# A Control Theoretic Scheme for Efficient Video Transmission over IEEE 802.11e EDCA WLANs

PAUL PATRAS and ALBERT BANCHS, Institute IMDEA Networks/University Carlos III of Madrid
PABLO SERRANO, University Carlos III of Madrid

The EDCA mechanism of the IEEE 802.11 standard has been designed to support, among others, video traffic. This mechanism relies on a number of parameters whose configuration is left open by the standard. Although there are some recommended values for these parameters, they are fixed independent of the WLAN conditions, which results in suboptimal performance. Following this observation, a number of approaches in the literature have been devised to set the EDCA parameters based on an estimation of the WLAN conditions. However, these previous approaches are based on heuristics and hence do not guarantee optimized performance. In this article we propose a novel algorithm to adjust the EDCA parameters to carry video traffic which, in contrast to previous approaches, is sustained on mathematical foundations that guarantee optimal performance. In particular, our approach builds upon (i) an analytical model of the WLAN performance under video traffic, used to derive the optimal point of operation of EDCA, and (ii) a control theoretic designed mechanism which drives the WLAN to this point of operation. Via extensive simulations, we show that the proposed approach performs optimally and substantially outperforms the standard recommended configuration as well as previous adaptive proposals.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: control theory, EDCA, IEEE 802.11, video transmission

**ACM Reference Format:**

Patras, P., Banchs, A., and Serrano, P. 2012. A control theoretic scheme for efficient video transmission over IEEE 802.11e EDCA WLANs. ACM Trans. Multimedia Comput. Commun. Appl. 8, 3, Article 29 (July 2012), 23 pages.
DOI = 10.1145/2240136.2240142 http://doi.acm.org/10.1145/2240136.2240142



## 1. INTRODUCTION

IEEE 802.11-based wireless LANs (WLANs) have been widely deployed in the recent years. The use of unlicensed spectrum, the availability of low cost devices, and their ease of management has lead to a plethora of WiFi Access Points, used not only in office environments or as public hot-spots but also to connect residential users and their multimedia devices to the Internet. According to the IEEE

The research leading to these results was funded by the EU's Seventh Framework Programme (FP7-ICT-2009-5) under grant agreement n.258053 (MEDIEVAL project) and by the Spanish Ministry of Science and Innovation under the QUARTET project (TIN2009-13992-C02-01).

Authors' address: P. Patras, A. Banchs, and P. Serrano, Universidad Carlos III de Madrid, Escuela Politécnica Superior, Avenida de la Universidad 30, Leganés (MADRID, E-28911 Leganés (Madrid) Spain, Correspondence email: patras@ieee.org.



DOI 10.1145/2240136.2240142 http://doi.acm.org/10.1145/2240136.2240142

802.11 standard [2007], there are two different channel access mechanisms, a centralized one, known as the Point Coordination Function (PCF), and a distributed one, the Distributed Coordination Function (DCF). However, most of the current WLANs are based on the latter, that is, a CSMA/CA mechanism that only provides with a best effort service, while the PCF mechanism has received relatively little attention from manufacturers.

Although the first physical layer specification supported only 2 Mbps capacity, due to the increasing bandwidth demands extensions were adopted over the years, such that nominal rates of up to 54 Mbps are achievable with, for example, IEEE 802.11g [2003]. This rate increase has enabled the use of WLANs also for real-time applications, such as for example voice over IP, video streaming or video conferencing.[1] However, these delay and bandwidth sensitive applications are properly supported only in over-provisioned scenarios, where the best-effort based scheme of DCF is enough to fulfill the QoS requirements.

In order to overcome this limitation, the revised version of the standard specifies an improved channel access scheme, the Hybrid Coordination Function (HCF), which consists of two access mechanisms, the HCF Controlled Channel Access (HCCA) and the Enhanced Distributed Coordination Access (EDCA) [IEEE 802.11e-2005]. The former is based, like PCF, on a centralized controller that schedules the transmissions in the WLAN, while the latter is an extension of DCF that supports service differentiation through four different Access Categories (namely voice, video, best-effort and background). These Access Categories can be configured with different values of the contention parameters, leading to statistical service differentiation. However, the configuration of both mechanisms is left open, as the standard only specifies a simple scheduler to provide CBR services for the case of HCCA, and a set of recommended values of the contention parameters for EDCA.

The EDCA mechanism is intended to be used for video traffic and, indeed, some specific recommendations for this traffic type are given. However, the use of the fixed set of recommended values for the EDCA parameters results in poor efficiency for most scenarios, as the optimal configuration of the channel access parameters depends on the WLAN conditions, including the number of stations and their load [Bianchi 2000; Banchs et al. 2003]. Thus, when the WLAN is heavily loaded, the performance of real-time applications, and in particular the delay experienced by video traffic, is severely degraded. Following this observation, previous work proposed to improve video performance by adapting the channel access protocol to the WLAN conditions. These works can be classified as follows.

—*Cross-Layer Approaches* [Ksentini et al. 2006; Foh et al. 2007a; He et al. 2008]. Most of these approaches classify the frames of a layered-encoded video according to their relevance, and map them to different ACs. [He et al. 2008] employs a controller to drive the delay to an application-specific reference, by employing packet classification at a newly introduced middleware layer. A major disadvantage of these approaches is their complexity, as they require rather complex interactions between the application and the MAC layers, and moreover they either require specific video sources, or modifications of the protocol stack.

—*Nonstandard Compliant Approaches* [Argyriou 2008; Bucciol et al. 2004; Nafaa and Ksentini 2008]. These approaches have the key drawback of requiring additional changes to the MAC layer and therefore cannot be implemented with current WLAN cards.

—*HCCA Compliant Approaches* [Grieco et al. 2003; Boggia et al. 2007; Yang et al. 2007]. These approaches are compliant with the 802.11 specifications and do not require changes to the standard, but they are based on the centralized mechanism (namely HCCA) which has seen a much smaller deployment than the EDCA mechanism. Moreover, some of them [Yang et al. 2007] rely on feedback

[1] Indeed, most of today's new laptops are provided with an integrated video camera.

information from the clients, which is not readily available within current device drivers and do not guarantee system stability.

—*EDCA Compliant Approaches* [Xiao et al. 2004, 2007; Freitag et al. 2006; Zhang et al. 2008; Chen 2007]. These approaches rely on the EDCA standard mechanism and dynamically update the EDCA parameters and/or the video codec behavior based on the observed WLAN conditions. Their major drawback is that they are based on heuristics and lack analytical support, and hence do not guarantee optimized performance.

In this article we propose a novel algorithm that dynamically adjusts the EDCA configuration to the conditions of the WLAN with the goal of minimizing the video traffic delay. In contrast to the previous approaches, our proposal has the following strengths.

(1) It is tailored to video applications, as our goal is to optimize the delay performance, which results in a better quality of experience (QoE) of the video traffic.
(2) It is based on a well established analytical model of the MAC operation [Foh et al. 2007b], which provides the foundations to guarantee optimal performance.
(3) It requires no additional signaling, since the AP drives the WLAN to the optimal point of operation only by observing the behavior of the WLAN and it is fully standard compliant, therefore can be implemented on available EDCA hardware.
(4) It guarantees simultaneously quick reaction to the changes in the network and stable operation by means of control theory.
(5) It supports graceful degradation of video flows by implementing a priority-based dropping policy, in line with the efforts of IEEE 802.11TGaa [2010] for robust streaming of audio video transport streams.

The rest of the article is organized as follows. In Section 2, we briefly summarize the IEEE 802.11e EDCA mechanism. In Section 3, we analyze the delay performance of EDCA and derive the optimal collision probability that minimizes the delay performance. In Section 4, we introduce our adaptive algorithm which drives the WLAN to the optimal point of operation, by means of a Proportional Integrator controller. The performance of the proposed algorithm and the accuracy of the analytical model are validated by means of simulations in Section 5. In Section 6, we present a prototype implementation of our algorithm with real devices and finally Section 7 concludes the article.

