RZ 3464 (# 93859) 10/28/02
Electrical Engineering 22 pages

Research Report

Performance Evaluation of the QoS Enhanced IEEE 802.11e MAC Layer

Gianluca Vannuccini

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

Performance Evaluation of the QoS Enhanced IEEE 802.11e MAC Layer

Gianluca Vannuccini
IBM Research, Zurich Research Laboratory
8803 Rüschlikon, Switzerland

Abstract

Wireless Local Area Networks (WLANs) are expected to become one of the most widespread solutions for wireless local access to the Internet, both in public environments as well as in the home-network. Many market forecasts show that WLANs could also overcome $3G$ solutions in the wireless access market, because they provide cheaper and faster access to data networks than cellular wireless systems do. However, a serious limitation to the WLAN success is the fact that they were originally conceived as an extension of the wired-LANs and, therefore, have no differentiation in the offered service. To make WLANs capable of supporting real-time services as well as best-effort data traffic, a large research effort has recently been focused on the optimization of the Quality of Service (QoS)-MAC performance for the upcoming high-data-rate next-generation WLANs. In particular, the IEEE 802.11 Task Group E has been working on a draft proposal for a QoS-aware MAC protocol for the most widespread WLAN technology, namely, IEEE 802.11. This draft considers several service differentiation mechanisms, based on both contention and polling schemes, whose actual effectiveness is still under investigation by the Task Group. In this work we evaluate the performance of the 802.11e MAC by means of computer simulations, in which the specifications given in the draft have been implemented in great detail. To test the QoS capability of the IEEE 802.11e MAC with respect to the actual requirements of the voice, video and high-data-rate applications, real-life scenarios have been defined as test cases, such as home-networking and hot spot environments. Simulation results have been evaluated to define an upper bound of the network capacity in terms of the maximum number of voice and high-data-rate streams that could be fitted in each scenario with the required QoS.

I. INTRODUCTION

In the next few years, high-data-rate Wireless Local Area Networks (WLANs) are expected to become widely deployed in specific environments, such as the *home-networking* marketplace, in which high-quality video and audio devices will be connected through the wireless medium, and the so-called *hot spots*, in which several users will be accessing the same Access Point (AP) to use voice, streaming video, and Web-browsing data services in public areas for information purposes. To this end, the next-generation high-data-rate WLANs will be required to support audio and video real-time services, as well as the typical best-effort data services [1]. However, such networks were developed as an extension of the existing connectionless wired LANs, which were historically designed only for best-effort data. Therefore, many proposals are currently under development to render WLANs capable of supporting real-time multimedia streams. To this end, standardization groups and researchers are working on new proposals to improve the Medium Access Control (MAC) layer performance of the upcoming high-data rate WLANs. The widespread IEEE 802.11 MAC layer considers two different access periods, namely the Contention Free Period (CFP), in which a Point Coordination Function (PCF) supports the access of real-time services to the network with a polling-based access scheme, and the Contention Period (CP), in which a Distributed Coordination Function (DCF) is used for the asynchronous best-effort service access to the channel, according to a contention-based multiple access scheme in which collisions may occur [2]. Although the PCF was conceived for real-time services, it has several drawbacks [3] and, at least to our knowledge, it has never been implemented. Therefore, new enhancements for the IEEE 802.11 MAC are needed to make it capable of supporting QoS. In the literature, two main approaches exist to improve the performance of the IEEE 802.11 MAC, based on service differentiation in the DCF contention-based access scheme, and proposals for more efficient polling schemes similar to, or based on, the PCF principle. In particular, in [4] and [5] the contention access parameters of the IEEE 802.11 DCF are adaptively changed depending on the service requirements and on the network conditions estimated. In these works, the backoff time evaluation, the maximum frame length, and the DCF Inter-Frame Space (DIFS) time are dynamically changed, depending on which class of service a packet belongs to. Moreover, in [4] admission control is also considered in order to optimize the real-time traffic allocation. On the other hand, in [6] and [7] modifications of the IEEE 802.11 PCF are proposed, and various other priority-based polling schemes are considered to support QoS in legacy IEEE 802.11 WLANs. All of these schemes are based on the assumption that the IEEE 802.11 PCF is not an acceptable solution to provide support to real-time services within 802.11 WLANs. At the same time, in the past few years, the IEEE 802.11 Task Group E (TGe) has been working on a proposal for QoS Enhancements to the IEEE 802.11 MAC, namely the IEEE 802.11e proposed standard [8]. This proposal is still far from being a definitive one, and much research is currently being done to define the new IEEE 802.11 QoS-aware MAC behavior. In the meantime, several WLAN technology vendors are starting to build their proprietary solutions to solve the QoS issue [9], because the WLAN marketplace grows very fast, and no standardized solutions will be released in the next few months that will allow WLANs to support real-time traffic. To evaluate the performance of the IEEE 802.11e MAC protocol, we have implemented a system simulator for a WLAN with an 802.11a [10] OFDM-based PHY and an 802.11e MAC layer. To build the system simulator, we have very closely followed the latest 802.11e draft version, namely D3.3 as of October 2002 [8]. Moreover, the starting point of the simulator was the SDL model of the IEEE 802.11 MAC, which was enhanced with the IEEE 802.11e MAC. This model has then been translated into a C++ discrete-event simulator, thus allowing us to obtain computer simulations that are much faster than with SDL and, at the same time, as accurate and close to the draft as SDL modelling allows. The aim of the performance evaluation was to first devise possible and reasonable application scenarios, and then to understand which of the differentiation mechanisms offered by the 802.11e MAC were sufficient to support a large number of services with guaranteed QoS for these scenarios. To this end, we have considered the two above-mentioned scenarios, namely, the hot spot and the home-networking environments, in which the IEEE 802.11e MAC features will most probably be applied. The traffic sources have been chosen according to real-life applications. In particular, voice traffic has been modelled with an ON/OFF source with a 64 kbit/s ITU-T $G.711$ codec [11], which emulates a typical high-quality packet-voice stream among wireless devices, and also an 8 kbit/s $G.729$ codec [12] has been considered, which emulates a lower-quality voice chat on the Internet. The video source has been modelled according to the two video services suggested in the IEEE 802.1D standard [13], also reported in the 802.11e reference scenario settings [8]. In particular, two different streaming video services are defined: a low-quality video service, that can be matched with the Excellent Effort (EE) video, which emulates the so-called *CEOs best effort*, i.e., the best-effort type service that an information services organization would deliver to its most important customers, and a high-quality video, corresponding to the Controlled Load (CL) video, which may require more stringent bandwidth guarantees [13]. Best-effort data traffic has been modelled as a source generating packets with exponentially distributed inter-arrival times and frame length whose distribution has been derived from a real Ethernet network traffic trace. The performance received by these traffic types in the IEEE 802.11e framework has been evaluated by means of computer simulations. In particular, we have evaluated the performance offered by the differentiation of the CWmin, CWmax, the MSDULifeTime, and the AIFS, with regards to the actual QoS requirements of the specific traffic type. The simulation results presented in this paper aim at ultimately evaluating if and when the use of polling-based access is mandatory and for what types of applications, as well as determining in which cases it is sufficient to adopt a simpler, distributed, contention-based differentiation scheme. Note that the delay and delay variation limitations required by many video applications no longer are as stringent as they were before, because these applications are endowed with strong network-adaptive sending and receiving algorithms, which are able to mitigate the effects of network congestion. For such applications the requirement of implementing a polling-based access scheme could be no longer needed. Moreover, when working at high data rates, the amount of time required for MAC and PHY processing at each WSTA is strongly reduced, and the IEEE 802.11a PHY imposes processing time on the order of a few microseconds. The digital processing time limit could be easily reached when considering implementation of the more complex IEEE 802.11e MAC access schemes within digital processors. Although a hardware-based implementation could achieve better performance even at higher data rates, it would require a longer implementation time and less flexible solutions. For this reason, it is important to consider the simplest feasible MAC solution capable of providing acceptable QoS, in order to accelerate the implementation process of IEEE 802.11-related high-performance products. As we will discuss in this paper, simulation results have shown that the simplest contention-based access scheme, with a specific set of differentiation parameters, allows to obtain a large margin of network capacity, while at the same time allowing a feasible implementation of the IEEE 802.11e MAC in the application scenarios considered. In Sec. II, a detailed description of the IEEE 802.11e MAC, as of its latest draft, is presented, and in Sec. III the implementation of this protocol in a C++ discrete-event simulator is extensively discussed. In Sec. IV the network scenarios considered are presented, and the performance evaluation of the MAC protocol is discussed.

II. The IEEE 802.11e MAC Draft Proposal

A. Basic Concepts

The set of Wireless Stations (WSTAs) that make use of the QoS-enhanced MAC protocol defined in the IEEE 802.11e draft is called QoS Basic Service Set (QBSS). The IEEE 802.11e MAC architecture, as shown in Fig. 1, is conceived as an extension of the previous IEEE 802.11 MAC.

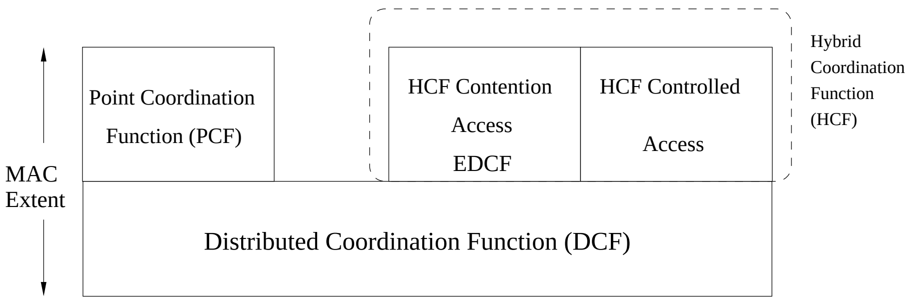

Fig. 1. IEEE 802.11e MAC Architecture.

A Hybrid Coordination Function (HCF) includes two access schemes, namely the contention-based, called Enhanced Distributed Coordination Function (EDCF), and the polling-based access, governed by the Hybrid Coordinator (HC), located at the AP. In contrast to the 802.11 MAC, where DCF operated in the CP, while the PCF operated in the CFP, in the 802.11e MAC the EDCF and the HCF no longer are logically separated, which means that the HCF-polled access can be started by the HC even during EDCF contention-based access. Moreover, the EDCF is defined as a part of the HCF access mechanism.

