An Enhancement to the IEEE 802.11e EDCA Providing QoS Guarantees

Hamidian, Ali; Körner, Ulf

Published in:
Telecommunication Systems

DOI:
10.1007/s11235-006-6520-z

2006

Telecommun Syst (2006) 31: 195–212
DOI 10.1007/s11235-006-6520-z

An enhancement to the IEEE 802.11e EDCA providing QoS guarantees

Ali Hamidian · Ulf Körner

Abstract One of the challenges that must be overcome to realize the practical benefits of ad hoc networks is quality of service (QoS). However, the IEEE 802.11 standard, which undeniably is the most widespread wireless technology of choice for WLANs and ad hoc networks, does not address this issue. In order to support applications with QoS requirements, the upcoming IEEE 802.11e standard enhances the original IEEE 802.11 MAC protocol by introducing a new coordination function which has both contention-based and contention-free medium access methods. In this paper, we consider the contention-based medium access method, the EDCA, and propose an extension to it such that it can be used to provide QoS guarantees in WLANs operating in ad hoc mode. Our solution is fully distributed, uses admission control to regulate the usage of resources and gives stations with high-priority traffic streams an opportunity to reserve time for collision-free access to the medium.

Keywords QoS – IEEE 802.11e – distributed MAC · Admission control and scheduling

1. Introduction

Today, *wireless local area networks* (WLANs) are deployed in many universities, homes, cafés, train stations, airports and even in airplanes. The key for these networks to become even more useful and popular is, among other things, to support applications with *quality of service* (QoS) requirements, such as video, audio, voice over IP and other multimedia applications. However, the original 802.11 standard [1] has not addressed the QoS issues sufficiently. The later 802.11a/b/g standards offer only higher maximum data rates by enhancing the *physical layer* (PHY) of the original standard – they all use the same medium access method that does not support QoS. Therefore, the 802.11 working group formed task group E (802.11e) to address the QoS issues in the *medium access control* (MAC) layer.

A. Hamidian · U. Körner
Department of Communication Systems, Lund University, Sweden, Box 118, 221 00 Lund
e-mail: {alexh, ulfk}@telecom.lth.se

The upcoming 802.11e standard [2] defines a new coordination function, called *hybrid coordination function* (HCF), that includes two medium access methods: *enhanced distributed channel access* (EDCA) and HCF *controlled channel access* (HCCA). The EDCA is distributed and contention-based, making it suitable for ad hoc networks while the HCCA is centralized[1] and contention-free – thus, it cannot be used in infrastructure-less networks. One drawback with the EDCA compared to the HCCA is that a station cannot reserve the medium and access it without needing to contend for it; instead the station must contend for access to the medium by possibly starting a backoff timer of random length. Moreover, there is no distributed admission control algorithm. Therefore, it is not possible to provide QoS guarantees using the EDCA [3].

There has been a lot of research on QoS for WLANs and ad hoc networks. Among these studies, many have focused on a solution at the MAC sublayer. This is explained by the fact that QoS provisioning is not possible unless supported by the MAC protocol. In other words, a QoS-aware MAC protocol is necessary but perhaps not sufficient (at least for multi-hop ad hoc networks where e.g. a QoS-aware routing protocol is required to find a route, satisfying the QoS requirements, between a source and a destination). Most of these works have been made for infrastructure-based networks. Among the studies focusing on the MAC sublayer and infrastructure-less networks, a few propose a solution that is compatible with 802.11. It is our belief that, since the 802.11 standard is so widely spread, any realistic proposal must be compatible with this technology.

The *fuzzy CW allocation control* (FCWAC) scheme is proposed in [4]. It aims to solve the problems occurring due to a) the doubling of the *contention window* (CW) after each unsuccessful transmission and b) the resetting of the CW to *CWmin* after each successful transmission. The first event results in very large delays after just a few unsuccessful retransmissions while the second event may result in a so called *bursty collision* since the current network conditions are not taken into consideration. Instead, after each collision or successful data transmission, FCWAC dynamically adjusts the CW based on the current queue length, loss probability and packet waiting time. Although FCWAC tries to improve the EDCA, it is still based on a random waiting time before accessing the medium and it does not have any distributed admission control algorithm. In other words, it is not possible to guarantee QoS because the scheme is based on service differentiation just as the EDCA.

In [5], the *Simple Scheduler*, the *Grilo Scheduler* [6] and an extension to the the Grilo Scheduler are presented. The Simple Scheduler is an example of a scheduler mentioned in [2] where each station can schedule medium time of fixed length at constant intervals. The Grilo Scheduler has extended this functionality by allowing the stations to schedule medium time of variable length at different intervals for each station. One drawback in the Simple Scheduler that has not been considered by the Grilo Scheduler is that the duration of the medium time to be reserved is calculated based on average traffic rates or estimations. In [5] the calculation is instead based on actual requirements; this is achieved by using two fields with information about the queue size and the requested duration of the medium time to be reserved. The proposed enhancement can be used in our scheme instead of the Simple Scheduler given as an example in [2].

Distributed MAC schemes based on 802.11 and designed for providing QoS are studied in [7]. The schemes are classified into priority-based and fair-scheduling-based approaches. The priority-based schemes, like the EDCA, provide service differentiation by allowing faster access to the channel to traffic classes with higher priority. The authors do not consider these schemes in their simulations since they are unfair: as the number of high-priority streams increase, they tend to grab the channel, preventing fair access for low-priority streams. Thus, the authors make a

[1] The HCCA manages access to the medium using a QoS access point (QAP).

simple comparison between the three approaches using fair scheduling: *distributed weighted fair queuing* (DWFQ) [8,9], *distributed fair scheduling* (DFS) [10] and their own proposal *distributed deficit round robin* (DDRR) [11], which is based on the concept of *deficit round robin* (DRR) (ref. 18 in [7]). In DDRR, each traffic class determines its allotted *service quantum rate* based on its throughput requirements and maintains a deficit counter of accumulated quanta. The deficit counter is decreased by the size of the transmitted frame and a traffic class can transmit only when the counter is positive. As the authors themselves note, the proposed mechanisms only provide service differentiation; none of them can guarantee QoS since they do not have any mechanism for admission control or resource allocation.

An Admission Control and Dynamic Bandwidth Management scheme is proposed in [12] to provide fairness and soft guarantees in single-hop ad hoc networks. The main piece of the scheme is a centralized (but wireless) *Bandwidth Manager* (BM) used to dynamically allot each stream a share of the channel, depending on the requirements of the stream relative to the requirements of other streams in the network. The BM is also in charge for the admission control management. The authors consider a single-hop ad hoc network as we do, but their proposal can only give soft QoS guarantees and, in addition, it relies on a central station (the BM) and operates at the application layer.

