A Control Theoretic Approach for Throughput Optimization in IEEE 802.11e EDCA WLANs

Paul Patras · Albert Banchs · Pablo Serrano

Abstract The MAC layer of the 802.11 standard, based on the CSMA/CA mechanism, specifies a set of parameters to control the aggressiveness of stations when trying to access the channel. However, these parameters are statically set independently of the conditions of the WLAN (e.g. the number of contending stations), leading to poor performance for most scenarios. To overcome this limitation previous work proposes to adapt the value of one of those parameters, namely the CW, based on an estimation of the conditions of the WLAN. However, these approaches suffer from two major drawbacks: *i*) they require extending the capabilities of standard devices or *ii*) are based on heuristics. In this paper we propose a control theoretic approach to adapt the CW to the conditions of the WLAN, based on an analytical model of its operation, that is fully compliant with the 802.11e standard. We use a Proportional Integrator controller in order to drive the WLAN to its optimal point of operation and perform a theoretic analysis to determine its configuration. We show by means of an exhaustive performance evaluation that our algorithm maximizes the total throughput of the WLAN and substantially outperforms previous standard-compliant proposals.

P. Patras (✉)
IMDEA Networks, Avenida del Mar Mediterráneo,
22, 28918 Leganés, Madrid, Spain
e-mail: paul.patras@imdea.org

P. Patras · A. Banchs · P. Serrano
Universidad Carlos III de Madrid,
Leganés, Madrid, Spain

1 Introduction

The CSMA/CA mechanism used in IEEE 802.11 WLANs is based upon a set of parameters that controls the way stations access the channel. In particular, the Contention Window (CW) parameter controls the probability that a station defers or transmits a frame once the medium has become idle.

The CW configuration used by the 802.11 standard [1] is statically set, independently of the number of contending stations. This static configuration leads to poor performance in most scenarios. In particular, when there are many stations in the WLAN it would be desirable to have larger CWs in order to avoid too frequent collisions, while with few stations smaller CWs would help to reduce the channel idle time. Following this, it has been shown that for given a number of actively contending stations there exists an optimal CW configuration that maximizes throughput performance [2, 3].

Following the above observations, many authors have proposed to dynamically adapt the CW by estimating the number of active stations in the WLAN. These works can be classified in two different groups:

1. Distributed approaches [4–13], that require every node on the WLAN to implement a mechanism for adjusting the backoff behavior. The main disadvantage of these approaches is that they change the rules of the 802.11 standard and require introducing modifications to the existing hardware.

2. Centralized approaches [14, 15], based on a single node that periodically distributes the set of MAC layer parameters to be used by every station. These approaches are compatible with the 802.11e standard. However, because they are based on heuristic algorithms and lack analytical support, they do not guarantee optimal performance.

In this paper we propose a novel adaptive algorithm to dynamically adjust the CW configuration of 802.11-based Wireless LANs. We share the same goal with previous approaches, i.e., to maximize the overall throughput performance of the wireless network. Compared to the existing schemes our proposal benefits from the following key improvements:

– It is fully compatible with the 802.11e standard and does not require any modification to existing hardware, since the dynamic adjustment is based only on observing successfully received frames at the Access Point (AP).
– It is based on a well established scheme from discrete-time control theory, namely the Proportional Integrator (PI). We optimally tune the parameters of the PI controller by conducting a control theoretic analysis of the system.

The rest of the paper is structured as follows. In Section 2 we briefly describe the EDCA mechanism of the IEEE 802.11e standard. In Section 3 we analyze the throughput performance of EDCA and find the collision probability for which this is optimized. In Section 4 we present the proposed algorithm, which aims at driving the system to the optimal collision probability obtained in the previous section by using a PI controller. The parameters of the controller are set following a control theoretical analysis of our system. The performance of the proposed scheme is validated by means of simulation experiments in Section 5. Finally, Section 6 concludes the paper.

2 IEEE 802.11e EDCA

This section briefly summarizes the EDCA mechanism. This mechanism has been defined in the 802.11e standard [16] and will be included in the ongoing new revision of the 802.11 standard [17].

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 them. Once a station becomes active, each CAF executes an independent backoff process to transmit its frames.

A station with a new frame to transmit monitors the channel activity. If the medium is idle for a period of time equal to the arbitration interframe space parameter ($AIFS$), the CAF transmits. Otherwise, if the channel is sensed busy (either immediately or during the $AIFS$ period), the CAF continues to monitor the channel until it is measured idle for an $AIFS$ time, and, at this point, the backoff process starts.

