Ad Hoc Networks xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks

Anna Lina Ruscelli [a,*], Gabriele Cecchetti [a,**], Angelo Alifano [b], Giuseppe Lipari [a]

[a] *ReTiS Lab., Scuola Superiore S. Anna, Via G. Moruzzi 1, 56124 Pisa, Italy*
[b] *Dip. di Ing. dell'Informazione, Università di Pisa, Via Diotisalvi 2, 56122 Pisa, Italy*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Wireless Lan
Quality of Service
Scheduling algorithms
Performance evaluation

ABSTRACT

The IEEE 802.11e standard introduces Quality of Service support for wireless local area networks through two MAC functions: Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA). While the former provides prioritized contention-based access to the medium, the latter uses a parameterized contention-free polling scheme. Several studies have proposed enhancements to EDCA or improved scheduling algorithms for HCCA to properly support VBR traffic. However, the cooperation between these functions has only marginally been considered and the solutions vary depending on specific traffic requirements.

In this paper we propose a novel approach to address the problem of scheduling VBR traffic streams. Our scheduler, named Overboost, uses HCCA to negotiate a minimum bandwidth and deals with traffic streams that require more bandwidth than the negotiated one by redirecting the excess bandwidth to the EDCA function. An analytical evaluation has been conducted and the results has been corroborated by an extensive set of simulations. They show that the overall scheduler improves the performance with respect to other HCCA schedulers in terms of null rate, throughput, access delay, and queue length.

1. Introduction

In recent years, the consolidated role of the IEEE 802.11 standard [1] as reference for *wireless local area networks* (WLANs) has fostered their wide diffusion. The *Medium Access Control* (MAC) layer of the IEEE 802.11b [2] standard is designed for best effort data transmissions. However, the support for multimedia applications (VoIP, videoconference, multimedia streaming, High Definition TV, etc.) is fast-growing and promising, stressing the current technology for improvements to the offered network service. In order to support the *Quality of Service* (QoS) requirements of the multimedia applications – expressed for instance in terms of guaranteed bandwidth and bounded delay and jitter – the IEEE 802.11 Working Group for WLAN Standards has produced the release of the IEEE 802.11e amendment [3].

The IEEE 802.11e amendment introduced traffic and service differentiation at the MAC layer so that it can deliver multimedia streams with the respect of their QoS and real-time (i.e. timing constraints expressed in terms of flows deadlines) requirements, in addition to regular best effort traffic. To this end, it introduces the *Hybrid Coordination Function* (HCF) that deals with both contention-based and contention-free access mechanisms and provides prioritized and parameterized QoS. In particular two new MAC functions are added to the pre-existing ones: the *Enhanced Distributed Channel Access* (EDCA) and the *HCF Controlled Channel Access* (HCCA). EDCA improves the mandatory and contention-based *Distributed Coordination Function* (DCF) by introducing traffic prioritization. On the other hand, HCCA enhances the optional *Point*

* Corresponding author. Tel.: +39 050882111.
** Principal corresponding author.
E-mail addresses: a.ruscelli@sssup.it (A.L. Ruscelli), g.cecchetti@sssup.it (G. Cecchetti), alifangelo@gmail.com (A. Alifano), g.lipari@sssup.it (G. Lipari).

1570-8705/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2010.09.014

Coordination Function (PCF) polling scheme with a parameterized traffic classification. The standard also suggests an HCCA reference scheduling algorithm that respects a minimum set of performance requirements.

The numerous studies that have provided insight on the HCCA function through analytical techniques and simulations [4–8] have demonstrated its good QoS performance for *Constant Bit Rate* (CBR) traffic. However, those studies also highlighted the poor performance of the HCCA function in the case of *Variable Bit Rate* (VBR) traffic. This is due to the fixed values the HCCA function assigns to the transmission parameters. Hence, some scheduling algorithms, alternative to the suggested reference scheduler, have been proposed to improve the HCCA QoS support.

To the best of our knowledge, very few works have considered the possibility to integrate the service provided by HCCA with the resources available for EDCA. The IEEE 802.11e standard describes a further access policy, the *HCCA–EDCA Mixed Mode* (HEMM), where both these functions are used. However, HEMM is not well documented and very few studies [9–11] have analyzed the QoS provisioning of the whole HCCA–EDCA system.

In this paper we present a scheduler for supporting VBR traffic streams with QoS requirements. Our scheduler allows the network node to use both the IEEE 802.11e MAC functions for the transmission of the same traffic stream. Our scheduler achieves this by sending the traffic that exceeds the assigned HCCA transmission time using the EDCA function, i.e., tries to boost the network performance. For this reason we named the proposed local scheduler "*Overboost*". The rest of the paper is organized as follows: in Section 2 the IEEE 802.11 standard is illustrated. In Section 3 we summarize some relevant works that improve the HCCA reference scheduler and some proposals about using HEMM mode. In Section 4, the proposed scheduler is described. In Section 6, we evaluate its performance through simulations. Finally, in Section 7 we present our conclusions.

2. IEEE 802.11e MAC protocol

The IEEE 802.11e compensates for the lack of QoS and real-time support of the IEEE 802.11b standard by introducing two new functions. HCF, located in the *QoS Access Point* (QAP), coordinates the global access of the period by establishing an *hyperperiod*, divided into many intervals, which can be contention-free (managed by the HCCA function) or contention-based (managed by the EDCA function). In order to overcome the starvation problem due to the unbounded channel occupation by low rate stations (which is one of the shortcomings of the legacy DCF function of IEEE 802.11.b), a constraint in the transmission time is added in both access methods. Each *station with QoS* (QSTA) can transmit no more than a fixed time interval, named *Transmission Opportunity* (*TXOP*). Each QSTA can transmit one or more *Traffic Streams* (TSs) with QoS requirements.

2.1. HCF Controlled Channel Access

The HCCA function introduces a parameterized QoS support in the centralized polling mechanism of PCF. HCCA classifies the traffic into height TSs. Each TS, that can be *uni-directional* (uplink or downlink) or *bi-directional* (both of them), corresponds to a specific service level identified by the values of the *Traffic Specification* (TSPEC) protocol parameters. The TSPEC mandatory fields are illustrated in Table 1. By the negotiation of these parameters during the admission control phase HCCA can guarantee TSs a parameterized QoS access to the medium. In order to be included in the polling list, a QSTA sends to the *QoS-aware Hybrid Coordinator* (HC), usually located at the QAP, a QoS reservation request for each of its TSs, using the QoS management frame *Add Traffic Stream* (ADDTS). Therefore, if the acceptance of the requesting TSs does not jeopardize the service guarantees of the already admitted ones, the QAP notifies the admission to the QSTA with a positive acknowledgement, containing the TS service start time. Moreover, HC aggregates the QSTA TSPECs in the following transmission parameters, used to manage the available resource with QoS and sent to the QSTA at the polling time: the *Service Interval* (*SI*), which is the time interval between two successive polls of the node, and the Transmission Opportunity (*TXOP*) which is the node transmission duration, based on the mean application data rates of its TSs.

Table 1
Glossary of used notation.

TSPEC parameters (units)	Symbol
Mean data rate (b/s)	R
Nominal SDU size (B)	L
Minimum PHY rate (b/s)	Γ
Delay bound (s)	D
Maximum Service Interval (s)	MSI
Other parameters (units)	**Symbol**
Peak data rate (b/s)	Π
Peak frame rate (frames/s)	Φ
Burstiness factor	B
Frame size (b)	L
Frame Interarrival time (s)	τ
Interarrival time upper tolerance (s)	δ_u
Interarrival time lower tolerance (s)	δ_l
Service Interval (s)	SI
Transmission Opportunity interval (s)	$TXOP$
Number of transmitted SDU	N
Amount of transmitted data (b)	χ

The *Controlled Access Phase* (CAP) is started by the QAP. It listens the medium and if this is idle for a *PCF Interframe Space* (PIFS), it gains the control of the channel. It then polls a QSTA at turn, according to the polling list, generated by the scheduler considering the QoS and real-time requirements. If the polled QSTA does not have data messages for the considered TS, i.e. the TS is not backlogged, or if the head-of-line frame does not fit into the remaining *TXOP* duration, the QSTA sends a *QoS CF-Null* frame to the QAP. The HCCA frames exchange sequence is shown in Fig. 1.

2.2. Enhanced Distributed Channel Access

The EDCA function provides a prioritized QoS to the DCF contention mechanism based on *Carrier Sense Multiple Access /Collision Avoidance* (CSMA/CA). During the contention phase each station listens to the medium. If the medium is

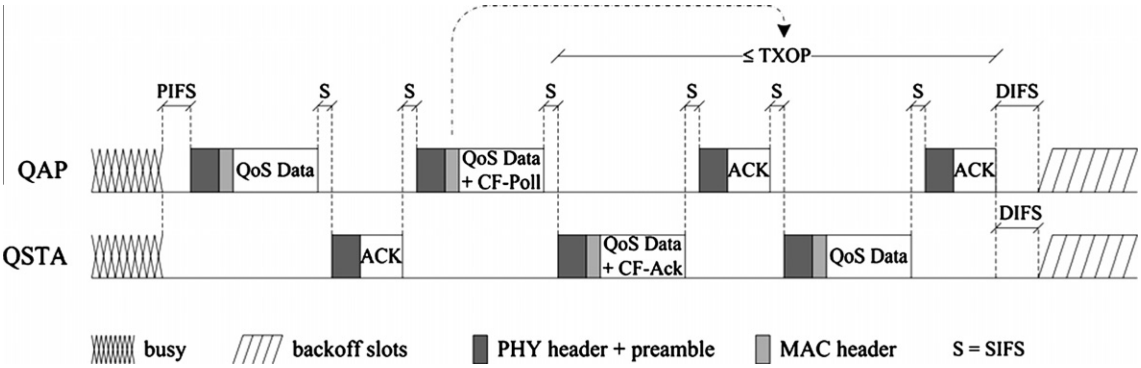


Fig. 1. HCCA sample frames exchange sequence.

