Feedback-based Control for Providing Real-time Services with the 802.11e MAC

G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo

Abstract

The 802.11e working group has recently proposed the Hybrid Coordination Function (HCF) to provide service differentiation for supporting real-time transmissions over 802.11 Wireless Local Area Networks. The HCF is made of a contention-based channel access, known as Enhanced Distributed Coordination Access, and of a HCF Controlled Channel Access (HCCA), which requires a Hybrid Coordinator for bandwidth allocation to nodes hosting applications with QoS requirements. The 802.11e proposal includes a simple scheduler providing a Constant Bit Rate service, which is not well suited for bursty media flows. This paper proposes two feedback-based bandwidth allocation algorithms to be used within the HCCA, which have been referred to as Feedback Based Dynamic Scheduler (FBDS) and Proportional-Integral (PI)-FBDS. These algorithms have been designed with the objective of providing services with bounded delays. Given that the 802.11e standard allows queue lengths to be fed back, a control theoretic approach has been employed to design the FBDS, which exploits a simple proportional controller, and the PI-FBDS, which implements a proportional-integral controller. Proposed algorithms can be easily implemented since their computational complexities scale linearly with the number of traffic streams. Moreover, a Call Admission Control scheme has been proposed as an extension of the one described in the 802.11e draft. Performance of the proposed algorithms have been theoretically analyzed and computer simulations, using the *ns*-2 simulator, have been carried out to compare their behaviors in realistic scenarios where video, voice, and FTP flows, coexist at various network loads.

Simulation results have shown that, unlike the simple scheduler of the 802.11e draft, both FBDS and PI-FBDS are able to provide services with real-time constraints. However, while the FBDS admits a smaller quota of traffic streams than the simple scheduler, PI-FBDS allows the same quota of traffic that would be admitted using the simple scheduler, but still providing delay bound guarantees.

Authors are with the "Dipartimento di Elettrotecnica ed Elettronica - Politecnico di Bari". Via Orabona, 4 – 70125 BARI, Italy. (e-mail: {g.boggia, camarda, a.grieco, mascolo}@poliba.it).

S. Mascolo is the corresponding author.

Index Terms

QoS, Wireless Networks, real-time applications.

I. Introduction

IEEE 802.11 Wireless Local Area Networks (WLANs) are widely employed for ensuring ubiquitous networking due to their easy installation, flexibility and robustness against failures [1]. At the present, they allow data rates up to 54 Mbps, using Orthogonal Frequency Division Multiplexing techniques [2], [3].

Despite of its very broad diffusion, 802.11 Medium Access Control (MAC) cannot support real time applications, characterized by strict constraints on packet delay and jitter [4], [5], [6]. Recently, to overcome this limitation, the 802.11e working group has proposed: (1) the Hybrid Coordination Function (HCF) as an enhanced access method; (2) a Call Admission Control (CAC) algorithm; (3) specific signaling messages for service request and Quality of Service (QoS) level negotiation; (4) four Access Categories (ACs) with different priorities to map the behavior of traffic flows with users' QoS requirements [7].

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, called Hybrid Coordinator (HC), generally located at the access point. EDCA operates as the basic DCF access method [1], but using different contention parameters per AC (see Tab. I). In this way, a service differentiation among traffic streams is statistically pursued [8]. Tuning EDCA parameters to provide prioritization of ACs is a current research topic [9]. A method for setting EDCA parameters has been described in [10]. Regarding the goal of providing delay guarantees, several papers have pointed out that the EDCA can provide a real-time service to highest priority flows, at the price of starving flows with lower priority, especially at high network load [11], [12], [13]. Moreover, EDCA can provide only a relative differentiation among service classes, but not absolute guarantees on throughput/delay performance [8], [14].

To overcome those limitations, adaptive algorithms that dynamically tune EDCA parameters have been recently proposed in [15], [16]; however, the effectiveness of these schemes have been proved only using simulations and no theoretical bounds on their performance in a general scenario have been derived.

TABLE I

EDCA CONTENTION PARAMETERS

Designation	AC	CW_{min}	CW_{max}	AIFS
Background	AC_BK	CW_{min}	CW_{max}	7
Best Effort	AC_BE	CW_{min}	CW_{max}	3
Video	AC_VI	$(CW_{min}+1)/2-1$	CW_{min}	2
Voice	AC_VO	$(CW_{min}+1)/4-1$	$(CW_{min}+1)/2-1$	2

With the HCCA, the HC is responsible for assigning the right to transmit at nodes hosting applications with QoS requirements, i.e., to perform dynamic bandwidth allocation within the WLAN. However, the 802.11e draft does not specify an effective bandwidth allocation algorithm; it only suggests a simple scheduler that uses static values declared by data sources for providing a Constant Bit Rate (CBR) service. As a consequence, this scheduler is not well suited for bursty media flows [17]. An improved bandwidth allocation algorithm has been proposed in [18], which schedules transmission by taking into account both the average and the maximum declared source rates. An adaptive version of the simple scheduler, which is based on the Delay-Earliest Due-Date algorithm, has been proposed in [17]. However, this scheduler does not exploit any feedback information from mobile stations, but implements a trial and error procedure for discovering the optimal amount of resources to assign to each AC. The Fair Scheduling scheme proposed in [19] allocates the WLAN channel bandwidth to wireless nodes in order to fully deplete transmit queues, which are estimated by taking into account the delayed feedbacks from the wireless nodes.

This paper proposes two feedback-based bandwidth allocation algorithms exploiting HCCA to provide service with guaranteed bounded delays: (1) the Feedback Based Dynamic Scheduler (FBDS) and (2) the Proportional Integral (PI)-FBDS. They have been designed using classic discrete-time feedback control theory. Proposed algorithms have low computational costs and can be easily implemented in wireless network interface cards. Theoretical results have been exploited to properly select the parameters of the FBDS and PI-FBDS bandwidth allocation algorithms in order to provide both system stability and real-time service. In particular, they have shown that using PI-FBDS there are more degrees of freedom in the choice of the parameter sets of the bandwidth allocation algorithm with respect to the case of FBDS, so that the parameter set can be properly selected in order to optimize system performace. Simulation results, obtained using the *ns*-2 simulator [20], have shown that, unlike the simple scheduler proposed by the 802.11e working group, both FBDS and PI-FBDS provide a real-time service with QoS guarantees in terms of delay, regardless of the network load. Moreover, when the PI-FBDS is used, the best trade-off between the one-way packet delay and the network utilization is achieved.

The rest of the paper is organized as follows: Section II gives an overview of the HCCA method; in Section III FBDS and PI-FBDS algorithms are proposed; Section IV shows simulation results; finally, the last Section draws the conclusions.

II. Overview of the HCCA Method

The core of the 802.11e proposal is the HCF, which is in charge of assigning TXOPs (Transmission Opportunities) to each AC in order to satisfy its QoS needs. TXOP is defined as the time interval during which a station has the right to transmit and is characterized by a starting time and a maximum duration. Contiguous time during which TXOPs are granted to the same station with QoS capabilities (i.e., a QoS station, QSTA) are called Service Period (SP). The interval T_{SI} between two successive SPs is called Service Interval [4], [7].

HCCA method combines some EDCA characteristics with some features of the Point Coordination Function (PCF) scheme, which is an optional contention-free access method defined by the basic 802.11 standard [1], [4]. The time is divided into repeated periods, called *SuperFrames* (SFs). Each superframe starts with a beacon frame after which, for legacy purpose, there could be a contention free period (CFP) for PCF access. The remaining part of the superframe forms the Contention Period (CP), during which QSTAs contend for the access to the radio channel using the EDCA mechanism (see Fig. 1). At least one CP interval, long enough to transmit a data frame with the maximum size at the minimum rate, must be contained in each superframe; the CP interval can be also used for management tasks, such as associations of new stations, new traffic stream negotiations, and so on.