In this paper we consider single-hop ad hoc networks, i.e. WLANs operating in ad hoc mode. One advantage of such networks, instead of the ones operating in the traditional infrastructure mode, is that all frames do not need to pass through an *access point* (AP) waisting bandwidth and making the communication inefficient. If an AP is used as an intermediary, direct one-hop transmissions needlessly become two-hop transmissions [12]. Moreover, the AP is a single point of failure and can thus cause the whole network to fail. Instead of relaying peer-to-peer transmissions between stations within the wireless network, the AP should be used only as a gateway toward the wired Internet. Hence, we propose an extension to 802.11e, which provides QoS guarantees in single-hop ad hoc networks by making use of the advantages of the HCCA and integrate them into the EDCA. Our solution is fully distributed and gives the stations with high-priority traffic an opportunity to reserve medium time.

It is worth mentioning that, when talking about QoS guarantees, we must keep in mind that since a wireless medium is much more unpredictable and error-prone than a wired medium, QoS cannot be guaranteed as in a wired system, especially in unlicensed spectra. However, it is possible to provide techniques that increase the probability that certain traffic classes get adequate QoS and that can provide QoS guarantees in controlled environments [2].

The remainder of this paper is organized as follows: Section 2 gives an overview of the original 802.11 and its QoS limitations. Moreover, it describes the 802.11e draft with focus on the EDCA. Our proposed solution is presented in Section 3. The simulation results are presented and discussed in Section 4. Finally, Section 5 concludes this paper and gives some directions for future work.

2. IEEE 802.11 and IEEE 802.11e

Since the 802.11 standard did not address the QoS issues sufficiently, the 802.11 working group formed task group E. However, although 802.11e is an important enhancement of 802.11, its contention-based medium access method only provides service differentiation and hence, there is no way to provide QoS guarantees in networks independent of any centralized devices.

Table 1 MAC parameters for 802.11b PHY (DSSS)

SIFS	PIFS	DIFS	SlotTime	CWmin	CWmax
10 μs	30 μs	50 μs	20 μs	31	1023

2.1. IEEE 802.11 MAC and its QoS limitations

The 802.11 standard [1] has defined two medium access methods: the *distributed coordination function* (DCF) and the *point coordination function* (PCF). DCF provides a best effort data service and is mandatory while PCF is optional and provides a time-bounded service. For these access methods, four different parameters are used for controlling the waiting time before medium access.

Short interframe space (SIFS): The shortest waiting time, and thus the highest priority for medium access. The SIFS is used by short control messages, such as *clear to send* (CTS) frames, *acknowledgment* (ACK) frames, or polling responses.

PCF interframe space (PIFS): A waiting time longer than SIFS but shorter than DIFS (and thus a medium priority). The PIFS is used only by stations operating under PCF, e.g. by the AP polling other stations.

DCF interframe space (DIFS): A waiting time longer than both SIFS and PIFS and thus the lowest priority for medium access. The DIFS is used only by stations operating under DCF transmitting data frames or management frames.

Extended interframe space (EIFS): The longest waiting time used by stations operating under DCF, but only when a transmission failure occurs. A station that receives an incorrect frame must wait for EIFS before starting its transmission in order to give other stations enough time to acknowledge the frame that the station received incorrectly.

The parameters *SIFS* and *SlotTime* are fixed per physical layer whereas DIFS and PIFS are derived from these two parameters. Table 1 shows these and some other MAC parameters (explained below) specified for 802.11b PHY [13]; *direct sequence spread spectrum* (DSSS).

2.1.1. The coordination functions of 802.11 – DCF and PCF

DCF is based on *carrier sense multiple access with collision avoidance* (CSMA/CA) which works as follows. If the medium is determined to be idle for at least the duration of DIFS, a station can start transmitting. If the medium is determined to be busy, a station defers its transmission until the end of the ongoing transmission. After deferral, the station selects a random backoff time as follows:

$$\text{backoff time} = \text{random()} \times \text{SlotTime}$$

where random() is a uniformly distributed integer in the interval [0,CW] and CW is an integer between *CWmin* and *CWmax*, i.e., CWmin $\leq$ CW $\leq$ CWmax (see Table 1).

A station performing the backoff procedure uses the carrier-sense mechanism to determine whether the medium is busy each time slot. As long as the medium is sensed to be idle for the duration of a time slot, the backoff procedure decrements its backoff time by a slot time.

Whenever the medium is determined to be busy, the backoff procedure is suspended; that is, the backoff timer does not decrement for that slot. The medium shall be sensed to be idle for the duration of DIFS before the backoff procedure is allowed to resume. Once the backoff timer expires the station begins transmitting.

If two or more stations start transmitting at the same time a collision will occur. In this case, the CW is doubled and a new backoff procedure is started. The CW starts with CWmin and doubles up to a maximum of CWmax. Once it reaches CWmax, the CW maintains that value until it is reset. The CW shall be reset to CWmin after each successful attempt to transmit a frame. This process will continue until the transmission is successful or discarded.

DCF cannot guarantee a maximum access delay or minimum transmission bandwidth. To provide time-bounded service such as voice, audio or video, PCF has been specified. PCF is dependent on DCF and can thus not be used alone. Moreover, it requires an access point that controls the medium access and polls the stations; therefore it is only usable on infrastructure network configurations, i.e., infrastructure-less networks cannot use this function.

This access method uses a *point coordinator* (PC), which operates at the access point, to determine which station has the right to transmit. PCF is actually a polling medium access method with the PC performing the role of the polling master. The PC maintains a polling list of registered stations and polls each station one by one according to the list. No station is allowed to transmit unless it is polled, and stations receive data from the access point only when they are polled.

2.1.2. QoS limitations of DCF and PCF

Some applications, such as data, audio and video, have different requirements in data rate, delay and jitter. However, in DCF all stations and data flows have the same priority to access the medium, i.e. in a first come first serve, best effort manner. Thus, there is no way to guarantee QoS, that is, there is no differentiation mechanism to guarantee data rate, delay or jitter for applications which are sensitive to these parameters.