## 2. IEEE 802.11E EDCA

In this section we briefly summarize the EDCA mechanism. This is a CSMA/CA-based protocol that extends DCF to provide service differentiation by means of the parameters that control the way stations access the wireless medium. The channel access of a station is performed by four Channel Access Functions (CAFs), each of them running an independent backoff process which is regulated by a number of configurable parameters. For the configuration of these parameters, the standard groups the CAFs by Access Categories (ACs) and assigns the same configuration to all the CAFs of an AC.

If a station with a new frame to transmit senses the channel idle for a period of time equal to the arbitration interframe space parameter ($AIFS$), the station transmits. Otherwise, if the channel is busy (either immediately or during the $AIFS$ period), the station continues to monitor the channel until it is measured idle for an $AIFS$ time, and then executes a backoff process. $AIFS$ takes a value of the form $DIFS + kT_e$, where $DIFS$ and $T_e$ are constants dependent on the physical layer and $k$ is a nonnegative integer.

When the backoff process starts, stations compute a random number uniformly distributed in the range $(0, CW - 1)$, and initialize their backoff time counter with this value. The $CW$ value is called the contention window, and depends on the number of failed transmission attempts. For the first transmission attempt the minimum contention window ($CW_{min}$) is used. In case of a collision its value doubles, up to a maximum value $CW_{max}$.

As long as the channel is sensed idle the backoff time counter is decremented once every time slot $T_e$. When a transmission is detected on the channel, the backoff time counter is "frozen", and reactivated after the channel is sensed idle for a certain period (equal to $AIFS$ if the transmission is received with a correct Frame Check Sequence (FCS), and equal to $EIFS - DIFS + AIFS$ otherwise). When the backoff time counter reaches zero, the station transmits its frame in the next time slot.

A collision occurs when two or more stations start transmitting simultaneously. An acknowledgment (Ack) frame is used to notify the transmitting station that the frame has been successfully received. If the Ack is not received within a given timeout, the station reschedules the transmission by reentering the backoff process. After a first failed attempt, all the retransmissions of the same frame are sent with the retry flag set in order to avoid duplicates. If the number of failed attempts reaches a predetermined retry limit $R$, the frame is discarded. Once the backoff process is completed (either successfully or unsuccessfully), $CW$ is set again to $CW_{min}$.

When the station gains access to the channel, it is allowed to retain the right to access it for a duration equal to the transmission opportunity limit parameter ($TXOP$). If this parameter is set to zero, a station is allowed to transmit only one frame upon accessing the channel. Using a larger $TXOP$ value helps minimizing the delay experienced by real-time traffic by ensuring that the transmission queues will not grow.

In order to provide service differentiation IEEE 802.11e [2005] recommends different values for the channel access parameters. However, these values are statically set, independently of the network conditions, thus yielding suboptimal performance for most scenarios. The standard also specifies that the Access Point (AP) can periodically broadcast through beacon frames (typically every 100 ms) the EDCA parameters to be used by all stations. In this article, we take advantage of this feature to adjust the EDCA configuration, in order to drive a WLAN operating under video traffic to the optimal point of operation.

## 3. ANALYTICAL MODEL

In this section, we present the analytical model upon which our adaptive algorithm is sustained. We first analyze the delay performance of a WLAN under video traffic and then, based on this analysis, we compute the collision probability that provides optimal delay performance. The algorithm proposed in the next section aims at driving the collision probability to this value.

### 3.1 Parameters Configuration

As discussed in Section 2, the operation of EDCA depends on four configurable parameters, namely $AIFS$, $TXOP$, $CW_{max}$ and $CW_{min}$. Based on the following arguments, we fix the first three parameters when there is only video traffic present in the WLAN.

—$AIFS = DIFS$. We set this parameter to its minimum possible value, as otherwise additional time is unnecessarily lost after every transmission. Indeed, this parameter aims at providing differentiation between different traffic types and it is not needed when there is only one traffic type present in the WLAN.

—$CW_{max} = CW_{min}$. When all parameters are statically set, $CW_{max}$ is typically set larger than $CW_{min}$, so that after a collision the $CW$ increases and thus the probability of a new collision is reduced. However, this is not necessary in our case, as our algorithm dynamically adjusts $CW_{min}$ so that the resulting collision probability corresponds to optimal operation. In addition, if we set $CW_{max}$ larger than $CW_{min}$, the delay of the packets that suffer one or more collision drastically grows, which harms jitter performance.[2]

—$TXOP = TXOP_{max}$. Considering the strict delay requirements of video traffic, it is desirable that, upon accessing the channel, all the waiting packets in the station's queue are transmitted in order to minimize their delay. To achieve this, we set the TXOP parameter to its maximum allowed value.

These settings build on our previous works [Serrano et al. 2007; Banchs and Vollero 2006] where we have shown that the optimal operation of the WLAN can be achieved without utilizing the $AIFS$ and $CW_{max}$ differentiation mechanisms, by solely employing an appropriate configuration of the $CW_{min}$. Additionally, our simulation results included in Appendix B.2 show that the best performance is achieved when TXOP is set to the maximum value. Consequently, we have that the only parameter whose configuration is left open is $CW_{min}$. The rest of this section is devoted to the analysis of performance as a function of this parameter, while in the next section we propose an adaptive algorithm that sets this parameter dynamically. To simplify notation, hereafter, we refer to the $CW_{min}$ parameter with $CW$.

### 3.2 Average Delay

Following this, in this article, we aim at finding the optimal value of the $CW$ parameter. This optimal $CW$ corresponds to a tradeoff between too large and too small $CW$s. Indeed, if stations contend with overly small $CW$s, the collision rate will be very high and therefore delay performance will be penalized. Similarly, if stations contend with too large $CW$s, the channel will be idle most of the time and the delay performance will also be degraded. Therefore, it follows from this that there exists an intermediate $CW$ value that minimizes the average delay of the WLAN; hereafter, we refer to this value as the *optimal CW*.

In the rest of the article we aim at designing an algorithm that drives the $CW$ of the WLAN to its optimal value and thus minimizes the average delay suffered by video frames. As a first step towards this algorithm, we next analyze the delay as a function of the $CW$. The key assumptions behind our analysis are as follows.

—Following the findings of Duffy et al. [2005] and Malone et al. [2007], we neglect the probability that a station accumulates more than one video frame in its transmission queue.

—We assume that the aggregate arrivals follow a Poisson process. Considering a sufficiently large number of stations, and given their independence, this assumption is sustained by the Palm-Khintchine Theorem [Heyman and Sobel 2004].

—We consider that access delays are exponentially distributed. This is supported by the observation that delay is mainly dominated by the number of attempts, which follows a geometric distribution, and that such a discrete distribution can be approximated by an exponential one in the continuous domain.

With these assumptions, the WLAN can be analyzed based on the Markov chain of Figure 1, where state $i$ represents the case where there are $i$ backlogged stations with a video frame to transmit, $\lambda$ is the aggregate arrival rate, computed as the individual arrival rate times the number of stations, denoted by $n$, and $\mu_i$ is the aggregate departure rate at state $i$.

To compute the $\mu_i$'s, we follow the assumption of Foh et al. [2007b] that the aggregate departure rate when there are $i$ backlogged stations can be approximated by the departure rate of the WLAN with $i$

[2]Experiments conducted with $CW_{max} = 2^6 \cdot CW_{min}$ and with $N = 25$ stations, report jitter values of up to 15 times larger than for a fixed $CW$ setting, which is inline with our assumptions.

