Mobile Networks and Applications 8, 223–235, 2003

Quality of Service Schemes for IEEE 802.11 Wireless LANs – An Evaluation

ANDERS LINDGREN, ANDREAS ALMQUIST and OLOV SCHELÉN
Division of Computer Science and Networking, Department of Computer Science and Electrical Engineering, Luleå University of Technology, SE-971 87 Luleå, Sweden

Abstract. This paper evaluates four mechanisms for providing service differentiation in IEEE 802.11 wireless LANs. The evaluated schemes are the Point Coordinator Function (PCF) of IEEE 802.11, the Enhanced Distributed Coordinator Function (EDCF) of the proposed IEEE 802.11e extension to IEEE 802.11, Distributed Fair Scheduling (DFS), and Blackburst. The evaluation was done using the *ns*-2 simulator. Furthermore, the impact of some parameter settings on performance has also been investigated. The metrics used in the evaluation are throughput, medium utilization, collision rate, average access delay, and delay distribution for a variable load of real time and background traffic. The simulations show that the best performance is achieved by Blackburst. PCF and EDCF are also able to provide pretty good service differentiation. DFS can give a relative differentiation and consequently avoids starvation of low priority traffic.

Keywords: Quality of Service, wireless LANs, performance evaluation, MAC protocols

1. Introduction

Wireless networks are superior to wired networks with regard to aspects such as ease of installation and flexibility. They do, however, suffer from lower bandwidth, higher delays, higher bit-error rates, and higher costs than wired networks. With the advent of Wireless Local Area Networks (WLANs), bandwidth has increased and prices have decreased on wireless networking solutions. These factors have made WLANs a very popular wireless networking solution. Given the coverage and low price, it is likely that demands for the ability to run real-time applications such as voice over IP over these networks will increase. If such applications shall be usable, considering the characteristics of wireless networks, some kind of service differentiation must be employed to let certain types of traffic get better performance.

The IEEE 802.11 standard [8] for WLANs is the most widely used WLAN standard today. Since it uses a shared medium, it has some inherent problems, such as low medium utilization, risk of collisions and problem of providing differentiation between different types of traffic. There is a mode of operation in IEEE 802.11 that can be used to provide service differentiation, but it has been shown to perform poorly and give poor link utilization [15], so several new service differentiation schemes have been proposed. We study and evaluate four schemes for providing Quality of Service (QoS) over IEEE 802.11 wireless LANs: the PCF mode of the IEEE 802.11 standard [8], Distributed Fair Scheduling [14], Blackburst [12], and Enhanced DCF [2].

This paper summarizes work previously published as position papers [9,10], and does a more thorough analysis than the previous papers. The rest of the paper is organized as follows. Section 2 provides an overview of IEEE 802.11 and the proposed schemes for service differentiation. Section 3 describes our simulation scenarios and metrics. In section 4 we present the results of our simulations, section 5 discusses some issues, and section 6 concludes.

2. Overview of evaluated schemes

In this section we describe the QoS mechanisms we have evaluated. For further details we refer to [2,8,12–14].

2.1. IEEE 802.11

IEEE 802.11 has two different access methods, the mandatory Distributed Coordinator Function (DCF) and the optional Point Coordinator Function (PCF). The latter aims at supporting real-time traffic.

2.1.1. Distributed coordinator function

The DCF is the basic access mechanism of IEEE 802.11. It uses a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm to mediate the access to the shared medium. Before a data frame is sent, the station senses the medium. If it is idle for at least a DCF interframe space[1] (DIFS) period of time, the frame is transmitted. Otherwise, a backoff time B (measured in time slots) is chosen randomly in the interval $[0, CW)$, where CW is the so called Contention Window. After the medium has been detected idle for at least a DIFS, the backoff timer is decremented by one for each time slot the medium remains idle. If the medium becomes busy during the backoff process, the backoff timer is paused, and is restarted when the medium has been sensed idle for a

[1] An interframe space, IFS, is the time a station waits when the medium is idle before attempting to access it. IEEE 802.11 defines several IFSs, and by using shorter IFS, the medium is accessed prior to stations using a longer IFS. This is, e.g., used to ensure that an acknowledgment frame is sent before any other station can send data.

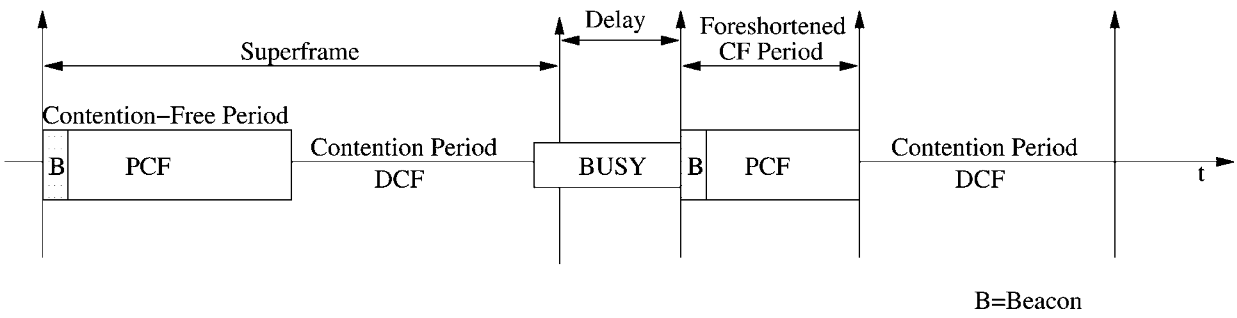


Figure 1. The superframe of IEEE 802.11.

DIFS again. When the backoff timer reaches zero, the frame is transmitted. Upon detection of a collision (which is detected by the absence of an acknowledgment frame to the data frame), the contention window is doubled according to

$$CW_i = 2^{k+i-1} - 1, \tag{1}$$

where i is the number of attempts (including the current one) to transmit the frame that has been done, and k is a constant defining the minimum contention window, $CW_{\min} = 2^k - 1$. A new backoff time is then chosen and the backoff procedure starts over. The backoff mechanism is also used after a successful transmission before sending the next frame. After a successful transmission, the contention window is reset to $CW_{\min}$.

2.1.2. Point coordinator function

PCF is a centralized, polling-based access mechanism which requires the presence of a base station that acts as Point Coordinator (PC). If PCF is supported, both PCF and DCF coexist and in this case, time is divided into superframes as shown in figure 1. Each superframe consists of a contention period where DCF is used, and a contention free period (CFP) where PCF is used. The CFP is started by a special frame (a *beacon*) sent by the base station. Since the beacon is sent using ordinary DCF access method, the base station has to contend for the medium, and therefore the CFP may be shortened.

The PC keeps a list of mobile stations that have requested to be polled to send data. During the CFP, it sends poll frames to the stations when they are clear to access the medium. Upon reception of a poll frame, the station sends a data packet if it has any packet queued. To ensure that no DCF stations are able to interrupt this mode of operation, the IFS between PCF data frames is shorter than the usual DIFS. This space is called a PCF interframe space (PIFS). To prevent starvation of stations that are not allowed to send during the CFP, there must always be room for at least one maximum length frame to be sent during the contention period.

2.2. IEEE 802.11e – Enhanced DCF

Task group E of the IEEE 802.11 working group are currently working on an extension to the IEEE 802.11 standard, called IEEE 802.11e. The goal of this extension is to enhance the access mechanisms of IEEE 802.11 and provide a distributed access mechanism that can provide service differentiation. All the details have not yet been finalized, but a new access mechanism called Enhanced DCF (EDCF) has been selected [2]. This is an extension of the basic DCF access mechanism in the original standard. Since devices complying with the old standard are widely deployed, great care was taken to ensure that EDCF should be inter-operable with the old DCF. The EDCF mechanism allows traffic to be classified into 8 different traffic classes, by modifying the minimum contention window ($CW_{\min}$) and the interframe space used for data transmissions. Choosing a smaller default contention window for a station will cause that station to generate shorter backoff intervals, thus gaining priority over a station with a larger $CW_{\min}$ which generates longer backoff intervals.

To be able to further differentiate between stations using the same contention window, different interframe spaces are used by different traffic classes. Instead of waiting a DIFS before trying to access the medium, or starting to decrement the backoff timer as in ordinary DCF, an interframe space called Arbitration Interframe Space (AIFS) is used. Each traffic class uses its own AIFS which equals a DIFS plus a number of time slots (possibly zero). This means that traffic using a large AIFS (many "extra" time slots) will have lower priority than traffic using a small AIFS, since they will wait longer before trying to access the medium or starting to decrement the backoff timer.