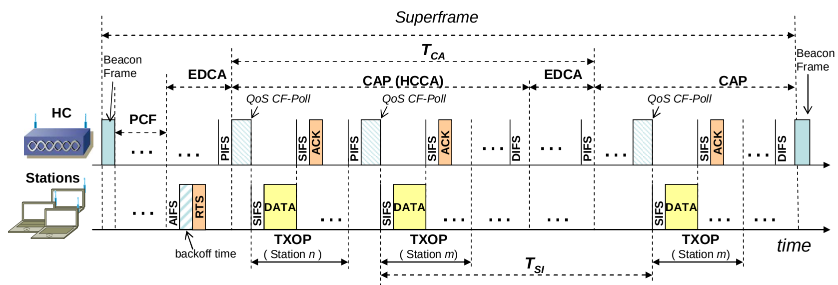

Fig. 1. Scheme of a superframe using the HCF controlled access method.

During the CP, the HC can start a Contention Access Phase (CAP)[1], during which only QSTAs that are polled and granted with the *QoS CF-Poll frame* are allowed to transmit during the assigned TXOPs. Thus, the HC implements a prioritized medium access control.

The number of CAPs and their locations in each superframe are chosen by the HC in order to satisfy QoS needs of each station. CAP length cannot exceed the value of the system variable *dot11CAPLimit*, which is advertised by the HC in the beacon frame when superframes start [7].

The simple scheduler designed in the draft [7] states that the $TXOP_i$ assigned to the i^{th} traffic queue in the WLAN should be computed as follows:

$$TXOP_i = max\left\{\frac{N_i \cdot L_i}{C_i} + H, \frac{M}{C_i} + H\right\} \quad (1)$$

[1]HCCA can be also enabled during the CFP with a procedure similar to the one described in this Section.

where, i refers to the i^{th} queue, L_i is the nominal size of MAC Service Data Units (MSDUs), C_i is the rate at which the data are transmitted over the wireless channel, H is the protocol overhead, M is the maximum MSDU size; and $N_i = \left\lceil \frac{T_{SI} \cdot \rho_i}{L_i} \right\rceil$, where ρ_i is the Mean Data Rate associated with the queue and T_{SI} is the Service Interval.

According to IEEE 802.11e specifications, each QSTA can feed back queue length of each AC to the HC in the frames' headers. As will be shown in this paper, this information can be fruitfully exploited to design novel dynamic bandwidth allocation algorithms based on HCCA, using feedback control theory [21].

A. QoS signalling

In the 802.11e proposal [7] each Traffic Stream , i.e., a data flow with QoS needs, is described by a *Traffic SPECification* (TSPEC), which indicates the main characteristics of the stream (e.g., Nominal/Maximum size of MAC frames, Maximum burst size, Delay Bound, and so on) [7]. The TSPEC is similar to the one introduced in [22] for *IP FlowSPecs* definition and adopted in IntServ [23] and DiffServ [24] architectures. Specific signalling has been introduced to manage new traffic stream requests and QoS provisioning. In particular, to start a new stream, the QSTA issues a setup phase by generating a message, known as *Mac Layer Management Entity (MLME)-ADDTS request*, containing the TSPEC of the stream [7]. This request message is sent to the HC which decides whether or not to admit the stream with the specified TSPEC, or to suggest an alternative TSPEC. The decision of the HC is transmitted with the *ADDTS response* message. After the reception of this response, the QSTA sends a message specifying whether the HC response meets its needs or not; if not, the whole process can be repeated [7].

B. Call Admission Control

In IEEE 802.11e networks the HC is used as an admission control unit. There are two distinct admission control schemes: one for the contention-based access and the other for the controlled-access. Herein, we will focus on the latter mechanism. Details regarding the contention-based admission control can be found in [7].

Let m be the number of admitted flows. In the presence of a new admission request for a traffic stream, the admission control unit calculates the TXOP duration needed by the stream ($TXOP_{m+1}$) as imposed by the simple scheduler (see Eq. (1)). The stream is admitted if the following inequality is satisfied:

$$\frac{TXOP_{m+1}}{T_{SI}} + \sum_{i=1}^{m} \frac{TXOP_i}{T_{SI}} \leq \frac{T - T_{CP}}{T} \tag{2}$$

where T is the superframe duration, and T_{CP} is the time during which EDCA is used for frame transmissions in the superframe. It is worth noting that the terms $TXOP_i$ used in the CAC test (2), are computed using the Eq. (1). This implies that the CAC does not take into account the actual network load but only an estimation based on static values declared by data sources in their TSPECs when deciding wheter to admit or not a new stream.

III. FBDS AND PI-FBDS ALGORITHMS

In this section, FBDS and PI-FBDS algorithms will be designed using feedback control theory. We will assume that both algorithms, running at the HC, allocate the WLAN channel bandwidth to wireless stations hosting real-time applications, using HCCA functionalities. This allows the HC to assign TXOPs to ACs by taking into account their specific time constraints and transmission queue levels [13]. We will refer to a WLAN system made of an Access Point and a set of quality of service enabled mobile stations (QSTAs). Each QSTA has up to 4 queues, one for each AC in the 802.11e proposal. Let T_{CA} be the time interval between the starting of two successive CAPs (see Fig. 1). Every time interval T_{CA}, which is assumed to be constant, the HC must allocate the bandwidth that will drain each queue during the next CAP. We assume that at the beginning of each CAP, the HC is aware of all the queue levels $q_i,\ i = 1, \dots, M$ at the beginning of the previous CAP, where M is the total number of traffic queues in the WLAN.[2].

The following discrete time linear model describes the dynamics of the i^{th} queue:

$$q_i(n+1) = q_i(n) + d_i(n) \cdot T_{CA} + u_i(n) \cdot T_{CA}, \qquad i = 1, \dots, M, \tag{3}$$

where, $q_i(n) \geq 0$ is the queue level at the beginning of the n^{th} CAP; $u_i(n) \leq 0$ is the average depletion rate (i.e., its absolute value represents the bandwidth assigned to drain the queue);

[2]This is a worst case assumption. In fact, queue levels are fed back using frame headers as described in Sec. II; as a consequence, if the i^{th} queue length has been fed at the beginning of the previous CAP, then the feedback signal might be delayed up to T_{CA} seconds.

$d_i(n) = d_i^s(n) - d_i^{CP}(n)$ is the difference between $d_i^s(n) \geq 0$, which is the average input rate at the queue during the n^{th} T_{CA} interval, and $d_i^{CP}(n) \geq 0$, which is the average output rate at the queue during the n^{th} T_{CA} interval using the EDCA.

The input $d_i(n)$ is unpredictable since it depends on the behavior of the source that feeds the i^{th} queue and on the number of packets transmitted using EDCA. Thus, from a control theoretic perspective, $d_i(n)$ can be modelled as a disturbance [25]. Without loss of generality, the following piece-wise constant model for the disturbance $d_i(n)$ can be assumed:

$$d_i(n) = \sum_{j=0}^{+\infty} d_{0j} \cdot 1(n - t_j) \tag{4}$$

where $1(n)$ is the unitary step function, $d_{0j} \in \mathbb{R}$, and t_j is a time lag [25].

Due to the assumption (4), the linearity of the system described by eq. (3), and the superposition principle that holds for linear systems, we will design the feedback control law by considering only a step disturbance: $d_i(n) = d_0 \cdot 1(n)$ [25].

A. The closed loop control scheme

Our goal is to design a control law that drives the queuing delay τ_i, experienced by each frame going through the i^{th} queue, to a desired target value τ_i^T that represents the QoS requirement of the AC associated to the queue.