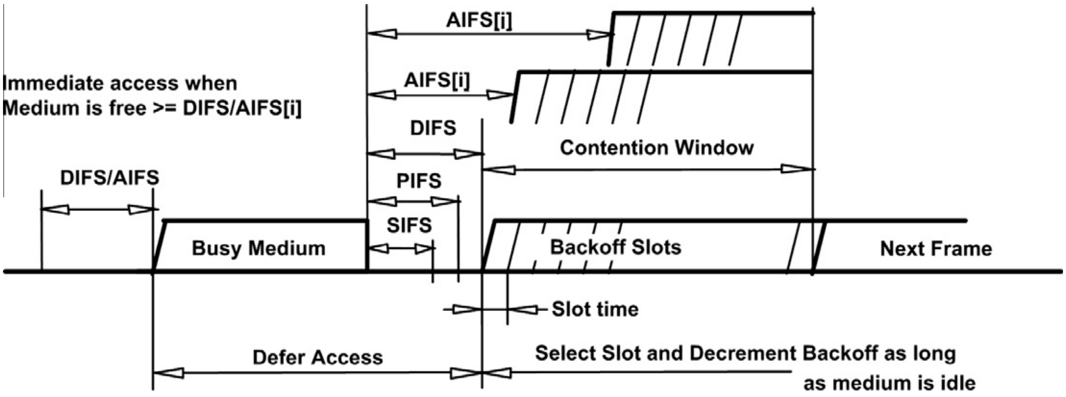

AIFS[i]
AIFS[i]
Immediate access when Medium is free >= DIFS/AIFS[i]
DIFS
Contention Window
DIFS/AIFS
PIFS
SIFS
Busy Medium
Backoff Slots
Next Frame
Slot time
Defer Access
Select Slot and Decrement Backoff as long as medium is idle

Fig. 2. Basic access method and IFS relationships.

free for an interval equals to the *Arbitration Inter-Frame Space* (AIFS) plus a random *backoff* time, the station sends its packets (see Fig. 2). Otherwise the station freezes its waiting timer until the medium becomes again available and then restarts its countdown. An exponential backoff reduces the collision probability.

EDCA uses eight different *User Priorities* (UPs) to classify the traffic. These UPs are mapped into four *Access Categories* (AC), implemented as *First In First Out* queues (see Table 2). Therefore, different kinds of applications (background, best effort, video and voice traffic) can receive a diverse service accordingly to their QoS requirements. The prioritization is obtained by assigning each AC different contention phase parameters: *i)* an increasing AIFS for decreasing AC priority *ii) Contention Window Min* (CW_{min}) and *Contention Window Max* (CW_{max}) used to compute different backoff periods. A QSTA with higher priority AC traffic has a backoff interval shorter than that of QSTAs with lower priority traffic which allows it to acquire a higher right of access to the medium. Also, a virtual contention handler solves the internal collisions of QSTAs that end their contention backoff intervals at the same time. It forces the QSTA with higher priority to gain the control of the medium by increasing the contention windows of the lower priority ones.

Table 2
UP to AC mappings.

Priority	User Priority	Access Priority	Traffic type
Lowest	1	AC_BK	Background
	2	AC_BK	Background
	0	AC_BE	Best effort
	3	AC_BE	Best effort
	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Voice
Highest	7	AC_VO	Voice

3. Related works

In this section the scheduling algorithms we considered in our work are analyzed, along with the IEEE 802.11e HCCA reference scheduler. To the best of our knowledge they are representative of the different available solutions introduced to improve the HCCA QoS support, without the aim to provide a survey on this topic. Also, we present a brief summary of the few works on the HEMM mode.

3.1. The HCCA schedulers

The IEEE 802.11e reference scheduler proposes how to compute the main protocol parameters, *SI* and *TXOP*, suitable to meet the requirements globally expressed by each QSTA. *SI* is computed as a unique value for all non-AP

QSTAs with admitted streams, whereas different values of *TXOP* are computed for each QSTA. Hence all the QSTAs are polled with the same period *SI* and different TSs of a station $QSTA_i$ are served with the same computation time $TXOP_i$. In particular, *SI* is proposed to be less than the beacon interval, so that the QSTAs will be polled at least once during the beacon duration. It also has to be less than the minimum of the *Maximum SI* (*MSI*) of each QSTA. This ensures no deadline misses. *SI* and $TXOP_i$ have fixed values, based on *worst case* conditions, and they are recomputed only if a new TS arrives with a MSI_i greater than the preexistent ones.

The numerous studies on the IEEE 802.11e HCCA reference scheduler [4,5,7] have proven through analysis and simulations that it has good QoS and real-time performance only with CBR traffic. Instead, due to the fixed values assigned to *TXOP* and *SI*, it provides a non-optimal resource utilization in the case of VBR traffic, (videoconference, video streaming, etc.). Same conclusions are drawn for the admission control. It misses on the necessary flexibility and applies stringent admission conditions with the acceptance of fewer TSs than possible, by wasting available resources. As a result, several alternative scheduling algorithms have been proposed to improve the QoS provisioning of IEEE 802.11e HCCA networks in the case of VBR traffic, [12–17] and few works have evaluated the real-time issues of the reference scheduler [6,7,12,18–21], and proposed possible solutions.

Fair HCF (FHCF) [14] assigns variable *TXOPs* by using a model of TSs queues length that distinguishes between the packet queuing delay and the waiting time delay. The packet queuing delay is influenced by the variations in packet size and data rate. The waiting time delay is defined as the interval between the packet arrival time and the QSTA polling time.

The *Real-Time HCCA* (RTH) algorithm [19] ensures a fixed capacity for the traffic streams during a fixed period. This scheduler, based on the *Earliest Deadline First* (EDF) [22] and *Stack Resource Policy* (SRP) [23] algorithms, takes into account the non-preemptability of the frame transmissions which are considered as critical sections.

The *Wireless Capacity Based Scheduler* (WCBS) [18,7] is suitable for serving soft real-time applications and is derived from the *Constant Bandwidth Server* [24]. At admission control stage, it assigns TSs periodic time budget, then it schedules the TXOP dynamically distributing this budget taking into account the EDF-based order of the deadlines associated to each data message. Moreover it allows to postpone the deadline whenever the higher traffic variability cannot be served with the assigned budget.

3.2. The HCCA–EDCA Mixed Mode

Very few works have considered the opportunity to enhance the HCCA QoS and real-time support by taking advantage of the resources available for the EDCA function. Nonetheless the IEEE 802.11e standard specifies three different access policies: HCCA, EDCA, and HCCA–EDCA Mixed Mode (HEMM). In particular, the HEMM mode permits a QSTA to jointly use the former MAC mechanisms. It is not adequately documented and to the best of our knowledge, the related studies are focused on the models of the used channel capacity and on the adjustment of the HCCA/EDCA ratio.

In [9] a model of the channel utilization is presented, considering both HCCA and EDCA modes. It shows that incrementing the portion of HCCA increases the medium utilization of large WLAN in saturation conditions and the determinism in the channel control. Instead large EDCA networks are affected by growing collisions that degrade their performance.

In [10] the efficient resource control for elastic traffic over EDCA and HCCA functions is analyzed using an economic model that highlights how the CW_{min} parameter, the CSMA/CA scheme, and the RTS/CTS procedure affect channel congestion and throughput. The optimal value of the HCCA–EDCA ratio is found by means of optimization techniques.

In [11] the *Adaptively Tuned HCF* (AT-HCF) algorithm is introduced. It dynamically adapts and tunes the HCCA and the EDCA durations to the different type of traffic, until they converge to the optimal values to improve the throughput of the overall system.

4. The Overboost scheduler

The proposed *Overboost* scheduler is local to each node. It improves performance of the HCCA schedulers without replacing them and deals with the TSs that are served by HCCA that still have some data to transmit at the end of CAP phase. As a result, before the contention period begins, it moves the TSs data messages from HCCA queue to EDCA queue assigning them to a highest priority EDCA Access Category (Figs. 3, 4), which is `AC_VO` (Voice AC) (see Table 2). Hence, the traffic that exceeds the assigned HCCA *TXOP* (the HCCA transmission time threshold) will not be served with parameterized QoS but will be served with prioritized QoS. The detailed description of the Overboost algorithm is the following:

1. when the CAP phase ends, HC transfers the control of the medium to the Overboost mechanism;
2. it checks if the HCCA is empty: in that case it leaves the control to the EDCA function;
3. otherwise, it moves the data message of HCCA queue to the EDCA function; and
4. if the EDCA period is not yet finished, it starts over from the point 2).

Overboost is a local node scheduler which collaborates with the MAC reference scheduling algorithm or with any alternative one. The centralized scheduler located in the QAP continues to manage the QSTAs that request to send and performs admission control, which remains unchanged. It then computes the scheduling parameters and creates the polling list, and finally it polls the admitted QSTAs. Instead Overboost, located in each QSTA, takes action only if the transmitting QSTA does not deliver all enqueued TSs data messages.