Upon starting the backoff process, a random value uniformly distributed in the range $(0, CW - 1)$ is chosen and the backoff time counter is initialized with this number. The CW value is called the contention window, and depends on the number of failed transmissions of a frame. At the first transmission attempt, CW is set equal to the minimum contention window parameter (CW_{min}).

As long as the channel is sensed idle, the backoff time counter is decremented once every empty slot time T_e. When a transmission is detected on the channel the backoff time counter is "frozen", and reactivated again after the channel is sensed idle for a certain period. This period is equal to $AIFS$ if the transmission is received with a correct FCS, and $EIFS - DIFS + AIFS$ otherwise, where $EIFS$ (the extended interframe space) and $DIFS$ (the distributed interframe space) are physical layer constants.

As soon as the backoff time counter reaches zero, the CAF transmits its frame. A collision occurs when two or more CAFs start transmitting simultaneously. An acknowledgement (Ack) frame is used to notify the transmitting station that the frame has been successfully received. The Ack is immediately sent upon the reception of the frame, after a period of time equal to the physical layer constant SIFS (the short interframe space).

If the Ack is not received within a time interval given by the *Ack_Timeout* physical layer constant, the CAF assumes that the frame was not received successfully. The transmission is then rescheduled by reentering the backoff process, which starts at an $AIFS$ time following the timeout expiry. After each unsuccessful transmission, CW is doubled, up to a maximum value given by the CW_{max} parameter. If the number of failed attempts reaches a predetermined retry limit R, the frame is discarded.

After a (successful or unsuccessful) frame transmission, before sending 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. In this situation, the station is allowed

to send a new frame a SIFS period after the completion of the previous one. The period of time a CAF is allowed to retain the right to access the channel is limited by the transmission opportunity limit parameter ($TXOP\_limit$).

In the case of a single station running more than one channel access function, if the backoff time counters of two or more CAFs 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* 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_{min}, CW_{max}, $AIFS$ and $TXOP\_limit$. These are configurable parameters that can be set to different values for different CAFs. The CAFs are grouped by Access Categories (ACs), all the CAFs of an AC having the same configuration. The Access Point (AP) announces periodically (every 100 ms) the parameters of each AC by means of beacon frames.

In this paper, our goal is to find the EDCA parameters that maximize the throughput of the WLAN, while fairly sharing the bandwidth among the competing stations. Following this goal, we use the following configuration for the stations:[1]

– Each station executes a single CAF and transmits one frame upon accessing the channel.
– The $AIFS$ parameter is set to the minimum value (DIFS) for all stations.
– All stations contend with the same CW_{min} and CW_{max} parameters.

The rest of the paper is devoted to the design of an adaptive algorithm that adjusts the configuration of CW_{min} and CW_{max} with the goal of maximizing the overall WLAN throughput. This algorithm is executed at the AP, which uses beacon frames to announce the computed CW_{min} and CW_{max} values to the stations.

3 Throughput analysis and optimization

In this section we present a throughput analysis of an EDCA WLAN configured according to the rules given in the previous section. Based on this analysis, we find the collision probability of an optimally configured WLAN, which is the basis of the algorithm presented in the following section.

We start by analyzing the case when all stations are saturated[2] and consider later the case when some stations are not saturated. Let us define τ as the probability that a saturated station transmits at a randomly chosen slot time. This can be computed according to [2] as follows:

$$\tau = \frac{2}{1 + W + pW \sum_{i=0}^{m-1} (2p)^i} \qquad (1)$$

where $W = CW_{min}$, m is the maximum backoff stage ($CW_{max} = 2^m CW_{min}$) and p is the probability that a transmission collides. In a WLAN with n stations,

$$p = 1 - (1 - \tau)^{n-1} \qquad (2)$$

The throughput obtained by a station can be computed as follows

$$r = \frac{P_s l}{P_s T_s + P_c T_c + P_e T_e} \qquad (3)$$

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

The probabilities P_s, P_c and P_e are computed as

$$P_s = n\tau(1 - \tau)^{n-1} \qquad (4)$$

$$P_e = (1 - \tau)^n \qquad (5)$$

$$P_c = 1 - n\tau(1 - \tau)^{n-1} - (1 - \tau)^n \qquad (6)$$

and the slot time durations T_s and T_c as

$$T_s = T_{PLCP} + \frac{H}{C} + \frac{l}{C} + SIFS + T_{PLCP}$$
$$+ \frac{Ack}{C} + DIFS \qquad (7)$$

$$T_c = T_{PLCP} + \frac{H}{C} + \frac{l}{C} + DIFS \qquad (8)$$

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

---

[1]The reader is referred to [18] for a detailed justification of these configuration choices.