We will consider the closed loop control system shown in Fig. 2, where the set point q_i^T has been set equal to zero, which means that we would ideally target empty queues. Regarding the transfer function $G_i(z)$ of the controller, we will focus on two very simple controllers: a proportional (P) controller $G_i(z) = k_{p_i}$, and a proportional-integral (PI) controller $G_i(z) = k_{p_i}\left(1 + \frac{z}{z-1} \cdot \frac{1}{T_{I_i}}\right)$. The corresponding bandwidth allocation algorithms will be referred to as Feedback Based Dynamic Scheduler (FBDS), and PI-FBDS.

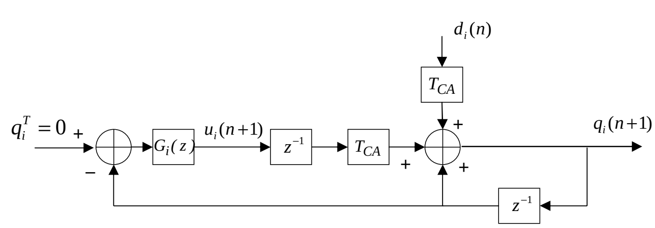

Fig. 2. Closed-loop control scheme based on HCCA.

1) The case of a proportional controller:

Proposition 1: The system reported in Fig. 2, where $G_i(z) = k_{p_i}$, is asymptomatically stable if and only if the following inequality holds:

$$0 < k_{pi} < 1/T_{CA}. \tag{5}$$

Proof: By considering the control scheme in Fig. 2 with $G_i(z) = k_{p_i}$, it is straightforward to compute the $\mathcal{Z}$-transforms of $q_i(n)$ and $u_i(n)$:

$$Q_i(z) = \frac{z \cdot T_{CA}}{z^2 - z + k_{pi} \cdot T_{CA}} \cdot D_i(z); \qquad U_i(z) = -\frac{k_{pi} \cdot T_{CA}}{z^2 - z + k_{pi} \cdot T_{CA}} \cdot D_i(z) \tag{6}$$

where $D_i(z) = \mathcal{Z}[d_i(n)]$.

From eqs. (6) it results that the system poles are $z_p = \frac{1 \pm \sqrt{1 - 4k_{pi} \cdot T_{CA}}}{2}$; thus, the system is asymptotically stable if and only if $|z_p| < 1$, that is: $0 < k_{pi} < 1/T_{CA} \quad \square$.

Proposition 2: By considering the system reported in Fig. 2, where $d_i(n) = d_0 \cdot 1(n)$ and $G_i(z) = k_{p_i}$, the following inequality has to be satisfied in order to achieve a steady-state delay smaller than the target delay τ_i^T:

$$k_{pi} \geq 1/\tau_i^T. \tag{7}$$

Proof: By considering that the $\mathcal{Z}$-transform of the step function $d_i(n) = d_0 \cdot 1(n)$ is $D_i(z) = d_0 \cdot \frac{z}{z-1}$, if we apply the final value theorem [21] to Eqs. (6), it results:

$$u_i(\infty) = \lim_{n \to +\infty} u_i(n) = \lim_{z \to 1} (z-1) U_i(z) = -d_0; \qquad q_i(\infty) = d_0 / k_{pi},$$

which implies that the steady state queueing delay is:

$$\tau_i(\infty) = |q_i(\infty)/u_i(\infty)| = 1/k_{pi}$$

The proof can now be easily derived by imposing $\tau_i(\infty) \leq \tau_i^T \quad \square$.

Remark 1: It is worth noting that $q(\infty) > 0$ even if $q_i^T = 0$, which means that the proportional controller is not able to fully reject the step disturbance $d_0 \cdot 1(n)$.

Remark 2: From inequalities (5) and (7), the T_{CA} parameter must satisfy the constraint:

$$T_{CA} < \min_{i=1..M} \tau_i^T. \tag{8}$$

Remark 3: From propositions 1 and 2 it turns out that the gain k_{pi} can vary in the range $\left[\frac{1}{\tau_i^T}, \frac{1}{T_{CA}}\right[$. We will set k_{pi} at its lowest admissible value $1/\tau_i^T$, allocating the lowest bandwidth that guarantees the target delay. In this way a cautious usage of the WLAN channel is achieved.

2) The case of a Proportional-Integral Controller:

Proposition 3: The system reported in Fig. 2, where $G_i(z) = k_{p_i}\left(1 + \frac{z}{z-1} \cdot \frac{1}{T_{I_i}}\right)$, is asymptomatically stable if and only if the following inequalities hold:

$$k_{p_i} < 1/T_{CA}; \qquad T_{I_i} > 1/(1 - T_{CA}k_{p_i}) \ . \tag{9}$$

Proof: By considering the control scheme in Fig. 2 with $G_i(z) = k_{p_i}\left(1 + \frac{z}{z-1} \cdot \frac{1}{T_{I_i}}\right)$, it is straightforward to compute the $\mathcal{Z}$-transforms of $q_i(n)$ and $u_i(n)$:

$$Q_i(z) = \frac{T_{CA}T_{I_i}z(z-1)}{T_{I_i}z^3 - 2T_{I_i}z^2 + (T_{I_i} + T_{CA}k_{p_i}T_{I_i} + T_{CA}k_{p_i})z - T_{CA}k_{p_i}T_{I_i}}D_i(z) \tag{10}$$

$$U_i(z) = \frac{T_{CA}k_{p_i}T_{I_i} - T_{CA}k_{p_i}T_{I_i}z - T_{CA}k_{p_i}z}{T_{I_i}z^3 - 2T_{I_i}z^2 + (T_{I_i} + T_{CA}k_{p_i}T_{I_i} + T_{CA}k_{p_i})z - T_{CA}k_{p_i}T_{I_i}}D_i(z) \tag{11}$$

where $D_i(z) = \mathcal{Z}[d_i(n)]$ and $Q_i^T(z) = \mathcal{Z}[q_i^T]$.

The proof follows by applying the Jury criterion [21] to $\mathcal{Z}$-transforms (10) and (11) $\square$.

Proposition 4: By considering the system reported in Fig. 2, where $d_i(n) = d_0 \cdot 1(n)$ and $G_i(z) = k_{p_i}\left(1 + \frac{z}{z-1} \cdot \frac{1}{T_{I_i}}\right)$, the steady-state delay is zero.

Proof: By applying the final value theorem to Eq. (10), where $D_i(z) = d_0 \cdot \frac{z}{z-1}$, it turns out:

$$q_i(\infty) = \lim_{n\to+\infty} q_i(n) = \lim_{z\to 1}(z-1)Q_i(z) = 0,$$

which implies that the steady state queuing delay is zero. $\square$.

Remark 4: A steady state queuing delay equal to zero is due to the integral action of the controller, which is able to fully reject the step disturbance at steady state.

Remark 5: It is worth noting that parameters of the PI regulator are subject only to the stability constraints (9). As a consequence, there are more degrees of freedom when chosing k_{p_i} and T_{I_i} with respect to the case of the proportional controller [3].

[3]We will show in the last section the impact of k_{p_i} and T_{I_i} in a realistic scenario involving video, voice, and FTP flows.

Remark 6: When the PI controller is used, it might happen that the depletion rate $|u_i(n+1)|$ computed by the controlled is larger than $q_i(n)/T_{CA}$, which is the amount of bandwidth required to fully deplete the i^{th} queue during the n^{th} CAP. This assignement would obviously waste WLAN resources. To overcome this drawback, we will employ the following shortcut:

$$u_i(n+1) \leftarrow max\,\{u_i(n+1), -q_i(n)/T_{CA}\} \tag{12}$$

where, the term $-q_i(n)/T_{CA}$ is the depletion rate needed to fully empty the queue.