In order to operate service differentiation, a specific tagging process is defined for frames used in IEEE 802.11e MAC, as shown in Fig. 2. In particular, each MSDU is tagged with a Traffic Identifier (TID), which can take up to 16 values. TID values from 0 to 7 are mapped into Traffic Category Identifiers

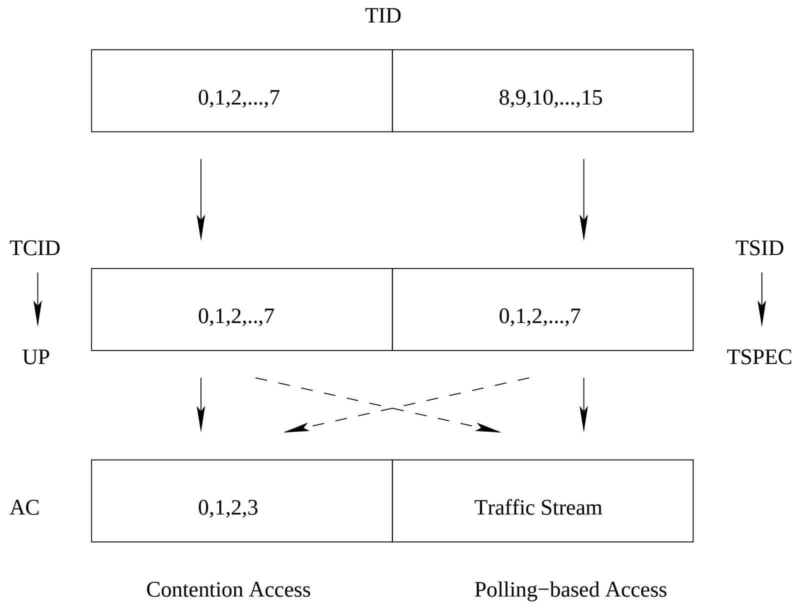

Fig. 2. IEEE 802.11e MAC Traffic Identification.

(TCIDs), which are first mapped $1:1$ into User Priorities (UPs), then are mapped into up to four Access Categories (ACs), according to [8]. TCID and ACs are basically intended as identifiers to differentiate the local treatment that has to be assigned to MSDUs belonging to these categories, within the contention-based process, and thus refer to a connectionless, per-class-based service. On the other hand, TID values from 8 through 15 are mapped into Traffic Stream Identifiers (TSIDs). Each TSID then selects a corresponding Traffic Specification Element (TSPEC). Each TSPEC describes the traffic characteristics of the MSDU with that specific TSID, such as the required bandwidth, delay, and delay variation. For this reason, the access-scheme based on the AC is often referred to as $Prioritized\ QoS$, whereas the access scheme based on TSPEC is called $Parameterized\ QoS$. According to [8], up to 8 traffic streams and 4 ACs can be active in parallel for each wireless link. The possibility to serve ACs with polling and TSPEC with contention-based access (dashed lines in Fig. 2) still is not explicitly denied in the draft, but more likely the access scheme chosen will be the one shown with solid lines in Fig. 2. The HCF Controlled Access scheme aims at providing a certain degree of QoS guarantee, as specified in the TSPEC. However, several issues within the wireless channel and the access network may prevent this target to be achieved, such as unpredictable channel behavior and asynchronous behavior of the users access. Similar concepts of access priority and user priority were already defined in the previous IEEE standards for VLAN tagging and bridge operation, i.e., in IEEE 802.1Q [14] and IEEE 802.1D [13], respectively. Following these standards principles, the latest 802.11e draft proposal contains a UP to AC mapping table, shown in Table I.

TABLE I
USER PRIORITY TO ACCESS CATEGORY MAPPING.

User Priority	Access Category	Designation
1	0	Best-effort
2	0	Best-effort
0	0	Best-effort
3	1	Video Excellent-effort
4	2	Video Controlled Load
5	2	Video
6	3	Voice
7	3	Voice/Network Control

The basic access unit to the channel in the 802.11e MAC is called Transmission Opportunity (TXOP), which replaces the CFP and CP used in the legacy 802.11 MAC. TXOPs can be either gained by WSTAs through contention in the EDCF mode, and are then referred to as EDCF TXOPs, or they can be granted by the HC under the HCF-controlled access operation, and are then called Polled TXOPs. In the following, a more detailed description of the IEEE 802.11e MAC operation is given.

B. EDCF Contention-based Access

EDCF access scheme is governed through a mechanism similar to the 802.11 DCF. In particular, backoff is used by WSTAs to minimize collisions, following a $CSMA/CA$ scheme that includes real and virtual channel sensing, by defining the Network Allocation Vector (NAV). Instead of the legacy 802.11 DIFS, an Arbitration Inter-Frame Space (AIFS) is used for the minimum specified idle duration time, which can be even shorter than the DIFS, to give prioritization to EDCF access over the 802.11 DCF operation. Different from the 802.11 DCF is that the EDCF contention parameters, such as the minimum and maximum values for the Contention Window (CWmin and CWMax) and the AIFSs, have different values for different ACs [8], thus leading to a differentiation in the backoff mechanism as discussed in the Introduction. The above-mentioned contention parameters, together with the TXOP limit parameter, which gives the maximum allowed duration of an EDCF TXOP, are specified in the QoS Parameter Set element, an information object contained in MSDUs exchanged during the EDCF. Only the HC can modify the fields of the QoS Parameter Set element. If the TXOP limit is zero only one MPDU, i.e., one fragment of an MSDU, can be sent during each EDCF TXOP, either with or without the RTS/CTS mechanism. If the TXOP limit is larger than zero, then many MSDU can be fitted into this TXOP, and if an RTS/CTS is needed, then it has to be issued only at the first MSDU of the burst. This procedure is called EDCF TXOP bursting, and its effectiveness is still under evaluation by the TGe. According to [8], when the WSTA, following the successful transmission of an MPDU, retains the right to transmit, it is said to enter into a Continuation TXOP period. As each UP is mapped into an AC, and each AC represents an instance of the EDCF access scheme, then for each WSTA multiple EDCF-based contention schemes can be active in parallel, with different UP and AC values, and thus different contention parameters, leading to internal contention within each WSTA. This multiple internal contention in each WSTA is a relatively new approach to distributed contention-based differentiation, when compared with previous work on the subject [4] [5]. The AC with the lowest values of CWmin, CWMax, and AIFS will experience a lower waiting time during the backoff process. This AC will contend with all other ACs inside the WSTA, as well as with the ACs of other WSTAs, and it will most probably be able to gain an EDCF TXOP before the other ACs running simultaneously within the same WSTA. The length of the TXOP limit will then determine the length of the gained EDCF TXOP, and how many MSDUs, with TIDs mapped into this AC, the WSTA will be able to transmit during the TXOP. According to [8], the MSDULifeTime, namely, the maximum amount of time that an MSDU can wait inside the WSTA buffer before being transmitted, can also assume different values depending on which TID the MSDU belongs to. As it will be shown later, it could be possible to tune this parameter in a useful manner to limit and control the overall transfer time for real-time-traffic MSDUs. Recently, a distributed admission control algorithm has been added to the latest version of the draft [8], which defines specific network occupation timers and transmission budget counters that are used by each WSTA to limit the amount of network occupation for each service. This feature has only recently been introduced, and more details on its implementation will be found in the next draft versions.

C. HCF Controlled Access

The contention mechanism of the EDCF represents a flexible solution for service differentiation. However, for some types of real-time services the contention-based access may not provide the required QoS guarantees. In view of this, the HCF polling-based mechanism has been included in the 802.11e MAC as well as the PCF for the legacy 802.11. The HC is able to start a Controlled Access period (CAP) whenever required, even during an active EDCF period, in order to serve traffic with specific QoS requirements. Moreover, the HCF adds a new access scheme, namely, the Controlled Contention Interval (CCI), in which the HC sends a Controlled Contention (CC) message to a specific subset of WSTAs, thus allowing them to contend, in order to send their polling request, while all other WSTAs are kept silent by means of the NAV setting caused by the CC message. Each WSTA is able to send a polling request to the HC by indicating the TID of the traffic for which the request is being made and either the number of bytes of this traffic type enqueued in the WSTAs or the requested polled TXOP duration. Each WSTA also has the possibility to set up a stream to the HC with a specific QoS that is detailed in the TSPEC element submitted to the HC in order to start the QoS-stream. Many details on the HCF-polling-based access are still under investigation within the TGe, such as the scheduling behavior and its management through proper information elements (the Schedule Element has recently been added to notify WSTAs of the current scheduler status). Many issues are still unsolved for the HCF implementation, such as how the HC should handle the polling of a large number of real-time services without harming the services using EDCF contention. Moreover, the QoS requirements for real-time traffic need a periodic polling, which may no longer be possible when many real-time streams are active in the network, and the polled TXOP duration becomes longer than the inter-poll time. As done for the EDCF, also in the HCF an admission control has been recently introduced in the latest draft version [8], thus allowing the HC to control the network load and to restrict the access to new services if the current network conditions do not allow the requested QoS to be guaranteed. Nevertheless, the effectiveness and feasibility of the admission control in the HC are still under investigation.

III. System Model

In order to evaluate the performance of the IEEE 802.11e MAC layer, a C++ simulator has been implemented, while referring, with as much detail as possible, to the latest version of the draft [8]. The starting point was the SDL model of the IEEE 802.11 MAC layer [2], which was modified to develop the logical structure defined in the 802.11e MAC layer. Next, the SDL model of the IEEE 802.11e MAC was translated into a discrete-event C++ simulator, in order to improve the simulation speed and the flexibility regarding simulating different scenarios, while maintaining a close accordance with the draft specifications. In the following, we will give a brief overview of the implemented simulator, whereas a more detailed description can be found in the Appendix.

A. General overview

The overall system, the first version of which was used in [15], is shown in Fig. 3, and consists of the following functional blocks:

- Simulation Management,
- Traffic Source,
- Terminal,
- MAC,
- PHY,
- Channel.