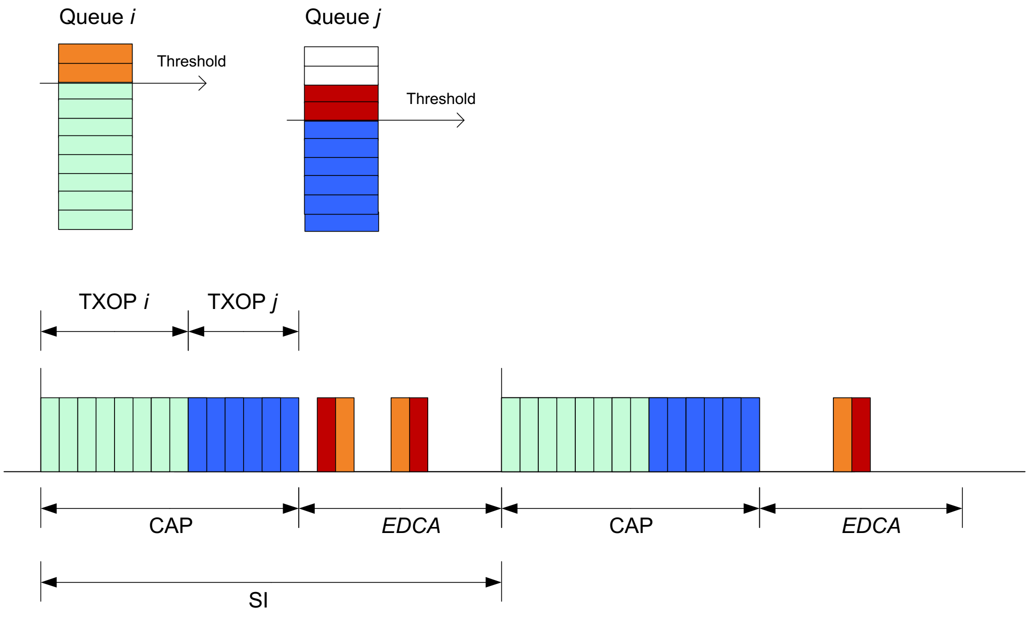


Fig. 3. The Overboost swapping queues mechanism.

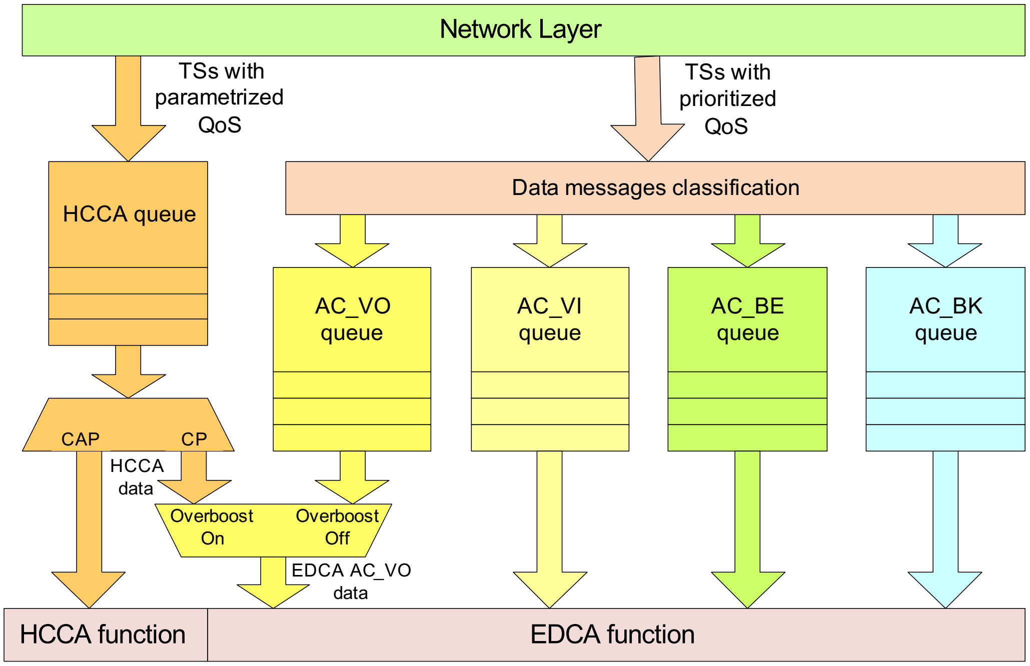


Fig. 4. The Overboost mechanism.

In the rest of the paper we will show that this mechanism limits the delay experienced by TSs and improves their performance. The local scheduler integrates the services offered by both HCCA and EDCA mechanisms, "boosting" the network performance of TSs by the HCCA scheduler. Hence we named this scheduler "*Overboost*".

5. Scheduling analysis

In this section we analyze the Overboost impact on the resource scheduling, tuning the admission control in presence of different types of traffic. The bursty traffic highlights how Overboost reacts to VBR sources like videostreams,

Please cite this article in press as: A.L. Ruscelli et al., Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks, Ad Hoc Netw. (2010), doi:10.1016/j.adhoc.2010.09.014

whereas the VoIP traffic shows that the CBR streams are not affected by Overboost. Moreover the queue length analysis illustrates the scheduler effects on buffer size, throughput and delay.

5.1. Tuning the admission control

In the following we evaluate how to improve the admission control test to take advantage of the resources made available by Overboost.

The general expression of the admission control that considers the time intervals assigned to both EDCA and HCCA functions is

$$T_{CAP} + T_{EDCA} = \sum_{i=1}^{K} (TXOP_i + t_{Pi}) + T_{EDCA} \leqslant T \tag{1}$$

where T is the hyperperiod, T_{CAP} is the length of CAP phase that is equal to the sum of the assigned $TXOP_i$ and of the polling times $t_P\boldsymbol{i}$ of the K admitted QSTAs, and T_{EDCA} is the duration of the EDCA. When Overboost takes action this expression is modified as follows:

$$T'_{CAP} + T_{OV} + T'_{EDCA} \leqslant T \tag{2}$$

where $T'_{CAP} \leqslant T_{CAP}$ is the new duration of CAP phase, T_{OV} is the time used by the Overboost scheduler during EDCA, $T'_{EDCA} = T_{EDCA} - T_{OV} \leqslant T_{EDCA}$ is the time duration assigned to the QSTAs using only EDCA.

The clause 7.3.2.29 of IEEE 802.11 standard defines the maximum length of the EDCA *TXOP* for each AC, the $TXOP_{limit}$ (contained in the *Parameter Record* of the *EDCA Parameter Set Element*). In particular the $TXOP_{limit}$ of `AC_VO` is equal to 3264 ms or 1504 ms, depending on the adopted physical specification. Thus $T_{OV} \leqslant K \cdot TXOP_{limit}$. Considering the previous results we can state the following: when Overboost is used the assigned HCCA $TXOP_i$ can be reduced by taking advantage of the transmission time recovered from EDCA and Eq. (2) becomes:

$$\sum_{i=1}^{K} (TXOP'_i + t_{Pi}) + K \cdot TXOP_{limit} + T'_{EDCA} \leqslant T \tag{3}$$

where in general $TXOP'_i \leqslant TXOP_i$. In particular the minimum value of $TXOP'_i$ assigned during the admission control, without jeopardizing the guaranteed QoS, is $TXOP'_{imin} = TXOP_i - TXOP_{limit}, 1 \leqslant i \leqslant K$.

This result provides more flexibility to the admission control. It is possible to admit the same number of TSs:

- with the same $TXOP_i$ values $(TXOP'_i \equiv TXOP_i, \forall i)$, improving their performance through the resources made available by Overboost, or
- with a reduced $TXOP_i$ values $\left(\sum_{i=1}^{K} TXOP'_i < \sum_{i=1}^{K} TXOP_i\right)$, integrating the missing resources by Overboost. This lets to keep the initial QoS using less resources and eventually to admit more TSs.

This behavior will also be shown through simulation in Section 6.7.

5.2. Transmission opportunity analysis

The HCCA reference scheduler suggests how to compute each $TXOP_i$, assuming $N_i = \lceil (SI \cdot R_i)/L_i \rceil$, for $1 \leqslant i \leqslant K$:

$$TXOP_i = max\left(\frac{N_i \cdot L_i}{R_i}, \frac{M_i}{R_i}\right) + O \tag{4}$$

where M_i is the maximum allowable size of MSDU (2304 bytes), and O is the overheads in time units. In Table 1 the used symbols are listed. Since these formulae use the TSPEC parameters whose values depend on the TSs, it is possible to adapt these expressions and that of the admission control to different types of traffic, replacing the hardwired standard parameters with those derived from the statistical traffic analysis and compare the network behavior with and without Overboost.

5.2.1. Bursty traffic

The bursty traffic, characterized by variable bit rate and packets interarrival times, impacts on the admission control and allows to highlight how Overboost reacts to VBR traffic with active/silence intervals. The mean frame interarrival time $1/\lambda$ of a bursty traffic is variable in the interval $(1/\lambda - \delta_u, 1/\lambda + \delta_l)$, where δ_u is the *upper tolerance* during the burst and δ_l is the *lower tolerance*, when the interarrival time is longer. A meaningful burstiness statistic is the *Burstiness factor*:

$$B = \frac{R}{\Pi}$$

where R is the mean data rate evaluated during a long time interval, and Π is the peak data rate during the activity interval. The peak frame rate is equal to $\Phi = R/(B \cdot L)$ and the bursty frame interarrival time is $1/\lambda = (B \cdot L)/R$. Hence, the CBR applications are characterized by $B = 1$, whereas the VBR traffic sources by $B \leqslant 1$. For $1 \leqslant i \leqslant K$, each $TXOP_i$ can be computed including the traffic parameter B_i, taking into account the traffic bursts when the resources are assigned

$$TXOP_{Bi} = max\left(\frac{N_i \cdot L_i}{B_i \cdot \Pi_i}, \frac{M_i}{B_i \cdot \Pi_i}\right) + O \tag{5}$$

obtaining an expression increased by the factor $\frac{1}{B_i} > 1$ with respect to Eq. (4), computed considering the mean data rate. Hence the scheduler assigns a greater $TXOP_i$ that overestimates the mean requirements.

