Dynamic Contention Window Adaptation (DCWA) in IEEE 802.11e Wireless Local Area Networks

Rastin Pries, Stefan Menth, Dirk Staehle, Michael Menth, Phuoc Tran-Gia
University of Wuerzburg, Institute of Computer Science, Wuerzburg, Germany
Email: pries@informatik.uni-wuerzburg.de

***Abstract*— The Wireless LAN Medium Access Control (MAC) protocol is responsible for an efficient sharing of the limited communication bandwidth. With the demand for Quality of Service (QoS), which is supported by different traffic classes in 802.11e, this task becomes even more complex.**

In this paper, we present a Dynamic Contention Window Adaptation (DCWA) algorithm which adapts the contention window parameters for a single traffic class such that they are as small as possible to meet realtime conditions and large enough to cope with the current network load. The results reveal that the DCWA increases the overall cell capacity significantly compared to the fixed IEEE standard settings, it adaptively minimizes the contention delay when possible and maximizes the throughput when needed. Furthermore, we extended the method to handle competing high and low priority traffic and showed that the new mechanism can effectively protect voice from best effort traffic.

***Index Terms*— 802.11e, Contention Window Adaptation, QoS**

I. INTRODUCTION

The continuous standardization of *Wireless Local Area Networks* (Wireless LANs) is a success story. Since the first release of the IEEE 802.11 Wireless LAN standard in 1997 [1], it gradually improved its performance and evolved into a very flexible and well-understood technology. Today, this wireless technology is a standard equipment in laptops and other portable and mobile devices. They provide convenient wireless access to the Internet for users at home, in public facilities, and emerge more and more in everyday's life.

The steady growth of Wireless LAN and the rising popularity of multimedia applications indicate that there will be an increasing demand for wireless communication supporting *Quality of Service* (QoS). However, due to the limited nature of the radio spectrum, the demand for wireless resources is likely to surpass the available wireless resources in the future. Therefore, an efficient usage of the available resources to support a certain QoS level is inevitable.

The IEEE 802.11 working group published the IEEE 802.11e [2] extension in late 2005 which enables service differentiation in Wireless LAN to a limited extent. However, it does not provide QoS guarantees. One reason for this is the lack of a load control for Wireless LAN. Further, resource efficiency has severely decreased through the service differentiation extension due to the use of small and static *Channel Access Parameters* (CAPs). As a result, time-varying loads cause heavily varying contention levels leading to an inefficient channel usage. In the worst case, traffic performance is degraded and QoS requirements cannot be met.

In this paper, we propose a measurement-based parameter adaptation scheme which succeeds in achieving a much better resource efficiency as compared to the standard. At the same time, service differentiation is maintained and even QoS guarantees can be given to a certain extent. Resource efficiency is achieved through a dynamic CAP adaptation process according to the current channel contention level at runtime. Updates of the CAPs resulting from adaptations are broadcast via beacon frames. The mechanism is called *Dynamic Contention Window Adaptation* (DCWA). It achieves resource efficiency by choosing an appropriate contention window with respect to the current channel contention. Thus, Wireless LAN resources available to voice flows significantly improve, become more robust, and are still protected.

This work is organized as follows. In Section II the work related to contention window adaptation mechanisms is reviewed. Section III presents the DCWA algorithm. The optimal parameters for the algorithm are evaluated in Section IV and Section V shows the performance of the algorithm for two service classes. Finally, Section VI concludes this work.

II. RELATED WORK

Contention window adaptation techniques in the literature can be mainly divided into *Multiplicative Increase / Multiplicative Decrease* (MIMD) and *Additive Increase / Additive Decrease* (AIAD) schemes. The authors in [3]–[6] use MIMD. The multiplication factor is either defined by a function of the priority and the collision rate, or simply by using a fixed value. The authors in [7]–[9] use AIAD and determine additive changes of the contention window through the collision rate, the priority, the distance between the *minimum contention window* (CWmin) and the *maximum contention window* (CWmax), or simply use fix values. The complex MIMD and AIAD methods estimate the ratio between the collision rate and the contention window size. However, in practice the emerging patterns might change and are not captured properly in these a priori relations.

