A Control Theoretic Approach to Distributed Optimal Configuration of 802.11 WLANs

Paul Patras*†, Albert Banchs*, Pablo Serrano* and Arturo Azcorra*†
*University Carlos III of Madrid, Spain
†IMDEA Networks, Madrid, Spain

***Abstract*—The optimal configuration of the contention parameters of a WLAN depends on the network conditions in terms of number of stations and the traffic they generate. Following this observation, a considerable effort in the literature has been devoted to the design of distributed algorithms that optimally configure the WLAN parameters based on current conditions. In this paper we propose a novel algorithm that, in contrast to previous proposals which are mostly based on heuristics, is sustained by mathematical foundations from multivariable control theory. A key advantage of the algorithm over existing approaches is that it is compliant with the 802.11 standard and can be implemented with current wireless cards without introducing any changes into the hardware or firmware. We study the performance of our proposal by means of theoretical analysis, simulations and a real implementation. Results show that the algorithm substantially outperforms previous approaches in terms of throughput and delay.**

***Index Terms*—Wireless LAN, IEEE 802.11, DCF, adaptive MAC, distributed algorithm, multivariable control theory**

I. Introduction

The throughput performance of the DCF mechanism of 802.11 Wireless LANs (WLANs) depends on the number of active stations and the Contention Window (CW) with which they contend. If too many stations use too small CW's, then the collision rate will be very high and consequently throughput performance will be low. Similarly, if few stations contend with too large CW's, the attempt rate will be low and the channel will be underutilized most of the time, yielding a poor throughput performance also in this case. In line with this explanation, many works in the literature (e.g. [1], [2]) have shown that, given a number of actively contending stations, there exists an optimal CW configuration that maximizes the throughput performance.

The CW configuration used by the 802.11 standard [3] is statically set, independently of the number of contending stations. As a result, it does not provide optimal performance. In particular, standard 802.11 stations contend with overly small CW's, which yields a degraded performance as the number of contending stations in the WLAN increases. In order to avoid this undesirable behavior, many schemes have been proposed in the literature to dynamically adapt the CW to the current WLAN conditions. Although the various mechanisms differ in the details, their common aim is to adjust the CW configuration to the optimal value corresponding to the number of currently active stations and thereby maximize the WLAN throughput performance.

The approaches proposed so far for the configuration of 802.11 can be classified in the following two groups:

- *Centralized approaches* [4]–[7]. These approaches are based on a single node (the Access Point) that periodically computes the set of MAC layer parameters to be used and signals this configuration to all stations.
- *Distributed approaches* [8]–[11]. With these approaches, each station independently computes its own configuration. Among other advantages, this removes the signaling overhead and naturally fits the *ad-hoc* mode of operation of 802.11 which uses no Access Point.

In this paper we propose a novel distributed algorithm, based on control theory, that adaptively adjusts the CW configuration of the WLAN with the goal of maximizing the overall performance. The key advantages of the proposed algorithm over existing distributed approaches are:

- The proposed algorithm is sustained by mathematical foundations from the multivariable control theory field that guarantee convergence and stability while ensuring a quick reaction to changes. In contrast, most of the previous proposals are based on heuristics that lack these foundations.
- Our mechanism is standard-compliant and can be implemented with existing hardware. In contrast, the existing proposals change the 802.11 mechanism, which introduces additional complexity and requires modifying the hardware and/or firmware of existing wireless cards.
- In contrast to all previous proposals, which modify the contention parameters of all stations upon congestion, our algorithm only acts on those stations that are contributing to congestion; as a result, it provides stations that are not contributing to congestion with a better delay performance.

The rest of the paper is organized as follows. In Section II we summarize the 802.11 DCF mechanism. Section III presents the proposed algorithm. In Section IV we conduct a steady state analysis of the WLAN to derive the optimal collision probability that maximizes performance (which is used as the reference signal of our control system). In Section V we perform a control theoretic analysis of the system and based on this analysis we configure the controller parameters to guarantee a proper tradeoff between stability and speed of reaction to changes. Section VI validates the algorithm by means of simulations, and Section VII presents a prototype that proves the algorithm can be implemented with current hardware. Section VIII reviews related work and finally Section IX concludes the paper.

II. IEEE 802.11 DCF

In this section we briefly summarize the 802.11 DCF mechanism [3]. With DCF, a station with a new frame to transmit senses the channel. If this remains idle for a period of time equal to the DCF interframe space parameter ($DIFS$), the station transmits. Otherwise, if the channel is detected busy, the station monitors the channel until it is measured idle for a $DIFS$ time, and then executes a backoff process.

When the backoff process starts, the station computes a random number uniformly distributed in the range $(0, CW-1)$, and initializes its backoff time counter with this value. CW is called the contention window and for the first transmission attempt the minimum value (CW_{min}) is used. In case of a collision CW is doubled, 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. 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 failed attempt, all the retransmissions of the same frame are sent with the retry flag set. If the number of failed attempts reaches a predetermined retry limit, the frame is discarded. Once the backoff process is completed, CW is set again to CW_{min}.

As it can be seen from the above description, the behavior of a station depends on the CW_{min} and CW_{max} parameters. In the revised version of the standard [12], which incorporates the mechanisms defined in 802.11e [13], these are configurable parameters that can be set to different values for different stations.

The rest of the paper is devoted to the design of a standard compliant mechanism for the optimal setting of the above parameters. In order to benefit from the features of the binary exponential backoff algorithm, we set $CW_{max} = 2^m CW_{min}$, taking the m value of the default configuration (which is $m = 6$ in IEEE 802.11g), and concentrate on adaptively adjusting the CW_{min} parameter.

III. DAC Algorithm

In this section we present the proposed algorithm, hereafter referred to as *Distributed Adaptive Control* (DAC) algorithm. DAC adjusts the CW_{min} parameter of each station with the goal of driving the WLAN to the optimal point of operation.

To achieve the above goal, DAC uses a classical system from multivariable control theory [14] which is shown in Fig. 1. In this system, each station runs an independent controller that gives the CW_{min} value to be used by the station. In this paper we have chosen to use a well known controller from classical control theory, namely a *proportional-integral* (PI) controller.

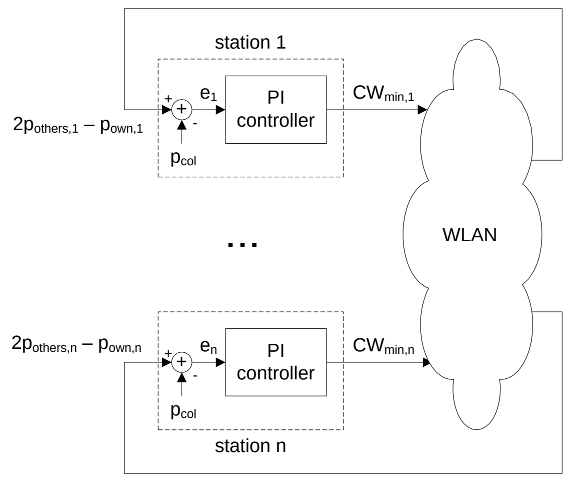

Fig. 1. DAC Algorithm

As it can be seen from Fig. 1, the PI controller of station i takes as input the error signal e_i and gives as output the $CW_{min,i}$ configuration of the station. The choice of the error signal e_i is a critical part of the design of the DAC algorithm, as it drives the system behavior both under steady and transient conditions.

