A Novel MAC Scheme for Solving the QoS Parameter Adjustment Problem in IEEE 802.11e EDCA

Thomas Nilsson[†] and Jahanzeb Farooq[‡]
[†]Dept. of Computing Science, Umeå University, Sweden, nilsson@cs.umu.se
[‡]Planete Project, INRIA, Sophia Antipolis, France, jfarooq@sophia.inria.fr

Abstract

We present a novel MAC scheme for solving the QoS parameter adjustment problem in IEEE 802.11e EDCA. The default values of the QoS parameters in EDCA, Contention Window (CW), arbitration Interframe Space (IFS) etc only yield good performance for few scenarios. We first propose a simple adaptation scheme, called a-EDCA, where the access point adapts the CWs based on the network conditions. The main contribution of this work is the second approach, called i-EDCA, where we make modifications to the backoff phase of EDCA and introduce a random IFS scheme. Simulation results show that both approaches result in stable capacity ratios between priority classes and very high channel utilization, compared to EDCA, when the number of stations is increased. Moreover, i-EDCA yields close to optimal channel utilization for a large number of scenarios without any need of adapting the parameters. In i-EDCA, the capacity for each class is directly proportional to its CW. Results also show that i-EDCA has improved fairness and prevents low priority classes to be starved under higher loads, a problem found in EDCA.

1 Introduction

Task Group E has recently approved the new wireless LAN standard IEEE 802.11e, capable of supporting QoS through service differentiation. The standard defines two access methods, one controlled channel access and one contention based channel access called Enhanced Distributed Channel Access (EDCA). EDCA defines four priority classes or access categories for traffic with different QoS requirements. Each access category (AC) uses a specific set of contention parameters, including the minimum and maximum Contention Window (CW), and the new arbitration IFS (AIFS), to contend for the channel.

Earlier studies have shown that the default values of the contention parameters are only good for scenarios with few high priority ACs and under moderate traffic loads, e.g. see [11]. The access point has the flexibility to adjust the contention parameters but no algorithm for this purpose is provided in the standard [4, Chap. 9.1.3.1].

One problem in EDCA is how to adjust the contention parameters, under varying network conditions, to achieve a certain capacity ratio between the ACs and at the same time achieve high channel utilization. Previous studies have shown that adaptation of multiple contention parameters may not be desirable to achieve the two goals [6]. There is a complex relationship between the contention parameters used by an AC and the relative share of the total capacity it achieves. It would be desirable to know this relationship to effectively adjust the contention parameters for each AC to achieve a certain capacity proportional to the parameter values used. This can be achieved by using CW differentiation [6].

In this work two completely different approaches to achieve high channel utilization and stable capacity ratios among ACs, in EDCA, under varying network conditions are proposed. First, we propose a simple parameter adaptation scheme, called adaptive EDCA (a-EDCA), where the access point is responsible for updating the CWs for each AC based on the congestion level in the network. The main contribution of this work is the second approach, where we make modifications to the backoff phase, including the removal of the the Binary Exponential Backoff (BEB), and the AIFS scheme in the EDCA protocol. The result is an improved EDCA (i-EDCA) with stable capacity ratios and very high and almost constant channel utilization, close to optimal, for various congestion levels without the need of adapting the contention parameters. The a-EDCA also serves as a comparison to show that similar performance can be achieved with the use of a centralized approach.

The rest of the paper is organized as follows. We first review some related work in the remainder of this section. In Section 2 we propose the a-EDCA and discuss its parameters. The new i-EDCA scheme is described in Section 3 with an example scenario of how it works. In Section 4, we evaluate the performance of our two approaches by simula-

978-1-4244-2100-8/08/$25.00

tions. A simple model to predict the throughput share for each AC is presented in Section 5. We conclude this work by summarizing our key findings in Section 6.

1.1 Related Work

In EDCA, packets arriving from higher layers are tagged with eight different user priorities and each priority is mapped to one out of four ACs (implemented by each station). Each AC maintains a local queue and an independent backoff instance parametrized with a specific set of contention parameters. All ACs independently contend for access to the channel and internal collisions may occur, but are solved by allowing the AC with the highest priority to gain access to the channel. The CW parameters CW_{min} and CW_{max} and AIFS are used to differentiate between ACs. Instead of waiting the normal DIFS time, each AC waits a specific AIFS time. Higher priorities have lower values of the CW parameters and AIFS. This leads to a higher fraction of the capacity and lower delays since the channel access frequency is increased. An additional parameter is the transmission opportunity (TXOP) that specifies for how long the channel may be occupied by a station. Depending on this limit, one or several packets may be transmitted when a AC has acquired the channel.

A large number of modifications and improvements of EDCA have been proposed, e.g. see [10, 12, 8]. The problem of adjusting the CW in EDCA is inherited from the original 802.11 DCF and it is well known that the BEB scheme used to adjust the CW is far from optimal [3]. A key problem in optimizing the CWs is to acquire an estimate of the number of stations in the network. By monitoring the number of empty time slots between transmission attempts the number of stations can be estimated and hence the optimal CW can be selected [3]. Several techniques have been proposed to improve the performance of EDCA by allowing the stations to adapt their CWs according to the channel state, e.g see [1]. Adaption of AIFS to achieve stable capacity ratios between ACs has also been studied in [14].