Although PCF was specified to provide a time-bounded service, this access method has a few problems which leads to poor QoS performance: a) the lack of possibility for stations to communicate QoS requirements to the access point makes it hard to optimize the polling algorithm performance in the PC; b) the unpredictable beacon delays result in shortened *contention-free period* (CFP) and c) the transmission time of the polled stations is unknown, which makes it hard for the PC to predict and control the polling schedule for the remainder of the CFP. Therefore, PCF does not fulfill its task despite the fact that it uses an access point controlling access to the medium.

2.2. IEEE 802.11e

In order to solve the above-mentioned problems with PCF and DCF, the upcoming standard 802.11e [2] defines a new coordination function: *hybrid coordination function* (HCF). HCF has both contention-based and contention-free (controlled) medium access methods in a single medium access protocol, which explains why it is called *hybrid* coordination function.

The contention-based medium access method is called *enhanced distributed channel access* (EDCA) and provides prioritized QoS support by delivering traffic based on differentiating user priorities.

The controlled medium access method is called *HCF controlled channel access* (HCCA) and provides support for parameterized QoS by allowing for the reservation of transmission time. The HCCA manages access to the medium using a hybrid coordinator operating at a *QoS access point* (QAP).

In HCF, the concept of *transmission opportunity* (TXOP) is introduced. A TXOP is a bounded time interval, defined by a starting time and a maximum duration, that specifies when a station has the right to initiate transmissions to the wireless medium. During this time interval, a station

Table 2 Default EDCA parameter set for 802.11b PHY (DSSS)

AC	CWmin	CWmax	AIFSN	TXOP limit
AC_BK	31	1023	7	0
AC_BE	31	1023	3	0
AC_VI	15	31	2	6.016ms
AC_VO	7	15	2	3.264ms

can transmit multiple frames if the duration of the transmissions does not extend beyond the maximum duration. If a frame is too large to be transmitted in a single TXOP, it should be fragmented into smaller frames.

2.2.1. Enhanced distributed channel access (EDCA)

In the EDCA every station has four transmission queues, or *access categories* (ACs), where each behaves like a virtual station. The four ACs are AC_BK (for background traffic), AC_BE (for best effort traffic), AC_VI (for video traffic) and AC_VO (for voice traffic). Thus, as opposed to DCF where all traffic shared a common queue, in the EDCA each traffic type is queued in the appropriate AC. By varying the following parameters for a specific AC, a differentiated medium access is realized:

- the length of the contention window to be used for the backoff
- the amount of time a station has to defer before backoff or transmission
- the duration a station may transmit after medium is accessed

Table 2 shows how this medium access differentiation is achieved by assigning certain parameters (explained below) different values.

The parameters *CWmin* and *CWmax* are the minimum and maximum value of the contention window. The contention window is used to calculate the number of time slots to backoff before accessing the medium. By assigning low values to CWmin and CWmax, we can give the AC a higher priority.

The *arbitration interframe space number* (AIFSN) is the number of time slots after a SIFS duration a station has to defer before either invoking a backoff or starting a transmission. AIFSN affects the *arbitration interframe space* (AIFS), which specifies the duration (in time instead of number of time slots) a station must defer before backoff or transmission. Thus, by assigning a low value to AIFSN, we give the AC a high priority. AIFS can be derived from the relation

$$AIFS[AC] = SIFS + AIFSN[AC] \times \text{SlotTime}$$

The parameter *TXOP limit* specifies the length (or maximum duration) of the TXOP. A TXOP limit higher than zero means that a station[2] can transmit multiple frames as long as the duration of the transmissions does not extend beyond the TXOP limit. A TXOP limit value of zero indicates that only one data or management frame (plus a possible RTS/CTS exchange) can be sent. Thus, by assigning a high value to the TXOP limit, we give the AC a high priority.

Each AC contends independently for TXOPs based on the parameters described above. Once the AC has sensed the medium idle for at least the duration of AIFS[AC], it starts its backoff timer. If two or more ACs within a single station get ready for transmitting at the same time slot, an internal collision occurs. The collision is resolved within the station such that the data

[2] In fact it is better to say an AC because during a TXOP, a station is not permitted to send frames from other ACs than the one that won the TXOP, even though there is time left in the TXOP.

frames from the higher-priority AC receive the TXOP and the data frames from the lower-priority colliding AC(s) behave as if there were an external collision on the wireless medium.

3. Proposed approach

Although the distributed EDCA is an important enhancement of DCF, it is not enough to provide QoS guarantees due to its non-deterministic nature where stations use a backoff timer of random length to contend for access to the medium. Moreover, the EDCA does not have any distributed admission control, but the administration of the admission control is done at the QAP. On the other hand, the centralized HCCA, where e.g. a QAP controls the medium access and allows for TXOP reservations, cannot be used in networks independent of any centralized infrastructure.

Therefore, the purpose of our solution is to provide QoS guarantees in a infrastructure-less WLAN by transferring the best techniques from the HCCA and integrate them with the EDCA. In other words, our goal is to distribute the admission control and the scheduling algorithms and enhance every station in the distributed network with the QoS capabilities of a QAP.

3.1. Traffic specification

The *traffic specification* (TSPEC) element contains information about the characteristics and QoS expectation of a traffic stream by specifying a set of parameters such as mean data rate, nominal frame size, service start time, maximum service interval, burst size, delay bound and medium time. The parameter *service start time* specifies the time when the service period starts, i.e. when the station expects to be ready to send frames and *maximum service interval* specifies the maximum time interval between the start of two consecutive service periods.

The information contained in the TSPEC helps other stations to schedule the TXOPs effectively. Most of the above-mentioned parameters are typically set according to the requirements from the application while other parameters are generated locally within the MAC.

The TSPEC element is sent within an *add traffic stream* (ADDTS) request frame, which is a management frame with subtype *action* [2].

3.2. Scheduling and admission control

Contention-based medium access is susceptible to significant performance degradation when overloaded. As the network becomes overloaded, the contention windows become large, leading to more time spent in backoff delays rather than sending data. This necessitates some admission-control mechanism to regulate the amount of traffic streams contending for access to the medium.

In 802.11e, the administration of the admission control is done at the hybrid coordinator, which is located in the QAP. The 802.11e draft gives an example of a simple scheduler and an admission control unit but it is possible to modify these to improve the performance [5]. In our solution the scheduler and admission control algorithm are modified and moved from the central QAP to the stations. For scheduling TXOPs for an admitted traffic stream, the scheduler calculates two parameters:

scheduled service interval (SI): the interval between TXOPs, which is the same for all stations. To calculate the SI, the scheduler calculates the minimum *m* of all maximum service intervals for all admitted streams. Then SI equals a value lower than *m* and a submultiple of the beacon interval. SI must be recalculated when a new traffic stream is admitted that has a maximum service interval smaller than the current SI. To improve the performance of the scheduler, it can for example be modified to generate different SIs for different stations [6].