In steady conditions, a key requirement for the choice of e_i is that there exists a single stable point of operation that yields optimal performance. This requirement is analyzed in Section IV, which shows that the system reaches the optimal point of operation by driving the collision probability to a desired value.

In transient conditions, we set the following requirements when choosing the error signal:

i) When the collision probability is far from its desired value, the error signal needs to be large in order to trigger a quick reaction towards the desired value.

ii) When the collision probability is around its desired value but stations do not share bandwidth fairly, the error should also be large in order to achieve a fair bandwidth sharing.

iii) In case of congestion, only the saturated stations[1]) should increase their $CW_{min,i}$, thus avoiding that the non-saturated stations (which are not contributing to congestion) are unnecessarily penalized.

In order to satisfy the above requirements, we take the error signal as the sum of two terms. The first one is:

$$e_{collision,i} = p_{others,i} - p_{col} \tag{1}$$

where $p_{others,i}$ is the probability that a transmission of a station different from i collides and p_{col} is the desired value for the collision probability. This term ensures that if the WLAN is operating at a different collision probability from the desired one, the error is large, achieving thus the first of the three requirements stated above.

[1]Following [1], with saturated station we refer to a station that always has packets ready for transmission.

The second term of the error signal is:

$$e_{fairness,i} = p_{others,i} - p_{own,i} \tag{2}$$

where $p_{own,i}$ is the probability that a transmission of station i collides. This term ensures that if two stations do not share the bandwidth fairly due to having different $CW_{min,i}$'s, the error will be large. Indeed, a station with a small $CW_{min,i}$ transmits with a large probability, and therefore its $p_{others,i}$ will be larger than $p_{own,i}$, yielding a large $e_{fairness,i}$. This fulfills the second requirement.

Additionally, the $e_{fairness,i}$ term also ensures that in case of congestion only the saturated stations increase their $CW_{min,i}$, which yields the last of the requirements stated above. This is caused by the fact that saturated stations have a larger transmission probability; as a result, their $p_{others,i}$ is larger and their $p_{own,i}$ smaller, which makes their $e_{fairness,i}$ larger.

The combination of Eqs. (1) and (2) yields the following error signal:

$$\begin{aligned} e_i &= e_{collision,i} + e_{fairness,i} \\ &= 2p_{others,i} - p_{own,i} - p_{col} \end{aligned} \tag{3}$$

where, as depicted in Fig. 1, the term $2p_{others,i} - p_{own,i}$ corresponds to the feedback signal measured from the WLAN and p_{col} is the reference signal, whose value is given in Section IV.

Having chosen the error signal as given by the above expression, the remaining key challenge for its computation is the measurement of the values of $p_{own,i}$ and $p_{others,i}$. In particular, the challenge lies in measuring these values by using only functionality available in current wireless cards. To achieve this, we proceed as follows.

To compute the own collision probability at station i, $p_{own,i}$, we take advantage of the following statistics which are readily available from wireless cards: the number of successful transmission attempts, denoted by T, and the number of unsuccessful attempts, F. $p_{own,i}$ is then computed by applying the following formula

$$p_{own,i} = \frac{F}{F+T} \tag{4}$$

The probability $p_{others,i}$ cannot be computed following the above procedure since with current hardware it is not possible to measure the unsuccessful attempts of other stations. Instead, we compute $p_{others,i}$ by looking at the retry flag of the frames successfully transmitted observed by station i. Let S be the number of frames with the retry bit unset, and R be the number of frames with the retry bit set. Then, if we assume that no frames are discarded due to reaching the retry limit, the collision probability $p_{others,i}$ can be computed as

$$p_{others,i} = \frac{R}{R+S} \tag{5}$$

With the above, each station i periodically measures $p_{others,i}$ and $p_{own,i}$ and computes the error signal e_i from these measurements. This error signal is then fed into the controller which triggers an update of $CW_{min,i}$. As a safeguard against too large and too small values of CW_{min}, when updating $CW_{min,i}$ we force that it can neither take values below a given lower bound nor above an upper bound. In particular, the values that we have chosen for the lower and upper bounds in this paper are the default CW_{min} and CW_{max} values used by the DCF standard (with the 802.11g physical layer, these are 16 and 1024, respectively).

Regarding the frequency with which the $CW_{min,i}$ is updated, in this paper we choose to update it every beacon interval, by triggering the algorithm upon the reception of a beacon frame. The key advantages of this choice are:

- It ensures compatibility with existing hardware, since WLAN cards conforming to the IEEE 802.11 revised standard are able to update the configuration of the CW_{min} parameter at the beacon frequency.
- It is a simple way to ensure that all the stations update their configuration with the same frequency.

As an exception to the above, if the number of samples used to compute $p_{others,i}$ or $p_{own,i}$ at the moment of receiving the beacon frame is smaller than 20, the update is not triggered but deferred until the next beacon. The reason is to avoid that a too small number of samples induces a high degree of inaccuracy in the estimation of these parameters. In what follows, we assume that there are always enough samples available and updates are never deferred.

From the above description of DAC, it can be seen that the algorithm relies on p_{col} as well as the parameters of the PI controller (namely K_p and K_i) [15]. The following two sections address the issue of properly configuring these parameters.

IV. Steady State Analysis

In the following we analyze the DAC algorithm under steady conditions and, based on this analysis, we compute the value of the p_{col} parameter that maximizes the throughput obtained in steady state. The analyses of this and the following section assume saturation conditions, while the simulation results presented in Section VI also cover the non-saturated case.

To analyze the system under steady conditions, we proceed as follows. Since the controller includes an integrator, this ensures that there is no steady state error [15]. The steady solution can therefore be obtained from imposing

$$e_i = 0 \ \forall i \tag{6}$$

from which

$$2p_{others,i} - p_{own,i} - p_{col} = 0 \tag{7}$$

Let τ_i be the probability that station i transmits at a given slot time [1]. $p_{own,i}$ and $p_{others,i}$ can be computed as a function of the τ_i's as follows. $p_{own,i}$ is the probability that a transmission of station i collides

$$p_{own,i} = 1 - \prod_{k \neq i} (1 - \tau_k) \tag{8}$$

$p_{others,i}$ is the average collision probability of the other stations measured by station i, which is computed by adding the individual collision probabilities of the other stations weighted by their transmission probability

$$p_{others,i} = \sum_{k \neq i} \frac{\tau_k}{\sum_{l \neq i} \tau_l} \left(1 - \prod_{l \neq k} (1 - \tau_l)\right) \tag{9}$$

By using the above expressions for $p_{others,i}$ and $p_{own,i}$, we can express Eq. (7) as a system of equations on the τ_i's. Theorem 1[2] guarantees the uniqueness of this system of equations and shows that all stations have the same transmission probability in the steady state solution:

$$\tau_i = \tau_j \ \forall i, j \tag{10}$$

Note that the above result given by Theorem 1 is of particular importance since it guarantees the existence of a unique stable point of operation for the system. Indeed, while the existence of a unique point of operation can be easily guaranteed in a centralized system by imposing the same configuration for all stations, it is much harder to guarantee this in a distributed system in which each station chooses its own configuration.

Substituting $\tau_i = \tau$, given by Eq. (10), into Eqs. (7), (8) and (9) yields

$$p_{col} = 1 - (1 - \tau)^{n-1} \tag{11}$$

From the above equation, it follows that by setting the p_{col} parameter in our control system, we fix the *conditional collision probability under steady conditions*. In the following, we analyze how this parameter should be set in order to maximize the throughput of the WLAN.