A randomized Arbitration Interframe Space Number (AIFSN) algorithm is presented in [5]. High priority ACs suffer from increasing collision probability when more stations enter the AC. By introducing a randomized AIFS for each AC, with different intervals, there is an additional level of separation between stations and hence the collision probability may be decreased. It is well known that AIFS have this property. However, the main purpose with AIFS in the EDCA standard is to achieve capacity differentiation.

A multi-class model is derived in [6] to adapt the EDCA contention parameters, by the access point, to varying network conditions using optimal parameters. The model only focuses on CW adaptation to provide proportional service differentiation, pre-specified throughput ratios among the ACs and high channel utilization. Furthermore, it may not be desirable to allow adaptation of both the CW and the AIFS, instead only adaptation of the CWs should be preferred [6].

An algorithm to adapt the CW_{min} to varying network conditions is proposed in [13]. Similar to our first approach, the throughput measurement at the access point is used to adapt the CW_{min} to achieve high channel utilization. However, in their work a system not compliant to EDCA is considered.

2 Adaptive EDCA

A parameter adjustment mechanism is presented that dynamically adapts parameters (CW_{min} and CW_{max}) based on network conditions. The access point continuously monitors the network traffic, determines new CW values using a parameter adapting algorithm and broadcasts these values in beacon frames. The stations on receiving the beacon frames then use the updated parameter values to contend for the medium.

The criterion used by the parameter adapting algorithm to update the parameter values is the aggregated throughput. The access point monitors the throughput and at specific intervals based on whether the throughput is decreasing or increasing, the parameters are adapted.

The parameter adapting algorithm works as follows: at any given time, the algorithm remains in one of the two states; either in the state of increasing or decreasing parameter values. The algorithm remains in a particular state as long as throughput keeps increasing. As soon as a drop in throughput is monitored, transition to the other state is made. The algorithm is presented in pseudo-code in Algorithm 1. A threshold value determines the minimum required difference between the current and previous throughput to update the parameter values (See line 2 in the Algo-

Algorithm 1 Adaptive EDCA

```
 1: Set $diff := abs(1 - newThr/oldThr)$
 2: if ($diff > thrshld$) then
 3:     if ($state = inc$) then
 4:         if ($newThr > oldThr$) then
 5:             Set $CW[ACi] := CW[ACi] \times$PF
 6:         else
 7:             Set $CW[ACi] := CW[ACi]/$PF, $state := dec$
 8:         end if
 9:     else
10:         if ($newThr > oldThr$) then
11:             Set $CW[ACi] := CW[ACi]/$PF
12:         else
13:             Set $CW[ACi] := CW[ACi] \times$PF, $state := inc$
14:         end if
15:     end if
16: end if
17: Broadcast updated parameter values
```

rithm 1). For instance if set to 0.1, the parameters are updated only if throughput is increased or decreased by at least 10 percent of the previous throughput.

A Persistence Factor (PF) is applied while updating the parameters (e.g. see line 5 in the Algorithm 1). For instance, the parameter values are doubled if PF is 2 and set to three times if PF is 3, and decreased in similar fashion.

Note that every time the parameters are adapted, CW_{min} and CW_{max} for all four ACs are updated. Hence the ratio between the CW values for the four ACs remains fixed. Also note that the parameters are never decreased beyond the default values specified in the standard, as shown in Table 1.

3 Improved EDCA

The use of a randomized AIFS scheme provides an additional level of separation between stations and has shown to yield lower collision probability [5]. Following a busy channel, only a part of the stations is allowed to start decrementing their backoffs in the first few slots and consequently the collision probability is decreased.

We have adopted a similar random IFS scheme in our i-EDCA, but only for the purpose of increasing the channel utilization and not for achieving differentiation between ACs [5].

It is well known that the BEB scheme adopts the CW in a non optimal way. This is one of the causes behind the need of adapting the contention parameters in EDCA. A collision in CSMA/CA forces the colliding stations to double their CWs. However, this information about congestion is not used by other stations to adjust their CWs. Furthermore, a successful transmission forces the station to reset its CW to the minimal value that is only optimal for limited number of stations. The BEB scheme also works in a discrete way which means that there is little precision in adjusting the CWs according to the contention level in the network. Our approach to adjust the backoff is not based on collisions between stations, but only on the channel idle status. Furthermore, the CWs are fixed to one value and the adjustment is only made to the backoff counter.

In i-EDCA stations select, when the channel becomes idle, a new AIFS value uniformly from the interval $\{1, 2, \ldots, H\}$, we refer to this as the Random IFS (RIFS). All ACs select their RIFS value from the same interval. Thus, RIFS is not used for priority differentiation. Each station in ACi selects a new backoff uniformly from the interval $\{0, 1, \ldots, CW[ACi]-1\}$ following each of its transmissions. When the channel has been idle for RIFS slots, the sender starts to decrement its backoff counter, similar to in EDCA. If the channel becomes busy before RIFS idle slots have elapsed, the sender will sample a random value, k, uniformly from the interval $\{1, 2, \ldots, K\}$, and add this value to its backoff counter. The motivation behind this is that heavy congestion in the network is signaled by decreasing number of idle slots between transmissions [3, 2, 7]. In i-EDCA, this leads to a higher fraction of stations that are unable to sense the channel idle for their sampled RIFS number of slots. These stations will then increase their backoff counters and the congestion level will be reduced. This is a faster method to respond to congestion than in CSMA/CA, where a station must pay the cost of a collision to adjust its CW.