Further, there are differences of how to change the contention windows based on measurements. A common method is a threshold-based approach [4]–[7]. An alternative to this is to define the contention window directly as a function of parameters such as the collision rate, the number of stations,

---

This work is funded by Deutsche Forschungsgesellschaft (DFG) under grant TR 257/19-2. The authors alone are responsible for the content of the paper.

and the priority [3], [8], [9]. The problem with the latter methods is that the parameter correlation for the contention window adaptation is not clear.

III. Dynamic Contention Window Adaptation

For our measurement-based approach, the CAPs are set according to the channel status. The *Access Point* (AP) adapts the CWmin and CWmax based on measurements of the average collision probability of the access category for voice traffic. Since the AP decides about the parameter updates and distributes them from there, it is most convenient to conduct the collision probability measurements there as well. If we can assume that each station including the AP experiences a similar collision rate and delay, then we could rely on measurements at the AP. However, in [10], we discovered that the AP can behave very differently compared to its associated stations. Due to queuing and contention effects that are differently pronounced at an individual station and the AP, the collision probabilities and delays can differ significantly.

The collision probability perceived at the AP and at a station can differ up to a factor of ten. This means that a contention window which is efficient regarding the collision probability of the AP may entail a high collision rate at its associated stations, leading to a poor overall channel utilization. We can cope with this unfair channel access using an explicit feedback mechanism from the stations. The feedback contains measurements of their individual collision probability and are transmitted in the Wireless LAN header block. Based on that feedback, the AP is able to make reasonable contention window adaptation decisions and broadcasts the new parameters in the beacon frames. Thereby, the contention status is assessed using the reported feedback from the stations. The AP maintains a hash table containing the most recent feedback of all stations. When the AP receives a frame from a station s containing feedback information $R_{i,s}$ related to beacon interval i, it inserts the value pair $(s, R_{i,s})$ into the hash table. Any older values $R_{j,s}$, $j < i$ are overwritten. Thus, the hash table reflects the most recent contention status of all stations. The contention window adaptation decision itself is performed based on the maximum value of the reported feedback. This metric is defined as follows.

1) Contention Window Control Parameter (CWCP): In beacon interval i, the hash table of the AP may contain feedback from a station $s \in S$, where S is the set of all stations in the *Basic Service Set* (BSS). A feedback value $R_{j,s}$ refers to the smoothed average number of retransmissions throughout beacon interval j $(j < i)$ at station s. Then, R_i^{max} is the CWCP in beacon interval i and defined by

$$CWCP := R_i^{max} = \max_{s \in S}(R_{j,s}) \qquad (1)$$

The value R_i^{max} reflects the retransmission level of the worst station within the BSS and serves as input for the DCWA algorithm.

We briefly summarize the described feedback mechanism in Fig. 1. The stations report estimates $R_{i,s}$ of their smoothed, empirical collision probability to the AP. The AP stores the received feedback in a hash table. The channel status is assessed by the maximum metric R_i^{max} as defined in Equation 1. The DCWA algorithm at the AP determines an appropriate contention window based on R_i^{max}. Updates of the *Enhanced Distributed Channel Access* (EDCA) parameter set are then distributed via the beacon frame throughout the network.

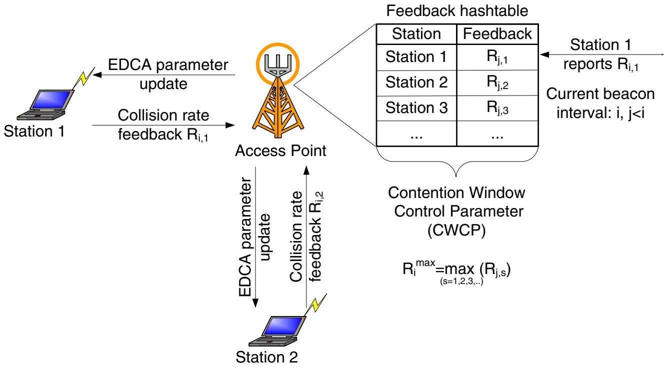


Fig. 1. Feedback mechanism supporting EDCA parameter adaptation

2) DCWA Algorithm: The objective of the DCWA algorithm is to keep the CWCP within a target range over time, independently of the current network load. This target range represents the preferred operation condition where the contention in the network is within an efficient range. Algorithm 1 describes the DCWA in detail. Before broadcasting a beacon frame, the AP performs the DCWA procedure. The maximum value R_i^{max} of the reported feedback contained in the hash table serves as an input variable for the DCWA. Initially, the AP distributes the default IEEE 802.11e parameters. The contention windows are updated - increased or decreased depending on the current value of R_i^{max}. An update can be repeated after the *inter-adaptation time* τ has elapsed.