Mechanisms similar to EDCF that use different backoff algorithms and interframe spaces for different priority levels have previously been proposed by for example Deng and Chang [5] and Barry et al. [1].

In IEEE 802.11e there is also the possibility to use *packet bursting* [3] to enhance the performance, and achieve better medium utilization. The packet bursting concept means that once a station has gained access to the medium through ordinary contention, it can be allowed to send more than one frame without contending for the medium again. After getting access to the medium the station is allowed to send as many frames it wishes as long as the total access time does not exceed a certain limit (TxOpLimit). To ensure that no other station interrupts the packet burst, the interframe space used between the reception of an acknowledgment, and the transmission of the next data frame in the packet burst is a SIFS (Short Interframe Space), which is the same IFS that is used between data and acknowledgment frames. If a collision occurs (no acknowledgment frame is received), the packet burst is terminated. Since packet bursting might increase the jitter, it is recommended that TxOpLimit is chosen such that it is not longer than the time required for the transmission of a data frame of maximum size.

2.3. Distributed fair scheduling

It is not always desirable to completely sacrifice the performance of low priority traffic in order to give very good service to high priority traffic. Often it can be good to be able to provide relative differentiation, for example specifying that one type of traffic should get twice as much bandwidth as some other type of traffic. Vaidya et al. propose an access scheme called Distributed Fair Scheduling (DFS) which applies the ideas behind fair queuing in the wireless domain [14].

There exist several fair queuing schemes that provide fair allocation of bandwidth between different flows on a node [6,7]. In this context, fair means that each flow gets bandwidth proportional to some *weight* that has been assigned to it. These schemes are centralized in the sense that they run on a single node which has access to all information about all the flows. Since different weights can be assigned to the flows, this can be used for differentiation between flows.

The Distributed Fair Scheduling scheme is based on the fair queuing mechanism known as Self-Clocked Fair Queueing [6], and uses the backoff mechanism of IEEE 802.11 to determine which station should send first. Before transmitting a frame, the backoff process is always initiated, even if no previous frame has been transmitted. The backoff interval is calculated as shown in

$$B = \left\lfloor \rho \times \left\lfloor \mathit{Scaling\_Factor} \times \frac{\mathit{size}_{\mathrm{packet}}}{\phi} \right\rfloor \right\rfloor, \qquad (2)$$

where $\mathit{size}_{\mathrm{packet}}$ is the size of the packet to send, ϕ is the weight of the station, ρ is a random variable with mean 1 (paper [14] uses a uniform random variable in the interval [0.9, 1.1], and so will we in our evaluations), and *Scaling_Factor* is used to scale the backoff intervals to values of suitable magnitude. Since the backoff interval will be longer the lower the weight of the sending station is, differentiation will be achieved. Further, fairness is achieved by using the size of the packet to be sent in the calculation of the backoff interval. This will cause larger packets to get longer backoff intervals than small packets, allowing a station with small packets to send more often so that the same amount of data is sent.

If a collision occurs, a new backoff interval is calculated using the backoff algorithm of the IEEE 802.11 standard where the contention window is given by equation (1), with $CW_{\min}$ set to 3. The reason for choosing such a short contention window even though a collision has occured is that DFS tries to maintain fairness among nodes, and thus a node that was "scheduled" to send a packet, should be able to send it as soon as possible. Otherwise fairness would suffer.

2.4. Blackburst

To improve the performance of real time streams in wireless LANs, Sobrinho and Krishnakumar proposed a scheme called Blackburst [12], and some enhancements to that in [13]. The main goal of Blackburst is to minimize the delay for real time traffic, and it is somewhat different from the other schemes since it imposes certain requirements on the traffic to be prioritized. Blackburst requires that all high priority stations try to access the medium with constant intervals, t_{sch} (this interval has to be the same for all high priority stations). Further, Blackburst also requires the ability to jam the wireless medium for a period of time.

When a high priority station wants to send a frame, it senses the medium to see if it has been idle for a PIFS and then sends its frame. On the other hand, if the medium is found busy, the station waits until the channel has been idle for a PIFS and then enters a black burst contention period. The station now sends a so called black burst by jamming the channel for a period of time. The length of the black burst is determined by the time the station has been waiting to access the medium, and is calculated as a number of *black slots*. After transmitting the black burst, the station listens to the medium for a short period of time (less than a black slot) to see if some other station is sending a longer black burst. That would imply that the other station has waited longer and, thus, should access the medium first. If the medium is idle, the station will send its frame, otherwise it will wait until the medium becomes idle again and enter another black burst contention period. By using slotted time, and imposing a minimum frame size on real time frames, it can be guaranteed that each black burst contention period will yield a unique winner [12].

After the successful transmission of a frame, the station schedules the next access instant (when the station will try to transmit the next frame) t_{sch} seconds in the future. This has the nice effect that real-time flows will synchronize, and share the medium in a TDM fashion [12]. This means that unless there is a transmission by a low priority station when an access instant for a high priority station occurs, very little blackbursting will have to be done once the stations have synchronized. Low priority stations use the ordinary DCF access mechanism of IEEE 802.11.

Two different modes of operation of Blackburst, with and without feedback from the MAC layer to the application, exist [13]. If the application is not Blackburst-aware, and the mode without feedback is used, a slack time, δ, is used to ensure stability of the system. The access intervals are scheduled δ before the time the packet is expected to arrive at the MAC layer. This is to ensure that delayed access instants caused by interfering traffic does not make the system unstable.

3. Simulations

To evaluate the methods described in section 2, we use the network simulator *ns*-2 [11] which has IEEE 802.11 DCF functionality. We extended the simulator by implementing IEEE 802.11 PCF[2] and Blackburst, and by adding implemen-

[2] Our *ns* implementation of PCF can be found at `http://www.sm.luth.se/~dugdale/index/software.shtml`

tations of DFS and EDCF made by other people,[3] and ran the simulation scenarios described below to measure five different metrics: throughput, medium utilization, collision rate, access delay and delay distribution.

3.1. Scenarios

In our simulations modeling a 2 Mbit/s wireless LAN, the wireless topology consisted of several wireless stations and a base station connected to a wired node which serves as a sink for the flows from the wireless domain. In an IEEE 802.11 network in infrastructure mode, the mobile nodes always communicate directly with the base station, so the results would be similar even if the mobile nodes communicated with each other. An example of the topology can be seen in figure 2. The parameters for the wired link were chosen to ensure that the bandwidth bottleneck of the system is within the wireless LAN. All wireless stations are located such that every station is able to detect a transmission from any other station, and there is no mobility in the system.

Our simulations consist of traffic that has been chosen to be similar to data generated by for example a variable bit rate audio or video encoder. The high priority stations generate packets with packet sizes taken from a normal distribution with mean 300 bytes, and standard deviation 40 bytes. We have used inter-packet intervals of 25 and 40 ms, which gives us data flows with an average bit rate of 96 and 60 kbit/s. We will refer to these as high and low bit-rate high priority traffic. The low priority stations generate packets every 50 ms, with a packet size taken from a normal distribution with mean 800 bytes, and standard deviation 150 bytes (corresponding to a mean bit-rate of 128 kbit/s). Our measurements started after a warm-up period that allowed initial control traffic like ARP to be exchanged so it would not affect our results. We have had some fixed numbers of low priority stations (3 and 12 stations), and gradually increased the number of high priority stations to increase the load of the system.

When choosing the parameter settings to use for the different schemes, we have tried to use settings specified in the standards or papers where the schemes are specified [8, 12–14], or that has elsewhere been shown to be sensible choices [2]. Unfortunately, the impact of the parameter settings for the different schemes are not always well investigated, and the preferable parameter setting may also vary depending on the expected traffic pattern. Table 1 shows the parameter values we have used in our simulations for the comparison of the schemes (where a time unit, TU, equals 1024 μs). The main parameter that can be modified for PCF is the size of the superframe, which actually determines how often a station can be polled. The length of the superframe is not specified in the standard, nor are any recommendations of superframe size given. Thus, we have investigated by simulation the impact this has on the throughput and medium utilization. The results of this can be found in section 4.1. During a CFP, the Point Coordinator (the base station) polls the stations in its polling list in a round robin fashion. If all stations have been polled once, the CFP will be ended prematurely. If there is not enough time to poll all stations in the current CFP, the next station in the list will be polled first in the next CFP. The IEEE 802.11 working group has still not decided on what values of AIFS and $CW_{\min}$ that should be used for the different traffic classes. However, we use values which have been shown to be reasonable from simulations done by Chesson et al. [3]. The TxOpLimit value was set to the time required to transmit a frame of maximum size, as recommended. Furthermore, we assume that most applications will not be aware of Blackburst, and thus, decide to use the mode without feedback in our simulations to make the results comparable with the results from the other methods, and to make them more applicable to real life.

