



QoS in Wireless LAN: A comparison between feedback-based and earliest due-date approaches

Giuseppe Piro*, Luigi Alfredo Grieco, Gennaro Boggia, Pietro Camarda

DEE – Dip. di Elettrotecnica ed Elettronica, Politecnico di Bari, Via Orabona 4, 70125 Bari, Italy

ARTICLE INFO

Article history:

Received 19 November 2010
Received in revised form 31 May 2011
Accepted 6 October 2011
Available online 14 October 2011

Keywords:

QoS
Dynamic bandwidth allocation
Real-time applications
Wireless networks

ABSTRACT

The Hybrid Coordination Function is a very powerful mechanism to provide differentiated services in IEEE 802.11 WLANs. In recent years, its adoption in conjunction with centralized scheduling algorithms has lead to many interesting solutions to support services with very sharp delay constraints. Notably, two main approaches have been proposed to face real-time service provisioning, i.e., using the well known earliest due date scheme or feedback control theory. Unfortunately, it is still missing a deep comparison among them. To bridge this gap, we propose herein a novel feedback-based mechanism for infrastructure WLANs and compare it with respect to the most important delay-EDD algorithm, in complex scenarios with video, voice, and best effort flows. Simulation results, obtained using ns-2, demonstrate that (i) both algorithms are able to guarantee upper bounded packet delays, (ii) a more efficient bandwidth usage and a higher network throughput is achieved using the feedback based approach. In particular, it has been shown that the feedback based approach allows a throughput gain ranging from 10% to 20% with respect to delay-EDD.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Due to the growing diffusion of ubiquitous applications, the design of innovative and efficient bandwidth allocation schemes, able to differentiate the offered Quality of Service (QoS) in 802.11 Wireless LANs (WLANs) [1], has become a very challenging research topic. For this reason, the new Hybrid Coordination Function (HCF) has been recently included into the latest version of the standard, released in 2007 [1].¹ In addition, a Call Admission Control (CAC) algorithm and four Access Categories (ACs), with different priorities to map the behavior of traffic flows with users QoS requirements, have been introduced [1].

The HCF is made of a contention-based channel access, known as Enhanced Distributed Coordination Access (EDCA), and of a HCF Controlled Channel Access (HCCA), which requires a centralized controller (i.e., the Hybrid Coordinator, HC), generally located at the access point. EDCA operates as the basic DCF (Distributed coordination function) access method of a 802.11 WLAN, but using different contention parameters per AC. In this way, a service differentiation among traffic streams is statistically pursued [3]. With HCCA, the HC is responsible for assigning the right to transmit to

nodes hosting applications with QoS requirements, i.e., to perform dynamic bandwidth allocation.

Unfortunately, the 802.11 standard does not specify an effective bandwidth allocation algorithm; it only suggests a simple scheduler that uses static values declared by data sources during the call admission phase for providing a constant bit rate service. As a consequence, this scheduler is not well suited for real bursty multimedia flows [4] which will constitute the greatest part of the Internet traffic [5].

To overcome this limitation, in the last years many solutions have been proposed in literature. Among them, the algorithm based on estimated transmission times-earliest due date (SETT-EDD) [6] is considered as one of the most powerful for QoS enabled WLANs. SETT-EDD is based on the well known delay-EDD, which distributes communication resources among different flows by tacking into the account, for each one, the delay of the head of line packet [7,8]. But, in 802.11 wireless networks where the access point distributes radio resources among users, the perfect knowledge of the arrival time of packets is possible only for the downlink. In the uplink, instead, the access point is not able to know both the delays of head of line packets and the quota of bandwidth required by each flow. Thus, the SETT-EDD algorithm estimates these values by using a token-bucket model associated to each flow. However, this approach leads to an incorrect vision of the effective load of the network and, as a consequence, to a non optimal resource allocation.

The drawback of SETT-EDD can be overcome using feedback based schedulers [4]. Such schedulers, in fact, can dynamically

* Corresponding author. Tel.: +39 080 5963301; fax: +39 080 5963410.

E-mail addresses: g.piro@poliba.it (G. Piro), a.grieco@poliba.it (L.A. Grieco), g.boggia@poliba.it (G. Boggia), camarda@poliba.it (P. Camarda).

¹ Note that HCF was initially introduced by 802.11e working group in its proposal [2]. Now, this amendment has been integrated in the latest release of the standard.

List of Acronyms

ACs	Access Categories	HCF	Hybrid Coordination Function
ARF	Auto Rate Fallback	HCCA	HCF Controlled Channel Access
BIBO	Bounded Input Bounded Output	LAN	Local Area Network
CAC	Call Admission Control	MSDU	MAC Service Data Unit
CAP	Controlled Access Phase	TDMA	Time Division Multiple Access
CBR	Constant Bit Rate	TSPEC	Traffic SPECification
DCF	Distributed Coordination Function	TXOP	Transmission Opportunity
EDCA	Enhanced Distributed Coordination Access	VBR	Variable Bit Rate
EDD	Earliest Due Date	WLAN	Wireless LAN
HC	Hybrid Coordinator		

adjust the amount of bandwidth assigned to each station by exploiting the knowledge of transmission queue lengths, which, according to the 802.11 standard, are fed back in apposite header fields.

The aim of this work is to compare delay-EDD based and feedback control based schedulers in order to provide a better understanding of packet scheduling in 802.11 WLANs. More precisely, we will compare the SETT-EDD algorithm with respect to a novel feedback based strategy we propose herein, namely Guaranteed Delay Scheduler (GDS), designed following the approach proposed in [9]. Ns-2 simulations [10], carried out on a broad set of WLAN scenarios in the presence of both real-time multimedia and best effort flows, clearly show that both solutions are able to guarantee upper bounded packet delays, but a smaller network capacity is required when feedback control is adopted. In particular, we have shown that GDS made two order of magnitude of bandwidth allocated in HCCA mode with respect to SETT-EDD, thus granting for a network throughput gain ranging from 10% to 20%.

The rest of the paper is organized as follows: in Section 2 related work on WLAN scheduling algorithms are discussed; Sections 3 and 4 describe SETT-EDD and GDS allocation schemes, respectively; Section 5 shows ns-2 simulation results, and finally the last section draws conclusions and states future research.

2. Related work on scheduling in 802.11 WLANs

The latest version of the IEEE 802.11 standard, released in 2007, enriches WLANs with new QoS differentiation capabilities [1]. In particular, this result has been obtained integrating the 802.11e amendment [2]. The HCF is introduced to provide service differentiation in WLANs. As stated in the introduction, it is made of a contention-based channel access (i.e., the EDCA) and of a controlled channel access (i.e., the HCCA), which requires the HC for bandwidth allocation to nodes hosting applications with QoS requirements. At any time during two successive beacon frame (i.e., the *beacon interval*), the HC can start a Controlled Access Phase (CAP), during which only polled stations can transmit, according to the allocated transmission opportunity (TXOP).

It is important to note that, despite the MAC layer enhancements introduced by the 802.11e amendment [11], the IEEE 802.11 WLAN standard has been mainly designed for data applications and, for this reason, it does not provide any QoS support to multimedia applications [12]. As a consequence, in last years, the issue of scheduling in 802.11 WLANs has lead to many interesting proposals from the scientific community [13].

The problem has been faced in [14] with a multi-polling mechanism that allocates TXOPs to real-time flows by taking into account average and maximum source transmission rates, which are declared at admission time.

To provide adaptability to changing network conditions, a scheduling algorithm that allocates TXOPs by taking into account

both transmission queue lengths and their estimates has been developed in [15]. Basically, the algorithm estimates the actual needs of each station and then tries to transmit all pending data in the transmission queues.

In [16], a cross-layer optimization has been proposed to jointly exploit the features of HCCA and of scalable video coding algorithms. Multiple sub-flows are created from one global video flow, each with its own traffic specifications. Then, to maximize the utilization of the wireless channel, a linear-programming solution is exploited, effectively allocating the optimal TXOP to each generated sub-flow. In this way, both the number of admitted stations and the video quality at each admitted station are improved.

In [17] a new polling-based medium access control protocol, namely Unified Point coordination Function, is reported. It has been designed to furnish QoS guarantees to multimedia applications in WLANs. At admission time, each flow declares its own need by means of a guaranteed TXOP, which is computed using the one-sided Chebyshev inequality. With this method, the guaranteed TXOP will be larger than the actual transmission needs with a given probability, which is a key parameter of the algorithm. During the data transfer, instead, each application can ask for a demanded TXOP, which is different from the guaranteed one. The actual allocated TXOP is computed by taking into account the guaranteed one, the declared one, and the total network load.