Motivated by the above discussion, the proposed DCWA uses a threshold-based contention window adaptation, and we suggest a simple MIMD method using a fixed multiplicative factor. Only the voice and the best effort *Access Categories* (ACs) are taken into account. This is not critical because the algorithm could be extended to manage other traffic categories. In case the contention R_i^{max} exceeds an upper control threshold θ_{up}, the CWmin and the CWmax of both *Access Category Voice* (AC_VO) and *Access Category Best Effort* (AC_BE) are doubled respecting the rules in the IEEE 802.11e standard [2]. The contention window parameters are increased as long as $CWmin\,[AC\_VO]$ does not exceed $maxCWmin\,[AC\_VO]$. In case the contention R_i^{max} drops below a lower control threshold θ_{lo}, the CWmin and the CWmax of both AC_VO and AC_BE are decreased by half. They can be decreased as long as $CWmin\,[AC\_VO]$ reaches $minCWmin\,[AC\_VO]$. Then, the CWmin and the CWmax are set again to the default values. The DCWA relies on measurements and thresholds to choose an appropriate contention window. The stability and efficiency of the hysteresis is determined by the following four parameters:

Algorithm 1 DCWA Algorithm

```
$R_i^{max}$: CWCP in beacon interval i
$\theta_{up}$: upper control threshold triggering CW increase
$\theta_{lo}$: lower control threshold triggering CW decrease
$maxCWmin\,[AC\_VO]$: maximum $CWmin\,[AC\_VO]$
$minCWmin\,[AC\_VO]$: minimum $CWmin\,[AC\_VO]$
$last\ \ CW\ \ update\ \ time$: time of the last contention window update
$\tau$: inter-adaptation time, minimum time to elapse before the next update

if $(current\ time - last\ CW\ update\ time) > \tau$ then
  if $R_i^{max} > \theta_{up}$ and $CWmin\,[AC\_VO] < maxCWmin\,[AC\_VO]$ then
    $CWmin\,[AC\_VO] = 2 \cdot CWmin\,[AC\_VO] + 1$
    $CWmax\,[AC\_VO] = 2 \cdot CWmax\,[AC\_VO] + 1$
    $CWmin\,[AC\_BE] = 2 \cdot CWmin\,[AC\_BE] + 1$
    $CWmax\,[AC\_BE] = 2 \cdot CWmax\,[AC\_BE]) + 1$
  else if $R_i^{max} < \theta_{lo}$ and $CWmin\,[AC\_VO] > minCWmin\,[AC\_VO]$ then
    $CWmin\,[AC\_VO] = (CWmin\,[AC\_VO] - 1)/2$
    $CWmax\,[AC\_VO] = (CWmax\,[AC\_VO] - 1)/2$
    $CWmin\,[AC\_BE] = (CWmin\,[AC\_BE] - 1)/2$
    $CWmax\,[AC\_BE] = (CWmax\,[AC\_BE] - 1)/2$
  else
    channel contention $R_i^{max}$ is within target range $\theta_{lo} \leq R_i^{max} \leq \theta_{up}$
  end if
end if
```

- *M*: memory of the Time-Exponentially Weighted Moving Average (TEWMA) [11] and smoothing factor for R_i^{max}
- τ: inter-adaptation time, minimum time between two consecutive adaptations
- θ_{up}: upper control threshold
- θ_{lo}: lower control threshold

The smoothing factor *M* determines the decay of the measured values and thus the agility of R_i^{max}. A small value of *M* assigns a higher importance to the currently measured values, which means that R_i^{max} will quickly reflect the recent contention status. With a large value of *M*, it will take longer until fundamental changes of the contention status are indicated by R_i^{max}.