The throughput obtained by a station in a saturated WLAN can be computed as follows

$$r = \frac{P_s l}{P_s T_s + P_c T_c + P_e T_e} \tag{12}$$

where l is the average packet length, and T_s, T_c and T_e are the duration of a success, a collision and an empty slot time, respectively, and P_s, P_c and P_e are the respective probabilities,

$$P_s = n\tau(1 - \tau)^{n-1} \tag{13}$$

$$P_e = (1 - \tau)^n \tag{14}$$

$$P_c = 1 - n\tau(1 - \tau)^{n-1} - (1 - \tau)^n \tag{15}$$

Following the analysis of [6], it can be seen that the total WLAN throughput is maximized with the following approximate expression for the optimal τ,

$$\tau_{opt} \approx \frac{1}{n}\sqrt{\frac{2T_e}{T_c}} \tag{16}$$

With the above τ_{opt}, the corresponding optimal conditional collision probability is equal to

$$p_{col} = 1 - (1 - \tau_{opt})^{n-1} = 1 - \left(1 - \frac{1}{n}\sqrt{\frac{2T_e}{T_c}}\right)^{n-1} \tag{17}$$

which can be approximated by

$$p_{col} \approx 1 - e^{-\sqrt{\frac{2T_e}{T_c}}} \tag{18}$$

[2]The theorems and their proofs are included in the Appendix.

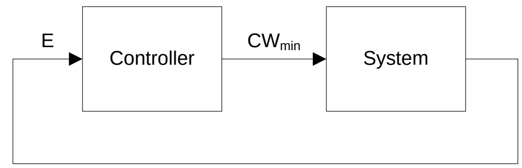

Fig. 2. Control system

From the above, we have that under optimal operation the conditional collision probability in the WLAN, p_{col}, is a constant independent of the number of stations. The fact that p_{col} is constant is a key result of our analysis, since it allows us to configure this parameter to a fixed value independent of the WLAN conditions.

V. Stability Analysis

We next conduct a stability analysis of DAC and, based on this analysis, we compute the configuration of the K_p and K_i parameters of the PI controller. The DAC system presented in Fig. 1 can be expressed in the form of Fig. 2, where

$$CW_{min} = \begin{pmatrix} CW_{min,1} \\ \vdots \\ CW_{min,n} \end{pmatrix} \tag{19}$$

and

$$E = \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} = \begin{pmatrix} 2p_{others,1} - p_{own,1} - p_{col} \\ \vdots \\ 2p_{others,n} - p_{own,n} - p_{col} \end{pmatrix} \tag{20}$$

Our control system consists of one PI controller in each station i that takes e_i as input and gives $CW_{min,i}$ as output. Following this, we can express the relationship between E and CW_{min} as follows

$$CW_{min}(z) = C \cdot E(z) \tag{21}$$

where

$$C = \begin{pmatrix} C_{PI}(z) & 0 & 0 & \dots & 0 \\ 0 & C_{PI}(z) & 0 & \dots & 0 \\ 0 & 0 & C_{PI}(z) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & C_{PI}(z) \end{pmatrix} \tag{22}$$

with $C_{PI}(z)$ being the z transform of a PI controller

$$C_{PI}(z) = K_p + \frac{K_i}{z - 1} \tag{23}$$

In order to analyze our system from a control theoretic standpoint, we need to characterize the Wireless LAN system with a transfer function that takes CW_{min} as input and gives the E as output.

Since the probabilities $p_{others,i}$ and $p_{own,i}$ are measured every 100 ms interval, we can assume that the obtained measurements correspond to stationary conditions and therefore the system does not have any memory. With this assumption,

E can be computed from the $CW_{min,i}$'s with Eq. (20), where $p_{own,i}$ and $p_{others,i}$ are computed as a function of the τ_i's following Eqs. (8) and (9).

Furthermore, the τ_i's can be calculated as a function of the $CW_{min,i}$'s from the following nonlinear equation [1],

$$\tau_i = \frac{2}{1 + CW_{min,i}(1 + p_{own,i} \sum_{k=0}^{m-1} (2p_{own,i})^k)} \tag{24}$$

where $p_{own,i}$ is a function of τ_i as given by Eq. (8).

The above equations give a nonlinear relationship between E and CW_{min}. In order to express this relationship as a transfer function, we linearize this relationship when the system suffers small perturbations around its stable point of operation. A similar approach was used in [16] to analyze RED from a control theoretical standpoint, although the analysis of [16] focused on a single-variable system while we analyze a multivariable system. In the following, we study the linearized model and force that it is stable. Note that the stability of the linearized model guarantees that our system is locally stable [16].

We express the perturbations around the point of operation as follows:

$$CW_{min,i} = CW_{min,i,opt} + \delta CW_{min,i} \tag{25}$$

where $CW_{min,i,opt}$ is the $CW_{min,i}$ value that yields the transmission probability τ_{opt} given by Eq. (16).

With the above, the perturbations suffered by E can be approximated by

$$\delta E = H \cdot \delta CW_{min} \tag{26}$$

where

$$H = \begin{pmatrix} \frac{\partial e_1}{\partial CW_{min,1}} & \frac{\partial e_1}{\partial CW_{min,2}} & \cdots & \frac{\partial e_1}{\partial CW_{min,n}} \\ \frac{\partial e_2}{\partial CW_{min,1}} & \frac{\partial e_2}{\partial CW_{min,2}} & \cdots & \frac{\partial e_2}{\partial CW_{min,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_n}{\partial CW_{min,1}} & \frac{\partial e_n}{\partial CW_{min,2}} & \cdots & \frac{\partial e_n}{\partial CW_{min,n}} \end{pmatrix} \tag{27}$$

The above partial derivatives can be computed as

$$\frac{\partial e_i}{\partial CW_{min,j}} = \frac{\partial e_i}{\partial \tau_j} \frac{\partial \tau_j}{\partial CW_{min,j}} \tag{28}$$

where from Eq. (24) we have

$$\frac{\partial \tau_j}{\partial CW_{min,j}} = -\tau_j^2 \frac{(1 + p_{own,j} \sum_{k=0}^{m} (2p_{own,j})^k)}{2} \tag{29}$$

which, evaluated at the stable point of operation, $p_{own,j} = p_{col}$ and $\tau_j = \tau_{opt}$, yields

$$\frac{\partial \tau_j}{\partial CW_{min,j}} = -\tau_{opt}^2 \frac{(1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k)}{2} \tag{30}$$

To compute $\partial e_i / \partial \tau_j$ for $j \neq i$ we proceed as follows

$$\frac{\partial e_i}{\partial \tau_j} = 2\frac{\partial p_{others,i}}{\partial \tau_j} - \frac{\partial p_{own,i}}{\partial \tau_j} \tag{31}$$

By calculating the two partial derivatives of the above equation and evaluating them at $\tau = \tau_{opt}$ we obtain

$$\frac{\partial p_{others,i}}{\partial \tau_j} = \frac{(n-2)(1-\tau_{opt})^{n-2}}{(n-1)} \tag{32}$$

and

$$\frac{\partial p_{own,i}}{\partial \tau_j} = (1-\tau_{opt})^{n-2} \tag{33}$$

From the above,

$$\frac{\partial e_i}{\partial \tau_j} = \frac{(n-3)(1-\tau_{opt})^{n-2}}{(n-1)} \tag{34}$$

