

DYNAMIC TUNING OF THE CONTENTION WINDOW MINIMUM (CW_{MIN}) FOR ENHANCED SERVICE DIFFERENTIATION IN IEEE 802.11 WIRELESS AD-HOC NETWORKS

Lassaad Gannoune and Stephan Robert

Ecole d'Ingénieurs du Canton de Vaud (EIVD), Rue de Galilée 15, CH-1400 Yverdon-les-Bains, Switzerland
 {lassaad.gannoune, stephan.robert}@eivd.ch

Abstract- The proposed IEEE 802.11e draft standard defines new MAC protocols for QoS in wireless networks, mainly HCF and EDCF. 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 minimum contention window (CW_{min}) that enables each station to tune the size of the CW_{min} used in its back-off algorithm at run time. The purpose of our scheme is to reduce delay and jitter and increase the efficiency of the transmission channel. Priorities between access categories are provisioned by tuning the size of the CW_{min} according to application requirements and channel conditions. The performances of the IEEE 802.11e EDCF, enhanced with our adaptation algorithm, are extensively investigated by simulations. Results obtained indicate that CW_{min} adaptation scheme outperforms the 802.11e EDCF standard in terms of channel utilization, throughput, and packet delay. Indeed, the delay for high priority access category decreases by up to 30%, total throughput increases by up to 18%, and channel capacity is at least 15% higher.

Keywords: IEEE 802.11e, QoS, EDCF, back-off algorithm, contention window minimum (CW_{min}).

I. 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]. Indeed, 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] and [5]. It is part of HCF for infrastructure networks and 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 Category 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 provisioned 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 different contention window (CW) parameters. 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} .

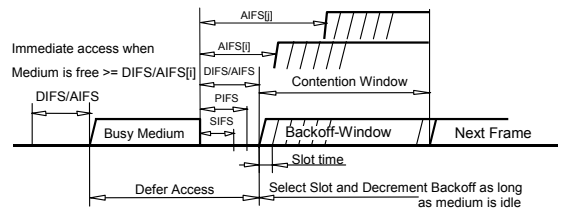


Fig. 1. Some IFS Relationship

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 as shown in equation (1):

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

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 differently 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.

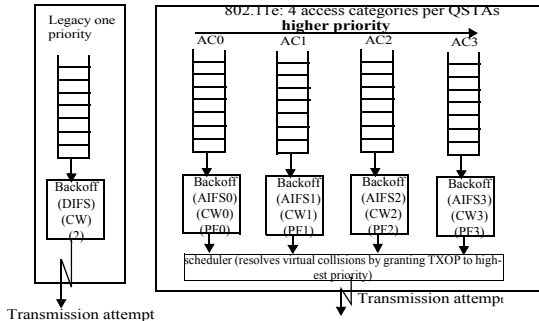


Fig. 2. Queue based EDCF vs legacy DCF

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 tuning of the minimum contention window (CW_{min}) with a special designed scheme. We compare the performance of the proposed scheme with both EDCF and the Slow Decrease scheme (SD) [14].

This paper is organized as follows. In section II, we present the proposed CW_{min} adaptation scheme. In section III, the simulation topology and parameters are described. We discuss the simulations results and the performance of our scheme in section IV. Finally, in section V we give some concluding remarks along with our future work plan.

II. CW_{MIN} ADAPTATION

Both in the legacy DCF [11] and EDCF [2], upon a successful transmission, the backoff algorithm reduces the contention window size to CW_{min} . The issue is that a such aggressive reduction of the CW could lead to more collisions when the transmission channel is loaded or in a congested state. The main idea behind the dynamic adaptation of CW_{min} is to adapt periodically the $CW_{min}[i]$ value for a certain access category i to the traffic load and channel conditions. In the rest of the paper, we assume 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.

The purpose of our scheme is to dynamically adapt the CW_{min} for each access category i by setting a higher value to $CW_{min}[i]$ when the channel is estimated to be congested and a smaller one (closest to its static $CW_{min}[i]$ value according EDCF) when the channel load is estimated to be low. So, we believe that adapting the $CW_{min}[i]$ parameter according to the traffic load will lead to reduce the overall

number of collisions and reduce the delay and jitter for the TCs in the different access categories.