[3] Thanks to Nitin H. Vaidya and Greg Chesson.

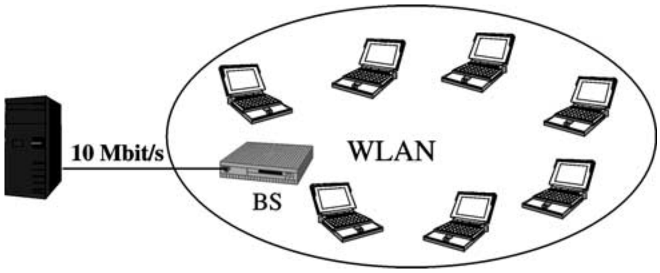


Figure 2. Example of our simulation setup.

Table 1
Simulation parameter values.

Parameter	Value
Time slot	20 μs
DIFS	50 μs
PIFS	30 μs
Superframe	20 TU
Max CFP	18.85 TU
c_{bitrate}	2 Mbit/s
$CW_{\min}$	31
Blackburst	
Black slot	20 μs
Slack time δ	5 ms
EDCF	
$\text{AIFS}_{\text{high prio}}$	50 μs
$CW_{\min}$ high prio	31
$\text{AIFS}_{\text{low prio}}$	90 μs
$CW_{\min}$ low prio	63
TxOpLimit	0.00953 s
DFS	
High weight	0.075
Low weight	0.025
Scaling_Factor	0.002

3.2. Metrics

The metrics we have used in our evaluation are throughput, medium utilization, collision rate, access delay, and cumulative delay distribution.

The *average throughput* for the stations at each priority level shows how well the QoS schemes can provide service differentiation between the various priority levels. To be able to compare the graphs from different levels of load, we have

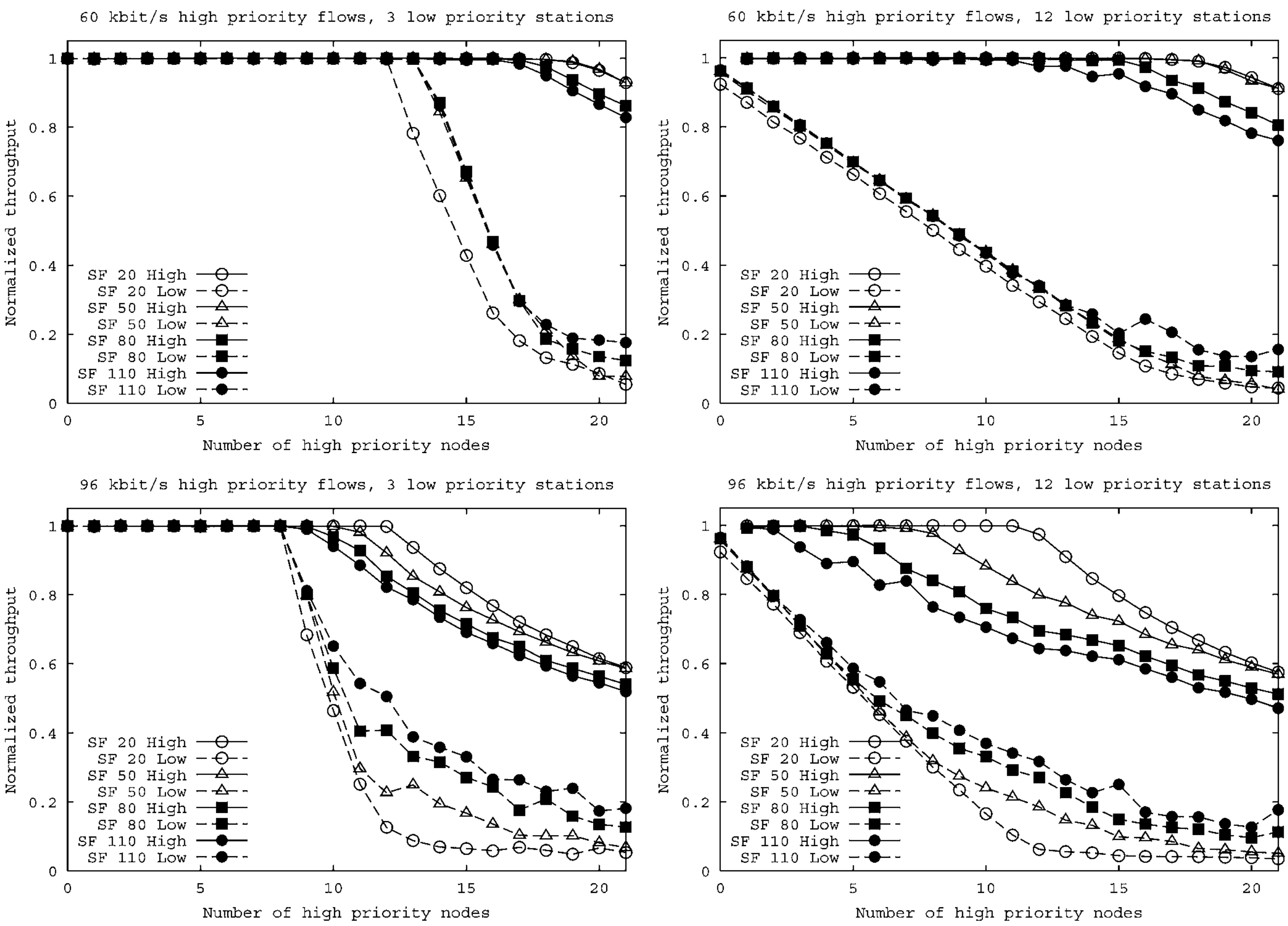


Figure 3. Average throughput for different superframe sizes.

chosen to plot a normalized throughput on the y axis, rather than the absolute throughput. The normalized throughput is calculated as the percentage of the offered data that is actually delivered to the destination.

Wireless bandwidth is a scarce resource, so, efficient use of it is vital. Therefore, we also study the *medium utilization* of the different schemes. To do this, we measure how large percentage of time that is used for successful transmission of data frames. Thus we can see how much of the time (and the medium capacity) that is used for data transmission, and how much that is wasted on other things.

The *collision rate* is the average number of collisions that occur per second. A large number of collisions will reduce the performance.

We define *access delay* as the time the Head-of-Line data packet spends at the MAC layer before being successfully transmitted out on the wireless medium. The reason for studying average access delay is that many real-time applications are very sensible to high delays, after which the data will be useless. Therefore, it is important to provide low delay for real-time flows. Because real-time applications often have a delay bound after which the data is useless, it does not suffice to just study the average access delay, since the average might be rather low even if a large part of the packets have unacceptable delays. We present the *cumulative distribution of the access delays* for high priority traffic to find out the percentage of the packets that are below certain delay bounds.

4. Results

4.1. Determining PCF superframe size

To validate the comparison between the different schemes, it is important to ensure that all schemes have reasonable parameter settings that does not affect the performance of the scheme adversely. Here we investigate the impact the superframe size has on performance for PCF, and determine what size to be used in the further comparisons. Using a short superframe increases the number of control frames sent, which might waste resources if there is not enough traffic to accommodate the polls. On the other hand, it also causes stations to be polled more frequently, allowing the high priority stations to send more traffic if the load is high. Using a longer superframe will reduce the amount of control frames sent, but will also cause stations to be polled less often, which might lead to too high delays and too low throughput. It is reasonable to believe that the best performance for high priority traffic would be achieved by having a superframe size similar to the

