

# Providing Service Guarantees in 802.11e EDCA WLANs with Legacy Stations

Albert Banchs, *Member, IEEE*, Pablo Serrano, *Member, IEEE*, and Luca Vollero, *Member, IEEE*

**Abstract**—Although the EDCA access mechanism of the 802.11e standard supports legacy DCF stations, the presence of DCF stations in the WLAN jeopardizes the provisioning of the service guarantees committed to the EDCA stations. The reason is that DCF stations compete with Contention Windows (*CW*s) that are predefined and cannot be modified, and as a result, the impact of the DCF stations on the service received by the EDCA stations cannot be controlled. In this paper, we address the problem of providing throughput guarantees to EDCA stations in a WLAN in which EDCA and DCF stations coexist. To this aim, we propose a technique that, implemented at the Access Point (AP), mitigates the impact of DCF stations on EDCA by skipping with a certain probability the Ack reply to a frame from a DCF station. When missing the Ack, the DCF station increases its *CW*, and thus, our technique allows us to have some control over the *CW*s of the legacy DCF stations. In our approach, the probability of skipping an Ack frame is dynamically adjusted by means of an adaptive algorithm. This algorithm is based on a widely used controller from classical control theory, namely a Proportional Controller. In order to find an adequate configuration of the controller, we conduct a control-theoretic analysis of the system. Simulation results show that the proposed approach is effective in providing throughput guarantees to EDCA stations in presence of DCF stations.

**Index Terms**—WLAN, 802.11, 802.11e, EDCA, DCF, ACKS, legacy stations, throughput guarantees, control theory.

## 1 INTRODUCTION

THE Wireless LAN (WLAN) technology is nowadays widely used for the Internet access. One of the shortcomings of traditional WLANs, based on the 802.11 standard [1], is that they provide no means to offer service guarantees to users. This is a significant drawback, in particular, due to the inherent resource limitation in radio systems. This shortcoming has been identified by the research community, who has devoted considerable effort over the last decade to the design of wireless local area networks (WLANs) with Quality of Service (QoS) support. Along this effort, the Enhancements Task Group (TGe) was formed under the IEEE 802.11 WG to recommend an international WLAN standard with QoS support. This standard is called 802.11e [2] and will be included in the ongoing new revision of the 802.11 standard [3].

The 802.11e standard defines two different access mechanisms: the *Enhanced Distributed Channel Access* (EDCA) and the *HCF Controlled Channel Access* (HCCA). This paper focuses on the former. The EDCA mechanism of 802.11e was designed as an extension of the *Distributed Coordination Function* (DCF) mechanism of the legacy 802.11

standard. One of the key design goals of the EDCA mechanism was the backward compatibility with the legacy DCF mechanism. Following this goal, EDCA was designed such that legacy stations using DCF could operate in an 802.11e WLAN under EDCA.

One of the main problems of the EDCA mechanism is that, although legacy DCF stations can interoperate in a WLAN under EDCA, they substantially degrade the performance of the WLAN and preclude the provisioning of service guarantees to the EDCA stations. Indeed, as we have noted in [4], [5], the fact that DCF (in contrast to EDCA) competes with predefined contention parameters that cannot be modified prevents controlling the aggressiveness of DCF stations. As a result, if EDCA stations competing against aggressive DCF stations are to receive service guarantees, they will need to behave aggressively as well, and this will severely degrade the overall WLAN performance.

Some effort in the literature has been devoted to the analysis of WLANs in which EDCA and DCF stations coexist (see, e.g., [6], [7], [8], [9]). Additionally, a number of proposals have been made to improve the performance of EDCA in presence of DCF stations, namely [10], [11], [12] in addition to our previous works [4], [5].<sup>1</sup> The main drawback of [10], [11], [12] is that they require introducing modifications into the DCF or the EDCA stations. In contrast, our proposal of [4], [5] leaves the EDCA and DCF stations untouched, which represents a major advantage from a deployment perspective.

Following our previous works [4], [5], in this paper, we address the problem of providing throughput guarantees to EDCA stations in a WLAN with legacy DCF stations. To tackle this, we propose the *Dynamic ACK Skipping* (DACKS)

• A. Banchs is with the Departamento de Ingenieria Telematica, Universidad Carlos III de Madrid, Avda de la Universidad, 30, E-28911 Leganés (Madrid), Spain, and also with IMDEA Networks. E-mail: banchs@it.uc3m.es.

• P. Serrano is with the Departamento de Ingenieria Telematica, Universidad Carlos III de Madrid, Avda de la Universidad, 30, E-28911 Leganés (Madrid), Spain. E-mail: pablo@it.uc3m.es.

• L. Vollero is with the CIR—Centro Integrato di Ricerca, Università Campus Bio-Medico di Roma, Via Alvaro del Portillo, 21, 00128 Roma, Italy. E-mail: l.vollero@unicampus.it.

Manuscript received 21 May 2009; revised 6 Nov. 2009; accepted 3 Dec. 2009; published online 16 Mar. 2010.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-05-0185. Digital Object Identifier no. 10.1109/TMC.2010.52.

1. In [13], the authors used a similar idea to that of [4], [5] for a different purpose, namely to provide service differentiation in a WLAN with DCF stations only.

technique, which mitigates the impact of legacy stations on an 802.11e WLAN under the EDCA mechanism by implementing a small modification in the 802.11e Access Point (AP). The main contributions of the paper are summarized as follows:

- We propose the DACKS technique. The key feature of the approach (as compared to our previous works [4], [5]) is that the system is dynamically controlled based on the observed behavior of the WLAN. In particular, DACKS is based on a common controller from control theory.
- We develop a model of a WLAN with DACKS under stationary conditions. Based on this model, we determine the optimal configuration of the EDCA parameters in order to provide EDCA stations with throughput guarantees.
- We develop a model for the transient response of a WLAN controlled by DACKS. With this model, we analyze the dynamics of our system from a control-theoretic standpoint to tune the DACKS parameters.

A longer version of this paper, containing proofs of the theoretical results as well as additional simulation results, is available in [14].

## 2 802.11 DCF AND 802.11E EDCA

DCF and EDCA execute a similar algorithm to transmit their frames. In the following, we first present the 802.11e EDCA mechanism and then we describe the differences between 802.11e EDCA and 802.11 DCF.

EDCA regulates the access to the wireless channel on the basis of the *channel access functions* (CAFs). A station may run up to 4 CAFs, and each of the frames generated by the station is mapped to one of these CAFs. Then, each CAF executes an independent backoff process to transmit its frames. A CAF  $i$  with a new frame to transmit monitors the channel activity. If the channel is idle for a period of time equal to the arbitration interframe space parameter ( $AIFS_i$ ), the CAF transmits. Otherwise, if the channel is sensed busy (either immediately or during the  $AIFS_i$  period), the CAF starts a backoff process. The arbitration interframe space ( $AIFS_i$ ) takes a value of the form  $DIFS + nT_e$ , where  $DIFS$  is the DCF interframe space,  $T_e$  is the duration of an empty slot time, and  $n$  is a nonnegative integer.

Upon starting the backoff process, the CAF computes a random integer value uniformly distributed in the range  $(0, CW_i - 1)$  and initializes its backoff time counter with this value. The  $CW_i$  value is called the contention window and depends on the number of transmissions failed for the frame. At the first transmission attempt,  $CW_i$  is set equal to the minimum contention window parameter ( $CW_i^{min}$ ). As long as the channel is sensed idle, the backoff time counter is decremented once every time interval  $T_e$ , and “frozen” when a transmission is detected on the channel.

When the backoff time counter reaches zero, the CAF transmits. A collision occurs when two or more CAFs start transmission simultaneously. An acknowledgment (Ack) frame is used to notify the transmitting CAF that the frame has been successfully received. The Ack is immediately transmitted at the end of the frame, after a period of time

equal to the SIFS (the short interframe space). If the Ack is not received within a time-out given by the  $Ack\_Timeout$ , the CAF assumes that the frame was not received successfully and reschedules the transmission by reentering the backoff process. The CAF then doubles  $CW_i$  (up to a maximum value given by the  $CW_i^{max}$  parameter), computes a new backoff time, and starts decrementing the backoff time counter at an  $AIFS_i$  time following the time-out expiry. If the number of failed attempts reaches a pre-determined retry limit  $R$ , the frame is discarded.

After a (successful or unsuccessful) frame transmission, before transmitting the next frame, the CAF must execute a new backoff process. As an exception to this rule, the protocol allows the continuation of an EDCA transmission opportunity (TXOP). A continuation of an EDCA TXOP occurs when a CAF retains the right to access the channel following the completion of a transmission and transmits several frames back-to-back. The period of time a CAF is allowed to retain the right to access the channel is limited by the transmission opportunity limit parameter ( $TXOP\_Limit_i$ ).

In the case of a single station running more than one CAF, if the backoff time counters of two or more CAFs of the station reach zero at the same time, a scheduler inside the station avoids the *internal collision* by granting the access to the channel to the highest priority CAF. The other CAFs of the station involved in the internal collision react as if there had been a collision on the channel, doubling their  $CW_i$  and restarting the backoff process.

As it can be seen from the description of EDCA given in this section, the behavior of a CAF depends on a number of parameters, namely  $CW_i^{min}$ ,  $CW_i^{max}$ ,  $AIFS_i$ , and  $TXOP\_Limit_i$ . These are configurable parameters that can be set to different values for different CAFs. The standard draft groups CAFs by Access Categories (ACs), all the CAFs of an AC having the same configuration, and limits the maximum number of ACs in the WLAN to 4. An EDCA station that wants to enter the WLAN must issue a signaling request indicating the AC that it wants to join. If admitted, the EDCA station can join the WLAN with a CAF configured according to the parameters of the corresponding AC. The parameters of each AC are announced periodically by means of beacon frames.

A DCF station executes a very similar backoff process to the one described above for an EDCA CAF, albeit with some differences. One difference is the way the backoff counter is managed. In EDCA, the backoff counter is resumed one slot time before the AIFS expiration, while in DCF, it is resumed after the expiration. Moreover, in DCF, a station transmits immediately when the counter decrements to 0, while in EDCA, it transmits in the next slot time.<sup>2</sup>

Another key difference between DCF and EDCA is that, while in 802.11e EDCA, the contention parameters are configurable and can be set to different values for different ACs, in DCF, the values of these parameters are fixed by the standard as follows:

- The  $AIFS_i$  parameter in DCF is set equal to  $DIFS$ .
- The configuration of the  $CW_i^{min}$  and  $CW_i^{max}$  parameters is predefined by the 802.11 DCF standard.

