# Achieving Service Differentiation and High Utilization in IEEE 802.11*

Vasilios A. Siris** and Matina Kavouridou

Technical Report 322 - June 2003
Institute of Computer Science (ICS)
Foundation for Research and Technology - Hellas (FORTH)
P.O. Box 1385, GR 711 10 Heraklion, Crete, Greece
{vsiris,mkavouri}@ics.forth.gr

**Abstract.** Service differentiation in wireless LANs is a growing demand, due to their increasing use for applications with different requirements and the scarcity of wireless channel resources, and is becoming particularly important with the emergence of wireless LAN *hotspots*. The contribution of this paper is twofold: First, we investigate how various parameters of the IEEE 802.11 MAC protocol affect service differentiation, in terms of both throughput and delay. Second, we propose a new approach for providing service differentiation, while achieving high wireless network utilization. Simulation investigations demonstrate that our approach can effectively adapt to varying network conditions, hence can achieve high overall network utilization. The approach can be implemented at the access point of a wireless LAN, while the wireless stations need only to support the emerging IEEE 802.11e standard.

*Keywords:* weighted fairness, throughput monitoring, wireless LANs, Wi-Fi

## 1 Introduction

The wireless LAN (WLAN) area is a field of wide development and great activity over the last few years. The two main WLAN standards are the ETSI High Performance European Radio (HIPERLAN) and the IEEE 802.11 WLAN, with the latter appearing to dominate the market. These standards focus mainly on the Physical and Medium Access Control (MAC) layers. In IEEE 802.11, the MAC layer is responsible for controlling medium access, hence for sharing wireless channel resources. This is achieved using the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) algorithm, which is the primary media access mechanism of IEEE 802.11. As currently defined, the CSMA/CA algorithm lacks support for service differentiation.

* The first part of this work was supported by British Telecommunications (BT), UK.
** Corresponding author: V.A.Siris, vsiris@ics.forth.gr, tel.: +30 2810 391726, fax: +30 2810 391601. Also with the Department of Computer Science, University of Crete

The capacity of wireless networks is much smaller than that of fixed networks. Moreover, unlike fixed networks, there is a limited ability for increasing the wireless channel capacity. At the same time, there is an increasing use of wireless networks for multimedia and delay sensitive applications, and an increasing deployment of wireless LAN hotspots. For all these reasons, it is becoming increasingly important to support service differentiation in wireless LANs.

The contribution of this paper is twofold. First, we investigate mechanisms for supporting service differentiation in IEEE 802.11 networks, both in terms of throughput and in terms of delay. Moreover, we investigate the weighted fairness of the various schemes, by presenting the results as the ratio of throughput achieved for the corresponding ratio of values of the parameter used to provide differentiation. Second, we propose and investigate a new approach for supporting service differentiation, while at the same time achieving high wireless network utilization. The approach involves dynamically adjusting parameters of the IEEE 802.11 MAC protocol, to track varying conditions of the network load. The approach can be implemented solely at the access point of an 802.11 network, without requiring changes to the wireless stations; the latter need only to support the emerging IEEE 802.11e standard.

This rest of the paper is organized as follows. In Section 1.1 we briefly describe the CSMA/CA protocol, including the extensions being developed for IEEE 802.11e. In Section 1.2, we review related work on supporting service differentiation in wireless LANs. In Section 2 we present and discuss results from our investigations on how various parameters of IEEE 802.11 affect service differentiation, in terms of both throughput and delay. In Section 3 we present and investigate a new approach for achieving service differentiation while attaining high network utilization. Finally, in Section 4 we present some concluding remarks, identifying areas for further investigation.

### 1.1 IEEE 802.11

The IEEE 802.11 standard [1] covers the MAC (Medium Access Control) sublayer and the physical layer of the OSI model. The standard supports two medium access control modes: *Distributed Coordination Function* (DCF) and *Point Coordination Function* (PCF). In the first, which is the primary mode and best suited for traffic without strict delay requirements, wireless stations have to contend for use of the wireless channel at every frame transmission. In the second mode, which is optionally supported and best suited for traffic with strict delay requirements, wireless medium allocation is controlled by the access point, which poles wireless stations to use the channel.

**Distributed Coordination Function DCF** . The basic scheme for DCF is *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). A collision can be caused by two or more stations trying to transmit a frame at the same time. After each frame transmission, the sender waits for an acknowledgment (ACK) from the receiver. If no ACK is received, a collision must have occurred and the frame is retransmitted.

Frames can have different priorities by adjusting the time interval, called *inter-frame spacing* (IFS), the channel must be sensed idle prior to their transmission. In DCF, two different IFSs are defined: Short IFS (*SIFS*) and DCF IFS (*DIFS*), where *SIFS* is smaller than *DIFS*. Hence, an ACK, for which *SIFS* is used, has a higher probability of being transmitted before a new data frame, for which *DIFS* is used.

The objective of the *collision avoidance* part of CSMA/CA is to avoid simultaneous frame transmissions right after the channel is sensed idle, since simultaneous transmissions from two or more stations would result in a collision. Such is achieved if a station, prior to transmitting a frame, waits for the channel to be idle for some random backoff interval. This backoff interval is a multiple of slot times, and is selected randomly from the interval $[0, CW - 1] \cdot \text{slot\_time}$, where $CW$ is the contention window, whose initial value is $CWmin$. After each unsuccessful transmission, detected by the absence of an ACK, the contention window is doubled, until it reaches a maximum value $CWmax$.