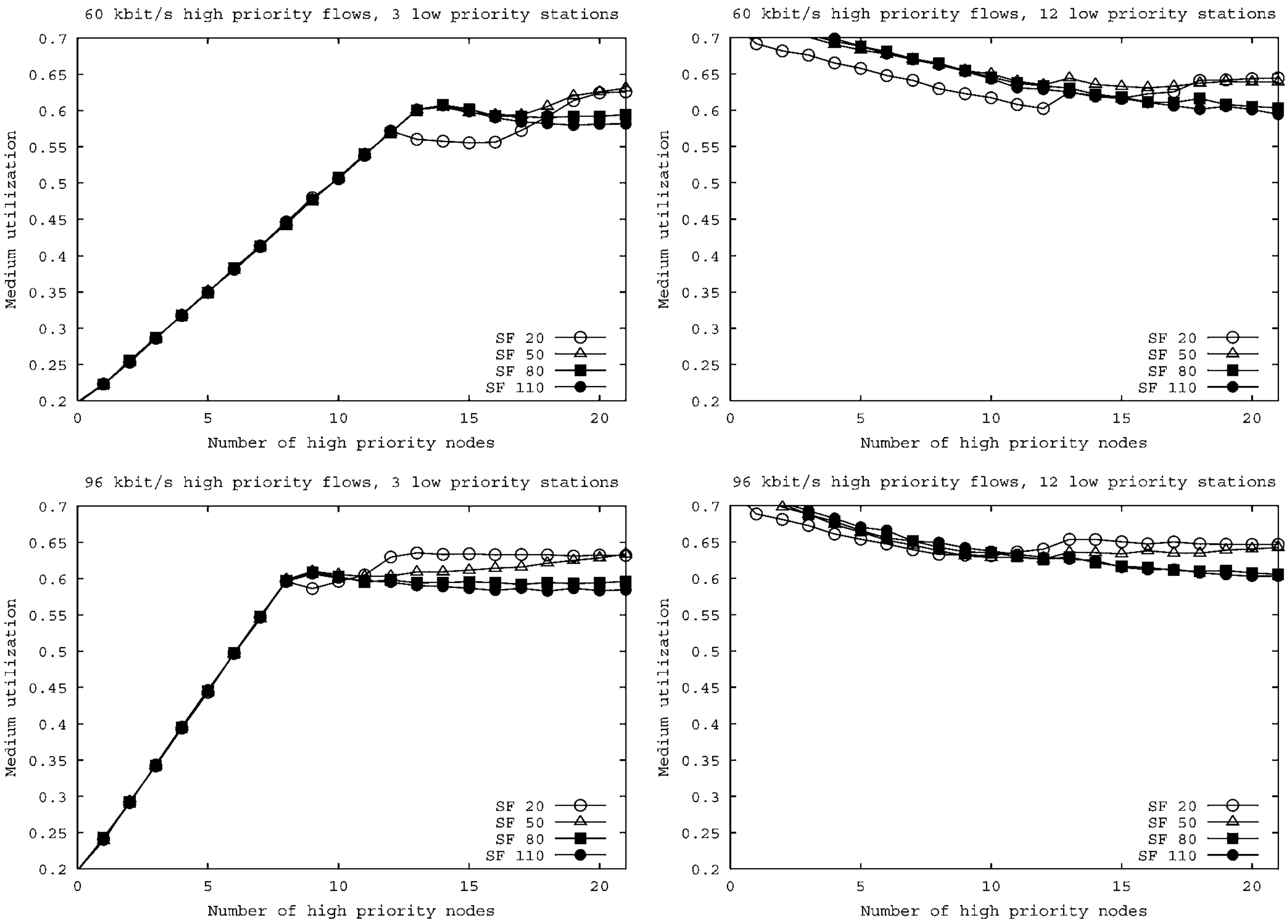


Figure 4. Medium utilization for different superframe sizes.

interval between the frames generated by the nodes. In many cases there might not be such a thing as a typical interval between frames, and then other considerations must be made when selecting superframe size. In this paper we do however only focus on the scenario where the high priority traffic is indeed periodic.

Figure 3 shows the throughput achieved for some different superframe sizes, chosen to range from just under the packet inter-arrival time of the high priority flows with high bit rate (a superframe of 20 time units (TU)), up to 110 time units, indicated in the graphs by "`SF` x `High/Low`" for a superframe size of x TU, and `High` or `Low` indicating if the curve is for high or low priority traffic. We can see that smaller sizes of the superframe (close to the inter-packet intervals) give better performance to high priority traffic. This, of course, impacts the performance of the low priority traffic in a negative way. For the scenario with low bit-rate high priority traffic, similar performance is achieved for high priority traffic with superframe sizes of 20 and 50 time units, but for high priority traffic with higher bitrate, the performance is better with a superframe size of 20 time units (which is close to the packet inter-arrival time of 25 ms for those flows). In figure 4 we can see how the medium utilization is affected by the size of the superframe. It is interesting to see that for lower loads of high priority traffic, the medium utilization for the cases with the smallest superframe size is lower than for the other superframe sizes, but as the load increases, the utilization gets better than for the other superframe sizes. We believe the reason of this is that when the load is low and the superframe is short, poll frames are sent more often than data packets are generated at the mobile nodes, rendering those poll frames useless to the stations receiving them, thus only wasting bandwidth that could otherwise have been used by the low priority traffic. When the load and the number of stations to be polled increases, the polling of all high priority stations might not fit into a single superframe. This means that the demand for bandwidth is higher than the supply, so whenever a high priority station receives a poll frame, it has some data to send, thus increasing the utilization.

It seems like our initial feeling that it should be good with a superframe that is similar to the packet inter-arrival time of high priority traffic was valid. While the smallest superframe size gives the best result (in terms of high priority traffic throughput and medium utilization) in the scenarios with high bit-rate high priority traffic, a somewhat larger superframe (50 TU in our simulations) seems better in the scenarios with low bit-rate high priority traffic, where the packet inter-arrival time is closer to 50 TU than 20 TU.

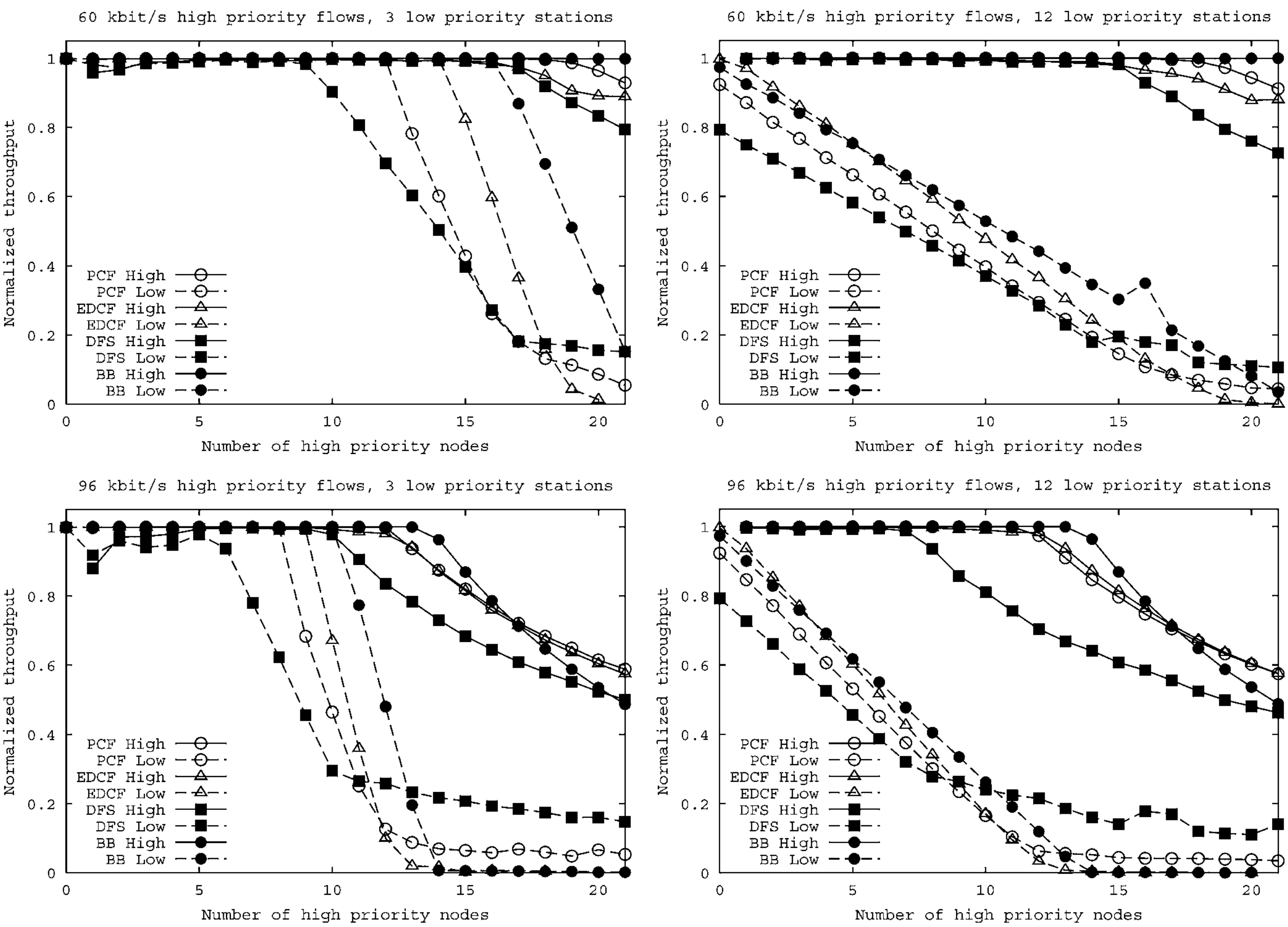


Figure 5. Average throughput for a station at the given priority level.