When Overboost is turned on, taking into account the traffic burstiness statistic B_i, the new $TXOP'_{Bi}$ is:

$$\begin{aligned} TXOP'_{Bi} &= TXOP_{Bi} - TXOP_{limit} \\ &= max\left(\frac{N_i \cdot L_i}{B_i \cdot \Pi_i}, \frac{M_i}{B_i \cdot \Pi_i}\right) + O - TXOP_{limit} < TXOP_{Bi}. \end{aligned}$$

This simple result confirms the usefulness of the Overboost mechanism that can be used to manage the traffic spikes over, without overprovisioning the assigned HCCA bandwidth.

In the following the max admissible burstiness $B_{i\ max}$ is evaluated when $TXOP_i$ are not changed and Overboost manages the traffic variability. At the beginning of current $TXOP_i$, before the burst arrives, if the HCCA queue is empty, the total amount of traffic that can be delivered by the overall system composed by the centralized HCCA scheduler and Overboost is $\chi = (TXOP_i + TXOP_{limit}) \cdot \Gamma$, the corresponding peak frame rate is:

$$\Pi_i = \frac{\chi}{L_i \cdot TXOP_i} = \frac{(TXOP_i + TXOP_{limit}) \cdot \Gamma}{L_i \cdot TXOP_i}$$

and the max frame interarrival time is:

$$\frac{1}{\lambda_i} = \frac{L_i \cdot TXOP_i}{(TXOP_i + TXOP_{limit}) \cdot \Gamma} = \frac{B_{imax} \cdot L_i}{R_i}.$$

Thus, knowing $TXOP_{limit}$, it is possible to derive the maximum burstiness tolerable by the overall scheduler without changing $TXOP_i$:

$$B_{imax} = \frac{R_i \cdot TXOP_i}{(TXOP_i + TXOP_{limit}) \cdot \Gamma}.$$

Finally, adopting a different point of view, the modified mathematical expression of $TXOP'_i$ needed to deliver a TS_i with a burstiness B_i, by taking advantage of the Overboost feature, is:

$$TXOP'_i = \frac{B_i}{R_i - B_i\Gamma} \cdot TXOP_{limit}. \tag{6}$$

Overboost allows to serve more traffic with respect to that delivered by only HCCA and without changing $TXOP_i$, since the variations typical of the bursty traffic are "absorbed" by Overboost using EDCA. These deductions are also valid for VBR traffic, like MPEG streams. These conclusions will be confirmed though simulation by the bandwidth underprovisioning analysis (see Section 6.7).

5.2.2. VoIP traffic

The VoIP streams can be modeled as an ON/OFF source: during the ON (*talkspurt*) periods the traffic is modeled as CBR with parameters that depend on the encoding scheme; during the OFF (*silence*) periods no frames are generated. Talkspurt and silence periods are distributed according to the Weibull distribution [25] that models a one-to-one conversation.

Assuming $R_{ON} = 1/T_{ON}$ as the constant frame transmission rate during the ON period, where T_{ON} is the mean duration of the ON period, the mean frame rate during the overall period $T = T_{ON} + T_{OFF}$ is $\overline{R} = R_{ON} \cdot T/(T_{ON} + T_{OFF})$ [26] where T_{OFF} is the mean duration of the silence period, when no frames are generated. During the talkspurt period the VoIP traffic source generates CBR frames, characterized by constant data rate, SDU size and frame interarrival time. Thus in the $TXOP_i$ expression it is correct to use parameters related to the ON period: R_i is set to $\overline{R}$, SI to T, and L_i to the SDU size. In particular, the values of these parameters depend on the VoIP codec, as shown in Section 6. This choice does not overestimate the needed resources since $\overline{R_i}$ allows to distribute the frames actually generated during T_{ON} in whole T.

Moreover the max number of transmitted SDU can be approximated as:

$$N_i = \left\lceil \frac{SI \cdot R_i}{L_i} \right\rceil = \left\lceil \frac{E[T_{ON}] \cdot \overline{R}_i}{L_i} \right\rceil \tag{7}$$

and since the talkspurt period is distributed according to the Weibull distribution, whose probability density function (p.d.f.) is:

$$p_{T_{ON}}(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \cdot \left(\frac{x}{\lambda}\right)^{k-1} \cdot e^{\left(-\frac{x}{\lambda}\right)^k} & x \geqslant 0 \\ 0 & x < 0 \end{cases}$$

where k is the *shape* parameter and λ is the *scale* parameter that depend on the streams, the mean value of T_{ON} is: $E[T_{ON}] = \lambda \cdot \Gamma(1 + 1/k)$ where Γ is the *Gamma* function, $\Gamma(z) = \int_0^\infty t^{z-1} \cdot e^{-t} dt$. Thus

$$TXOP_i = max\left(\frac{\lceil \lambda_i \cdot \Gamma(1 + 1/k_i) \cdot \overline{R}_i/L_i \rceil \cdot L_i}{\overline{R}_i}, \frac{M_i}{\overline{R}_i}\right) + O \tag{8}$$

where λ_i and k_i are the parameters of the Weibull distribution that depends on the considered TS.

Taking into account the previous considerations we can state that Eq. (8) does not differ from Eq. (4) since the used parameters are derived from the mean value statistics during the ON period. Thus the CBR traffic, like VoIP, is not affected by the Overboost mechanism since the centralized scheduler can efficiently manage the application requirements.

5.3. Queue length analysis

In this section the improvement introduced by Overboost in the performance of the HCCA scheduling is analytically studied in terms of emptying HCCA queue rate that affects the experienced delay and throughput. These considerations will be confirmed by the introduction of a simple queue length model that shows how the amount of enqueued traffic is efficiently reduced by Overboost.

When Overboost is turned off the maximum $QSTA_i$ enqueued traffic, supposed backlogged, that is delivered during a single hyperperiod H, is equal to $TXOP_i \cdot \Gamma$, whereas when Overboost is turned on, it is equal to $(TXOP_i + TXOP_{limit}) \cdot \Gamma$. Hence the maximum increase in the dispatched traffic $\eta_{i\ max}$ introduced by Overboost for the $QSTA_i$ is:

$$\eta_{maxi} = \frac{(TXOP_i + TXOP_{limit}) \cdot \Gamma - TXOP_i \cdot \Gamma}{TXOP_i \cdot \Gamma} = \frac{TXOP_{limit}}{TXOP_i}.$$

Considering all the K admitted $QSTAs$, the maximum total increase in the dispatched traffic during H is:

$$\eta_{max} = \frac{\sum_{i=1}^{K} TXOP_{limit}}{\sum_{i=1}^{K} TXOP_i} \cdot f = \frac{K \cdot TXOP_{limit}}{\sum_{i=1}^{K} TXOP_i} \cdot f$$

where f is a factor that takes into account the TSs type which affects the Overboost activity (in the case of CBR TSs $f = 0$, whereas for backlogged VBR TSs $f = 1$).

Analyzing the emptying queue process, when Overboost is turned off, at time t after the start of the j^{th} hyperperiod, H_j, the enqueued traffic τ_{ij} of $QSTA_i$ is:

$$\tau_{ij} = Q_{i,j-1} + t \cdot (R_i + \Gamma)$$

where $Q_{i,j-1}$ is the residual traffic from the previous hyperperiod H_{j-1}, $t \cdot R_i$ is the incoming traffic in the queue during

H_j until t, and $t \cdot \Gamma$ is the outcoming one. $Q_{i,j-1}$ is due to the traffic not delivered during H_{j-1}, considering both the residual traffic from the previous hyperperiods and the incoming and the outcoming traffic during H_{j-1}, $Q_{i,j-1} = Q_{i,j-2} + TXOP_i \cdot (R_i + \Gamma)$. By iterating: $Q_{i,j-2} = Q_{i,j-3} + TXOP_i \cdot (R_i + \Gamma)$ ... $Q_{i,j-1} = (R_i - \Gamma) \cdot \sum_{t=1}^{j-1} TXOP_i$ and being $TXOP_i$ constant for each hyperperiod, finally we get:

$$Q_{i,j-1} = (j-1) \cdot TXOP_i \cdot (R_i - \Gamma).$$

Thus the $QSTA_i$ traffic enqueued in the HCCA queue length during H_j is:

$$\tau_{ij} = (j-1) \cdot TXOP_i \cdot (R_i - \Gamma) + t \cdot (R_i + \Gamma).$$

and the total enqueued traffic of all the K admitted *QSTAs*, i.e. the HCCA queue length, is:

$$\tau_j = (j-1) \cdot (R_i - \Gamma) \cdot \sum_{i=1}^{K} TXOP_i + t \cdot \sum_{i=1}^{K} (R_i + \Gamma).$$

When Overboost is turned on, considering a backlogged traffic, the queue length is:

$$\tau_{jOV} = (j-1) \cdot (R_i - \Gamma) \cdot \sum_{i=1}^{K} TXOP_i - K \cdot J \cdot TXOP_{limit} + t \cdot \sum_{i=1}^{K} (R_i + \Gamma) < \tau_j. \tag{9}$$

Eq. (9) illustrates how Overboost efficiently contributes in the emptying HCCA queue process. This positively impacts on scheduling performance evaluated in terms of throughput and experienced delay.

6. Performance analysis

This section presents the results of Overboost performance evaluation obtained through simulations. We analyzed the benefits of the new local scheduler when combined with previously described HCCA schedulers: IEEE 802.11e reference [3], WCBS [18], RTH [27], and FHCF [14]. A performance analysis showing the main features and differences among these scheduling algorithms can be found in [7].

After describing the simulation settings and the used traffic model, we discuss the results about the schedulers efficiency in terms of the experienced null rate, the throughput, the access delay, and the QSTA queue length. Then we evaluate the performance of the overall scheduler in comparison with the bandwidth *over-provisioning* and *under-provisioning* and with the only EDCA function.