Each AC keeps a fixed CW, CW$[AC0]$, CW$[AC1]$, $\ldots$, CW$[ACM]$ and stations do not double their CWs after unsuccessful attempts (collisions). The i-EDCA algorithm is described in pseudo–code in Algorithm 2. Here BO_{max} (See line 14 in Algorithm 2) sets an upper limit on the size of the backoff counter. Figure 1 shows an example with three stations, S_1, S_2 and S_3, contending for access. Gray upside-down triangles are the selected RIFS times. The tables contain the values of RIFS and BO for each contention phase (1 to 3) and a bolded value means that the BO has been updated. In the first contention phase, the stations select 10, 5 and 1 RIFS slots (See line 5 in Algorithm 2). S_3 waits 1 slot ($t = 1$) and starts to decrement its backoff counter and at $t = 5$ S_2 starts its countdown. At $t = 9$, the backoff counter of S_2 reaches 0 and it starts the transmission and the other stations freeze their backoff counters. S_1 has not started its countdown when the channel becomes busy and therefore adds a random value to its backoff counter (See line 14 in Algorithm 2). S_3 continues to decrement its backoff after two slots ($t = 2$) in the second contention phase. The other two stations also start to decrement their counters after their specific RIFS slots. However, S_1 has added 4 slots to its backoff because it did not start to decrement its backoff in the previous contention phase. In contention phase 3, S_3 does not start to decrement its backoff and will consequently add a random value to its backoff in the subsequent contention phase (not shown here). It should be noted

Algorithm 2 Improved EDCA

```
 1: if Previous transmission on channel has ended then
 2:     if $BO = 0$ then
 3:         $BO$ := UniRnd{0,...,$CW[ACi]$}
 4:     end if
 5:     $RIFS$ := UniRnd{1,...,$H$}                          ▷ Random IFS
 6:     if Channel idle for $RIFS$ slots then
 7:         while channel is idle do
 8:             Set $BO := BO - 1$
 9:             if $BO = 0$ then
10:                 TRANSMITMESSAGE
11:             end if
12:         end while
13:     else                                                ▷ $BO < BO_{max}$
14:         $BO := BO + \text{UniRnd}\{1, \min(K, BO_{max} - BO)\}$
15:     end if
16: end if
17: Repeat from (1)
```

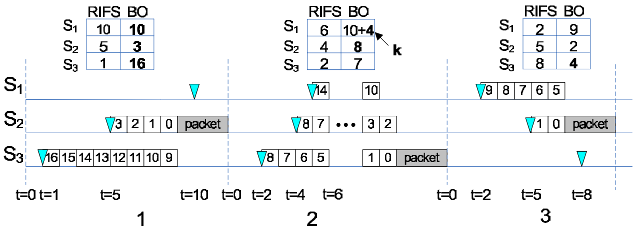

Figure 1: An i-EDCA scenario with three stations.

that i-EDCA can support more than one AC per station, by granting access to the highest priority AC in case of virtual collisions between ACs within the same station.

3.1 Capacity differentiation in i-EDCA

In i-EDCA differentiation between ACs is only based on the use of different CWs. First we associate a weight ϕ_i to each ACi and its weight determines the throughput r_i allocated to ACi. In other words, capacity is allocated to each AC according to its weight. The relative capacity ratio for ACi compared to ACj is

$$r_{i,j} = \frac{r_i}{r_j} = \frac{\phi_j}{\phi_i}. \tag{1}$$

In this work, a higher weight means a lower share of the capacity. To enforce the desired capacity ratios, the CW for each AC is computed as follows

$$CW[ACi] = CW_{\text{min}} \times \frac{\phi_i}{\phi_0}, \tag{2}$$

where ϕ_0 is the weight for the highest priority AC, i.e. AC0. To have $r_{0,1} = \frac{\phi_1}{\phi_0} = 2$, then $\phi_0 = 1$ and $\phi_1 = 2$. The CWs simply become CW[AC1] = CW$_{\text{min}}$ × 2 and CW[AC0] = CW$_{\text{min}}$ × 1.

4 Simulation Results

We assume a single network with $i := 1, 2, \ldots, M$ priority classes or ACs with n_i stations in each AC and one access point. We restrict our work to the case where there is one AC per station although EDCA supports four. Furthermore, we assume a total of four ACs but the validity of this work is not only limited to this number. In EDCA, the four ACs are named, AC_VO, AC_VI, AC_BE and AC_BK, but here we will use the notation AC0, …, AC3.

We have implemented the a-EDCA and i-EDCA protocols in the GloMoSim environment [15]. Our focus is on the MAC layer performance and therefore only the simple Two-ray model is used to model the pathloss, and no fading is assumed. For the physical layer we are assuming the IEEE 802.11a standard. The simulation area is of size 300 x 300 meters with an access point located in the center. The stations are uniformly distributed and there is no mobility. We assume full connectivity in the network, i.e. no hidden terminals. Table 1 shows the parameter values used in the simulations. These are the default values unless otherwise stated.

Table 1: Default simulation parameters for a-EDCA and i-EDCA.