A. Scheme description

In the basic EDCF scheme [4],[2], after each successful transmission the contention window is reduced to $CW_{min}[i]$. So, we propose that upon successful transmission of a frame from an access category i , we compute a dynamic value of the minimum contention window, i.e., $DCW_{min}[i]$ and the contention window is reduced to that value. We note that we use both an adaptive mechanism for computing the $DCW_{min}[i]$ according to both traffic load and the static value of $CW_{min}[i]$. There is also a relationship between $DCW_{min}[i]$ and the priority level i related to the corresponding access category.

B. Setting CW after each successful transmission

In our scheme we propose that each access category updates each $CW_{min}[i]$ parameter in an adaptive way using the estimated collision rate at regular update period T_{update} expressed in time slots. In the proposed scheme we re-use the same method defined in [5] to estimate the average collision rate as seen by a station p . Instantaneous collision rate f_{curr}^j at the j^{th} update period T_{update} is calculated using the number of collisions and the number of packets sent during that period and given by (2):

$$f_{curr}^j = \frac{Num(collisions_j[p])}{Num(data_sent_j[p])} \quad (2)$$

Where $Num(collisions_j[p])$ is the number of collisions at station p during the period (update period or step) j and $Num(data_sent_j[p])$ 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_{curr}^j). Equation (3) gives the corresponding value of collision rate at step j .

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

Where α is a smoothing factor in the range $[0, 1]$, j refers to the j^{th} update period T_{update} . f_{curr}^j stands for the instantaneous collision rate. Using the estimated average of collision rate f_{avg}^j at step j the numerical expression of the proposed scheme for $CW_{min}[i]$ adaptation is presented in equation (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 $DCW_{min}[i]$ stands for the adaptive value of contention window minimum for an access category i , $CW_{min}[i]$ is the minimum contention window (according to EDCF) assigned for the same access category i and \hat{p}_{avg} represents the estimated collision rate at step j . We propose here to perform a *slow adaptation* for high priority access category and a *fast adaptation* for low priority access category. This leads to a fast increase in CW_{min} for lower priority ACs and a slow increase in CW_{min} for higher priority ACs. Both slow adaptation and fast adaptation of CW_{min} can be achieved by introducing the level of priority i for an access category in the formula of DCW_{min} .

The dynamic contention window minimum for AC i obtained in equation (4) varies between a lower bound of $CW_{min}[i]$, when the collision rate equals to zero, and an upper bound of $(CW_{max}[i] - CW_{min}[i]) * 2^{i-2}$, when the collision rate equals to 1. Thus, this upper bound depends on the priority level of the access category and limits the increase of the $DCW_{min}[i]$ which results in a slow adaptation of higher priority traffic. Indeed, this upper bound of $DCW_{min}[i]$ 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 equations 4) uses the static value of CW_{max} according to EDCF along with the following formula:

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

We note that, this is not useful for our specific simulation scenario, since we have only used three access categories and the upper bound of $DCW_{min}[3]$ is $(CW_{max}[3] - CW_{min}[3])$ which is obviously less than $CW_{max}[3]$.

C. Setting CW after each collision

In our scheme we are using a PF which equals to 2 for all access categories, so that, the contention window is doubled while remaining less than the maximum contention window, i.e, $CW_{max}[i]$.

$$CW_{new}[i] = \min(CW_{max}[i], 2 \times CW_{old}[i]) \quad (6)$$

In this case, we do not modify the basic EDCF scheme as illustrated by equation (III).

III. SIMULATION PARAMETERS AND TOPOLOGY

We implemented our proposed scheme under the network simulator ns-2 [7] using EDCF semi-package to support QoS enhancements features (provided by Atheros). This section presents the generic simulation topology used in order to evaluate the performance of the dynamic CW_{min} adaptation scheme as well as a detailed analysis for a proper selection of α and T_{update} parameters used in the proposed scheme.

A. Generic 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 UDP CBR 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). The highest priority queue 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 traffic queue, 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 queue sending rate is 128Kbps which represents a packet size of 200 bytes and an inter-arrival packet of 12.5 ms.

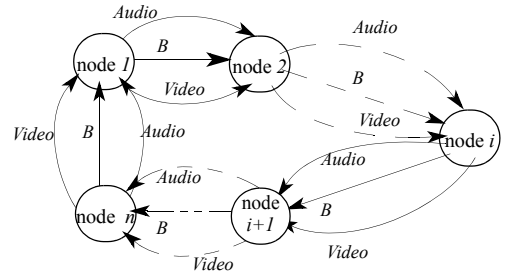


Fig. 3. Simulation topology

RTS/CTS mode is not used. In addition, 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 500×500 m² dimensions in 2D space. Table 1 shows the different MAC parameters for the three access categories used for all QoS STA and in the different simulation scenarios.