6.1. Simulation settings

The software implementation of the Overboost algorithm has been developed for *ns-2* [28] simulator by introducing an extension able to use both HCCA and EDCA. Regarding the HCCA software implementation we focused on the framework proposed in [29], due to its modularity. We then implemented the EDCA part and added the necessary code to integrate the two access functions [30]. Finally the Overboost algorithm was implemented as a local scheduler for each QSTA.

Table 3
MAC/PHY simulation parameters.

Parameter	Value (µs)	Parameter	Value
SIFS	10	PHY header	192 µs
PIFS	30	Data rate	11 Mb/s
DIFS	50	Basic rate	1 Mb/s
Slot time	20	Bit error rate	0 b/s

The used physical layer parameters are specified by the *High Rate-Direct Sequence Spread Spectrum* (HR-DSSS), also known as 802.11b, and reported in Table 3. MAC level fragmentation, multirate support and RTS/CTS protection mechanism are disabled. We also assume that all nodes can directly communicate with each other without the hidden node problem.

The analysis has been carried out using the method of independent replications. Specifically, we ran independent replications of 3600 s, each with 100 s warm-up period until the 95% confidence interval is reached for each performance measure. Confidence intervals are not drawn whenever negligible.

6.2. Traffic model

For simulations we used two types of uplink (UL) traffic streams that require QoS guarantees: VoIP and video.

The VoIP traffic is simulated using a VoIP generator module for ns-2 described in [31]. The VoIP streams are modeled as an ON/OFF source as described in Section 5.2.2. The Weibull distribution parameters are listed in Table 4. The employed encoding scheme is the G.729A [32] with the parameters as shown in Table 5. The TSPEC delay bound is set to the packet interarrival time (*period*) and the mean data rate to the peak rate during talkspurts.

The video stream traffic is generated using pre-encoded MPEG4 trace files from the Internet archive of traces [33]. An MPEG4 encoder produces streams of variable size frames at fixed intervals [34]. They are chosen to represent a videoconference session (*LectureHQ-Reisslein* trace file) and a video streamed over the network (*Jurassic Park High Quality* trace file). The TSPEC parameters are shown in Table 6.

Table 4
Voice activity detection model for VoIP traffic streams.

Type	λ_{ON} (s)	k_{ON}	λ_{OFF} (s)	k_{OFF}	$E[ON]$ (s)	$E[OFF](s)$
O2O	1.423	0.824	0.899	1.089	1.58	0.87

Table 5
G.729A VoIP encoding scheme parameters.

Codec	G729A
Frame size (B)	10
Period (s)	0.02
Sample per packet	2
Payload size (B)	20
IP/UDP/RTP Header size (B)	40
SDU size (B)	60
Data rate (b/s)	24,000

Please cite this article in press as: A.L. Ruscelli et al., Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks, Ad Hoc Netw. (2010), doi:10.1016/j.adhoc.2010.09.014

Table 6
Traffic parameters for video streams.

Parameter	VideoConf.	VideoStr.
Mean frame size (B)	660	3800
Max frame size (B)	11,386	16,745
Period (s)	0.033333	0.040
Mean data rate (b/s)	157,712	770,000
Peak data rate (b/s)	2,732,640	3,300,000

All the simulations include one station with background data traffic operating in asymptotic conditions (i.e. it always has a frame to transmit). The packet length of data traffic is constant and equal to 1500 bytes and the best effort data traffic is transmitted using legacy DCF.

6.3. Efficiency analysis

In this section we analyze the efficiency of the four global schedulers, intended as a measure of how well they utilize the network resources. The evaluation is performed using the analysis of the null rate experienced during the polling of the QSTAs. The null rate is defined as the number of *Null* frames received by the QAP when it has sent a *CF-Poll* frame and the QSTA has no frames to transmit. When evaluating the null rate, we can check if the polling time computation is suitable for the considered traffic or the QAP is polling the QSTAs more frequently than necessary (which may increase the system overhead). In Fig. 5, we show the null rate value for the four analyzed schedulers with and without the Overboost mechanism, in a scenario composed by three VoIP G.729A uplink TSs and three video stream uplink TSs. In the case of G.729A TSs the Overboost does not improve the schedulers performance and the null rate is almost the same for all the schedulers, with and without Overboost. The schedulers compute the same *TXOP* value and the same *SI* value that is equal to the packet interarrival time, hence they have the suitable polling rate to empty the queue. The null frames are only due to the silence periods. Instead with video stream TSs the Null frames significantly increase when Overboost is activated: at the end of the *TXOP*, the data messages in the HCCA queue are swapped in the higher priority EDCA queue. Therefore, when the QAP polls the QSTA in the next CAP, there is more chance to find the queue empty. However, even in the case of the increased null rate the delay is reduced, as shown in Section 6.6.

The polling analysis was not conducted because the local Overboost scheduler does not modify the HCCA scheduling algorithm, its polling list, and timings. Therefore the results remain untouched with respect to [7].

6.4. Throughput analysis

Fig. 6 shows the throughput of each algorithm with (*left column*) or without Overboost (*right column*) when VoIP or video stream TSs are transmitted. It highlights that, in the case of CBR traffic the Overboost mechanism does not affect any of the analyzed schedulers. This means that they assign a *TXOP* large enough to send all the frames spawn during one *SI*. On the other hand, in the case of VBR traffic we note that some of the frames are sent in EDCA manner. As a consequence, all the schedulers use less CAP time and, in particular, the reference, RTH, and WCBS schedulers improve their throughput. FHCF does not change its behavior because its queue model is already flexible enough, by

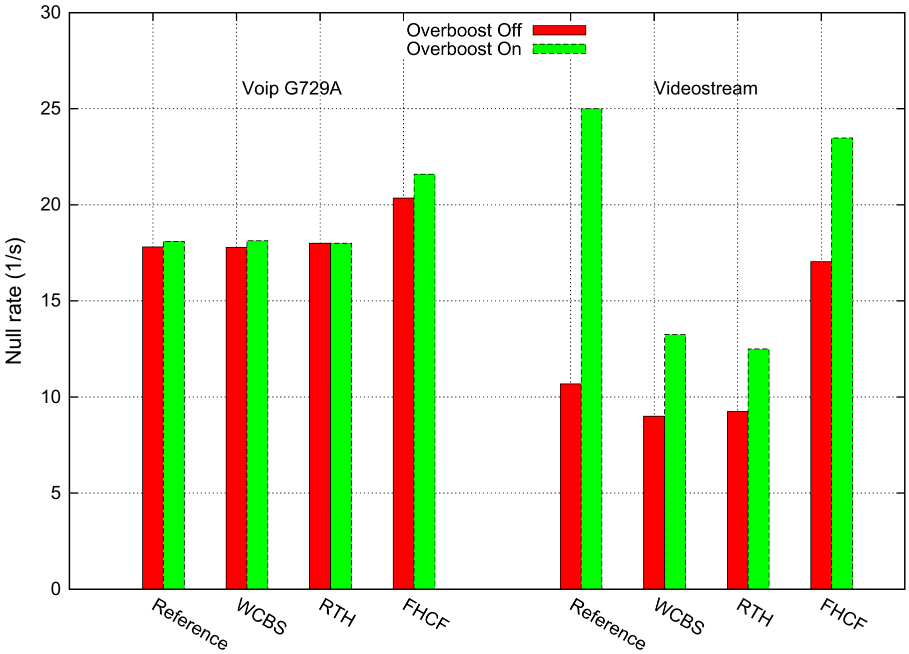


Fig. 5. Null rate of UL VoIP G.729A and UL video stream TSs.

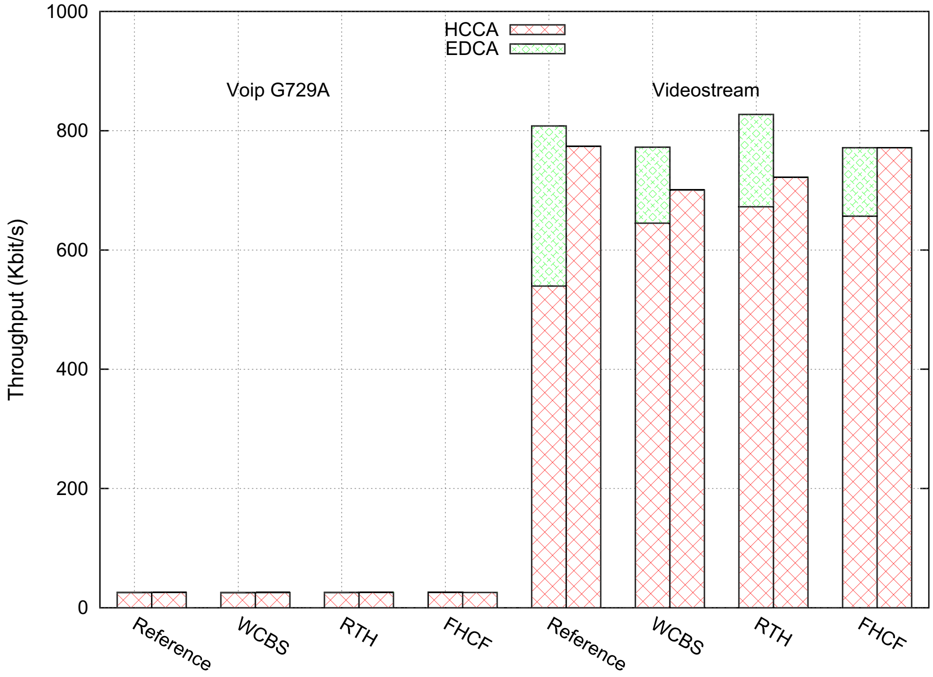


Fig. 6. Throughput of two UL VoIP G.729A, four UL video conference and two UL video stream TSs.

assigning variable *SI* and *TXOP* suitable to follow the traffic variations.