2. The reader is referred to [7] for further details about the backoff behavior of EDCA and DCF.

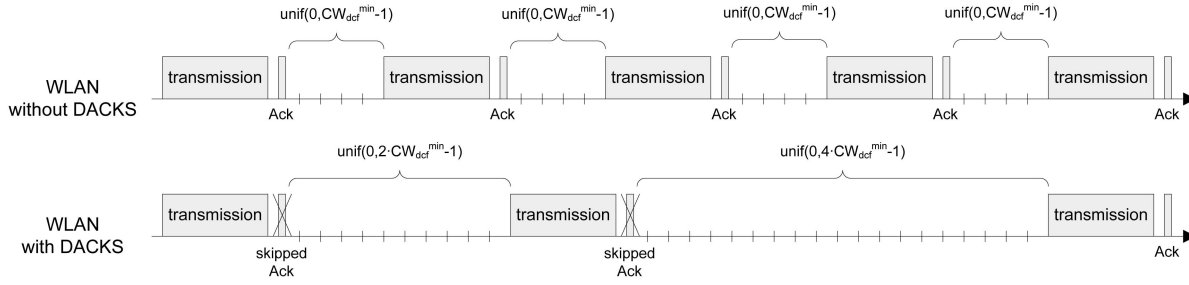


Fig. 1. DACKS technique.

We refer to the values given by the standard as  $CW_{dcf}^{min}$  and  $CW_{dcf}^{max}$ , respectively.

- Upon accessing the medium, DCF stations transmit a single packet, and hence, do not use the  $TXOP_{limit_i}$  parameter.

While EDCA has been designed to allow coexistence with legacy DCF stations, the fact that the contention parameters with which DCF stations compete are fixed jeopardizes the provisioning of service guarantees to EDCA stations. The rest of the paper is devoted to overcoming this limitation.

### 3 DACKS TECHNIQUE

As we have seen in the previous section, legacy DCF stations start the backoff process with a  $CW$  equal to  $CW_{dcf}^{min}$ . This initial  $CW$  is fixed by the standard to a small value, and it only doubles after each failed attempt. These small  $CW$  values of DCF stations raise problems in a WLAN in which EDCA stations are to receive service guarantees. Indeed, no matter whether the  $CW$ s of the EDCA stations are configured with small or large values, the following drawbacks are observed when there are a nonsmall number of stations in the WLAN:

1. If EDCA stations were configured with small  $CW$  values in order to give them a higher priority than DCF stations, we would have both DCF and EDCA stations with small  $CW$ s and the resulting overall efficiency of the WLAN will be low, due to the fact that small  $CW$  values result in a high collision rate.
2. If EDCA stations were configured with large  $CW$  values in order to avoid the above problem, DCF stations would compete with smaller  $CW$ s than EDCA and would consume most of the WLAN resources, leaving EDCA stations with little resources, and thus, failing to meet their service guarantees.

None of the above two alternatives is desirable, as in both cases the service received by the EDCA stations is seriously degraded as a consequence of the impact of legacy stations. Instead, it would be desirable to increase the  $CW$  of legacy stations; in this way, EDCA stations could receive service guarantees without compromising the overall efficiency. The DACKS technique achieves this goal without modifying the legacy DCF stations.

DACKS is based on the following behavior of DCF: after sending a packet, a DCF station waits for an Ack frame, and if the frame is not received within an Ack time-out, it assumes a collision and increases its  $CW$ . The central idea is then the following: if the AP skips the Ack reply to legacy

DCF stations with a certain probability (hereafter referred to as  $P_{skip}$ ), these stations will “see” a collision rate higher than the actual one, and will contend with larger  $CW$ s, resulting this in a smaller impact on the EDCA stations.

The above behavior of DACKS is illustrated in Fig. 1. In the figure, the behavior of a DCF station in a WLAN without DACKS is compared against the behavior of a DCF station in a WLAN that uses the DACKS technique. It can be observed that in the latter case, by skipping the Ack reply with some probability, DACKS achieves the objective of increasing the average  $CW$  with which the DCF station contends for channel access, and hence, reduces the number of times that the DCF station transmits.

The challenge with the DACKS technique is the configuration of the probability  $P_{skip}$ . This adds to the inherent difficulty in 802.11e of configuring the EDCA contention parameters in order to provide the desired behavior. In [4], [5] we proposed some algorithms to compute  $P_{skip}$  statically. The main drawbacks of a static configuration are the following:

- A static configuration has to compute the configuration assuming the worst case in which all DCF stations are constantly active. This requires a much more aggressive behavior than needed against DCF stations. In particular, when all DCF stations are active, Ack frames need to be skipped with a high probability to ensure the desired throughput guarantees for EDCA. In contrast, if some DCF stations are not active, a smaller skipping probability is enough to provide EDCA stations with the desired service.
- Similarly to the above, a static configuration has to assume that all admitted EDCA stations are active, since this is the worst case to ensure the desired guarantees. This assumption forces a high probability of skipping Ack frames, degrading, thus, DCF performance. In the case some EDCA stations are not active, the desired service could be provided while reducing the degradation suffered by DCF.

We conclude from the above that a static configuration degrades the performance of DCF stations unnecessarily when not all the (EDCA and DCF) stations are active. In this paper, we propose an alternative scheme that, by dynamically adjusting the skipping probability to the current behavior of the WLAN, minimizes the disruption suffered by the DCF stations.

### 4 EDCA CONFIGURATION

It follows from the above explanations that a major challenge for an EDCA WLAN with DACKS is the configuration of both

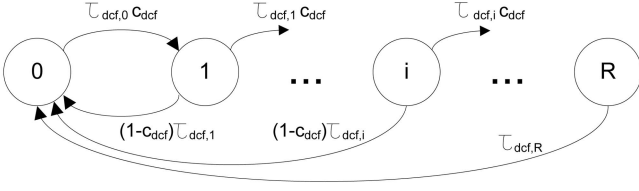


Fig. 2. Markov chain model of a DCF station.

the EDCA parameters and the DACKS skipping probability. In this section, we analyze the EDCA configuration,<sup>3</sup> while the DACKS configuration is analyzed in the next section.

#### 4.1 Scenario and Assumptions

In the following, we describe the scenario considered in this paper as well as the assumptions upon which our analysis is based:

- Our scenario consists of a WLAN, where EDCA and DCF stations coexist. Our goal is to provide EDCA stations in this scenario with throughput guarantees.
- We consider that each EDCA station executes only one CAF and joins a given AC  $i$  depending on its throughput requirements. We denote by  $R_i$  the throughput guarantee given to the EDCA stations of AC  $i$ .
- We assume that, over a time period, a station is either constantly backlogged<sup>4</sup> or does not transmit any traffic. We refer to the former as an *active* station and the latter as *inactive*.
- We denote by  $N_{def}$  the number of active DCF stations in the WLAN and by  $N_i$  the number of active EDCA stations that belong to AC  $i$ .
- Following our previous results of [16], we use the following configuration for the EDCA stations:  $AIFS_i = DIFS$  and  $CW_i^{min} = CW_i^{max}$ , since [16] shows (both analytically and via simulation) that no other configuration provides better throughput performance. We denote  $CW_i = CW_i^{min} = CW_i^{max}$ .
- Following [17], we assume that backoff times are geometrically distributed, i.e., a station at a given backoff stage transmits with a constant and independent probability in each slot time.
- Upon accessing the channel, both EDCA and DCF stations transmit a single packet of length  $l$ .

#### 4.2 DCF Station Model

We start our analysis by computing the probability that a DCF station transmits at a randomly chosen slot time,  $\tau_{def}$ , as a function of the probability that a transmission attempt of a DCF station collides,  $c_{def}$ .

Fig. 2 illustrates our model of a DCF station. The states represent the backoff stage of the station, i.e., the number of collisions suffered by the current frame. At state 0, the station's  $CW$  is equal to  $CW_{def}^{min}$ , yielding the following transmission probability [15]:

3. In the EDCA configuration proposed here, we focus only on EDCA stations with throughput guarantees. In [14], we extend the configuration to *Best-Effort* EDCA stations.

4. [15] refers to constantly backlogged stations as *saturated*. In the rest of the paper, we use the terms "constantly backlogged" and "saturated" indistinctly.

$$\tau_{def,0} = \frac{2}{CW_{def}^{min} + 1}. \quad (1)$$

Let  $m$  be the maximum backoff stage defined by  $CW_{def}^{max} = 2^m CW_{def}^{min}$ . Note that in DCF, we have  $m < R$  [1]. At state  $i \leq m$ , the  $CW$  has been doubled  $i$  times, yielding the following transmission probability:

$$\tau_{def,i} = \frac{2}{2^i CW_{def}^{min} + 1}. \quad (2)$$

At state  $i > m$ , the  $CW$  has already reached  $CW_{def}^{max}$ , yielding

$$\tau_{def,i} = \frac{2}{2^m CW_{def}^{min} + 1}. \quad (3)$$

In the rest of the paper, we use the following simplifying approximation for  $\tau_{def,i}$ :

$$\tau_{def,i} \approx \frac{2}{2^{\min(i,m)} (CW_{def}^{min} + 1)} = \frac{\tau_{def,0}}{2^{\min(i,m)}}. \quad (4)$$

Following the above, we have that at state  $i$ , the station transmits in each slot time with probability  $\tau_{def,i}$ . If the transmission collides (which occurs with probability  $c_{def}$ ), the station moves to the next state, and doubles its  $CW$  if  $i < m$ . If it succeeds, the station goes back to the initial state 0 and sets the  $CW$  equal to  $CW_{def}^{min}$ . When the station reaches the maximum retry limit at state  $R$ , it moves back to state 0 no matter if the transmission succeeds or collides. This leads to the state transition probabilities given in Fig. 2.

Let us denote by  $P_i$  the probability that the station is at state  $i$ . The probability of entering state  $i$  is equal to the probability of being at state  $i-1$  and performing a failed transmission. The probability of leaving this state is equal to the probability of performing a (failed or successful) transmission. By forcing equilibrium between these two probabilities, we have

$$P_{i-1} \tau_{def,i-1} c_{def} = P_i \tau_{def,i}. \quad (5)$$

Following (4), we have

$$\tau_{def,i} = \begin{cases} \tau_{def,i-1}/2, & i \leq m \\ \tau_{def,i-1}, & i > m, \end{cases} \quad (6)$$

which yields

$$P_i = \begin{cases} P_{i-1} 2c_{def}, & i \leq m \\ P_{i-1} c_{def}, & i > m. \end{cases} \quad (7)$$

Applying the above recursively leads to

$$P_i = \begin{cases} P_0 (2c_{def})^i, & i \leq m \\ P_0 2^m c_{def}^i, & i > m. \end{cases} \quad (8)$$

By forcing  $\sum_{i=0}^{\infty} P_i = 1$  and isolating  $P_0$ , we obtain

$$\begin{aligned} P_0 &= \frac{1}{\sum_{i=0}^m (2c_{def})^i + \sum_{i=m+1}^R 2^m c_{def}^i} \\ &= \frac{1}{\frac{1-(2c_{def})^{m+1}}{1-2c_{def}} + \frac{2^m c_{def}^m (1-c_{def}^{R-m})}{1-c_{def}}}. \end{aligned} \quad (9)$$

With the above, we can compute the transmission probability of a DCF station as follows:

$$\begin{aligned}\tau_{dcf} &= \sum_{i=0}^R P_i \tau_{dcf,i} \\ &= \frac{(1 - 2c_{dcf})(1 - c_{dcf}^{m+1}) +}{(1 - c_{dcf})(1 - (2c_{dcf})^{m+1}) +} \\ &\quad \frac{(1 - 2c_{dcf})c_{dcf}^{m+1}(1 - c_{dcf}^{R-m})}{(1 - 2c_{dcf})2^m c_{dcf}^{m+1}(1 - c_{dcf}^{R-m})} \tau_{dcf,0},\end{aligned}\quad (10)$$

which terminates our model of a DCF station.