Parameter	a-EDCA	i-EDCA
CW$_{\text{min}}$[AC0]	$(\text{CW}_{\text{min}}+1)/4-1$	CW$_{\text{min}}$ × 1
CW$_{\text{min}}$[AC1]	$(\text{CW}_{\text{min}}+1)/2-1$	CW$_{\text{min}}$ × 2
CW$_{\text{min}}$[AC2]	CW$_{\text{min}}$	CW$_{\text{min}}$ × 4
CW$_{\text{min}}$[AC3]	CW$_{\text{min}}$	CW$_{\text{min}}$ × 8
CW$_{\text{max}}$[AC0]	$(\text{CW}_{\text{min}}+1)/2-1$	-
CW$_{\text{max}}$[AC1]	CW$_{\text{min}}$	-
CW$_{\text{max}}$[AC2]	CW$_{\text{max}}$	-
CW$_{\text{max}}$[AC3]	CW$_{\text{max}}$	-
CW$_{\text{min}}$	31	31
CW$_{\text{max}}$	1023	-
AIFSN[…]	2,2,3,7	RIFS
BO$_{\text{max}}$, K, H	-	1023, 15, 10

For the simulations of the a-EDCA, the access point transmits beacon frames every 500 milliseconds. The parameter values are updated every 8 beacons, i.e., access point monitors the throughput for 8 x 500 milliseconds and updates the parameter values accordingly. We have found that this is the minimum time period required for the access point to have consistent and sufficient data to adapt the parameter values. Only the simulation results for PF=2, yielding the best results, are presented in this work.

We assume that all ACs use the same packet size of 1500 bytes. This assumption is somewhat unrealistic since AC0 (voice) and AC1 (video) most likely use shorter packet sizes. However, in this work we are only interested in how to adjust the contention parameters in an optimal way and achieve a certain capacity allocation among the ACs.

We are assuming a channel bit rate of 6 Mbps in all simulation scenarios. Each AC always has a new packet in queue ready for transmission. Thus, the network operates in saturation condition. The RTS/CTS mechanism is not considered. The simulation results in Figure 2,3,4 and 5 are obtained from one simulation over 300 seconds to show the instantaneous effects on the performance when the network conditions change. The rest of the results are obtained as an average of 20 simulations, each lasting 500 seconds.

4.1 Fairness metric

In i-EDCA, the size of the backoff counter is independent of the number of collisions experienced by a station. This is not the case for CSMA/CA that has problems with fairness since the BEB favors the station that succeeded last and vice versa. We have therefore included fairness in the simulation study.

There exist several fairness metrics for evaluating MAC protocols. We have chosen the dimensionless fairness metric proposed by Jain et al.[9]

$$F_{\text{Jain}_i} = \frac{(\sum_{l=1}^{n_i} x_l)^2}{n_i \sum_{l=1}^{n_i} x_l^2},$$

where x_l is the capacity allocated to station l in ACi and n_i is the number of stations in this AC. This metric is in the range of $[0, 1]$. In this work, the fairness is only evaluated for stations within the same AC. It would be possible to normalize each rate x_i with its corresponding weight ϕ_j and compute the fairness for all stations. However, it is not clear how to set the contention parameters in EDCA to achieve a certain capacity ratio between ACs.

4.2 Simulation scenario

To test the adaptability of the two approaches to cope with varying network conditions, a scenario with increasing and decreasing number of stations during the simulation is considered. In this scenario, the network starts with one station in each AC and the number is doubled every 20s until the network has a total of 64 stations (16 in each AC). The network is then left unchanged for 70s and then the number in each class is divided in half every 20s until only one station remains in each AC. The simulation is run for 300 seconds.

4.3 Results for adaptive EDCA

Figure 2 shows the throughput results for the scenario described in Section 4.2 for EDCA and a-EDCA. The CW$_{\text{min}}$ values for the highest priority AC (AC0) are drawn at the second y-axis in order to examine how parameter values are adapted. Since the ratios between the CW$_{\text{min}}$ and CW$_{\text{max}}$ values for the four ACs are maintained, displaying only the CW$_{\text{min}}$[AC0] value is sufficient to determine the rest of the values.

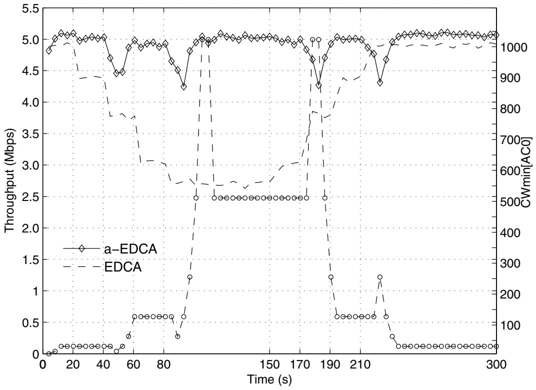

Figure 2: The aggregated throughput for EDCA and a-EDCA.

The access point successfully enhances the throughput by adapting the CW values based on the throughput achieved, see Figure 2. Initially the CWs are increased step by step until no further improvements in throughput are achieved, resulting in a quick adapting to the optimal CWs. Once set to the optimal value, it stays there as long as no significant change in throughput is monitored.

As soon as the access point monitors a drop in throughput as new stations are added, it adapts the CW to higher values optimal for that particular number of stations, resulting in the CW set to the highest value when the number of stations reaches to 64 and then gradually set back to the previous values as stations are removed. However, the adapted throughput suffers from fluctuations in form of dips at the points stations are added or removed. Furthermore, as discussed in Section 2, while experiencing a drop during a dip period, the access point does not make transition to the other state until it meets the threshold. However, once met it quickly recovers from the situation by adapting new CW values for the current number of stations. The dips experienced when stations are added are due to the fact that the CW values in effect that time are too small for the increased number of stations. A drop in throughput therefore occurs as a result of collisions. Closely examining a dip, for example in the 80-100s interval, further reveals a weakness of the algorithm. Here while the first drop occurred due to the reason discussed above, the second consecutive drop occurred primarily because the access point wrongly transited to the decreasing state (it was earlier in the increasing state), resulting in further drop in the throughput. This weakness is further observed during the intervals 170-190s and 210-230s. However as before the access point quickly controlled the situation by entering into the correct state once it realized further drop in throughput.

