWmax} - AG_{EDCF}}{AG_{EDCF}}$$

- Mean delay: stands for the average delay for all flows having the same priority in the different stations, e.g., mean audio delay. This metric is used to evaluate how well the scheme can accommodate real-time flows.
- Collision rate: represents the number of collisions per second.
- Medium utilization (M_u): the medium utilization represents the percentage of time used for the transmission of data frames. it is given by:

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

For the different scenarios used in this section, all three traffic categories (associated with the three access categories) are launched at around 3.0 seconds with small individual offsets to have accurate CDFs (Cumulative Distribution function) of the latency. The simulation duration is 18 seconds. Values of α and *Tupdate* are respectively 0.6 and 5000 time slots. We run 20 simulations and results are averaged over these simulations.

Figure 8 shows the mean audio delay as a function of traffic load for both CW_{max} adaptation scheme and EDCF. The mean audio delay improves a lot in CW_{max} compared to EDCF. Indeed, CW_{max} adaptation scheme maintains a lower audio delay than EDCF even in low traffic load environment, i.e., for a number of stations less than 15. As the load traffic increases, CW_{max} adaptation is able to maintain a lower delay than EDCF. The audio delay in CW_{max} adaptation scheme is up to 31% lower than in EDCF for a traffic load of 30 stations and 43% lower for a traffic load of 45 stations.

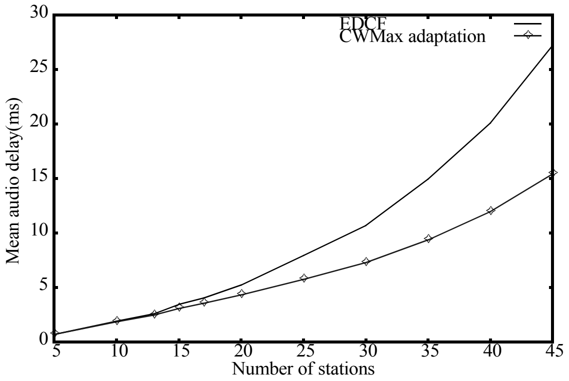

Fig. 8. Audio delay for CWmax adaptation and EDCF

This gain in delay for CW_{max} adaptation scheme can be explained by the adaptation algorithm that performs better than static CW_{max} values and especially for medium and highe loaded environment. As illustrated by figure 9 both EDCF and CW_{max} adaptation have the same mean audio delay when the traffic load is low and medium, i.e., less than 13 stations. However, there is a slight improvement in CW_{max} adaptation as the traffic load increases and especially in high loaded conditions, e.g., starting from 35 stations. Indeed, in high traffic load system, video delay is higher, by just up to 3 %, in CW_{max} adaptation than in EDCF.

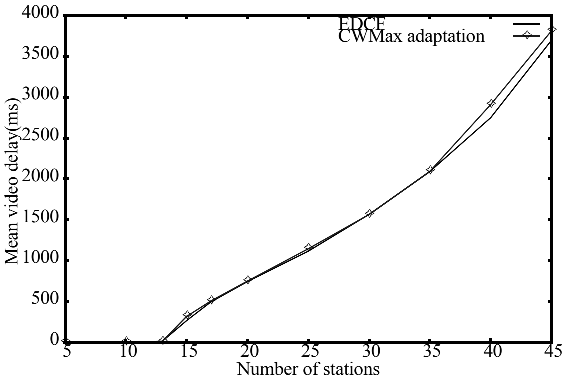

Fig. 9. Mean video delay for CWmax adaptation and EDCF

Figure 10 illustrates the collision rate in CW_{max} adaptation scheme and EDCF as a function of the traffic load. We can see that both schemes have the same collision rate for traffic loads up to 25 stations. However, there is a slight improvement for EDCF (up to 7% lower) over CW_{max} adaptation for higher traffic loads. Indeed, improvement in delay for high priority traffic is done at the cost of a slight increase in the collision rate.

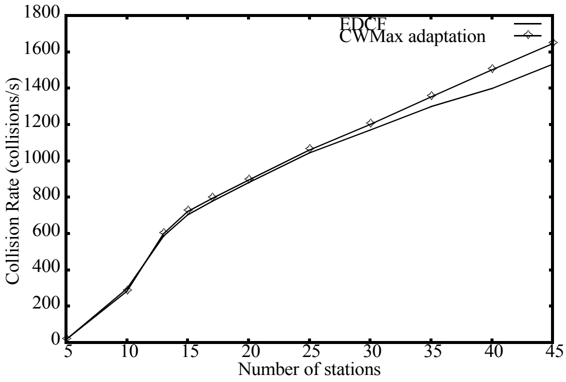

Fig. 10. Collision rate for CWmax adaptation and EDCF

Medium utilization is illustrated by figure 11 for both EDCF and CW_{max} adaptation scheme.