We decided to use a superframe of 20 time units for PCF in the comparison of the different QoS schemes. This decision was made to try to achieve as good throughput as possible for high priority traffic. The problem with this setting is the comparably low medium utilization in some scenarios, but since one of the main objectives of this evaluation is to determine the service differentiation capabilities of the scheme, we believe that this is the right choice to make.

4.2. Performance comparison

4.2.1. Throughput

The first metric investigated is throughput. Figure 5 shows the normalized throughput for low and high priority stations versus the number of high priority stations for some fixed number of low priority stations. Each graph represents different load conditions by setting different numbers of low priority stations and bit-rates of high priority stations. We can see that the Blackburst scheme provides the best performance for high priority traffic with regard to throughput. Blackburst is able to provide perfect service (in the sense that all packets are delivered) for at least up to 21 low bit-rate high priority stations. Stations using the other schemes experience a performance loss when a certain number of high priority stations is reached, and it is only at very high loads (many high bit-rate high priority stations) that the performance of Blackburst drops below that of the other schemes. DFS is the scheme for which high priority traffic starts to lose performance first. It should however be noted, that this should be considered the correct behaviour, since the objective of DFS is to provide fair (relative) differentiation (instead of trying to give perfect service to high priority traffic at any cost). This can also be seen from the performance of low priority traffic. Unlike the other schemes, DFS always allocates a share of the bandwidth for low priority traffic and avoids starvation. It might seem strange that the DFS low priority stations are not able to send all data even when they are alone on the medium. This is however logical due to the rather long backoff intervals used by the low priority stations. Summing the transmission and backoff times for all stations show that the total time required to transmit all data is longer than the interval between low priority data packets, thus all cannot be sent. Noteworthy is also that for lower loads, Blackburst is the scheme that gives the best performance to low priority traffic as well as high priority traffic.

When the high priority traffic have higher bit rate, all schemes can facilitate less flows with good service. Inter-

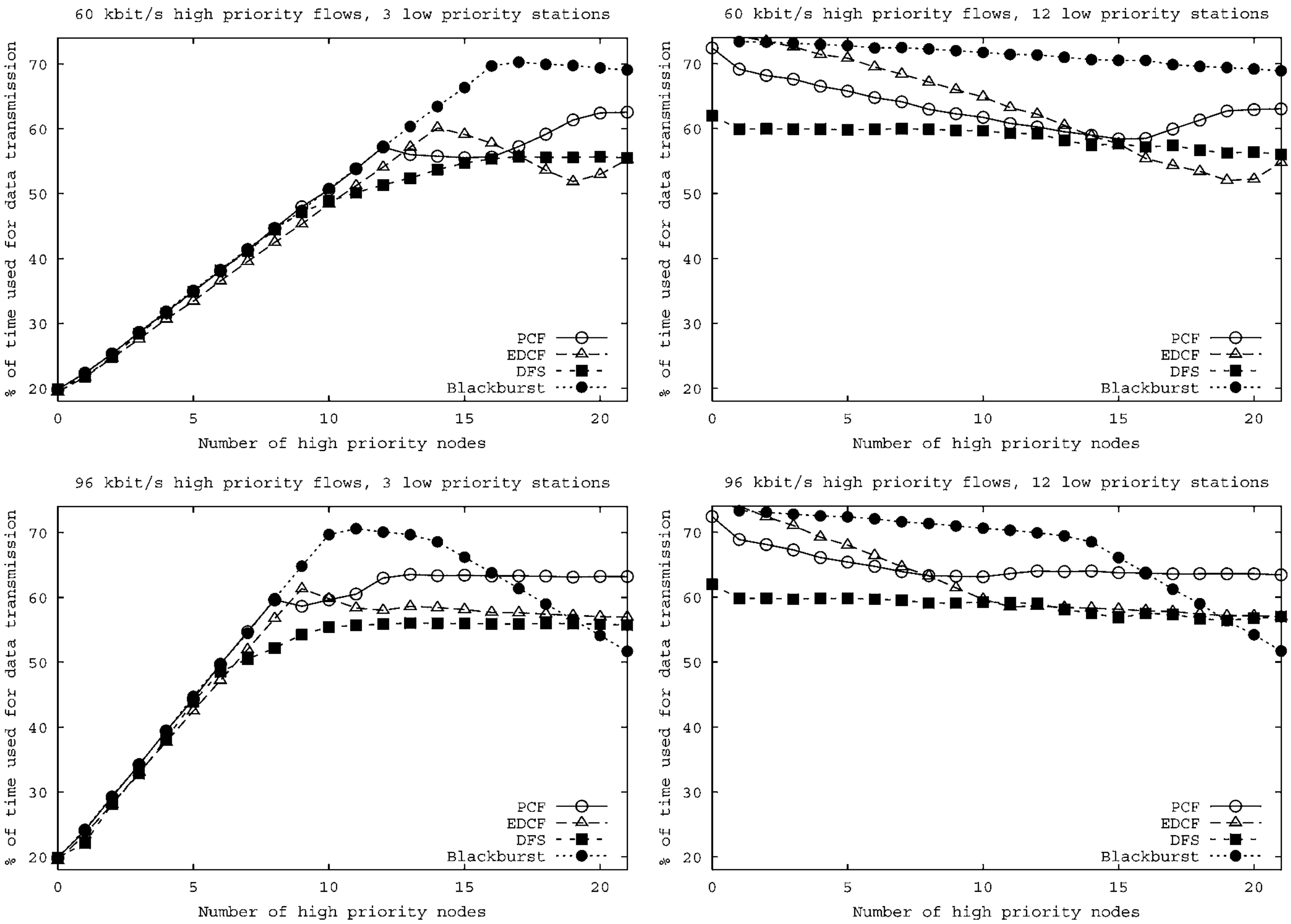


Figure 6. Medium utilization for the different QoS schemes.

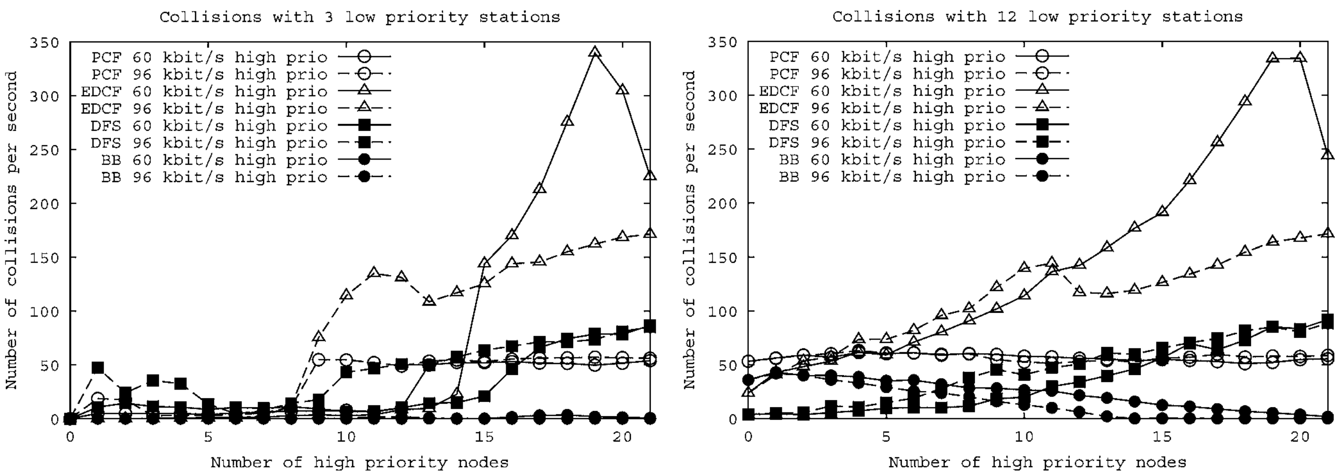


Figure 7. Collisions for the different schemes.

esting to note is that the difference in performance between Blackburst, PCF and EDCF is now quite small. Blackburst is able to give good performance to slightly more stations than EDCF, but for high numbers of high priority stations, EDCF get better performance. Both EDCF and Blackburst unfortunately starve low priority traffic rather fast, and PCF only gives a very small share of the bandwidth to low priority traffic. At high loads, there is a significant difference between the performance of low priority traffic for these schemes and for DFS. While not giving very good performance to high priority traffic, it does not starve the low priority traffic, but always let it have a portion of the bandwidth.