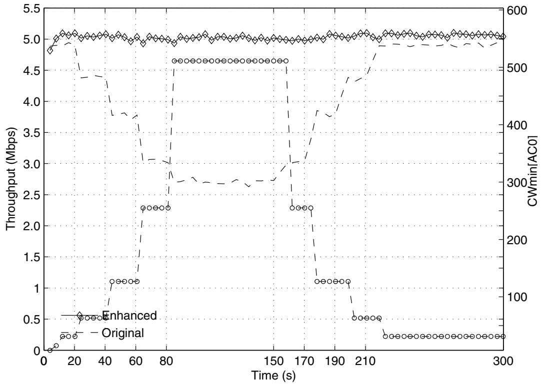

Figure 3: The aggregated throughput for EDCA and the improved a-EDCA.

A small addition is made to the algorithm to overcome the above discussed weakness. Now in addition to throughput, the number of stations is also chosen as a criterion to

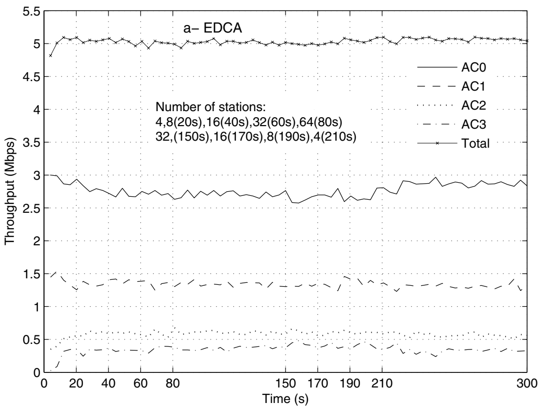

Figure 4: The throughput results for each AC for the improved a-EDCA.

update the parameters. Hence as soon as a change in the number of stations is monitored, the access point automatically enters into the increasing state if the number of stations increases, and vice versa, thus avoiding any possibilities of wrong transitions. Thus the addition helps in adapting the CW values even before any drop in the throughput is experienced as the stations are added or removed.

Figure 3 illustrates the results for the updated algorithm. The new addition improves the algorithm performance. The degradations seen previously are completely overcome now, resulting in very consistent throughput throughout the simulation. The effectiveness of the algorithm is clear, it maintains the throughput by adapting the optimal CW values for varying number of stations, while the throughput for EDCA keeps dropping. Figure 4 shows the same results with the throughput shares achieved by each AC for EDCA and a-EDCA. As it is seen the access point successfully maintains the relative throughput for the individual ACs, which is completely different in the original EDCA where the lower priority AC are easily starved, as a number of studies have shown [11].

Note that since the algorithm is entirely based on throughput monitoring, its performance is not supposed to degrade in the scenarios with variable packet sizes or traffic rates. It will eventually adapt to the optimal parameter values based on the throughput performance provided that the network conditions do not change too frequently.

4.4 Results for improved EDCA

Figure 5 shows the throughput results for i-EDCA for the scenario described in Section 4.2. There is a small decrease in aggregated throughput when doubling the number of stations. The throughput ratios between the ACs remain the same during the simulation. The result in this figure should be compared to Figure 4, showing the case when the access point continuously adapts the CWs. The results for a-EDCA and i-EDCA are almost identical even though the parameters in i-EDCA are not adapted. This clearly demonstrates the benefits of i-EDCA compared to a-EDCA. In the next simulation scenario, we have compared the throughput results for i-EDCA with a fixed CW scheme for increasing number of stations in each AC. In the fixed CW scheme the CWs are kept constant for each AC. Here, we are using the optimal CW sizes that maximizes the channel utilization according to (41) in the analytical model proposed by [6]. This model is based on p–persistent CSMA and can be used to compute the CWs that yield specific capacity ratios between the ACs and maximizes the channel utilization. The p–persistent version of CSMA has been shown to closely approximate IEEE 802.11 DCF [3]. The throughput results using the optimal CWs represent an upper bound for CSMA/CA protocols. In Figure 6 the throughput for each AC is shown for an increasing number of stations. The performance of i-EDCA is compared to a fixed CW scheme using optimal CWs for each AC. Here the CWs are computed for both schemes according to $r_{0,1} = 2$, $r_{0,2} = 4$, $r_{0,3} = 8$.

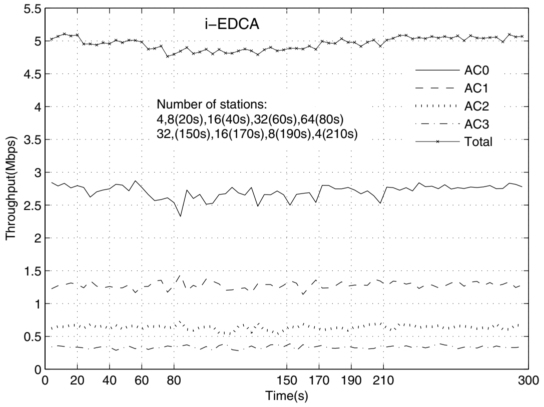

Figure 5: The throughput results for each AC for the i-EDCA.