### 4.3 Throughput Analysis

Based on the model of a DCF station presented above, we now analyze the throughput performance of DCF and EDCA stations in the WLAN. Our analysis is based on the following: 1) after each transmission, there is a slot time in which DCF stations have not yet decremented their backoff counter and only EDCA stations may transmit, 2) we assume that EDCA and DCF stations transmit with a constant and independent probability in those slot times where they are allowed, and 3) when computing their transmission probabilities, we account for the fact that EDCA stations wait for one extra slot time after the backoff counter reaches 0.

Equation (10) gives the transmission probability of a DCF station as a function of the collision probability. The transmission probability of the EDCA stations, whose configuration satisfies  $CW_i = CW_i^{min} = CW_i^{max}$ , can be easily computed as follows:<sup>5</sup>

$$\tau_i = \frac{2}{CW_i + 3}. \quad (11)$$

Further, the collision probability of the DCF stations can be expressed as<sup>6</sup>

$$c_{dcf} = 1 - P_{ack}(1 - \tau_{dcf})^{N_{dcf}-1} \left( \prod_i (1 - \tau_i)^{N_i} \right), \quad (12)$$

where  $P_{ack}$  is the probability that, upon successfully receiving a packet from a DCF station, the AP sends the corresponding Ack—i.e., the probability that the DACKS technique does not skip this Ack:

$$P_{ack} = 1 - P_{skip}. \quad (13)$$

With the above, we can compute the transmission probability of all the stations of the WLAN as follows:

- The transmission probability of the EDCA stations,  $\tau_i$ , can be computed from their configured  $CW_i$  with (11).
- Given all  $\tau_i$ s, we can compute  $\tau_{dcf}$  by solving the nonlinear equation formed by (10) and (12).<sup>7</sup>

Once all the transmission probabilities have been obtained, we can compute the probability  $P_t$  that a given slot time contains a transmission (either a success or a collision) as follows: If the previous slot time was empty, all stations may transmit, otherwise, only EDCA stations may transmit. Thus,

$$\begin{aligned}1 - P_t &= (1 - P_t)(1 - \tau_{dcf})^{N_{dcf}} \prod_i (1 - \tau_i)^{N_i} \\ &\quad + P_t \prod_i (1 - \tau_i)^{N_i},\end{aligned}\quad (14)$$

which yields

$$P_t = \frac{1 - (1 - \tau_{dcf})^{N_{dcf}} \prod_i (1 - \tau_i)^{N_i}}{1 + \prod_i (1 - \tau_i)^{N_i} - (1 - \tau_{dcf})^{N_{dcf}} \prod_i (1 - \tau_i)^{N_i}}. \quad (15)$$

With the above, we can proceed to compute the throughput experienced by an EDCA station of AC  $i$ ,  $r_i$ , and the throughput experienced by a DCF station,  $r_{dcf}$ , as follows:

$$r_i = \frac{\tau_i c_i l}{(1 - P_t)T_e + P_t T_t} \quad (16)$$

and

$$r_{dcf} = \frac{P_{ack} \tau_{dcf} (1 - \tau_{dcf})^{N_{dcf}-1} \left( \prod_{j \neq i} (1 - \tau_j)^{N_j} \right) l}{(1 - P_t)T_e + P_t T_t}, \quad (17)$$

where  $T_e$  is the duration of an empty slot time,  $T_t$  is the duration of a slot time with a transmission, and  $c_i$  is the probability that a transmission attempt of an EDCA station of AC  $i$  collides,

$$\begin{aligned}c_i &= (1 - P_t)(1 - \tau_i)^{N_i-1} (1 - \tau_{dcf})^{N_{dcf}} \prod_{j \neq i} (1 - \tau_j)^{N_j} \\ &\quad + P_t (1 - \tau_i)^{N_i-1} \prod_{j \neq i} (1 - \tau_j)^{N_j}.\end{aligned}\quad (18)$$

The duration of an empty slot time ( $T_e$ ) is fixed by the standard, while the duration of a slot time that contains a success and a collision is equal to, respectively:<sup>8</sup>

$$T_s = T_{PLCP} + \frac{H}{C} + \frac{l}{C} + SIFS + T_{PLCP} + \frac{ACK}{C} + DIFS, \quad (19)$$

$$T_c = T_{PLCP} + \frac{H}{C} + \frac{l}{C} + EIFS, \quad (20)$$

where  $T_{PLCP}$  is the Physical Layer Convergence Protocol (PLCP) preamble and header transmission time,  $H$  is the MAC overhead (header and FCS),  $ACK$  is the size of the acknowledgment frame, and  $C$  is the channel bit rate.

Since the standard fixes the value of  $EIFS$  equal to the time required to send an Ack, we have that the duration of a collision and a success are equal, and we can thus compute the duration of a slot time with a transmission as

$$T_t = T_s = T_c. \quad (21)$$

With the above, we can compute, given the configuration of the  $CW_i$  and  $P_{ack}$  parameters, the throughput of each of the DCF and EDCA stations in the WLAN, which terminates the throughput analysis. In the following sections, we address the configuration of these parameters.

### 4.4 $CW_i$ Configuration

We now address the issue of calculating the optimal configuration of the WLAN. The goal of the optimal

5. Note that (11) differs from (1) as it takes into account that an EDCA station transmits at the next slot time when its backoff counter reaches 0.

6. Note that, by collision, here, we understand both the case when the transmission actually collides and the case when, even if there is not a real collision, the Ack is omitted by the DACKS technique.

7. The reader is referred to [18] for a discussion on the uniqueness of the solution.

8. Note that, in case of a skipped Ack, the slot time duration is given by  $T_s$ , since stations update their NAV to the duration of a successful transmission, and defer channel access during this time.

configuration is to provide the desired throughput guarantees while maximizing the overall throughput performance.

Upon changing the  $CW_i$  configuration, the AP needs to distribute the new configuration to the stations by means of signaling. This signaling limits the frequency with which the  $CW_i$ s values can be updated. In contrast to the  $CW_i$ s, the  $P_{ack}$  parameter is local and its value need not be sent to the stations. As a result,  $P_{ack}$  can be updated as frequently as needed with no associated signaling cost. Following this, in this paper, we make the following choices:

- The  $CW_i$  parameters are statically set based on information that does not change frequently, and therefore, does not trigger frequent updates of their values.
- The  $P_{ack}$  parameter is configured based on a dynamic algorithm that constantly updates its value following the observed behavior of the WLAN.

In the remaining of this section, we address the configuration of the  $CW_i$  parameters, while the dynamic algorithm that updates  $P_{ack}$  is presented in the next section.

Following the above argumentation, the computation of the  $CW_i$  configuration needs to be based on data that do not change frequently. In particular, we use the following data:

- The number of EDCA stations admitted in the WLAN and their required throughputs. These data are available at the AP since EDCA stations, prior to entering the WLAN, have to issue an admission control request with this information.
- The number of DCF stations present in the WLAN. This information is available as DCF stations need to go through an authentication/association process before they enter the WLAN.

In contrast to the above data,  $P_{ack}$  is constantly updated, and therefore, cannot be taken into account in the computation of the  $CW_i$ s. This raises an issue since the optimal  $CW_i$  configuration actually depends on the setting of this parameter. In order to overcome this problem, the approach that we take in this paper is to compute the configuration of the  $CW_i$ s considering that  $P_{ack}$  is set to 0. This suboptimal solution has the following advantages:

- The first advantage is that the solution becomes optimal when the WLAN is stressed with many throughput guarantee requests from the EDCA stations. This is due to the fact that, when the WLAN is stressed, the DACKS technique forces DCF stations to reduce drastically their transmission rate by setting  $P_{ack} = 0$ , thereby making the computed  $CW_i$  configuration optimal.
- The other advantage of the proposed configuration is that it allows maximizing the number of throughput guarantee requests that can be admitted. Indeed, if a request cannot be admitted when  $P_{ack}$  is set to 0, this means that the request can never be admitted.

To compute the optimal configuration, we start by imposing the following condition, which ensures that the throughput will be distributed among stations proportionally to their requests [19]:

$$\frac{\tau_i(1 - \tau_j)}{\tau_j(1 - \tau_i)} = \frac{R_i}{R_j}, \quad (22)$$

where  $R_i$  is the throughput guarantee of AC  $i$ .

We note that, with the above equation, if we assume that the value of a given  $CW_i$  is known, we can compute the value of all the other  $CW_i$ s. From the throughput analysis of Section 4.3 and taking  $P_{ack} = 0$ , we can then compute all the throughputs.

With the above, we proceed as follows to find the optimal  $CW_i$  configuration: We conduct a numerical search using the *golden section search* method over the  $CW_i$  of the AC with the lowest throughput guarantee (without loss of generality, we assume it is AC 1). For each  $CW_1$  value evaluated in the search, we compute the other  $CW_i$ s from (22), and from these, we compute  $r_1$ . With the numerical search, we find thus the  $CW_1$  value that leads to the largest  $r_1$ . In order to avoid a large degree of unfairness with DCF, we impose in the search that  $CW_1$  cannot be smaller than  $CW_{def}^{min}$ . Once the search finds  $CW_1$ , we then compute all the other  $CW_i$ s, which terminates the algorithm.

Note that a requirement that must be met by the  $CW_i$  configuration given by the above algorithm is that the resulting  $r_i$ s are larger than the corresponding  $R_i$ s. If this condition is not satisfied, this means that there exists no set of  $CW_i$  values that meets the desired throughput guarantees even when  $P_{ack}$  is set to 0. In this case, the requested guarantees cannot be satisfied and the request that triggered this computation must, therefore, be rejected.<sup>9</sup>

Note also that the above computation has been performed considering that  $P_{ack}$  is set to minimally disrupt EDCA stations (specifically,  $P_{ack} = 0$ ). As a consequence, if  $P_{ack}$  is not properly adjusted and takes larger values, there is the risk that the throughput guarantees are not met. In the following section, we present an algorithm that adjusts  $P_{ack}$  to ensure that the committed throughput guarantees are never harmed.

## 5 DACKS CONFIGURATION

In this section, we present an algorithm that updates the  $P_{ack}$  parameter dynamically. We start by analyzing the conditions that must be met by the setting of  $P_{ack}$ . Next, we propose a system based on control theory that, following these conditions, dynamically adjusts  $P_{ack}$ . In order to analyze the overall controlled system, we develop a linearized model of the system. Based on this linearized model, we conduct a stability analysis to determine the region of the system parameters that guarantees a stable behavior. Finally, we obtain the setting of the parameters of the controlled system within the stability region.

### 5.1 $P_{ack}$ Configuration

Our goals for the setting of the  $P_{ack}$  parameter are the following ones:

- Given the  $CW_i$  configuration obtained in the previous section, we want to ensure that backlogged EDCA stations see their throughput guarantees satisfied.
- As long as the throughput guarantees for EDCA stations are met, we want to minimize the throughput degradation of the DCF stations by setting  $P_{ack}$  as large as possible.