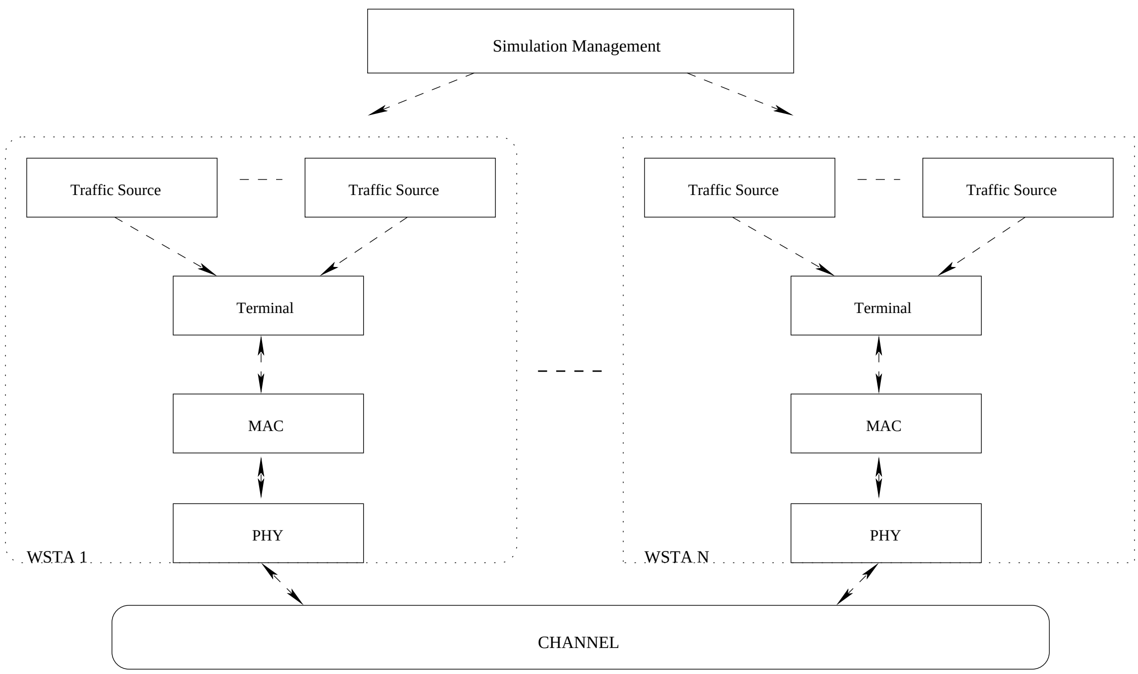

Fig. 3. System Model.

The Simulation Management block is the core of the implemented system, and essentially is a discrete-event simulator, in which the driving events are managed by a system scheduler (see Appendix). It is responsible for creating the objects related to the simulator, namely, one or more Traffic Source blocks, a MAC and PHY block for each WSTA, and the Channel object. Each Traffic Source block, as shown in Fig. 3, is associated with a Terminal block, which can be either of a WSTA type or an AP type. Multiple traffic types can be associated with each Terminal block. Performance measurements are collected for each Terminal block, and then sent to the Simulation Management block, which gathers this information and derives the overall system simulation results for each iteration. We consider an infrastructure-type WLAN; hence all traffic within the QBSS is handled by the AP, no direct links among WSTAs are simulated, and ad-hoc operation has not been implemented.

B. Traffic sources

Traffic sources were selected by carefully taking into account the environment in which the 802.11e MAC might most probably be applied. As many market research studies envisage, the next-generation, high-data-rate WLANs will most probably be employed in two main scenarios, namely the home-networking environment, in which multiple consumer devices will be connected in the home through the WLAN, and the hot spot, where the WLAN will provide mobile users with high-data-rate connectivity in airport lounges, hotels, convention centers, and shopping centers, thus enabling a large-scale spreading of wireless Internet services, such as MPEG streaming, e-mail, Web Browsing, and real-time services like voice and interactive video. In view of this, we have chosen numerical parameters in such a way that the traffic sources, even if built with simple models, reflect as closely as possible the network load conditions of the real-life scenarios. Numerical default values for the traffic sources are given in Table II.

TABLE II

TRAFFIC PARAMETERS.

Symbol	Definition	Default value
Generic System Parameters		
N_v	number of Voice Flows	
N_{lvd}	number of Low-Quality Video Flows	
N_{hvd}	number of High-Quality Video Flows	
N_d	number of Best-Effort Data Flows	
	Each Flow is associated to one link AP-WSTA	
Voice Traffic Parameters		
	Voice-Packet Length (G.711)	160 bytes
	Voice-Packet Length (G.729)	20 bytes
	Voice-Traffic Bit-rate (G.711)	64 kbit/s
	Voice-Traffic Bit-rate (G.729)	8 kbit/s
	RTP/UDP/IP Compressed Header Length	4 bytes
	Voice-Packet max acceptable Delay	20 ms
ϑ_t	Mean Talkspurt Time	1 s
ϑ_s	Mean Silence Time	1.35 s
	Voice TID	6
	Uplink/Downlink Ratio	1/1
High-Quality Video Parameters		
	CBR High-Quality Video Bit-rate	5 Mbit/s
	CBR High-Quality Video Packet Length	1500 byte
	High-Quality Video TID	4
	Uplink/Downlink Ratio	0/1
Low-Quality Video Parameters		
	CBR Low-Quality Video Bit-rate	500 kbit/s
	CBR Low-Quality Video Packet Length	1500 byte
	Video Low-Quality TID	3
	Uplink/Downlink Ratio	0/1
Best-Effort Data Traffic Parameters		
	Mean Packet Length	501 byte
	Offered Load	variable
	Best-Effort TID	0
	Uplink/Downlink Ratio	0.2/1

B.1 Voice Source

Voice signals are known to have a two-state ON/OFF behavior, where speech periods (Talkspurts) are followed by silent (Silence) periods [16]. Many speech codecs have been studied in the literature, and we have chosen two of the most widely deployed ones. The ITU-T $G.711$ codec [11], a 64 kbit/s speech codec, with 160-bytes-long packets generated every 20 ms in the Talkspurt period and no packets generated in the Silence period, is used in good-quality voice calls, even if lower-bit-rate and lower-quality codecs are used when low cost and bandwidth limitation are the issue, such as in the modem-based home access. In such cases, many applications use simpler codecs, such as the ITU $G.729$ [12] 8 kbit/s codec, at the price of a lower quality. The $G.729$ codec selected generates a 20-bytes-long packet every 20 ms. The RTP/UDP/IP header overhead has been considered, but as the packet length in both cases is very limited, header compression has been assumed, which allows us to compress the 40 bytes of the RTP/UDP/IP header into 4 bytes [17]. The duration of the Talkspurt and Silence periods is an exponential random variable with mean equal to 1 s, and 1.35 s, respectively [16]. Unless otherwise stated, in the following performance evaluation we will use $G.711$ as the default voice codec, because it represents the worst-case, having a higher bit rate and packet length. The main performance parameters for voice traffic are voice-packet transfer delay, delay variation and packet loss [18]. In order to preserve the end-user-perceived voice quality, commonly accepted maximum values for these parameters in an end-to-end connection over an IP-based network are 150 ms for the one-way delay, a few milliseconds for the delay variation, and 3% packet loss [18]. However, since the WLAN represents only the last hop of the end-to-end connection, we have chosen more stringent values for the voice transfer delay, delay variation, and packet loss. Accordingly, we have set the voice transfer delay threshold at 20 ms, which can also be used as the IEEE 802.11e MSDU LifeTime value to control voice transfer delay. For the delay variation, comparable or even lower values than the latency will be considered acceptable. Finally, if an MSDU LifeTime is fixed at the terminal buffer for voice packets, after which the packet is discarded, a packet discard rate of 1% is considered as the limit for acceptable voice quality. The voice-packet loss rate is a widely used parameter to evaluate voice performance. However, in order to have a worst-case measurement condition, we decided to measure the percentage of voice packets transmitted with a transfer delay greater than 20 ms, namely $P_{20} = Prob(delay > 20ms)$. In order to measure P_{20}, we have of course removed the MSDU LifeTime differentiation, and thus delayed voice packets were no longer discarded at the transmitting WSTA, but rather their transfer delay was measured at the receiving WSTA. By evaluating the final results, we confirmed that P_{20} is indeed related to the voice-packet discard rate when the MSDU LifeTime is 20 ms. However, we noticed that the voice-packet discarding at the transmitting WSTA resulted in a slight improvement of the voice transfer delay performance measured at the receiving WSTA, because the packets were discarded, which entailed a slight decrease of the network congestion. We then chose to consider the worst-case condition, namely the P_{20} of voice packets, as a voice QoS parameter, while assuming a value of 1% as its limit value for acceptable voice-call quality.

B.2 Video Source

Many video models have been proposed in the literature, each of them belonging to specific services. Again, the possible services and applications in a home-networking environment and a hot spot were investigated. In the former case, inside a home network, we could think of having a low-quality MPEG-like video-downlink streaming, characterized by a relatively low bit rate, such as 500 kbit/s, which could correspond to the Excellent Effort (EE) service specified in [8] [13], and some high-quality video streaming, such as flows connecting DVDs with TVs and other audio or video devices. These high-quality streams could easily attain bit rates of 5 Mbit/s, that therefore should be guaranteed, to avoid a loss of quality, and they could correspond to the Controlled Load (CL) video streams as in [8] and [13]. Such flows are not only downlink (they are often referred to as peer-to-peer communications), because, for example, some DVD players might be connected to the TV via the AP, and in this case we would have one uplink and one downlink flow. Therefore, each high-quality video connection will be hereafter implemented with two simplex AP-WSTA links, one for the uplink and one for the downlink connection. In the hot spot environment, the main video application could be the low-quality MPEG-like video streaming (in this case it could be only downlink, emulating users watching video streams delivered from the fixed network via the AP), say, short advertising spots in a shopping center, or some kind of video trailers in a cinema lounge. The above-mentioned video services can in general be thought of as stored multimedia data delivered to the end user from a storage database, be it a DVD player or a Web Video Content Server. Therefore, we can assume that these flows can be modelled by means of a Constant Bit Rate (CBR) source. Furthermore, we can assume that widely deployed adaptive algorithms are used at the receiving-station video buffer, thus mitigating effects of high delay and jitter values for video packets. For this reason, such flows are not as delay-sensitive as interactive video streams, such as real-time video-conferencing, can be. However, even adaptive algorithm are useless when throughput degradation is concerned. For this reason, when evaluating simulation results, we will focus more strongly on throughput guarantees for video applications, instead of their delay performance, especially for the high-quality video case.

B.3 Data Source

The best-effort data traffic has been modelled with a source generating packets with Poisson-distributed inter-arrival times, and a data packet-length distribution taken from a LAN monitoring trace, with a mean length of 501 bytes. Best-effort offered load has been varied in the different scenarios in order to reach reasonable network occupation levels. To emulate the asymmetrical behavior of Web-browsing-like services, an uplink/downlink ratio of $0.2/1$ has been chosen for the data source.

C. MAC Design

In order to follow the detailed specifications given in the draft D3.3 [8], it was necessary to separate the logical functions of the MAC layer into multiple sub-blocks, while translating the state-machine described in the 802.11 standard for the SDL implementation into a discrete-event C++ simulator for the 802.11e MAC. This made it easier to manage the complex MAC scheme of the IEEE 802.11e MAC. As shown in Fig. 4, the MAC is decomposed into the following parts:

- $Classifier$,
- $TxCoordination$,
- $AccessCategory$,
- $StreamManager$,
- $RxFunctions$,
- $CsFunctions$,
- $PhyAccess$.