6.5. The queue length analysis

Figs. 7 and 8 show the 99th percentile of the queue length of QSTA transmitting respectively one videoconference TS and one video stream TS. The EDF-based schedulers WCBS and RTH need a queue greater than others because they strictly reserve the needed bandwidth using the mean data rate value, without over-provisioning (like the reference scheduler) or adopting a flexible scheduling scheme (like FHCF). Instead, when Overboost is activated, we have a significant decrease in the queue length because every *SI* the

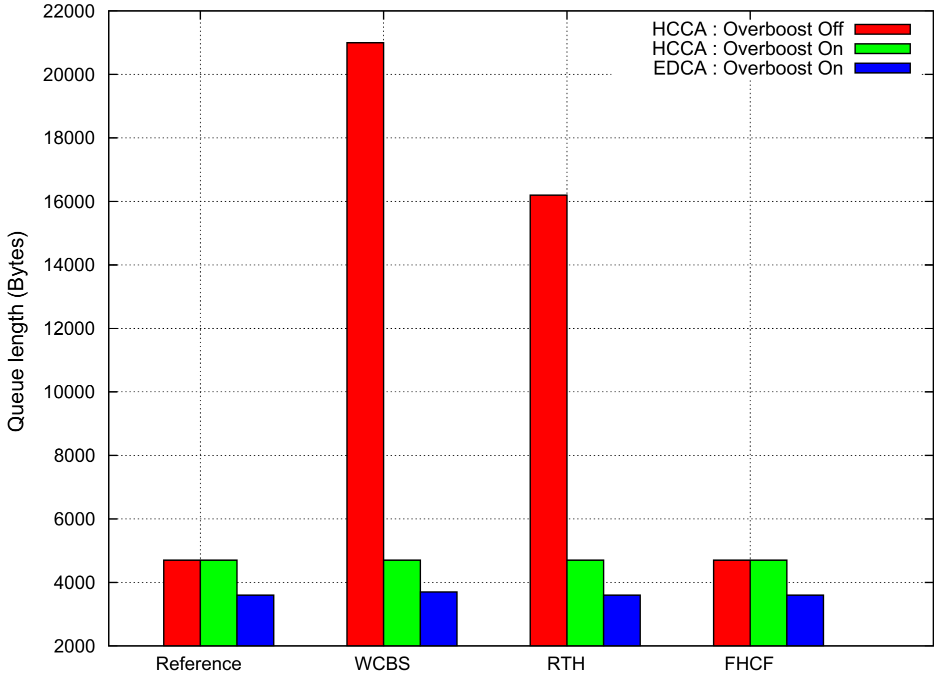


Fig. 7. Ninety ninth percentile of buffer size when video conference TSs are transmitted.

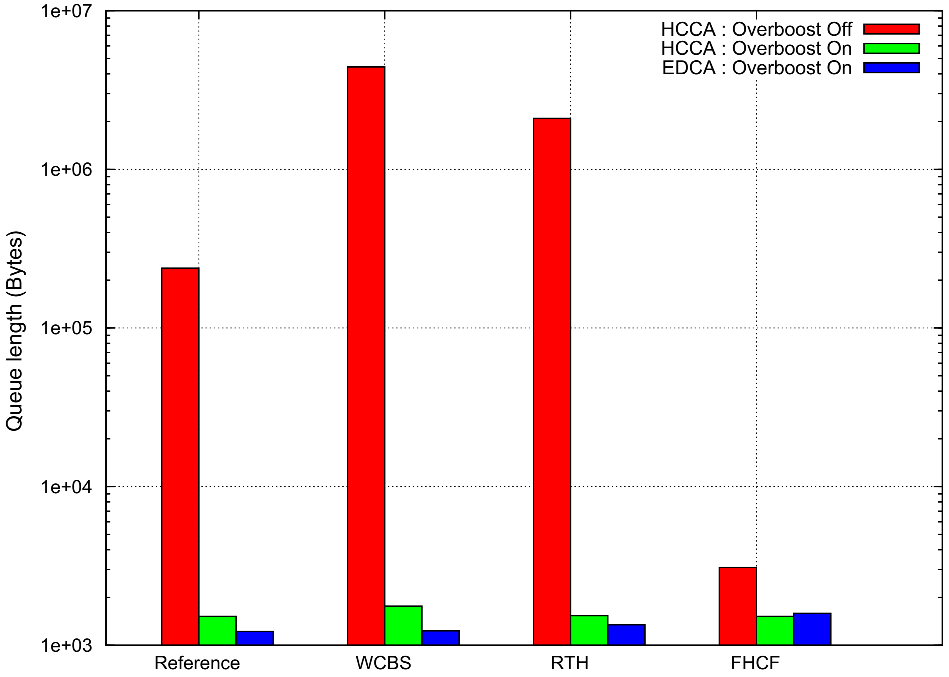


Fig. 8. Ninety ninth percentile of buffer size when video stream TSs are transmitted.

data messages are moved from the HCCA queue to the EDCA queue as analytically shown in Section 6.5. This effect is evident in the case of the video stream TSs.

6.6. Delay analysis

In this section we investigate the access delay defined as the time elapsed from the time the frame reached the MAC layer to the time when the frame is successfully acknowledged. We consider a scenario with an increasing number of VoIP G.729.A TSs. The *Cumulative Distributed Function* (CDF) of the access delay (Fig. 9) shows that the access delay of all the schedulers does not change, hence confirming that Overboost does not affect this kind of traffic.

Finally we consider a mixed scenario with two VoIP G.729A uplink TSs, four videoconference TSs and two video

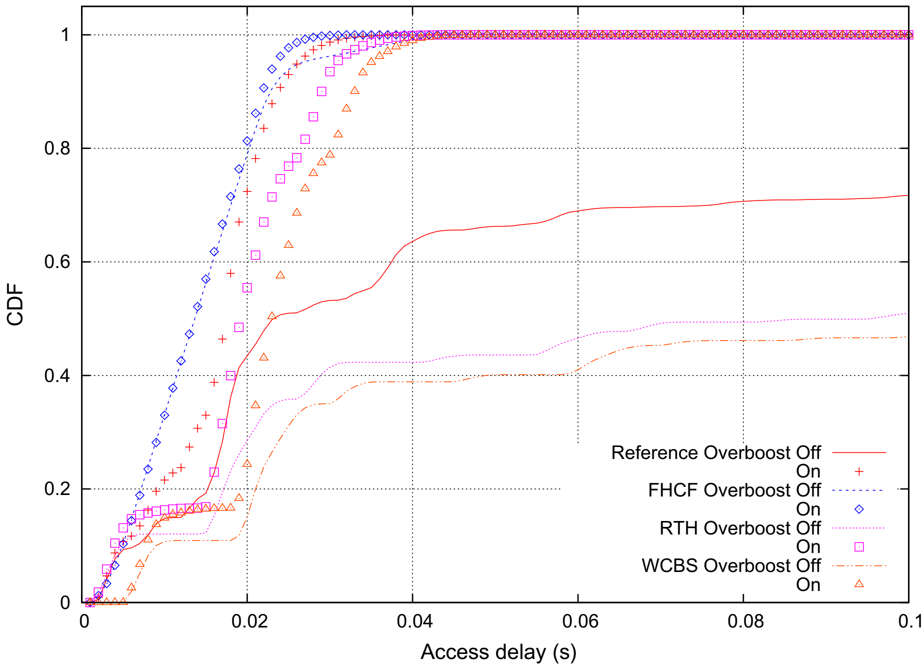


Fig. 9. CDF of access delay when eight UL VoIP G.729.A TSs are transmitted.

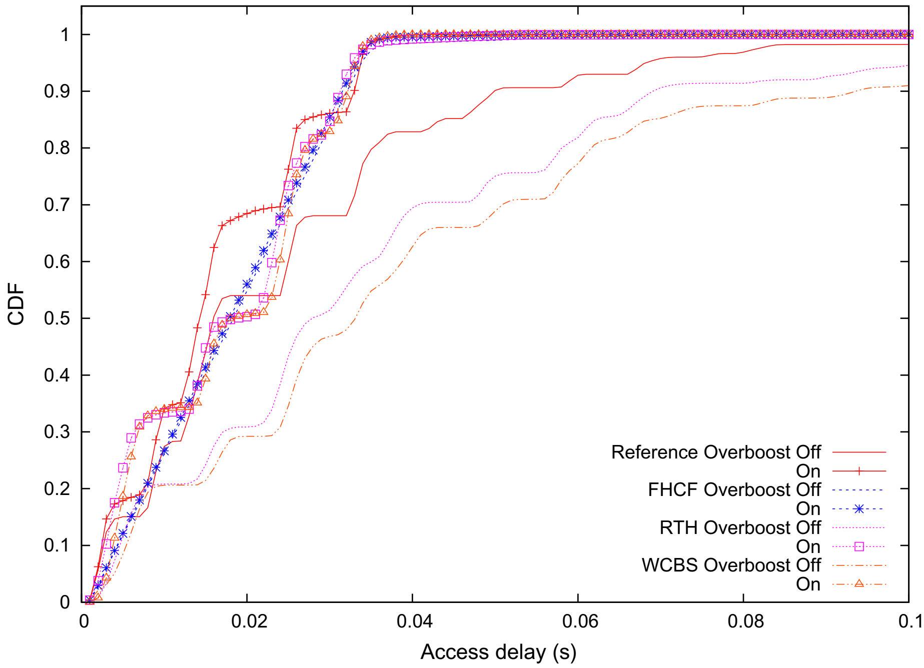


Fig. 10. CDF of access delay when eight UL video conference TSs are transmitted.

stream TSs. Fig. 10 shows the CDF of the access delay of last admitted QSTA transmitting videoconference TS. The slope of each probability curve is increased, meaning that every scheduler has more probability to keep the access delay under a specific value. Also, in this case the delay is lower when Overboost is activated, especially with EDF-based schedulers. This is an important result since the EDF-based schedulers produce a mean delay greater than the others, executing a new sorting for each CAP phase, while the other schedulers maintain a fixed order of TSs. In presence of Overboost we note how this gap between the schedulers is reduced.