In addition to *physical carrier sensing*, IEEE 802.11 supports *virtual carrier sensing*. The latter is achieved using time information in the data frames that indicate the duration, called Network Allocation Vector (NAV), that the source will occupy the channel.

**802.11e** . An important activity of IEEE is its work towards 802.11e, an extension of 802.11 aiming to improve its medium access mechanism, and to add support for service differentiation [2]. A new access method called *Hybrid Coordination Function* (HCF) is introduced, which is a queue-based service differentiation scheme that uses both DCF and PCF enhancements.

Enhanced DCF (EDCF) is the contention-based HCF channel access. Although all the details have not yet been finalized, EDCF supports different classes, with different values for *CWmin*, *CWmax*, *DIFS* (now called Arbitration IFS - *AIFS*), and the persistence factor *PF*, which determines the increase of the contention window after a collision. Smaller values of *CWmin*, *CWmax*, *AIFS*, or *PF* correspond to a higher priority. Up to eight priority traffic classes are supported. Each station can have many flows, which can belong to different classes.

### 1.2 Related work

Next we review some representative work that is related to the work presented in this paper; this is not an exhaustive survey of the area.

The work in [3] investigates, through analysis and simulation, the use of different backoff increase rates and *DIFS* intervals for providing service differentiation. Simulation experiments show that such schemes work well for UDP traffic, but not so well for TCP traffic. The work of [4] investigates how the above mechanisms, in addition to *CWmin* differentiation, can be adjusted to support per flow differentiation, rather than per station differentiation, considering in particular TCP flows. In both works, differentiation is in terms of throughput, and the wireless channel utilization is not considered.

Another research direction involves the development of new distributed access control algorithms for supporting service differentiation [5–9]. The algorithm proposed in [6] tries to distribute bandwidth in a fair manner to wireless stations, while supporting service differentiation through weights. In particular, a station's backoff timer is initially determined by the weight factor and the frame size. Upon collision, the backoff interval follows the exponential backoff procedure of 802.11. The work in [7] presents an alternative approach for supporting service differentiation based on weights, where each wireless station adjusts its contention window based on a fairness index, which measures the degree of fairness of the station's throughput, relative to some reference value. In earlier work, the MACAW protocol [5] uses a multiplicative increase/additive decrease algorithm for adjusting the backoff timer: the timer is doubled for each frame loss, and is decreased by one upon each successful transmission.

The work in [8] deals with fairness and wireless channel utilization, and proposes a scheme for selecting the backoff interval for different traffic classes, based on the classes' weights and the estimated number of stations contending for the wireless channel. The work in [9] introduces virtual MAC and virtual source algorithms that monitor the radio channel, and passively determine whether the channel can support new service requests in terms of delay and loss. The schemes described in the last two paragraphs all require changes to the MAC layer running at the wireless stations.

# 2 Service differentiation mechanisms

In this section we investigate how various parameters of IEEE 802.11, and 802.11e in particular, affect service differentiation in terms of both average throughput and delay. The parameters we consider are the following:

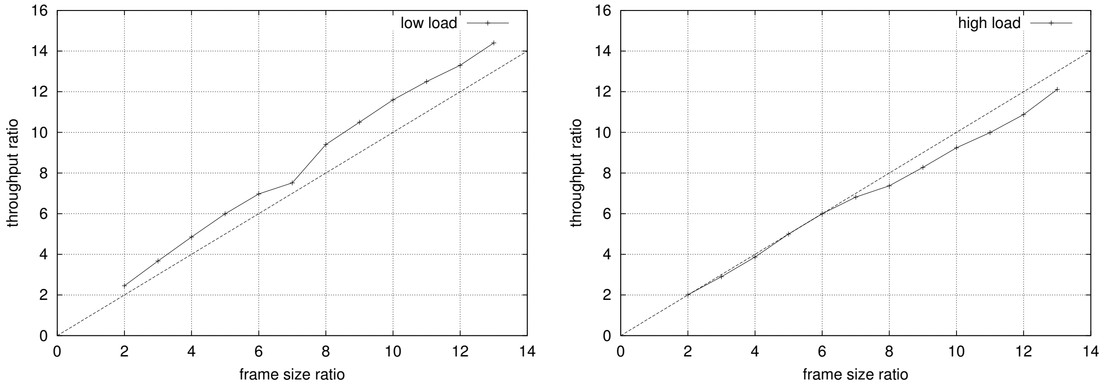
- Maximum frame size
- *DIFS*: interval the channel must be sensed idle prior to frame transmission
- *CWmin*: minimum value of the contention window

Other parameters that can affect service differentiation include *CWmax* (the maximum value for the contention window), and the persistence factor $PF$. Both these parameters will affect service differentiation in the presence of collisions, which however should be avoided since they decrease the channel utilization. Indeed, different *CWmax* values would affect differentiation only in situations of a high percentage of collisions. Differentiation based on altering the persistence factor $PF$, which has the default value of 2 in 802.11, after an unsuccessful transmission has been investigated in [3], where it was shown that large values of the backoff increase rate can lead to unstable behaviour.