After a contention window adaptation, the algorithm waits for the duration of τ before changing the contention window again. The rationale behind this is to wait until effects of the contention window change are indicated by R_i^{max}. Otherwise, there might be the risk of increasing the contention windows too quickly, resulting in an immediate decrease of the contention window. This behavior is called oscillation of R_i^{max}. In order to prevent oscillation and to obtain a stable system, it is important to jointly adjust the smoothing factor *M* and the inter-adaptation time τ. As explained before, for any network condition there is an optimal contention window that balances the costs of idle and collision time. Thus, an efficient operation condition will generate a certain level of contention. This efficient contention level shall be included in the target range of R_i^{max} which is determined by the upper and lower control thresholds θ_{up} and θ_{lo}.

For simulation purposes, the DCWA is implemented as an extension to the OPNET Modeler 12.0 Wireless LAN MAC layer [12]. The goal is to find optimal parameters in order to minimize the delay and packet loss in the network. An illustration of the DCWA algorithm and its parameters is shown in Fig. 2. The x-axis shows the simulation time in seconds. The left y-axis marks the maximum packet collision rate and the second y-axis shows the currently used CWmin. After about 30 seconds of simulation, the collision rate exceeds the upper threshold θ_{up} and the DCWA algorithm increases the CWmin stepwise to 31. The contention window is again increased after 55 seconds. However, whenever the maximum collision rate drops below θ_{lo}, the CWmin is decreased again, as shown after 40 seconds, 75 seconds, and 95 seconds of simulation.

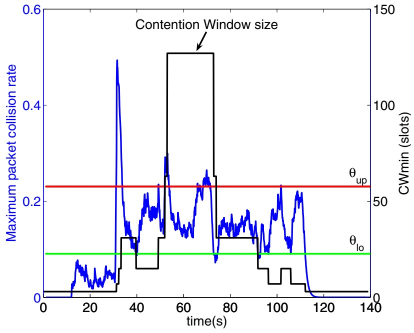


Fig. 2. DCWA algorithm and its parameters

IV. DCWA PARAMETER EVALUATION

The DCWA algorithm can be configured by four parameters θ_{up}, θ_{lo}, τ, and M. One of the main contributions of this work is to identify good values for these parameters in practice. In order to evaluate the impact of a single parameter on the system performance, the concerned parameter is iterated through a value range while keeping the other three parameters fixed.

1) Influence of θ_{up}: The first parameter to be evaluated is θ_{up} which is the upper threshold for the CWCP. It represents a limit for the maximum tolerable average retransmissions per packet over all stations in a Wireless LAN cell. There is a tradeoff between the collision probability and the performance which can be tuned by the choice of a suitable contention window. If the contention window is chosen too small, many station will be competing for the same transmission slot. Choosing the contention window too large, more slots will not be used for any transmission. Hence, to achieve an optimum system performance, a certain level of retransmissions per packet is required. In order to study the influence of the upper threshold, θ_{up} is varied from 0.1 to 0.5. The other DCWA parameters are fixed and set to $\theta_{lo} = 0.05$, $\tau = 1\,s$, and

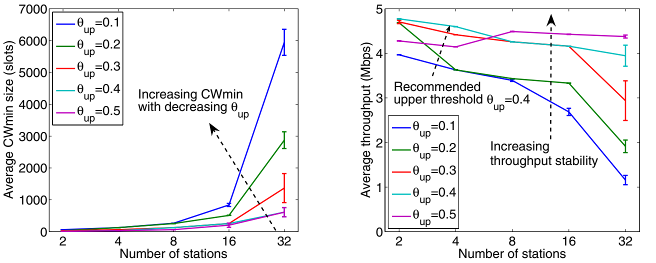
$M = 1$. Other parameter combinations have been simulated as well, but they have shown the same behavior and therefore, only selected parameter combinations are presented.

For the first parameter evaluation, a saturated UDP traffic model is used with a MAC layer packet size of 1257 bits, similar to the ITU-T G.711 voice codec packet size. All stations start with the initial contention windows of $CWmin[AC\_VO] = 3$ and $CWmax[AC\_VO] = 15$ and no bursting is used, meaning that the transmission opportunity limit of the IEEE 802.11e standard is set to one packet. We distinguish between the transient phase and the steady state and evaluate appropriate measures in both phases. During the transient phase, new stations start their transmission and the DCWA adapts the contention window until the number of stations reaches a fixed level. The duration of this phase is influenced by all parameters. The steady state considers the time when all stations have started their transmissions and there are only a few more contention window adaptations until the end of the simulation.