9. Note that this request can come either from an EDCA or a DCF station. In the latter case, the AP can reject the request by not completing the association process initiated by the station. Note that many of today's APs already apply similar policies to deny association of stations based, e.g., on their MAC address or on the AP's current load.

Following the above, the main goal for the dynamic algorithm that computes  $P_{ack}$  is to set it to the largest possible value that satisfies the throughput requirements of the EDCA stations. We build the algorithm around the probability  $P_t$  that a randomly chosen slot time contains a transmission. Note that (16) can be rewritten as a function of  $P_t$ :

$$r_i = \frac{\tau_i(1 - P_t)l}{(1 - \tau_i)((1 - P_t)T_e + P_tT_r)}. \quad (23)$$

Our algorithm is based on the following two observations:

- Given the  $CW_i$  configuration of AC  $i$ , there exists a maximum  $P_{t,max,i}$  value such that, as long as  $P_t \leq P_{t,max,i}$ , the throughput guarantee of AC  $i$  is met. This can be seen from (23).
- The larger the  $P_t$  we allow, the smaller the probability of skipping an Ack frame needs to be. One of the goals that we have stated above was precisely to make the probability of skipping an Ack frame as small as possible, in order to minimize the disruption suffered by the DCF stations.

With the above observations, our objective can be formulated as to finding the  $P_{ack}$  configuration that yields a transmission probability equal to

$$P_{t,max} = \min_i \{P_{t,max,i}\}, \quad (24)$$

since this is the  $P_t$  value that minimizes the degradation suffered by the DCF stations while meeting the throughput guarantees of all EDCA stations.

$P_{t,max,i}$  can be obtained by imposing  $r_i \geq R_i$  and isolating  $P_t$  from (23). Given the  $P_{t,max,i}$ s, we can then compute from (24) the value of  $P_{t,max}$ . Note that this value is a constant that depends only on the  $CW_i$  configuration obtained in the previous section.

The remaining challenge is to design an adaptive algorithm that, by observing the transmission probability  $P_t$  in the channel, adjusts  $P_{ack}$  such that the channel's transmission probability is equal to  $P_{t,max}$ . Note that the key advantage of the proposed algorithm is that, by monitoring the WLAN's behavior, we can adjust the probability of skipping an Ack to the minimum value that current conditions allow, and thus, we disrupt legacy stations as little as possible. Specifically, note the following:

- With our algorithm,  $P_{ack}$  is adjusted dynamically to the behavior of the DCF stations. Indeed, as only the DCF stations currently active contribute to  $P_t$ , these are the only ones taken into account when adjusting  $P_{ack}$ .
- $P_{ack}$  is also dynamically adjusted to the behavior of the EDCA stations. Indeed, if some of the EDCA stations are not active, those do not contribute to  $P_t$ , and therefore, the setting of  $P_{ack}$  is not unnecessarily penalized because of them.

Following the above, we next design an algorithm based on control theory that adjusts  $P_{ack}$  as a function of the  $P_t$  observed in the channel with the goal of forcing that this  $P_t$  equals the target  $P_{t,max}$ .

## 5.2 DACKS Control System

Our goal is to design a control law that drives the transmission probability  $P_t$  to the desired target value  $P_{t,max}$

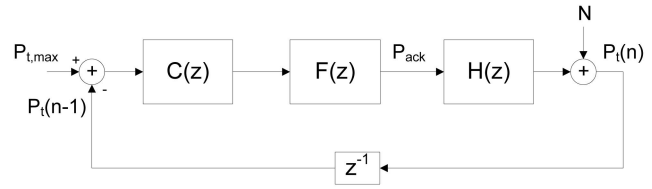


Fig. 3. Block diagram of the controlled system.

computed in (24). To this aim, we build the closed-loop control system illustrated in Fig. 3, which consists of the following blocks:

- $H(z)$  represents the WLAN system. The system is controlled by  $P_{ack}$  and its output is the occupation of each slot time (where an output of “1” means that a slot time is occupied and “0” that it is empty). We consider that this occupation function is given by the average transmission probability of the WLAN system,  $P_t$ , added to some noise of zero mean, which we represent by  $N$ .
- $C(z)$  is the controller module. It takes the error, given by  $P_{t,max} - P_t$ , as input, and computes from this error the control signal.
- In order to eliminate the noise fed from  $N$  into the control signal, we introduce (following the design guidelines of [20]) a low-pass filter  $F(z)$  to eliminate this undesired noise. The resulting control signal free from noise is the probability of replying a frame from a DCF station with an Ack,  $P_{ack}$ .

For the transfer function of the controller  $C(z)$ , in this paper, we focus on a very simple controller from classical control theory, namely the Proportional Controller [21]:

$$C(z) = K_p. \quad (25)$$

For the low-pass filter  $F(z)$ , we use a simple exponential smoothing algorithm of parameter  $\alpha$  [22],

$$F_{out}[n] = \alpha F_{in}[n] + (1 - \alpha)F_{out}[n - 1], \quad (26)$$

where  $F_{in}$  and  $F_{out}$  are the input and output signals of the filter, respectively.

Since the output of the filter  $F(z)$  is the probability  $P_{ack}$ , we need to enforce that it stays between 0 and 1. We do this by setting

$$P_{ack}[n] = \max(0, \min(1, F_{out}[n])), \quad (27)$$

which generates the following clipping error:

$$e[n] = \max(0, \min(1, F_{out}[n])) - F_{out}[n]. \quad (28)$$

In order to eliminate this error, we follow the strategy of [23] of subtracting the error of the previous sample into the input of the following one. With this, (26) is rewritten as

$$F_{out}[n] = \alpha(F_{in}[n] - e[n - 1]) + (1 - \alpha)F_{out}[n - 1]. \quad (29)$$

In the analysis of the rest of this section, we assume that  $F_{out}$  keeps always in the range  $(0, 1)$ , and we neglect the effect of the clipping error. With this assumption,  $F(z)$  behaves as a first order filter with the following transfer function:

$$F(z) = \frac{\alpha}{1 - (1 - \alpha)z^{-1}}. \quad (30)$$

It can be seen from the above that our control system relies on two parameters, namely  $K_p$  and  $\alpha$ . The rest of this section is devoted to analyzing the system with the goal of finding an appropriate setting for these parameters.

### 5.3 Transient Analysis of 802.11

In the system illustrated in Fig. 3, we need to characterize the WLAN transfer function  $H(z)$ . To this aim, the transient response of an 802.11 WLAN system has to be studied. While 802.11 has been widely analyzed under stationary conditions (including our analysis presented in Section 4), its transient response to changing conditions has received much less attention. Indeed, although a number of papers have studied different aspects of the transient response of 802.11 [24], [25], [26], to the knowledge of the authors, ours is the first attempt to analyze the transient response of the complete 802.11 protocol under general conditions.<sup>10</sup>

In our analysis, we will assume that the number of active DCF stations and the number of active EDCA stations are constant. Note that, with this assumption, the effect of all EDCA stations can be captured with the probability that a slot time contains the transmission of at least one EDCA station. We denote this probability by  $P_{edca}$ . We further assume in what follows that stations never reach the maximum  $CW$ , which leads to the following simplified expressions for (8) and (10):<sup>11</sup>

$$P_i = P_0(2c_{def})^i, \quad (31)$$

$$\tau_{def} = \frac{(1 - 2c_{def})}{(1 - c_{def})} \tau_{def,0}. \quad (32)$$

To model the transient behavior of the WLAN, our goal is to compute the probability that a DCF station transmits at a slot time  $n$ ,  $\tau_{def}[n]$ , given the transmission probability of the DCF station in the previous slot time,  $\tau_{def}[n-1]$ , and the probability  $P_{ack}$ . Note that in stationary conditions, we will have  $\tau_{def}[n-1] = \tau_{def}[n]$ .

The key approximation upon which we base our transient analysis is the following. We assume that the relationship between the state probability  $P_i$  and the transmission probability  $\tau_{def}$  given by (31) and (32), which has been derived under stationary conditions, also holds during transients. Specifically, we assume that at a given slot time  $n-1$ , we have

$$P_i[n-1] = \left(1 - 2 \frac{\tau_{def}[n-1] - \tau_{def,0}}{\tau_{def}[n-1] - 2\tau_{def,0}}\right) \cdot \left(2 \frac{\tau_{def}[n-1] - \tau_{def,0}}{\tau_{def}[n-1] - 2\tau_{def,0}}\right)^i, \quad (33)$$

where  $\tau_{def}[n-1]$  is the transmission probability at this slot time.

Given  $P_i[n-1]$ , the state probabilities at the next slot time  $n$  can be computed as follows: if the station does not transmit at time  $n-1$ , it stays in the same state at time  $n$ ; if

it transmits successfully, it moves to state 0; if it collides, it moves to state  $i+1$ . This yields

$$P_i[n] = P_i[n-1](1 - \tau_{def,i}) + P_{i-1}[n-1]\tau_{def,i-1}c_{def}, i > 0 \quad (34)$$

and

$$P_0[n] = P_0[n-1](1 - \tau_{def,0}) + \sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i}(1 - c_{def}), \quad (35)$$

where  $c_{def}$ , the probability that a transmission at slot time  $n-1$  collides, is given by

$$1 - c_{def} = (1 - P_{edca})(1 - \tau_{def}[n-1])^{N_{def}-1}(1 - P_{ack}). \quad (36)$$

With the above, we can compute  $\tau_{def}[n]$  as follows: By definition,

$$\tau_{def}[n] = \sum_{i=0}^{\infty} P_i[n]\tau_{def,i}. \quad (37)$$

Applying (35) and (34) to  $P_i[n]$  in the above equation, we have

$$\begin{aligned} \tau_{def}[n] &= P_0[n-1](1 - \tau_{def,0})\tau_{def,0} \\ &+ \sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i}(1 - c_{def})\tau_{def,0} \\ &+ \sum_{i=1}^{\infty} P_i[n-1](1 - \tau_{def,i})\tau_{def,i} \\ &+ \sum_{i=1}^{\infty} P_{i-1}[n-1]\tau_{def,i-1}c_{def}\tau_{def,i}. \end{aligned} \quad (38)$$

Recombining the above terms and considering that  $\tau_{def,i} = \tau_{def,i-1}/2$ , we obtain

$$\begin{aligned} \tau_{def}[n] &= \sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i} - \sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i}^2 \\ &+ (1 - c_{def})\tau_{def,0} \sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i} \\ &+ \frac{c_{def}}{2} \sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i}^2, \end{aligned} \quad (39)$$

where the first term of (38) has been integrated into the first two sums of the above equation.