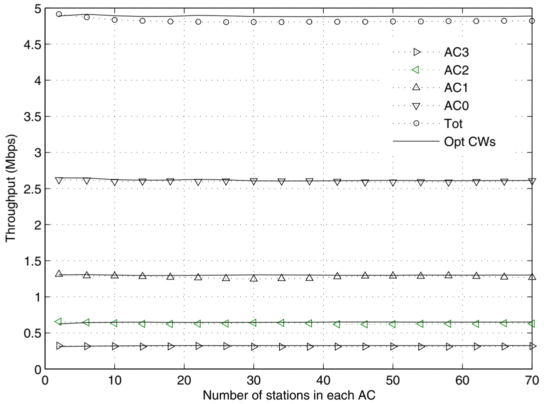

Figure 6: The throughput results for each AC for i-EDCA and the fixed CW scheme using optimal CWs.

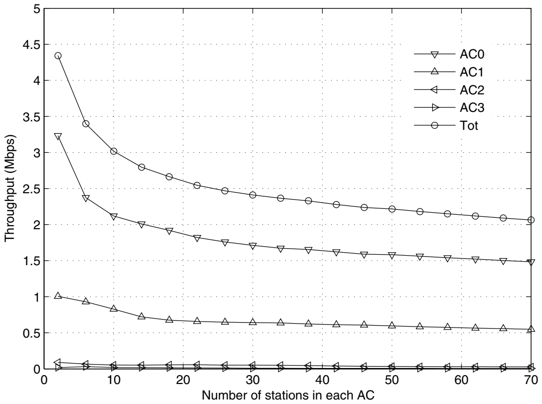

Figure 7: The throughput results for each AC for EDCA when using default values of the contention parameters.

The results for i-EDCA are extremely close to that of the optimal (there are eight curves in Figure 6). It is clear that the throughput ratios are stable when increasing the number of stations in each AC. The aggregated throughput is very high and appears to be almost independent of the number of stations. Clearly there is little need to adjust the parameters in i-EDCA. It should be noted that the optimal performance of i-EDCA might be higher from that of p–persistent CSMA assumed in [6].Using optimal CWs requires the presence of an access point, as in a-EDCA, whereas i-EDCA is fully decentralized and still yields similar performance.

The same scenario, for EDCA, is shown in Figure 7. The throughput for each AC is rapidly decreasing when increasing the number of stations. AC2 and AC3 experience starvation very quickly and the throughput ratios between the ACs do not remain constant. There is no starvation for the low priority ACs in i-EDCA, see Figure 6. In the next scenario we are studying the fairness properties of i-EDCA. A network with 8 stations in each AC, a total of 32 stations, is considered. In Figure 8 and 9 the Jain fairness for each AC is plotted as a function of the throughput sampling period, starting from 1s to 50s. This way of plotting the Jain index gives temporal information about long and short term fairness. For a shorter time scale none of the systems show very good fairness but converges to better fairness over time.

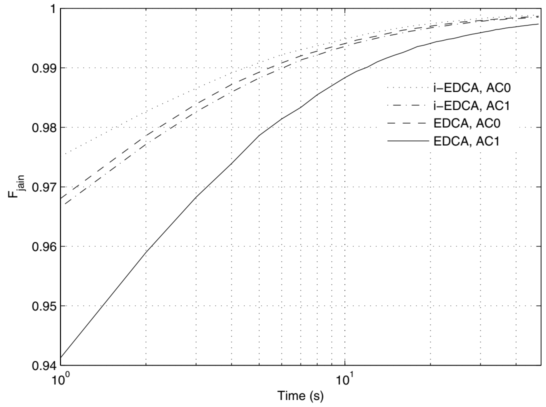

Figure 8: The Jain fairness index, F_{jain}, as a function of the throughput sampling period.

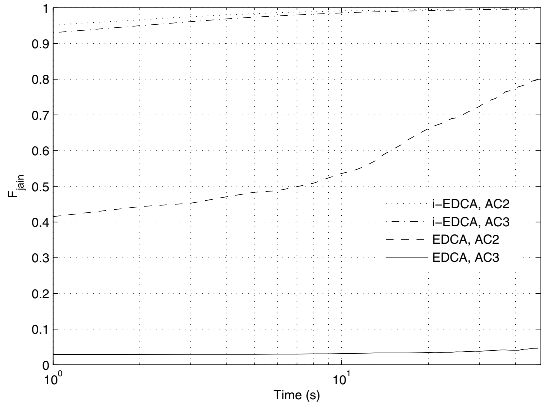

Figure 9: The Jain fairness index, F_{jain}, as a function of the throughput sampling period.

There is little difference in fairness for AC0 and AC1 between EDCA and i-EDCA as shown in Figure 8. In EDCA, the AC0 and AC1 stations are only allowed to double their CWs once, AC0 from 7 to 15 and AC1 from 15 to 31. This results in reasonably good fairness but at the expense of high collision probability when the number of stations increases. For AC2 and AC3 the difference is much greater, where i-EDCA shows very good fairness compared to EDCA as seen in Figure 9. One reason for the poor fairness in the EDCA is that the range of the CW for AC2 and AC3 is 31 to 1023. In the following scenario we look at how the parameter K affects the performance of i-EDCA. Only one AC is considered. Figure 10 shows the aggregated throughput when the number of stations increases. Increasing the value of K results in higher throughput. These results suggest that K should be reasonably large. However, from Figure 11 that shows the jain fairness index, F_{jain}, for the same scenario, it is clear that larger values of K give a fairness problem. The choice of K results in a trade–off between throughput and fairness. In this scenario the sampling time for the throughput is 5s and that gives a fairness measure on a relatively short time scale. We have also studied how the performance is affected when changing the parameter H. These results (not presented here) suggest that $H \approx 10$ yields good performance for most scenarios.