In [18], the priority-oriented adaptive control with QoS guarantee is developed as centralized channel access mechanism able to support multimedia applications in WLAN. It provides delay and jitter guarantees for both constant bit rate and variable bit rate traffic. This access mechanism is not based on polling, but on a TDMA scheme. The access point uses the beacon signal to inform the stations of the assigned slots for the current beacon interval. The number of slots assigned to a traffic stream can be dynamically varied upon the reception of a request from the station hosting that stream. The final objective is to support as many traffic streams as possible at the best possible quality.

The scheduling scheme, known as prediction and optimization-based HCCA, is illustrated in [19]. It adapts dynamically TXOPs by taking into account the actual load. In particular, the TXOP adaptation exploits a variable bit rate traffic prediction based on wavelet and an on-line optimization. The effectiveness of the approach has been demonstrated only in one simple WLAN scenario with three stations.

In [20], a static TXOP scheduling algorithm has been proposed to provide bounded delays. Being static, it cannot track the real-time needs of data flows, thus loosing the gain of statistical multiplexing. Furthermore, it has not been validated using realistic network scenarios.

In order to provide a further comparison among the wide variety of aforementioned approaches, Tables 1 and 2 report scheduling targets and parameters used for computing TXOPs, respectively.

Table 1
Explicit scheduling targets.

Name	High channel utilization	Delay guarantees	Fairness	Maximization of admitted flows	Video quality improvement	Power conservation
Contention-based multipolling mechanism [14]	x	x				
FHCF [15]		x	x			
Optimized scalable video streaming [16]				x	x	
UPCF [17]		x				x
POAC-QG [18]	x					
PRO-HCCA [19]		x				
Equal-SP [20]	x					

Table 2
Parameters for TXOPs computation.

Name	TSPEC	Instantaneous source rate	Source rate prediction	Mean sending rate	Queue length	Standard compliant
Contention-based multipolling mechanism [14]	x					NO
FHCF [15]	x			x	x	YES
Optimized scalable video streaming [16]	x					YES
UPCF [17]	x					NO
POAC-QG [18]	x	x				NO
PRO-HCCA [19]	x		x			YES
Equal-SP [20]	x					YES

Looking at them, it is easy to see that, while all considered scheduling strategies take into account TSPECs in TXOP computation, only algorithm in [15,18,19] are able to encompass typical source rate oscillations of VBR real-time applications. We note also that some approaches are not compliant with the standard, e.g., those proposed in [14,17,18]. Finally, we remark that only algorithms proposed in [14,15,17,19] are able to provide delay guarantees, but none of them is able to guarantee any upper bound on the packet delay.

In [4,21], a novel approach for dynamic TXOP allocations has been proposed. The idea is to exploit packet headers to carry transmission queue lengths of QoS capable stations, as proposed in the 802.11 standard. Using this signal as feedback, the HC can properly allocate TXOPs exploiting a linear regulator. In these studies, the effectiveness of proportional, proportional integral, and proportional integral derivative regulators has been investigated. Theoretical analysis have demonstrated how regulator parameters have to be tuned to avoid control loop instability. Anyway, absolute delay bounds were evaluated only using computer simulations. In other words, the proposed feedback based dynamic schedulers are able to significantly outperform the scheduler proposed in the 802.11 standard, but absolute performance bounds cannot be known in advance. To bridge this gap, the present work proposes the use of the GDS algorithm. The innovation lies in the design strategy and in the queuing model, which is valid also in the presence of time-varying sampling intervals. It is also important to note that GDS timing rules are very simple and not depend on the network scenario.

To study the effectiveness of GDS in the considered application scenario, we have compared it with respect to the SETT-EDD scheme, because delay-EDD algorithms are considered as close to optimal scheduling strategy in a wide set of real-time scheduling problems [8].

3. The SETT-EDD scheduler

SETT-EDD represents a realistic implementation in an 802.11 wireless network of the well known delay-EDD algorithm. It allows the HC to decide, at the beginning of each CAP, the order of the stations to poll and the related TXOP assignments.

SETT-EDD supposes there are S active stations, each one hosting one or more traffic streams. Let N_i be the number of active traffic streams for the i th station, i.e., the whole number of traffic streams in the network is $N = \sum_i N_i$.

It is important to note that all decisions are carried out considering the traffic specification (TSPEC) declared by stations for all active traffic streams at admission time; they are expressed as aggregate parameters. The most important of them are:

- Minimum TXOP duration (mTD): the time required to transmit a MAC Service Data Unit (MSDU) with the maximum size at the minimum physical data rate.
- Maximum TXOP duration (MTD): the time required to transmit the aggregate burst size of a station at the minimum physical data rate. The aggregate burst size is equal to the sum of the burst size of all the flows belonging to a station. Moreover, the burst size represents the maximum burst of MSDUs that a flow can generate at the peak data rate.
- Number of MSDUs (F): a number of MAC frames that a flow generates at the mean data rate during the service interval. For each flow, the service interval is obtained as the ratio between the nominal MSDU size and the mean data rate declared into its TSPEC.
- TXOP duration ($TXOP_i$): it represents a fixed duration of a TXOP, which is calculated for each station. In particular, for the i th station, $TXOP_i$ is obtained as:

$$TXOP_i = \sum_{j=1}^{N_i} \max \left(\frac{F_j \cdot L_j}{C_m} + O_j, \frac{M_j}{C_m} + O_j \right), \quad (1)$$

where O is the time overhead due to ACK packets and interframe spaces (see the 802.11 standard [1] for details about these interframe intervals), L_j (M_j) is the Nominal (Maximum) MSDU size of the j th flow, and C_m is the minimum physical data rate.

- Minimum service interval (mSI): the minimum time interval between the beginning of two consecutive TXOPs allocated to the same station. It is equal to the smallest service interval for any flow of given station.
- Maximum service interval (MSI): the maximum time allowed between the starting of successive TXOPs allocated to the same

station. As described in [6], it is computed considering the lowest delay bound (D) among all the station's flows. It is obtained as:

$$MSI = (D - MTD). \quad (2)$$

When the HC decides to poll a station, it computes a TXOP whose duration is obtained dynamically considering some aggregate service parameters as described before. It is worth to note that the definition of the TXOP duration is the most important aspect of SETT-EDD algorithm.

The HC handles a TXOP timer for each station. The timer increases at a constant rate equal to TD/mSI and it is set to 0 every time the station receives a new poll. When the HC decides to poll the i th station, the TXOP duration is deducted from the TXOP timer associated to that station. Let Δt_i be the time interval between the last polling time of the i th and the current time instant. The TXOP timer (i.e., Γ_i) associated to that station and the related TXOP duration are obtained by the following equations.

$$\Gamma_i = \frac{TXOP_i}{mSI_i} \Delta t_i, \quad (3)$$

$$TXOP_i = \begin{cases} 0, & \text{if } \Gamma_i < mTD_i, \\ \Gamma_i, & \text{if } mTD_i \leq \Gamma_i \leq MTD_i, \\ MTD_i, & \text{if } \Gamma_i > MTD_i. \end{cases} \quad (4)$$

According to Eq. (4), TXOP is not allocated if the timer is less than mTD because this grants that at least one packet is transmitted during a TXOP. Furthermore, the maximum TXOP is less than the MTD defined for a given station.

To implement a delay-EDD algorithm, SETT-EDD assigns to each station a deadline. The HC tries to poll stations before the expiration of their deadlines. In particular, if the station receives a poll at time t , the time deadline is set to $t + MSI$.

At the beginning of each CAP, the HC orders stations based on their deadlines. Then, it assigns TXOPs, starting from those stations whose time deadline is going to expire. The allocation in the current CAP will finish when the sum of allocated TXOPs reach the part of the CAP used with HCCA.

To summarize the working of this allocation scheme, the SETT-EDD pseudo code has been reported in Algorithm 1.

Algorithm 1. SETT-EDD pseudo code

```

INPUT:
stations, a list of  $S$  stations hosting uplink flows;
 $deadLine_i$ , a time dead-line associated to the station  $i$ , with
 $i = 1, \dots, S$ ;
00: BEGIN
01: FOR each CAP DO
02:   WHILE CAP ends DO
03:     FIND  $station_i$  IN  $stations$  whose  $deadLine_i$  is going
       to expire
04:     update the TXOP timer  $\Gamma_i$  of the  $station_i$  using Eq.
       (3)
05:     IF  $\Gamma_i \geq mTD_i$ 
06:       compute  $TXOP_i$  using Eq. (4)
07:       assign  $TXOP_i$  TO  $station_i$ 
08:        $\Gamma_i = 0$ 
09:        $deadLine_i \leftarrow now + MSI_i$ 
10:     END IF
11:   END WHILE
12: END FOR
13: END

```

It is worth to note that, SETT-EDD assigns radio resource according to the TSPEC associated to flows hosted into each station, without the explicit knowledge of the actual bandwidth needs.