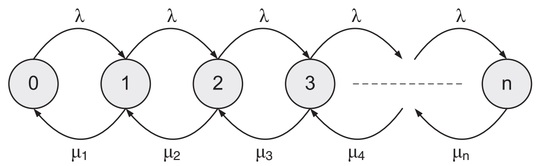


Fig. 1. Markov chain model of the WLAN.

saturated stations, which yields

$$\mu_i = \frac{r_i^{sat}}{L} \tag{1}$$

where $L$ is the average length of a video frame and $r_i^{sat}$ is the total throughput with $i$ saturated stations. $r_i^{sat}$ is computed following Bianchi [2000]

$$r_i^{sat} = \frac{P_s L}{P_s T_s + P_c T_c + P_e T_e}, \tag{2}$$

where $P_s$, $P_c$ and $P_e$ are the probabilities that a slot time contains a successful transmission, a collision and is empty, respectively, and $T_s$, $T_c$ and $T_e$ are the corresponding average slot time durations. The probabilities are computed as

$$P_s = i\tau(1-\tau)^{i-1}, \quad P_e = (1-\tau)^i, \quad P_c = 1 - P_s - P_e, \tag{3}$$

where $\tau$ is the probability that a backlogged station transmits in a randomly chosen slot time, which can be computed as a function of the $CW$ following Bianchi [2000]

$$\tau = \frac{2}{CW+1}. \tag{4}$$

The average slot time durations $T_s$ and $T_c$ can be computed from the video frame length distribution as follows. Let $P_l$ be the probability that the length of a video frame equals $l$. Then,

$$T_s = \sum_l P_l T_{s,l}, \tag{5}$$

where $T_{s,l}$ is the duration of a transmission of a video frame of length $l$. Note that, since a video frame may be larger than the maximum size of a layer 2 (L2) frame, which we denote by $l_{max}$, it may need to be transmitted in several back-to-back L2 frames. Thus,

$$\begin{aligned} T_{s,l} \; = \; & (N-1)\left(T_{PLCP} + \frac{H + l_{max}}{C} + SIFS + T_{ack} + SIFS\right) \\ & + T_{PLCP} + \frac{H + l - (N-1)l_{max}}{C} + SIFS + T_{ack} + DIFS, \end{aligned} \tag{6}$$

where $N = \lceil l/l_{max} \rceil$ is the total number of L2 frames in which the video frame is divided, $T_{PLCP}$ is the Physical Layer Convergence Protocol preamble and header transmission time, $H$ is the L2 overhead (header and FCS), $T_{ack}$ is the duration of the acknowledgment frame and $C$ is the channel bit rate.

To compute $T_c$, we neglect the probability that more than two stations collide. With this assumption, $T_c$ can be computed as

$$T_c = \sum_l \sum_k P_l P_k \max(T_{c,l}, T_{c,k}), \tag{7}$$

where $T_{c,l}$ is the duration of a slot time that contains a collision in which the largest colliding frame is of size $l$. Note that in case the video frame is larger than $l_{max}$, the collision is detected after the first L2 frame transmission and no further L2 frames are sent. Thus,

$$T_{c,l} = T_{PLCP} + \frac{H + \min(l, l_{max})}{C} + EIFS. \tag{8}$$

With this, we can compute the $\mu_i$ values. Once these values have been obtained, the next step is to calculate the state probabilities of the Markov chain. Let $P_i$ be the probability that the Markov chain is in state $i$. From the balance equations, we have

$$P_i = P_{i-1}\frac{\lambda}{\mu_i} \tag{9}$$

and applying this recursively

$$P_i = P_0 \prod_{j=1}^{i} \frac{\lambda}{\mu_j}. \tag{10}$$

By forcing that all $P_i$'s add to 1, we have

$$P_0 = \frac{1}{1 + \sum_{i=1}^{n} \prod_{j=1}^{i} (\lambda/\mu_j)}. \tag{11}$$

From Eqs. (10) and (11), we can compute all state probabilities $P_i$, and from the $P_i$'s we then calculate the average number of backlogged stations,

$$n_b = \sum_{i=1}^{n} i P_i. \tag{12}$$

Finally, by applying Little's formula [Kleinrock 1975], we obtain the average delay

$$D = \frac{n_b}{\lambda}, \tag{13}$$

which terminates the delay performance analysis.

### 3.3 Optimal Collision Probability

We next compute the optimal collision probability that minimizes the average delay calculated in the previous section. By collision probability we mean the conditional probability that a station encounters a collision upon attempting a transmission.

Our optimal collision probability computation is based on the observation that, in order to minimize the average number of backlogged stations (and therefore the delay, since the arrival rates of the Markov chain of Figure 1 are fixed), we need to find the collision probability that maximizes the departure rates $\mu_i$'s.

We next compute the collision probabilities that maximize the different $\mu_i$'s. We first note that in state $i = 1$, where there is only one backlogged station, the collision probability is necessarily zero, since never more than one station will attempt to transmit in this state.

For $i > 1$, we proceed as follows. According to Eq. (1), maximizing $\mu_i$ is equivalent to maximizing $r_i^{sat}$. Eq. (2) can be rearranged to obtain

$$r_i^{sat} = \frac{L}{T_s - T_c + \frac{P_e(T_e - T_c) + T_c}{P_s}}, \tag{14}$$

As $L$, $T_s$, and $T_c$ are constant, maximizing the following expression will result in the maximization of $r_i^{sat}$,

$$\hat{r}_i = \frac{P_s}{P_e(T_e - T_c) + T_c}. \tag{15}$$

Given $\tau \ll 1$, $\hat{r}$ can be approximated by

$$\hat{r}_i = \frac{i\tau - (i(i-1)/2)\tau^2}{i\tau(T_e - T_c) + T_c}. \tag{16}$$

The optimal value of $\tau$, $\tau_{opt}$, that maximizes $\hat{r}$ can then be obtained by

$$\left.\frac{d\,\hat{r}_i}{d\,\tau}\right|_{\tau=\tau_{opt}} = 0, \tag{17}$$

which yields

$$i^2(i-1)(T_c - T_e)\tau^2 + 2i(i-1)T_e\tau + iT_e = 0. \tag{18}$$

Isolating $\tau_{opt}$ from this yields

$$\tau_{opt} = \sqrt{\left(\frac{T_e}{i(T_c - T_e)}\right)^2 + \frac{2T_e}{i(i-1)(T_c - T_e)}} - \frac{T_e}{i(T_c - T_e)}, \tag{19}$$

Given $T_e \ll T_c$, we finally obtain the following approximate solution for the optimal $\tau$,

$$\tau_{opt} \approx \frac{1}{i}\sqrt{\frac{2T_e}{T_c}}. \tag{20}$$

With $\tau_{opt}$, the corresponding collision probability is equal to

$$p_{col} = 1 - (1 - \tau_{opt})^{i-1} = 1 - \left(1 - \frac{1}{i}\sqrt{\frac{2T_e}{T_c}}\right)^{i-1}, \tag{21}$$

which can be approximated by

$$p_{col} \approx 1 - e^{-\sqrt{\frac{2T_e}{T_c}}}. \tag{22}$$

Note that the key result from the above approximations is that $p_{col}$ does not depend on the number of backlogged stations $i$.

From this, we conclude the following.

—When a station transmits in state $i = 1$, the collision probability is always zero.

—When a station transmits in a state $i > 1$, the optimal collision probability is equal to $p_{col}$, which is a constant independent of $i$.

The combination of these two leads to the following collision probability seen by a station in a WLAN under optimal operation:

$$p_{opt} = P(i = 1) \cdot 0 + P(i > 1) \cdot p_{col} = P(i > 1)p_{col}, \tag{23}$$

where $P(i = 1)$ is the probability that a transmission by a station is attempted in state $i = 1$ and $P(i > 1)$ is the probability that a it is attempted in state $i > 1$.

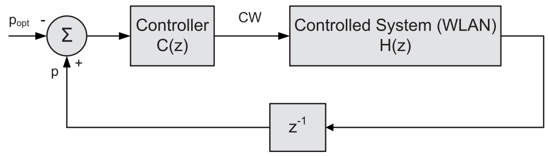


Fig. 2. Control system.