TXOP duration: to calculate the TXOP duration for an admitted stream, the scheduler uses the following parameters: mean data rate (p) and nominal frame size (L) from the TSPEC, the SI as calculated above, physical transmission rate (R), maximum allowable frame size, i.e. 2304 bytes (M) and overhead due to MAC and PHY headers (O). First, the scheduler calculates the number of data frames that arrived at the mean data rate during the SI:

$$N_i = \left\lceil \frac{SI \times \rho_i}{L_i} \right\rceil$$

Then the scheduler calculates the TXOP duration as the maximum of the time to transmit N_i frames at R_i plus overhead and the time to transmit one maximum size data frame at R_i plus overhead:

$$TXOP_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O\right)$$

To improve the performance of the scheduler, it can for example be modified to consider retransmissions while allocating TXOP durations.

Once SI and TXOP duration are calculated, the admission control decision is easy. If there are k admitted streams, a new stream (k + 1) can be admitted if it satisfies the following inequality:

$$TXOP_{k+1} + \sum_{i=1}^{k} TXOP_i \leq SI - T_{CP}$$

where T_{CP} is the duration of the contention period. The last term ensures that some amount of time is saved for contending low-priority streams.

3.3. Resource reservation

As long as there is no station that needs to reserve TXOPs for its high-priority traffic stream, our solution works like the EDCA. Once a station (sender) wishes to send traffic with strict QoS requirements, i.e. a high-priority traffic stream in either AC_VI or AC_VO, it requests admission for its traffic stream. The admission control request is not sent to any central station such as a QAP, but is handled internally within the sender. The sender either admits or rejects the requested traffic stream according to the admission control algorithm described in Section 3.2.

What happens if the traffic stream is rejected is described later, but if the traffic stream is accepted, the sender schedules its traffic by setting the SI and the service start time parameters. The SI is calculated as described previously and the service start time is set to the end time of the last TXOP in a service interval. Figure 1 shows an example where two stations have scheduled their TXOPs and a third station is about to schedule its TXOPs. If there are no TXOPs previously reserved, the service start time can instead be set to any appropriate value, which defines the start of the newly established service interval. Hence, during a service interval, the first part is used as a contention-free period by traffic streams that have reserved TXOPs and the second part is used as a contention period for low-priority streams.

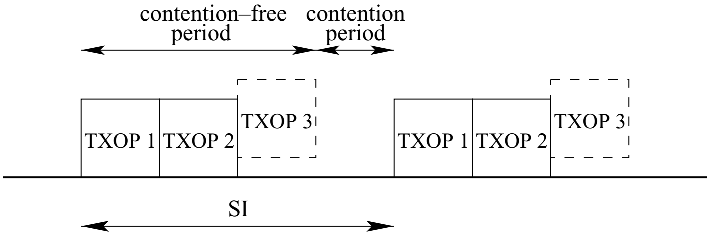

Fig. 1 The scheduling of the reserved TXOPs

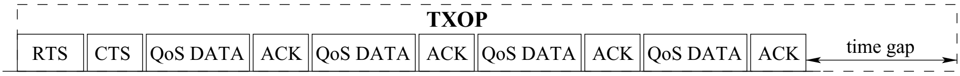

Fig. 2 An example of a frame sequence during a TXOP

Figure 2 shows an example of a frame exchange sequence in which four data frames are transmitted during a TXOP. It is worth mentioning that the sequence starts with an RTS/CTS exchange in order to prevent hidden stations from trying to access the medium while the sender transmits its frames. Thus, the hidden station problem is handled the same way (through RTS/CTS exchange) as in 802.11(e).

In addition, Fig. 2 shows a time gap at the end of the TXOP. These time gaps between two TXOPs are caused by the scheduler's calculation of the duration of the TXOP, which is based on average values rather than exact values. The time gaps are used as an advantage by allowing the stations to transmit important control messages (such as ADDTS request, ADDTS response, messages from the routing protocol, etc.) or low-priority data frames instead of having to wait until the end of the last TXOP in the service interval.

To further increase the performance of the protocol and decrease the time it takes for the sender to reserve TXOPs, an AC (AC_MA) has been added that is used only by management frames, such as ADDTS request and ADDTS response, and routing messages. AC_MA has the same parameters as AC_VO except for TXOP limit, which is set to zero since a station does not need to send multiple management frames in a short time period. The reason behind the choice of AC_MA's parameters is to give the management frames a short waiting time before access to the medium.

Next, the sender broadcasts an ADDTS request containing a TSPEC element with information such as mean data rate, nominal frame size, service start time and SI. All stations that receive the ADDTS request store the information of the sender's service start time and SI and schedule the new traffic stream exactly as the sender. This ensures that no station starts a transmission that cannot finish before a reserved TXOP starts and thus collision-free access to the medium is guaranteed for the streams with reserved TXOPs. All neighbours have to unicast an ADDTS response back to the sender. This is to make sure that the neighbours agree on the schedule and to keep the schedules synchronized.

Every time the sender receives an ADDTS response from a neighbour, it stores the address of the neighbour. After receiving a response from all neighbours, the sender waits until the service start time specified in the TSPEC element and initiates a transmission. If the time instant when all responses are received occurs later than the advertised service start time, the transmission is initiated at the next TXOP instead. During a TXOP, the sender can transmit multiple frames but it must stop sending when the remaining time of the TXOP is less than the transmission time of another data frame plus its corresponding ACK. Once the TXOP is finished, the station waits until the next TXOP, which occurs after an SI. A station that has reserved TXOPs for a traffic stream with strict QoS requirements, is not allowed to transmit frames belonging to that stream at time instants other than during the reserved TXOPs. In other words, the station can transmit frames only at

$$t = \text{service start time} + n \times SI, \quad n = 0, 1, 2 \ldots$$

Of course the station is allowed to transmit frames from other traffic streams, in other ACs, by contending for access to the medium. However, these streams and other low-priority streams from other stations must make sure to finish their transmission before a TXOP starts; otherwise the contending station must backoff and the frames are not allowed to be sent until after the reserved TXOP(s).

When a transmission failure occurs during a TXOP, the station does not start a backoff procedure. Instead, it retransmits the failed frame after SIFS if there is enough time left in the TXOP to complete the transmission.