Table 1. MAC parameters for the three ACs

Parameters	High	Medium	Low
CW_{min}	7	15	31
CW_{max}	200	500	1023
AIFS(μ 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

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).

B. Impact of the smoothing factor and the update period

As described, in section II, the proposed scheme uses two main parameters, a smoothing factor α and an update period T_{update} . In order to select proper parameters, we have done a several simulation experiments. First, to deal with 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. Simulations are run for a fixed number of stations, i.e., 20 stations. Results are averaged over 20 simulations. Two performance criteria are used, the total throughput (or goodput) and the mean audio delay.

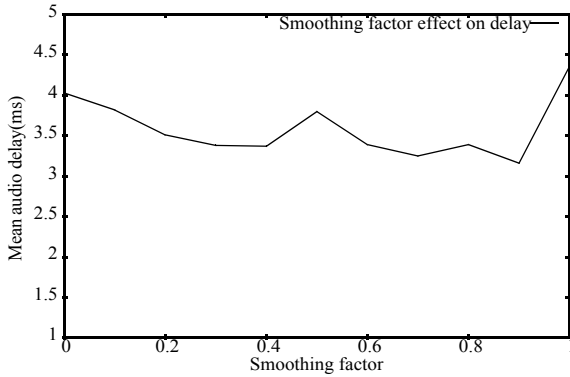


Fig. 4. Impact of smoothing factor on audio delay

Goodput is defined as the total application layer received bytes divided by the total simulation time. Figure 4 and 5 show resp. the effect of smoothing factor on total goodput and mean audio delay. It can be seen in figure 4 that, a value of α in the range $[0.6, 0.9]$ achieves a lower delay and the lowest audio delay is for $\alpha=0.9$. From figure 5, we can see that, a value of α in the range $[0.55, 0.7]$ achieves a higher goodput with a maximum of goodput for $\alpha=0.6$. In addition, 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 $[0.55, 0.7]$.

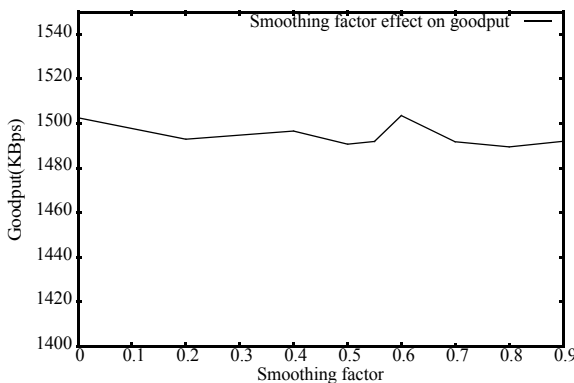


Fig. 5. impact of smoothing factor on goodput

Therefore, 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 resp. total goodput and mean audio delay as a function of the update period expressed in time-slots.

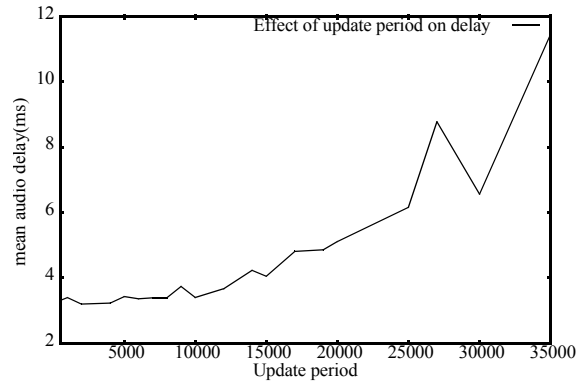


Fig. 6. Impact of update period on audio delay

The selection of the update period, T_{update} , value should take into account that higher values make adaptations less useful and smaller ones could hurt the adaptation scheme since high frequent updates of CW_{min} could be influenced by channel fluctuations. As illustrated in figure 6, lower audio delay is achieved with values of T_{update} in the range of $[500, 10000]$ time-slots with a lower audio delay at $T_{update}=4000$ time-slots.

Figure 7 shows the total goodput as a function of the update period and it can be seen that we have a higher goodput for update periods in the range of $[500, 6000]$ and also for T_{update} value of 500 time-slots. So, a value of $T_{update}=4000$ time-slots achieves a trade-off between a higher goodput and a lower delay.