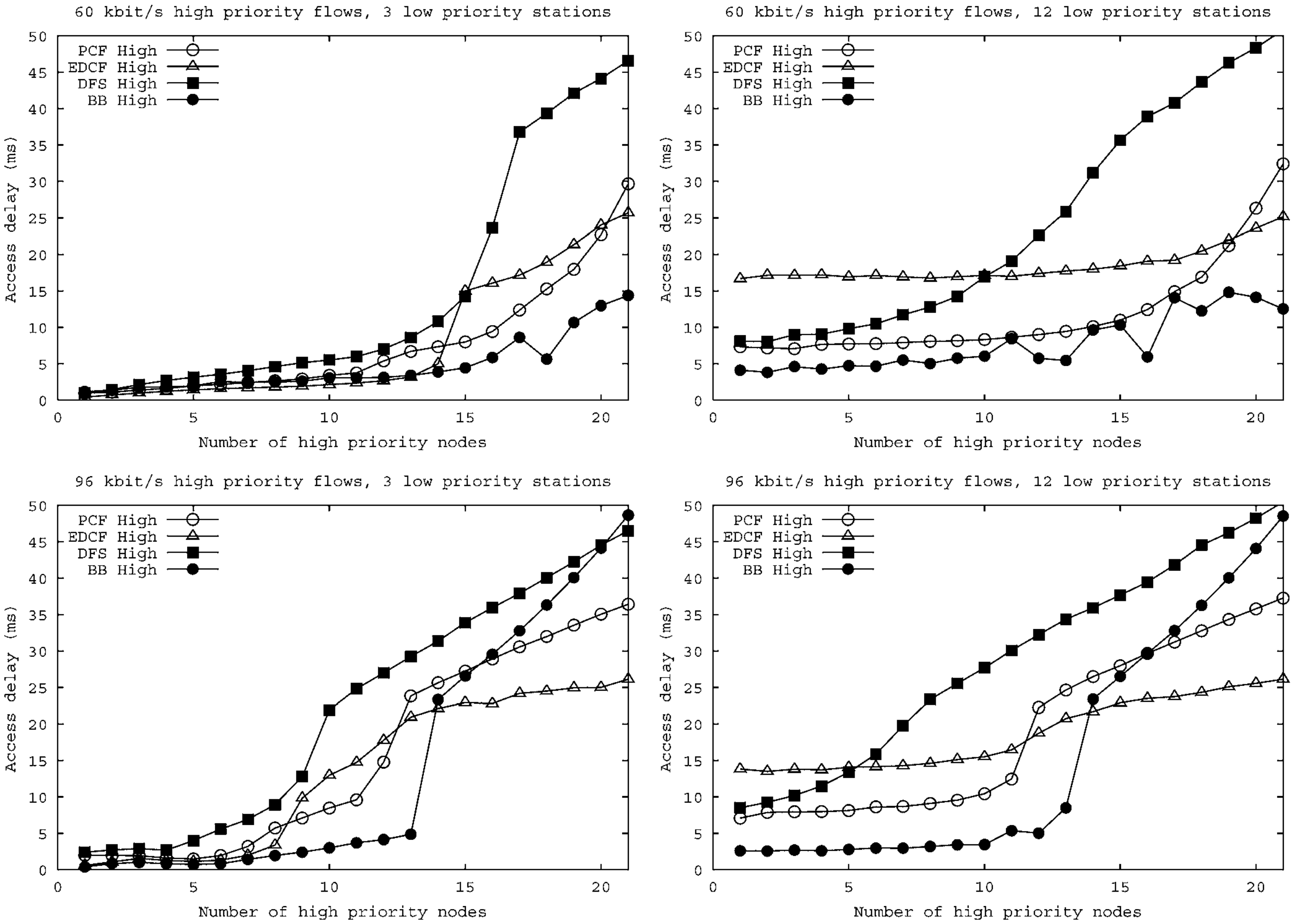


Figure 8. Average access delay for high priority stations.

4.2.2. Medium utilization

Figure 6 shows the medium utilization of the different schemes. Note that for the graphs where the low priority load is low, the utilization first increases linearly up until a certain level, because there is no more data to send at those loads. Thus, the interesting part of the graph is after the slope of the curve starts to decline. We can see that Blackburst has significantly higher medium utilization than the other schemes except for at very high loads. At times it differs almost 10% between Blackburst and the scheme that is closest. One remarkable thing is the rather low utilization of EDCF at higher loads. The packet bursting scheme used by EDCF was introduced to enhance the medium utilization over ordinary DCF. However, looking at the collision rates for the schemes shown in figure 7 we see that EDCF still suffers from a large number of collisions, especially at high loads. This explains why the medium utilization of EDCF is comparably low since much time is wasted on collisions. We can also see that Blackburst has an extremely low number of collisions. This confirms that Blackburst is able to completely avoid collisions between high priority stations since the collision rate goes down to zero at the point where the low priority traffic is starved (in figure 5), so the collisions seen prior to that point must be caused by low priority stations accessing the medium. It is interesting that the medium utilization of Blackburst drops dramatically when the load gets very high. To find out the cause of this, we studied how much time that is "wasted" on blackbursting instead of sending useful data. In figure 10 we can see how large fraction of the time that was used for black burst in the different scenarios. We can see that for very high loads, the amount of time where the medium is jammed by blackbursting stations increases rapidly to a high level, explaining the decrease of medium utilization at the same point. For comparison, we also measured the amount of time used by PCF to send control frames. Figure 10 shows the overhead caused by these transmissions. The amount of overhead could be expected to be a function of the superframe size (how often we send beacon and poll frames), and the number of high priority stations (how many poll frames to send during each contention-free period). Since we only have one superframe size, the impact of that cannot be seen in the graph, but it clearly shows that the overhead increases with the number of stations to be polled. Thus, the overhead is not largely affected by the amount of low priority station or the bit rate of high priority traffic. However, already at fairly low loads the transmission of control frames occupy over 10% of the time (and channel bandwidth).