The remaining challenge to obtain $p_{opt}$ is the computation of $P(i > 1)$. We want to compute this probability by using only data that can be easily measured at the AP. To this aim, we make the following approximations: i) we assume an infinite number of stations, and ii) we neglect the protocol overhead on the $\mu_i$'s by taking $\mu_i = 1/T_s \; \forall i$. With these approximations,

$$P_i = \left(\frac{\lambda}{\mu}\right)^{i-1} P_1 \tag{24}$$

and

$$P(i > 1) = 1 - \frac{P_1}{\sum_{j=1}^{n} P_j} = \frac{\lambda}{\mu}. \tag{25}$$

Finally, combining the preceding equations, we obtain

$$p_{opt} = p_{col} \frac{\lambda}{\mu}, \tag{26}$$

which terminates the analysis of the optimal collision probability. This expression represents the theoretical optimal at which we would like our system to operate. We note that the expression obtained in Eq. (26) depends only on the parameters $\lambda$, $\mu$ and $T_c$ which can be easily measured at the AP as we will show in the next section.

## 4. ADAPTIVE ALGORITHM

In this section, we present our adaptive algorithm. This algorithm runs at the AP and consists of the following two steps, which are executed iteratively.

—During each beacon interval (100 ms), the AP measures the collision probability of the WLAN resulting from the current $CW$ configuration.

—At the end of the period, the AP computes the new $CW$ configuration based on the measured collision probability and distributes it to the stations in the new beacon frame.

Our adaptive algorithm relies on a *Proportional Integrator* (PI) *controller* to drive the WLAN to its optimal point of operation. We note that previous works have successfully employed a PI controller to address performance issues in communication networks [Hollot et al. 2001; Cavendish et al. 2004]. A key advantage of using a PI controller is that it is simple to design, configure and implement with existing hardware, as we show in Section 6.

In the following, we first describe our system from a control theoretical standpoint. Next, we analyze the system by linearizing the behavior of the WLAN. Finally, we use this analysis to adequately configure the parameters of the PI controller.

### 4.1 Control System

Our system can be regarded from a control theoretic perspective as the composition of the two modules depicted in Figure 2.

—The *controller* $C(z)$ is located at AP and implements the adaptive algorithm that controls the WLAN. Our proposal uses a classical scheme from discrete-time control theory for the controller module, namely the PI controller. The AP estimates the collision probability and provides it to the controller, which takes as input the difference between the estimated collision probability and its desired value as given by Eq. (26). With this input, the controller computes the $CW$ value.

—The *controlled system* $H(z)$ is the WLAN system itself. As specified by the standard, the AP distributes the new $CW$ configuration to the stations every 100 ms.

The transfer function of the controller is given by

$$C(z) = K_p + \frac{K_i}{z-1}. \tag{27}$$

With this transfer function, at every beacon interval $t$, the controller will take as input the estimated error signal $e = p - p_{opt}$ and give as output the new $CW$ value to be used by the contending stations.

$$CW[t] = K_p \cdot e[t] + K_i \sum_{k=0}^{t-1} e[k]. \tag{28}$$

Note that implementing Eq. (28) would be highly inefficient as it would require storing all error samples in the past. A much more efficient implementation that only requires storing the previous values of $CW$ and $e$ is

$$CW[t] = CW[t-1] + K_p \cdot e[t] + (K_i - K_p) \cdot e[t-1]. \tag{29}$$

The estimated error signal $e$ is the difference between the actual collision probability $p$ observed in the WLAN and the target value $p_{opt}$ given in Eq. (26) which yields the optimal performance. In order to compute the error signal $e$, we first need to estimate the collision probability $p$ considering only information available at the AP without requiring any modifications to the station nodes. The estimation of the collision probability is performed at the AP over a 100-ms period as follows. Let $S$ be the number of frames received by the AP during this period with the retry flag unset, and $R$ be the number of frames received with the retry flag set. Then, if we assume that no frames are discarded due to reaching the retry limit, the collision probability $p$ can be computed as

$$p = \frac{R}{R+S} \tag{30}$$

since this is precisely the probability that the first transmission attempt of a frame collides.

In addition to this, the AP also needs to compute the optimal collision probability $p_{opt}$ as given by Eqs. (22) and (26), which requires the computation of $\lambda$, $\mu$ and $T_c$. These parameters are estimated by the AP over each 100-ms period as follows: $\lambda$ is measured by counting the number of video frames received ($R+S$) during the period, $\mu$ is computed from the average length of the frames received during the period, and $T_c$ is calculated by applying Eq. (7) to the received frames.

Note that with the above, the AP can measure the collision probability and compute its optimal value by simply analyzing the frames successfully received, which can be easily done with no modifications to the AP's firmware and hardware.

Based on the measurements taken by the AP, the controller adjusts the $CW$ parameter to drive the collision probability to the optimal value. In order to provide a safeguard against too large and too small values of the $CW$, we force that $CW$ can neither take values below $CW_{lb} = 16$ (which is the minimum standard recommendation for video traffic) nor larger than $CW_{ub} = 1024$ (which is the maximum $CW$ value for best-effort traffic). In the rest of the article we assume that the $CW$ always takes values within these bounds and do not further consider this effect.

### 4.2 Transfer Function Characterization

In order to analyze our system from a control theoretic standpoint, we need to characterize the WLAN with a transfer function that takes the $CW$ as input and gives the collision probability $p$ as output. Since the collision probability is measured every 100-ms interval, we can safely assume that the obtained measurement corresponds to stationary conditions and therefore the system does not have any memory. With this assumption and the analysis of Section 3,

$$p = \sum_i P(i)\left(1 - (1-\tau)^{i-1}\right), \tag{31}$$

where $P(i)$ is the probability that a transmission is attempted at state $i$ and $\tau$ is a function of the $CW$,

$$\tau = \frac{2}{CW+1}. \tag{32}$$

Thus, there exists a nonlinear relationship between $p$ and $CW$. In order to express this relationship as a transfer function, we further proceed with linearizing it when the system is perturbed around its stable point of operation (note that a similar approach has been used in for example Hollot et al. [2001] and Patras et al. [2009]),[3]

$$CW = CW_{opt} + \delta CW, \tag{33}$$

where $CW_{opt}$ is the $CW$ value that yields the optimal collision probability $p_{opt}$ given by Eq. (26).

The oscillations of the collision probability around its point of operation $p_{opt}$ can be approximated by

$$p \approx p_{opt} + \frac{\partial p}{\partial CW}\delta CW. \tag{34}$$

This partial derivative can be computed as

$$\frac{\partial p}{\partial CW} = \frac{\partial p}{\partial \tau}\frac{\partial \tau}{\partial CW}. \tag{35}$$

Eq. (31) can be approximated by

$$p \approx \sum_i P(i)(i-1)\tau \tag{36}$$

from which

$$\frac{\partial p}{\partial \tau} \approx \sum_i P(i)(i-1). \tag{37}$$

Additionally, we have

$$\frac{\partial \tau}{\partial CW} = -\frac{2}{CW^2}. \tag{38}$$

By taking these two partial derivatives and using the approximation $\tau \approx 2/CW$, we obtain

$$\frac{\partial p}{\partial CW} \approx -\sum_i P(i)(i-1)\frac{\tau^2}{2}. \tag{39}$$

[3]By linearizing the WLAN behavior around its stable point of operation, we accurately model the behavior of the transfer function around the point of operation, but we may not be accurate in regions far from this point. As a result, our analysis guarantees only local stability.

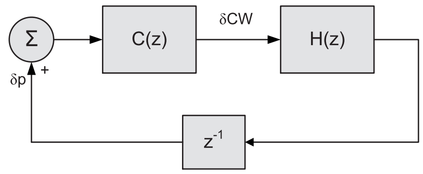


Fig. 3. Linearized system.