Following a similar procedure we obtain

$$\frac{\partial e_i}{\partial \tau_i} = 2(1-\tau_{opt})^{n-2} \tag{35}$$

Combining all the above,

$$H = K_H \begin{pmatrix} 2 & \frac{n-3}{n-1} & \frac{n-3}{n-1} & \cdots & \frac{n-3}{n-1} \\ \frac{n-3}{n-1} & 2 & \frac{n-3}{n-1} & \cdots & \frac{n-3}{n-1} \\ \frac{n-3}{n-1} & \frac{n-3}{n-1} & 2 & \cdots & \frac{n-3}{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{n-3}{n-1} & \frac{n-3}{n-1} & \frac{n-3}{n-1} & \cdots & 2 \end{pmatrix} \tag{36}$$

where

$$K_H = -\tau_{opt}^2 (1-\tau_{opt})^{n-2} \frac{(1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k)}{2} \tag{37}$$

With the above, we have our system fully characterized by the matrices C and H. The next step is to configure the K_p and K_i parameters of this system. Following Theorem 2, we have that as long as the $\{K_p, K_i\}$ setting meets the following condition the system is guaranteed to be stable:

$$-(n-1)K_H(K_p - K_i) - 1 < (n-1)K_H(K_p - K_i) + 1 \tag{38}$$

In addition to guaranteeing stability, our goal in the configuration of the $\{K_p, K_i\}$ parameters is to find the right tradeoff between speed of reaction to changes and oscillations under transient conditions. To this aim, we use the *Ziegler-Nichols* rules [17], which have been designed for this purpose, 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:

$$K_p = 0.4K_u \tag{39}$$

and

$$K_i = \frac{K_p}{0.85T_i} \tag{40}$$

In order to compute K_u we proceed as follows. From Eq. (38) with $K_i = 0$ we have

$$K_p < \frac{1}{-(n-1)K_H} \tag{41}$$

Combining the above with Eq. (37) yields

$$K_p < \frac{2}{(n-1)\tau_{opt}^2 (1-\tau_{opt})^{n-2} (1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k)} \tag{42}$$

Since $p_{col} = 1 - (1-\tau_{opt})^{n-1} \approx (n-1)\tau_{opt}$, the above can be rewritten as

$$K_p < \frac{2}{p_{col}\tau_{opt}(1-\tau_{opt})^{n-2} (1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k)} \tag{43}$$

Since the above is a function of n (note that τ_{opt} depends on n) and we want to find an upper bound that is independent of n, we proceed as follows. From $p_{col} = 1 - (1 - \tau_{opt})^{n-1}$, we observe that τ_{opt} is never larger than p_{col} for $n > 1$[3]. Furthermore, we have $(1-\tau_{opt})^{n-2} < 1$. With these observations, we obtain the following constant upper bound (independent of n):

$$K_p < \frac{2}{p_{col}^2 \left(1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k\right)} \tag{44}$$

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

$$K_u = \frac{2}{p_{col}^2 \left(1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k\right)} \tag{45}$$

and set K_p according to Eq. (39),

$$K_p = \frac{0.4 \cdot 2}{p_{col}^2 \left(1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k\right)} \tag{46}$$

With the K_p value that makes the system become unstable, a given set of input values may change their sign up to every time slot, yielding an oscillation period of two slots ($T_i = 2$). Thus, from Eq. (40)

$$K_i = \frac{0.4}{0.85 p_{col}^2 \left(1 + p_{col} \sum_{k=0}^{m} (2p_{col})^k\right)} \tag{47}$$

which completes the configuration of the PI controller parameters. The stability of this configuration is guaranteed by Corollary 1.

VI. Performance Evaluation

In this section we evaluate DAC by conducting an extensive set of simulations under different traffic scenarios and compare its performance against the following approaches: the standard default configuration (DCF) [12], the static optimal configuration derived in [1] and several other adaptive algorithms, namely the *Enhanced 802.11* [8], *Idle Sense* [9] and the *Dynamic 802.11* [11]. Unlike these previous papers, which assume that all stations are saturated (i.e. they always have a packet ready for transmission), we analyze the saturated and non-saturated scenarios as well as the mixed one.

For the simulations, we have implemented our algorithm as well as the different existing proposals in OMNET++[4]. The physical layer parameters of IEEE 802.11g and a fixed payload size of 1000 Bytes have been used in all the experiments. For the obtained results, average and 95% confidence intervals are given.

A. Saturated scenario

First we evaluate the performance of DAC in a WLAN operating under saturation conditions. For this purpose, we compare the total throughput achieved by DAC for an increasing number of saturated stations n against the static optimal configuration, DCF and the other adaptive schemes.

Results are depicted in Fig. 3 (which includes a zoom in subplot). We observe from these results that DAC closely

[3] Note that for $n = 1$ the system is stable for any K_p.

[4] http://www.omnetpp.org/

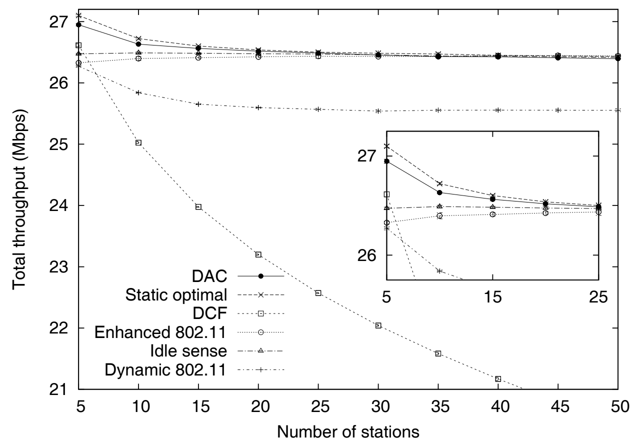

Fig. 3. Saturated scenario

follows the static optimal configuration for any n, slightly outperforms *Enhanced 802.11* and *Idle Sense* for a small number of active stations, and substantially outperforms *Dynamic 802.11* and DCF. Note that the static optimal configuration requires to know a priori the number of stations in the network, which challenges its practical use. Additionally, the other adaptive mechanisms introduce extra complexity and are not standard compliant, which makes them more difficult to deploy.

We conclude from the above that DAC achieves the objective of maximizing the total throughput in saturated conditions, without requiring to estimate the number of stations and avoiding complex and non-standard mechanisms.

B. Non-saturation scenario

We next analyze the behavior of the proposed algorithm in a non-saturated scenario where all stations send Poisson traffic with an average bit rate of 500 Kbps. Note that, in a non-saturated scenario, all stations see their throughput demands satisfied, and performance is given by delay.

Fig. 4 illustrates the average delay in the above scenario as a function of the number of stations. From the results, we observe that our proposal minimizes the average delay. It performs similarly to the other adaptive approaches, and outperforms the static optimal configuration (which is based on the assumption that all stations are saturated and thus enforces an overly large CW) and DCF (which uses a small fixed value of the CW_{min} which degrades performance for large n values).

We conclude that, in addition to maximizing the total throughput under saturation, DAC also minimizes the average delay under non-saturation.

C. Mixed scenario

We next address a mixed scenario in which some of the stations are saturated and some are not. In particular, we take half of the stations saturated and the other half sending Poisson traffic at an average bit rate of 500 Kbps.