When a traffic stream finishes and has no more frames to send, it broadcasts a *delete traffic stream* (DELTS) frame notifying other stations to delete the traffic stream and to reschedule the TXOPs of any remaining traffic stream.

If a traffic stream is rejected by the admission control algorithm, the sender can try to lower its QoS demands and retry. The demands should be lowered such that a lower TXOP duration is required. If this compromise is not enough, meaning that no TXOPs can be reserved, the sender has two options left: a) the priority of the traffic stream is lowered such that the stream sends from another AC (with lower priority) that does not require admission control or b) the priority and thus the AC is kept, but the TXOP limit is set to zero. The second option means that, a traffic stream that cannot reserve TXOPs, does not necessarily have to move to another AC with lower priority and longer waiting time before medium access; instead the rejected traffic stream remains in the high-priority AC and contends for access to the medium using the parameters assigned to the high-priority AC, but once it gains access to the medium, it is not allowed to transmit more than one single data frame.

The advantages of this solution are that it is fully distributed, protects against network overload using an admission control algorithm and offers the possibility for stations with high-priority traffic to schedule their traffic in advance such that the QoS requirements of the traffic streams are satisfied. Moreover, since the solution is compatible with the widely used 802.11 standard and based on the upcoming 802.11e standard, it will be possible to integrate it into 802.11e without much difficulties. The proposed mechanism requires modifications only to the software of 802.11 e, i.e. additional hardware is not needed.

4. Evaluation

In order to evaluate the performance of our solution, we have been working on a detailed implementation in the network simulator ns-2 [14]. Since the standard 802.11 implementation in ns-2 is rather simple, we used another more advanced 802.11 implementation [15] for ns-2, which also implements 802.11 a/b/g and some features of 802.11 e. This code was then modified and extended according to our solution described above. The implementation has been used to, by means of simulation, compare our solution against 802.11e concerning QoS guarantees in WLANs independent of centralized devices.

4.1. Simulation scenario

The simulation scenario consists of a number of stations, all within transmission range of each other. The transmission range is 250 meters. The stations use 802.11b DSSS in the physical layer and 802.11e EDCA or our modified version of 802.11e in the MAC sublayer. An error model is used causing 1% of the packets to be damaged. The high-priority streams are assumed to have a maximum delay bound of 10 ms so the parameter SI is 10 ms.

The high-priority streams are sent from AC_VO and use a constant bit rate traffic generator to generate UDP packets with a size of 210 bytes each third ms. The low-priority streams are sent from AC_BE and generate TCP segments according to an FTP application which always has data to transmit.

We have studied both the transient and the stationary behaviour of our solution and compared it toward 802.11e's medium access method EDCA. Regarding the transient behaviour, we study the impact of additional traffic streams on existing traffic by starting the applications at different time instants ten seconds apart. More specifically, we calculated the throughput and end-to-end delay (both at the transport layer) of all traffic streams when additional applications were started. In this scenario, there is one low-priority stream and four high-priority streams. Each traffic stream is sent from a unique source to a unique destination so the total number of stations is ten, i.e. five sources and five destinations. During these simulations, the size of the TCP segments is 210 bytes, i.e. equal to the size of the UDP packets.

Regarding the stationary behaviour, we study the impact of an increasing number of low-priority streams on one high-priority stream. More specifically, we calculated the average end-to-end delay[3], jitter[4] and squared coefficient of variance of the end-to-end delay ($C^2[d]$) for the high-priority stream when the number of low-priority streams was varied between zero and five. Each traffic stream is sent from a unique source to a unique destination so the number of stations is varied between two (one high-priority source-destination pair) and twelve (one high-priority and five low-priority source-destination pairs). The size of the TCP segments was increased to 1000 bytes during the stationary simulations. The reason for this is that we wanted to stress the compared MAC schemes regarding the QoS provisioning to high-priority streams, by increasing the proportion of low-priority traffic load in the network.

For each of the six data points (the number of low-priority streams from zero to five) we ran 150 simulations each during 200 simulated seconds. Then, we calculated the average of the 150 averaged values for each data point and plotted them. Because of the extensive simulations, we could calculate the 99% confidence interval of the average end-to-end delay without getting too large intervals.

4.2. Simulation results

We start by studying the transient behaviour of our scheme compared to the EDCA. There is a low-priority TCP-stream that is started at the 1st second. Then four high-priority UDP-streams are started at the 11th, 21st, 31st and 41st second. All streams continue sending until the 60th second.

[3] The end-to-end delay is calculated as the time when a frame is received by the destination's application layer minus the time when the frame was generated at the application layer at the source.

[4] The jitter is calculated as the variance of the end-to-end delay.

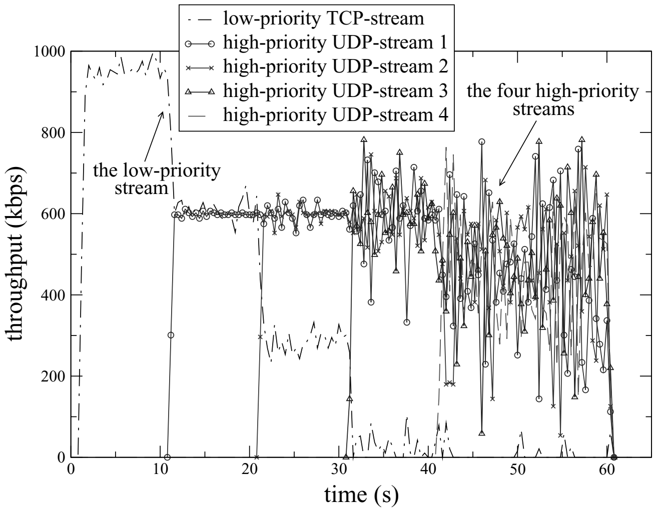

Fig. 3 Throughput – EDCA

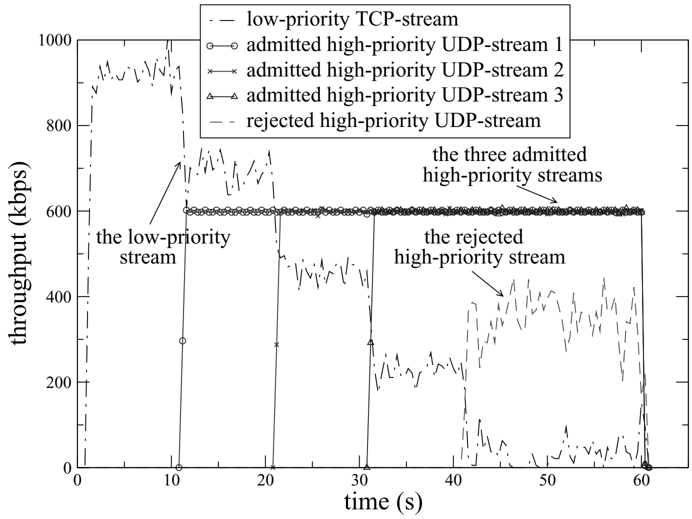

Fig. 4 Throughput – our scheme

Figures 3 and 4 show the throughput of the two schemes at the transport layer. Consequently we must keep in mind that the bit rate on the wireless medium is much higher due to the overhead at the network, data link and physical layer.