The term  $\sum P_i[n-1]\tau_{def,i}$  is, by definition, equal to  $\tau_{def}[n-1]$ . The term  $\sum P_i[n-1]\tau_{def,i}^2$  can be expressed as follows:

$$\begin{aligned} \sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i}^2 &= \sum_{i=0}^{\infty} \left(1 - 2 \frac{\tau_{def}[n-1] - \tau_{def,0}}{\tau_{def}[n-1] - 2\tau_{def,0}}\right) \\ &\cdot \left(2 \frac{\tau_{def}[n-1] - \tau_{def,0}}{\tau_{def}[n-1] - 2\tau_{def,0}}\right)^i \left(\frac{\tau_{def,0}}{2}\right)^2, \end{aligned} \quad (40)$$

which, solving the series, yields

$$\sum_{i=0}^{\infty} P_i[n-1]\tau_{def,i}^2 = \frac{2\tau_{def}[n-1]\tau_{def,0}^2}{3\tau_{def,0} - \tau_{def}[n-1]}. \quad (41)$$

10. In particular, Cali et al. [24] analyze a dynamic 802.11 protocol, which is different from the standard one; Challa et al. [25] analyze the start-up of a simplified version of the protocol in which  $CW_{min} = CW_{max}$  and Foh and Zukerman [26] analyze the recovery time under a disaster scenario. None of these analyses models the transient behavior with a transfer function that can be used for a control theory study.

11. We note that while this assumption was not necessary in the previous sections, it is needed here to make the transient analysis (which is more complex) tractable.



Finally, combining all the above, we obtain the following equation that describes the system behavior under transient conditions:

$$\begin{aligned} \tau_{def}[n] = & \tau_{def}[n-1] + (1 - c_{def})\tau_{def,0}\tau_{def}[n-1] \\ & - (1 - c_{def}/2) \frac{2\tau_{def}[n-1]\tau_{def,0}^2}{3\tau_{def,0} - \tau_{def}[n-1]}, \end{aligned} \quad (42)$$

where  $c_{def}$  is a function of  $P_{ack}$  given by (36).

Note that by imposing stationary conditions (i.e.,  $\tau_{def}[n-1] = \tau_{def}[n]$ ), the above equation results in (32).

#### 5.4 Linearized Model

The above transient analysis has resulted in a nonlinear relationship between  $\tau_{def}$  and  $P_{ack}$ . In order to analyze the problem from a control-theoretic standpoint, we need to obtain a linear relationship that can be captured by a transfer function. To achieve this, we linearize (42) around the stable point of operation of the system.<sup>12</sup>

The stable point of operation of the WLAN can be obtained from forcing  $\tau_{def}[n-1] = \tau_{def}[n]$  in (42) and isolating  $\tau_{def}$ . We express the perturbations around this point as  $\Delta\tau_{def} + \Delta\tau_{def}$ . When these perturbations are small, they can be approximated by

$$\Delta\tau_{def}[n] \approx \frac{\partial\tau_{def}[n]}{\partial\tau_{def}[n-1]} \Delta\tau_{def}[n-1] + \frac{\partial\tau_{def}[n]}{\partial P_{ack}} \Delta P_{ack}, \quad (43)$$

where  $\tau_{def}[n]$  is the right-hand-side expression of (42).

The above expression provides a linear relationship between  $\tau_{def}[n]$  and  $P_{ack}$ ; however, in order to obtain  $H(z)$ , we need to find a linear relationship between  $P_t[n]$  and  $P_{ack}$ . We do this as follows:

$$\Delta P_t[n] \approx \frac{\partial P_t[n]}{\partial P_t[n-1]} \Delta P_t[n-1] + \frac{\partial P_t[n]}{\partial P_{ack}} \Delta P_{ack}, \quad (44)$$

where

$$\begin{aligned} \frac{\partial P_t[n]}{\partial P_t[n-1]} &= \frac{\partial P_t[n]}{\partial\tau_{def}[n]} \frac{\partial\tau_{def}[n]}{\partial\tau_{def}[n-1]} \frac{\partial\tau_{def}[n-1]}{\partial P_t[n-1]} = \\ &= \frac{\partial\tau_{def}[n]}{\partial\tau_{def}[n-1]} \end{aligned} \quad (45)$$

and

$$\frac{\partial P_t[n]}{\partial P_{ack}} = \frac{\partial P_t[n]}{\partial\tau_{def}[n]} \frac{\partial\tau_{def}[n]}{\partial P_{ack}}. \quad (46)$$

With the above, we have the following expression for the relationship between  $\Delta P_t$  and  $\Delta P_{ack}$ :

$$\Delta P_t[n] = H_1 \Delta P_t[n-1] + H_2 \Delta P_{ack}, \quad (47)$$

where the expressions for the coefficients  $H_1$  and  $H_2$  are computed from (45) and (46) in [14, Appendix I].

By doing the Z-transform of the above equation, we obtain

$$\Delta P_t(z) = H_1 \Delta P_t(z) z^{-1} + H_2 \Delta P_{ack}(z) \quad (48)$$

from where by isolating  $\Delta P_t(z)/\Delta P_{ack}(z)$ , we finally obtain  $H(z)$ :

12. A similar approach was used in [27] to analyze RED from a control-theoretic standpoint.

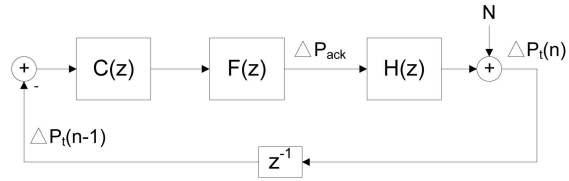


Fig. 4. Block diagram of the linearized system.

$$H(z) = \frac{H_2}{1 - H_1 z^{-1}}. \quad (49)$$

#### 5.5 Stability Analysis

We now study the system when it suffers perturbations around its point of operation and analyze the conditions that guarantee local stability.

Fig. 4 illustrates the linearized model when working around the stable operation point:

$$P_t = P_t + \Delta P_t, \quad (50)$$

$$P_{ack} = P_{ack} + \Delta P_{ack}. \quad (51)$$

Note that, as compared to the model of Fig. 3, in Fig. 4, only perturbations around the stable operation point are considered.

The closed-loop transfer function of the system of Fig. 4 is given by

$$T(z) = \frac{C(z)H(z)F(z)}{1 + z^{-1}C(z)H(z)F(z)}. \quad (52)$$

Substituting (25), (30), and (49) into the above yields

$$T(z) = \frac{K_p \alpha H_2}{(1 - (1 - \alpha)z^{-1})(1 - H_1 z^{-1}) + z^{-1}K_p \alpha H_2}, \quad (53)$$

which can be rewritten as

$$T(z) = \frac{K_p \alpha H_2 z^2}{z^2 + a_1 z + a_2}, \quad (54)$$

with

$$a_1 = K_p \alpha H_2 - H_1 - (1 - \alpha), \quad (55)$$

$$a_2 = H_1(1 - \alpha). \quad (56)$$

A sufficient condition for stability is that the poles of the above polynomial fall within the unit circle  $|z| < 1$ . This can be ensured by choosing coefficients  $\{a_1, a_2\}$  of the characteristic polynomial that belong to the stability triangle [28]:

$$a_2 < 1, \quad (57)$$

$$a_1 < a_2 + 1, \quad (58)$$

$$a_1 > -1 - a_2. \quad (59)$$

Equation (57) is met given that  $1 - \alpha < 1$  and  $H_1 < 1$ . Equation (59) is met given that

$$-1 - a_2 = -1 - H_1 + H_1 \alpha < 1 - H_1 + \alpha < a_1. \quad (60)$$

Equation (57) imposes the following restriction:

$$K_p \alpha H_2 - H_1 - (1 - \alpha) < H_1(1 - \alpha) + 1 \quad (61)$$

from which we obtain the following restriction on  $K_p$ :

$$K_p < \left( \frac{2 - \alpha}{\alpha} \right) \frac{1 + H_1}{H_2}. \quad (62)$$

As long as the configuration of  $K_p$  is smaller than the above expression, the system is guaranteed to be stable. However,  $H_1$  and  $H_2$  in the above expression are a function of the number of active DCF stations,  $N_{dcf}$ , and the behavior of the EDCA stations, given by  $P_{edca}$ . These values are not known a priori and may vary with time.

In order to assure stability, we need to find some upper bound for  $K_p$  that guarantees stability independent of  $N_{dcf}$  and  $P_{edca}$ . This bound is given by Theorem 1 (in [14, Appendix II]), which shows that as long as  $K_p$  is configured smaller than  $K_p^{max}$ , the system will be stable,  $K_p^{max}$  being a constant value given by the following expression:

$$K_p^{max} = \left( \frac{2 - \alpha}{\alpha} \right) \frac{1 + H_1^{min}}{H_2^{max}}, \quad (63)$$

where the expressions for  $H_1^{min}$  and  $H_2^{max}$  are given in [14, Appendix II]. This terminates the stability analysis.

## 5.6 Parameter Setting

The stability analysis conducted in the previous section provides a range for the parameters values, where the system is guaranteed to be stable. In this section, we propose specific rules for setting the parameters  $\alpha$  and  $K_p$  within this range. The proposed rules aim at 1) ensuring that the system behaves stably while reacting quickly to changes and 2) eliminating from the system the noise caused by the oscillations of  $P_t$ . In the following, we first fix  $\alpha$  and then, with the given value of  $\alpha$ , we set  $K_p$  such that these two objectives are met.

The parameter  $\alpha$  of the low-pass filter is fixed as follows: The goal of the low-pass filter is to eliminate the fluctuations introduced to the system by  $P_t$ . Since, with a transmission probability of  $P_{t,max}$ , there is approximately one transmission every  $P_{t,max}$  samples, the frequency that needs to be filtered out is approximately equal to  $2\pi/P_{t,max}$ . Following this reasoning, we impose as design criterion that the low-pass filter reduces this frequency by a factor  $G_F$ :

$$|F(2\pi/P_{t,max})| = G_F. \quad (64)$$

With the above, the problem of configuring  $\alpha$  is reformulated as to finding the value that satisfies (64). Combining this with (30) yields

$$\left| \frac{\alpha}{1 - (1 - \alpha)[\cos w + j \sin w]} \right|^2 = G_F^2, \quad (65)$$

where

$$w = \frac{2\pi}{P_{t,max}}. \quad (66)$$

Operating on the above, we obtain a second order equation from which we can isolate  $\alpha$ :

$$\alpha = \frac{-(1 - \cos w) + \sqrt{(1 - \cos w)^2 + 2(G_F^2 - 1)(1 - \cos w)}}{G_F^2 - 1}, \quad (67)$$

which terminates the setting of  $\alpha$ .

Given the above  $\alpha$  setting, we next address the configuration of the parameter  $K_p$  in order to meet the two goals set at the beginning of this section. We start by analyzing the setting of  $K_p$  following stability considerations.