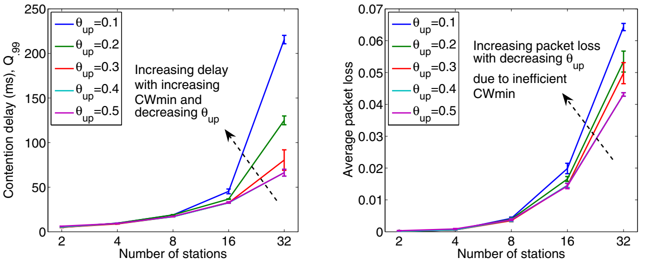
At first, we take a look at the contention window, since the DCWA controls its size throughout the simulation. The development of the average CWmin size for the steady state is shown in Fig. 3(a). An increasing CWmin with an increasing number of stations can be observed, which reflects the higher contention when many stations compete concurrently for medium access. Further, the smaller θ_{up} is set, the higher gets the CWmin. This is obvious since the lower the θ_{up}, the sooner the threshold will be reached, and the contention window will be enlarged. Informally, a contention window adaptation decision made by the DCWA is called stable when the contention window size reaches a steady state and is not changed by the DCWA anymore. The stability depends on both a proper adjustment of the parameter M and τ, and the hysteresis control range $[\theta_{lo}; \theta_{up}]$. Having selected a fixed $\theta_{lo} = 0.5$ the too narrow control ranges $[\theta_{lo}; \theta_{up}] = [0.05; 0.1 - 0.2]$ lead to a high number of contention window adaptations during the steady state phase. In this case, the contention window suffers from oscillation. Wider control ranges enable the DCWA to choose a stable contention window size, i.e. there are only a few more adaptations necessary throughout the simulation.

The choice of an appropriate control range significantly affects the performance in terms of throughput. Fig. 3(b) depicts the impact of the upper control threshold θ_{up} on the average throughput during the steady state. We recognize a decrease in throughput the more stations are active in the cell; this can generally be traced back to a loss of resources due to higher contention. As mentioned above, there is a tradeoff between the packet collision probability and the achievable performance. Up to a value of θ_{up}, the average throughput increases with larger values of θ_{up} throughout all scenarios.

There are two main impacts on the contention delay which is depicted in Fig. 3(c). First, the more stations are active and communicate, the larger is the average medium busy time a particular stations has to wait, and the higher is the contention for medium access. An increasing number of stations in the system result in an increased delay. Second, the higher the contention windows, the larger are the backoff times, and the larger is the contention delay.



(a) Average CWmin size in the steady state (b) Average throughput in the steady state

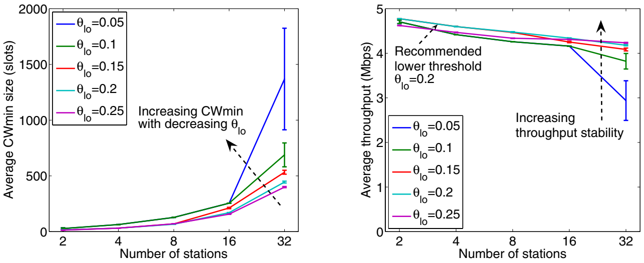


(c) Contention delay in the steady state (d) Average packet loss in the transient phase

Fig. 3. Impact of the DCWA parameter θ_{up} on the Wireless LAN performance

The impact of the DCWA on the packet loss during the transient phase is shown in Fig. 3(d). During this transient phase, packets are dropped due to high contention. An average packet loss of up to 7 % can be observed and even a slightly higher packet loss for $\theta_{up} = 0.1$ which occurs due to CWmin oscillations. During steady state, the packet loss rate is extremely low and in an order of magnitude from 10^{-4} to 10^{-5} on average. This is an indicator for the robustness of the DCWA and can be observed in all results of the simulation series. Since not experiencing an overall throughput gain for values above 0.4, and having the objective to maintain a system that is still sensitive to traffic changes, an upper control threshold of $\theta_{up} = 0.3 - 0.4$ is recommended.