In the final study we have performed simulations to study the throughput ratios between stations in each AC. Table 2 shows the average throughput ratio for a station in each AC relative to a station in AC0. In this study there are 9 scenarios with different number of stations in each AC. Three different weight allocations among the ACs are simulated. From the table it is clear that the relative throughput for a

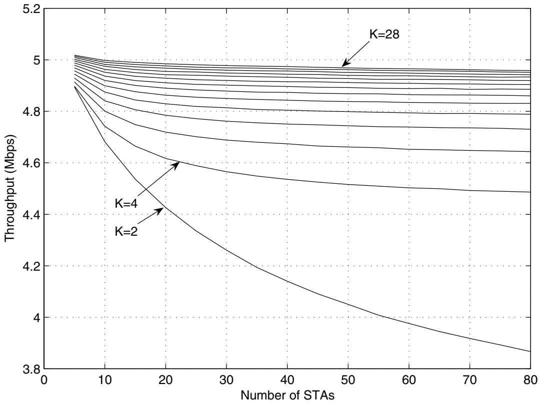

Figure 10: The aggregated throughput as a function of the number of stations.

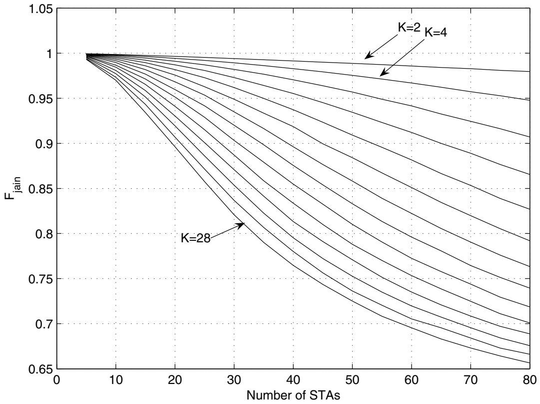

Figure 11: The Jain fairness index, F_{jain}, as a function of the number of stations.

station is inversely proportional to its weight. The values $r_{i,0}$ should be compared to that of ϕ_0/ϕ_i. The aggregated throughput remains almost the same independent of what weights are used by the ACs and the number of stations per AC.

Table 2: The average throughput ratio for a single station in each AC compared to a station in AC0.

N^1	Φ^1	$r_{1,0}$	$r_{2,0}$	$r_{3,0}$	Tot
(5,5,5,5)	(1,2,4,8)	0.504	0.253	0.127	4.75
	(1,4,8,16)	0.253	0.127	0.0634	4.85
	(1,3,6,9)	0.336	0.169	0.113	4.83
(5,10,12,8)	(1,2,4,8)	0.505	0.254	0.127	4.75
	(1,4,8,16)	0.253	0.127	0.0636	4.80
	(1,3,6,9)	0.337	0.169	0.113	4.78
(10,20,30,40)	(1,2,4,8)	0.504	0.253	0.127	4.70
	(1,4,8,16)	0.254	0.127	0.0637	4.72
	(1,3,6,9)	0.338	0.169	0.113	4.70

[1] For ease of notation we define $N = (n_0, n_1, n_2, n_3)$ and $\Phi = (\phi_0, \phi_1, \phi_2, \phi_3)$.

5 Model to predict throughput shares

The simulation results indicate that the capacity remains almost constant independent of the number of stations and weights. Furthermore, the relative capacity for each AC is inversely proportional to its weight. These results can be used to compute the approximate throughput for all stations in the network, if assuming an ideal channel. This can be used by the access point for QoS guarantees and admission control purposes.

The throughput for one station in AC0 is ϕ_i/ϕ_0 higher than a station in ACi, and if we consider the number of stations in each AC, we can easily create a linear equation system as follows

$$\begin{aligned}
(\phi_0/n_0)\,r_0 - (\phi_1/n_1)\,r_1 &= 0\\
(\phi_0/n_0)\,r_0 - (\phi_2/n_2)\,r_2 &= 0\\
&\vdots\\
(\phi_0/n_0)\,r_0 - (\phi_{\rm M}/n_{\rm M})\,r_{\rm M} &= 0\\
\sum_{i=0}^{M} r_i &= C.
\end{aligned}$$

This linear system is easily solved since it is non-singular with a unique solution. The capacity C is the estimate of the maximum achievable throughput. This model is however simplistic since it implicitly assumes that all stations use the same packet size. The model can be extended to incorporate different packets sizes but then becomes slightly more complicated.

The access point has the ability to monitor the throughput for each AC but it would be difficult to predict how these throughput shares would change if new stations from specific ACs enter the network. By solving this linear system the access point can predict such a situation and possibly

Table 3: The predicted throughput for each AC compared to the simulated throughput.