[2]Following [2], by saturation we mean that a station always has a packet ready for transmission.

The above terminates our throughput analysis. We next address, based on this analysis, the issue of optimizing the throughput performance of the WLAN. To this aim, we can rearrange Eq. 3 to obtain

$$r = \frac{l}{T_s - T_c + \frac{P_e(T_e - T_c) + T_c}{P_s}} \quad (9)$$

As l, T_s, and T_c are constants, maximizing the following expression will result in the maximization of r,

$$\hat{r} = \frac{P_s}{P_e(T_e - T_c) + T_c} \quad (10)$$

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

$$\hat{r} = \frac{n\tau - n(n-1)\tau^2}{T_e - n(T_e - T_c)\tau + \frac{n(n-1)}{2}(T_e - T_c)\tau^2} \quad (11)$$

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

$$\left. \frac{d\hat{r}}{d\tau} \right|_{\tau = \tau_{opt}} = 0 \quad (12)$$

which neglecting the terms of higher order than 2 yields

$$a\tau^2 + b\tau + c = 0 \quad (13)$$

where

$$a = -\frac{n^2(n-1)}{2}(T_c - T_e) \quad (14)$$

$$b = -2n(n-1)T_e \quad (15)$$

$$c = nT_e \quad (16)$$

Isolating τ_{opt} from the above yields

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

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

$$\tau_{opt} \approx \frac{1}{n}\sqrt{\frac{2T_e}{T_c}} \quad (18)$$

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

$$p_{opt} = 1 - (1 - \tau_{opt})^{n-1} = 1 - \left(1 - \frac{1}{n}\sqrt{\frac{2T_e}{T_c}}\right)^{n-1} \quad (19)$$

which can be approximated by

$$p_{opt} \approx 1 - e^{-\sqrt{\frac{2T_e}{T_c}}} \quad (20)$$

This implies that, under optimal operation with saturated stations, the collision probability in the WLAN is a constant independent of the number of stations. The key approximation of this paper is to assume that, when some of the stations are saturated and some are not, the optimal collision of the WLAN takes the same constant value.

In the following section we design an adaptive algorithm that adjusts the WLAN configuration with the goal of driving the collision probability to the above value. Note that, since this a constant value, our algorithm does not need to know the number of stations in the WLAN.

4 Adaptive algorithm

We next present our adaptive algorithm; this algorithm runs at the AP and consists of the following two steps which are executed iteratively:

– During the period between two beacon frames (which lasts 100 ms), the AP measures the collision probability of the WLAN resulting from the current CW configuration.
– At the end of this period, the AP computes the new CW configuration based on the measured collision probability and distributes it to the stations in a new beacon frame.

Our algorithm uses a PI controller to drive the WLAN to its optimal point of operation. In the following, we explain how the CW configuration is adjusted using a control signal. We then describe our system from a control theoretical standpoint. Next, we analyze our system by linearizing the behavior of the WLAN. Finally, we use this analysis to adequately configure the parameters of the PI controller.

4.1 CW configuration

Following the previous section, our goal is to adjust the *CW* parameters of EDCA (CW_{min} and CW_{max}) in order to force the collision probability given by Eq. 20.

Since the default CW values given by the 802.11e standard[3] ($CW_{min}^{default}$ and $CW_{max}^{default}$) are typically too small, yielding a too aggressive behavior, in order to achieve optimal operation these CW parameters should be increased.

Following the above reasoning, our algorithm increases the default CW_{min} of the standard by some CW_{offset},

$$CW_{min} = CW_{min}^{default} + CW_{offset} \qquad (21)$$

while keeping the default value for the maximum back-off stage, i.e.

$$CW_{max} = 2^m CW_{min} \qquad (22)$$

where m is the maximum backoff stage of the default configuration.

In order to ensure that our algorithm never underperforms the standard default configuration by using overly small CW values, we force that CW_{offset} cannot take negative values, which guarantees that CW_{min} will never take smaller values than the standard's default. In addition, we also force that CW_{offset} cannot take values that yield a CW_{min} larger than $CW_{max}^{default}$. These bounds provide a safeguard against too large and too small values of CW_{min}, respectively. In the rest of the paper we assume that CW_{offset} always takes values within these bounds and do not further consider this effect.

4.2 Control system

From a control theoretic standpoint, our system can be seen as the composition of the two modules depicted in Fig. 1: the *controller* C(z), which is the adaptive algorithm that controls the WLAN, and the *controlled system* H(z), which is the WLAN itself. In our proposal we use for the controller module a classical scheme from discrete-time control theory, namely the *Proportional Integrator* (PI) *Controller*.