Our experiments were performed using the ns-2 network simulator [10]. For the experiments investigating *DIFS* and *CWmin* differentiation, we used the EDCF modules developed by Atheros Communications. The procedure for dynamically adjusting the *CWmin* values was implemented on top of these modules. Except for the first experiment regarding service differentiation based on the maximum frame size, for which the channel capacity was 2 Mbps, the other experiments were for channel capacity 11 Mbps.

## 2.1 Differentiation based on maximum frame size

We first consider varying the maximum frame size that each station is allowed to transmit. The traffic used was constant bit rate (CBR) sources over UDP connections. The wireless LAN considered had capacity 2 Mbps, and carried 4




(a) Low load (= 1.5 Mbps) (b) High load (= 2 Mbps)

**Fig. 1.** Service differentiation with varying maximum frame size.

sources with an aggregate traffic of 1.5 Mbps. Fig. 1(a) shows that the ratio of throughput is approximately proportional to the ratio of frame sizes; each point in the graph corresponds to an experiment where the frame payload size for 2 of the sources was 300 bytes and for the other 2 a multiple of this value. Indeed, it appears that the source with a larger frame size receives proportionally higher throughput compared to a source with a smaller frame size.

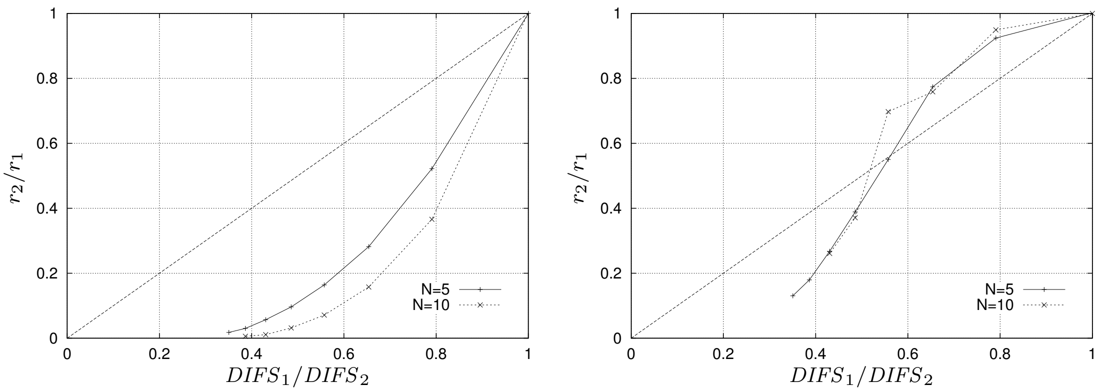
The latter is not the case when the load is increased, as shown in Fig. 1(b), which was for 10 sources with an aggregate rate of 2 Mbps. Indeed, this figure shows that sources with large frame sizes receive somewhat less capacity than that suggested by the ratio of frame sizes. We conjecture that this is due to the fact that larger frames have a higher collision probability.

Finally, we note that varying the frame size cannot achieve delay differentiation. To achieve delay differentiation, we need to use one of the other two parameters, the DCF inter-frame spacing interval *DIFS* or the minimum contention window *CWmin*, which we investigate next.

### 2.2 Differentiation based on *DIFS*

Next we investigate the service differentiation that is achieved with different values of *DIFS*, or *AIFS* as the inter-frame spacing interval is called in 802.11e. Hence, we can distinguish various service classes by assigning to them different *DIFS* values, with a smaller *DIFS* assigned to the class with higher priority.

Our experiments involved both UDP and TCP traffic. The default value of *DIFS* is 34 $\mu$s; this value corresponds to the class with highest priority. The lower




(a) CBR-over-UDP (b) TCP

**Fig. 2.** Throughput ratio $r_2/r_1$ as a function *DIFS* ratio $DIFS_1/DIFS_2$, for UDP and TCP traffic, and $N = 5$ and $N = 10$ sources.

priority classes have a *DIFS* value larger than the default by some number of time slots, with each slot time equal to 9 $\mu$s.

Fig. 2(a) shows the differentiation in terms of throughput, for 5 and 10 CBR sources over UDP connections with aggregate rate 20 Mbps and 40 Mbps, respectively, and payload size 500 bytes. Observe that the dependence of the throughput is far from being inversely proportional to the value of *DIFS*. Fig. 2(b) shows that throughput differentiation in the case of TCP traffic is quite different than that for UDP traffic; indeed, observe that for small values of *DIFS*, the effect of increasing *DIFS* is small. Also observe in Fig. 2(a) that for UDP traffic the number of sources affects the dependence of the throughput ratio on the *DIFS* ratio.