4. The guaranteed delay scheduler

Following a methodology similar to the one proposed in [9], GDS has been designed using the linear discrete-time control theory. It is assumed an 802.11 wireless network made by: a shared channel, N transmission queues (i.e., N traffic streams), and a scheduling algorithm. Such an algorithm, knowing transmission queue levels, should properly distribute the channel bandwidth with the goal of the provisioning of strict bounds on queuing delays. Unlike [9], the algorithm proposed herein is centralized and, as a consequence, a stronger enforcement on delay bounds can be provided.

To distribute the transmission opportunities scheduled by GDS, we suppose that HC starts a CAP every T_{CA} seconds (i.e., T_{CA} is the time interval between two consecutive CAPs). In the sequel of the work, the impact of the T_{CA} value on the system performance will be analyzed both theoretically and using simulations. In the time intervals between the ending of a CAP and the starting of the next one, stations will access to the channel using standard EDCA.

In detail, each T_{CA} seconds, the HC computes, using GDS, the bandwidth that should be allocated to each station during the upcoming CAP. In this way, each real-time application will receive the opportunity to transmit every T_{CA} .

In particular, for each flow, HC evaluates the TXOP duration by using the GDS control law (see below for details). To this aim, the HC needs to know transmission queue levels of wireless stations, which are feed back in packet headers according to the IEEE 802.11 standard. To reduce bandwidth waste, stations advertising empty queues are not served.

In the next section, the system model used for the GDS design and the control law used to compute the TXOP in a centralized environment are explained. Moreover, starting from this model, stability, complexity, and performance bounds of GDS will be also theoretically studied and demonstrated.

4.1. The system model

As stated before, there are N active traffic flows sharing the wireless channel. Associated to each flow, there is a queue, where packets are stored waiting for transmission. GDS scheduler evaluates the transmission needs of each queue in terms of TXOPs.

Let $\hat{t}_i(k)$ be the starting time instant of the k th TXOP of the i th queue. Thus, the k th transmission opportunity ends at the time instant $t_i(k+1) = TXOP_i(k) + \hat{t}_i(k)$ (see Fig. 1). We will consider such an instant $t_i(k)$ as the sampling instant in our system, i.e., $\Delta t_i(k) = t_i(k+1) - t_i(k)$ is the sampling interval.

Now, the following equation holds:

$$q_i(k+1) - q_i(k) = d_i(k) - u_i(k), \quad (5)$$

where

- $q_i(k)$ is the i th queue length at time $t_i(k)$;
- $q_i(k+1)$ is the i th queue length at time $t_i(k+1)$;

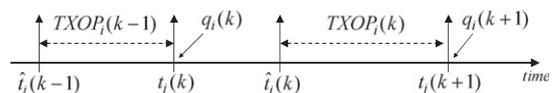


Fig. 1. Transmission opportunities of the i th queue.

- $u_i(k)$ corresponds to the amount of data that will be transmitted during the $TXOP_i(k)$ (i.e., the k th transmission opportunity);
- $d_i(k)$ is the amount of data that filled the queue during the time interval $[t_i(k), t_i(k+1)]$, i.e., it models the behavior of the data source feeding the i th queue.

It is worth noting that $TXOP_i(k)$ can be expressed as a function of $u_i(k)$ and vice versa, by taking into account protocol overhead, transmission rates, and packet size.

In the following, we will refer to $Q_i(z)$, $D_i(z)$, and $U_i(z)$ as the Z -transforms [22] of the signals $q_i(k)$, $d_i(k)$, and $u_i(k)$, respectively.

4.2. The control law

Our scheduling algorithm is based on a control law that has to compute, at each time instant $\hat{t}_i(k)$, the k th transmission opportunity duration $TXOP_i(k)$ (i.e., the interval duration which can be used by the i th queue for transmission), or, equivalently, $u_i(k)$. Such a control law should be properly designed in order to provide bounded delays to transmitted packets, assuring, at the same time, BIBO stability [22] to the system defined by Eq. (5).

We will assume the following general control law:

$$u_i(k) = h_i(k) * q_i(k) \quad (6)$$

where the '*' operator represents the discrete time convolution [22].

Eq. (6) means that the amount of data to be transmitted during the k th transmission opportunity is obtained by filtering the signal $q_i(k)$ (i.e., the queue level) through a time-invariant linear filter with pulse response $h_i(k)$ or, equivalently, with transfer function $H_i(z) = \mathcal{Z}[h_i(k)]$ [22].

Combining Eqs. (5) and (6), we obtain that our scheduling algorithm realizes the control loop shown in Fig. 2, with the set point $q_i^T = 0$. This means that our control algorithm tries to target empty queues using a linear regulator with transfer function $H_i(z)$.

In the following, the pulse response of the system will be referred to as $h_{s_i}(k)$, so that the following equality holds:

$$q_i(k) = h_{s_i}(k) * d_i(k). \quad (7)$$

Assuming $q_i(0) = 0$ (i.e., empty queues at the beginning), our design strategy is to find the proper function $H_i(z)$ that ensures BIBO stability to the system and guaranteed queuing delays. We will prove that these constraints could be fulfilled if the closed-loop response to the Kronecker pulse $\delta(k)$ [22] (i.e., the system pulse response) has the following expression:

$$h_{s_i}(k) = \sum_{n=0}^{M_i} c_i(n) \delta(k-n), \quad (8)$$

where $c_i(n)$ are real finite coefficients, i.e., $c_i(n) \in \mathbb{R}$.

The system transfer function $H_{s_i}(z)$ is by definition just the Z -transform of the system pulse response, $h_{s_i}(k)$, assuming $q_i(0) = 0$. With reference to Fig. 2, we have

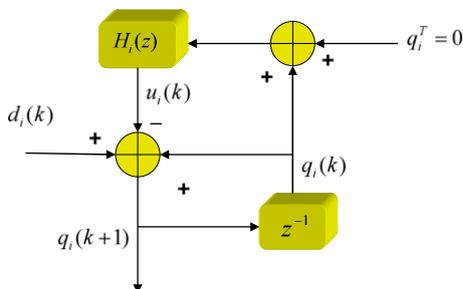


Fig. 2. Control loop of the bandwidth allocation algorithm.

$$H_{s_i}(z) = \frac{Q_i(z)}{D_i(z)} = \frac{1}{z-1+H_i(z)} = \mathcal{Z}[h_{s_i}(k)]. \quad (9)$$

Considering Eq. (8), then:

$$\mathcal{Z}[h_{s_i}(k)] = \mathcal{Z} \left[\sum_{n=0}^{M_i} c_i(n) \delta(k-n) \right] = \sum_{n=0}^{M_i} c_i(n) z^{-n}. \quad (10)$$

Thus, combining Eqs. (9) and (10), Eq. (8) holds when the following transfer function for the controller is used:

$$H_i(z) = \frac{U_i(z)}{Q_i(z)} = \frac{(1-z) \sum_{n=0}^{M_i} c_i(n) z^{-n} + 1}{\sum_{n=0}^{M_i} c_i(n) z^{-n}}. \quad (11)$$

Now, we can demonstrate that with this controller (i.e., the GDS controller) we can ensure BIBO stability and bounded delays.

First of all, it is straightforward to show that Eq. (8) ensures BIBO stability [22]. In fact:

$$\sum_{k=0}^{+\infty} |h_{s_i}(k)| = \sum_{k=0}^{M_i} |c_i(k)| < +\infty. \quad (12)$$

We have to prove also that Eq. (8) can guarantee bounded delays by a proper selection of coefficients $c_i(n)$.

Considering Eq. (8), obviously the queue response cannot be negative. If we consider a Kronecker pulse as input to the queue, this means that $h_{s_i}(k)$ should be not negative, i.e., $h_{s_i}(k) \geq 0$. Therefore, it is necessary that $c_i(n) \geq 0$. Moreover, the queue cannot contain more data than its input (i.e., a pulse with width equal to 1); that is $h_{s_i}(k) \leq 1$, which is equivalent to require $c_i(n) \leq 1$.

To guarantee the system causality, we have to set $c_i(0) = 0$ and $c_i(1) = 1$. In fact, a pulse of data arriving during the first sampling interval $[t_i(0), t_i(1)]$ will be enqueued during that interval. It will be transmitted not before the second sampling interval $[t_i(1), t_i(2)]$. In other words, assuming at time $t=0$ an empty queue, i.e., $q_i(0) = 0$, and a single data pulse as system input, i.e., $d_i(k) = \delta(k)$, we have to impose that the queue is filled only by the data pulse at time $t=1$, i.e., $q_i(1) = 1$. This means, equivalently, that it should be $c_i(0) = 0$ and $c_i(1) = 1$ in Eq. (8).