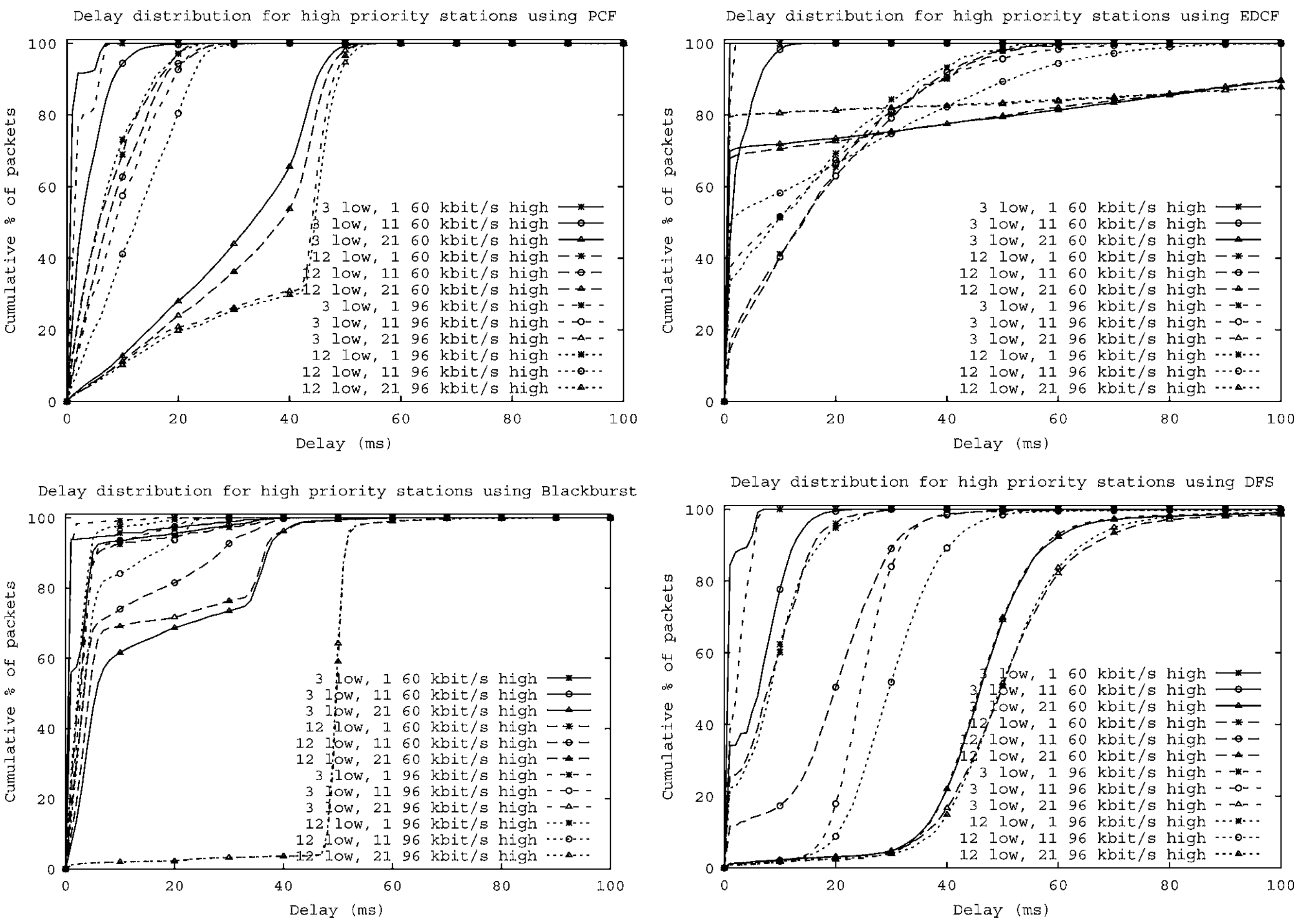


Figure 9. Cumulative delay distribution for high priority traffic.

4.2.3. Access delay

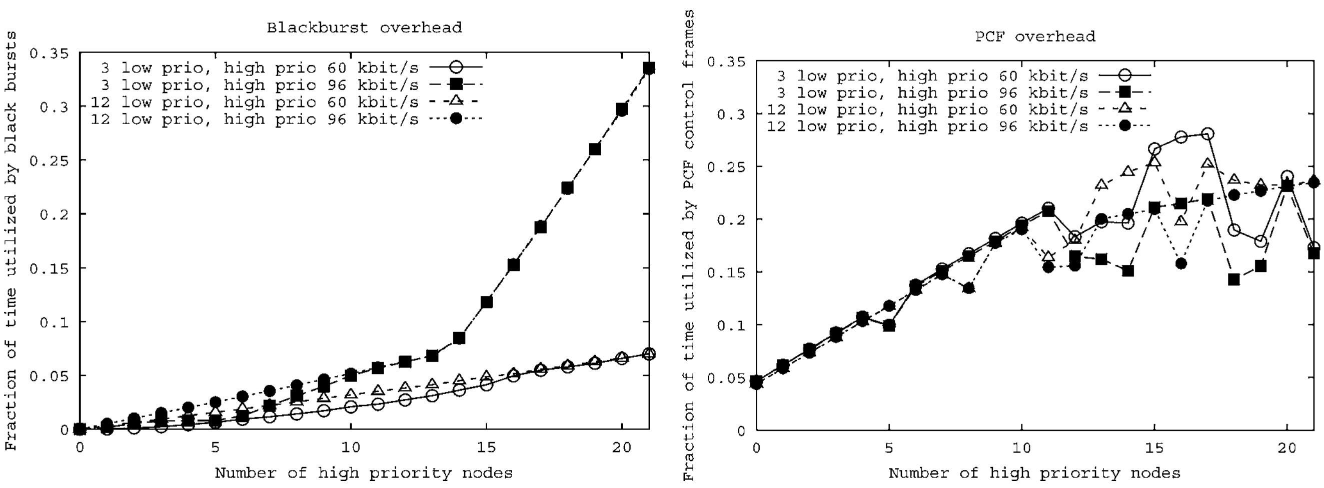
Investigating our third metric, average access delay for high priority traffic, figure 8 shows that Blackburst has very low delays in most cases, even though the delays increases when the load gets very high. However, all the schemes have acceptable delays, even though DFS in most cases incur a longer delay than the other schemes.

Even if a scheme can give low *average* access delay to high priority traffic, there might still be many packets that get rather high delays, rendering them useless to a time critical application (for example, voice over IP). Therefore, it is interesting to investigate how the delays of the packets are distributed. In figure 9 we plot the cumulative percentage of packets that have an access delay below certain values up to 100 ms[4] for the different schemes and for three choices of number of high priority stations for each low priority load, and high priority bit-rate.

The distribution of the access delay for PCF is quite good. At all loads, all packets have a rather acceptable access delay. It is however interesting to see that when there are many high priority stations, a large part (around 70%) of the packets have an access delay of approximately two superframe lengths. This indicates that the load is too high for the Point Coordinator to be able to poll all high priority nodes during one superframe, so many nodes has to wait more than one superframe before being polled.

When investigating EDCF, it can be seen that at low loads all delays are very low, and there is not much jitter. At "medium" load, the delay starts to spread over a larger range, but still the delay has a upper bound around 50 ms, which should be acceptable. When studying the distribution of the delay for the scenarios with high load, some interesting things can be noted. A very large part of the packets (up to 80%) has very low delays (below a few milliseconds), while the rest of the packets have rather high delays (over 10% of the packets have delays over 100 ms). This tendency with a large percentage of the packets having very low delay could also be noted, although not to the same extent, at the "medium" levels of load. This behaviour is due to the use of packet bursting in EDCF. The stations need to contend for the medium for the first packet in each packet burst which thus gets high delay (especially since everybody else also is packet bursting, so it will be hard to get access to the medium). However, after getting hold of the medium, packets are transmitted with a very short interval between them, so the access delay for these packets will be minimal.

[4] This limit of 100 ms was chosen because packets with a delay of more than 100 ms would most likely be useless to most real-time applications.



(a) (b)

Figure 10. Overhead incurred by Blackburst and PCF. (a) Blackburst, (b) PCF.

The distribution of Blackburst is as expected. For the two cases with highest load (the ones with 21 high priority stations with 96 kbit/s flows), almost all packets have an access delay around 50 ms, which is rather acceptable at high loads like this. For the cases where the load is a bit lower, we can see that most packets have very low delay, while some have a little longer delay, from contending for the medium with black bursts. The majority of packets will not have to do that since the stations then are "synchronized" and will not have to contend for access to the medium.

Finally, when studying DFS, we see that for all levels of load, the majority of the packets have access delays that are at acceptable levels, and in each case there is an acceptable upper bound that the delays do not exceed (unlike, for example, EDCF where in some cases there were packets with delays above 100 ms). Looking at the shape of the curves, it seems like the access delays come from a normal distribution. This is due to the fact that the packet sizes of the packets sent come from a normal distribution, and the packet sizes are used in the calculation of the backoff intervals. Thus, the delay experienced by the stations also conforms to a normal distribution.

5. Discussion

We have shown that all the schemes are able to provide service differentiation to some extent. However, we have also seen that if too many high priority stations are active, their performance degrade. Thus, it would be desirable to be able to perform admission control when a new station or flow wishes to use the higher priority class. One major advantage that PCF has over the other schemes is that since it is centralized, implementation of admission control would be rather simple [4]. Because of the distributed nature of the other schemes, admission control and service enforcement are harder to realize for those schemes. However, Barry et al. recently presented some interesting work on distributed admission control [1].

One problem with Blackburst and EDCF is that with the settings we have used, they completely starve low priority traffic when the load is increased, which is not desirable. If it is important that low priority traffic is not starved (and if admission control is not available to ensure that there is not too much high priority traffic allowed), DFS might be an alternative to consider. It is also very likely that the traffic class parameters of EDCF could be chosen such that starvation does not occur (for example, by using the same AIFS for all traffic classes and only vary $CW_{\min}$). However, if admission control is used, this will not be a problem since decisions could be made that a certain part of the bandwidth always should be available to low priority traffic.

One thing that might be problematic for Blackburst is that it requires high priority traffic to access the medium at constant intervals, which should be the same for all high priority traffic. However, one possible scenario where Blackburst certainly would provide great benefits would be if, for example, a company would wish to run telephony over the corporate WLAN. Then all the users using the WLAN for telephony would have the same kind of traffic, and all the users using the WLAN for ordinary data transfer would use the low priority (which in the Blackburst case is just normal IEEE 802.11 DCF, meaning that no special precautions have to be taken by those users). In such a scenario, it is likely that special purpose real time applications are used (for example, implemented in hardware in a phone), which opens the possibility to use the Blackburst mode with feedback to the application, probably yielding even better results than those presented in this paper.