Fig. 3(a) shows how the average delay for voice traffic depends on the number of slots added to the default value of *DIFS*. The traffic consisted of 2 UDP connections carrying voice traffic, and 4 CBR sources with an aggregate bandwidth of 30 Mbps. The modelled voice traffic was assumed to be G.729 encoded: a 60 byte packet is generated every 20 ms during the talk spurt period, and the length of talk spurt periods is exponentially distributed with mean 352 ms, whereas the length of silence (inactive) periods is exponentially distributed with mean 650 ms. Fig. 3(a) shows that, as expected, the average delay increases with the number of slots; indeed, the rate of increase is larger for a larger number of slots added to *DIFS*. Also shown in the figure is the 90% confidence interval, estimated from 10 independent runs of the experiment.

Fig. 3(b) shows the average delay of TCP traffic, in the case of 2 TCP connections, and 3 CBR sources with total bandwidth 30 Mbps. Comparison with

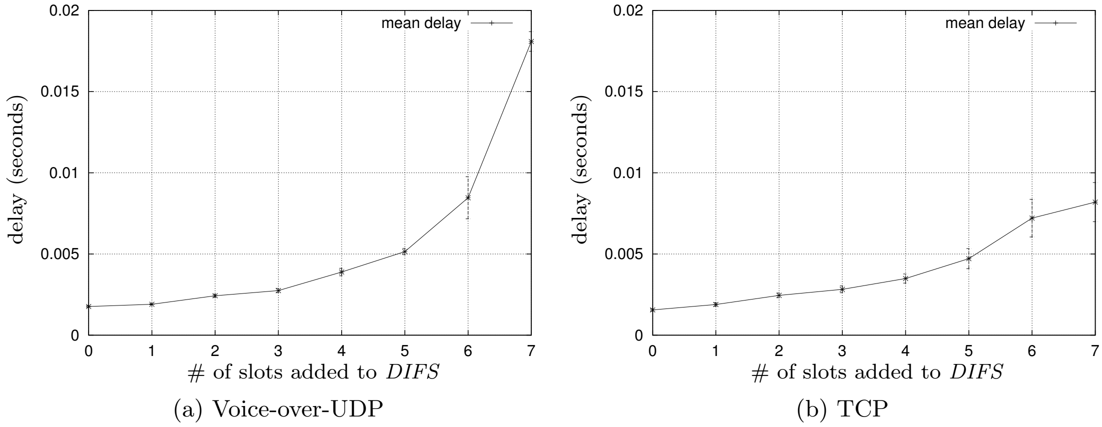



**Fig. 3.** Delay differentiation for different *DIFS* values.

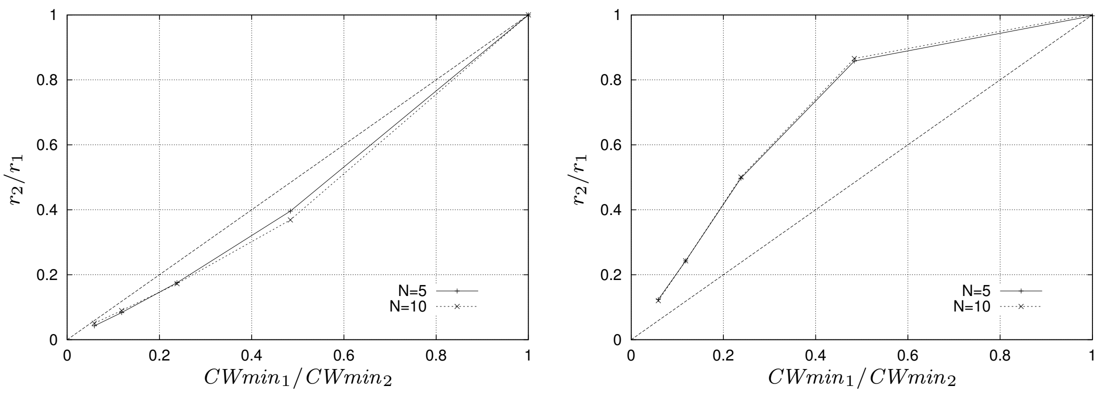
Fig. 3(a) shows that the addition of up to 6 slots results in an average delay that is the same as in voice-over-UDP traffic. The addition of more than 6 slot times results in a higher delay for voice-over-UDP traffic. Hence, the average delay appears to be less affected by *DIFS* for TCP traffic, compared to UDP traffic.

### 2.3 Differentiation based on *CWmin*

As already discussed, after each collision the contention window *CW* is doubled, until it reaches an upper bound *CWmax*. Backoff times are set to a random value in the interval $[0, CW - 1] \cdot \text{slot\_time}$. After a successful transmission, *CW* is reset to its initial value *CWmin*. For two or more sources entering the backoff procedure simultaneously with different *CWmin*, the source with smaller *CWmin* has a higher probability of transmitting its frame first.

Fig. 4(a) shows the throughput differentiation, for 5 and 10 CBR sources with aggregate rate 20 Mbps and 40 Mbps, respectively, and payload size 500 bytes. Observe that the ratio of throughput is approximately inversely proportional to the ratio of *CWmin* values. Fig. 4(b) shows that the differentiation in the case of TCP traffic is less effective; see also [4]. Indeed, increasing the value of *CWmin* from 16 to 32 results in a decrease in the throughput of only 15%. Also observe in Figs. 4(a) and 4(b) that the effect of the number of connections on the throughput ratio as a function of the *CWmin* ratio is minimal; this is not the case with *DIFS* differentiation with UDP traffic, Fig. 2(a).