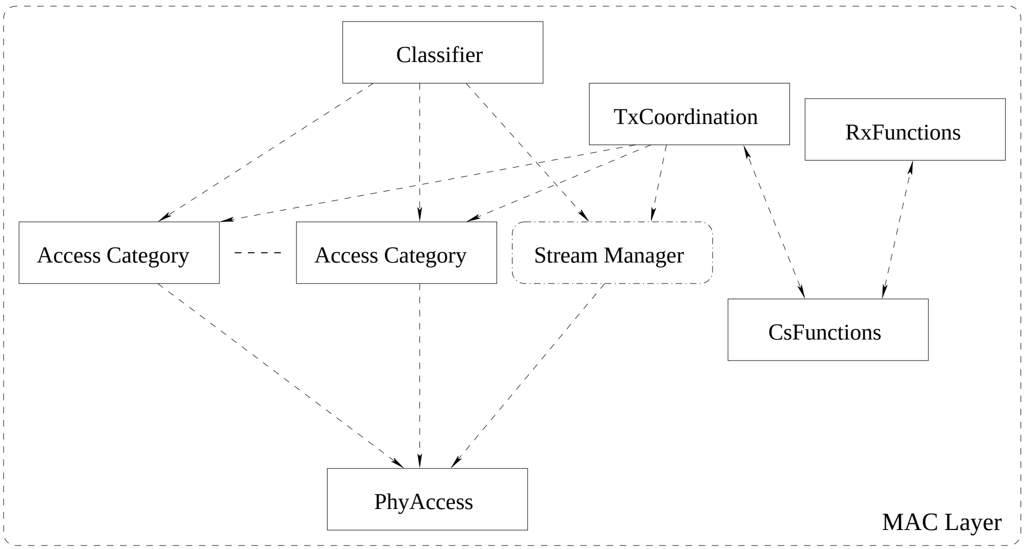

Fig. 4. MAC Layer Model.

The $Classifier$ block receives the MSDUs from the upper layers, reads their TID, and then selects the corresponding access scheme to be associated to each MSDU. If the TID is smaller than 8, a contention-based EDCF scheme is associated with this MSDU, which is then sent to the corresponding $AccessCategory$ for transmission. For each TID, the specific AC associated with an $AccessCategory$ block is derived from the mapping scheme shown in Table I. On the other hand, if the TID is greater than or equal to 8, the MSDU is associated to a Traffic Stream, and therefore is sent to the corresponding $StreamManager$ for a polling-based access scheme. The $StreamManager$ block (and therefore the HCF Controlled Access scheme) has not yet been implemented in the simulator. The $TxCoordination$ is responsible for controlling the multiple $AccessCategory$ and $StreamManager$ blocks that can be activated for each WSTA in the transmitting function. Each $AccessCategory$ installs an EDCF state-machine in the WSTA, and up to four $AccessCategory$ blocks can be active for each WSTA in parallel. The $StreamManager$ coordinates the polling-based access scheme governed by the HCF, and one $StreamManager$ is instantiated for each of the active Traffic Streams. The $RxFunctions$ block is responsible for interpreting the MPDUs received, and for sending the proper status indication messages to the MAC macro-block, confirming a successful reception of each MPDU. The $CsFunctions$ block is responsible for carrier sensing, NAV management and sending Free and Busy signals to the $TxCoordination$ block, which will then provide the proper information to each of the active $AccessCategory$ and $StreamManager$ blocks. Finally, the $PhyAccess$ block manages internal collisions caused by the simultaneous access to the PHY by multiple $AccessCategory$ or $StreamManager$ blocks. This control is necessary because, even if contention is allowed among multiple $AccessCategory$ and $StreamManager$ within the same WSTA, there is only one transceiver for all of them, and thus a proper signalling between these blocks must be conceived to control access to the PHY. The above-described MAC functional scheme is applied to both WSTAs and AP terminals, even if, in the case of polling, the AP MAC would have different functionalities, especially in the $StreamManager$ block. Of the various features considered in the IEEE 802.11e draft [8], there are some that we currently have not implemented in the MAC block shown in Fig. 4, namely, the FEC MAC processing, the EDCF TXOP bursting, the admission control algorithm, and the management part of the MAC. All the other specifications given in the draft for the HCF contention-based access have been carefully taken into account, first in the SDL model and then in the C++ discrete-event simulator.

D. PHY and Channel Model

The PHY and Channel blocks have been taken from [15], where an 802.11a PHY and OFDM channel have been studied in detail. The PHY is therefore compliant with the 802.11a specifications [10] and the channel model has been assumed as frequency-flat and Rayleigh-fading with Jakes' Doppler spectrum [19].

Default values for MAC, PHY and Channel blocks are given in Table III. In particular, for the contention parameters that can take different values depending on the AC, we have used the values suggested in [8], which will be referred to as $default$ differentiation values in the following performance evaluation. As shown in the table, for these parameters, a specific value is given for each of the possible 4 ACs.

TABLE III
MAC, PHY AND CHANNEL DEFAULT VALUES.

Symbol	Defi nition	Default value
MAC Parameters		
	802.11e MAC Header Data Frame Length	30 byte
RTSThreshold		10000
FragmentationThreshold		10000
LongRetryLimit		7
ShortRetryLimit		7
CWMin	Contention Window min	15;15;;7;3
CWMax	Contention Window max	1023;1023;1023;511
AIFS	Arbitrary Inter-Frame Space	2;1;1;1
MSDULifetime	in KUsec (1024*μs)	512;512;512;512
MaxQueuelength		100;100;100;100
Up2acmap	Mapping from UP to AC	0;0;0;1;2;0;3;0
PHY Parameters		
NoiseVariance	noise variance at receiver in dBm	-95.0
CCASensitivity	carrier sensing sensitivity (packets with receive power below this level are ignored)	-98.0
TxMode	Transmission Rate in Mbit/s	M24 and M54
TxPower	maximum transmit power in dBm	40
Channel Parameters		
Radius	cell radius in meters	20 and 50
LossExponent	path loss exponent	3.0
RefLossdB	reference path loss at 1 m (according to Friis equation)	46.7
DopplerSpreadHz	maximum Doppler spread in Hz	5.0
NumberSinus	number of sinewaves to emulate Rayleigh fading	20

IV. Performance Evaluation

As shown, the IEEE 802.11e MAC layer has several parameters that can be tuned to control system performance. However, one first improvement with respect to the other proposed differentiation techniques seems to be the possibility of having multiple EDCF-contention state machines for each WSTA. Therefore, we first investigated the effect, on the QoS parameters, of having multiple ACs and then multiple queues for each WSTA, even without any differentiation in the contention parameters. Several configurations have been tested, such as the ones having 1, 2, 3 and 4 ACs at the AP, then at the WSTA, and then at both of them. We looked at the effect of having multiple queues on voice latency and video throughput. The study of this effect was proven to be not straightforward, due to the fact that congestion can take place mainly in two sites, namely the terminal buffer and the wireless channel. If we manage to reduce congestion in the first one, the other one may be affected, and vice-versa. To this extent, one first consideration that can be done is that, in the cases in which the waiting time is caused by congestion within the terminal buffer (as it is the case for high traffic load at one terminal, such as the AP), then the introduction of multiple queues for the different traffic types at this terminal has the effect of improving the throughput of each traffic type, and moreover of decreasing the transfer delay (i.e. the waiting time). However, while the terminal buffer congestion is decreased, the wireless channel congestion is increased, and if the offered load is more increased at the AP, then the WSTAs will experience a more busy channel, that will affect then their waiting time, even with more queues. Therefore, when the wireless channel gets more congested, either by means of increasing offered load or increasing number of WSTAs, then the introduction of multiple queues might cause higher throughput to be injected into the network from the most loaded terminal (i.e. the AP) than the case with one queue, and then the other WSTAs would get worse performance in terms of transfer delay. Such a situation is partially shown in Figs. 5 and 6, where the voice-packet transfer delay and the video throughput versus the total offered load are shown for a scenario with the following settings: $N_d = 8$, $N_v = 8$, $N_{lvd} = 2$, $N_{hvd} = 1$, with a transmission rate of 24 Mbit/s, and 1 and 4 $AccessCategory$ blocks are active in each of the 8 WSTAs and in the AP. Note that WSTAs always have two ACs fewer than the AP, even with 4 ACs at each side of the link, because in this example video streams are only downlink.

As shown in Fig. 5, the voice transfer delay in the downlink benefits from the introduction of multiple ACs in the AP, whereas no clear effect is seen for the uplink voice delay, which only exhibits a slightly worse performance with 4 ACs rather than with 1 AC under a heavy network load. At the same time, as shown in

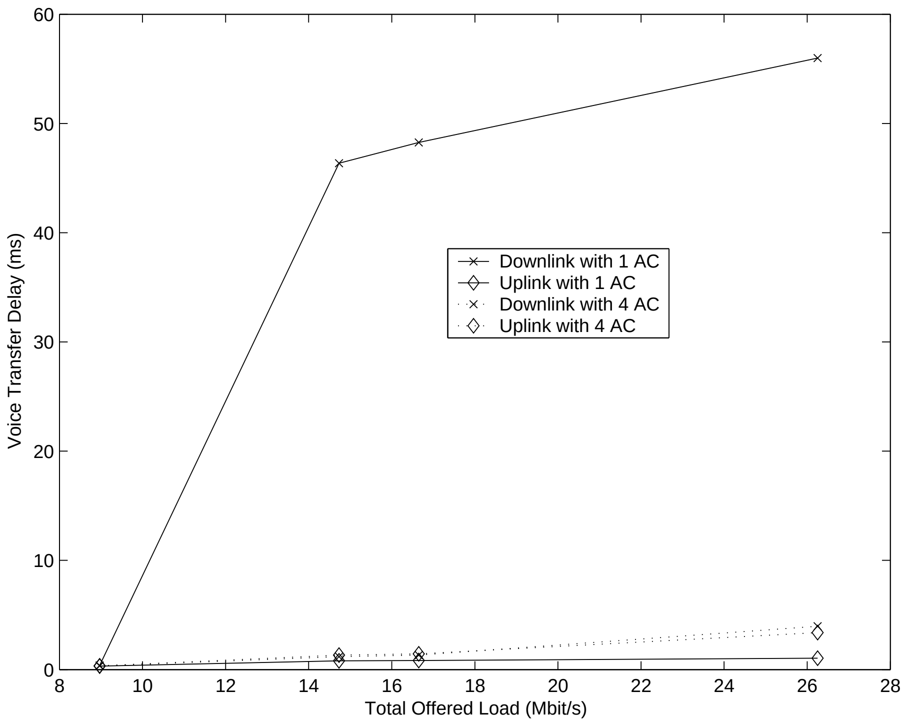

Fig. 5. Effect of having multiple ACs on voice latency (8 WSTAs, $N_v = 8 = N_d$, $N_{lvd} = 2$, $N_{hvd} = 1$, 24 Mbit/s tx rate).