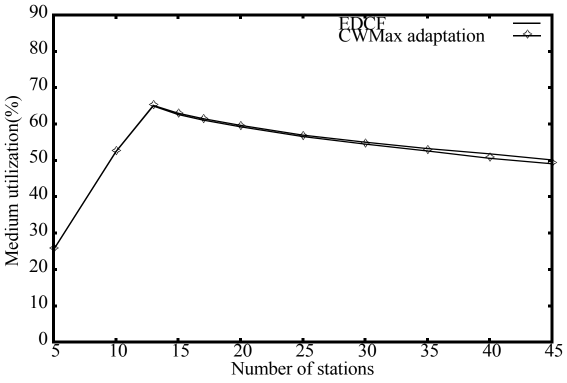

Fig. 11. Medium utilization for CWmax adaptation and EDCF

Figure 11 shows a slight improvement in medium utilization for EDCF (up to 1% higher) over CW_{max} adaptation and this is for high traffic loads, i.e., 45 stations.

Figure 12 shows the gain in goodput for CW_{max} scheme over EDCF. We can see, that both CW_{max} adaptation and EDCF have the same goodput and there is not a significant difference between both schemes. In addition, we note that there is a just a slight drop, by up to 3%, in goodput for CW_{max} compared to EDCF for high loaded environment.

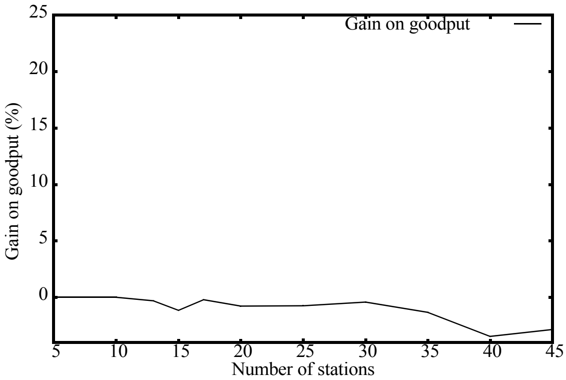

Fig. 12. Goodput gain

Results obtained related to the defined performance metrics indicate that CW_{max} adaptation performs better than EDCF and maintains a lower delay for high priority access category (i.e., AC *0* for audio). Compared to EDCF, CW_{max} adaptation, have the same performance for medium utilization and stable delay and jitter for low priority access categories. We point out that there is a slight drop in goodput in high loaded environment for CW_{max} adaptation scheme at the cost of a huge delay decrease for higher priority traffic (or access category).

We can also conclude that by tuning CW_{max} for lower priority ACs and setting a low CWmax for higher priority AC enables to enhance the performance for high priority AC and at the same time maintaining stable performance for lower priority ACs. However, a small static CWmax value for all or some access categories can not reach similar results as the dynamic adaptation scheme because it contributes to a higher collision rate and hence to higher delay and jitter for lower priority ACs.

4. Conclusions and future work

In this paper we have proposed a new dynamic scheme for the adaptation of the contention window maximum (CW_{max}) in order to enhance the service differentiation for 802.11 WLANs. We have extended the basic EDCF scheme by a distributed algorithm that enables each station to tune the size of the CW_{max} used in its back-off algorithm at run time. The tuning is differentiated for each access category *i* and performed according to the channel traffic conditions. The performances of the proposed adaptation scheme investigated by simulations have indicated that our scheme improves delay and jitter for higher priority traffic while maintaining a stable performance for lower 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 EDCF. On one hand, results have shown that audio delay associated with high priority access category, improves greatly and decreases by up to 43%. On the other hand, performance for lower priority access category, such as video traffic, remains stable. Moreover, we have observed a slight drop (up to 3%) in throughput in high system load, but throughput is stable in light and medium system load for CW_{max} adaptation scheme.

Further work could include implementation of the dynamic CW_{max} adaptation on top of AEDCF [5]. Moreover, we could deal with implementing and evaluating a hybrid adaptation approach of CW_{min} and CW_{max}. Finally, in a future work we could also investigate how to apply the proposed adaptation algorithm for infrastructure networks especially during the contention period.

5. References

[1] D-J. Deng and R-S. Chang. A priority scheme for IEEE 802.11 DCF access method. IEICE Transactions on Communications, E82-B(1), January 1999.

[2] 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.

[3] N. H. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in wireless LAN. In Sixth Annual International Conference on Mobile Computing and Networking, Boston, August 2000.

[4] 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.

[5] Romdhani, L., Ni, Q., and Turletti, T., "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks ", in Proceedings of the IEEE WCNC'03.

[6] Imad Aad and Claude Castellucia. Differentiation mechanisms for IEEE 802.11. In Proceedings of IEEE Infocom 2001, Anchorage, Alaska, April 2001.

[7] Ns-2 network simulator home page. available at: http://www.isi.edu/nsnam/ns/

[8] A. Veres, A.T. Campbell, M. Barry, and L-H. Sun. 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.

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

[10] IEEE WG home page: http://grouper.ieee.org/groups/802/11

[11] IEEE WG, "802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11 Standard", Sept. 1999.

[12] 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).

[13] F. Cali, M. Conti and E. Gregori. IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism. In IEEE JSAC, vol. 18, no. 9, september 2000.