N^1	Φ^1	i	$r_i{}^{\rm s}$	r_i	Tot (Mbps)
(2,2,3,3)	(1,2,4,8)	0	2.40	2.30	4.94
		1	1.20	1.15	
		2	0.895	0.864	
		3	0.450	0.432	
(5,10,12,8)	(1,2,4,8)	0	1.68	1.70	4.75
		1	1.70	1.70	
		2	1.02	1.02	
		3	0.342	0.339	
	(1,3,6,9)	0	2.12	2.12	4.78
		1	1.43	1.41	
		2	0.854	0.847	
		3	0.382	0.376	
(10,20,24,16)	(1,2,4,8)	0	1.67	1.70	4.70
		1	1.68	1.70	
		2	1.01	1.02	
		3	0.339	0.339	

[1] See table 2. [s] Simulated values of r_i.

deny new stations from entering the network based on earlier QoS guarantees. Table 3 shows the predicted throughput for each AC, solved according to the linear equation, for each AC compared to the simulated throughput. In these results C is set to 4.75 Mbps. The predictions of the throughput shares are of-course best when C is closer to 4.75. The table shows three configurations with different number of stations in each AC and in one case with two different sets of weights. Generally, the predicted throughput shares matches, with reasonably good accuracy, the simulated shares indicating that this approach may be feasible.

6 Conclusions

We have proposed two different approaches to solve the parameter adjustment problem in IEEE 802.11e EDCA. In the simpler approach, the access point is responsible to adapt the CWs according to the congestion level in the network. Simulation results show that this approach is feasible in a network to achieve high channel utilization and stable capacity ratios between ACs. However, this algorithm is centralized making it a less attractive solution. To get around the need of adjusting the contention parameters, we make modifications to the backoff phase of EDCA, including the removal of the binary exponential backoff, and introduce a random IFS scheme. We call this new improved fully decentralized protocol improved EDCA (i-EDCA). Simulation results show that i-EDCA gives close to optimal performance, compared to a p–persistent CSMA, for a large number of scenarios without any need of adapting its parameters. Deterministic capacity ratios between ACs can easily be achieved by setting the CWs for each ACi according to a weight ϕ_i. Our results show that the capacity ratio for each AC is inversely proportional to its weight. We also present a simple model to predict the approximate throughput for each AC. The results also show that i-EDCA has better fairness properties than EDCA because the size of the CW or backoff is independent of the number of collisions experienced by a station. Furthermore, i-EDCA prevents low priority ACs from starving under higher loads that can easily happen in EDCA.

We are currently working on a Markov model to further study the performance of the i-EDCA protocol. Future work also include studies of how TCP can benefit from a more fair MAC protocol like i-EDCA and how sensitive this protocol is when hidden terminals are present. The compatibility issue is also of interest in scenarios when there exist both EDCA and i-EDCA stations in the same network.

References

[1] H. Artail, H. Safa, J. Naoum-Sawaya, B. Ghaddar, and S. Khawam. A simple recursive scheme for adjusting the contention window size in IEEE 802.11e wireless ad hoc networks. *Comput. Commun.*, 29(18):3789–3803, 2006.

[2] L. Bononi, M. Conti, and E. Gregori. Runtime optimization of IEEE 802.11 wireless LANs performance. *IEEE Trans. Parallel Distrib. Syst.*, 15(1):66–80, 2004.

[3] F. Calì, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transaction on Networking*, 8(6):785–799, 2000.

[4] T. Cole, editor. *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*. IEEE, New York, NY, USA, 2005.

[5] S. Gaur, C. Tavares, and T. Cooklev. Improved performance of CSMA/CA WLAN using a random inter-frame spacing algorithm. In *IWCMC '06*, pages 407–412, New York, NY, USA, 2006. ACM Press.

[6] Y. Ge, J. C. Hou, and S. Choi. An analytic study of tuning systems parameters in IEEE 802.11e enhanced distributed channel access. *Comput. Networks*, 51(8):1955–1980, 2007.

[7] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. *SIGCOMM Comput. Commun. Rev.*, 35(4):121–132, 2005.

[8] A. Iera, G. Ruggeri, and D. Tripodi. Providing throughput guarantees in 802.11e WLAN through a dynamic priority assignment mechanism. *Wirel. Pers. Commun.*, 34(1-2):109–125, 2005.

[9] R. K. J. Jain, D.-M. W. Chiu, and W. R. Hawe. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. *ArXiv Computer Science e-prints*, Sept. 1998.

[10] T. H. Kim, L. Marwitz, and D. K. Kim. Dynamic offset contention window (DOCW) algorithm for wireless MAC in 802.11e based wireless home networks. *Lecture Notes in Computer Science*, 2524:162–172, 2003.

[11] S. Kuppa and R. Prakash. Service differentiation mechanisms for IEEE 802.11 based wireless networks. In *IEEE Wireless Communications and Networking Conference*, volume 2, pages 796–801, March 2004.

[12] L. Romdhani, Q. Ni, , and T. Turletti. Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless ad hoc networks. In *IEEE WCNC'03*, volume 2, pages 1373–1378, March 2003.

[13] V. A. Siris and M. Kavouridou. Achieving service differentiation and high utilization in IEEE 802.11. In *PWC*, pages 128–137, 2003.

[14] Y. Tanigawa, J.-O. Kim, H. Tode, and K. Murakami. Proportional control and deterministic protection of QoS in IEEE 802.11e wireless LAN. In *IWCMC '06*, pages 1147–1152, New York, NY, USA, 2006. ACM Press.

[15] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: A library for the parallel simulation of large scale wireless networks. In *the 12th Workshop on Parallel and Distributed Simulation*, pages 154–161, 1998.