Figure 3 shows the case with the EDCA. In the beginning there is a single low-priority TCP-stream, which transmits around 950 kbps. When the first high-priority UDP-stream starts after 11 seconds, the throughput of the low-priority stream drops down to 600 kbps and it continues to fall for each newly started high-priority stream. Once the fourth high-priority stream is started, the throughput of the low-priority stream falls to extremely low levels. Regarding the high-priority streams, it can be seen that the network behaves pretty well until the 31th second when the third high-priority stream is started. From that time on, the throughput of all high-priority streams starts to fall and the variance of the throughput increases drastically.

Figure 4 shows the case with our scheme. Again the simulation starts with a single low-priority TCP-stream, which transmits around 950 kbps. However, in this case the high-priority streams will reserve TXOPs, so if their traffic requests are admitted by the admission control algorithm, they get the amount of bandwidth they require. Thus, we see that the first three high-priority streams have been admitted while the fourth has been rejected. In the case with our scheme, three advantages can be noted compared to the case with the EDCA. First, the low-priority stream has higher throughput due to the fact that the high-priority streams do not collide. Second, the throughput of the admitted high-priority traffic streams is not decreased when new streams are started. Third, the variance of the throughput of the admitted high-priority streams is very low; i.e. the throughput is almost constant around 600 kbps.

For this transient scenario, our simulations also show that the end-to-end delay and the jitter for the three accepted high-priority streams are kept very low with our scheme, independent of the number of existing streams. It is only the rejected high-priority stream that envisions a notable delay and jitter. This is quite in contrast to the EDCA, which shows much larger delay as well as jitter for all high-priority streams.

Next, the stationary behaviour is studied. Figure 5 shows the average end-to-end delay for a high-priority stream when the number of low-priority streams is increased. As expected, the figure shows that using the EDCA, the end-to-end delay increases when the number of low-priority streams is increased. It is worth mentioning that what we see is a typical behaviour of contention-based medium access schemes and it is this kind of behaviour that we want to avoid for streams with strict QoS requirements, where e.g. the delay must be bounded. In addition, we see that the average end-to-end delay is smaller for the EDCA compared to our solution when the number of low-priority streams is very small. This is also expected, since random-access schemes are known to work well in very lightly loaded networks [16] since the medium access time is very low. When the high-priority stream is the only active stream in the network, there is no waiting time at all for the frames. Consequently, the average end-to-end delay of a frame is almost equal to its transmission time plus the time it takes for the stations to process the frame

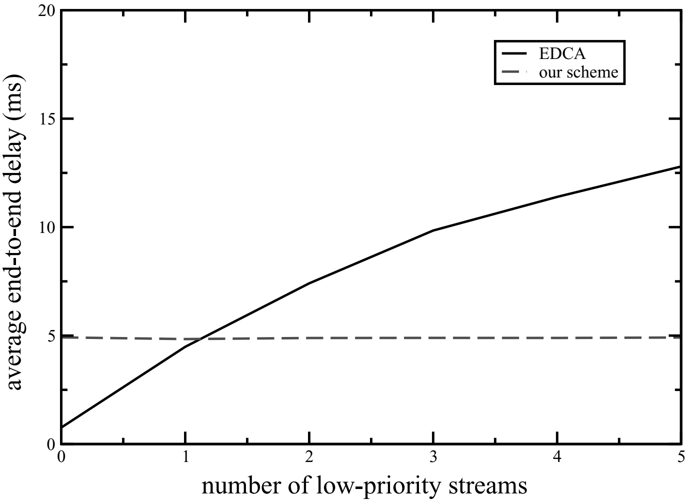

Fig. 5 Impact of low-priority streams on average end-to-end delay for a high-priority stream

Table 3 99% Confidence interval of the average end-to-end delay

Nbr of LP-Streams	Confidence interval (ms)	
	EDCA	Our scheme
0	(0.7609, 0.7612)	(4.8604, 4.9723)
1	(4.4188, 4.5478)	(4.7827, 4.8968)
2	(7.2580, 7.5599)	(4.8319, 4.9426)
3	(9.6543, 10.0295)	(4.8345, 4.9514)
4	(11.2196, 11.5705)	(4.8309, 4.9543)
5	(12.4882, 13.0977)	(4.8502, 4.9727)

(i.e. the time it takes for the sender/receiver to send the packet down/up from/to the transport layer). In fact, if it would not be for the lightly error-prone medium causing retransmission from time to time, the average end-to-end delay of a frame would be exactly equal to its transmission time plus the processing time. Hence, the EDCA is working under ideal conditions when the number of low-priority streams is zero. Using our scheme, on the other hand, the high-priority stream reserves TXOPs and can transmit during the reserved TXOPs only. It can be argued that the high-priority stream reserves TXOPs according to its needs (traffic specification), also under very light loads, and as long as the specification is not violated the goal of the traffic stream is fulfilled. The reservation of TXOPs by the high-priority stream results in guaranteed periodical access to the medium. This explains why the end-to-end delay (around 4.9 ms) is unaffected by the number of low-priority streams, i.e. the end-to-end delay is constant, no matter how much background traffic there is in the network.

In Table 3 we can see the 99% confidence interval for the data points in Figure 5. The table shows that the confidence intervals are pretty small in general. It is worth mentioning that the confidence intervals for the EDCA are bigger than the corresponding intervals for our scheme, except when there are no low-priority streams in the network. In addition, we can see that the confidence intervals for the EDCA increase as the number of low-priority traffic streams increase while this is not the case for our scheme. These observations are explained by the random nature of the EDCA where high-priority streams must contend for access to the medium using the random backoff time resulting in large variances. In our scheme, this randomness is eliminated for the high-priority streams.