Fig. 5(a) shows how the average delay of voice traffic depends on the value of *CWmin*, in the case of 2 UDP connections carrying voice traffic, and 3 CBR




(a) CBR-over-UDP (b) TCP

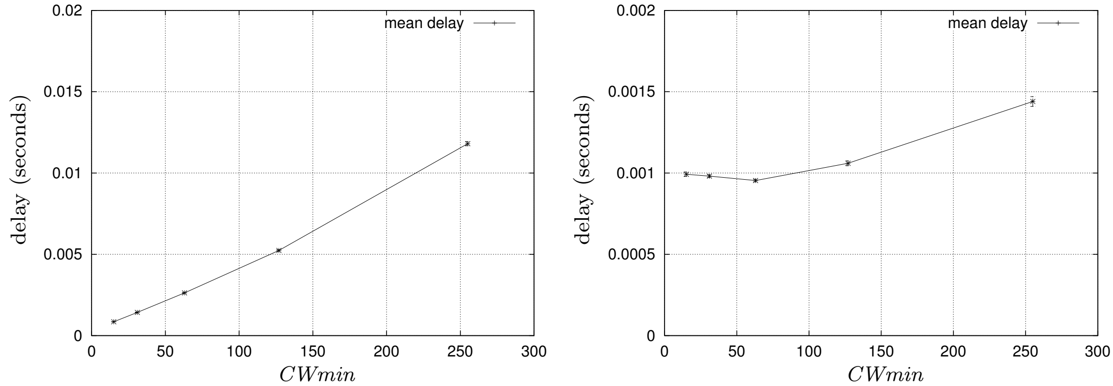
**Fig. 4.** Throughput ratio $r_2/r_1$ as a function $CWmin$ ratio $CWmin_1/CWmin_2$, for UDP and TCP traffic, and $N = 5$ and $N = 10$ sources.

sources with an aggregate bandwidth of 30 Mbps. The values for $CWmin$ that we consider are $16, 32, 64, 128$, and 256 time slots. As expected, the delay increases with increasing $CWmin$.

Fig. 5(b) shows how the average delay of TCP traffic depends on $CWmin$, in the case of 2 TCP connections, and 3 CBR sources with an aggregate bandwidth of 30 Mbps. First observe that the delay in the TCP case is an order of magnitude smaller than the delay in the UDP case. Furthermore, observe that for values of $CWmin$ up to approximately 64 time slots, there is no increase of the average delay; indeed in this range the delay decreases with increasing $CWmin$. For values of $CWmin$ above 64 time slots, the average delay increases with increasing $CWmin$. Further work seeks to investigate the extent to which this behaviour is due to TCP's congestion control algorithm.

## 3 Algorithm for assignment and adaptive recalculation of *CWmin*

In the previous section we investigated the service differentiation that can be achieved with different values of $CWmin$. The optimal, in terms of efficient network utilization, values of $CWmin$ will depend on the number of stations that are contending for the channel, hence no static values can be optimal for all cases. In this section we describe an approach for assigning $CWmin$ values to sources belonging to different classes, where each class has a corresponding weight, and for




(a) Voice-over-UDP (b) TCP

**Fig. 5.** Delay differentiation for different *CWmin* values.

adjusting these values when the network conditions change, in order to achieve high network utilization.

### 3.1 Assignment of *CWmin* values

Assume that different classes are associated with different weights, where a larger weight is assigned to a higher priority class. We consider the minimum contention window *CWmin* as the parameter used for differentiation. Based on the results of Section 2.3, the values of *CWmin* can be assigned inversely proportion to the weight of each class. Hence, if $\phi_i$ is the weight for class $i$, the value of the minimum contention window $CWmin_i$ for class $i$ is calculated using[1]

$$CWmin_i = \left\lfloor SF \cdot \frac{Li}{\phi_i} \right\rfloor , \tag{1}$$

where $SF$ is some scaling factor and $L_i$ is the frame size; the frame size is added in the above calculation so that differentiation depends solely on the weight, or equivalently classes with the same weight but different frame sizes achieve the same throughput. In the case of collisions, the usual exponential backoff algorithm of 802.11 is performed.

The above assignment of *CWmin* values for different classes, based on the class' weight and the frame size, is similar to the calculation of the backoff interval in the scheme proposed in [6]; our approach differs in that we assign different values of *CWmin* rather than the backoff interval, hence we do not change the

[1] More precisely, the left-hand side should be $CWmin_i - 1$, since the backoff is selected from $[0, CW - 1] \cdot \text{slot\_time}$. However, because we consider $CWmin \geq 16$, the difference in practise is insignificant.