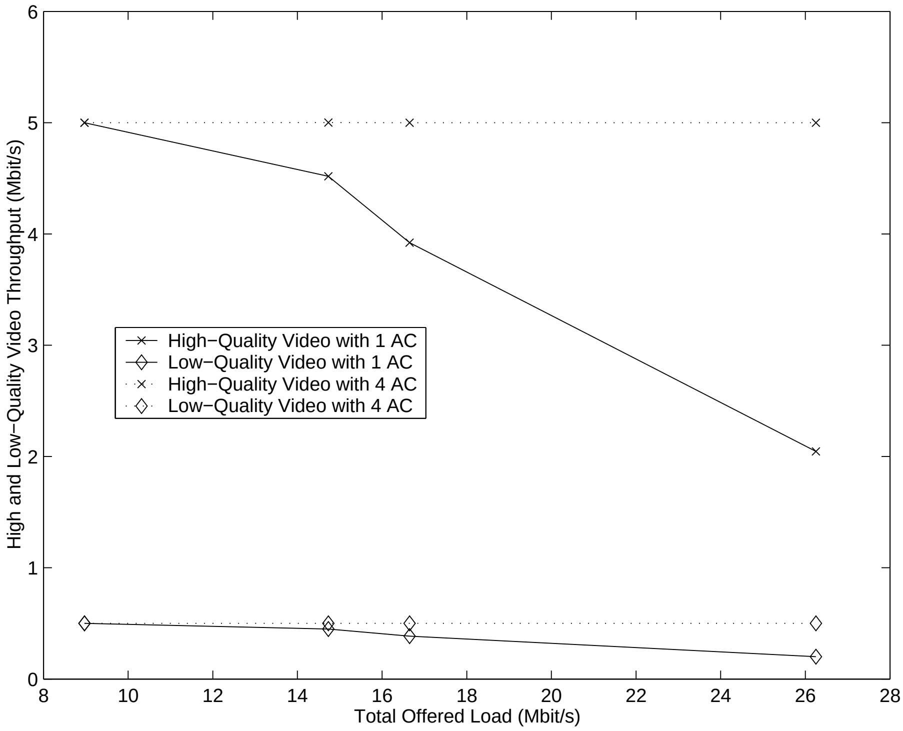

Fig. 6. Effect of having multiple ACs on video throughput (8 WSTAs, $N_v = 8 = N_d$, $N_{lvd} = 2$, $N_{hvd} = 1$, 24 Mbit/s tx rate).

Fig. 6, the downlink video throughput performance always benefits from the introduction of multiple ACs in the AP. This confirms the above considerations, i.e., when multiple ACs are present on a terminal (in this case the AP), ACs belonging to that terminal will on average get more access to the channel, having more EDCF state-machines running in parallel, and then having the possibility to select on average lower backoff times and to access the channel more frequently. On the other hand, the other terminal with less ACs (in this case the WSTA) will experience a busier channel, and will have to wait longer for its traffic to be delivered (hence the slightly higher transfer delay for the uplink case with 4 ACs).

A. Home-Networking Scenario

In order to properly evaluate the effectiveness of the contention parameters of the IEEE 802.11e MAC layer, it is important first to understand what is the aim of such an evaluation. In particular, we want to devise what could be an acceptable scenario for the home-network, and what could be the network capacity target for this case. If we consider a 20 m-radius environment, served with a QBSS, a reasonable performance target could be the allocation of a maximum of 14 WSTAs, supporting a maximum of 14 best-effort flows, 8 low-quality EE MPEG-like 500 kbit/s-streaming videos, 14 voice calls, and the largest possible number of high-quality video streams, representing CL DVD-like video streams. With this target, we have tuned the contention parameters, while referring to [8], where the suggested default parameters shown in Table III have been proposed. In Fig. 7 we show the effect of the default differentiation on P_{20} for voice uplink packets, when the number of voice services in the network is increased, and $N_d = 8$ (for the cases with $N_v \geq 10$ we put $N_d = N_v$ to maintain high network utilization), $N_{lvd} = 4, 8$, $N_{hvd} = 2$ (namely one high-quality video connection between two WSTAs and the AP) and the transmitting rate is 24 Mbit/s. The offered load for each best-effort link is 500 kbit/s. If we fix the acceptable value of P_{20} to 1%, then we conclude from Fig. 7 that the default differentiation is not sufficient to allocate more than 10 voice flows in the home-network with $N_{lvd} = 4$, and more than 5 if $N_{lvd} = 8$. If we want to achieve a larger number of voice flows, we have to select a lower value for the voice TID, namely CWmin[voice]$= 0$, which allows up to 14 voice services to be active in the network with acceptable QoS, even when 8 low-quality MPEG video streams are active. The effect of the increasing number of voice flows on the video throughput is shown in Fig. 8. The performance of the low-quality MPEG-like video is basically not influenced by the presence of the voice services with $N_{lvd} = 4$, whereas it is decreased if $N_{lvd} = 8$ for $N_v > 8$. The high-quality DVD

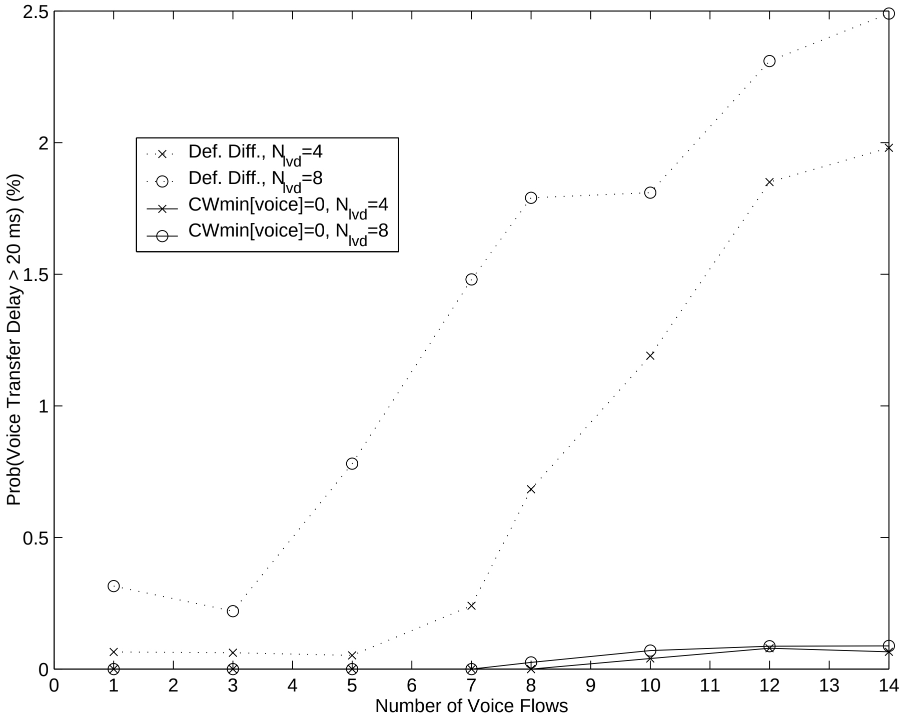

Fig. 7. Home-networking scenario: Voice QoS with default differentiation (10 WSTAs, $N_d = 10$, $N_{lvd} = 4$, $N_{hvd} = 2$ and 24 Mbit/s).

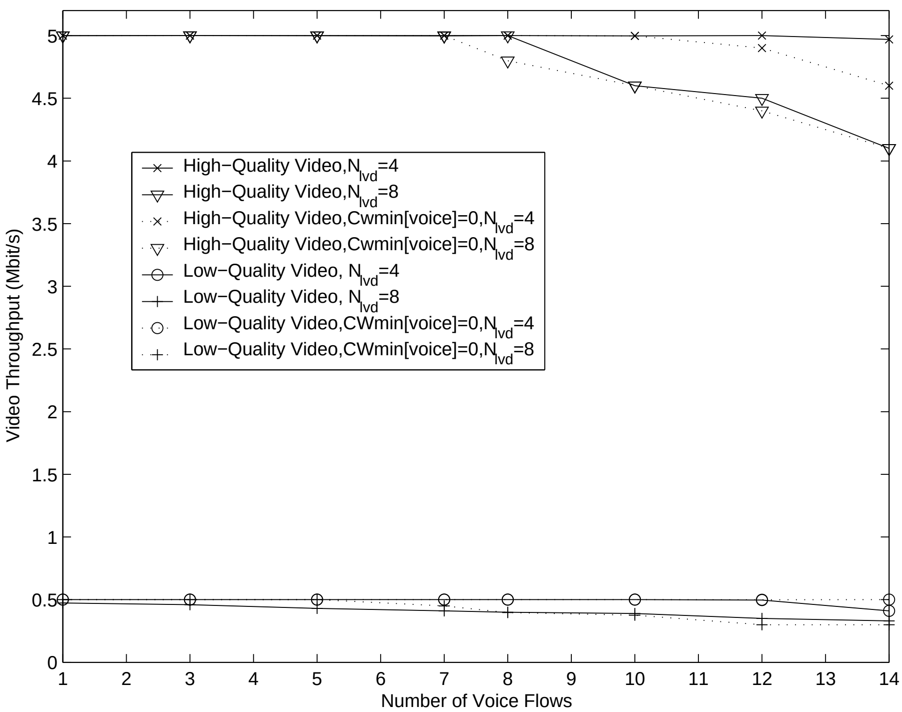

Fig. 8. Home-networking scenario: Video QoS with default differentiation (10 WSTAs, $N_d = 10$, $N_{lvd} = 4$, $N_{hvd} = 2$ and 24 Mbit/s).