B. TXOP assignment

We have seen in Sec. II that, every time interval T_{CA}, the HC allocates TXOPs to mobile stations in order to meet the QoS constraints. Herein, we shows how to transform the bandwidth $|u_i|$ into a $TXOP_i$ assignment. In particular, if the i^{th} queue is drained at rate C_i, the following relation holds:

$$TXOP_i(n) = \frac{|u_i(n) \cdot T_{CA}|}{C_i} + H(n) \tag{13}$$

where $TXOP_i(n)$ is the TXOP assigned to the i^{th} queue during the n^{th} CAP, and $H(n)$ is the time overhead due to ACK packets and interframe spaces (see Fig. 1).

The extra quota of TXOP due to the overhead $H(n)$ depends on the number of MSDUs corresponding to the amount of data $|u_i(n) \cdot T_{CA}|$ to be transmitted. $H(n)$ could be estimated by assuming that all MSDUs have the same nominal size specified into the TSPEC. Moreover, when $|u_i(k) \cdot T_{CA}|$ does not correspond to a multiple of MSDUs, the TXOP assignment is rounded in excess in order to guarantee a queuing delay equal or smaller than the target value τ_i^T.

C. Call Admission Control and Channel saturation

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 maximum CAP duration, which is defined by the system variable *dot11CAPLimit*; this value can be violated when the network is saturated. In order to avoid heavy channel saturations, we propose a CAC scheme which is an improved version of the one proposed by the 802.11e working group. However, since transient network overloads cannot be avoided due to the burstiness of the multimedia flows, when for a given n_0 we have that: $\sum_{i=1}^{M} TXOP_i(n_0) > dot11CAPLimit$, each computed $TXOP_i(n_0)$ is decreased by an amount $\Delta TXOP_i(n_0)$, so that the following capacity constraints is satisfied:

$$\sum_{i=1}^{M} [TXOP_i(n_0) - \Delta TXOP_i(n_0)] = dot11CAPLimit \ . \tag{14}$$

In particular, the generic amount $\Delta TXOP_i(n_0)$ is evaluated as a fraction of the total amount $\Delta = \sum_{j=1}^{M} TXOP_i(n_0) - dot11CAPLimit$, as follows:

$$\Delta TXOP_i(n) = \frac{TXOP_i(n)C_i}{\sum_{j=1}^{M} [TXOP_j(n)C_j]} \Delta. \tag{15}$$

Notice that Eq. (15) provides a $\Delta TXOP_i(n_0)$, which is proportional to $TXOP_i(n_0)C_i$; in this way, connections transmitting at low rates are not too much penalized.

When the number of multimedia flows sharing the WLAN increases, the channel saturates and delay bounds cannot be guaranteed [16]. In order to avoid heavy channel saturations, we propose a new CAC scheme, which exploits the 802.11e CAC proposal (see Sec. II).

In particular, starting from the TXOPs allocated to the active traffic streams in each CAP, a new flow request is admitted if

$$\frac{TXOP_{m+1}}{T_{CA}} + \sum_{i=1}^{m} \frac{TXOP_i}{T_{CA}} \leq \frac{T - T_{CP}}{T} \tag{16}$$

where m is the number of admitted flows, T is the superframe duration, and T_{CP} is the time used by EDCA during the superframe.

Notice that the proposed CAC scheme given by eq. (16) has been obtained from the CAC suggested in the 802.11e draft [7] (see sec. II-B), by replacing the constant TXOPs used by the simple scheduler with the time-varying ones allocated by the proposed bandwidth allocation algorithms. For the same reason, the T_{SI} term in (2) has been replaced by the T_{CA} term in (16). In this way, our proposed CAC takes into account the bandwidth actually used by the flows and not the sum of the average source rates declared in the TSPECs. In other terms, the proposed CAC is a measurement-based CAC [26], provideing a better protection against network overloads with respect to the CAC test (2) proposed in [7].

D. Computational Complexity of the bandwith allocation algorithms

Herein, we estimate the computational complexity of the proposed allocation algorithms.

Channel saturation episodes will be neglected because we assume they are sporadic due to the effectiveness of the admission control scheme.

Proposition 5: In a WLAN system with M active traffic streams, the computational complexity of the FBDS algorithm is $O(2M)$.

Proof: Every time interval T_{CA}, the HC computes the bandwidth assignment for each one of the M active traffic streams, using Eq. (13). With FBDS, from Fig. 2, the control law is:

$$u_i(n+1) = -k_{pi} \cdot q(n). \tag{17}$$

Thus, Eq. (13) becomes:

$$TXOP_i(n) = \beta_i q(n) + H(n) \tag{18}$$

where $\beta_i = k_{pi} T_{CA}/C_i$.

As a consequence a single bandwidth assignment consists of two multiplications and one sum. The first multiplication takes into account the term $\beta_i q(n)$, the second one estimates the protocol overhead, which is proportional to $\beta_i q(n)$. Thus, we need $2M$ multiplication plus M sums for each T_{CA} interval, i.e., the computational complexity is $O(2M)$ $\square$.

Proposition 6: In a WLAN system with M active traffic streams, the computational complexity of the PI-FBDS algorithm is $O(4M)$.

Proof: For each active stream, the HC computes the bandwidth by using Eq. (13). When PI-FBDS is used, the control law is:

$$u_i(n+1) = -k_{pi} \cdot q_i(n) - \frac{k_{pi}}{T_{Ii}} \sum_{h=0}^{n} q_i(h) \tag{19}$$

which can be also written as

$$u_i(n+1) = u_i(n) + k_{pi} q_i(n-1) - k_{pi} \left(1 + 1/T_{Ii}\right) q_i(n). \tag{20}$$

Thus, Eq. (13) becomes:

$$TXOP_i(n) = (T_{CA}/C_i) \left[u_i(n) + k_{pi} q_i(n-1) - k_{pi} \left(1 + 1/T_{Ii}\right) q_i(n)\right] + H(n) \tag{21}$$

Now, considering that the overhead is estimated using 1 multiplication, a single bandwidth assignment consists of 4 multiplications and 3 sums. As a consequence, we need $4M$ multiplication plus a $3M$ sums for each T_{CA} interval. Thus, the computational complexity is $O(4M)$ $\square$.

From the above propositions, we can conclude that the computational complexities of both FBDS and PI-FBDS scale linearly with the number of active streams. Thus, such schemes can be easily implemented in real wireless network interface cards.

IV. Performance Evaluation

In order to assert the validity of FBDS and PI-FBDS in realistic scenarios, computer simulations involving voice, video and FTP data transfers have been run using *ns*-2 [20]. We have considered a scenario where a 802.11a wireless channel at 54 Mbps is shared by a mix of 3α voice flows encoded with the G.729 standard [27], α MPEG-4 encoded video flows [28], α H.263 video flows [29], and α FTP best effort flows. Therefore, in such a scenario the traffic load is proportional to the parameter α.

For video flows, we have used traffic traces available from the video trace library [30]. G.729 sources are modeled using Markov ON/OFF sources [31]. The ON period is exponentially distributed with mean 3 s; the OFF period has a truncated exponential pdf with an average value of 2.23 s and an upper limit of 6.9 s. During the ON period, the voice source sends packets of 20 bytes every 20 ms (i.e., the source data rate is 8 kbps; also we are considering two G.729 frames combined into one packet [32]). By taking into account the overheads of the RTP/UDP/IP protocol stack the total rate over the wireless channel is 24 kbps, during the ON periods. During the OFF period the rate is set to zero since we assume the presence of a Voice Activity Detector (VAD).

During CAPs, stations access the channel using the HCCA method, otherwise they use EDCA. In simulations, EDCA parameters have been set as suggested in [7].