Following the above, our control system consists of the following two modules:

– The controller module located at the AP, that is based on the *Proportional Integrator* (PI) controller. The AP estimates the collision probability and provides it to the controller, which takes as input the difference between the estimated collision probability and its desired value as given by
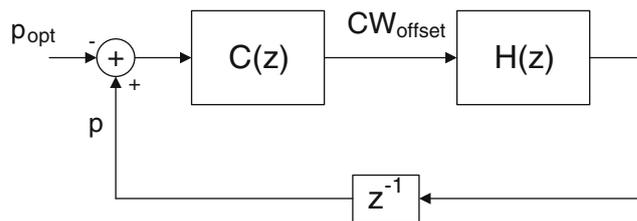

Figure 1 Control system

Eq. 20. With this input, the controller computes the CW_{offset} value.

– The controlled module is the 802.11e EDCA WLAN system. As specified by the standard, the AP distributes the new CW configuration to the stations every 100 ms. This configuration is obtained from the CW_{offset} value given by the controller, following Eqs. 21 and 22.

The estimation of the collision probability over a 100 ms period is performed at the AP as follows. Let S be the number of frames received by the AP during this period with the retry bit unset, and R be the number of frames received with the retry bit set. Then, if we assume that no frames are discarded due to reaching the retry limit, the collision probability p can be computed as

$$p = \frac{R}{R + S} \qquad (23)$$

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

Note that with the above method, the AP can compute the probability p by simply analyzing the header of the frames successfully received, which can be easily done with no modifications to the AP's hardware and driver.

4.3 Transfer function characterization

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_{offset} as input and gives the collision probability p as output. Since the collision probability is measured every 100 ms interval, we can safely assume that the obtained measurement corresponds to stationary conditions and therefore the system does not have any memory. With this assumption,

$$p = 1 - (1 - \tau)^{n-1} \qquad (24)$$

---

[3]Although the 802.11e parameters are configurable, the standard includes a default setting for these parameters [16].

where τ is a function of CW_{offset} as given by Eq. 1,

$$\tau = \frac{2}{1 + \left(CW_{min}^{default} + CW_{offset}\right)\left(1 + p\sum_{i=0}^{m-1}(2p)^i\right)} \tag{25}$$

The above equations give a nonlinear relationship between p and CW_{offset}. In order to express this relationship as a transfer function, we linearize this relationship when the system is perturbed around its stable point of operation,[4] i.e.

$$CW_{offset} = CW_{offset,opt} + \delta CW_{offset} \tag{26}$$

where $CW_{offset,opt}$ is the CW_{offset} value that yields the optimal collision probability p_{opt} computed in Eq. 20.

With the above, the oscillations of the collision probability around its point of operation p_{opt} can be approximated by

$$p \approx p_{opt} + \frac{\partial p}{\partial CW_{offset}}\delta CW_{offset} \tag{27}$$

The above partial derivative can be computed as

$$\frac{\partial p}{\partial CW_{offset}} = \frac{\partial p}{\partial \tau}\frac{\partial \tau}{\partial CW_{offset}} \tag{28}$$

where

$$\frac{\partial p}{\partial \tau} \approx n - 1 \tag{29}$$

and

$$\frac{\partial \tau}{\partial CW_{offset}} = -\frac{2\left(1 + p\sum_{i=0}^{m-1}(2p)^i\right)}{\left(1 + CW_{min}\left(1 + p\sum_{i=0}^{m-1}(2p)^i\right)\right)^2} \tag{30}$$

Evaluating the partial derivative at the stable point of operation $p = p_{opt}$, and using the approximation $p_{opt} \approx (n-1)\tau_{opt}$ given by Eq. 19 and the expression for τ_{opt} given by Eq. 1, yields

$$\frac{\partial p}{\partial CW_{offset}} \approx -p_{opt}\tau_{opt}\frac{1 + p_{opt}\sum_{i=0}^{m-1}(2p_{opt})^i}{2} \tag{31}$$

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

$$\delta P(z) = H(z)\,\delta CW_{offset}(z) \tag{32}$$

we obtain from Eqs. 27 and 31 the following expression for the transfer function,

$$H(z) = -p_{opt}\tau_{opt}\frac{1 + p_{opt}\sum_{i=0}^{m-1}(2p_{opt})^i}{2} \tag{33}$$

Figure 2 illustrates the above linearized model when working around its stable operation point:

$$\begin{cases} p = p_{opt} + \delta p \\ CW_{offset} = CW_{offset,opt} + \delta CW_{offset} \end{cases} \tag{34}$$

Note that, as compared to the model of Fig. 1, in Fig. 2 only the perturbations around the stable operation point are considered.