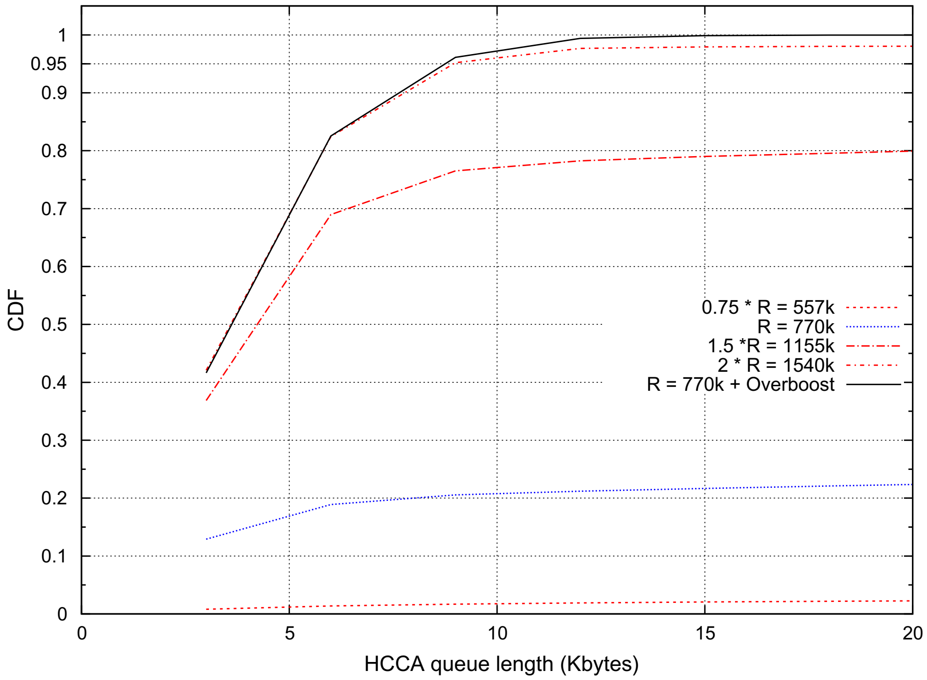


Fig. 11. CDF of the HCCA queue length in the case of *overprovisioning* and *underprovisioning* versus *Overboost*.

Please cite this article in press as: A.L. Ruscelli et al., Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks, Ad Hoc Netw. (2010), doi:10.1016/j.adhoc.2010.09.014

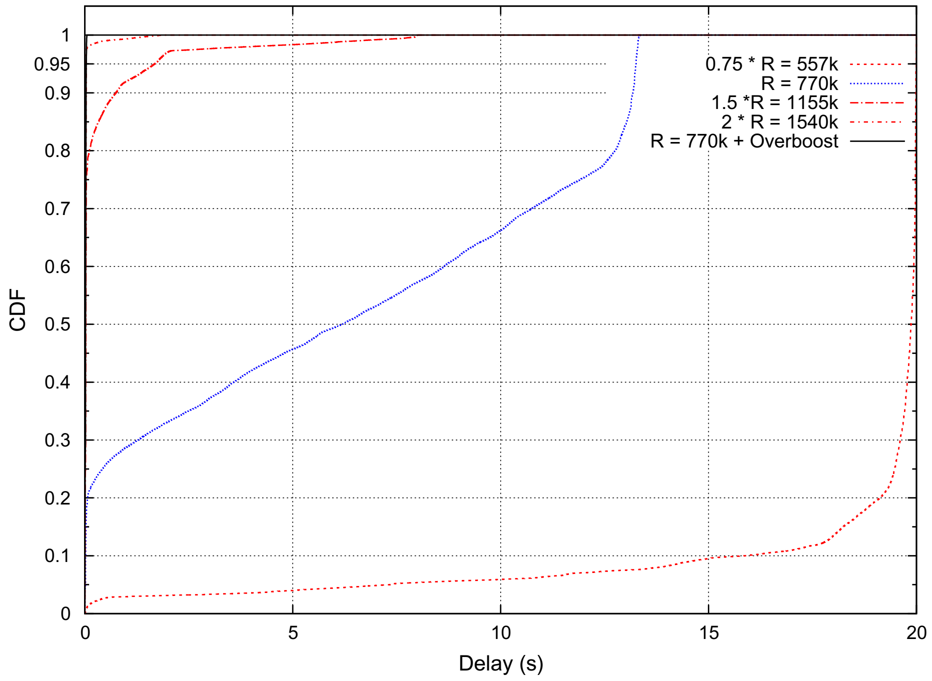


Fig. 12. CDF of access delay in the case of *overprovisioning* and *underprovisioning* versus *Overboost*.

6.7. HCCA bandwidth over-provisioning and under-provisioning versus Overboost

Finally we analyze the effects of Overboost on the number of the admitted QSTAs and on the provided service level in terms of node queue length and access delay. Usually the HCCA schedulers admit such TSs by evaluating their mean data rate or a related function. As a result, the VBR TSs can suffer delays when instant data rate is greater than mean data rate.

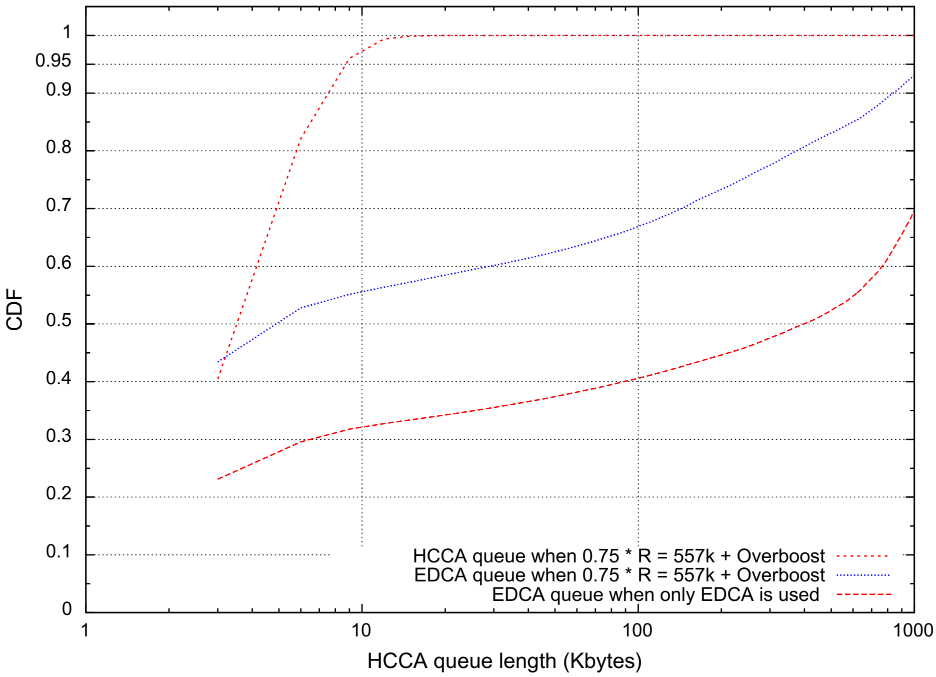


Fig. 13. CDF of the EDCA queue: EDCA function versus *underprovisioning* plus *Overboost*.

Please cite this article in press as: A.L. Ruscelli et al., Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks, Ad Hoc Netw. (2010), doi:10.1016/j.adhoc.2010.09.014

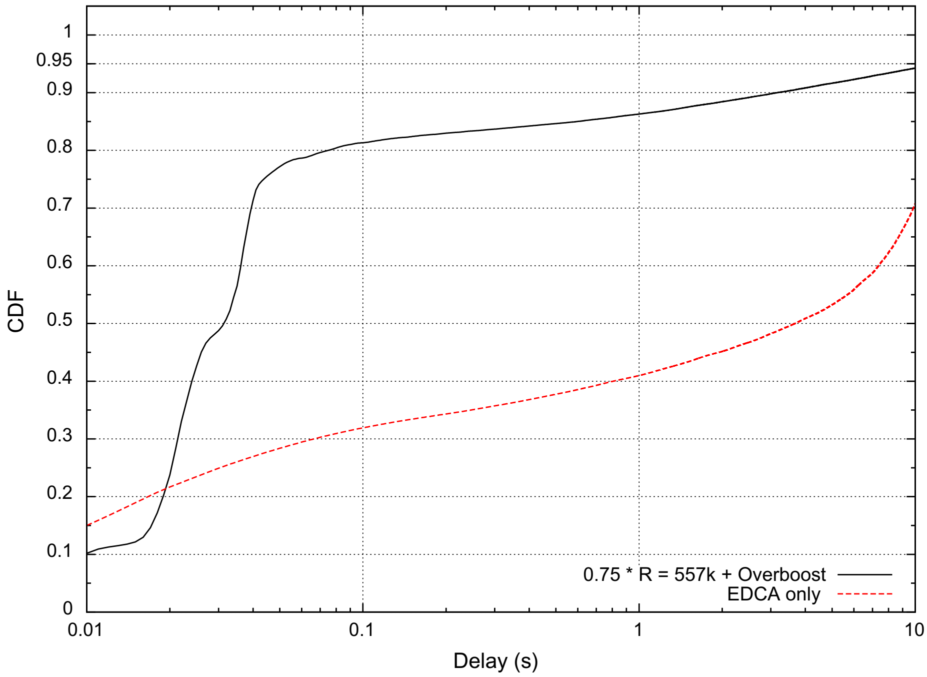


Fig. 14. CDF of the EDCA access delay: EDCA function versus *underprovisioning* plus *Overboost*.