Continuing the study of the stationary behaviour, Table 4 shows the jitter and the $C^2[d]$ for the high-priority stream when the number of low-priority streams is increased. We can see that the jitter is constant low for our scheme independent of the number of low-priority streams. This is exactly what we want to achieve with our scheme. For the EDCA, on the other hand, the jitter starts from very low values and increases to high values as the number of low-priority streams increase. This behaviour is not acceptable for multimedia applications with QoS requirements on constant jitter. The reason for why the jitter and the $C^2[d]$ is very low for the EDCA when the high-priority stream is the only active stream in the network, is the same as why the average end-to-end delay is very low for the EDCA; i.e. there is no waiting time for the frames, which leads to the average end-to-end delay of a frame becoming almost equal to its transmission time plus its processing time. As for the $C^2[d]$, the table shows that the $C^2[d]$ is about 6-7 times larger for the EDCA compared to our scheme (except for the case when the number of low-priority streams is zero, i.e., when the EDCA works under ideal conditions).

Table 4 Jitter – our scheme vs. EDCA

Nbr of LP-streams	jitter ($10^{-6}s^2$)		$C^2[d]$	
	EDCA	Our scheme	EDCA	Our scheme
0	0.074	6.6	0.13	0.27
1	37	6.8	1.84	0.29
2	125	7.0	2.28	0.29
3	223	6.9	2.30	0.29
4	275	6.9	2.12	0.29
5	351	6.9	2.14	0.29

5. Conclusion and future work

In this paper, we have presented a distributed MAC scheme based on 802.11e for providing QoS guarantees in WLANs operating in ad hoc mode. One advantage with this solution is that it regulates the medium access with a distributed admission control algorithm. Moreover, there is a resource reservation mechanism allowing the stations wishing to send traffic with strict QoS requirements to reserve TXOPs. These TXOPs are scheduled by a distributed scheduler, ensuring that all neighbours have the same schedule. Once a traffic stream is admitted and the TXOPs are reserved and scheduled, the station does not need to contend for medium access for that traffic stream anymore. The distributed scheduler ensures that no station starts a transmission that cannot finish before a reserved TXOP starts; in other words, a station with reserved TXOPs has collision-free deterministic access to the medium. Since our solution is based on existing commonly used protocols and easy to implement, it is a credible candidate for solving the QoS issues in single-hop ad hoc networks.

Our scheme has been compared to 802.11e's contention-free medium access method, the EDCA, which cannot provide any strict QoS guarantees. Through simulations we have shown that our scheme performs better than the EDCA except when the traffic load is very light. The simulations show that our scheme is able to guarantee constant throughput, delay and jitter to multimedia applications with QoS requirements.

The aim in future work will be to further evaluate and enhance our solution. Regarding the enhancement, we plan to add support for e.g. dynamic resource reservation, retransmitting lost ADDTS requests, removing and rescheduling reserved TXOPs for traffic streams that have completed their transmission and handling stations that move in to and out from the network. Finally, it is our aim to develop the implementation further such that it can be used in a multi-hop ad hoc network and reserve resources along a multi-hop route, perhaps with the aid of a QoS-aware ad hoc routing protocol.

Appendix – A lower bound for the average end-to-end delay

The curves in Fig. 5, i.e. the average packet delay for the EDCA and our MAC scheme, are results of very detailed and comprehensive simulations. Note that the 99% confidence intervals are very small as seen in Table 3. Our simulations were run on a standard 1.7 GHz PC and the values for each point in the curves, i.e. for each value of the low-priority streams, required about 10 hours of simulation. As mentioned, the simulations are built on ns-2 with a detailed 802.1 le implementation for the EDCA [15]. A rough numerical calculation of the presented delays in our proposal validates, at least to some extent, our simulation results for the stationary case.

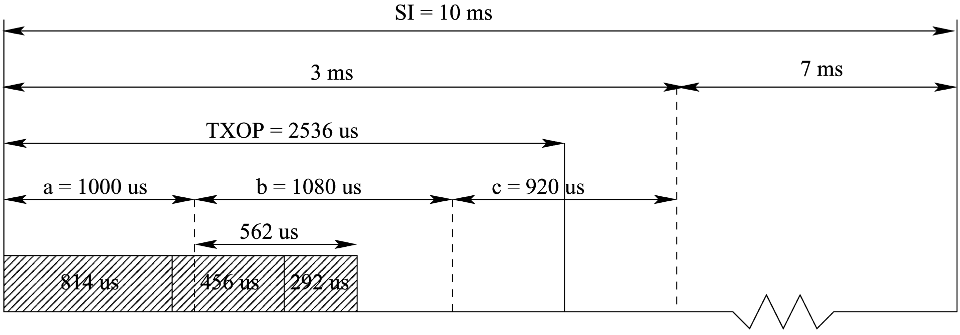

Fig. 6 Time relationships

Regarding our scheme, it is obvious that the end-to-end delay is not affected at all by the number of low-priority streams. As seen from the diagram, the curve is flat slightly below 5 ms. Remember that packets are generated every third ms. The scheduled service interval, SI, is 10 ms and the TXOPs are 2.536 ms long according to the QoS requirements set up by the high-priority source. During each SI, 10/3 packets are generated and thus, on average during each SI 10/3 packets arrive to the MAC sublayer and must be transmitted. Of those packets that are generated during an SI, not more than one may be transmitted during that SI since packets are generated every third ms and the TXOP is just 2.536 ms long.

To calculate the probability that the first packet arriving in an SI (henceforth referred to as $packet_1$), will be transmitted during that SI, we note that it must arrive not only before the TXOP ends but also in time to be transmitted within that TXOP (see Figure 6). The time it takes to transmit $packet_1$ is composed of SIFS (10 μs) + QoS DATA (284 μs) + SIFS (10 μs) + ACK (152 μs) $\Rightarrow$ 456 μs. Hence, the packet must arrive within the first 2536 μs − 456 μs = 2080 μs of the SI. Consequently, the probability that the first packet, generated during an SI, will be transmitted during that SI is 2080/3000. Thus the mean number of packets generated in an SI that also will be sent out in the TXOP in that SI is 2080/3000, i.e. 0.69 packets. This also means that the mean number of packets generated in an SI but sent out in the next SI is $10/3 - 2080/3000 \approx$ 2.64 packets. In other words, the average queue length when a new TXOP starts is 2.64 packets.

If a packet arrives during the first 2080 μs of an ongoing TXOP, its delay depends on where in the TXOP it arrives. The probabilities for where it arrives during those first 2080 μs and the corresponding delays it will experience can be calculated very straightforward. Given this, then it is easy to calculate the delays for the rest of the packets that arrive during that SI. Note that these remaining packets (on average 2.64 packets) will all be transmitted in the TXOP found in the next SI.