In Fig. 5 we analyze the performance of our algorithm in terms of total throughput. We observe that DAC succeeds

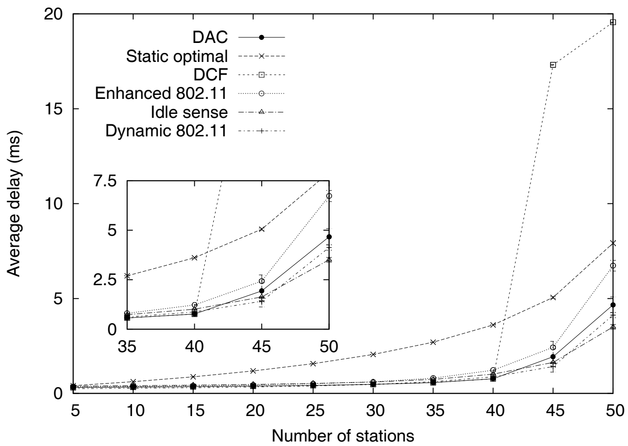

Fig. 4. Non-saturation scenario

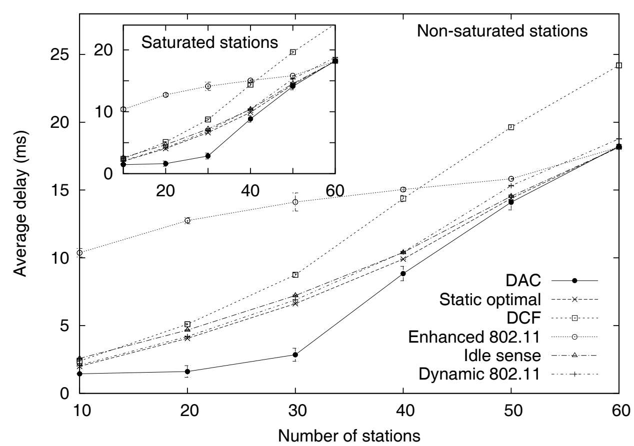

Fig. 6. Average delay under mixed traffic conditions

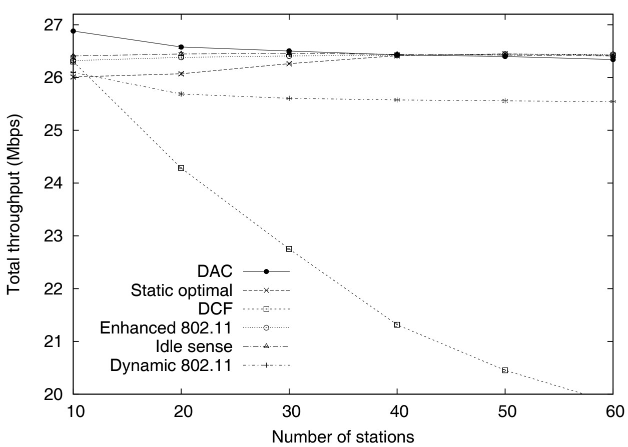

Fig. 5. Throughput performance under mixed traffic conditions

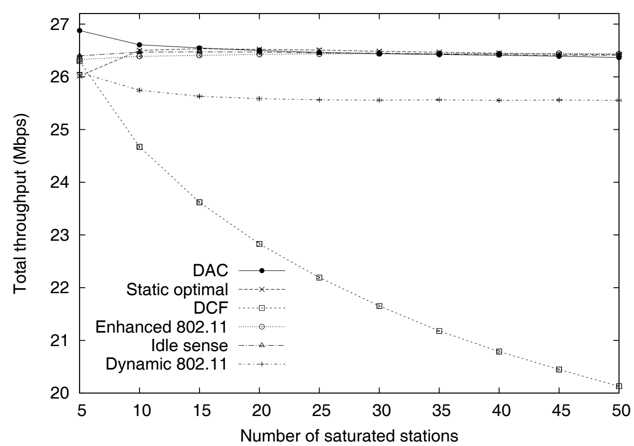

Fig. 7. Throughput performance of the mixed unbalanced scenario

in maximizing the throughput also for a mixed scenario, since it outperforms all other approaches and in particular it substantially outperforms the static optimal configuration.

In addition to the throughput evaluation, we also analyze the delay performance of DAC in the same scenario by measuring the average delay experienced by the non-saturated and saturated stations. Results are depicted in Fig. 6 (the delay of the saturated stations is given in a subplot). We can see from the figure that DAC substantially outperforms all the other approaches, since it provides the non-saturated stations with smaller delays without harming the delay performance of the saturated stations. The reason why our approach outperforms the other adaptive approaches is that, upon detecting congestion, the other approaches increase the CW of all stations (the saturated and the non-saturated ones), harming thus the delay performance of the non-saturated stations. In contrast, our algorithm is designed to increase only the CW of the saturated stations, which are the ones contributing to congestion.

We conclude from the above that DAC performs better than any other approach when saturated and non-saturated stations coexist in the WLAN, as it minimizes the delay performance of non-saturated station while neither harming the total throughput of the WLAN nor the delay of the saturated stations.

D. Mixed unbalanced scenario

In the previous experiment we had the same number of saturated and non-satured stations. In order to show the impact of having an unbalanced scenario with a different number of saturated and non-satured stations, we repeat the experiment for 5 non-saturated stations and a variable number of satured stations. Fig. 7 shows the resulting total throughput and Fig. 8 illustrates the average delay of non-saturated stations. We observe from these results that DAC outperforms all other approaches both in terms of throughput and delay also for this case.

E. Convergence

Our analysis guarantees that, after some transient, the CW_{min} of all stations converge towards a common value. In order to illustrate this behavior, we perform the following experiment. In a WLAN with 5 stations, one new station joins every 20 s until a total of 10 stations is reached. In this experiment, we analyze the CW_{min} of one of the initial stations as well as the CW_{min} of each one of the new stations joining. The results, depicted in Fig. 9, show that both the stations already present in the network and the new joining ones converge fast to the same CW_{min} value. Thus, this