Since at the stable point of operation $\tau = \tau_{opt}$ we have from Eq. (21) $p_{col} \approx (i-1)\tau_{opt}$ for $i > 1$, the above can be expressed as

$$\frac{\partial p}{\partial CW} \approx -P(i > 1) p_{col} \frac{\tau_{opt}}{2}, \tag{40}$$

and combining it with Eq. (23) yields

$$\frac{\partial p}{\partial CW} \approx -\frac{p_{opt}\tau_{opt}}{2}. \tag{41}$$

If we now consider the transfer function that allows us to characterize the perturbations of $p$ around its stable point of operation as a function of the perturbations in $CW$,

$$\delta P(z) = H(z)\,\delta CW(z), \tag{42}$$

we obtain from Eqs. (34) and (41) the following expression for the transfer function,

$$H(z) = -\frac{p_{opt}\tau_{opt}}{2}. \tag{43}$$

This linearized model is depicted in Figure 3. Note that, as compared to the model of Figure 2, only the perturbations around the stable operation point are considered:

$$\begin{cases} p = p_{opt} + \delta p \\ CW = CW_{opt} + \delta CW \end{cases}. \tag{44}$$

### 4.3 Controller Configuration

In what follows, we compute the configuration of the PI controller. According to Eq. (27), the transfer function of the PI controller depends on two parameters which need to be configured: $K_p$ and $K_i$. The objective when configuring these parameters is to achieve a proper tradeoff between speed of reaction to changes and stability. For this purpose, we use the Ziegler-Nichols method [Franklin et al. 1997]. This method works as follows. First, we compute the parameter $K_u$, defined as the $K_p$ value that leads to instability when $K_i = 0$, and the parameter $T_i$, defined as the oscillation period under these conditions. Then, $K_p$ and $K_i$ are configured as follows:

$$\begin{cases} K_p = 0.4 K_u \\ K_i = \frac{K_p}{0.85 T_i} \end{cases}. \tag{45}$$

In order to compute $K_u$ we proceed as follows. The system is stable as long as the absolute value of the closed-loop gain is smaller than 1,

$$|H(z)C(z)| = K_p \frac{p_{opt}\tau_{opt}}{2} < 1, \tag{46}$$

which yields the following upper bound for $K_p$,

$$K_p < \frac{2}{p_{opt}\tau_{opt}}. \tag{47}$$

This expression depends on $\tau_{opt}$, which is not known by the AP. Since we want to find an upper bound that can be computed at the AP, we proceed as follows. From Eq. (21), we have that $\tau_{opt}$ is never larger than $p_{col}$. With this observation, we obtain the following tighter upper bound:

$$K_p < \frac{2}{p_{opt}\, p_{col}}. \tag{48}$$

Following this, we take $K_u$ as the value where the system may turn unstable (given by the previous equation),

$$K_u = \frac{2}{p_{opt}\, p_{col}}, \tag{49}$$

and set $K_p$ according to Eq. (45). Thus,

$$K_p = \frac{0.4 \cdot 2}{p_{opt}\, p_{col}}. \tag{50}$$

For the $K_p$ value that turns the system unstable, the following holds:

$$H(z)C(z) = -1. \tag{51}$$

With such a closed-loop transfer function, a given input value changes its sign at every time interval, yielding an oscillation period equal to two intervals ($T_i = 2$). Consequently, from Eq. (45), we obtain

$$K_i = \frac{0.4}{0.85\, p_{opt}\, p_{col}}, \tag{52}$$

which completes the configuration of the PI controller. The stability of this configuration is guaranteed by Theorem 1, included in the Appendix.

## 5. PERFORMANCE EVALUATION

We validated the proposed algorithm by conducting an extensive set of simulations in order to assess the delay performance of the adaptive scheme, the robustness of the underlying analytical model and the configuration of the controller. For this purpose, we have extended the simulator used by Patras et al. [2009] and Banchs and Vollero [2006]; this is an event-driven simulator developed in OMNET++[4] that closely follows the details of the MAC protocol of 802.11e EDCA. The source code of the simulator and basic use instructions are available online at our OWSiM project page.[5]

For all the experiments we have used the physical layer parameters of IEEE 802.11b [1999]. In order to evaluate the performance of our adaptive scheme under video traffic we considered three of the most widely used codec types: H.263, MPEG-4 and H.264. The frame size distribution of the H.263 and MPEG-4 streams were extracted from the video traces of the films *Aladdin* and *Star Wars IV*, respectively, which are available from the Video Traces Library.[6] The H.263 video was a VBR encoded sequence with an unspecified target bitrate and a 20 fps average frame rate. The MPEG-4 trace had a fixed frame rate of 25 fps. We have also analyzed the operation of the adaptive algorithm under CBR video, using one of the 30 fps encoded H.264 test sequences of Lambert et al. [2006]. The average bitrates of these video sequences are 245 kbps, 288 kbps and 300 kbps, respectively. In our simulations we consider that all active stations are transmitting video traffic and no other traffic types are present in the WLAN. For the obtained results, averages and 95% confidence intervals are given.[7]

[4]http://www.omnetpp.org/.
[5]http://enjambre.it.uc3m.es/~ppatras/owsim/.
[6]http://trace.eas.asu.edu/.
[7]Additional results that validate the performance of our proposal under mixed traffic conditions can be found in the appendix.

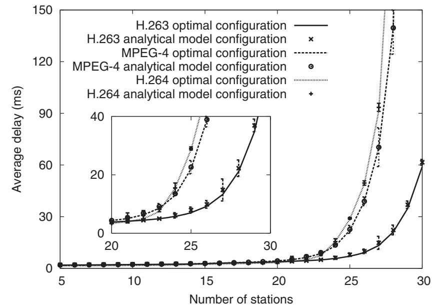


Fig. 4. Validation of the delay model.

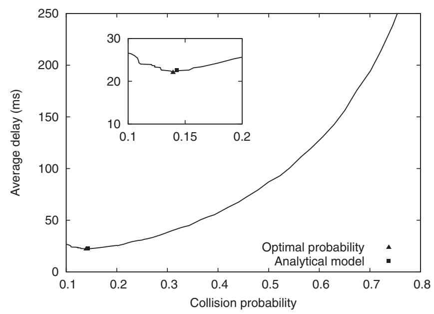


Fig. 5. Optimal collision probability.

## 5.1 Validation of the Analytical Model

We first validated the accuracy of the proposed analytical model upon which the adaptive algorithm is based. In particular, we verified that delay is minimized when the collision probability equals the optimal value given by Eq. (26), which is the basis of our analysis. To this aim, we simulated the average delay and collision probability from two different $CW$ configurations:

—the $CW$ value that yields a collision probability equal to the optimal collision probability given by our analysis (hereafter we refer to this configuration as the *analytical model configuration*);

—the $CW$ value that gives the smallest average delay, obtained from an exhaustive search on all the possible configurations of the $CW$ parameter (hereafter the optimal configuration).

Following our analysis of Section 3, the configuration resulting from the optimal collision probability should minimize the average delay, and therefore the delay resulting from the two above configurations should be very similar. Figure 4 shows the delay performance resulting from the two configurations for a varying number of stations and the different codecs considered. We observe that in all cases the two configurations provide a very similar delay performance, which validates our analytical model.

To show that the collision probability resulting from the optimal configuration is close to the optimal collision probability computed by our analysis, we plotted in Figure 5 the average delay as a function of the collision probability, for a WLAN with 25 stations sending each of them MPEG-4 video traffic. From the plot, we can see that the optimal collision probability given by our analysis (shown with a square) is very close to the collision probability for which the average delay is minimized (shown with a triangle).

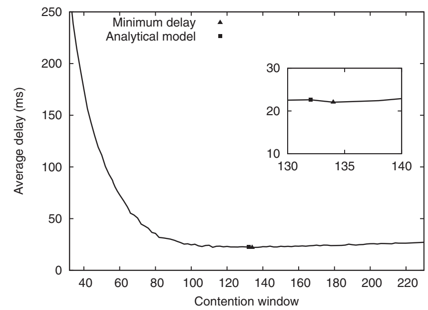


Fig. 6. $CW$ configuration.

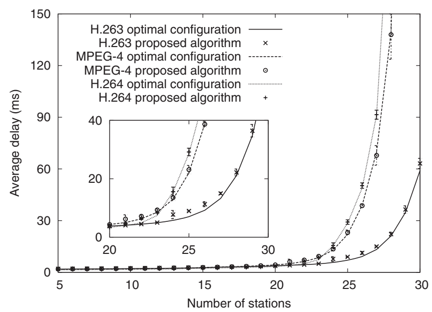


Fig. 7. Delay performance of the proposed algorithm.