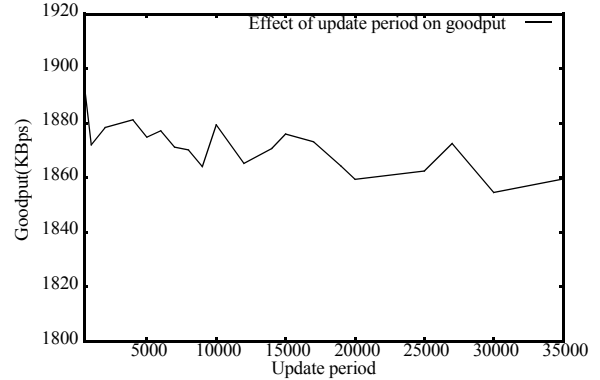


Fig. 7. impact of update period on goodput

In the following simulations we set α and T_{update} values resp. to 0.6 and 4000 time-slots.

IV. SIMULATION RESULTS

In order to evaluate the performance of the dynamic CW_{min} adaptation scheme, we investigate in this section the impact of traffic load and compare it to the basic EDCF

scheme. The different access categories ACs used for simulations are described in Table 1. We simulate various loads of the system by instantiating the simulation topology in figure 3 for a special number of stations. All the stations are located within the same Independent Basic Service Set (IBSS), so that, every station can detect the transmission from any other station. The following QoS metrics are used to evaluate the performance of the different simulations:

- Gain of goodput: stands for the gain (in %) on the average goodput (AG) of the proposed scheme (DCWmin) compared with basic EDCF:

$$\text{Gain_of_goodput} = \frac{AG_{DCWmin} - AG_{EDCF}}{AG_{EDCF}}$$

- Mean delay: stands for the average delay of all 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.
- Collision rate: represents the average number of collisions that occurs per second.
- Medium utilization (M_u): the medium utilization represents the percentage of time used for the transmission of data frames and it is given by:

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

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 latency. The simulation duration is 18 seconds. In order to have confidence in results obtained by simulations, we run 15 simulations and obtained results are averaged on these simulations.

In the following subsections, the performance comparisons are done between CW_{min} adaptation scheme, *Slow Decrease (SD)* [14] and *EDCF* schemes. In the SD scheme, CW is decreased more slowly than in EDCF, i.e., upon each successful transmission $CW_{new} = \max\{CW_{min}, 0.5 \times CW_{prev}\}$.

A. Channel utilization comparisons

Figure 8 shows the collision rate for CWmin adaptation, SD and EDCF. The collision rate is the same for CWmin adaptation, SD and EDCF for a very low traffic load, i.e., 5 stations. As the traffic increases, the collision rate in CWmin adaptation maintains a lower increase than in SD or EDCF and that starting from a system load of 10 stations. It can be seen that, for 25 stations, the collision rate in CWmin adaptation is 40% lower than in EDCF.

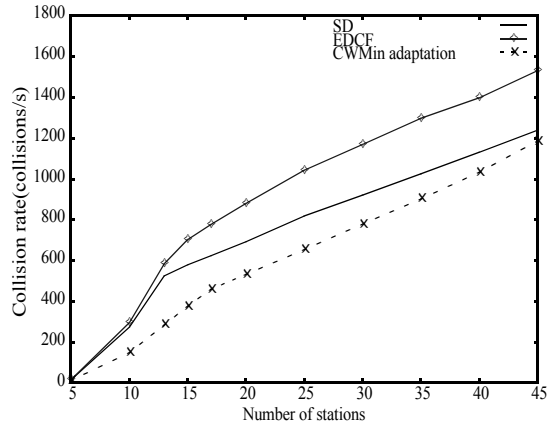


Fig. 8. Collision rate for CWmin adaptation, SD and EDCF

We believe that the dynamic adaptation of CWmin has contributed to reduce the number of collisions in the IBSS. Figure 9 shows the medium utilization for CWmin adaptation scheme, SD and EDCF. It can be seen that under most system loads, CW_{min} adaptation scheme has much better channel utilization than both SD and EDCF. This is, because CWmin adaptation adjusts the size of $CW_{min}[i]$ upon a successful transmission according to the network condition. However, EDCF just blindly resets $CW[i]$ to a static $CW_{min}[i]$ and as a result contributes to more collisions when the system load is not very low (as shown in figure 8). Furthermore, SD improves the medium utilization compared to EDCF but it is still under CW_{min} adaptation. This can be explained by the fact that SD adjusts statically the size of CW and not according to the network load, hence contributing to more collisions than our scheme when the system is loaded.