The target delay τ_i^T has been set equal to 30 ms for voice flows and 40 ms for video flows. According to 802.11 standard, in our *ns*-2 implementation T_{CA} is expressed in Time Units (TUs), each one equal to 1024 μs [1]; we assume a T_{CA} of 29 TUs in order to satisfy inequality (8). The value of system variable *dot11CAPlimit* has been set in order to allow the transmission of at least 10 MSDUs of maximum size using EDCA, between the starting of two successive CAPs.

When FBDS is used, the proportional gain k_{p_i} is set equal to $1/\tau_i^T$ (see Sec. II). When PI-FBDS is used, we have simulated the scenario using many parameter sets (results will be reported in Sec. IV-B) and we have selected the set $(k_{p_i} = 15s^{-1}, T_{I_i} = 4)$, which provide a good trade-off between the ratio of admitted flows and the one-way packet delay.

The main characteristics of the considered multimedia flows are summarized in Table II.

TABLE II

MAIN FEATURES OF THE CONSIDERED MULTIMEDIA FLOWS.

Type of flow	*Nominal (Maximum) MSDU Size* [Byte]	*Mean (Maximum) Data Rate* [kbps]	*Inactivity Interval* [s]
MPEG-4	1536 (2304)	770 (3300)	3
H.263 VBR	1536 (2304)	450 (3400)	3
G.729 VAD	60 (60)	13.76 (24)	10

Before starting data transmission, a multimedia source has to set up a new Traffic Stream as specified in Sec. II-A. If the reply to the stream admission message is not received within a Δ_{TO} timeout interval, the request is repeated up to a maximum number N_{Adm} of times; in our simulations, we have chosen $N_{Adm} = 10$ and $\Delta_{TO} = 1.5$ 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 Δ_{defer} with average value equal to 1 min. The duration of video flows is deterministic and equal to 10 min, whereas voice flows durations are exponentially distributed with average value of 120 s. When a multimedia flow terminates, a new stream of the same type is generated after an exponentially distributed random time, with an average value equal to 1 min. Each terminated flow is withdrawn from the polling list by the HC after that no more packets from that flow are received for a time equal to the Inactivity Interval reported in Table II. Each simulation lasts 1 hour.

A. Performance comparison of PI-FBDS, FBDS, and the Simple scheduler

Tab. III reports the ratio of admitted flows for various values of the network load parameter α when PI-FBDS or FBDS or the simple scheduler are used.

By looking at this table, it is straightforward to note that FBDS admits the smallest ratio of flows. On the other hand, almost all the flows are admitted when either the PI-FBDS or the Simple scheduler are used. The reason is that the Simple scheduler allocates TXOPs by taking into account the declared average source rates, which can be much smaller than the actual source rates; as a consequence, the CAC test realized with eq. (2) is always satisfied. When FBDS or PI-FBDS are used, the CAC test given by eq. (16) takes into account the actual load. The substantial difference between FBDS and PI-FBDS is that the PI regulator gives the opportunity to choice its parameter set in a broader space of values with respect to the case of a proportional controller. In particular, we can select the parameter set of the PI regulator to filter out high frequency components of the traffic load better than in the case of the P regulator. In other words, the PI-FBDS algorithm allows CAC test of eq. (16) to consider only low frequency components of network load, with the consequence that a larger number of flows are admitted.

TABLE III

RATIOS OF ADMITTED FLOWS.

α	PI-FBDS ($k_{p_i} = 15s^{-1}$, $T_{I_i} = 4$)	FBDS	Simple scheduler
5	100%	100%	100%
10	100%	89%	100%
12	99%	59%	100%

At this point, due to the smaller number of admitted traffic streams, we would expect lower one-way packet delays when FBDS is employed with respect to one-way packet delays obtained with the Simple scheduler or PI-FBDS. This is shown in Fig. 3 where the one-way packet delays experienced by the MPEG, H.263, and G.729 flows are reported. By looking at those figures, we can note that FBDS provides the smallest delays due the smallest quota of admitted flows. The surprising result is that, with high load (i.e., $\alpha = 12$) although the Simple scheduler and PI-FBDS admit the same ratio of flows, delays obtained with PI-FBDS are one order of magnitude smaller than those obtained with the Simple scheduler. The latter result highlights that if network resources are carefully managed, a good trade-off between the number of admitted flows and the one-way packet delay can be achieved, also with high load.

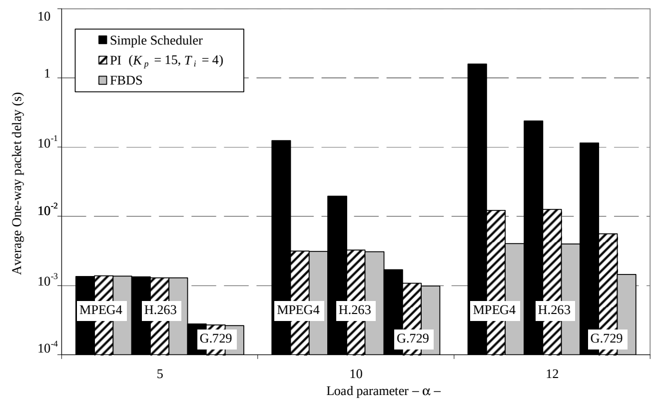

Fig. 3. Average one-way packet delay of real-time flows.

To look at those results from another point of view, Figs. 4, 5, and 6 show the CDF of the one-way packet delays obtained for MPEG4, H.263, and G.729 flows, respectively. They clearly highlights that as the load parameter α increases differences among the consider bandwidth allocation algorithms become more evident. In particular, while for $\alpha = 5$ and $\alpha = 10$ delays obtained with PI-FPDS, FBDS, and the Simple scheduler are similar to each other, when $\alpha = 12$ the Simple scheduler is no more able to provide a service with bounded delay, while PI-FBDS and FBDS guarantees real-time packet delivery. Obviously, FBDS provides delays smaller with respect to PI-FBDS due to the smaller quota of admitted flows.

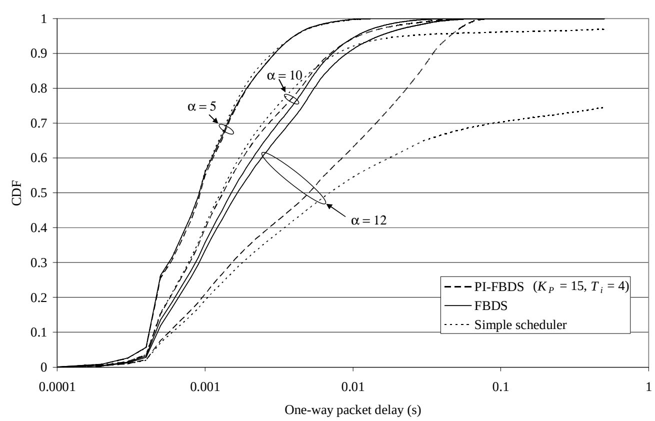

Fig. 4. CDFs of the one-way packet delays for MPEG4 flows when $\alpha = 5, 10, 12$.

Table IV reports the average and peak superframe utilization in HCCA mode, which is defined as the sum of TXOPs allocated during CAPs over the superframe duration. It shows that the

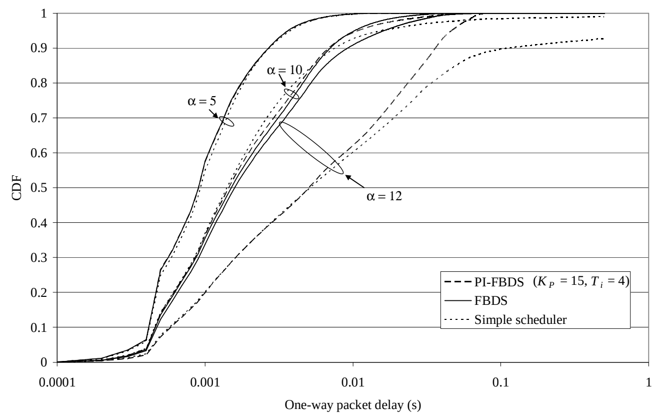

Fig. 5. CDFs of the one-way packet delays for H.263 flows when $\alpha = 5, 10, 12$.