From a stability standpoint, we have a trade-off between system stability and speed of reaction to changes. The larger  $K_p$ , the fastest the system reacts to changes; however, if  $K_p$  is chosen too large, the system becomes unstable (as we have seen in the previous section). In order to determine the right trade-off between these two effects in the setting of the  $K_p$  parameter, we follow the Ziegler-Nichols rules [21], which are widely used to configure proportional controllers. According to these rules, we impose that this parameter cannot be larger than one-half of the maximum value that guarantees stability,

$$K_p \leq K_p^{stability} = \frac{K_p^{max}}{2}. \quad (68)$$

In addition to the above,  $K_p$  also needs to be set according to the objective of eliminating the noise from the system. The noise caused by the fluctuations of  $P_t$  around frequency  $w$  is amplified into the input signal  $P_{ack}$  by  $|C(w)F(w)|$ . In order to avoid that this noise causes too large oscillations on the input signal, we impose as a design criterion that this gain is no larger than a factor  $G_{CF}$ :

$$|C(w)F(w)| = K_p G_F \leq G_{CF}. \quad (69)$$

Isolating  $K_p$  from the above equation, we obtain the largest  $K_p$  allowed by the considerations on noise:

$$K_p \leq K_p^{noise} = \frac{G_{CF}}{G_F}. \quad (70)$$

Finally, based on the above, we configure  $K_p$  as follows to guarantee that the two objectives set at the beginning of this section on stability and noise are met:

$$K_p = \min(K_p^{stability}, K_p^{noise}). \quad (71)$$

Note that the configuration proposed above depends on the setting of two parameters,  $G_F$  and  $G_{CF}$ . To provide appropriate filtering and attenuate noise, these parameters should take small values. Furthermore, to allow sufficiently large  $K_p^{noise}$  values, (70) imposes  $G_{CF} \gg G_F$ . Following these considerations, in this paper, we take  $G_{CF} = 10^{-2}$  and  $G_F = 10^{-4}$ .

## 6 PERFORMANCE EVALUATION

In order to evaluate the performance of DACKS, we have performed an exhaustive set of simulation experiments. For the simulations, we have extended the simulator used in [16], [29]; this is an event-driven simulator that closely follows the details of the MAC protocol of 802.11 EDCA. For all tests, we have taken a fixed frame payload size of 1,000 bytes and the system parameters of the IEEE 802.11b physical layer [30]. For the simulation results, average and 95 percent confidence interval values are given (note that in many cases, confidence intervals are too small to be appreciated in the graphs). Sections 6.1-6.9 focus on a single EDCA Access Category (AC 1) and saturated conditions, while the experiments of Sections 6.10 and 6.11 extend the evaluation to more than one AC and nonsaturation, respectively. Additional simulation results and more detailed explanations can be found in [14].

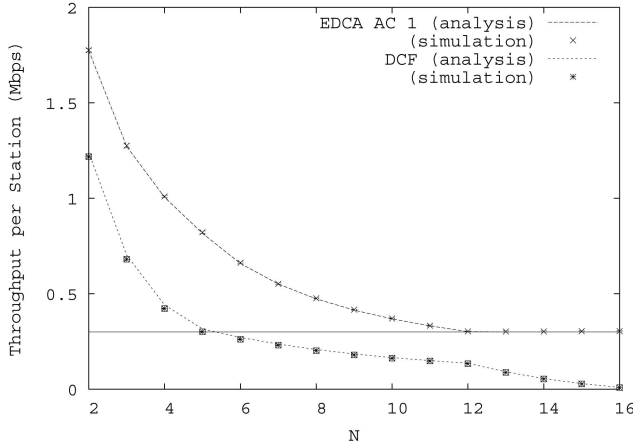


Fig. 5. Throughput guarantees.

### 6.1 Throughput Guarantees

In our first experiment, we evaluated the ability of DACKS to provide throughput guarantees to the EDCA stations. To this aim, we considered a scenario with  $N_{edca}$  EDCA stations, all belonging to the same AC (AC 1), and  $N_{dcf}$  DCF stations. The EDCA stations were given a throughput guarantee of 300 Kbps. We took  $N_{edca} = N_{dcf} = N$  and varied  $N$  from 2 to the maximum number of stations allowed by our admission control algorithm. The results of this experiment are illustrated in Fig. 5. Analytical results are represented with lines, and simulations with points with error bars. A horizontal line is used to show the guaranteed throughput. We conclude from the figure that the proposed DACKS technique is effective in providing throughput guarantees, since EDCA stations never have a throughput below 300 Kbps. Additionally, we observe that analytical results follow simulations closely, which validates our analytical model.

### 6.2 Number of DCF Stations

In the experiment of the previous section, the number of EDCA stations has been taken equal to the number of DCF stations. In order to evaluate the performance of DACKS in scenarios with different numbers of EDCA and DCF stations, we performed the following experiment: We fixed the number of EDCA stations ( $N_{edca}$ ) to 5 (low load), 10 (medium load), and 15 (high load) stations, and varied the number of DCF stations ( $N_{dcf}$ ) from 2 to 20. The resulting throughputs for EDCA and DCF stations (the latter in a subplot), obtained analytically and via simulation, are given in Fig. 6. Results confirm the effectiveness of DACKS under a variable number of DCF stations.

### 6.3 Total Throughput

In addition to providing throughput guarantees, one of our goals is also to optimize the overall throughput performance. In order to assess the performance of the  $CW_i$  configuration proposed in Section 4.4, we compared the total throughput obtained with our  $CW_i$  setting against the result of performing an exhaustive search over  $CW_i$  and choosing the best configuration. Specifically, in the exhaustive search, we evaluated all possible  $CW_i$  values, choosing for each one the largest  $P_{ack}$  that ensured the desired throughput guarantees, and took the  $CW_i$  value that provided the largest

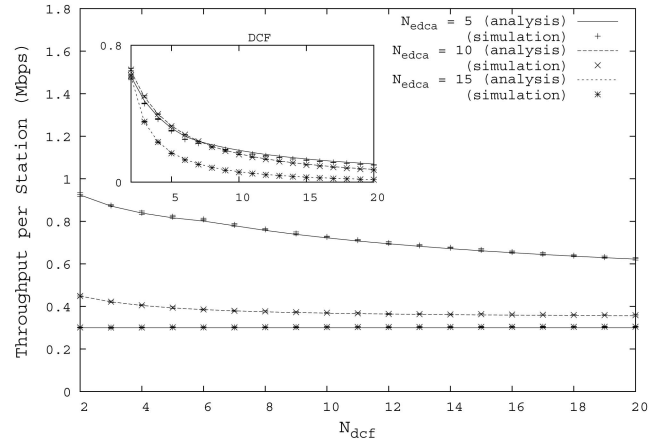


Fig. 6. Number of DCF stations.

total throughput. The results of this experiment are given in Table 1 as a function of  $N_{edca} = N_{dcf} = N$ . We can see that the total performance achieved by our configuration follows closely the one resulting from the exhaustive search. Based on these results, we conclude that our scheme is effective in optimizing the overall throughput performance.

### 6.4 WLAN without DACKS

In order to assess the benefits gained from DACKS, we compared its performance against a WLAN without DACKS configured according to the two following strategies:

- *Standard configuration*: EDCA stations are configured with the  $CW_i$  setting recommended by the standard [2] for voice traffic, which is the one that gives the highest priority to EDCA over DCF.
- *Optimal configuration*: For each  $N$  value, we configure EDCA stations with the  $CW_i$  setting that maximizes their throughput, which leads to maximizing the admissibility region.

Results on the total throughput and the throughput of EDCA and DCF stations are given in Fig. 7. We first observe that DACKS outperforms the strategies without DACKS in terms of total throughput. Looking at the per station throughputs, we see that the three approaches give similar throughput to EDCA stations, while DACKS provides a substantial larger throughput to DCF stations. The reasons for this improvement are further analyzed in the next experiment.

We further observe that DACKS allows admitting more EDCA stations while meeting the throughput guarantees. Indeed, up to 16 stations can be admitted with DACKS, while only 13 and 9 stations can be admitted with the optimal and standard configurations, respectively. We conclude that

TABLE 1  
Total Throughput (in megabits per second) for the Proposed Algorithm and the Exhaustive Search

N	Algorithm	Search	N	Algorithm	Search
2	6.0277	6.1101	10	5.3441	5.4177
4	5.7867	5.8089	12	5.2513	5.2599
6	5.5799	5.6504	14	4.9821	4.9925
8	5.4488	5.5325	16	4.9502	4.9518

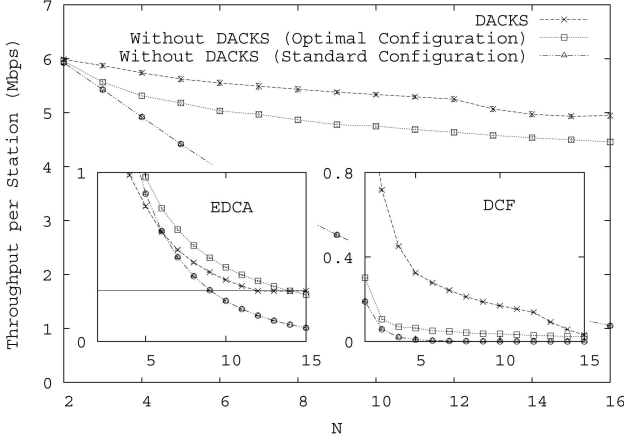


Fig. 7. WLAN without DACKS (main plot: total throughput; subplots: throughput per EDCA/DCF station).

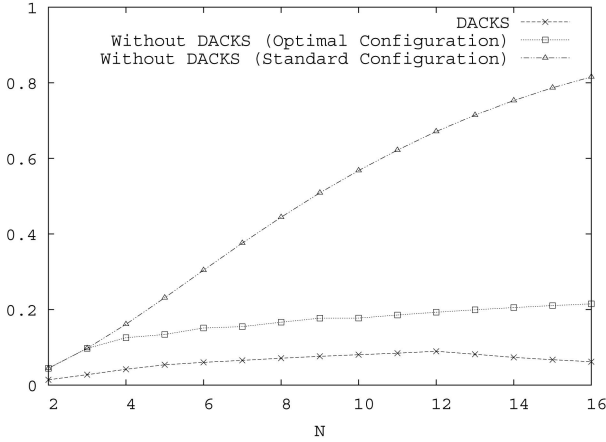


Fig. 8. Collision rate.

DACKS benefits both DCF stations (by providing them with more throughput) and EDCA stations (by increasing the number of stations that can be admitted).

### 6.5 Collision Rate

The reason for the performance improvement achieved with DACKS is that, although DACKS wastes some time in the retransmission of successful frames whose Acks are skipped, a WLAN without DACKS wastes much more time in collisions. Indeed, in a WLAN without DACKS, the aggressiveness of DCF stations cannot be controlled, and as a consequence, EDCA stations need to behave aggressively as well, which results in many collisions. In order to illustrate this behavior, Fig. 8 shows the collision rate with DACKS for the same scenario as the previous experiment and compares it against the collision rate for the strategies without DACKS. This result confirms that the collision rate with DACKS is indeed much smaller than that with the other approaches.