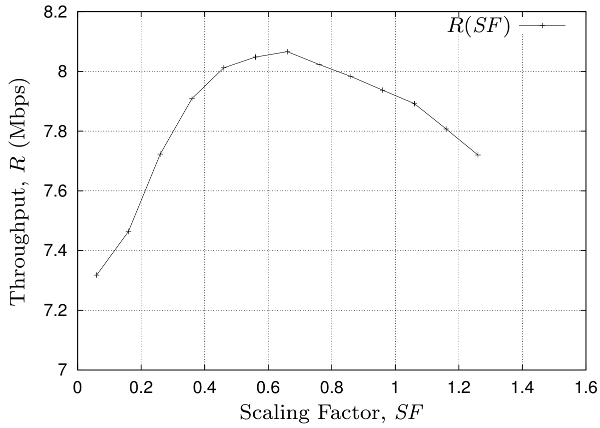



**Fig. 6.** Aggregate throughput as a function of the scaling factor. The measurement interval is $T_m =$ 10 seconds and the scaling factor step is $\Delta_{SF} = 0.1$.

behaviour of the exponential backoff algorithm in 802.11. This presents an important advantage, since our approach can be implemented solely at the access point of a wireless LAN, without requiring any changes at the wireless stations, assuming these support the IEEE 802.11e standard.

An important issue with the above approach for assigning *CWmin* values, and with the approach in [6], is the value of the scaling factor. As discussed in [6], the scaling factor affects the channel utilization, hence the overall throughput. Moreover, the optimal value of the scaling factor depends on the network load. In the following subsections we describe and evaluate an approach for dynamically adjusting the value of the scaling factor, and subsequently the values of *CWmin* according to Eq. 1, based on monitoring the actual throughput of the WLAN.

### 3.2 Adaptive recalculation of *CWmin*

The dependence of the aggregate throughput on the scaling factor is shown in Fig. 6; an identical result is presented in [6]. Observe that the throughput initially increases with the scaling factor up to a maximum value, after which it starts to decrease. Such a behaviour can be explained as follows: When the scaling factor is smaller than the optimum, the values of *CWmin* are small, resulting in many collisions, hence the utilization and throughput is low. On the other hand, when the scaling factor gets too large, the values of *CWmin* are large, resulting in a large percentage of idle times, hence the wireless channel is underutilized.

Another important observation is that the optimum scaling factor is different for different network loads; this motivates the need to recalculate the scaling

**Step 1.** The scaling factor is set to some initial value $SF_1$, and the aggregate throughput $R_1$ is measured at the access point (AP) over an interval $T_m$

**Step 2.** $SF_2 := SF_1 + \Delta_{SF}$, and the aggregate throughput $R_2$ is measured at the AP

**Step 3.** If $R_2 > R_1$ then $SF$ *increases* with step $\Delta_{SF}$ while the aggregate throughput increases else if $R_2 < R_1$ then $SF$ *decreases* with step $\Delta_{SF}$ while the aggregate throughput increases

**Step 4.** Let $SF^*$ be the optimal scaling factor, and $R^*$ the throughput when Step 3 ends
Let $SF_1 := SF^*$
For the same scaling factor, the AP continuously measures the current average throughput $R_1$
If $R_1 < aR^*$, where $a \in (0, 1)$ then goto Step 2

**Fig. 7.** Procedure for adjusting the scaling factor $SF$.

factor, hence the values of $CWmin$, adaptively when the network load changes. From the shape of Fig. 6, this can be achieved by measuring the average throughput, and moving the scaling factor in the direction that increases the aggregate throughput. Once such a procedure converges, the system would be operating with a scaling factor in the area around the peak in Fig. 6.

Based on the previous discussion, we propose the procedure for adjusting the scaling factor $SF$ that is shown in Fig. 7. After the scaling factor is adjusted, new values of $CWmin_i$ are calculated using Eq. 1.

In Step 4 of the procedure in Fig. 7, the current aggregate throughput $R_1$ can differ from the throughput $R^*$, which is measured when Step 3 ends, if there is a change in network conditions, e.g., when wireless stations arrive or depart.

The above procedure for adjusting the value of the scaling factor can be implemented at the access point. Moreover, each time the scaling factor changes, the access point would recalculate, using Eq. 1, the minimum contention window $CWmin_i$ for each class $i$, and distribute the new values to all the wireless stations; this communication can utilize the particular procedure that will be used in IEEE 802.11e.

The procedure described previously includes three parameters: the interval $T_m$ over which the throughput is measured, the scaling factor step size $\Delta_{SF}$, and the percentage $a$ used for deciding when to start the search for a new optimal scaling factor. We discuss each of these three parameters next.

**Throughput measurement period ($T_m$)** . The time interval $T_m$ over which the throughput is measured needs to be selected taking into account the tradeoff between measurement reliability and time for the procedure to converge to the optimal scaling factor. A large measurement interval would make the estimation more reliable; for example, Fig. 8(a) shows that a measurement period of $T_m =$ 1 second produces a lot of noise, hence the procedure outlined above cannot be applied without modification. On the other hand, a large measurement interval can lead to increased convergence time, and can result in not tracking changes that occurred between two successive measurements. In both cases, the overall wireless channel utilization would decrease.