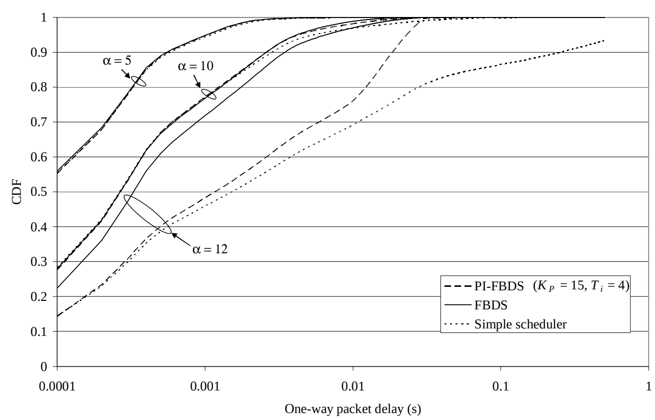

Fig. 6. CDFs of the one-way packet delays for G.729 flows when $\alpha = 5, 10, 12$.

Simple scheduler requires the highest average quota of WLAN resources. The reason is that the simple scheduler does not adapt the quota of allocated resources to the actual load because it provides a CBR service. For the same reason, the peak superframe utilizations achieved by the Simple scheduler for $\alpha = 10$ and $\alpha = 12$, i.e., at high traffic load, are smaller than those provided by FBDS and PI-FBDS. This result clearly highlights that the proposed control schemes enable a more proper usage of the bandwidth and allows the bandwidth requirements of the real-time flows to be tracked.

Finally, Tab. IV reports also the goodput achieved by the FTP flows when the PI-FBDS or the FBDS or the Simple scheduler are used. It is straightforward to note that the goodput of the FTP flows is strictly related to the average superframe utilization In fact, when FBDS is

TABLE IV

RESULTS IN HCCA MODE.

α	PI-FBDS ($k_{p_i} = 15s^{-1}$, $T_{I_i} = 4$)			FBDS			Simple scheduler		
	Superframe Utilization		Overall Goodput	Superframe Utilization		Overall Goodput	Superframe Utilization		Overall Goodput
	Average [%]	Peak [%]	FTP flows [Mbps]	Average [%]	Peak [%]	FTP flows [Mbps]	Average [%]	Peak [%]	FTP flows [Mbps]
5	6.8	29.5	4.57	6.95	31	4.58	25.62	32.42	4.57
10	15.2	80.2	4.33	15.22	82	4.38	52.4	62.58	4.27
12	29.3	80.9	2.52	17.66	80	4.09	63.75	74.98	2.32

used, FTP flows obtain the highest goodput, whereas a smaller goodput is achieved when the Simple scheduler or the PI-FBDS regulator are employed. The reason is that FTP flows grab the bandwidth left unused by real-time flows.

B. The Impact of the PI parameters

To investigate the impact of k_{p_i} and T_{I_i} parameters on the behaviour of PI-FBDS, we have repeated the simulations reported above using many parameter sets, all satisfying the stability inequalities (9). In particular, we have varied k_{p_i} between 5 and 25, and T_{I_i} between 4 and 64.

Fig. 7 shows the ratio of admitted flows for all the considered parameter sets and load factors. It shows that when $\alpha = 5$ all the considered parameter sets allow the admission of all flows. This result is not surprising because the network load is small. The differences between the considered parameter sets are evident under high load conditions, i.e., when $\alpha = 12$. In particular, by increasing k_{p_i}, a smaller ratio of flows is admitted: the larger k_{p_i} is, the faster becomes the closed loop control system. As a consequence, high frequency components of allocated bandwidth are more pronounced, and the probability that a new flow finds the channel busy increases. For similar reasons, the ratio of admitted flows diminishes as T_{I_i} decreases. Finally, it is worth noting that for $\alpha = 10$ intermediate results between the cases $\alpha = 5$ and $\alpha = 12$ have been obtained.

Fig. 8 shows that the average superframe utilization during CAPs follows the behaviour of the ratio of admitted flows reported in Fig. 7. The reason is that a higher number of admitted flows

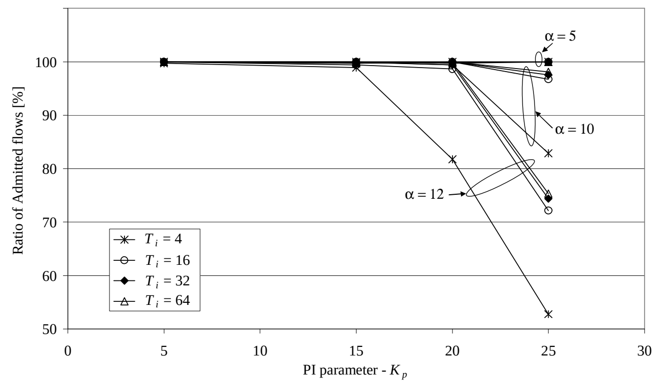

Fig. 7. Ratio of admitted flows obtained using the PI-PBDS algorithm with various parameter sets when $\alpha = 5, 10, 12$.

leads to a higher superframe utilization. On the other hand, Fig. 9 shows that the goodput achieved by the FTP flows, i.e., the amount of data transferred to the FTP receivers over the simulation time, has a complementary behaviour with respect to the average superframe utilization during CAPs. The reason is that FTP flows run over TCP, which grabs the bandwidth left unused by the real-time flows [33].

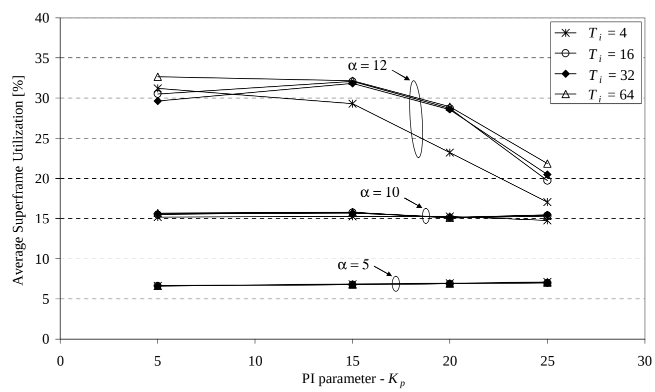

Fig. 8. Average superframe utilization, due to CAPs, using PI-PBDS scheme with various parameter sets when $\alpha = 5, 10, 12$.