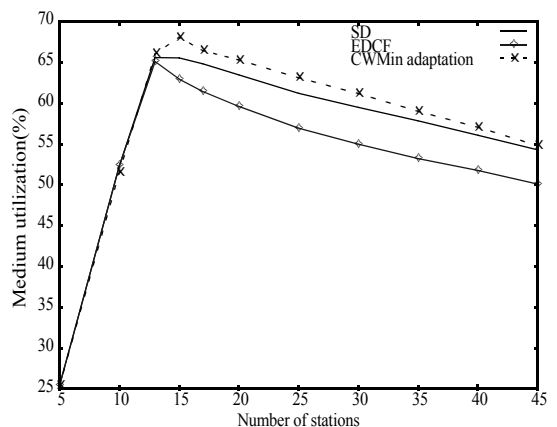


Fig. 9. Medium utilization for CWmin adaptation, SD and EDCF

The capacity in CWmin adaptation is higher than in EDCF (maximum channel utilization in EDCF is reached for 13 stations while in CWmin adaptation corresponds to 15 stations). Thus, the channel capacity is 15% higher than in EDCF. Furthermore, the maximum channel utilization

reached in EDCF (65.11%), corresponds to a channel capacity of 13 stations and 20 stations for CW_{\min} adaptation which leads to a gain of up to 54% in channel capacity.

B. Throughput comparisons

Figure 10 shows the gain of goodput for CW_{\min} adaptation and SD over EDCF.

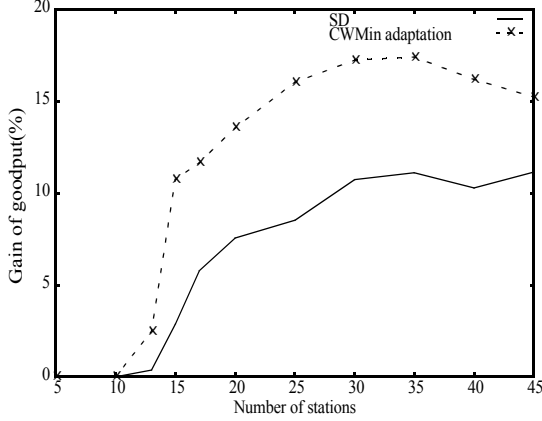


Fig. 10. Gain of goodput for CW_{\min} adaptation and SD over EDCF

The goodput improves in CW_{\min} adaptation and the gain of goodput for CW_{\min} adaptation over EDCF is up to 18%. Furthermore, the gain increases when traffic load is greater than 10 stations as shown in figure 10. Moreover, throughput for CW_{\min} adaptation is much better than for SD. So, according to system throughput CW_{\min} adaptation outperforms both SD and EDCF. The higher performance in throughput for CW_{\min} adaptation is due to the increase in channel utilization because of the CW_{\min} dynamic adaptation algorithm.

C. Packet delay comparisons

In this subsection, we compare the average packet delay under EDCF, SD and CW_{\min} adaptation scheme. Figure 11 shows the mean audio delay (high priority traffic) as a function of traffic load for both CW_{\min} adaptation, SD and EDCF. The mean audio delay improves significantly in CW_{\min} compared to EDCF. Indeed, CW_{\min} scheme maintains a lower audio delay than EDCF even in low traffic load, i.e., for a number of stations less than 10. As the traffic load increases, CW_{\min} is able to maintain a lower delay than EDCF.

The audio delay in CW_{\min} adaptation scheme is 34% lower than in EDCF for a traffic load of 30 stations and results in lower delay and jitter for high priority access categories. However, the SD scheme does not improve the audio delay (high priority AC), but contributes to a higher delay than EDCF.

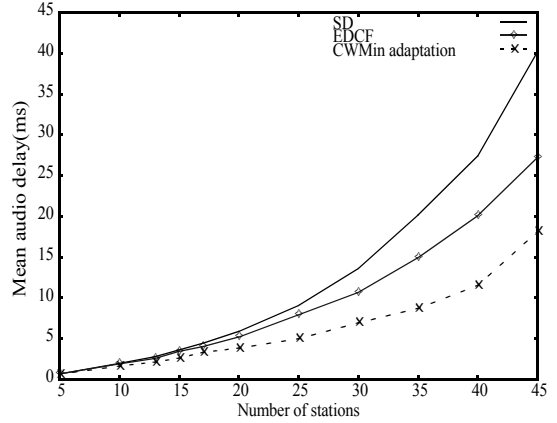


Fig. 11. Mean audio delay for CW_{\min} adaptation, SD and EDCF

This can be explained by the fact that SD does not adapt the CW according to traffic load (network condition) and the CW is decreased less aggressively than in EDCF. Moreover, the SD scheme may lead to poor performance when applied without PF differentiation. Figure 12 shows the mean video delay as a function of the traffic load for EDCF, SD and CW_{\min} adaptation scheme. As it can be seen, there is an improvement of the mean video delay (medium priority traffic) in CW_{\min} adaptation compared to EDCF. Both EDCF, SD and CW_{\min} adaptation have the same mean video delay when the traffic load is low, i.e., less than 13 stations.