With reference to Eq. (5), even considering the Kronecker pulse as system input and that $c_i(0) = 0$, we have:

$$\begin{aligned} u_i(k) &= d_i(k) + q_i(k) - q_i(k+1) \\ &= \delta(k) + \sum_{n=0}^{M_i} c_i(n) \delta(k-n) - \sum_{n=1}^{M_i} c_i(n) \delta(k+1-n), \end{aligned} \quad (13)$$

that is, after a bit of algebra:

$$u_i(k) = c_i(M_i) \delta(k-M_i) + \sum_{n=1}^{M_i-1} [c_i(n) - c_i(n+1)] \delta(k-n). \quad (14)$$

Now, considering that $u_i(k)$ should not be negative, it holds that $c_i(n) \geq c_i(n+1)$ for $n \geq 1$.

From the Eq. (14), it is clear that the choice of the $c_i(n)$ coefficients affects the way the bandwidth is allocated during the transient. In fact, each pulse of data entering the queue is spread over M_i transmission opportunities. Moreover, the amount of data transmitted during each transmission opportunity depends on the set of $c_i(n)$ coefficients. This can help the designer to tailor the control law trying to cast GDS in other technological context.

To summarize the analysis on coefficients $c_i(n)$, we have to impose in Eqs. (8) and (11) the constraints:

$$0 \leq c_i(n) \leq 1 \quad \forall n; \quad c_i(n) \geq c_i(n+1), \quad n \geq 1. \quad (15)$$

Now, we can prove that these constraints are able to provide upper bounded queuing delays. In particular, we can show that delays can be smaller than $M_i + 1$ sampling intervals. This requires that the queue backlog measured in $t_i(k+1)$ will be transmitted

in at most $M_i + 1$ sampling interval. In this way, a generic packet that entered the queue during the time interval $[t_i(k); t_i(k + 1)]$ will wait in queue for at most $M_i + 1$ sampling intervals. This can be expressed as:

$$\sum_{n=0}^{M_i} u_i(k+n) \geq q_i(k) \quad \forall k \geq 0, \quad (16)$$

which, by considering Eq. (5), can be equivalently rewritten as:

$$\sum_{n=0}^{M_i} d_i(k+n) \geq q_i(k+M_i+1) \quad \forall k \geq 0. \quad (17)$$

Considering Eq. (8), we obtain:

$$q_i(k) = h_{s_i}(k) * d_i(k) = \sum_{n=0}^{M_i} c_i(n) d_i(k-n). \quad (18)$$

Substituting Eq. (18) in (17), Eq. (17) is equivalent to the following inequality:

$$\sum_{n=0}^{M_i} d_i(k+n) \geq \sum_{n=0}^{M_i} c_i(n) d_i(k+M_i-n+1). \quad (19)$$

Imposing $m = M_i - n + 1$, it becomes:

$$d_i(k) + \sum_{n=1}^{M_i} d_i(k+n) \geq \sum_{m=1}^{M_i} c_i(M_i-m+1) d_i(k+m) \quad (20)$$

that is

$$d_i(k) + \sum_{n=1}^{M_i} [1 - c_i(M_i-n+1)] d_i(k+n) \geq 0. \quad (21)$$

Remembering that $d_i(k) \geq 0$ and $0 \leq c_i(n) \leq 1$, the last inequality (21) holds for all values of k . This proves that with the GDS controller, delays are smaller than $M_i + 1$ sampling intervals.

It is also possible to estimate the upper bound of the queuing delay, T_i^M , corresponding to $M_i + 1$ consecutive sampling intervals. We assume that each station is polled by the HC with a constant sampling time equal to the interval T_{CA} . It is important to note that GDS has been designed to spread each burst of data over M

consecutive transmission opportunities. Now, given that transmission opportunities are granted by the coordination function in each CAP, the packet delay depends only on T_{CA} . In this way, the target delay can be expressed as:

$$T_i^M = (M_i + 1) T_{CA}. \quad (22)$$

Thus, the proposed allocation algorithm assures that $\tau_i \leq T_i^M$.

To summarize the results, if we consider the GDS scheme using a controller with the transfer function (11) and the constraints on $c_i(n)$ coefficients given by Eq. (15), we can ensure BIBO stability to the system and guaranteed delays. A proper choice of the coefficients of the transfer function allow us to obtain the best behavior in terms of the average value for the delay and the average/maximum channel utilization.

4.3. TXOP assignments

After computing the bandwidth, GDS has to transform $u_i(k)$ into a $TXOP_i(k)$ assignment, according to the 802.11 standard.

If the packets in the i th queue are transmitted at rate C_i , the following relation is considered:

$$TXOP_i(k) = \frac{|u_i(k)|}{C_i} + O(k), \quad (23)$$

where $TXOP_i(k)$ is the TXOP assigned to the i th queue during the k th successful contention, and $O(k)$ is the time overhead due to ACK packets and interframe spaces (see the 802.11 standards [1] for details about these interframe intervals).

4.4. Computational complexity of GDS

The total computational complexity of GDS is $O(NM)$ where $M = \max_i\{M_i\}$ with $i = 1, \dots, N$.

In fact, GDS evaluates the amount of data that should be transmitted by each queue. Hence, its computational complexity only depends on the computation of the signal $u_i(k)$. By considering Eq. (11) in the time domain, the control law can be expressed as follows:

$$u_i(k) = q_i(k) + \sum_{n=2}^{M_i} [q_i(k-n+1) - q_i(k-n+2) - u_i(k-n+1)] c_i(n). \quad (24)$$

From Eq. (24), it is clear that, for each transmission queue, this computation requires $(M_i - 1)$ multiplications and $3(M_i - 1) + 1$ sums; that is, the computational complexity for each queue is $O(M_i)$. As a consequence, if in the wireless system there are N transmission queues, the total computational complexity is $O(NM)$ where $M = \max_i\{M_i\}$ with $i = 1, \dots, N$.

4.5. Channel saturation control

The above bandwidth allocation algorithm is based on the implicit assumption that the sum of the TXOPs assigned to each traffic stream is smaller than the $dot11CAPlimit$ value which is the maximum length of CAPs and is advertised periodically by the HC. This value can be violated when the network is saturated.

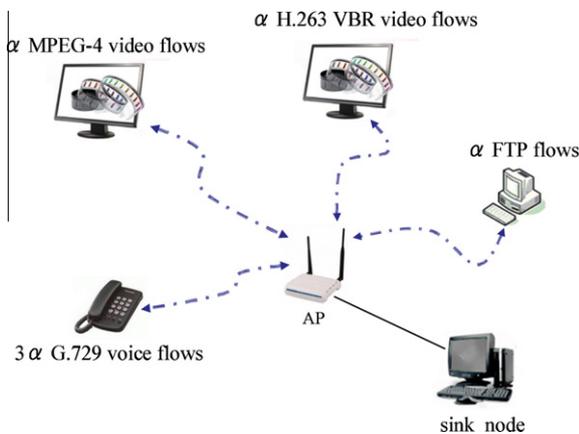


Fig. 3. Scenario with multimedia flows.

Table 3

Main features of the considered multimedia flows.

Type of flow	Nominal MSDU [byte]	Maximum MSDU [byte]	Burst size [byte]	Mean data rate [kbps]	Peak data rate [kbps]
MPEG-4 HQ	1536	2304	16745	770	3300
H.263 VBR	1536	2304	18168	450	3400
G.729 VAD	60	60	60	8.4	24

In the presence of a new admission request for a traffic stream, the admission control unit calculates the nominal transmission opportunity needed by the stream as proposed by the standard [1]. We choose this standard CAC in order to have a fair comparison with respect to the SETT-EDD scheduler.

It is important to note that the CAC scheme proposed by the standard is able to avoid heavy channel saturations because it is based on the declared average rate of each source. However, since the transient network overloads (due to the burstiness of the multimedia flows) cannot be avoided, a saturation control function has been designed.

Considering the k th time interval T_{CA} , a channel saturation occurs if and only if the following condition is verified:

Table 4
Target delays for GDS and SETT-EDD.

T_{CA}	$D = 4T_{CA}$	$D = 6T_{CA}$
15 TU	61.44 ms	92.16 ms
29 TU	118.78 ms	178.18 ms

Table 5
Percentage of Admitted Flows when $\alpha = 12$ and $T_{CA} = 15$ TU [%].