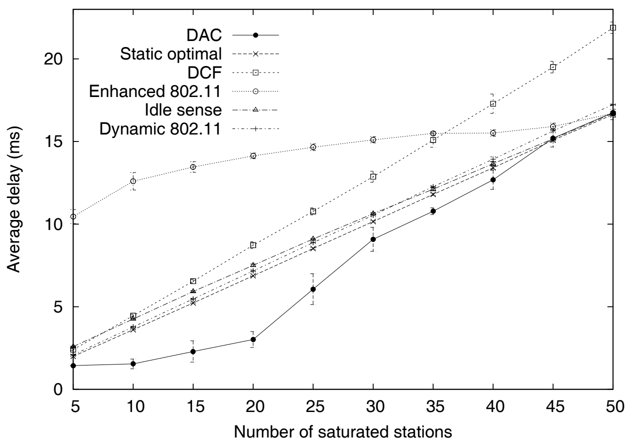

Fig. 8. Average delay of the mixed unbalanced scenario

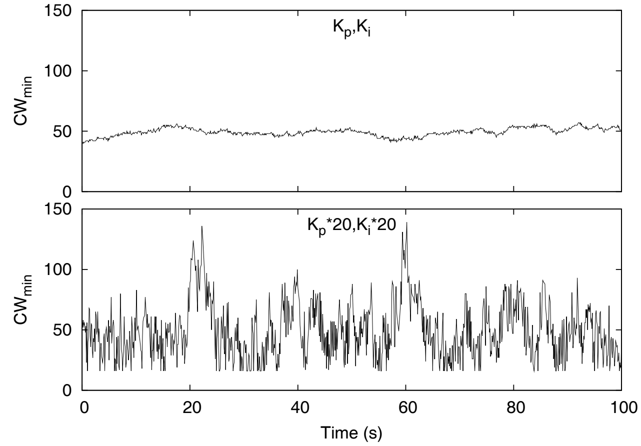

Fig. 10. Stability validation

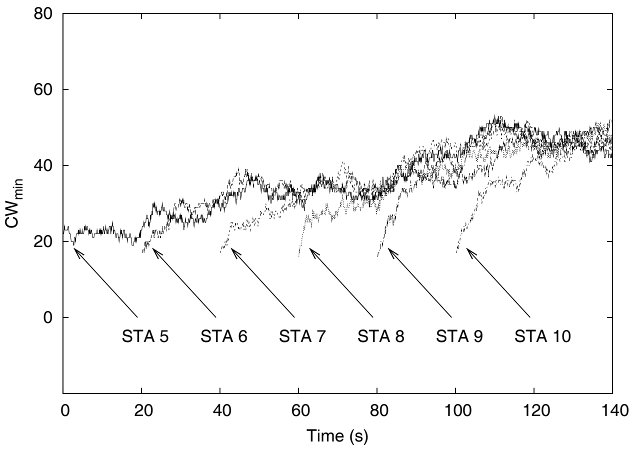

Fig. 9. Convergence

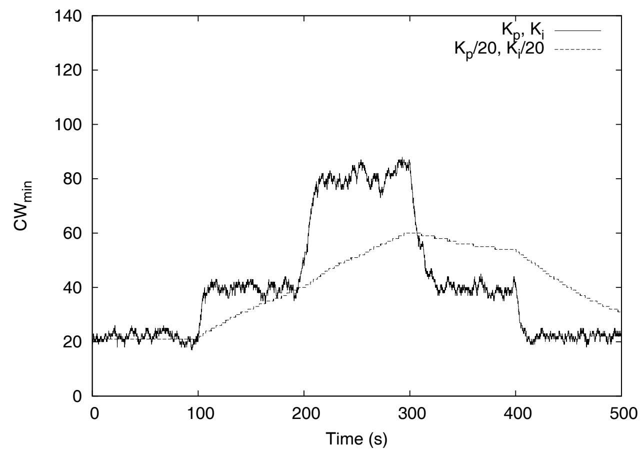

Fig. 11. Speed of reaction to changes

experiment confirms our theoretical result on the convergence of the proposed distributed algorithm.

F. Stability

The main objective in the configuration of the K_p and K_i parameters proposed in Section V is to achieve a proper tradeoff between stability and speed of reaction to changes. This objective is verified by the results presented in this and the following sections.

To validate that our system guarantees a stable behavior, we analyze the evolution in time of the control signal (CW_{min}) for our $\{K_p, K_i\}$ setting and a configuration with values of these parameters 20 times larger, in a network with 10 saturated stations. We observe form Fig. 10 that with the proposed configuration (label "K_p, K_i"), the CW_{min} only presents minor deviations around its stable point of operation, while if a larger setting is used (label "$K_p * 20, K_i * 20$"), the CW_{min} has a strong unstable behavior with drastic oscillations. We conclude that the proposed configuration achieves the objective of guaranteeing stability.

G. Speed of reaction to changes

In order to verify that our system has the ability to rapidly react to changes in the network, we conduct the following experiment. In a WLAN initially with 5 stations, 5 additional stations join the WLAN at time 100 s, and 5 more stations (yielding a total of 15) join 100 s afterwards. After additional 100 s, 5 stations leave the WLAN, and again 5 more stations leave, returning to the initial state with 5 stations. For this experiment, we examine the evolution over time of the CW_{min} used by one station of the initial group for our $\{K_p, K_i\}$ setting as well as for a smaller value of these parameters. From Fig. 11 we observe that with our setting (label "K_p, K_i"), the system reacts fast to the changes on the WLAN, as the CW_{min} reaches the new value almost immediately. In contrast, for a setting of these parameters 20 times smaller (label "$K_p/20, K_i/20$"), the system cannot keep up with the changes as CW_{min} reacts too slowly.

From this and the previous section, we conclude that the proposed setting of $\{K_p, K_i\}$ provides a good tradeoff between stability and speed of reaction, since with a larger setting the system suffers from instability and with a smaller one it reacts too slowly to changes.

H. Fairness

In Section VI-A we have evaluated the total throughput performance of our approach, but it is also relevant to analyze whether the total throughput is fairly shared among

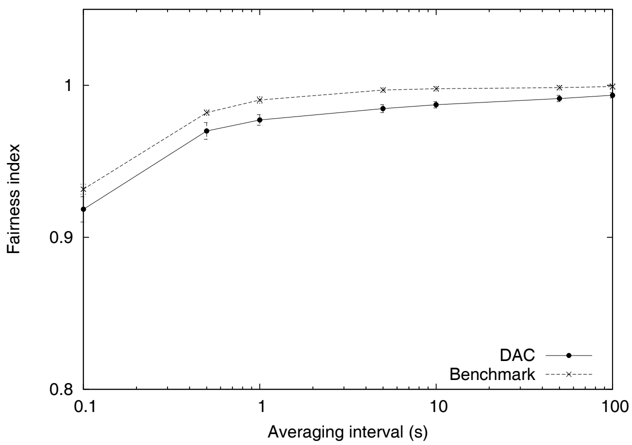

Fig. 12. Fairness

stations over short time scales and understand the impact of varying CW_{min} on fairness. Although our algorithm provides the same average CW_{min} to all stations over long time periods, at a given instant two stations may have slightly different CW_{min} values. In order to understand if this has any significant impact on short-term fairness we compare our approach against benchmark values. More specifically, we evaluate Jain's fairness index [18] over different averaging intervals for our approach and a configuration in which all stations use the same CW_{min}, whose value is equal to the average CW_{min} used by the adaptive algorithm.

The scenario consists of 10 stations always having a packet ready for transmission. The result of this experiment is depicted in Fig. 12. We conclude that our approach performs close to the benchmark configuration in terms of short-term fairness and the fairness index of DAC is close to 1 for reasonable periods of time.

VII. Implementation Experience

One of the key advantages of our algorithm over existing approaches is that it can be implemented with current off-the-shelf hardware. In order to prove this claim, we have deployed a small testbed consisting of two Linux-based laptops equipped with Atheros AR5212 cards operating in 802.11b mode. The implementation is based on kernel 2.6.24 and a modified version of the MadWifi v0.9.4 driver[5]. The adaptive algorithm runs as a user-space application and communicates with the driver by means of `IOCTL` calls.

The collision probability experienced by the neighboring nodes, p_{others}, is measured by a virtual device configured in promiscuous mode, which monitors the retry flag of all frames that belong to the same BSS and computes p_{others} every 100 ms by applying Eq. (5).

The collision probability observed by the station, p_{own}, is computed by gathering statistics from the device driver making `SIOCGATHSTATS IOCTL` requests every 100 ms and applying Eq. (4). More specifically, the driver provides detailed information about the total number of transmitted frames, the number of management frames and the number of retries within a `ath_stats` data structure.