**Scaling factor step ($\boldsymbol{\Delta_{SF}}$)** . The step $\Delta_{SF}$ for increasing or decreasing the scaling factor entails the tradeoff between accuracy in selecting the optimal scaling factor and convergence time. A large step can result in a less than optimal scaling factor when the procedure ends; on the other hand, a large step size would decrease the convergence time. Moreover, as Fig. 8(b) shows, a small step size, $\Delta_{SF} = 0.01$, can result in more noise. From Fig. 6 one can argue that a reasonable value for $\Delta_{SF}$ is 0.1, since the range of values of the scaling factor for which the throughput is within 2% of the maximum value, i.e., larger than approximately 7.9 Mbps, is a few times (approximately 6 in this example) of the value 0.1. Hence, independent of whether the peak in Fig. 6 is approached from left or right, with a step size of 0.1 the procedure will select a value for the scaling factor for which the aggregate throughput is within 2% of the maximum throughput.

**Throughput decrease threshold for initiating a new optimal scaling factor search ($\boldsymbol{a}$)** . In Step 4 of the procedure described previously, after the scaling factor has converged to some value, the access point continuously measures the aggregate throughput. If it observes a decrease of the average throughput larger than some percentage $(1-a)$, it initiates a search for a new optimal scaling factor. Increasing $a$ would make the procedure more reactive to network changes, but can also increase the variability of the scaling factor and the values of $CWmin$, hence of the performance observed by wireless stations, and would increase the number of $CWmin$ update messages sent by the access point to the stations.

### 3.3 Simulation Results

In this section we demonstrate the operation of the procedure described in the previous section for dynamically adjusting the values of $CWmin$, showing that it effectively reacts to changes of network conditions, maintaining a high overall

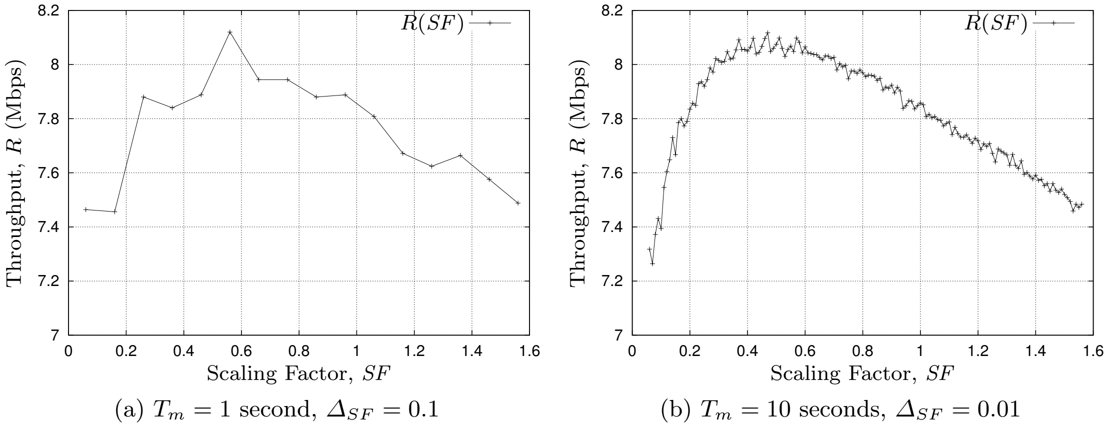


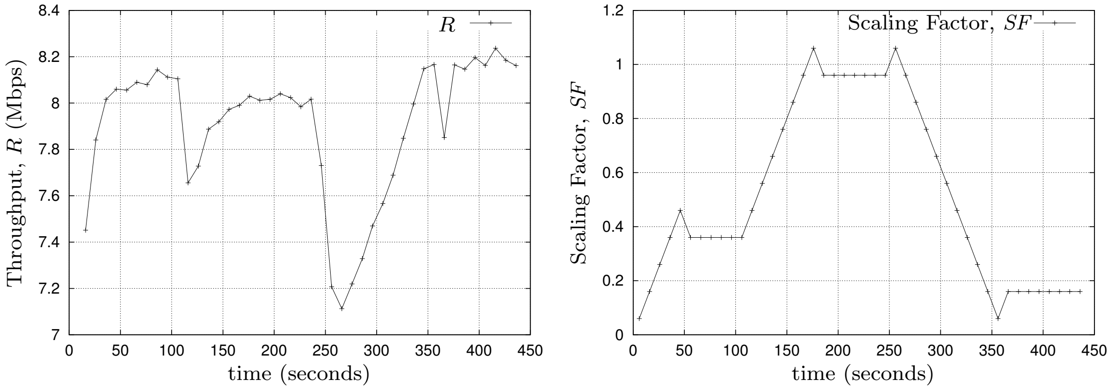

**Fig. 8.** Throughput as a function of scaling factor for different throughput measurement intervals $T_m$ and scaling factor steps $\Delta_{SF}$.

wireless channel throughput. The values for the three parameters of the procedure we consider are the following: throughput measurement interval $T_m = 10$ seconds, scaling factor step size $\Delta_{SF} = 0.1$, and throughput decrease threshold $a = 0.95$.

The wireless channel capacity is 11 Mbps. Initially, the wireless LAN contains 13 stations, each producing CBR traffic with rate 1 Mbps. At time 130 seconds, 20 more identical stations enter the WLAN; at time 240 seconds, 23 of the stations depart, leaving 10 stations in the WLAN. Fig. 9(a) shows the aggregate throughput as a function of time. Observe that the procedure for adjusting *CWmin* kicks in whenever the number of stations changes, hence tries to maintain a high throughput. Also observe that the maximum throughput is over 8 Mbps, but the exact value depends on the number of stations. Indeed, a smaller number of stations can achieve a larger aggregate throughput; this is due to the smaller probability of frame collisions when there are fewer stations. Fig. 9(b) shows how the scaling factor changes with time; observe that after each change, the scaling factor quickly converges to a new optimal value.