### 6.6 Stability

One of the objectives of the configuration setting computed in Section 5 is to ensure that the system is stable. In order to evaluate the stability of our configuration, we analyzed the evolution of the control signal ( $P_{ack}$ ) over time and compared it against a configuration with  $K_p$  set to a value 100 times larger. Fig. 9 depicts the time plots for our configuration (straight line) and for the configuration with larger  $K_p$

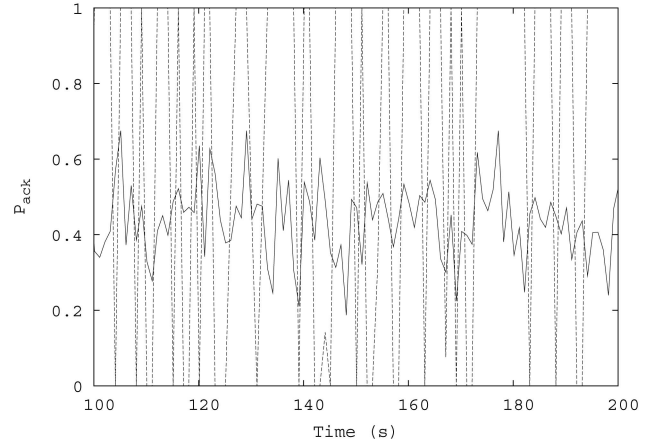


Fig. 9. Stability.

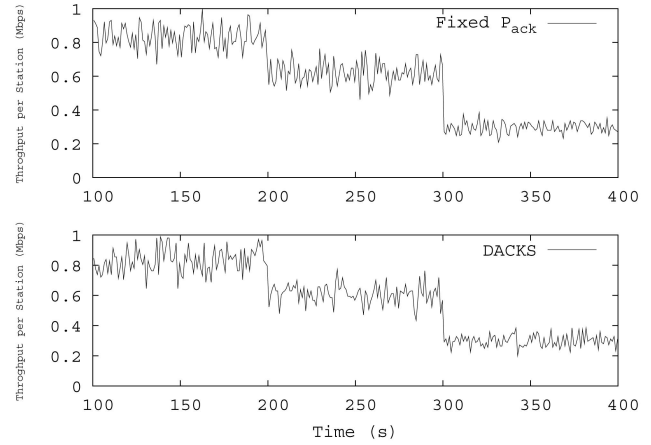


Fig. 10. Changing conditions.

(dotted line) for a scenario with  $N = 15$ . We observe from the figure that with our configuration,  $P_{ack}$  oscillates stably around the average value, while the configuration with larger  $K_p$  shows an unstable behavior with large oscillations of  $P_{ack}$  that go from 0 (where DCF stations are starved) to 1 (where DCF stations are uncontrolled). These results confirm the effectiveness of our configuration to ensure stability.

### 6.7 Changing Conditions

In addition to stability, another objective of the configuration setting computed in Section 5 is to ensure that the system reacts quickly upon changes. In order to study the system's transient response to changes, we performed the following experiment: Initially, we had the system operating with  $N_{edca} = N_{def} = 5$ . At some time instant ( $t = 200$  seconds), we introduced 10 additional DCF stations in the system ( $N_{def} = 15$ ). At some later instant ( $t = 300$  seconds), we introduced 10 further additional EDCA stations ( $N_{edca} = 15$ ), which (in contrast to the previous case) triggered the corresponding configuration update. Fig. 10 depicts the time plot of the throughput of one EDCA station. As a benchmark to assess the response of our system, we compare the instantaneous throughput with DACKS against that of a system where  $P_{ack}$  is immediately changed to a fixed new value upon the stations' arrival. We observe from the figure that DACKS reacts quickly and smoothly to the changes. This and the previous experiments confirm the proposed configuration setting in terms of stability and response to changes.

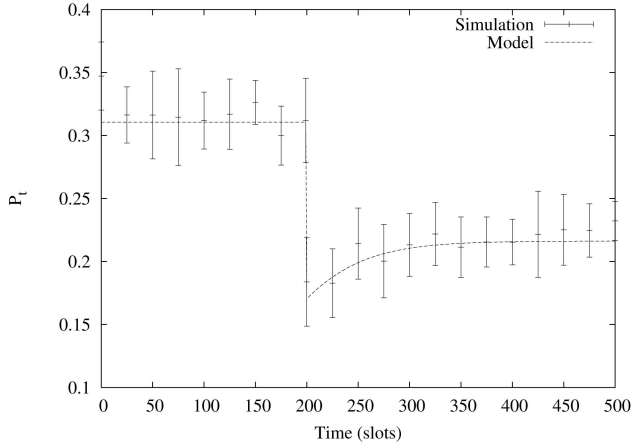


Fig. 11. Validation of the transient model.

### 6.8 Validation of the Transient Model

One of the main contributions of this paper is the transient analysis of 802.11 presented in Section 5.3. In order to validate the model proposed, we performed the following experiment: We had 10 DCF stations in the WLAN, and at slot time 200, five of the stations left. Fig. 11 illustrates the evolution of the total transmission probability in the channel,  $P_t$ , according to our transient model and simulations. For the simulations, the total probability is computed by taking into account the backoff stage of each station and the corresponding transmission probability at this backoff stage as given by (3). We observe that simulation results follow our model; although there is a large degree of variability in the simulations, caused by the inherent randomness of  $P_t$ , the results given by our model fall within the confidence intervals of simulation results, which confirms the validity of the model.

### 6.9 Inactive Stations

One of the design goals of the proposed DACKS scheme is its ability to dynamically adapt to the number of active DCF and EDCA stations. Specifically, the proposed scheme automatically adjusts  $P_{ack}$  to the traffic actually transmitted in the WLAN, in order to avoid degrading unnecessarily the throughput experienced by DCF stations. In order to evaluate this feature, we performed the following experiment: We had the WLAN configured to support  $N_{edca} = N_{dcf} = 16$  stations, with a throughput request of 300 Kbps for each EDCA station, and had that only  $N_{active}$  of the EDCA and DCF stations were active. To understand the benefit of adjusting  $P_{ack}$  dynamically, we compared DACKS against a static configuration, where  $P_{ack}$  was computed in order to provide the desired throughput guarantees with  $N_{edca} = N_{dcf} = 16$ . Fig. 12 illustrates the throughput of a DCF station resulting from this experiment. We observe that DACKS achieves the objective of minimizing the disruption suffered by the DCF stations while the static configuration severely degrades the DCF throughput. We conclude that the proposed adaptive DACKS approach outperforms very significantly the static approach proposed in [4].

### 6.10 Multiple ACs

The experiments performed so far involve one single EDCA Access Category with throughput guarantees. To gain insight into the performance of DACKS with more than

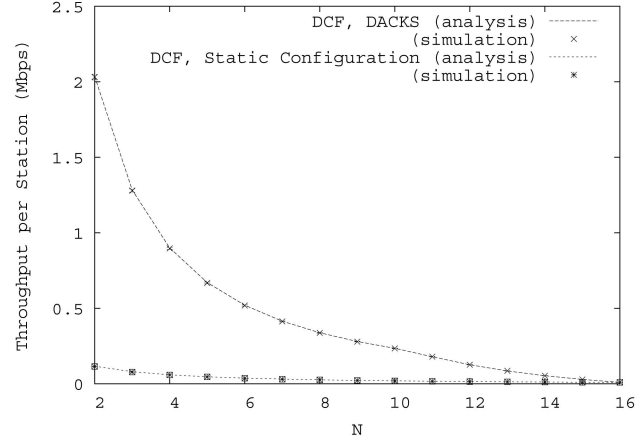


Fig. 12. Inactive stations.

one AC, we conducted the following experiment: We had four ACs, with throughput guarantees of 300, 150, 75, and 37.5 Kbps to AC 1, AC 2, AC 3, and AC 4, respectively. Fig. 13 illustrates the throughput obtained by the EDCA stations of the different ACs. The results confirm the effectiveness of DACKS under multiple ACs; in particular, the desired throughput guarantees are always met for all ACs.

### 6.11 Nonsaturated Traffic

All previous experiments have been performed with all stations saturated. In order to evaluate DACKS under different traffic conditions, we repeated the experiment of the previous section under nonsaturation. Specifically, we considered the following traffic models:

1. EDCA stations of AC 1 and DCF stations were saturated.
2. EDCA stations of AC 2 generated traffic at a constant rate.
3. EDCA stations of AC 3 generated traffic following a Poisson process with an average rate equal to its guaranteed rate.
4. EDCA stations of AC 4 generated traffic following a Pareto process of shape 2.

The results obtained, illustrated in Fig. 14, show that our technique is also effective under nonsaturated conditions.

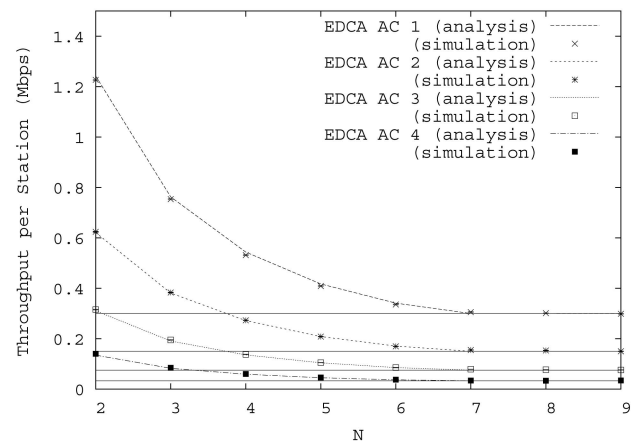


Fig. 13. Multiple ACs.

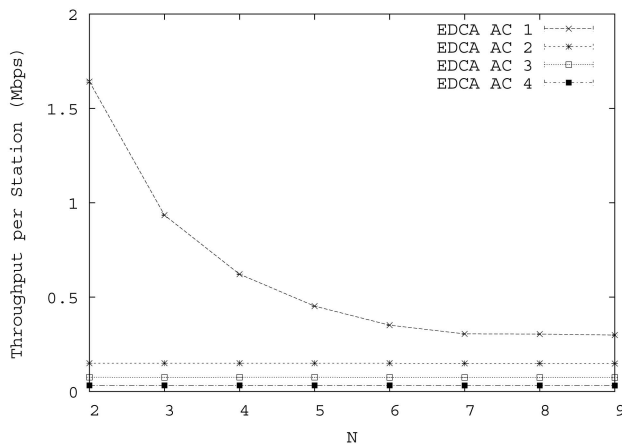


Fig. 14. Nonsaturated traffic.

In particular, all the ACs see their desired throughput guarantees satisfied independent of their arrival process.

## 7 SUMMARY AND FINAL REMARKS

The EDCA mechanism of the IEEE 802.11e standard is backward compatible, thereby allowing legacy DCF stations to interoperate in a WLAN working under the EDCA mechanism. However, the coexistence of EDCA and DCF stations in the same WLAN stations degrades performance substantially. In particular, the presence of DCF stations jeopardizes the service guarantees committed to the EDCA stations and degrades the overall efficiency of the WLAN. The reason for this performance degradation is that DCF stations compete with overly small  $CW$ s values, and these values cannot be modified since they are predefined by the standard.

In this paper, we have proposed the DACKS technique to overcome the above problem. With DACKS, upon receiving a frame from a DCF station, the AP skips the Ack reply with some probability. When missing the Ack reply, DCF stations assume that the transmitted frame collided and double their  $CW$ . This allows having some control on the average  $CW$ s used by the DCF stations and thereby overcoming the above problem which was caused by the lack of control on the  $CW$ s of the DCF stations.