[5]MadWifi project page, `http://madwifi-project.org/`

TABLE I
Validation of the implementation

Measured parameter	Implementation result	Simulation result
p_{meas} [%]	5.63 ± 0.75	6.27 ± 0.79
p_{own} [%]	5.91 ± 0.91	6.25 ± 0.80
Throughput [Mbps]	3.274 ± 0.086	3.278 ± 0.048

With the estimated values of p_{others} and p_{own}, CW_{min} is computed through Eq. (21). Finally, the CW_{min} parameter is updated every 100 ms by means of a private `IOCTL` call to the driver.

To validate the performance of our implementation we compared p_{others}, p_{own} and the achieved throughput against the values obtained by simulation. As shown in Table I, the values obtained in the real scenario closely follow the ones given by the simulation, which validates the implementation.

From the experiment reported in this section we conclude that the DAC algorithm can indeed be implemented with current devices, as it takes input data that can be easily obtained from the hardware and uses readily available primitives for the setting of the 802.11 parameters.

VIII. Related Work

In this section we provide a review of the related work in two areas, which are control theory techniques for 802.11 and distributed algorithms for WLANs, and we highlight the key differences between the previous works in these areas and ours.

Some previous papers in the literature have already used techniques from control theory to configure the 802.11 MAC parameters [6], [7], [10]. In [6], [7], control theoretic algorithms are proposed to optimally configure 802.11 nodes for maximizing the throughput and delay performance, respectively. However, these suffer the inherent limitations of centralized schemes. Furthermore, the analysis of a centralized control system differs very significantly from a distributed one as the former relies on a single CW for all stations (computed by a central entity) while in the latter system every station has its own CW.

The work in [10] proposes a distributed algorithm based on control theory to adaptively configure the Contention Window (CW) parameter of the stations. However, in this work the WLAN is modeled as a single variable system, and therefore the proposal only works as long as all stations use the same CW. This holds only when all stations simultaneously join the WLAN and change their CW in the same manner, but it does no longer hold if at a time instant a new station having a different CW joins. In contrast to [10], we model the WLAN as a multivariable system, where the CW of each station is a different variable and therefore CW's can take different values.

A number of papers have proposed distributed algorithms to optimize the WLAN performance [8]–[11]. A major drawback of these algorithms is that they require substantial modifications to the hardware of the existing wireless cards. The approaches of [8], [9] use as input of the algorithm low-level data which is currently not available in existing cards. The solutions proposed in [8]–[10] require modifying the CW on a per-packet basis, which is not achievable with existing hardware and brings substantial complexity. Furthermore, [11] modifies the contention algorithm of 802.11 which involves major hardware modifications.

In contrast to the above approaches, the mechanism proposed in this paper is compliant with the 802.11 standard [12] and can be implemented by existing cards with no modifications, as shown by the implementation experiences reported in Section VII. We argue that avoiding modifications to the standard 802.11 mechanism represents a major advantage, since this mechanism has been widely tested and its performance is well understood.

A major conceptual difference between existing distributed algorithms [8]–[11] and ours is the following. With the existing algorithms, each station configures its parameters based on the overall level of congestion observed in the WLAN *independently of how much the station is contributing to the overall congestion*. As a result, in case of congestion, a station that is not contributing to congestion increases nonetheless its CW and therefore sees its delay performance *unnecessarily penalized*. In contrast to this behavior, with our algorithm each station measures its own contribution to congestion, and adjusts the CW based on this contribution, yielding thus a better delay performance.

IX. Summary and Conclusions

In this paper we have proposed a distributed adaptive algorithm to optimally configure IEEE 802.11 networks. The key advantages of the proposed algorithm over existing approaches are: $i)$ the proposed algorithm is sustained by mathematical foundations that guarantee optimal performance, convergence and stability, $ii)$ the mechanism is standard-compliant and can be implemented with existing hardware, and $iii)$ it outperforms previous approaches in terms of throughput and delay.

The proposed algorithm executes an independent PI controller in each station that takes as input the measured error signal and gives as output the station's configuration. The error signal has been carefully chosen to ensure that $i)$ the stable point of operation gives optimal throughput performance, and $ii)$ when the WLAN operates at any other point, the error signal is large thus forcing the WLAN to quickly converge to the stable point.

The error signal is obtained by subtracting the reference signal from the feedback signal. We have taken as reference signal the optimal conditional collision probability. To compute this value, we have conducted a steady-state analysis of the WLAN. As a result of this analysis, we have shown that the optimal collision probability is a constant independent of the number of stations. This is a key result since a fundamental requirement when building a control system is to have a constant reference signal.

In order to configure the parameters of the PI controller, we have conducted a control theoretic analysis of our system. As the system relies on a number of independent variables (namely the configuration of each station), the analysis has been based on multivariable control theory. From this analysis, we have first obtained the stability region of the parameter values, and then we have chosen a configuration within this stability region that provides a proper tradeoff between stability and reaction to changes.

The performance of the proposed algorithm has been extensively evaluated by means of simulations. Results have shown that $i)$ our scheme substantially outperforms DCF in terms of throughput, $ii)$ it performs better than the static optimal configuration when not all stations are saturated, and $iii)$ it outperforms other distributed adaptive approaches in terms of delay. The approach has also been validated by means of a real prototype, which has proved that the scheme can be implemented with current hardware.

References

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

[2] A. Banchs and L. Vollero, "Throughput Analysis and Optimal Configuration of 802.11e EDCA," *Computer Networks*, vol. 50, no. 11, August 2006.

[3] IEEE 802.11 WG, *Information Technology - Telecommun. 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*. Standard, IEEE, August 1999.

[4] A. Nafaa, A. Ksentini, A. A. Mehaoua, B. Ishibashi, Y. Iraqi, and R. Boutaba, "Sliding Contention Window (SCW): Towards Backoff Range-Based Service Differentiation over IEEE 802.11 Wireless LAN Networks," *IEEE Network*, vol. 19, no. 4, pp. 45–51, July 2005.

[5] J. Freitag, N. L. S. da Fonseca, and J. F. de Rezende, "Tuning of 802.11e Network Parameters," *IEEE Communications Letters*, vol. 10, no. 8, pp. 611–613, August 2006.

[6] P. Patras, A. Banchs, and P. Serrano, "A Control Theoretic Approach for Throughput Optimization in IEEE 802.11e EDCA WLANs," *Mobile Networks and Applications (MONET)*, published online December 2008.

[7] A. Banchs *et al.*, "A Control Theoretic Scheme for Efficient Video Transmission over IEEE 802.11e EDCA WLANs," *submitted*, 2009.

[8] G. Bianchi, L. L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in *Proceedings of PIMRC96*, Taipei, Taiwan, October 1996.

[9] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans," in *Proceedings of SIGCOMM '05*. New York, NY, USA: ACM, 2005, pp. 121–132.

[10] Q. Xia and M. Hamdi, "Contention Window Adjustment for IEEE 802.11 WLANs: A Control-Theoretic Approach," in *Proceedings of IEEE ICC 2006*, Istanbul, Turkey, June 2006.

[11] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 9, Sep 2000.

[12] IEEE 802.11 WG, *Information Technology - Telecommunications and Information Exchange between Systems. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE 802.11-2007, Revision of Std. 802.11-1999, 2007.