2) Influence of θ_{lo}: After having found a suitable setting for θ_{up}, the performance influences of θ_{lo} is evaluated. θ_{up} is now fixed to 0.3 and θ_{lo} is varied between 0.05 and 0.25. This lower control threshold θ_{lo} represents the minimum empirical collision probability that is tolerable and is responsible for decreasing the contention windows as soon as the measured average collision probability drops underneath θ_{lo}. The lower it is set, the less the DCWA responds to traffic fluctuations. This behavior can be approved by Fig. 4(a). Small values of θ_{lo} result in higher average contention window sizes, while the highest value $\theta_{lo} = 0.25$ effects the smallest average contention window size, but it effects as well the highest number of contention window adaptations throughout the simulation. Here, the contention window oscillates because the hysteresis control range $[\theta_{lo}; \theta_{up}] = [0.25; 0.3]$ is too small, constantly triggering the contention window to be incremented and decremented again. In either case, the average throughput does not reach its achievable optimum which can be seen in Fig. 4(b). The average throughput is maximized for the values $\theta_{lo} \in 0.15; 0.2$ throughout all scenarios except for the 32 stations scenario, which is dominated by $\theta_{lo} = 0.25$. The 99 % quantile of the contention delays are always below 100 ms and are only slightly affected by different values for the θ_{lo}. Again, the average packet loss during the steady state phase is below 10^{-4} and partly even below 10^{-5} for all parameter configurations.



(a) Average CWmin size in the steady state (b) Average throughput in the steady state

Fig. 4. Impact of the DCWA parameter θ_{lo} on the Wireless LAN performance

We conclude that the control range should be at least of size $\delta = 0.3 - 0.2 = 0.1$. Hence, we recommend a control range for the DCWA from $[\theta_{lo}; \theta_{up}] = [0.15 - 0.2; 0.3 - 0.4]$ or expressed in collision probabilities $[\theta_{lo}; \theta_{up}] = [13\,\% - 17\,\%; 23\,\% - 29\,\%]$. The parameters M and τ were evaluated similar to θ_{lo} and θ_{up} and the optimum parameter setting for the DCWA can be found in Table I.

TABLE I

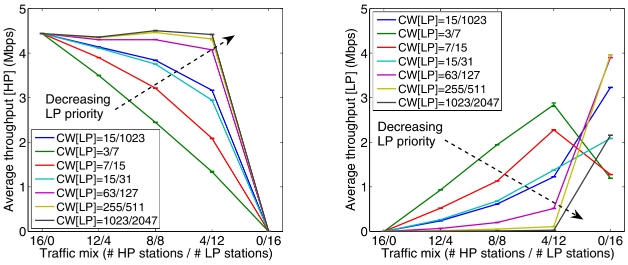
RECOMMENDATION FOR THE DCWA PARAMETER CONFIGURATION

DCWA parameter	**Value range**
θ_{up}	0.3-0.4
θ_{lo}	0.15-0.2
M	0.5-1.0
$\tau(s)$	0.5 s-1.0 s

V. PERFORMANCE OF THE DCWA ALGORITHM WITH TWO SERVICE CLASSES

This section shows the impact of the DCWA algorithm on a best effort service class. We call this class *Low Priority* (LP) class, while the voice traffic class is called *High Priority* (HP) class. For the evaluation, the DCWA is configured with θ_{up}=0.4, θ_{lo}=0.2, M=1, and $\tau(s)$=1 s. The simulation duration of a single simulation run is 100 s, and the first 20 s are considered as transient-phase. All of the following performance figures are generated on the basis of five replications by calculating the 95 % confidence interval. The design of this analysis allows us to observe influences resulting from the scenario size, such as the number of stations, the traffic mix, and the prioritization level.

A main result of the throughput analysis is that the DCWA successfully keeps the throughput stable over all scenario sizes and traffic mixes which is shown in Fig. 5. The figures display the average throughput for HP and LP traffic respectively for a scenario with a total of 16 stations. The x-axis depicts the traffic mix giving the number of HP and LP stations. The traffic mix starts with 100% HP traffic and continuously replaces the HP traffic in steps of 25 % with LP traffic. The y-axis depicts the average throughput as a function of the traffic mix. Up to a certain degree of prioritization, we observe the following expected behavior: the average HP throughput drops as the number of HP stations decreases, whereas the average LP throughput rises with increasing number of LP stations. However, in cases where the CW[LP] is equal or greater than 63/127, the average HP throughput drops only slightly. This is a result of the high prioritization level, leading to an almost total blockage of LP traffic.