Figure 6 shows the usual situation that the first generated packet in an SI is faced with. There are on average 2.64 packets that must be transmitted before $packet_1$ can be transmitted. In the TXOP, the first of these 2.64 packets will be transmitted during the first 814 μs of the TXOP, the second packet during the next 456 μs and the remaining 0.64 packets during the next 0.64*456 = 292 μs. In the figure, the three non-overlapping intervals a, b and c are depicted. If $packet_1$ arrives during interval a, it will be transmitted during the running TXOP and there will be another three arrivals in that SI. If $packet_1$ arrives during interval b, then it will be transmitted during the current TXOP, but there will be just another two arrivals. Finally, if $packet_1$ arrives during interval c, it will be transmitted first in the TXOP of the next SI and furthermore there are another two arrivals in the current SI.

The probability that $packet_1$ arrives in interval x is denoted $P_x (x \in \text{a,b,c})$. Furthermore, let W_x^i denote the average waiting time for the ith arriving packet in that SI, given that $packet_1$ arrives in interval x. Thus,

$$P_a = \frac{1000}{3000} \approx 0.333, \quad P_b = \frac{1080}{3000} \approx 0.360,$$
$$P_c = \frac{920}{3000} \approx 0.307$$

Given an arrival in interval x, the average remaining time of that interval after the arrival is half of the length of the interval. Thus, the average delay of an arrival in interval x consists of the remaining time of interval x (the first term), plus the transmission time of the remaining packets in the queue (any term in the middle), plus its own transmission time (the last term):

$$W_a^1 = \frac{1000}{2} + 562 + 456 = 1518\ \mu s,$$
$$W_b^1 = \frac{562}{1080} \times \frac{562}{2} + 456 \approx 602\ \mu s$$
$$W_c^1 = \frac{920}{2} + 7000 + 814 = 8274\ \mu s$$

Note that those packets that arrive during interval b will experience an average waiting time equal to $562/2\ \mu s$ and that with the probability 562/1080. In addition, note that remaining packet arrivals, on average 2.64, will be transmitted first during the next SI. The second packet arriving exactly $3000\ \mu s$ after the first one, given that the first one arrived in interval x, will experience an average delay equal to:

$$W_a^2 = 7000 - \frac{1000}{2} + 814 = 7314\ \mu s,$$
$$W_b^2 = 7000 - 1000 - \frac{1080}{2} + 814 = 6274\ \mu s$$
$$W_c^2 = W_c^1 - 3000 + 456 = 5730\ \mu s$$

where the last term is the transmission time. Similar reasoning gives the waiting time for the third packet:

$$W_x^3 = W_x^2 - 3000 + 456 = W_x^2 - 2544\ \mu s$$

A fourth packet will arrive during the SI only if $packet_1$ arrived in interval a. Then the waiting time for the fourth packet is:

$$W_a^4 = W_q^3 - 3000 + 456 = 2226\ \mu s$$

The average delay for all packets arriving in one and the same SI given that the first packet arrives in interval x is:

$$W_x = \begin{cases} \frac{1}{4}\sum\limits_{i=1}^{4} W_x^i,\ x = a \\ \frac{1}{3}\sum\limits_{i=1}^{3} W_x^i,\ x = b, c \end{cases}$$

This gives $W_a = 3957\mu s$, $W_b \approx 3535\ \mu s$ and $W_c = 5730\ \mu s$. The total average waiting time W is given by:

$$W = \sum_{x \in a,b,c} P_x \times W_x$$

Straightforward calculations then give $W \approx 4349\ \mu s$. To that comes the time it takes to process the packets, which is 25 μs for both the source and the destination, i.e. 50 μs in total. This gives a lower bound of 4399 μs for the average end-to-end delay, which should be compared to the simulation result of about 4900 μs. The missing microseconds are due to the error-prone wireless medium causing retransmissions and which in turn increase the average end-to-end delay.

References

1. ANSI/IEEE Std 802.11, Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
2. IEEE P802.11e/D10.0, Part11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Medium Access Control (MAC) Quality of Service (QoS) Enhancements, September 2004.
3. A. Shepard: Hybrid Change Makes WLAN QoS Come to Life. http://www.us.designreuse.com/articles/article7742.html, April 2004.
4. I. Hwang and C. Wang, Improving the QoS Performance of EDCA in IEEE 802.11e WLANs Using Fuzzy Set Theory, Active Networking Workshop. 2004.
5. D. Skyrianoglou, N. Passas and A. Salkintzis, Traffic Scheduling in IEEE 802.11e Networks Based on Actual Requirements, Mobile Venue '04 Mobile Location Workshop Athens, May 2004.
6. A. Grilo, M. Macedo and M. Nunes, A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks. IEEE Wireless Communications Magazine, June 2003, pp. 36–43.
7. W. Pattara-Atikom and P. Krishnamurthy, Distributed Mechanisms for Quality of Service in Wireless LANs. IEEE Wireless Communications Magazine, June 2003.
8. A. Branchs, A. and X. Perez, Providing Throughput Guarantees in IEEE 802.11 Wireless LAN, Proc. WCNC, 2002.
9. A. Branchs, A. and X. Perez, Distributed Weighted Fair Queuing in IEEE 802.11 Wireless LAN, Proc. IEEE ICC, 2002.
10. N.H. Vaidya, P. Bahl and S. Gupta, Distributed Fair Scheduling in a Wireless LAN, Proc. ACM MOBICOM, 2000.
11. W. Pattara-Atikom, S. Banerjee and P. Krishnamurthy, Starvation Prevention and Quality of Service in Wireless LANs, Proc. IEEE WPMC, 2002.
12. S.H. Shah, K. Chen and K. Nahrstedt, Dynamic Bandwidth Management in Single-Hop Ad Hoc Wireless Networks, Proc. IEEE Int'l. Conf. Pervasive Comp. and Commun., 2003.
13. IEEE Std 802.1 lb-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, September 2004.
14. S. McCanne and S. Floyd, The Network Simulator – ns-2. www.isi.edu/nsnam/ns/. K. Fall and K. Varadhan, "The ns Manual".
15. M. Moreton: 802.11e patch for ns-2. http://cvs.sourceforge.net/viewcvs.py/ns2-wlanpatch/patch_802_11/
16. A. Muir and J.J. Garcia-Luna-Aceves, Group Allocation Multiple Access in Single-Channel Wireless LANs, Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference, 1997.