Figs. 10-12 show the average one-way packet delays of MPEG4, H.263, and G.729 flows, respectively. For all the considered cases, they are smaller than the 40 ms target delay for video flows and the 16 ms for voice flows. Moreover, when $\alpha = 5$ delays are not affected by parameter set, which confirms that under low loads the system is not sensitive to the parameter set. On the other hand, at high loads, i.e., for $\alpha = 12$, delays diminishes for increasing values of k_{p_i}, and

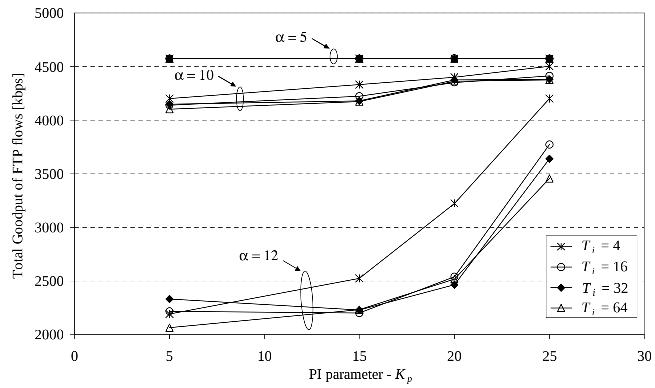

Fig. 9. Goodput achieved by FTP flows, using PI-PBDS scheme with various parameter sets when $\alpha = 5, 10, 12$.

decreasing values of T_{I_i}. In fact the number of admitted flows decreases for increasing values of k_{p_i}, and smaller values of T_{I_i}. For $\alpha = 10$ intermediate results are obtained.

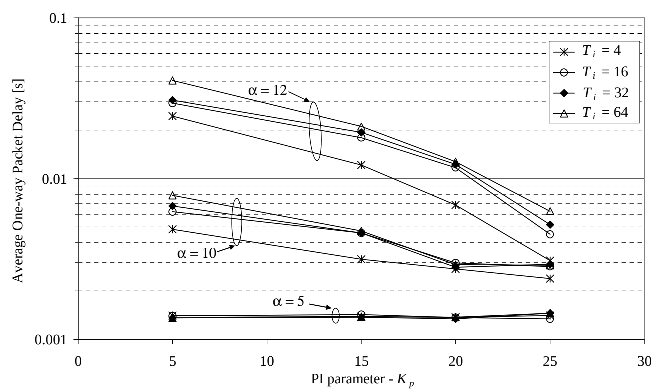

Fig. 10. Average one-way packet delays of MPEG4 flows, using PI-PBDS with various parameter sets when $\alpha = 5, 10, 12$.

Finally, Figs. 13-15 show that the standard deviations of the one-way packet delays follow the same behaviour of their average values reported in Figs. 10-12.

In order to give an insight into the time domain behaviour of PI-FBDS, Fig. 16 reports the superframe utilization using HCCA when three different parameter sets are used under the load factor $\alpha = 12$. It shows that as soon as k_{p_i} increases and T_i diminishes, high frequency components of the superframe utilization grow up, and its average value gets smaller. This confirms what we have shown before, i.e., that for increasing k_{p_i} and decreasing T_i the ratio of admitted flows diminishes due to the high frequency components of the allocated bandwidth

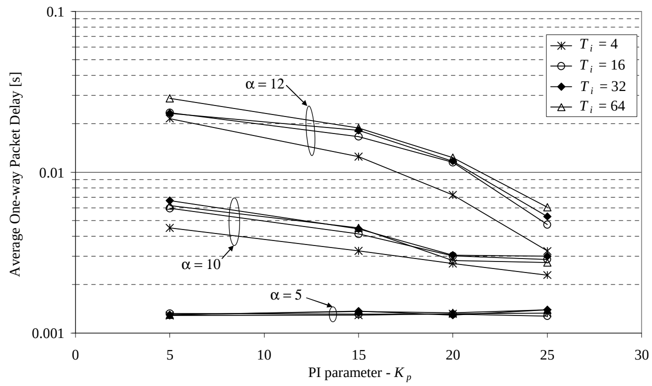

Fig. 11. Average one-way packet delays of H.263 flows, using PI-PBDS with various parameter sets when $\alpha = 5, 10, 12$.

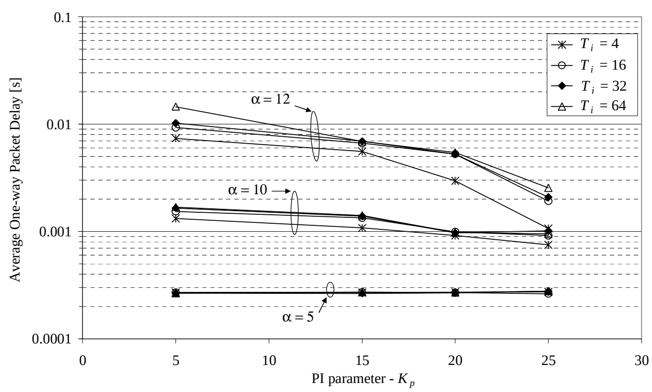

Fig. 12. Average one-way packet delays of G.729 flows, using PI-PBDS with various parameter sets when $\alpha = 5, 10, 12$.

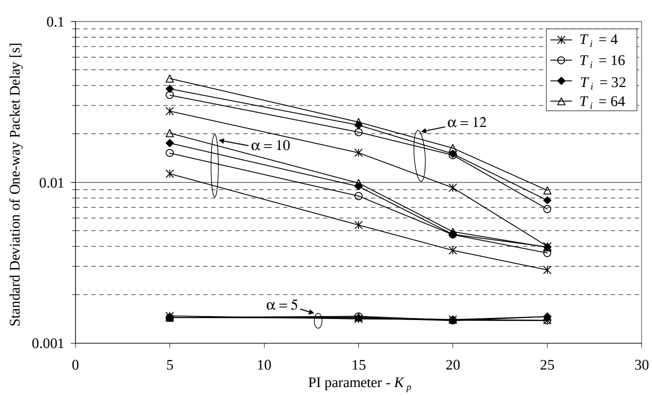

Fig. 13. Standard deviation of one-way packet delays of MPEG4 flows, using PI-PBDS scheme with various parameter sets when $\alpha = 5, 10, 12$.

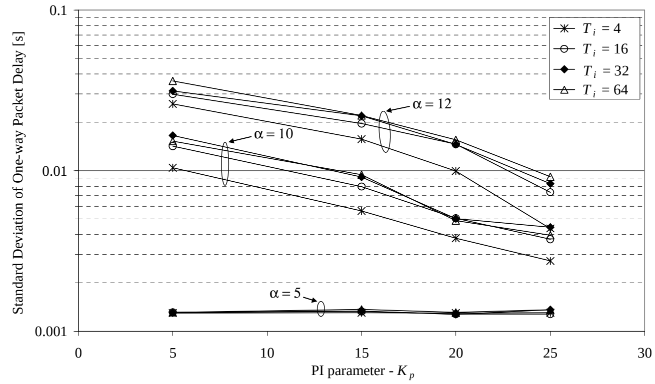

Fig. 14. Standard deviation of one-way packet delays of H.263 flows, using PI-PBDS scheme with various parameter sets when $\alpha = 5, 10, 12$.

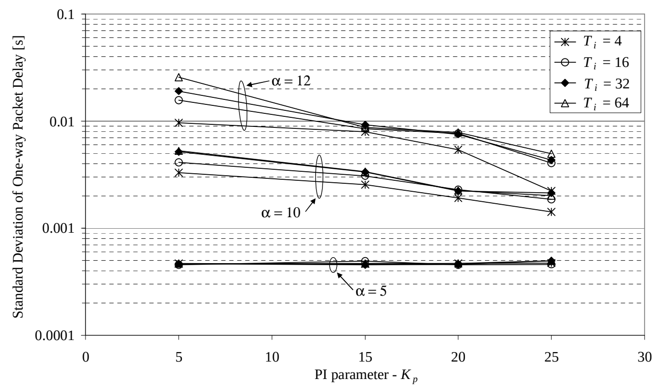

Fig. 15. Standard deviation of one-way packet delays of G.729 flows, using PI-PBDS scheme with various parameter sets when $\alpha = 5, 10, 12$.