(a) High Priority traffic (b) Low Priority traffic

Fig. 5. High Priority and Low Priority throughput of the 16 station scenario

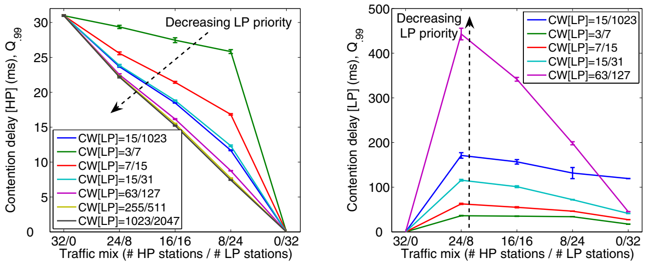
A strong relationship between prioritization and achieved throughput can be observed. The prioritization level directly determines the amount of throughput each Access Category can achieve. The higher the prioritization level, the higher the HP throughput and the lower the LP throughput and vice versa. A rise of the prioritization level results in a linear decrease of LP throughput. In presence of 100 % LP traffic, the behavior is different. In this case, the DCWA is not enlarging the contention windows. From Fig. 5(b) can be seen that the throughput degrades in case the contention window is too small (CW[LP]≤7/15). The choice of a 'broad' contention window of 15/1023 mitigates this problem and achieves a good throughput performance for each scenario. Its prioritization level correponds approximatley to 15/31 and additionally has the ability to better adapt to the given traffic mix.

We can conclude that the DCWA achieves the capacity enhancement for high priority traffic, and that an appropriate choice of the contention window prioritization level controls how the capacity is shared among the traffic categories.

As we have seen, we can realize a capacity enhancement with the DCWA and we can share it among HP and LP traffic with regard to specific prioritization needs. Besides throughput, there are the central performance metrics: packet loss and packet delay. Especially for the HP traffic class which targets to carry voice and video traffic, it is crucial to meet specific QoS requirements.

For the following delay analysis, a contention delay metric is generated by separately collecting all HP and LP packet delays of all stations. Then, the 99 % quantile of these sets of delay values is calculated and denoted $Q_{.99}$. A main result of the delay analysis is that prioritization of HP traffic can be adjusted so that HP delay is not affected by LP traffic.

Fig. 6(a) and Fig. 6(b) depict the contention delays for HP and LP traffic in a scenario with 32 stations.



(a) High Priority traffic (b) Low Priority traffic

Fig. 6. Impact of the traffic mix and the contention window prioritization on the contention delay

We recognize that the higher the prioritization level, the shorter the contention delay for HP traffic and the longer the contention delay for LP traffic. Up to a prioritization level of 5 (CW[LP]=63/127), the HP delays are reduced. The absolute reduction amounts to 20 ms, inferred from Fig. 6(a), data point 8/24. At the same time, LP delays exceed 200 ms. A prioritization beyond this level does not reduce HP delays anymore, but increases LP delays tremendously. The delay curves for prioritization levels of 7 and 9 are not included in Fig. 6(b) since the LP values exceed 2 s and 7 s respectively. The green curves in both figures represent the delays when no contention window prioritization among HP and LP traffic is used, i.e. $CW[HP] = CW[LP] = 3/7$. A quantitative comparison of both shows that the HP delay is at least 5 ms below the LP delay at comparable points. This part of delay prioritization of HP can be traced back to a shorter *Arbitration Interframe Space* (AIFS) of the high priority traffic. However, we recommend to set the contention windows for the low priority traffic class to 63/127 or to the initial settings 15/1023 as recommended by the IEEE 802.11e standard.

We conclude that the contention window prioritization is the main factor in order to guarantee short delays for the HP traffic class by delaying the LP packets by means of larger backoff times. Both the HP and LP delays decrease with a decreasing share of HP traffic. This is a result of the DCWA which enlarges the contention window and with it the backoff times when coping with more HP stations. This behavior can be observed for all scenario sizes.