Loss rate in the BAD state	GDS		SETT-EDD	
	$T_i^M = 4T_{CA}$	$T_i^M = 6T_{CA}$	$T_i^M = 4T_{CA}$	$T_i^M = 6T_{CA}$
0	62.62 ± 2.81	62.33 ± 2.20	62.12 ± 2.50	62.56 ± 2.12
0.001	63.66 ± 6.36	62.46 ± 6.23	63.21 ± 3.93	63.21 ± 3.93
0.01	62.22 ± 3.31	64.14 ± 5.06	63.33 ± 7.62	63.33 ± 7.62
0.1	64.71 ± 5.21	63.03 ± 4.03	63.91 ± 5.77	63.91 ± 5.77

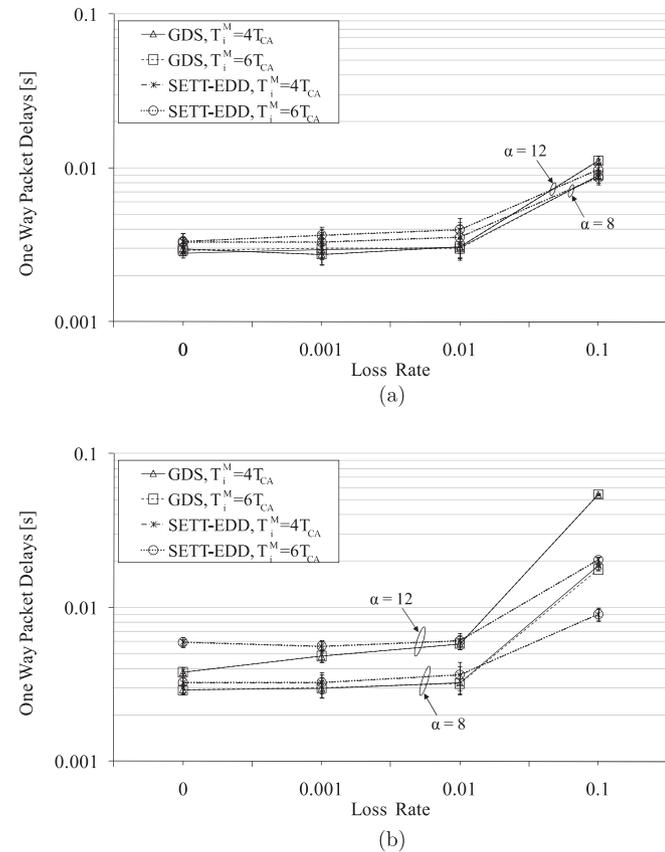


Fig. 4. 95th percentile delays of MPEG4 flows with (a) $T_{CA} = 15$ TU, (b) $T_{CA} = 29$ TU.

$$\sum_{i=1}^N TXOP_i(k) > dot11CAPlimit, \tag{25}$$

when N is the number of active flows.

If a channel saturation occurs, the computed $TXOP_i(k)$ is decreased by $\Delta TXOP_i(k)$, so that the following capacity constraints is satisfied:

$$\sum_{i=1}^N TXOP_i(k) - \Delta TXOP_i(k) = dot11CAPlimit. \tag{26}$$

In particular, the generic amount $\Delta TXOP_i(k)$ is evaluated as a fraction of the total amount $\Delta = \sum_{i=1}^N TXOP_i(k) - dot11CAPlimit$, the weight being the transmission data rate C_i ; that is:

$$\Delta TXOP_i(k) = \frac{TXOP_i(k)C_i}{\sum_{j=1}^N TXOP_j(k)C_j} \Delta. \tag{27}$$

Notice that Eq. (27) provides a $\Delta TXOP_i(k)$, which is proportional to $TXOP_i(k)C_i$; in this way, connections transmitting at low rates are not too much penalized.

5. Performance evaluation

In this section we compare the performance of GDS and SETT-EDD schedulers using the ns-2 simulator. Similarly to [4], we have considered a complex scenario (see Fig. 3) consisting of a 802.11 g WLAN shared by a mix of 3α voice flows encoded with the G.729 standard, α MPEG-4 encoded video flows, α H.263 video flows, and α FTP flows. Therefore, in such a scenario the traffic load is proportional to the parameter α .

For the video flows, we have used traffic traces available from the video trace library [23]. For voice flows, we have modeled

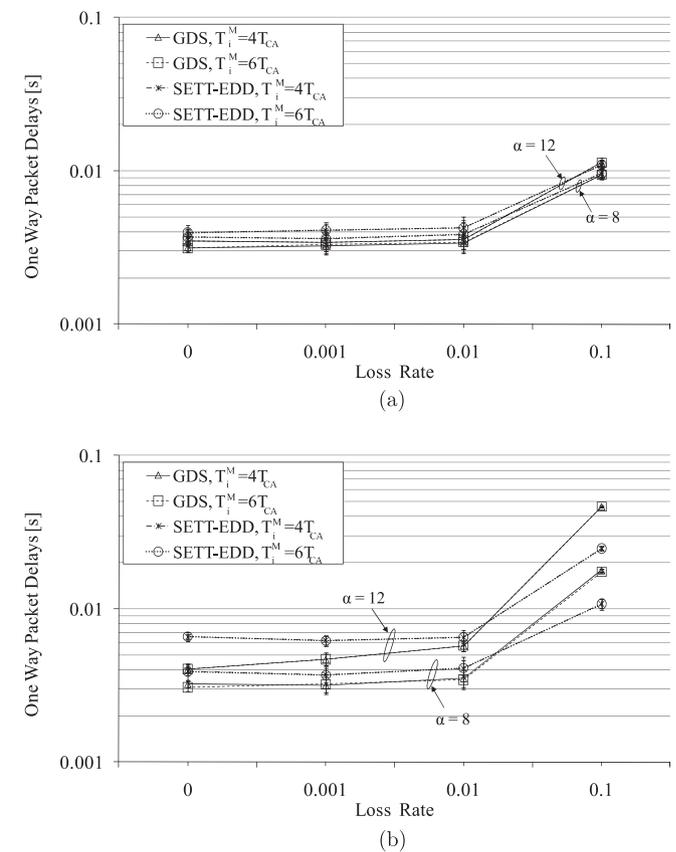


Fig. 5. 95th percentile delays of H263 flows with (a) $T_{CA} = 15$ TU, (b) $T_{CA} = 29$ TU.

G.729 sources using Markov ON/OFF sources, where the ON period is exponentially distributed with mean value 3 s, and the OFF period has a truncated exponential pdf with an upper limit of 6.9 s and an average value of 3 s [24]. During the ON period, the source sends 20 bytes long packets every 20 ms (i.e., the source data rate is 8 kbps; we are also considering two G.729 frames combined into one packet [25]). Considering the overheads of the RTP/UDP/IP protocol stacks, the rate is 24 kbps during the ON periods. During the OFF period the rate is zero because we assume the presence of a Voice Activity Detector. Each wireless node hosts a single multimedia or FTP sender. The HC is connected through a wired link with a sink node, hosting receivers. This link has capacity equal to 100 Mbps, propagation delay equal to 20 ms, and a maximum queue size equal to 50 packets. The main characteristics of the considered multimedia flows are summarized in Table 3.

To study the effect of random bursty losses, we have considered a Gilbert two state discrete Markov chain for modeling the loss process affecting the WLAN channel [26]. In particular, we assume a frame loss probability equal to 0, when the channel is in the Good state, and ranging from 0 to 0.1 when the channel is in the Bad state. The permanence times in the Good and Bad states are geometrically distributed with mean values equal to 0.1 s and 0.01 s, respectively. When a MAC frame is lost, it is retransmitted up to 7 times.

The data rate has been set to 54 Mbps for all wireless stations. During CAPs, stations access to the channel using the HCCA method, otherwise they use EDCA. In simulations, EDCA parameters have been set as suggested in [1]. According to 802.11 standard, in our ns-2 implementation T_{CA} is expressed in Time Unit (TU), which is equal to 1024 μ s [1]; we have studied scheduler performances with T_{CA} equal to 15 and 29 TU.

For both GDS and SETT-EDD, the target delay T_i^M has been set to either $4T_{CA}$ or $6T_{CA}$, for all real-time flows. These values correspond to set M_i to 3 or 5 in GDS, according to Eq. (22). Target delays in all operative conditions are reported in Table 4.