To gain further insight into the $CW$ configuration resulting from our analytical model, we plotted in Figure 6 the average delay as a function of the $CW$ for the same scenario as above with 25 stations and MPEG-4 traffic. We observe that the $CW$ configuration resulting from our analytical model is very close to the optimal one that yields the minimum delay, which further validates our analysis.

## 5.2 Adaptive Algorithm Performance

The main objective of our adaptive algorithm is to minimize the average delay of the WLAN. In order to validate that this objective is met, we compared the delay performance of a WLAN which implements our adaptive algorithm against the optimal configuration resulting from the search performed in the previous subsection. Results are depicted in Figure 7. We observe that the proposed mechanism achieves a delay performance almost identical to the minimum given by the optimal configuration, regardless of the codec used. We conclude that our algorithm fulfills its main objective of minimizing the delay for any video traffic pattern.

Note that the optimal configuration against which we compare our approach is the result of an exhaustive search and requires a priori knowledge of the number of active stations and their traffic pattern, which challenges its practical use. In contrast, the adaptive algorithm that we propose does not require any kind of a priori knowledge since it adjusts the WLAN configuration based only on the measurements taken by the AP.

## 5.3 Stability

One of the objectives of the configuration of the PI controller presented in Section 4.1 is to guarantee stable behavior of the system. To validate whether this objective is met, we analyzed the evolution of the $CW$ (our control signal) with our $\{K_p, K_i\}$ setting and for a larger configuration of these parameters,

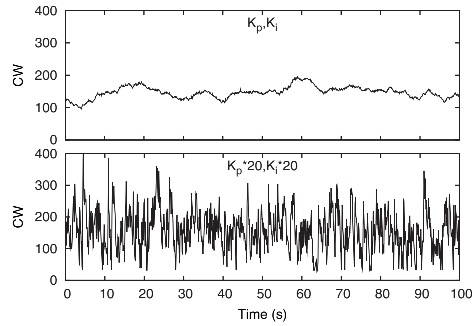


Fig. 8. Stability evaluation.

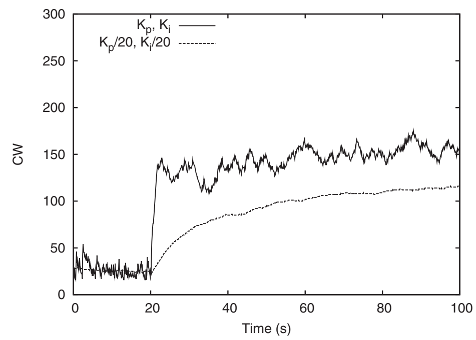


Fig. 9. Speed of reaction to changes.

in a WLAN with 25 stations, each sending MPEG-4 video traffic. From the results given in Figure 8 we observe form this figure that with the proposed configuration (label "$K_p$, $K_i$"), the $CW$ only has minor deviations around its stable point of operation, while if a larger setting is used (label "$K_p * 20, K_i * 20$"), the $CW$ has a strong unstable behavior with drastic oscillations. We conclude that the proposed configuration achieves the objective of guaranteeing stability.

### 5.4 Speed of Reaction to Changes

The other objective of the designed PI controller is to react quickly to changes in the WLAN. To verify whether this objective is fulfilled we ran the following experiment. We had a WLAN with 20 active stations sending MPEG-4 traffic and at $t = 20$ s, we added 5 more stations. We plot the evolution of the $CW$ for our $\{K_p, K_i\}$ setting in Figure 9 (label "$K_p$, $K_i$"). The system reacts fast to the changes on the WLAN, as the $CW$ reaches the new value almost immediately.

We have already shown in the previous section that large values for the parameters of the controller lead to unstable behavior. To analyze the impact of small values for these parameters, we plot on the same figure the behavior of the $CW$ for a $\{K_p, K_i\}$ setting 20 times smaller (label "$K_p/20, K_i/20$"). We observe that with such setting the system reacts too slow to the changes of the conditions on the WLAN.

### 5.5 Graceful Degradation of Video Quality

The recently created IEEE 802.11TGaa [2010] is standardizing a set of mechanisms to better support video streaming in WLANs. One of these mechanism consists of the so-called graceful degradation of

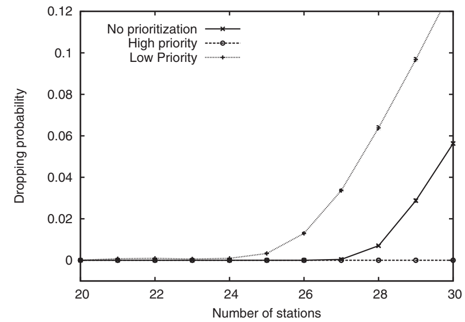


Fig. 10. Dropping probability evaluation.

video flows, whose purpose is to first discard less critical frames in case of congestion. Following these lines, in this section, we illustrate how our algorithm can be extended to support this new feature. This extension consists of introducing a queue-size threshold $Q_{th}$ which, if reached, triggers the discard of arriving frames marked with low priority. To assess the advantages of this enhancement, in Figure 10 we measure i) the dropping probability when there is no support for graceful degradation, and ii) the dropping probability of high-priority and low-priority frames, respectively, for the case of $Q_{th} = Q_{max}/2$. As the figure illustrates, this mechanism prevents high-priority frames from being discarded even in case of large traffic loads, thereby showing its ability to support a graceful degradation of video traffic.

### 5.6 Comparison Against Other Approaches

In order to better assess the advantages of our proposal, we compared it against the following approaches: i) the recommended configuration of IEEE 802.11e [2005], ii) the scheme proposed by [Nafaa and Ksentini 2008] (labeled as "Nafaa") and iii) two other standard compliant proposals, namely the one by [Xiao et al. 2004] (labeled as "Xiao")[8] and the one in Freitag et al. [2006] (labeled as "Freitag"), respectively.

Figure 11 depicts the average delay resulting from each of the above approaches as a function of the number of stations in the WLAN, and Table I shows the average total throughput that can be supported by each of the approaches while guaranteeing an average delay below the playback deadline of 150 ms.[9] We conclude from these results that our algorithm substantially outperforms all other approaches both in terms of delay and throughput.

Additionally, we compared the performance of our algorithm against the EDCA configuration and the other approaches in terms of perceptual quality of the reconstructed video, by evaluating the Mean Opinion Score (MOS) for different number of stations. For this purpose we assumed the same playback deadline of 150 ms. Based on this constraint we considered that the frames which experience delays above this limit are discarded by the decoder. With the obtained packet loss ratio we computed the MOS of the received sequence according to the method given in ITU-T [2007]. The results are shown in Figure 12. We conclude that our algorithm outperforms the standard recommended configuration as well as the other similar approaches, both in terms of average delay and perceptual video quality, being able to accommodate a substantially larger throughput (approximately 20%).

[8]The approaches by Xiao et al. [2004, 2007] are very similar and thus yield comparable performance.
[9]This is the maximum one-way delay as recommended by ITU-T [2001].

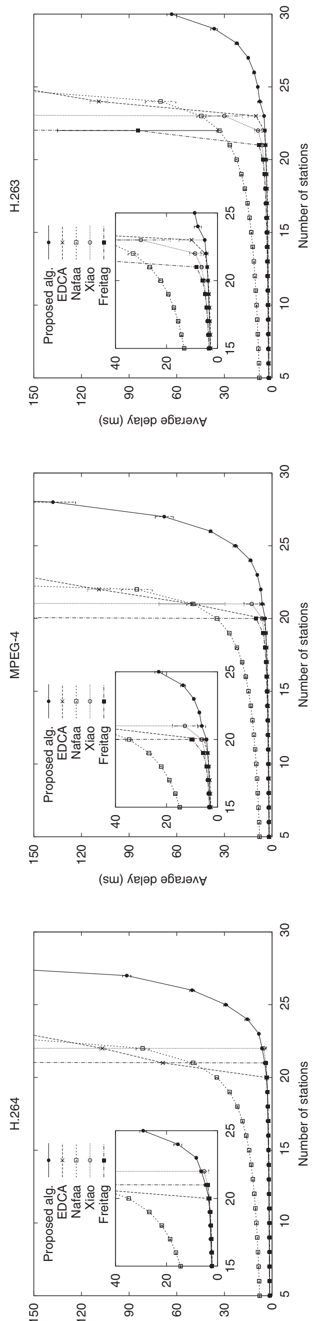


Fig. 11. Comparison against other approaches.