video throughput decreases only after 12 voice services have been activated in the network with $N_{lvd} = 4$, and even for $N_{lvd} = 8$ and $N_v = N_d = 14$ it never goes below 4 Mbit/s. Note that, for this last case, the total offered load is about 24.3 Mbit/s; therefore, we far exceed the maximum available throughput for an 802.11a network with 24 Mbit/s transmission rate. We can also note that the use of CWmin[voice]$= 0$ does not considerably affect the performance of video throughput. From the above graphs, we can derive that default differentiation and the use of CWmin[voice]$= 0$ allow an acceptable performance to be reached in the home network for a large number of active services, such as 14 voice calls, 14 data flows, 8 low-quality MPEG streaming, and 1 high-quality video connection among two devices (i.e., 2 high-quality simplex flows with the AP). We can assume that 14 voice calls represents a maximum case for a home-network with a cell radius of 20 m, therefore the considered service differentiation mechanism is well suited to support voice QoS within a home-network scenario, while the video throughput performance still remains at acceptable values. Moreover, results with the $G.729$ codec have shown that, as predictable, even better performance in terms of voice transfer delay and video throughput can be achieved when using lower bit-rate voice traffic, while accepting a lower quality of the voice call. Note that the default differentiation, or even the introduction of CWmin[voice]$= 0$, is relatively simple to implement, compared with other differentiation mechanisms such as TXOP bursting and the polling-based access scheme. However, the use of a larger number of high-quality video flows may be necessary, for example, to connect more devices, such as game consoles, and high-definition video and audio flows. To this end, we have to consider that the above results have been obtained by assuming a transmission rate of 24 Mbit/s. However, the use of the maximum allowed transmission rate for IEEE 802.11a PHY, namely 54 Mbit/s, would allow a larger number of high-data rate flows, while still assuming the differentiation shown in Table III. In Fig. 9 we show the maximum number of high-data rate connections (each composed of one uplink and one downlink high-quality video flow) that can be achieved within a home network with 54 Mbit/s transmitting rate and scenario settings as shown in the figure, while assuming default service differentiation. The video throughput is guaranteed up to 2 high-quality video connections (i.e., $N_{hvd} = 4$, that is, four 5-Mbit/s high-quality simplex links between WSTAs and the AP) if we assume to have 8 low-quality video streaming flows in the same network, and even 3 high-quality video connections could be supported if we accept a degradation of 1 Mbit/s in the high-quality video throughput (see Fig. 9).

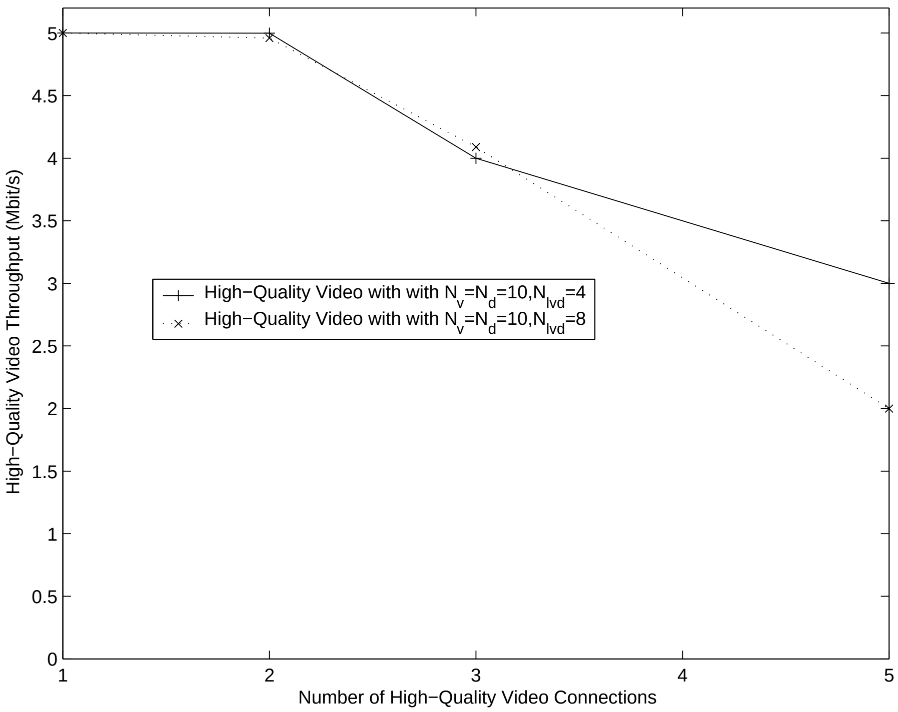

Fig. 9. Home-networking scenario: Maximum number of High-Quality Video connections with 54 Mbit/s transmission rate.

B. Hot Spot Scenario

In order to evaluate the performance for the hot spot scenario, we have followed the same approach as for the home-networking scenario, except that in this case the requirement is no longer to guarantee the throughput to a large number of high-data rate applications, but rather to allocate as many WSTAs as possible within the hot spot with acceptable QoS. The services considered were voice, low-quality video, and data, with the following settings: $N_{lvd} = 8, 15$, $N_d = 30$ (if $N_v \leq 30$, else $N_d = N_v$), best-effort offered load: 350 kbit/s. In particular, the aim here is to support, with guaranteed QoS on voice transfer delay, the largest possible number of voice services within the hot spot, together with a corresponding number of best-effort traffic and a reasonably large number (8 or 15) of low-quality video streams, which could, for example, emulate some downlink MPEG streaming services. To this extent, we have also considered the effect of a lower bit-rate codec voice service, such as the G.729 ITU speech codec, in order to achieve the highest possible network capacity in terms of voice users. As shown in Fig. 10, for a transmission rate of 24 Mbit/s, the use of the default contention parameters shown in Table III is insufficient when more than 30 voice ($G.711$) streams are simultaneously present in the network. However, the use of CWmin[voice]$= 0$ allows an acceptable delay performance to be maintained ($P_{20} < 1\%$) with up to 60 voice ($G.711$) services over 60 best-effort and 15 MPEG, at 24 Mbit/s transmission rate. In Fig. 11 we show the effect of such a differentiation on low-quality MPEG-like video throughput for the same scenario settings, and it is evident that for $N_v \geq 30$ and $N_{lvd} = 15$ the throughput performance decreases, even if it remains in acceptable values if $N_{lvd} = 8$ up to $N_v = 50$. If we accept a lower quality speech codec, such as the $G.729$ codec, we see that up to 80 voice users together with 80 best-effort and 15 MPEG streaming services can be allocated within the network, even for 24 Mbit/s tx rate. However, if we consider the case with 54 Mbit/s, we see from Fig. 10 that 80 higher quality $G.711$ voice flows, $N_d = 80$ and $N_{lvd} = 15$ can be supported in parallel with acceptable voice QoS, if CWmin[voice]$= 0$. In this case, the video throughput performance would still be decreased, as shown in Fig. 11, but only after 60 voice and 60 best-effort streams in the network. However, some degradation in the low-quality video throughput could be accepted in favor of a very large number of voice users within the hot spot. On the other hand, since the high-data rate connections would not be present in the hot spot, in this case it is possible to assign to MSDUs belonging to the low-quality video service a higher AC (namely AC2), in order to give them lower contention parameters (see Table III) and thus better performance in terms of throughput.

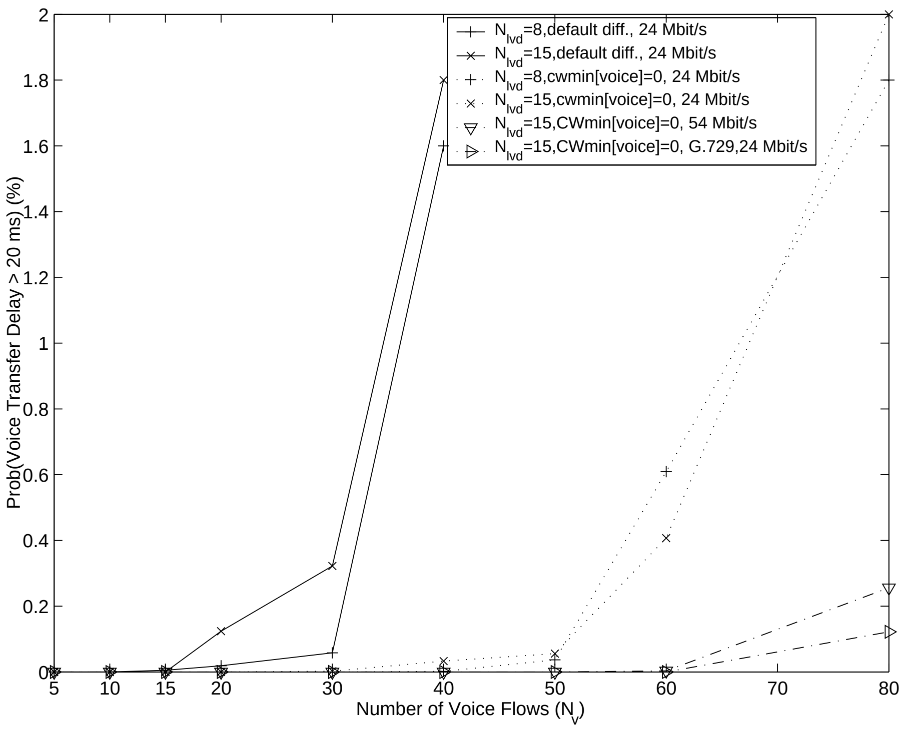

Fig. 10. Hot-spot scenario: Voice Transfer Delay Performance vs number of Voice Flows for 24 Mbit/s transmission rate.

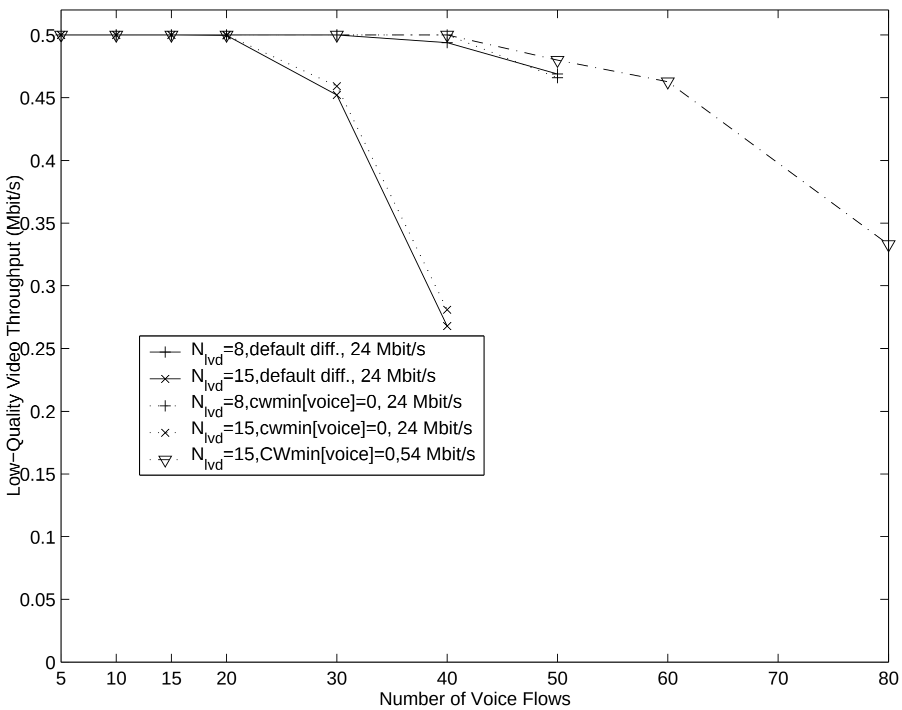

Fig. 11. Hot-spot scenario: Video Throughput Performance vs number of Voice Flows for 24 and 54 Mbit/s transmission rate.