Fig. 10(a) and 10(b) shows the behaviour of the proposed scheme in the case of exponential and pareto traffic sources, respectively. Observe that the procedure, as expected, reacts to the changes of the network load, achieving high overall utilization.

The results from the above figures show that, for the particular parameter values, the procedure's convergence time is of the order of 10s of seconds, which is suf-




(a) Throughput (b) Scaling Factor

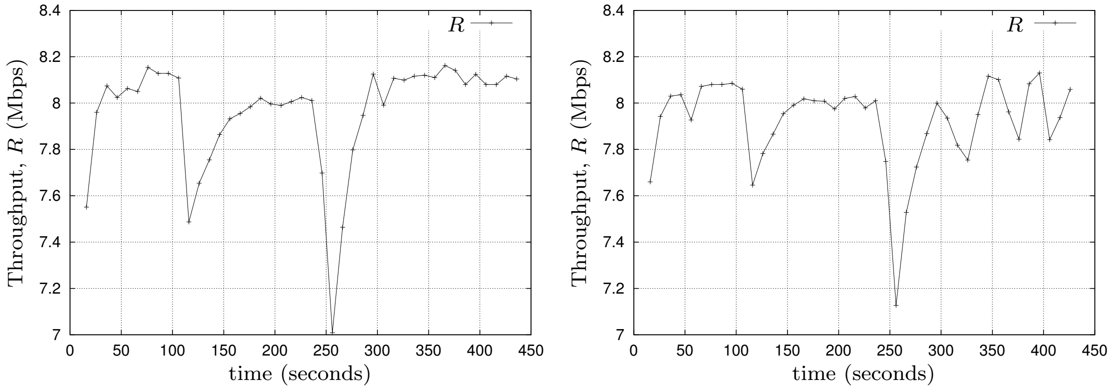
**Fig. 9.** Average throughput and scaling factor over time. CBR traffic.

ficient in environments where the number of wireless users change over timescales larger than a few minutes.

# 4 Conclusions

In this paper we first presented simulation experiments on service differentiation using various IEEE 802.11 MAC layer parameters, namely the maximum frame size, the minimum contention window, and the DCF inter-frame spacing interval. Our investigations showed the achievable service differentiation in terms of both throughput and delay.

Second, we proposed a simple yet effective procedure for adjusting the minimum contention window based on actual throughput measurements, in order to achieve high network utilization. An important advantage of our approach is that it can be implemented solely at the access point; the wireless stations are only required to support the emerging IEEE 802.11e standard. Our initial experimental results show that the approach is quite robust, and can effectively adjust the contention window to achieve high aggregate throughput. Further investigations focus on quantifying the tradeoffs of the various parameters of the proposed procedure. Such information will be useful for tuning these parameters in a real environment. Furthermore, the procedure's reactivity should depend on the timescales over which the network conditions change significantly; identifying these timescales can also assist in tuning the procedure's degree of reactivity.




(a) Exponential (b) Pareto

**Fig. 10.** Results for exponential and pareto traffic sources. Exponential: average "on" 800 ms, "off" 200 ms. Pareto: same average on/off as exponential, and shape parameter 1.5. The number of stations is the same as in Fig. 9.

Other important research issues include approaches for achieving service differentiation in the case of TCP flows, and modification of our approach for ad hoc wireless networks.

# References


1. IEEE: 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE (1999)
2. IEEE: 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 (PHY) specifications: MAC Enhancements for Quality of Service (QoS), IEEE 802.11e/D2.0, IEEE 802.11e WG. (2001)
3. Aad, I., Castelluccia, C.: Differentiation mechanisms for IEEE 802.11. In: Proc. of IEEE INFOCOM'01. (2001)
4. Aad, I., Castelluccia, C.: Remarks on per-flow differentiation in IEEE 802.11. In: Proc. of European Wireless 2002. (2002)
5. Bharghavan, V., Demers, A., Shenker, S., Zhang, L.: MACAW: a media access protocol for wireless LANs. In: Proc. of ACM SIGCOMM'94. (1994)
6. Vaidya, N.H., Bahl, P., Gupta, S.: Distributed fair scheduling in a wireless LAN. In: Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM). (2000)
7. Bensaou, B., Wang, Y., Ko, C.C.: Fair medium access in 802.11 based wireless ad-hoc networks. In: Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (2002)
8. Qiao, D., Shin, K.G.: Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under DCF. In: Proc. of IEEE/IFIP International Conference of Quality of Service (IWQoS'02). (2002)

9. Veres, A., Campbell, A.T., Barry, M., Sun, L.H.: Supporting service differentiation in wireless packet networks using distributed control. IEEE J. Select. Areas in Commun. **19** (2001) 2094–2104
10. UCB/LBNL/VINT: Network Simulator - ns (version 2). (`http://www-mash.cs.berkeley.edu/ns.html`)