In order to grant that each burst of data is spread uniformly over the M transmission opportunities, or equivalently $h_s(k)$ has a linear shape, the following $c_i(n)$ coefficients have been chosen for the GDS algorithm:

$$c_i(0) = 0; \quad c_i(n) = 1 - \frac{n-1}{M_i} \quad \text{with } n = 1, \dots, M_i. \quad (28)$$

In order to allow the call admission phase of new starting flows, the value of the system variable *dot11CAPlimit* has been set so that at least 10 frames of maximum size can be transmitted using EDCA, between the starting of two successive CAPs. Before data transmission, a multimedia source has to set up a new Traffic Stream. If the reply to the stream admission message is not received within a timeout interval, σ_{TO} , the request is repeated up to a maximum number, N_{Adm} , of times; as in [4], we have chosen $N_{Adm} = 10$ and $\sigma_{TO} = 1.5$ s. If after N_{Adm} admission tries no reply is received back, then the request is considered lost and a new admission procedure is initiated after an exponential distributed random time σ_{defer} with average value equal to 1 min. When a flow terminates, it is withdrawn from the polling list by the HC when no more packets of that flow are received for a time equal to the Inactivity Interval, which is equal to 3 s for video flows and 10 s for voice flows.

Each simulation lasts 800 s and all simulation results are averaged over 5 simulations. For all the obtained results, also the 95% confidence interval is reported.

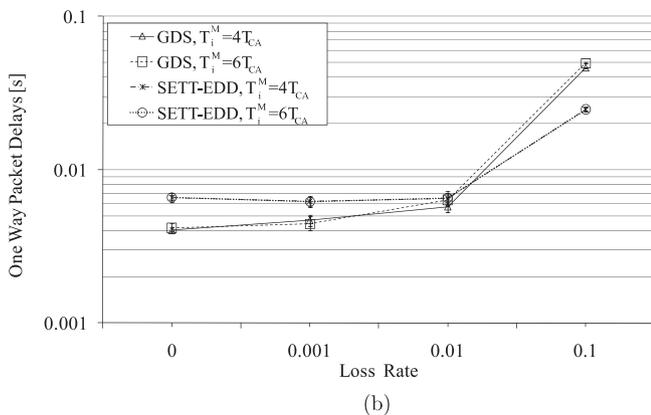
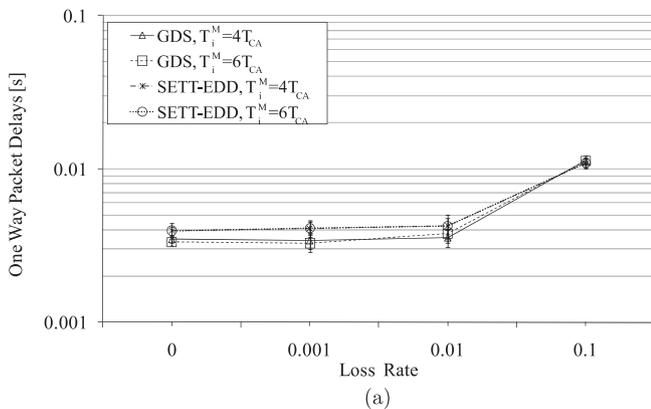


Fig. 6. 95% percentile delays of G729 flows with (a) $T_{CA} = 15$ TU, (b) $T_{CA} = 29$ TU, when $\alpha = 12$.

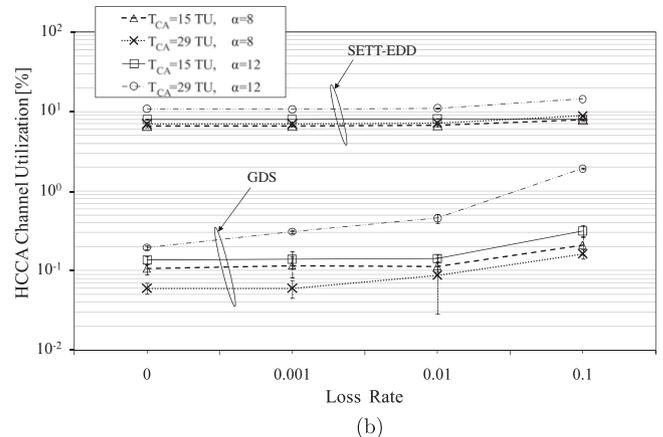
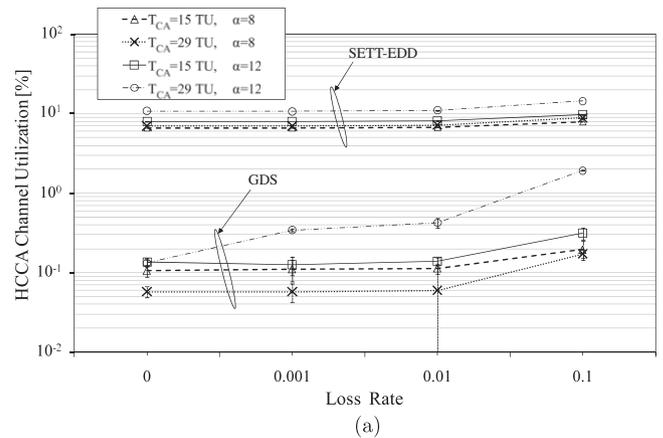


Fig. 7. Average HCCA channel utilization when (a) $T_i^M = 4T_{CA}$, (b) $T_i^M = 6T_{CA}$.

5.1. Comparison between GDS and SETT-EDD

In this section, GDS and SETT-EDD algorithms will be compared in scenarios with α equal to 8 and 12 (i.e., at average and high load conditions), for T_{CA} equal to 15 and 29 TU.

We found that both schedulers admit the same quota of flows in all considered scenarios. This result was expected because the same CAC has been adopted for both GDS and SETT-EDD. In particular, the ratio of admitted flows is equal to 100% in all scenarios other than the one with $\alpha = 12$ and $T_{CA} = 15$ TU. In the latter case

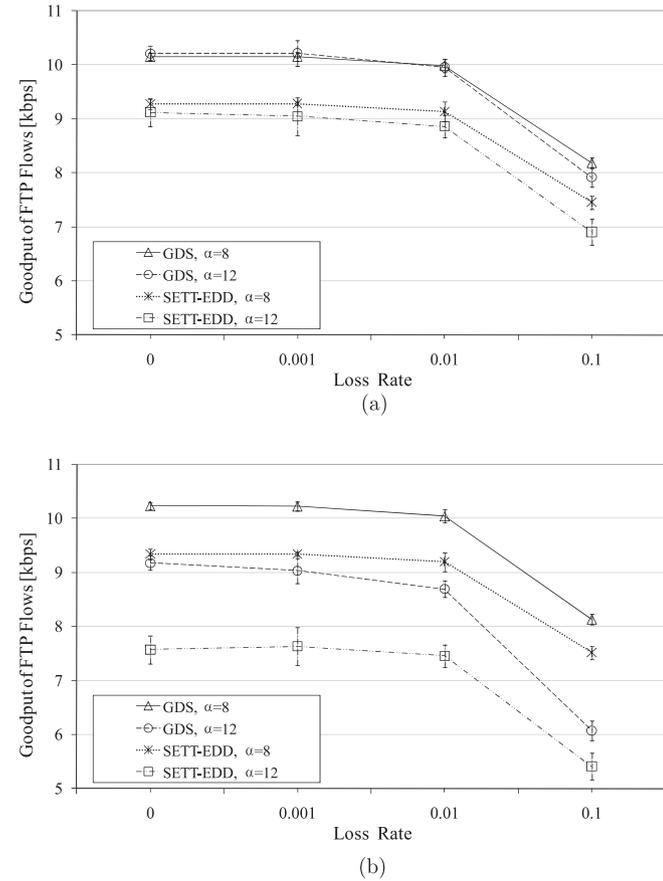


Fig. 8. Goodput of FTP flows with (a) $T_{CA} = 15$ TU, (b) $T_{CA} = 29$ TU, when $T_i^M = 4T_{CA}$.

(see results reported in Table 5) a smaller quota of flows is admitted because at high load ($\alpha = 12$) a smaller T_{CA} does not allow each flow to be polled at least one time for each CAP, as required by the CAC proposed in [1].

Fig. 4 shows that the 95% percentile of packet delays for MPEG4 flows are smaller than target delays for both GDS and SETT-EDD (see Table 4). Furthermore, it is important to note that a smaller T_{CA} reduces packet delay because transmission queues are polled more frequently. Moreover, delays increase with the network load (this is more evident when $T_{CA} = 29$ TU). Another important, but expected results, is that delays increase with the loss rate due to the packet retransmissions. Analogous results have been achieved for H.263 flows (see Fig. 5).

Smaller packet delays are obtained for G.729 flows, as reported in Fig. 6. This effect is due to the adopted EDCA parameter set (the same as in [1]), which treats voice packets with the highest priority. In this case, only results for $\alpha = 12$ have been displayed because very similar to those observed for $\alpha = 8$.