[13] ——, *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, November 2005.

[14] T. Glad and L. Ljung, *Control theory: multivariable and nonlinear methods*. Taylor & Francis, 2000.

[15] K. Aström and R. M. Murray, *Feedback Systems*. Princeton University Press, 2008.

[16] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A Control Theoretic Analysis of RED," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[17] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 2nd ed. Addison-Wesley, 1990.
[18] R. Jain, Chiu, D.M., and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems," *DEC Research Report TR-301*, 1984.
[19] K. Aström and B. Wittenmark, *Computer-controlled systems, theory and design*, 2nd ed. Prentice Hall International Editions, 1990.

Appendix

Theorem 1. *The system of equations defined by (7) has a unique solution that satisfies $\tau_i = \tau_j\ \forall i, j$.*

Proof: From Eq. (7) we have

$$2p_{others,i} - p_{own,i} - p_{col} = 0 \tag{48}$$

which following Section IV can be rewritten as

$$2\sum_{k\neq i} \frac{\tau_k}{\sum_{l\neq i}\tau_l} p_{own,k} - p_{own,i} - p_{col} = 0 \tag{49}$$

From Eq. (48) we have

$$2p_{others,i} - p_{own,i} - p_{col} - \frac{\sum_{k\neq i}\tau_l}{\sum_{k\neq j}\tau_l}\left(2p_{others,j} - p_{own,j} - p_{col}\right) = 0 \tag{50}$$

Applying Eq. (49) to the above yields

$$\frac{2\tau_j}{\sum_{k\neq i}\tau_k} p_{own,j} + \frac{\sum_{k\neq j}\tau_k}{\sum_{k\neq i}\tau_k} p_{own,j} - \frac{2\tau_i}{\sum_{k\neq i}\tau_k} p_{own,i} - p_{own,i} - p_{col} + \frac{\sum_{k\neq j}\tau_k}{\sum_{k\neq i}\tau_k} p_{col} = 0 \tag{51}$$

from where

$$\left(\tau_j + \textstyle\sum_k \tau_k\right) p_{own,j} - \left(\tau_i + \textstyle\sum_k \tau_k\right) p_{own,i} + (\tau_j - \tau_i) p_{col} = 0 \tag{52}$$

Substituting the expressions of $p_{own,j}$ and $p_{own,i}$ by Eq. (8) and operating on the above yields

$$(\tau_j - \tau_i)\left(1 - \sum_k \tau_k \prod_{k\neq i,j} 1 - \tau_k - \prod_{k\neq i,j} 1 - \tau_k - p_{col}\right) = 0 \tag{53}$$

Note that Eq. (52) can be rewritten as

$$\left(\tau_j + \textstyle\sum_k \tau_k\right)\left(p_{own,j} - p_{col}\right) - \left(\tau_i + \textstyle\sum_k \tau_k\right)\left(p_{own,i} - p_{col}\right) = 0 \tag{54}$$

from where $p_j \le p_{col} \le p_i$ or $p_i \le p_{col} \le p_j$, which forces that either $p_{col} \ge 1 - \prod_{k\neq i} 1 - \tau_k$ or $p_{col} \ge 1 - \prod_{k\neq j} 1 - \tau_k$. This leads to

$$p_{col} > 1 - \prod_{k\neq i,j} 1 - \tau_k \tag{55}$$

Combining the above with Eq. (52), we have the the second term of Eq. (52) is surely negative, which forces the first term to be 0. Thus,

$$\tau_i = \tau_j \tag{56}$$

which proves the second part of the theorem.

To proof uniqueness of the solution, we proceed as follows. From the above we have

$$\tau_i = \tau\ \forall i \tag{57}$$

Substituting this into Eq. (48) yields

$$(1-\tau)^{n-1} = 1 - p_{col} \tag{58}$$

Since the lhs of the above equation decreases from 1 to 0 with τ while the rhs is a constant between 0 and 1, we have that there exists a unique τ value that resolves the above equation. From Eq. (57) it further follows that the only solution to the system is $\tau_i = \tau\ \forall i$. The proof follows. ■

Theorem 2. *The system is guaranteed to be stable as long as K_p and K_i meet the following condition:*

$$-(n-1)K_H(K_p - K_i) - 1 < (n-1)K_H(K_p - K_i) + 1 \tag{59}$$

Proof: According to (6.22) of [14], we need to check that the following transfer function is stable

$$(I + CH)^{-1}C \tag{60}$$

Computing the above matrix yields

$$(I + CH)^{-1}C = \begin{pmatrix} a & b & b & \dots & b \\ b & a & b & \dots & b \\ b & b & a & \dots & b \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b & b & b & \dots & a \end{pmatrix} \tag{61}$$

where

$$a = \frac{C_{PI}(z)}{n}\left(\frac{1}{1 + (n-1)K_H C_{PI}(z)} + \frac{n-1}{1 + \left(2 - \frac{n-3}{n-1}\right) K_H C_{PI}(z)}\right) \tag{62}$$

and

$$b = \frac{C_{PI}(z)}{n}\left(\frac{1}{1 + (n-1)K_H C_{PI}(z)} - \frac{1}{1 + \left(2 - \frac{n-3}{n-1}\right) K_H C_{PI}(z)}\right) \tag{63}$$

Rearranging terms in a and b we obtain

$$a = \frac{P_1(z)}{(z^2 + a_1 z + a_2)(z^2 + a_1' z + a_2')} \tag{64}$$

and

$$b = \frac{P_2(z)}{(z^2 + a_1 z + a_2)(z^2 + a_1' z + a_2')} \tag{65}$$

where $P_1(z)$ and $P_2(z)$ are polynomials and

$$a_1 = -(n-1)K_H K_p - 1 \tag{66}$$

$$a_2 = (n-1)K_H(K_p - K_i) \tag{67}$$

$$a_1' = -\left(2 - \frac{n-3}{n-1}\right) K_H K_p - 1 \tag{68}$$

$$a_2' = \left(2 - \frac{n-3}{n-1}\right) K_H (K_p - K_i) \tag{69}$$

According to Theorem 3.5 of [14], a sufficient condition for the stability of a transfer function is that the zeros of its pole polynomial (which is the least common denominator of all the minors of the transfer function matrix) fall within the unit circle. Applying this theorem to $(I+CH)^{-1}C$ yields that the roots of the polynomials $z^2+a_1z+a_2$ and $z^2+a_1'z+a_2'$ have to fall inside the unit circle. This can be ensured by choosing coefficients $\{a_1, a_2\}$ and $\{a_1', a_2'\}$ that belong to the stability triangle [19]:

$$a_2 < 1 \tag{70}$$

$$a_1 < a_2 + 1 \tag{71}$$

$$a_1 > -1 - a_2 \tag{72}$$

and

$$a_2' < 1 \tag{73}$$

$$a_1' < a_2' + 1 \tag{74}$$

$$a_1' > -1 - a_2' \tag{75}$$

Eqs. (70), (72), (73) and (75) are satisfied for any $\{K_p, K_i\}$ setting. If Eq. (71) is satisfied, then Eq. (74) is also satisfied. Therefore, it is enough to guarantee that Eq. (71) is met. The proof follows. ■

Corollary 1. *The K_p and K_i configuration given by Eqs. (46) and (47) is stable.*

Proof: It is easy to see that Eqs. (46) and (47) meet the condition of Theorem 2. ■