On the other hand, when the data rate is smaller than the mean data rate, the CAP phase is automatically shortened by the protocol itself. Given the bandwidth recovery of Overboost for such kind of TSs, we analyze if Overboost can be used to admit more TSs, using an admission control test based on a value smaller than the mean data rate and keeping the same delay performance, or if it is able to reduce the experienced delay. The next scenario that we consider has an increasing number of video stream TSs. Each TS has a mean data rate of 770 Kbit/s (see Table 6). Fig. 11 shows that, when Overboost is turned off, increasing the mean data rate during the admission control phase decreases the HCCA queue length; when Overboost is turned on the queue length becomes the smallest, even when using the mean data rate.

Fig. 12 confirms these results in terms of access delay, which becomes smaller when Overboost is activated than when more bandwidth is reserved to the TSs. Thus the Overboost mechanism behaves better than the over-provisioning mechanism.

To further validate these results, we compare the overall scheduler composed by the WCBS HCCA scheduler and the Overboost local scheduler versus using the only EDCA transmission. Figs. 13 and 14 show that WCBS and Overboost reduce the queue length and the access delay with respect to EDCA only transmission, even when the network is overloaded with TS declaring a mean data rate smaller than its effective one by 25%.

7. Conclusions

In this paper we have presented, *Overboost*, a novel local node scheduler that is suitable for improving performance of HCCA schedulers of IEEE 802.11e networks by exploiting the EDCA function. Overboost switches the data traffic exceeding the HCCA transmission time limit to the queue of the highest priority EDCA access category. The analytical evaluation shows that it efficiently uses the wireless medium in presence of different types of traffic, and reduces the length of the HCCA queue. These results has been validated through simulations showing that Overboost improves the QoS support provided by the HCCA schedulers when variable bit rate traffic streams are transmitted. In particular it increases the algorithm efficiency expressed in terms of the null rate, throughput and access delay.

References

[1] IEEE802.11, Wireless LAN medium access control (MAC) and physical layer (PHY) specification, 2007.
[2] IEEE802.11b, Wireless LAN medium access control (MAC) and physical layer (PHY) specification: higher-speed physical layer extension in the 2.4 GHz band, 1999.
[3] IEEE802.11e, 802.11e-2005 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) specifications: Amendment 8: Medium access control (MAC) Quality of Service enhancements, 2005.
[4] A. Grilo, M. Nunes, Performance evaluation of IEEE 802.11e, in: Personal, Indoor and Mobile Radio Communications. The 13th IEEE International Symposium, vol. 1, Lisboa, Portugal, 2002, pp. 511–517.
[5] J. Cowling, S. Selvakennedy, A detailed investigation of the IEEE 802.11e HCF reference scheduler for VBR traffic, in: Proc. 13th International Conference on Computer Communications and Networks, Chicago, US, 2004, pp. 453–459.
[6] G. Cecchetti, A.L. Ruscelli, Performance evaluation of real-time schedulers for HCCA function in IEEE 802.11e wireless networks, in: Proc. ACM Q2SWinet, Vancouver, Canada, 2008.
[7] G.Cecchetti, A.L. Ruscelli, Real-Time Support for HCCA Function in IEEE 802.11e Networks: A Performance Evaluation, Security and

Communication Networks, 2010 (Special Issue on Security for QoS Assured Wireless Networks).

[8] Q.N.P. Ansel, T. Turletti, FHCF: a fair scheduling scheme for 802.11e WLAN, Technical Report 4883, INRIA, 2003.

[9] [9] C. Kuan, K. Dimyati, Utilization model for mixed mode, ETRI Journal 29 (2007) 829–831.

[10] V.A. Siris, C. Courcoubetis, Resource control for the EDCA and HCCA mechanisms in IEEE 802.11e networks, in: WiOpt, Boston, Massachusetts, pp. 22–27.

[11] W.K. Lai, C. Shien, C. Jiang, Adaptation of HCCA/EDCA ratio in IEEE 802.11 for improved system performance, International Journal on Innovative Computing, Information and Control 5 (2009) 4177–4188.

[12] A. Grilo, M. Macedo, M. Nunes, A scheduling algorithm for QoS support in IEEE 802.11e networks, IEEE Wireless Communications 10 (2003) 36–43.

[13] H. Fattah, C. Leung, An overview of scheduling algorithms in wireless multimedia networks, IEEE Wireless 9 (2002) 76–83.

[14] P. Ansel, Q. Ni, T. Turletti, FHCF: A simple and efficient scheduling scheme for IEEE 802.11e wireless lan, Mobile Networks and Applications 11 (2006) 391–403.

[15] C. Cicconetti, L. Lenzini, E. Mingozzi, G. Stea, An efficient cross layer scheduler for multimedia traffic in wireless local area networks with IEEE 802.11e HCCA, ACM Mobile Computing and Communications Reviews 11 (2007) 31–46.

[16] D. Skyrianoglou, N. Passas, A.K. Salkintzis, ARROW: an efficient traffic scheduling algorithm for IEEE 802.11e HCCA, IEEE Transaction on Wireless Communications 5 (2006) 3558–3567.

[17] Y.-S. Chen, Y.-W. Lee, G.-H. Huang, Modified HCCA mechanism for improving QoS support in IEEE 802.11e network, in: Mobility '08: Proceedings of the International Conference on Mobile Technology, Applications, and Systems, ACM, 2008, pp. 1–7.

[18] G. Cecchetti, A.L. Ruscelli, F. Checconi, W-CBS: a scheduling algorithm for supporting QoS in IEEE 802.11e, in: QSHINE '07: The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, ACM, Vancouver, British Columbia, Canada, 2007, pp.1–7.

[19] C. Cicconetti, L. Lenzini, E. Mingozzi, G. Stea, Design and performance analysis of the Real-Time HCCA scheduler for IEEE 802.11e WLANs, Computer Networks 51 (2007) 2311–2325.

[20] G. Boggia, P. Camarda, L.G. Grieco, S. Mascolo, Feedback-based control for providing real-time services with the IEEE 802.11e MAC, IEEE/ACM Transactions on Networking 2 (2007) 323–333.

[21] L. Yan, P-HCCA: a new scheme for real-time traffic with QoS in IEEE 802.11e based networks, in: APAN Network Research Workshop, Cairns, Australia, 2004.

[22] C.L. Liu, J.W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment, Journal of ACM 20 (1973) 46–61.

[23] T.P. Baker, Stack-based scheduling for real-time processes, Real-Time Systems 3 (1991) 67–99.

[24] L. Abeni, G. Buttazzo, Integrating multimedia applications in hard real-time sistems, in: IEEE Real Time-Systems Symposium, 1998, pp. 4–13.

[25] P.T. Brady, A model for generating on–off speech pattern in two way conversation, Bell System Technical Journal 48 (1969) 2445–2472.

[26] H. Hassan, J. Garcia, O. Brun, Generic modeling of multimedia traffic sources, in: Het-Nets, 2005.

[27] C. Cicconetti, L. Lenzini, E. Mingozzi, G. Stea, Efficient provisioning of real-time QoS guarantees in IEEE 802.11e WLANs, in: Proceedings of the European Wireless, Athens, 2006, p.7.

[28] Network Simulator 2, 1995. <http://www.isi.edu/nsnam/ns/>.

[29] C. Cicconetti, L. Lenzini, E. Mingozzi, G. Stea, A software architecture for simulating IEEE 802.11e HCCA, in: Proceedings of the IPS MoMe, Warsaw, Poland, 2005, pp. 97–104.

[30] A. Alifano, Design and implementation of a scheduler for HCCA and EDCA IEEE 802.11e networks, Master's Thesis, University of Pisa, 2010.

[31] A. Bacioccola, C. Cicconetti, G. Stea, User-level performance evaluation of voip using ns-2, in: ValueTools '07: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, Belgium, 2007, pp. 1–10.

[32] Cisco (Ed.), Traffic Analysis for Voice Over IP, Cisco Press, 2001.

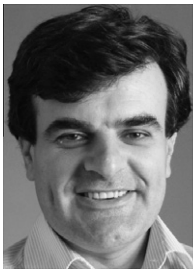
[33] <http://traces.eas.asu.edu/>, 2005.

[34] F.H.P. Fitzek, M. Reisslein, MPEG4 and H.263 video traces for network performance, IEEE Network 15 (2001) 40–54.

Anna Lina Ruscelli received the MS degree in Electronic Engineering from the University of Perugia, Italy, in 2005. She is currently a PhD student in Innovative Technologies of Information and Communications Engineering and Robotics at Scuola Superiore Sant'Anna, Pisa, Italy. Her research interests include Quality of Service support over heterogeneous networks, IEEE 802.11e networks, Wireless Sensor Networks and cross-layer design.

Gabriele Cecchetti is a researcher at the Scuola Superiore Sant'Anna University, Pisa, Italy. His current research activities span several areas, including: design and performance evaluation of scheduling algorithms for QoS provisioning in wireless computer networks with soft real-time constraints, design and evaluation of cross-layer framework architecture for providing QoS in heterogeneous wireless networks, and wireless link emulation.

Angelo Alifano received the MS degree in Computer Engineering from the University of Pisa, Italy, in 2010. His research interests are about Quality of Service support over IEEE 802.11 networks.



Giuseppe Lipari is Associate Professor of Computer Engineering at Scuola Superiore Sant'Anna. He is part of the RETIS lab of the Center of Excellence for Information Information, Communication and Perception Engineering. He is IEEE member since many years, and associate editor of IEEE Transactions on Computers. His research interests are in real-time systems, real-time operating systems, scheduling algorithms, embedded systems, wireless sensor networks. He is involved in many EU research projects (FRESCOR, RI-MACS, ARTIST 2) and Italian projects (ART-DECO, SensorNet).