The most important result of the comparison between the two algorithms is about the average HCCA channel utilization in HCCA mode, reported in Fig. 7. It is evident, from these graphs, that GDS requires a smaller amount of bandwidth (up to two orders of magnitude less) with respect to SETT-EDD. This result is a consequence of the cautious usage of the wireless channel granted by GDS. The bandwidth gain is pursued because GDS, knowing the status of transmission queue lengths, can spread over multiple transmission

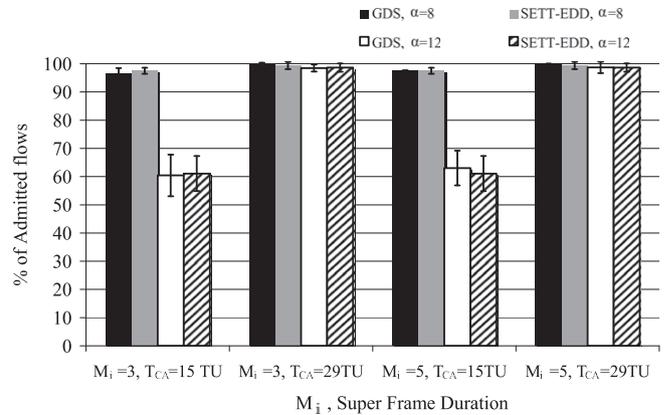


Fig. 10. Percentage of admitted flows in a scenario with realistic channel model and ARF algorithm.

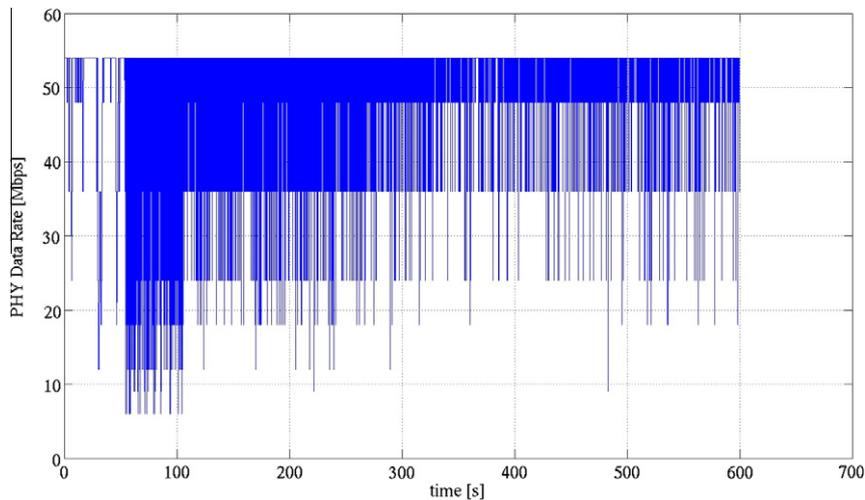


Fig. 9. Realization of the physical transmission rate in a scenario with realistic channel model and ARF algorithm.

opportunities each burst of data and avoid to poll stations with no packet to transmit. Both of scheduling algorithms show an increment of the HCCA channel utilization when the loss rate increases, because of the bandwidth is wasted by retransmissions. Finally, the HCCA channel utilization increases with α and T_{CA} , due to a larger amount of admitted flows.

Fig. 8 reports the goodput achieved by FTP flows when $T_i^M = 4T_{CA}$ (analogous results have been obtained also for $T_i^M = 6T_{CA}$). Note that the goodput of FTP flows is strictly related to the HCCA channel utilization and by the number of admitted real-time flows. In fact, FTP flows can only use the bandwidth left free by higher priority ones. An important results is that GDS allows a higher goodput (from 10% to 20% more) with respect to SETT-EDD, as a consequence of the lower HCCA channel utilization.

5.2. Impact of rate control in realistic channel conditions

In the previous section, we have compared GDS and SETT-EDD assuming that all stations use a fixed transmission rate equal to 54 Mbps. However, an important feature supported by the IEEE 802.11 standard is the multi-rate capability, used to counteract time-varying channel conditions by means of physical transmission rate adaptation. Nowadays, several rate control algorithms have been proposed in literature (see [27] for a survey). Therefore, it is important to evaluate the impact of a dynamic data rate change on the considered scheduling strategies.

As stress test, we chose in our analysis the Auto Rate Fallback (ARF) algorithm [28], considering highly oscillating rate dynamics. Fig. 9 shows a realization of the physical transmission rate obtained for a given station using the ARF algorithm.

To provide a more complete and useful discussion, we have also considered an improved wireless channel model, proposed in [29], which is able to better describe the frame error behavior at the link layer.

Results of this investigation confirm the findings of the previous sub-section (with only some minor differences). In fact, we can observe in Fig. 10 that the percentage of admitted flows has not changed with respect to previous results, because of the use of the same admission control algorithm.

The 95% percentile of packet delays for both video and voice flows is reported in Fig. 11. It shows that the introduction of a rate control algorithm translates into higher packet delays (see Figs. 4–6 for comparison), which are, in any case, kept below the target values in Table 4. The increase of packet delays was expected since now the ARF algorithm reduces the data rate in response to packet losses.

As further confirmation of the GDS effectiveness, Fig. 12 shows that the HCCA usage is kept to very low values with respect to SETT-EDD, thus granting for a higher network throughput (as shown in Fig. 13).

At the end of the comparison, we can conclude that both GDS and SETT-EDD are well designed to guarantee bounded packet delays for real-time flows. However, GDS is able to provide a more efficient bandwidth usage and a higher network throughput. We remark that this advantage is due to two different reason. The first one is the better resource allocation criterion used by GDS to allocate resources among flows. In fact, by spreading the incoming bursts of data over multiple transmission opportunities, a smaller HCCA channel utilization is achieved during the time. The second one is the capability of GDS to exploit the feedback of transmission queue lengths for dynamically adjusting the amount of bandwidth assigned to each station without wasting the network capacity.

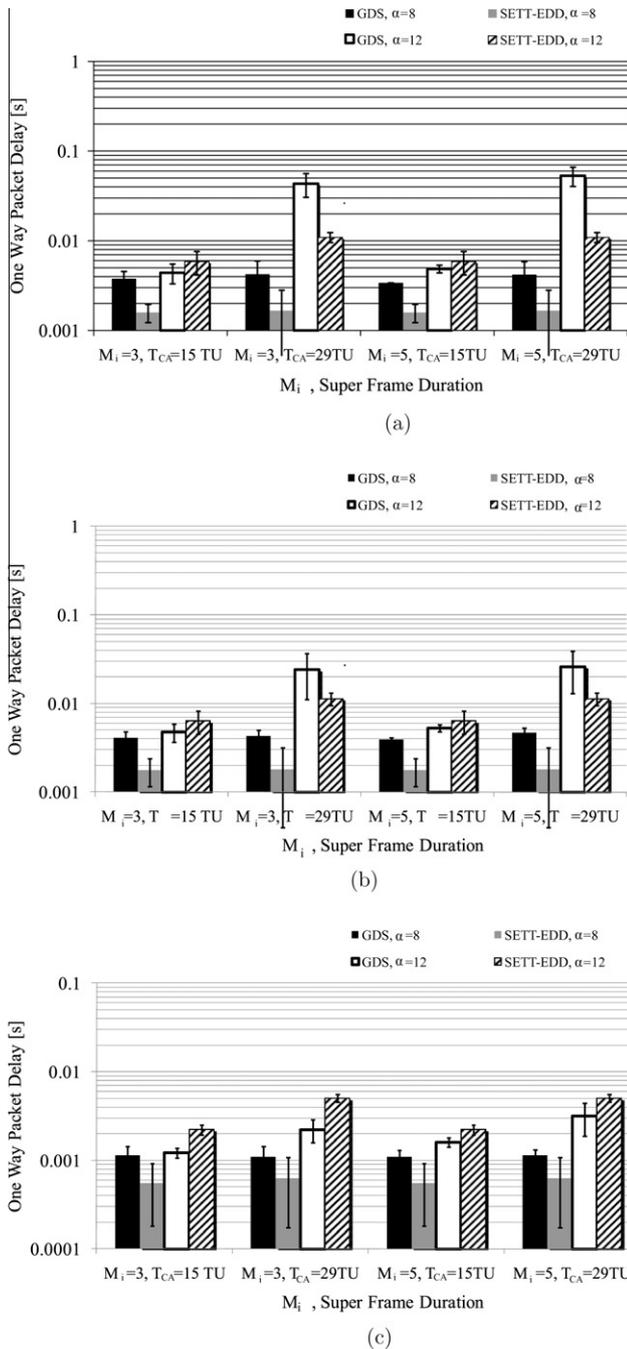


Fig. 11. 95% percentile delays of (a) MPEG4, (b) H263, (c) G729 flows in a scenario with realistic channel model and ARF algorithm.