V. Conclusions

In the next few years, WLANs will become a widespread solution in several marketplaces, such as wireless access to the Internet and to information services in hot spots, and the wireless connectivity of high-data-rate devices in the home-networking scenario. In order to achieve these goals, support for real-time services as well as best-effort data must be introduced into WLAN MAC protocols. To this end, the IEEE 802.11e proposed standard could be one possible choice, but several issues must first be solved in order to make this proposal a definitive standard. In this work, we have evaluated the performance of the IEEE 802.11e EDCF MAC access scheme, while taking into account the real size and needs of the possible application scenarios for such a technology. Simulation results have shown that a relatively large number of users and high-data-rate connections can be reached in the hot spot and home-network scenarios, respectively, by using service differentiation provided by the contention-based EDCF scheme. This would lead to a well-performing WLAN without requiring very complex implementation solutions. By assuming that the EDCF differentiation scheme is sufficient to support QoS for an acceptable number of users and services, the standardization process for the IEEE 802.11e could be much accelerated, and many vendors would have the possibility to implement a standard technology, rather than proprietary solutions as they are doing at the moment. As for possible enhancements, we could think of some form of admission control (for example by means of IP-based signalling while activating a new service to the QBSS), which is a currently evaluated solution even in the TGe. Polling-based HCF access could be still maintained as an optional solution, as also proposed in some recent contributions to the standard [20], in order to implement it in those specific cases for which it is required. Finally, it should be noted that possible positive effects coming from the introduction of TXOP bursting on video throughput guarantees (in particular for the home-network scenario) may entail negative effects on voice performance. For this reason, the use of bursting to improve throughput performance may be considered very carefully.

VI. Acknowledgments

The author thanks Pierre Chevillat and Hong Linh Truong for their precious logistic and technical support, André Noll Barreto for his helpful suggestions in the simulator implementation, and Charlotte Bolliger for the paper revision.

APPENDIX

A. Discrete-Event Simulations

In order to study the performance of networking systems, it is necessary to simulate real-life scenarios, in which several users access the network in an asynchronous, random way. This type of access leads to the analysis of systems in which actions are all consequence of discrete events occurring at specific times. Accordingly, the implemented simulator is a discrete-event network simulator in the sense that every action, for instance the activation of a timer, or the action of sending a message after a SIFS time, is implemented in the simulator as a sequence of discrete events. Every event that is activated is subscribed into a system scheduler. The system scheduler is the overall simulator manager, which takes into account all occurring events and which schedules their occurrence at specific times. For example, if we want to start the backoff timer equal to $UBackoff$, we will write the following:

```
 time-to-transmit = now + UBackoff;
schedule(Event(time-to-transmit,transmit());
```

this means that we do not really activate a timer for the backoff, whose maintenance would require to check its value every few milliseconds, but instead we first evaluate when the event following the end of the backoff (in this case the even $transmit$, implemented with the function $transmit()$) will occur, and then schedule this event within $UBackoff + now$. However, following this example for the backoff timer, an event may occur that could freeze the backoff timer, such as the arrival of a $Busy$ message from $Txcoordination$. In this case, the event $transmit()$ must be removed from the scheduler:

```
remove (Event(transmit());
```

B. Class Definitions and Naming Assumptions

The implemented simulator is built in C++ language. Therefore, every event, WSTA, and functional block is defined through classes and derived objects. Each functional block exchanges messages with other blocks, and consequently each class definition contains the messages (i.e., external functions) exchanged with the other objects. For example, if we want to define class $AccessCat$, representing a functional block $AccessCategory$ shown in Fig. 4, we will first define its main structure, namely, its internal methods (functions) and variables, in a class named $AccessCat - private$. Then, the class $AccessCat$, derived from $AccessCat - private$, will be defined in $AccessCat.h$, together with the messages that the class $AccessCat$ exchanges with other objects, and its constructor. At each class, we assume to define the messages (i.e.external functions) that will be sent to (i.e.called) this class by other objects. Finally, $AccessCat.cpp$ will contain the constructor implementation, and the implementation of the messages defined for the class $AccessCat$, for both internal and external functions. For example, a simplified version of the class $AccessCat - private$ is shown below:

```
#include "Scheduler.h"
#include "Packet.h"
#include "log.h"
#include "Terminal.h"

//these are the other classes with which class AccessCat will exchange messages
class Classifier;
class PHY;
class TxCoordination;
class MAC;
class PhyAccess;
////////////////////////////////////////////////////////////////////////////////
// class AccessCat_private                                                    //
//                                                                            //
//  declares private members and functions of class AccessCat                 //
////////////////////////////////////////////////////////////////////////////////
class AccessCat_private {
protected:
    Scheduler* ptr2sch; //pointer to simulation scheduler
    log_file* mylog; //pointer to log file
    MAC_EDCF* mymac; // pointer to MAC object
    PHY* myphy; //pointer to PHY object
    Terminal* term; //pointer to Terminal object
    random* randgen;  // pointer to random number generator

    // Priority-related variables
    unsigned tid; //traffic identifier of the MSDU received by Terminal
    unsigned up; //user priority of the MSDU received by Terminal
    .................
    ................

    //internal functions
    void send_data();
    void send_rts();
```

```
    void ack_timed_out(); //receives an ACK timed out message from Scheduler
    void cts_timed_out(); //receives a CTS-timed out message from Scheduler
   ............
    void transmit(); //transmit an MSDU

};
```

The actual $AccessCat$ class will then be derived from this $AccessCat-private$, and its definition will be given in $AccessCat.h$ as shown below:

```
#include "AccessCat_private.h"

////////////////////////////////////////////////////////////////////////////////
// class AccessCat                                                            //
////////////////////////////////////////////////////////////////////////////////
class AccessCat : AccessCat_private {

public:
//AccessCat class constructor definition

  AccessCat(MAC_EDCF* mac,  // pointer to MAC object containing this AccessCat
            Terminal* t,    // pointer to owner terminal
            Scheduler* s,   // pointer to simulation scheduler
            random* r,      // pointer to random number generator
            log_file* l,    // pointer to log
            Acc_struct acc, // Access Category parameters
            Up_params up2ac,// up2acmapping for this WSTA
            unsigned ac_num,// AC number of the current AccessCat object
            vector<unsigned> ac_up_map //vector containing all
                                       // the ups mapped into this AC
           );

// interface (external) functions

  ...........
  void receive_Free(timestamp time);
  // receive a message from TxCoordination saying that the channel is Free and
  // NAV is zero

  void receive_Busy(timestamp time);
  // receive a message from TxCoordination saying that the channel is Busy and
  // NAV is zero

  .........


};

#endif
```

where the interface functions will be the messages sent from the other objects to this class. Finally, the actual implementation of the $AccessCat$ class will be defined according to the following $AccessCat.cpp$ file sample:

```
#include"AccessCat.h"
#include "Classifier.h"
#include"PhyAccess.h"
#include"Phy.h"
#include"MAC.h"
#include"Profiler.h"
#include"TxCoordination.h"
////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
// class AccessCat                                                            //
////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////
// AccessCat constructor                                                      //
////////////////////////////////////////////////////////////////////////////////

AccessCat::AccessCat (MAC_EDCF* mac, Terminal* t, Scheduler* s,
                    random* r, log_file* l, Acc_struct acc,
                    Up_params up2ac, unsigned ac_num,
                    vector<unsigned> ac_up_map) {
    term = t;
    ptr2sch = s;
    .........
    long_retry_count = 0;
```

```
    short_retry_count =0;
    NAV = 0;
    tStartBkoff = 0;
    uBkOff = 0;
    ........

}


// external functions implementation
////////////////////////////////////////////////////////////////////////////////
// AccessCat_private::send_rts()                                              //
////////////////////////////////////////////////////////////////////////////////
void AccessCat_private::send_rts() {

  ......

  newnav = ptr2sch->now() + rts_duration + cts_duration +
             auxpck.get_duration() + ack_duration(which_mode) +
             3*SIFS;
  pck_rts = MPDU_RTS(term,msdu.get_target(),power_dBm,M6,newnav);
  (mymac->get_phyacc())->phyTxStartReq(pck_rts,true,ac);
  ptr2sch->schedule(Event(t, &wrapper_to_cts_timed_out,this));
}
```

C. Input and Output for the Simulator

The implemented simulator takes the configuration settings from the following $config.txt$ sample file:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation control parameters
MaxSimTime = 30  % maximum simulation time in seconds
TempOutputInterval = 4% interval between temporary outputs
TransientTime = 1 % ignore first TransientTime seconds
VoicePercentile = 27 %in ms
Confidence = .95 % for calculation of confidence interval (if more than one seed), default = .95
%Log = SETUP, TRAFFIC, ACCESSCAT, TXCOORDINATION, CHANNEL, PHY
Log = SETUP  %log simulation events (SETUP,PHY,MAC_DCF,CHANNEL,ADAPT,TRAFFIC, CLASSIFIER,
             % ACCESSCAT, CSFUNCTIONS, TXCOORDINATION, RXFUNCTIONS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% all following parameters accept comma-separated multiple values for several iterations

Seed = 11,23678
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Physical Parameters

NumberAPs = 1 % number of access points
NumberStas = 30% number of mobile terminals
Radius = 50 % cell radius in meters
% if NumberAPs > 1
APPosition_0 = (-50;0) APPosition_1 = (50;0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Channel Parameters
LossExponent = 3.0 % path loss exponent
RefLoss_dB = 46.7  % reference path loss at 1 m (according to Friis equation, Rappaport, pp. 72)
DopplerSpread_Hz = 5.0    % maximum Doppler spread in Hz
NumberSinus = 20 % number of sinewaves to emulate Rayleigh fading.
                 % It should be >10 for good statistical properties.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PHY Parameters
NoiseVariance_dBm = -95.0 % noise variance at receiver in dBm
CCASensitivity_dBm = -98.0 % carrier sensing sensitivity
                           % packets with receive power below this level are ignored
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Traffic scenario settings
%
% TrafficType can be VOICE,POISSON,CBR,VIDEO