Table I. Throughput Evaluation

| Codec | Total throughput [Mbps] | | | | |
|---|---|---|---|---|---|
| | Proposed alg. | EDCA | Nafaa | Xiao | Freitag |
| H.263 | 7.365 | 5.890 | 5.892 | 5.636 | 5.396 |
| MPEG-4 | 8.062 | 6.334 | 6.329 | 6.050 | 5.758 |
| H.264 | 8.095 | 6.896 | 6.598 | 6.595 | 6.297 |

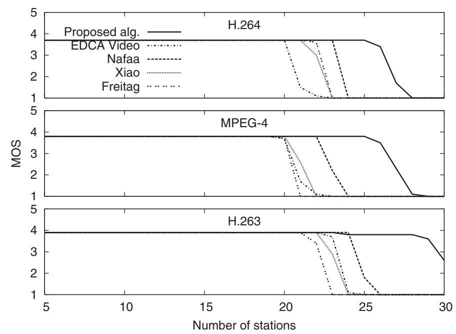


Fig. 12. MOS evaluation.

## 6. PROTOTYPE IMPLEMENTATION

A key advantage of our proposal as compared to existing approaches is that it can be implemented with current off-the-shelf devices without introducing any modifications into their hardware or driver. In order to show this feature, we have developed a prototype implementation of our algorithm and assessed its performance with real video traffic and real users. In this section, we provide a description of this implementation and discuss the results of the conducted experimental evaluation under a mixed traffic scenario.

Our implementation is based on Debian Linux kernel 2.6.26 and the popular open source MadWifi v0.9.4 driver.[10] The source code of our prototype is available online.[11] Note that a similar implementation approach has been used in Patras et al. [2010] for the distributed optimal configuration of 802.11 stations under data traffic. As depicted in Figure 13, our algorithm runs at the AP as a user-space application and relies on existing `IOCTL` calls to communicate with the driver and to signal the CW configuration to the stations. We also exploit the capability of the MadWifi driver to support multiple virtual devices operating in different modes (master/monitor) with a single physical interface.

The first step in the execution of the adaptive algorithm is the estimation of the collision probability $p$ in the WLAN. In our implementation this is computed according to Eq. (30) by the *Frame Sniffer* module, which uses a virtual device configured in promiscuous mode to monitor the retry flag of all the frames transmitted by the stations during a beacon interval. With the estimated collision probability resulting from the current CW configuration and the target optimal value given by Eq. (26), the CW configuration module computes the new CW setting every 100 ms, by applying Eq. (29). The CW configuration is then updated using a private `IOCTL` call and propagated to the stations with the next scheduled beacon frame. As specified by the standard, the stations will extract the configuration provided by the AP and employ it to access the channel until the next update.

[10] http://madwifi-project.org/.
[11] http://enjambre.it.uc3m.es/~ppatras/code/adaptive_video.tar.gz.

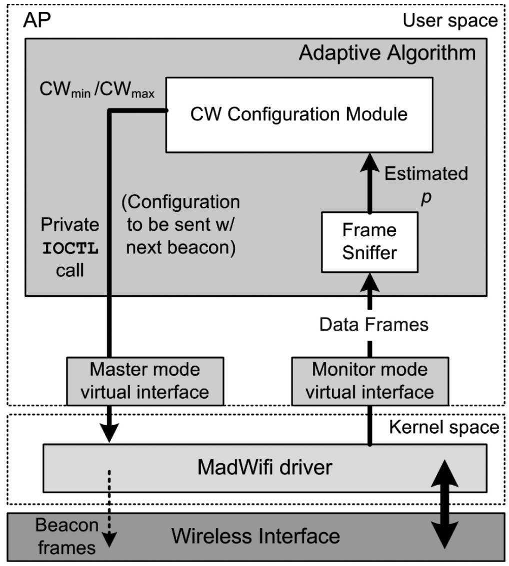


Fig. 13. Prototype implementation.

In order to validate our implementation, we deployed a simple testbed consisting of three laptops equipped with Atheros AR5212 cards operating in 802.11b mode, one acting as an AP and the other two as stations. The algorithm ran at the AP, while the two stations were unmodified 802.11 nodes that streamed video towards the AP using the VLC multimedia framework[12] and, simultaneously, transmitted saturated UDP flows using the `iperf` tool,[13] thus emulating best effort data transfers. As test video sequences we utilized a 30 seconds fragment from an Ice Age 3 trailer, encoded in MPEG-2 format at 2 Mbps. We ran 2 sets of experiments: in the first set, we employed our algorithm, while during the second one we used the default EDCA configuration. In each case, we launched 10 consecutive streaming sessions, recording the received sequences and logging the broadcast CW configuration for the video and best-effort access categories.

To assess the improvement of the video performance provided by our algorithm, a group of 20 people watched in random order the sequences streamed with both approaches and subjectively ranked the perceived quality for each video session on a scale ranging from 1 to 5, with 5 corresponding to the maximum QoE. With these ratings we computed the resulting MOS for the transmissions with the default EDCA configuration and with our algorithm, respectively. As shown in Figure 14, which depicts the average MOS values and their standard deviation for each case, our algorithm is able to significantly improve the quality of the received video by providing users with substantially better QoE.

To gain further insights into the behavior of our algorithm in a real environment and understand how the improved performance is achieved, we examined the evolution of the CW employed by our proposal throughout the duration of the experiments for video and best effort traffic, as compared to the configuration used by the standard. As depicted in Figure 15, our approach ensures a higher level of differentiation between the two types of traffic by employing larger CW values for the best effort flows, and thus enhances the quality of video. In contrast, the default EDCA setting is not sufficient to ensure proper quality for video traffic as previously shown.

[12]http://www.videolan.org/vlc/.
[13]http://sourceforge.net/projects/iperf/.

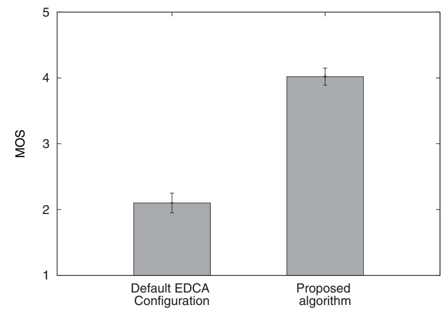


Fig. 14. MOS evaluation of the prototype.

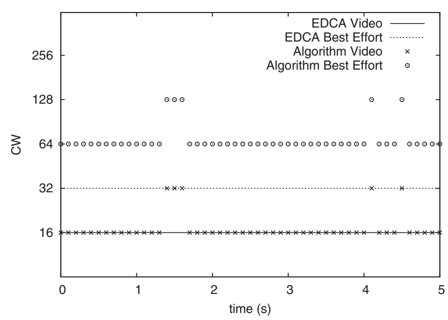


Fig. 15. CW differentiation.

## 7. CONCLUSIONS

In this article, we have proposed a novel algorithm to optimally adjust the configuration of the EDCA parameters for a WLAN operating under video traffic. The key design features of our approach are: *i*) we do not require any a priori knowledge about the number of active sources or their traffic patterns, *ii*) the approach is fully compatible with the 802.11e EDCA standard, and *iii*) the AP only needs to look at the successfully received frames, which can be easily done with no modifications of the hardware and firmware, as our prototype implementation demonstrates.