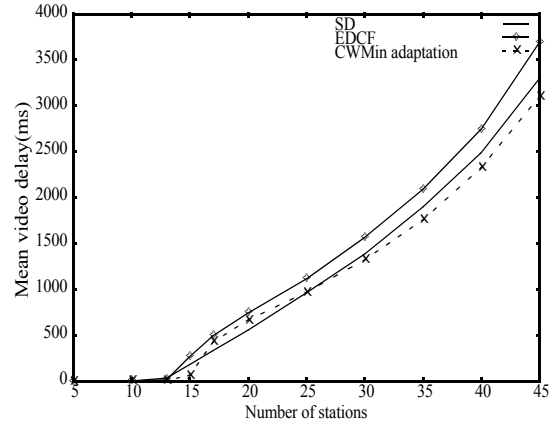


Fig. 12. Mean video delay for CW_{\min} adaptation, SD and EDCF

However, the delay improves in CW_{\min} adaptation as the traffic load increases. The video delay is 75% lower (65.08ms in our scheme and 267.44ms in EDCF) in CW_{\min} adaptation scheme than in EDCF for a system load of 15 stations. This can be explained by the dynamic adaptation algorithm used to adjust the size of $CW_{\min}[i]$ that performs better than a static CW_{\min} in medium and loaded channel system. However, the SD scheme maintains a lower video delay than CW_{\min} adaptation for only traffic loads between 16 and 25

stations. So, SD performs better than CWmin adaptation for medium priority traffic only after the system reaches its capacity. This is likely due to the aggressive (i.e., *fast adaptation* for lower priority) CWmin adaptation algorithm for medium priority AC when system load increases (i.e., when the collision rate increases). Otherwise, CWmin adaptation maintains lower delay than SD scheme.

From the different simulation experiments, we can conclude that CWmin adaptation scheme outperforms both EDCF and SD schemes in light, medium and high system load before and after the saturation of the channel system. Moreover, the channel capacity improves and is 15% higher than in EDCF. We note that SD scheme shows limitations when applied without PF differentiation according to ACs. Furthermore, in contrast to SD or AEDCF [5], our scheme can be easily applied to IEEE 802.11e [2] standard infrastructure networks since it does not use any PF differentiation and at the same time achieves better performance than EDCF and SD.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a new dynamic scheme for the adaptation of the minimum contention window (CW_{min}) in order to enhance the service differentiation for 802.11e WLANs. We have extended the basic EDCF scheme by a distributed algorithm that enables each station to tune the size of the CW_{min} used in its back-off algorithm at run time. The performances of the proposed adaptation scheme investigated by simulations have indicated that our scheme outperforms both EDCF and SD scheme and improves delay and jitter for all access categories. The mean audio delay in CW_{min} adaptation scheme is up to 34% lower than in EDCF. Also, the total throughput is improved by up to 18% and the overall channel capacity is 15% higher than in EDCF.

In a future work, we will enhance the CWmin dynamic adaptation with a more appropriate *slow adaptation* and *fast adaptation* functions and compare obtained performances with related work including CW adaptation schemes. Furthermore, based on this work, we are going to investigate how the proposed scheme can be adapted for infrastructure networks. We intend also to design and implement a hybrid adaptation approach of CW_{min} and CW_{max} parameters and study its performance.

ACKNOWLEDGEMENTS

This work has been supported by the Switzerland HES-SO Ra&D fund with grant number QoS-1505. Also, many thanks go to Swisscom Innovations for supporting this work.

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 Turetletti, T., "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks", in the Proceedings of the IEEE WCNC'03.

[6] I. Aad and C. 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.

[14] I.Aad, Q. Ni, C. Castelluccia, and T. Turetletti. "Enhancing IEEE 802.11 performance with slow CW decrease". IEEE 802.11e working group document 802.11-02/674r0, Nov. 2002.