One of the major challenges with the DACKS scheme is the configuration of the probability of skipping the Ack reply. This probability should be configured in order to preserve the committed service guarantees to the EDCA stations while minimizing the disruption suffered by the DCF stations. We argue that these goals require the skipping probability to be dynamically configured. Indeed, if the skipping probability was statically set, we would have to choose a conservative configuration to avoid failing to meet EDCA service guarantees when all stations are active. As a result, when some of the stations were inactive, the skipping probability would be too high and DCF stations would see their throughput performance unnecessarily reduced.

The system proposed to dynamically tune the skipping probability is based on the observation that, as long as the overall transmission probability in the WLAN does not exceed a certain threshold, EDCA stations are guaranteed to receive the committed service. Following this observation, the controller used by our system takes as input the observed

transmission probability and provides as output the skipping probability. The algorithm that we have chosen in this paper to compute the output control signal based on the measured input is based on a Proportional Controller.

One of the challenges of our DACKS system is the configuration of the gain of the proportional controller. This has been addressed in the paper in the following two steps: In the first step, we have conducted a performance analysis of our system under stationary conditions to obtain the maximum transmission probability in the WLAN that guarantees EDCA stations receive the committed throughputs, which has been used as the reference signal of the DACKS controller. In the second step, we have conducted an analysis of our system under transient conditions. Based on this analysis, we have studied our system from a control-theoretic standpoint and found the conditions that need to be met in order to guarantee that our system is stable. Following considerations from control theory, we have then set the gain of the Proportional Controller as a trade-off between stability and speed of reaction.

The proposed scheme has been exhaustively evaluated by means of simulations. The performance evaluation conducted has shown the following:

1. DACKS is effective in providing throughput guarantees to EDCA stations.
2. The chosen configuration maximizes the overall efficiency.
3. A WLAN with DACKS is more efficient than a WLAN that does not use the DACKS technique.
4. Our technique avoids disrupting DCF stations in case some of the (EDCA or DCF) stations are not active.
5. Our closed-loop system behaves stably while reacting quickly upon changing conditions.

Although the focus of this paper has been on providing EDCA stations with throughput guarantees, the proposed scheme can also be used to provide delay guarantees. Indeed, the key idea of DACKS is to regulate the DCF stations to ensure that the transmission probability in the channel does not exceed a given value. Following this, the value of the transmission probability that ensures the desired delay guarantees can be computed based on the model of [31], and then DACKS can be used to provide these guarantees.

## ACKNOWLEDGMENTS

The authors thank Dr. José Félix Kukiélka for having carefully read the manuscript. They are grateful to the anonymous referees for their valuable comments, which greatly helped in improving the paper. The research leading to these results received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 214994 (CARMEN project). It was also partly funded by the Ministry of Science and Innovation of Spain, under the QUARTET project (TIN2009-13992-C02-01).

## REFERENCES

- [1] IEEE 802.11 WG, *Information Technology—Telecomm. and Information Exchange between Systems. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, Aug. 1999.

- [2] IEEE 802.11 WG, *Amendment to Standard for Information Technology. LAN/MAN Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, Supplement to IEEE 802.11 Standard, IEEE, Nov. 2005.
- [3] IEEE 802.11 WG, *Information Technology—Telecomm. and Information Exchange between Systems. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11-REVma/D9.0, Revision of Standard 802.11-1999, IEEE, 2006.
- [4] L. Vollero, A. Banchs, and G. Iannello, "ACKS: A Technique to Reduce the Impact of Legacy Stations in 802.11e EDCA WLANs," *IEEE Comm. Letters*, vol. 9, no. 4, pp. 346-348, Apr. 2005.
- [5] A. Banchs, P. Serrano, and L. Vollero, "Reducing the Impact of Legacy Stations on Voice Traffic in 802.11e EDCA WLANs," *IEEE Comm. Letters*, vol. 11, no. 4, pp. 331-333, Apr. 2007.
- [6] G.-H. Hwang and D.-H. Cho, "Performance Analysis on Coexistence of EDCA and Legacy DCF Stations in IEEE 802.11 Wireless LANs," *IEEE Trans. Wireless Comm.*, vol. 5, no. 12, pp. 3355-3359, Dec. 2006.
- [7] G. Bianchi, I. Tinnirello, and L. Scalia, "Understanding 802.11e Contention-Based Prioritization Mechanisms and Their Coexistence with Legacy 802.11 Stations," *IEEE Network*, vol. 19, no. 4, pp. 28-34, Aug. 2005.
- [8] P.E. Engelstad and O.N. Osterbo, "Analysis of the Total Delay of IEEE 802.11e EDCA and 802.11 DCF," *Proc. IEEE Int'l Conf. Comm. (ICC '06)*, June 2006.
- [9] J. Villalon, P. Cuenca, and L. Orozco-Barbosa, "On the Capabilities of IEEE 802.11e for Multimedia Communications over Heterogeneous 802.11/802.11e WLANs," *Telecomm. Systems*, vol. 36, nos. 1-3, pp. 27-38, Nov. 2007.
- [10] J. Majkowski and F.C. Palacio, "QoS Protection for IEEE 802.11e in WLAN with Shared EDCA and DCF Access," *Proc. Conf. Comm. Systems and Networks (CSN)*, Aug. 2006.
- [11] H. Al-Mefleh and J.M. Chang, "A New ACK Policy to Mitigate the Effects of Coexisting IEEE 802.11/802.11e Devices," *Proc. IEEE INFOCOM*, Apr. 2008.
- [12] J. Villalon, P. Cuenca, L. Orozco-Barbosa, and A. Garrido, "B-EDCA: A QoS Mechanism for Multimedia Communications over Heterogeneous 802.11/802.11e WLANs," *Computer Comm.*, vol. 31, no. 17, pp. 3905-3921, Nov. 2008.
- [13] L. Vollero and G. Iannello, "ACK Skipping: Enabling QoS for Multimedia Communications in WiFi Hot Spots," *Int'l J. High-Performance Computing and Networking*, vol. 4, nos. 1/2, pp. 23-30, July 2006.
- [14] A. Banchs, P. Serrano, and L. Vollero, "Providing Service Guarantees in 802.11e EDCA WLANs with Legacy Stations," technical report, Univ. Carlos III of Madrid, [http://www.it.uc3m.es/banchs/papers/dynamic\\_acks.pdf](http://www.it.uc3m.es/banchs/papers/dynamic_acks.pdf), 2010.
- [15] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Selected Areas Comm.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.
- [16] A. Banchs and L. Vollero, "Throughput Analysis and Optimal Configuration of 802.11e EDCA," *Computer Networks*, vol. 50, no. 11, pp. 1749-1768, Aug. 2006.
- [17] F. Cali, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Networking*, vol. 8, no. 6, pp. 785-799, Dec. 2000.
- [18] V. Ramaiyan, A. Kumar, and E. Altman, "Fixed Point Analysis of Single Cell IEEE 802.11e WLANs: Uniqueness and Multistability," *IEEE/ACM Trans. Networking*, to appear.
- [19] A. Banchs, X. Pérez-Costa, and D. Qiao, "Providing Throughput Guarantees in IEEE 802.11e Wireless LANs," *Proc. 18th Int'l Teletraffic Congress (ITC-18)*, Sept. 2003.
- [20] B. Kristiansson and B. Lennartson, "Robust Tuning of PI and PID Controllers," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 55-69, Feb. 2006.
- [21] G.F. Franklin, J.D. Powell, and M.L. Workman, *Digital Control of Dynamic Systems*, second ed. Addison-Wesley, 1990.
- [22] A.K. Palit and D. Popovic, *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications*. Springer-Verlag, 2005.
- [23] H. Chen and A.M. Haimovich, "Iterative Estimation and Cancellation of Clipping Noise for OFDM Signals," *IEEE Comm. Letters*, vol. 7, no. 7, pp. 305-307, July 2003.
- [24] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE J. Selected Areas Comm.*, vol. 18, no. 9, pp. 1774-1786, Sept. 2000.
- [25] R.K. Challa, S. Chakrabarti, and D. Datta, "Modeling of IEEE 802.11 DCF for Transient State Conditions," *J. Networks*, vol. 2, no. 4, pp. 14-19, Aug. 2007.
- [26] C.H. Foh and M. Zukerman, "Performance Evaluation of IEEE 802.11," *Proc. IEEE Vehicular Technology Conf. (VTC)*, May 2001.
- [27] C.V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A Control Theoretic Analysis of RED," *Proc. IEEE INFOCOM*, Apr. 2001.
- [28] K. Åström and B. Wittenmark, *Computer-Controlled Systems, Theory and Design*, second ed. Prentice Hall, 1990.
- [29] P. Serrano, A. Banchs, and A. Azcorra, "A Throughput and Delay Model for IEEE 802.11e EDCA under Non Saturation," *Wireless Personal Comm.*, vol. 43, no. 2, pp. 467-479, Oct. 2007.
- [30] IEEE 802.11 WG, *Information Technology—Telecomm. and Information Exchange between Systems. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 2.4 GHz Band*, Supplement to IEEE 802.11 Standard, IEEE, Sept. 1999.
- [31] A. Banchs and L. Vollero, "A Delay Model for IEEE 802.11e EDCA," *IEEE Comm. Letters*, vol. 9, no. 6, pp. 508-510, June 2005.



**Albert Banchs** received the telecommunication engineering degree from the Polytechnical University of Catalonia in 1997 and the PhD degree from the same university in 2002. Since 2003, he has been with the University Carlos III of Madrid. His PhD dissertation received the national award for the best thesis on broadband networks. He was a visiting researcher at the International Computer Science Institute, Berkeley, in 1997, worked for Telefonica I+D in 1998, and worked for NEC Europe Ltd., Germany, from 1998 to 2003. He is the author of more than 50 publications in peer-reviewed journals and conferences and has four patents (two of them granted). He is an associate editor for *IEEE Communications Letters* and has been the guest editor for *IEEE Wireless Communications* and for *Computer Networks*. He has served on the program committees of a number of conferences and workshops including IEEE INFOCOM, IEEE ICC, and IEEE GLOBECOM, and is the program committee chair for European Wireless 2010. He is a member of the IEEE.



**Pablo Serrano** received the telecommunication engineering degree and the PhD degree from the Universidad Carlos III de Madrid (UC3M) in 2002 and 2006, respectively. He has been in the Telematics Department of UC3M since 2002, where he is currently an assistant professor. In 2007, he was a visiting researcher in the Computer Network Research Group at the University of Massachusetts Amherst, partially supported by the Spanish Ministry of Education under a José Castillejo grant. His current research focuses on the performance evaluation of wireless networks. He has more than 20 scientific papers in peer-reviewed international journals and conferences. He also serves as a program committee member of several international conferences, including IEEE GLOBECOM and IEEE INFOCOM. He is a member of the IEEE.



**Luca Vollero** received the MS degree in telecommunications engineering and the PhD degree in computer science from the University of Naples "Federico II" in 2001 and 2005, respectively. He is an assistant professor in the Laboratory on Elaboration Systems and Bioinformatics at the University Campus Bio-Medico of Rome. His research interests include networks modeling and simulations, design and evaluation of systems for mobility in heterogeneous networks, multimedia data elaboration, and image processing. He is a member of the IEEE, the IEEE Computer Society, the IEEE Communications Society, and the ACM.