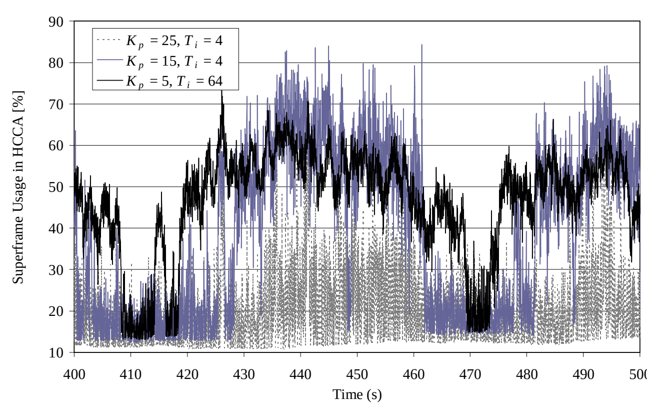

Fig. 16. Superframe utilization using HCCA for different parameter sets when the load factor $\alpha = 12$: $K_p = 5$, $T_i = 64$; $K_p = 15$, $T_i = 4$; $K_p = 25$, $T_i = 4$.

(see Fig. 7 for $\alpha = 12$), which can saturate the WLAN channel, thus causing the rejection of new traffic streams.

V. Conclusion

In this paper a control theoretic framework for addressing the first hop bandwidth allocation issue using the HCCA function of the 802.11e MAC has been proposed. Two dynamic bandwidth allocation algorithms, which have been referred to as FBDS and PI-FBDS, have been proposed along with a CAC mechanism. Simulation results obtained using the *ns*-2 simulator have shown that, unlike the simple scheduler proposed by the 802.11e WG, both FBDS and PI-FBDS generate bounded delays at high network loads also. Moreover, when the PI-FBDS is used, the best trade-off between one-way packet delays and network utilization is achieved.

References

[1] IEEE 802.11, *Information Technology - Telecommun. and Information Exchange between Systems. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN MAC and PHY Specifications*, 1st ed., ANSI/IEEE Std. 802.11, ISO/IEC 8802-11, 1999.

[2] ——, *Supplement to IEEE Standard for Information Technology. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN MAC and PHY Specifications: Higher-Speed Physical Layer Extension in the 5 GHz Band*, IEEE Std 802.11a, ISO/IEC 8802-11:1999/Amd 1:2000(E), 1999.

[3] ——, *Supplement to IEEE Standard for Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE Std 802.11g, 2003.

[4] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," in *European Wireless Conference 2002*, Florence, Italy, Feb. 2002.

[5] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal of Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[6] A. Koepsel, J. P. Ebert, and A. Wolisz, "A Performance Comparison of Point and Distributed Coordination Function of an IEEE 802.11 WLAN in the Presence of Real-Time Requirements," in *Proc. of 7th Intl. Workshop on Mobile Multimedia Communications (MoMuC2000)*, Tokio,Japan, October 2000.

[7] IEEE 802.11 WG, *Draft Amendment to Standard for Information Technology. LAN/MAN Specific Requirements - Part 11: Wireless MAC and PHY Specifications: MAC Quality of Service (QoS) Enhancements*, IEEE 802.11e/D13.0, Jan. 2005.

[8] G. Bianchi and I. Tinniriello, "Analysis of priority mechanisms based on differentiated inter frame spacing in CSMA-CA," in *IEEE 58th Veichular Technology Conference (VTC03), fall*, Orlando, October 2003.

[9] W. Pattara-Atikom and P. Krishnamurty, "Distributed Mechanisms for Quality of Service in Wireless LANs," *IEEE Wireless Communications*, pp. 1536–1284, June 2003.

[10] A. Banchs, X. Perez-Costa, and D. Qiao, "Providing throughput guarantees in IEEE 802.11e Wireless LANs," in *Providing Quality of Service in Heterogeneous Environments, ITC 2003*, Berlin, Germany, August 2003, pp. 1001–1010.

[11] A. Lindgren, A. Almquist, and O. Schelén, “Quality of service schemes for IEEE 802.11 wireless LANs - an evaluation,” *Mobile Networks and Applications*, vol. 8, no. 3, pp. 223–235, June 2003.

[12] Z. Kong, D. H. K. Tsang, B. Bensaou, and D. Gao, “Performance Analysis of IEEE 802.11e Contention-Based Channel Access,” *IEEE Journal of Selected Areas in Communications*, vol. 22, no. 10, pp. 2095–2106, December 2004.

[13] G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo, “Feedback based bandwidth allocation with call admission control for providing delay guarantees in IEEE 802.11e networks,” *Computer Communications, Elsevier*, vol. 28, no. 3, pp. 325–337, February 2005.

[14] Q. Ni, L. Romdhani, and T. Turletti, “A survey of QoS enhancements for IEEE 802.11 wireless LAN,” *Wireless Communications and Mobile Computing*, vol. 4, pp. 547–566, 2004.

[15] L. Romdhami, Q. Ni, and T. Turletti, “Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless Ad-Hoc networks,” in *Proc. of IEEE Wireless Commun. and Networking Conf. (WCNC)*, New Orleans, Louisiana, USA, Mar. 2003.

[16] Y. Xiao, H. Li, and S. Choi, “Protection and guarantee for voice and video traffic in IEEE 802.11e Wireless LANs,” in *IEEE Infocom*, Hong Kong, March 2004.

[17] A. Grilo, M. Macedo, and M. Nunes, “A scheduling algorithm for QoS support in IEEE 802.11e networks,” *IEEE Wireless Communications*, pp. 36–43, June 2003.

[18] S. C. Lo, G. Lee, and W. T. Chen, “An efficient multipolling mechanism for IEEE 802.11 wireless LANs,” *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 764–778, June 2003.

[19] P. Ansel, Q. Ni, and T. Turletti, “Fhcf: A fair scheduling scheme for 802.11e wlan,” Institut National de Recherche en Informatique et en Automatique (INRIA), Tech. Rep. 4883, July 2003.

[20] Ns-2, “Network simulator,” available at http://www.isi.edu/nsnam/ns, 2004.

[21] K. J. Astrom and B. Wittenmark, *Computer controlled systems: theory and design*, 3rd ed. Prentice Hall, 1995.

[22] C. Partridge, *A Proposed Flow Specification*, IETF RFC 1363, Sep. 1992.

[23] R. Braden, D. Clark, and S. Shenker, *Integrated Services in the Internet Architecture: an Overview*, IETF RFC 1633, June 1994.

[24] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Service*, IETF RFC 2475, Dec. 1998.

[25] S. Mascolo, “Congestion control in high-speed communication networks using the smith principle,” *Automatica, Special Issue on Control methods for communication networks*, vol. 35, pp. 1921–1935, December 1999.

[26] M. Grossglauser and D. N. N. Tse, “A framework for robust measurement-based admission control,” *IEEE/ACM Trans. on Networking*, vol. 7, no. 3, pp. 293–309, Jun 1999.

[27] International Telecommunication Union (ITU), *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP)*, ITU-T Recommendation G.729, Mar. 1996.

[28] MPEG-4 Video Group, “Mpeg-4 overview,” available at http://mpeg.telecomitalialab.com/, Mar. 2002.

[29] International Telecommunication Union, *Video coding for low bit rate communication*, ITU-T Recomm. H.263, Feb. 1998.

[30] “Video trace library,” available at http://trace.eas.asu.edu/.

[31] C. Chuah and R. H. Katz, “Characterizing Packet Audio Streams from Internet Multimedia Applications,” in *Proc. of International Communications Conference (ICC 2002)*, New York, NY, April 2002.

[32] W. Wang, S. C. Liew, and V. O. K. li, “Solutions to performance problems in VoIP over a 802.11 wireless LAN,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 1, pp. 366–384, Jan. 2005.

[33] M. Allman, V. Paxson, and W. R. Stevens, “TCP congestion control,” RFC 2581, April 1999.