TrafficType_0 = VOICE
PacketLength_0 = 164 % 160 bytes is the voice packet length, plus 4 bytes for RTP/UDP/IP compressed header
DataRate_0 = .0656   % offered load in Mbps per link
DownlinkFactor_0 = 1 % offered downstream load is DataRate * DownlinkFactor
UplinkFactor_0 = 1 % offered upstream load is DataRate * UplinkFactor
TID_0 = 6
Flows_0=AP0-MS0/AP0-MS1

TrafficType_1 = CBR %for video a fixed model is used: P-state rate =
PacketLength_1 = 1500
DataRate_1 = 0.5   % offered load in Mbps per link
DownlinkFactor_1 = 1 % offered downstream load is DataRate * DownlinkFactor
UplinkFactor_1 = 0 % offered upstream load is DataRate * UplinkFactor
TID_1 = 3
```

```
Flows_1= AP0-MS0/AP0-MS1/AP0-MS2/AP0-MS3

TrafficType_2 = POISSON
PacketLength_2 = 60(.12);160(.08);180(.23);190(.21);600(.06);1100(.21);1480(.09) % averg length: 501.5 bytes
DataRate_2 = 0.01,.35 % offered load in Mbps per link
DownlinkFactor_2 = 1 % offered downstream load is DataRate * DownlinkFactor
UplinkFactor_2 = 0.2 % offered upstream load is DataRate * UplinkFactor
TID_2 = 0
Flows_2= AP0-MS0/AP0-MS1/AP0-MS2/AP0-MS3


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAC Parameters

WhichMAC = EDCF          % type of MAC protocol: DCF or EDCF (802.11e)
RTSThreshold = 15000%   % packets with more than RTSThreshold bytes employ RTS/CTS
FragmentationThreshold = 100000 % maximum MPDU size, larger packets are fragmented

LongRetryLimit = 7
ShortRetryLimit = 7
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 802.11e MAC Parameters
CWMin = 15;15;15;15;7;15;3;15
CWMax=1023;1023;1023;1023;1023;1023;511;1023
AIFS =2;1;1;1;1;1;1;1
MSDULifetime=58254000;58254000;58254000;58254000;58254000;58254000;58254000;58254000
%in units of time slots, 802.11 states: 512KUsec = 512*1024 microseconds ~ 58254 slot time

MaxQueuelength = 100;100;100;100;100;100;100;100
% Mapping from User Priority to Access Category. Each WSTA can have its own mapping.
% Up2acmap_msi = 0;1;0;1;1;0;0;0  means that WSTA (i+1) has: user priority 0 maps Access Category 0
% user priority 1 maps Access Category 1, user priority 2 maps access category 0.
% WSTA (i+1) will instantiate 2 Access Categories, ac0 and ac1.
%AP
Up2acmap_ap = 0;0;0;1;2;0;3;0
%STA1
Up2acmap_ms0 = 0;0;0;1;2;0;3;0
```

From this script, the Simulation Management block gathers parameter values used to build the specific objects for the various system functional blocks shown in Fig. 3 and Fig. 4. The output of each simulation is then written to a $results.txt$ file, which looks as follows:

```
Iteration 1     Data Rate number:3 = 0.1 , seed = 11
Access Point
0(-50;  0): P(mW)=3211.1
 to Mobile Station 0, TID = 0, tp(Mbps)= 0.095, trt(ms)=    0.40, s_trt(ms)=    0.35, txt(ms)= 0.21, ..
 to Mobile Station 0, TID = 6, tp(Mbps)= 0.029, trt(ms)=    0.22, s_trt(ms)=    0.25, txt(ms)= 0.09,  ..

Mobile Station 0(-10;-12): P(mW)=175.1
 to Access Point 0, TID = 0, tp(Mbps)= 0.019, trt(ms)=    0.44, s_trt(ms)=    0.38, txt(ms)= 0.21, ..
 to Access Point 0, TID = 6, tp(Mbps)= 0.036, trt(ms)=    0.21, s_trt(ms)=    0.20, txt(ms)= 0.10, ..

 Total throughput = 7.26682 Mbps
 Average transfer time = 0.000339901s
 Average transmission time = 0.000184242s
 Packet loss rate = 0
 MSDU life time discard rate = 0
 Overflow rate = 0
 Queue size = 0

Iteration 2
    Data Rate number:3 = 0.1 , seed = 23678

Access Point 0(-50;  0): P(mW)=3216.5
    to Mobile Station 0, TID = 0, tp(Mbps)= 0.096, trt(ms)=    0.40, s_trt(ms)=    0.36, txt(ms)= 0.20,
    ...
   ......
 Total throughput = 7.2672 Mbps
 Average transfer time = 0.000319999s
 Average transmission time = 0.000184483s
 Packet loss rate = 0.000113263
 MSDU life time discard rate = 0
 Overflow rate = 0
 Queue size = 0

%%%% Final results %%%%
Data Rate number:3=       0.1        0.2

Throughput (Mbps) mean =7.253      8.217

conf. interval = 0.101  0.079

Transfer time (ms) mean = 0.336 0.393
```

```
Standard deviation of transfer time (ms) mean       = 0.301 0.400
Transmission time (ms) mean           = 0.189      0.200

Standard deviation of transmission time (ms) mean  = 0.161 0.228
Packet loss rate mean           =      0.000      0.000
conf.interval = 0.000      0.000

Queue size (pckts) mean = 0.000 0.000 .....
```

Basically, for each flow the simulator reports the performance in terms of the various parameters monitored. As shown in the scripts, several iterations can be defined with different random number generator seeds and also different values for the various parameters shown in $config.txt$. This allows the number of configuration scripts to be reduced and facilitates the launch of batch simulations.

D. Traffic Generators

The traffic generators follow the same event-driven approach as described above. Each MSDU generated is an object with specific characteristics, namely, the length, the TID, the source and target address and the packet-generation timestamp. Packet-generator constructors first calculate the inter-arrival time of each packet, and then schedule an event $new - packet$ within this time. Every time the event $new - packet$ is called from the scheduler, a new MSDU object is built, and this MSDU is sent to a function of the Terminal block, which then will forward it to the MAC object associated with this terminal. Finally, the next $new - packet$ event is scheduled. As an example, the generation of a packet belonging to the CBR flow will follow the steps below:

```
1. Calculation of the next MSDU inter-arrival time:
packs_per_sec=data_rate/(packlength * 8.0);
inter_arrival_time=1/packs_per_sec;
next_time_arrival=now+inter_arrival_time;

2. Generation of the MSDU:

MSDU pck(packlength,source,target,tid,time_generated);

3. Transmission of the current packet to the Terminal block:
term->macUnitdataReq(pck);

4. Scheduling of the next MSDU transmission:
ptr2sch->schedule(Event(time_arrival, &wrapper_to_new_packet));
```

REFERENCES

[1] M. Freytes, C.E. Rodriguez, and C.A. Marques, "Real-time H.263+ video transmission on 802.11 wireless LANs," *Proc. Int. Conf. on Information Technology: Coding and Computing, 2001.*, pp. 125–129, 2001.

[2] *IEEE Std 802.11-1999 Information Technology- telecommunications And Information exchange Between Systems-Local And Metropolitan Area Networks-specific Requirements-part 11: Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications*, 1999.

[3] A. Lindgren, A. Almquist, and O. Schelen, "Evaluation of quality of service schemes for IEEE 802.11 Wireless LANs," *Proc. 26th Annual IEEE Conference on Local Computer Networks,(LCN 2001).*, pp. 348–351.

[4] A. Banchs and X. Perez, "Providing throughput guarantees in IEEE 802.11 Wireless LAN," *Proc. IEEE Wireless Communications and Networking Conference, 2002 (WCNC2002).*, pp. 130–138.

[5] I.Aad and C.Castelluccia, "Differentiation Mechanisms for IEEE 802.11," *Proc. of IEEE INFOCOM, 2001.*, pp. 209–218.

[6] T.Suzuki and S.Tasaka, "Performance Evaluation of Priority-based Multimedia Transmission with the PCF in an IEEE 802.11 Standard Wireless LAN," *Proc. of the 12th IEEE Int.l Symp. on Personal, Indoor and Mobile Radio Communications, 2001*, pp. 70–77.

[7] J.Y.Yeh and C.Chen, "Support of Multimedia Services with the IEEE 802.11 MAC protocol," *Proc. of the IEEE Int.l Conf. on Communications, (ICC 2002).*, vol. 1, pp. 600–604.

[8] *IEEE Std 802.11e/D3.3 October 2002 Draft Supplement to IEEE standard for Telecommunications and Information exchange between systems - LANMAN specific requirements. Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)*.

[9] B.Mc Farland, G.Chesson, C.Temme, and T.Meng, "The 5-up protocol for unified multiservice wireless networks," *IEEE Comm. Mag.*, vol. 39, no. 11, pp. 74–80, Nov. 2001.

[10] *IEEE Std 802.11a-1999 Supplement to IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)*, 1999.

[11] *ITU-T Recommendation G.711 (11/88): "Pulse code modulation (PCM) of voice frequencies"*.

[12] *ITU-T Recommendation G.729 (03/96): "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)"*.

[13] *ISO/IEC 15802-3: 1998 ANSI/IEEE Std 802.1D, 1998 Edition - Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Common specifications Part 3: Media Access Control (MAC) Bridges*.

[14] *IEEE Std 802.1Q-1998 - Virtual Bridged Local Area Networks*.

[15] P. Chevillat, J. Jelitto, A. Noll Barreto, and H.L. Truong, "A dynamic link adaptation algorithm for ieee 802.11a wireless lans," *IBM Zurich Research Laboratory, RZ3442*, 2002.

[16] H. Xie and D.J. Goodman, "Prma System Performance with Variable Number of Calls," *Technical Report, WINLAB-TR-36*, Jun 1992.

[17] K. Svanbro, "Lower layer guidelines for robust RTP/UDP/IP Header Compression," *draft-ietf-rohc-rtp-lower-layer-guidelines-03.txt*, Dec 2001.

[18] P. Coverdale, "Itu-T Study Group 12: Multimedia QoS requirements from a user perspective," *Workshop on QoS and user perceived transmission quality in evolving networks*, Oct 2001.

[19] W.C. Jakes, "Microwave mobile communications," *Wiley, New York, 1974*.

[20] M. Sherman, "Should parameterized qos be optional," doc IEEE 802.11-02/620r1, IEEE Task Group E Contribution, Sept 2002.