If Blackburst could not be used (because of failure to comply with the traffic requirements or lack of the ability to jam the channel), EDCF could be a suitable alternative. It is able to give good differentiation and can give high priority traffic high throughput and low delay. One thing that speaks very strongly in favor of EDCF is that it is a part of the upcoming IEEE 802.11e standard, which means that it is very likely that most WLAN equipment will have EDCF functionality implemented by default, thus simplifying the deployment of the technology.

The packet bursting mechanism of EDCF could be applied to the other schemes as well to enhance their performance. This is something that should be studied in more detail; especially how the packet bursting impacts on real-time applications that are sensitive to jitter.

If the high priority traffic come from time critical real-time applications, the delay – and not least the distribution of it, is crucial. For example, when comparing EDCF and DFS at high loads, we saw in figures 5 and 8 that EDCF has both higher throughput and lower average delay than DFS for high priority traffic. Thus, it would seem like EDCF is the best choice in this case. However, assume that the real-time application the high priority data is meant for cannot use data older than a certain limit, and has to discard such data. If this limit would be, for example, 100 ms, the stations using DFS would deliver all its packets well under this delay, while the use of EDCF would cause 10–15% of the delivered packets to have delays higher than 100 ms, thus being discarded by the receiver (see figure 9). This means that the actual amount of data the application can use is rather similar for both schemes. Still, DFS is able to give much better service to low priority traffic, which EDCF completely starves at these loads, making DFS a better choice in this particular case.

When doing an evaluation like this, different settings used to create the different scenarios of course affect the final result. We have done tests with other settings of packet sizes and inter-packet intervals as well, and the comparative results presented in this paper still holds. Even though the overall performance probably would increase a bit with the use of larger packets with longer intervals between them (since most overhead is on a per-packet basis), we wanted the high priority traffic to model some real-time traffic, meaning that we cannot buffer data too long, and thus we cannot have very large packets.

6. Conclusions

We have evaluated four mechanisms for service differentiation in IEEE 802.11 wireless LANs. Evaluation includes both some tradeoffs in parameter settings for individual schemes, and performance comparisons between the different schemes.

Our simulations show that when using PCF, it is important to select a proper size of the superframe, which determines how often poll frames are sent to high priority stations. To get good performance for high priority traffic, without wasting resources on unnecessary control frames, the superframe should be approximately as long as the interval between packets generated by a high priority station.

When comparing the schemes, our simulations show that Blackburst gives the best performance of the evaluated schemes to high priority traffic both with regard to throughput and access delay. At low loads, it also gives rather good performance to low priority traffic, but it does however deteriorate to complete starvation of low priority traffic at very high loads. A drawback with Blackburst is the requirements of constant access intervals it imposes on high priority traffic. If these cannot be met, EDCF might be a suitable alternative. Although not being able to provide as good service as Blackburst and suffering from a high rate of collisions, it still can serve quite many high priority stations and give very low delay to high priority traffic.

Both Blackburst and EDCF starve low priority traffic at high loads of high priority traffic, which in many cases is not desirable. If relative differentiation is desired, DFS would do a better job. It ensures better service to high priority traffic, and still does not starve low priority traffic (ensuring that it gets its fair share of the bandwidth).

Further, our simulations show that the Blackburst scheme gives the best medium utilization at reasonable loads of high priority traffic. This is important, given the scarcity of bandwidth in wireless networks. We have also shown that Blackburst is very good at avoiding collisions between high priority stations, while EDCF suffers from a high rate of collisions.

Contrary to EDCF and DFS, Blackburst and PCF transmits bursts and control frames on the channel to determine which station should get access to the medium. During those transmissions, the channel is occupied and cannot be used for any useful transmission of data. We have shown that for Blackburst, this overhead is rather low up to a certain point of high priority load, where the amount of overhead increases rapidly. For PCF, the overhead is quite large, and increases with the number of high priority stations.

Finally, we conclude with the observation that it is nontrivial to say that one QoS scheme is better than another, since it largely depends on the context where it is to be used, and which results are desired. We can say that Blackburst works very well in many scenarios, but there certainly exist scenarios where some of the other schemes would be preferable. Before deciding on what QoS scheme to use in a network, an analysis of what the network should be used for, and what kind of services that is needed should be done.

Acknowledgments

We would like to thank João Sobrinho for helping us to gain a better understanding of Blackburst. We would also like to express our deepest gratitude to Nitin H. Vaidya, Paramvir "Victor" Bahl, and Seema Gupta for their benevolence in providing us with their *ns* implementation of DFS, saving us many long hours of implementation work. Greg Chesson has been very helping, providing us with a *ns* implementation of EDCF (without packet bursting), and also answering our questions about IEEE 802.11e.

This research is supported in part by Mäkitalo Research Centre, Vinnova, and EU structural funds.

References

[1] M. Barry, A.T. Campbell and A. Veres, Distributed control algorithms for service differentiation in wireless packet networks, in: *Proceedings of IEEE INFOCOM* (2001).

[2] M. Benveniste, G. Chesson, M. Hoeben, A. Singla, H. Teunissen and M. Wentink, EDCF proposed draft text, IEEE working document 802.11-01/131r1 (2001).

[3] G. Chesson, W. Diepstraten, D. Kitchin, H. Teunissen and M. Wentink, Baseline D-QoS proposal, IEEE 802.11 working document 802.11-00/399 (2000).

[4] C. Coutras, S. Gupta and N.B. Shroff, Scheduling of real-time traffic in IEEE 802.11 wireless LANs, Wireless Networks 6(6) (2000) 457–466.

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

[6] S.J. Golestani, A self-clocked fair queueing scheme for broadband applications, in: *Proceedings of IEEE INFOCOM* (1994).

[7] P. Goyal, H.M. Vin and H. Cheng, Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks, IEEE/ACM Transactions on Networking 5 (1997) 690–704.

[8] IEEE Standard 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications (1999).

[9] A. Lindgren, A. Almquist and O. Schelén, Evaluation of Quality of Service schemes for IEEE 802.11 wireless LANs, in: *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)* (2001).

[10] A. Lindgren, A. Almquist and O. Schelén, Quality of Service schemes for IEEE 802.11 – A simulation study, in: *Proceedings of the Ninth International Workshop on Quality of Service (IWQoS 2001)* (2001).

[11] S. McCanne and S. Floyd, *ns* Network Simulator version 2.1b8 (2001).

[12] J.L. Sobrinho and A.S. Krishnakumar, Real-time traffic over the IEEE 802.11 Medium Access Control layer, Bell Labs Technical Journal (1996) 172–187.

[13] J.L. Sobrinho and A.S. Krishnakumar, Quality-of-Service in ad hoc carrier sense multiple access networks, IEEE Journal on Selected Areas in Communications 17(8) (1999) 1353–1368.

[14] N.H. Vaidya, P. Bahl and S. Gupta, Distributed fair scheduling in a wireless LAN, in: *Sixth Annual International Conference on Mobile Computing and Networking*, Boston (2000).

[15] M.A. Visser and M. El Zarki, Voice and data transmission over an 802.11 wireless network, in: *Proceedings of PIMRC'95*, Toronto, Canada (1995) pp. 648–652.

Anders Lindgren received his M.Sc. degree in computer science and electrical engineering from Luleå University of Technology in 2001. He is currently a Ph.D. student at the Division of Computer Science and Networking at the Department of Computer Science and Electrical Engineering at Luleå University of Technology. His research interests include Quality of Service in wireless networks, ad hoc networking, and delay tolerant networks.
E-mail: dugdale@sm.luth.se
WWW: http://www.sm.luth.se/∼dugdale

Andreas Almquist is project manager and research engineer at Telia Research. He has a Masters degree in computer science, specialized in computer communication, at Luleå University of Technology. He is currently managing several projects examining future markets for Virtual Private Network (VPN) services and developing strategies and architectures to meet future customer needs.
E-mail: andreas.k.almquist@telia.se

Olov Schelén is a faculty member of the Computer Science and Networking group at Luleå University of Technology. He is also CTO and co-founder of Operax AB. He received his Ph.D. in 1998. R&D interests are network architectures and protocols for multimedia networking, including quality of service, routing, wireless communication and mobility.
E-mail: olov@sm.luth.se