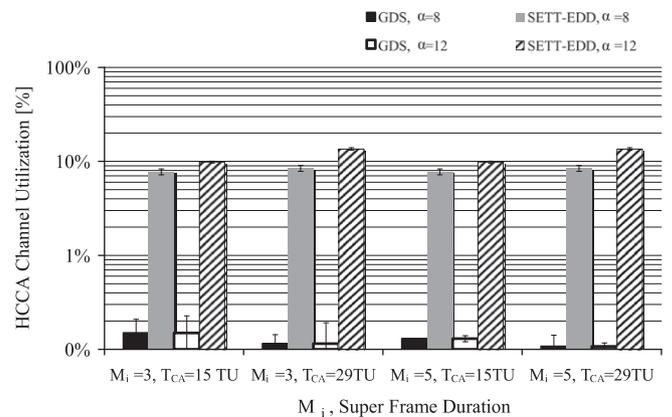


Fig. 12. Average HCCA channel utilization in a scenario with realistic channel model and ARF algorithm.

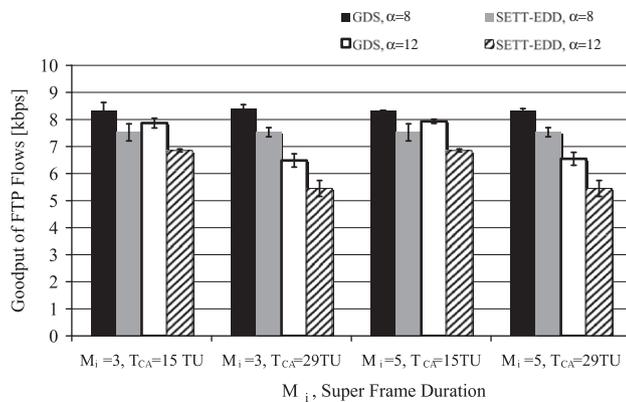


Fig. 13. Goodput of FTP flows in a scenario with realistic channel model and ARF algorithm.

Finally, we would note that the proposed algorithm is very general and that, differently from SETT-EDD, it can be applied also into a distributed environment (as we have demonstrated in [9]).

6. Conclusion

This paper has proposed a comparison between SETT-EDD and GDS, which represent valid implementations of the two main approaches, namely delay-EDD and feedback control based, recently proposed to face real-time service provisioning in IEEE 802.11 WLANs. First, both algorithms have been thoroughly described using theoretical arguments. Then, ns-2 simulations have been proposed to compare them in complex WLAN scenarios with many video, voice, and FTP flows. Results have highlighted that, despite its very simple tuning rules, GDS is able to respect delay bounds as SETT-EDD, but requiring a smaller network capacity and assuring a higher network throughput. Further research will consider the application of GDS to other wireless communication technologies and MAC schemes.

References

- [1] IEEE 802.11, IEEE Standard for Information Tech.-Telecommunications and Information Exchange between Systems, Local and Metropolitan Area Networks, Specific Requirements-Part 11: Wireless LAN MAC and PHY Specifications, IEEE Std. 802.11-2007 (rev. of IEEE Std. 802.11-1999), 2007.
- [2] IEEE 802.11 WG, Draft Amendment to Standard for Information Technology – Telecommunications and Information Exchange between Systems-LAN/MAN Specific Requirements-Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements, IEEE 802.11e/D13.0, Jan. 2005.
- [3] L. Romdhani, Q. Ni, T. Turetletti, Adaptive EDFC: Enhanced service differentiation for IEEE 802.11 wireless Ad-Hoc networks, in IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA, Mar. 2003, pp. 1373–1378.
- [4] G. Boggia, P. Camarda, L.A. Grieco, S. Mascolo, Feedback-based control for providing real-time services with the 802.11e mac, IEEE/ACM Trans. Netw. 15 (2) (2007) 323–333.
- [5] Cisco, Cisco Visual Networking Index: Forecast and Methodology, 2009–2014, white paper, June 2010.
- [6] A. Grilo, M. Macedo, M. Nunes, A scheduling algorithm for QoS support in IEEE 802.11e networks, IEEE Wireless Commun. 10 (3) (2003) 36–43.
- [7] D. Ferrari, D. Verma, A scheme for real-time channel establishment in wide-area networks, IEEE JSAC 8 (3) (1990) 368–379.
- [8] J. Stankovic, M. Spuri, K. Ramamritham, G. Buttazzo, Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms, Springer, 1999.
- [9] G. Boggia, P. Camarda, L.A. Grieco, G. Piro, Extended EDCA for delay guarantees in wireless local area networks, Pervasive and Mobile Comput. 5 (5) (2009) 402–418, doi:10.1016/j.pmcj.2009.07.018.
- [10] Ns-2, Network simulator. Available at: <<http://www.isi.edu/nsnam/ns/>>, 2009.
- [11] Q. Ni, Performance analysis and enhancements for IEEE 802.11e wireless networks, IEEE Netw. 19 (4) (2005) 21–27.
- [12] Q. Ni, L. Romdhani, T. Turetletti, A survey of QoS enhancements for IEEE 802.11 wireless LAN, Wireless Commun. Mobile Comput. 4 (2004) 547–566.
- [13] A. Nafaa, Provisioning of multimedia services in 802.11-based networks: facts and challenges, IEEE Wireless Commun. 14 (5) (2007) 106–112.
- [14] S.C. Lo, G. Lee, W.T. Chen, An efficient multipolling mechanism for IEEE 802.11 wireless LANs, IEEE Trans. Comput. 52 (6) (2003) 764–778.
- [15] P. Ansel, Q. Ni, T. Turetletti, FHCF: a simple and efficient scheduling scheme for IEEE 802.11e, J. Mobile Netw. Appl. (MONET) 11 (3) (2006) 391–403.
- [16] M. van der Schaar, Y. Andreopoulos, H. Zhiping, Optimized scalable video streaming over IEEE 802.11 a/e HCCA wireless networks under delay constraints, IEEE Trans. Mobile Comput. 5 (6) (2006) 755–768.
- [17] Z. Chou, C.C. Hsu, S.N. Hsu, UPCF: a new point coordination function with QoS and power management for multimedia over wireless LANs, IEEE/ACM Trans. Netw. 14 (4) (2006) 807–820.
- [18] T.D. Lagkas, G.I. Papadimitriou, P. Nicopolitidis, A.S. Pomportsis, Priority-oriented adaptive control with QoS guarantee for wireless LANs, IEEE Trans. Veh. Technol. 56 (4) (2007) 1761–1772.
- [19] M.M. Rashid, E. Hossain, V.K. Bhargava, Controlled channel access scheduling for guaranteed QoS in 802.11e-based WLANs, IEEE Trans. Wireless Commun. 7 (4) (2008) 1287–1297.
- [20] Q. Zhao, D.H.K. Tsang, An equal-spacing-based design for QoS guarantee in IEEE 802.11e HCCA wireless networks, IEEE Trans. Mobile Comput. 7 (12) (2008) 1474–1490.
- [21] G. Boggia, P. Camarda, L.A. Grieco, S. Mascolo, A. Stefanelli, Performance evaluation of feedback-based bandwidth allocation algorithms for 802.11e MAC, in: Proc. IEEE VTC Spring, Dublin, Ireland, 2007.
- [22] K.J. Astrom, B. Wittenmark, Computer Controlled Systems: Theory and Design, third ed., Prentice Hall, Englewood Cliffs, 1995.
- [23] Video trace library. Available at: <<http://trace.eas.asu.edu/>>.
- [24] C. Chuah, R.H. Katz, Characterizing packet audio streams from internet multimedia applications, in: Proc. Int. Commun. Conf., ICC, New York, USA, Apr. 2002, pp. 1199–1203.
- [25] W. Wang, S.C. Liew, V.O.K. Li, Solutions to performance problems in VoIP over a 802.11 wireless LAN, IEEE Trans. Veh. Technol. 54 (1) (2005) 366–384.
- [26] C. Barakat, E. Altman, Bandwidth tradeoff between TCP and link-level FEC, Comput. Netw. 39 (2002) 133–150.
- [27] S. Biaz, W. Shaoen, Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison, in: IEEE Symposium on Computers and Communications (ISCC) 2008, Jul. 2009, pp. 130–136.
- [28] M. Lacage, M. Manshaei, T. Turetletti, A practical approach to rate adaptation, IEEE/ACM MSWIM (2004).
- [29] G. Boggia, P. Camarda, A. D’Alconzo, Performance of Markov models for frame-level errors in IEEE 802.11 wireless LANs, Int. J. Commun. Syst. 22 (6) (2009) 695–718.