To design our algorithm, we have analyzed an 802.11e EDCA WLAN under video traffic and computed its optimal point of operation. A key observation resulting from our analysis is that the collision probability that yields optimal performance can be computed as a function of few data that can be easily measured at the AP. Following this observation, our approach adaptively adjusts the EDCA configuration to drive the collision probability to the optimal value as given by the analysis. The proposed mechanism is based on a PI controller, which has the key advantage being simple to design, configure and implement with existing hardware.

The performance of our approach has been exhaustively validated by means of simulations and a prototype implementation. From this study, we conclude that: i) the optimal point of operation corresponds to the collision probability resulting from our analysis, which validates the analytical model upon which the approach is sustained, ii) our adaptive algorithm yields optimal performance, as there is no other configuration that provides (significantly) better performance, iii) the chosen settings of the PI controller perform well both in terms of stability and speed of reaction to changes, iv) the proposed approach outperforms very substantially the standard's recommended configuration, and v) our algorithm is easily deployable with existing hardware.

## ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

### ACKNOWLEDGMENTS

The authors are thankful to the anonymous referees for their valuable comments which greatly helped in improving the article.

### REFERENCES

ARGYRIOU, A. 2008. Error-resilient video encoding and transmission in multirate wireless LANs. *IEEE Trans. Multimed. 10,* 5, 691–700.

ASTRÖM, K. AND WITTENMARK, B. 1990. *Computer-Controlled Systems: Theory and Design* 2nd ed. Prentice-Hall, Inc., Upper Saddle River, NJ.

BANCHS, A., PÉREZ-COSTA, X., AND QIAO, D. 2003. Providing throughput guarantees in IEEE 802.11e wireless LANs. In *Proceedings of the 18th International Teletraffic Congress (ITC18)*.

BANCHS, A. AND VOLLERO, L. 2006. Throughput analysis and optimal configuration of 802.11e EDCA. *Comput. Net. 50,* 11, 1749–1768.

BIANCHI, G. 2000. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Select. Areas Comm. 18,* 3, 535–547.

BOGGIA, G., CAMARDA, P., GRIECO, L., AND MASCOLO, S. 2007. Feedback-based control for providing real-time services with the 802.11e MAC. *IEEE/ACM Trans. Netw. 15,* 2, 323–333.

BUCCIOL, P., DAVINI, G., MASALA, E., FILIPPI, E., AND DE MARTIN, J. 2004. Cross-layer perceptual ARQ for H.264 video streaming over 802.11 wireless networks. In *Proceedings of IEEE GLOBECOM '04*. Vol. 5.

CAVENDISH, D., GERLA, M., AND MASCOLO, S. 2004. A control theoretical approach to congestion control in packet networks. *IEEE/ACM Trans. Netw. 12*, 893–906.

CHEN, C.-L. 2007. IEEE 802.11e EDCA QoS provisioning with dynamic fuzzy control and cross-layer interface. In *Proceedings of ICCCN*.

DUFFY, K., MALONE, D., AND LEITH, D. 2005. Modeling the 802.11 distributed coordination function in non-saturated conditions. *IEEE Comm. Letters 9,* 8.

FOH, C., ZHANG, Y., NI, Z., CAI, J., AND NGAN, K. 2007a. Optimized cross-layer design for scalable video transmission over the IEEE 802.11e networks. *IEEE Trans. Circ. Syst. Video Tech. 17,* 12.

FOH, C., ZUKERMAN, M., AND TANTRA, J. 2007b. A markovian framework for performance evaluation of IEEE 802.11. *IEEE Trans. Wire. Comm. 6,* 4, 1276–1265.

FRANKLIN, G., POWELL, D., AND WORKMAN, M. 1997. *Digital Control of Dynamic Systems*. Prentice-Hall.

FREITAG, J., DA FONSECA, N. L. S., AND DE REZENDE, J. F. 2006. Tuning of 802.11e Network Parameters. *IEEE Comm. Letters 10,* 8, 611–613.

GRIECO, L., BOGGIA, G., MASCOLO, S., AND CAMARDA, P. 2003. A control theoretic approach for supporting quality of service in IEEE 802.11e WLANs with HCF. In *Proceedings of the 42nd IEEE Conference on Decision and Control*. Vol. 2, 1586–1591.

HE, W., NAHRSTEDT, K., AND LIU, X. 2008. End-to-end delay control of multimedia applications over multihop wireless links. *ACM Trans. Multimedia Comput. Commun. Appl. 5*, 16:1–16:20.

HEYMAN, D. P. AND SOBEL, M. J. 2004. *Stochastic Models in Operations Research, Vol. I: Stochastic Processes and Operating Characteristics*. Dover Publications.

HOLLOT, C. V., MISRA, V., TOWSLEY, D., AND GONG, W. B. 2001. A control Theoretic Analysis of RED. In *Proceedings of IEEE INFOCOM*. 1510–1519.

IEEE 802.11. 2007. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Revision of IEEE Std 802.11-1999.

IEEE 802.11B. 1999. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher-speed physical layer extension in the 2.4 GHz band.

IEEE 802.11E. 2005. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: medium access control (MAC) enhancements for quality of service. Supplement to IEEE 802.11 Standard.

IEEE 802.11G. 2003. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 4: further higher data rate extension in the 2.4 GHz Band.

IEEE 802.11TGaa. 2010. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: amendment for robust streaming of audio video transport streams, draft 0.05.

ITU-T. 2001. Recommendation G.1010: End-user multimedia QoS categories.

ITU-T. 2007. Recommendation G.1070: Opinion model for video-telephony applications.

Kleinrock, L. 1975. Queuing Systems, Vol. 1: Theory. Wiley-Interscience.

Ksentini, A., Naimi, M., and Gueroui, A. 2006. Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture. *IEEE Comm. Mag. 44,* 1, 107–114.

Lambert, P., De Neve, W., De Neve, P., Moerman, I., Demeester, P., and Van de Walle, R. 2006. Rate-distortion performance of H.264/AVC compared to state-of-the-art video codecs. *IEEE Trans. Circ. Syst. Video Tech. 16,* 1, 134–140.

Malone, D., Duffy, K., and Leith, D. 2007. Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions. *IEEE/ACM Trans. Netw. 15,* 1.

Nafaa, A. and Ksentini, A. 2008. On Sustained QoS guarantees in operated IEEE 802.11 wireless LANs. *IEEE Trans. Paral. Distribut. Syst. 19,* 8, 1020–1033.

Patras, P., Banchs, A., and Serrano, P. 2009. A control theoretic approach for throughput optimization in IEEE 802.11e EDCA WLANs. *Mobile Net. Appl. 14,* 6, 697–708.

Patras, P., Banchs, A., Serrano, P., and Azcorra, A. 2010. A control theoretic approach to distributed optimal configuration of 802.11 WLANs. *IEEE Trans. Mobile Comput. 99,* PrePrints.

Serrano, P., Banchs, A., and Kukielka, J. 2007. Optimal configuration of 802.11e EDCA under voice traffic. In *Proceeding of the Global Telecommunications Conference. (GLOBECOM '07).* IEEE. 5107–5111.

Xiao, Y., Li, F. H., and Li, B. 2007. Bandwidth sharing schemes for multimedia traffic in the IEEE 802.11e contention-based WLANs. *IEEE Trans. Mobile Comput. 6,* 7, 815–831.

Xiao, Y., Li, H., and Choi, S. 2004. Protection and guarantee for voice and video traffic in IEEE 802.11e wireless LANs. In *Proceedings of IEEE INFOCOM*. Vol. 3, 2152–2162.

Yang, Y., Haverkort, B. R., and Heijenk, G. J. 2007. A centralized feedback control model for resource management in wireless networks. In *Proceedings of the 8th International Workshop on Performance Modeling of Computer and Communications Systems*.

Zhang, Y., Foh, C., and Cai, J. 2008. An on-off queue control mechanism for scalable video streaming over the IEEE 802.11e WLAN. In *Proceedings of ICC '08.* 4958–4962.

Received June 2010; revised November 2010 and April 2011; accepted May 2011