The packet loss for both HP and LP traffic is very low for all scenario sizes and traffic constellations. This a result of the DCWA, which selects an optimal contention window depending on the current HP traffic. HP packet loss lays in an order of magnitude of 0.01% to 0.001%, LP packet loss is between 0.1% to 0.01%.

VI. CONCLUSION

In this paper, we presented the *Dynamic Contention Window Adaptation* (DCWA) algorithm which keeps the channel contention in Wireless LAN at an efficient level, independent of the current network load. The DCWA algorithm chooses a suitable contention window regarding the channel contention level. An efficient DCWA parameter configuration was derived to optimize the achievable capacity and it was shown that the algorithm effectively increases the wireless resources available for high priority traffic. Up to 100 % more voice connections can be supported compared to the IEEE 802.11e standard while still meeting the Quality of Service requirements.

A key finding was that the contention window is a very powerful means to realize service differentiation. The amount of wireless resources granted to an Access Category and the experienced packet delivery delay are heavily impacted depending on the degree of the contention window prioritization. The DCWA was extended to simultaneously control the contention windows of both high priority and low priority traffic. By maintaining the prioritization level between the two service classes, the QoS requirements for voice traffic can still be met at any time and best effort traffic is granted the remaining wireless resources.

REFERENCES

[1] IEEE Std. 802.11-1997, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification,” 1997, iSO/IEC 8802-11:1997.

[2] IEEE 802.11e, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements,” November 2005, iEEE 802.11e-2005.

[3] L. Romdhani, Q. Ni, and T. Turletti, “AEDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, USA, March 2003. [Online]. Available: citeseer.ist.psu.edu/romdhani03adaptive.html

[4] V. Scarpa, G. Convertino, A. Palmieri, and J. C. D. Martin, “Adaptive Techniques to Guarantee QoS in a IEEE 802.11 Wireless LAN,” in *IEEE 59th Vehicular Technology Conference 2004, VTC 2004-Spring*, Milan, Italy, May 2004, pp. 3014 – 3018.

[5] L. Zhang and S. Zeadally, “A Framework for Efficient Resource Management in IEEE 802.11 WLANs: Research Articles,” *Wirel. Commun. Mob. Comput.*, vol. 4, no. 8, pp. 835–848, 2004.

[6] L. Scalia, I. Tinnirello, J. W. Tantra, and C. H. Foh, “Dynamic MAC Parameters Configuration for Performance Optimization in 802.11e Networks,” in *IEEE Globecom 2006*, San Francisco, CA, USA, November 2006.

[7] A. Nafaa, A. Ksentini, and A. Mehaoua, “SCW: Sliding Contention Window For Efficient Service Differentiation in IEEE 802.11 Networks,” in *IEEE WCNC 2005 The IEEE Wireless Communications and Networking Conference 2005*, New Orleans, LA, USA, March 2005.

[8] S. E. Housseini and H. Alnuweiri, “Adaptive Contention-Window MAC Algorithms for QoS-Enabled Wireless LANs,” in *International Conference on Wireless Networks, Communications, and Mobile Computing*, Maui, HI, USA, June 2005.

[9] L. Gannoune, “A Comparative Study of Dynamic Adaptation Algorithms for Enhanced Service Differentiation in IEEE 802.11 Wireless Ad Hoc Networks,” in *AICT-ICIW ’06: Proceedings of the Advanced Int’l Conference on Telecommunications and Int’l Conference on Internet and Web Applications and Services*, Guadeloupe, French Carribean, February 2006.

[10] R. Pries, D. Staehle, S. Oechsner, M. Menth, and S. Menth, “A New Perspective on the Unfair Channel Access Phenomenon in Wireless LANs,” Universtiy of Würzburg, Würzburg, Germany, Tech. Rep. 439, December 2007.

[11] M. Menth, J. Milbrandt, and J. Junker, “Time-Exponentially Weighted Moving Histograms (TEWMH) for Self-Adaptive Systems,” in *IEEE Globecom*, San Francisco, CA, USA, November 2006.

[12] OPNET Modeler, OPNET University Program: http://www.opnet.com/services/university/.