4.4 Controller configuration

We next address the issue of configuring the PI controller. The transfer function of the controller is given by

$$C(z) = K_p + \frac{K_i}{z - 1} \tag{35}$$

We observe from the above transfer function that the PI controller depends on the following two parameters to be configured: K_p and K_i. Our goal in the configuration of these parameters is to find the right tradeoff between speed of reaction to changes and stability. To this aim, we use the Ziegler-Nichols rules [20] which have been designed for this purpose. These rules are applied 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
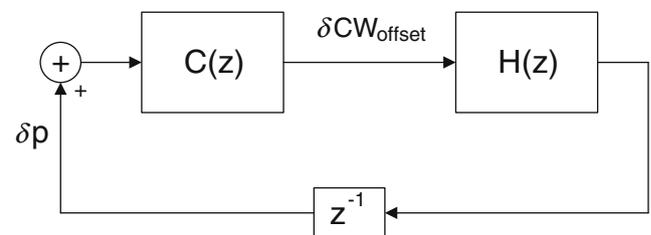
---

[4]A similar approach was used in [19] to analyze RED from a control theoretical standpoint.

Figure 2 Linearized system

period under these conditions. Then, K_p and K_i are configured as follows:

$$K_p = 0.4 K_u \tag{36}$$

and

$$K_i = \frac{K_p}{0.85 T_i} \tag{37}$$

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

$$|H(z)C(z)| = K_p p_{opt} \tau_{opt} \frac{1 + p_{opt} \sum_{i=0}^{m-1} (2p_{opt})^i}{2} < 1 \tag{38}$$

which yields the following upper bound for K_p,

$$K_p < \frac{2}{p_{opt} \tau_{opt} \left(1 + p_{opt} \sum_{i=0}^{m-1} (2p_{opt})^i\right)} \tag{39}$$

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 Eq. 19, we observe that τ_{opt} is never larger than p_{opt} for $n > 1$ (note that for $n = 1$ the system is stable for any K_p). With this observation, we obtain the following constant upper bound (independent of n):

$$K_p < \frac{2}{p_{opt}^2 \left(1 + p_{opt} \sum_{i=0}^{m-1} (2p_{opt})^i\right)} \tag{40}$$

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

$$K_u = \frac{2}{p_{opt}^2 \left(1 + p_{opt} \sum_{i=0}^{m-1} (2p_{opt})^i\right)} \tag{41}$$

and set K_p according to Eq. 36,

$$K_p = \frac{0.4 \cdot 2}{p_{opt}^2 \left(1 + p_{opt} \sum_{i=0}^{m-1} (2p_{opt})^i\right)} \tag{42}$$

With the K_p value that makes the system become unstable we have $H(z)C(z) = -1$. With such a closed-loop transfer function, a given input value changes its

sign at every time slot, yielding an oscillation period of two slots ($T_i = 2$). Thus, from Eq. 37,

$$K_i = \frac{0.4}{0.85 p_{opt}^2 \left(1 + p_{opt} \sum_{i=0}^{m-1} (2p_{opt})^i\right)} \tag{43}$$

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

5 Performance evaluation

In order to evaluate the performance of the proposed algorithm, we performed an exhaustive set of simulation experiments. For this purpose, we have extended the simulator used in [18, 21]. This is an event-driven simulator written in C/C++. It implements independently for each station the protocol details and timing of the 802.11 EDCA MAC, and supports both saturated and non-saturated sources. We integrated into the simulator the proposed approach as well as other centralized solutions [14, 15].

For all tests, we used a payload size of 1000 bytes and the system parameters of the IEEE 802.11b physical layer [22]. For the simulation results, average and 95% confidence interval values are given (note that in many cases confidence intervals are too small to be appreciated in the graphs). Unless otherwise stated, we assume that all stations are saturated.

5.1 Throughput performance

The main objective of the proposed algorithm is to maximize the throughput performance of the WLAN. To verify if the proposed algorithm meets this objective, we evaluated the total throughput obtained for different numbers of stations n. As a benchmark against which to assess the performance of our approach, we compared it against the static optimal configuration given by Eq. 18 and the default configuration given in the 802.11e standard [16]. Note that the static optimal configuration method requires the knowledge of the number of active stations, which challenges its practical use.

The results of the experiment described above are given in Fig. 3. We can observe from the figure that the performance of the proposed algorithm follows very closely the static optimal configuration in terms of total throughput. In contrast, the default configuration performs well for a small number of stations but sees its performance substantially degraded as the number
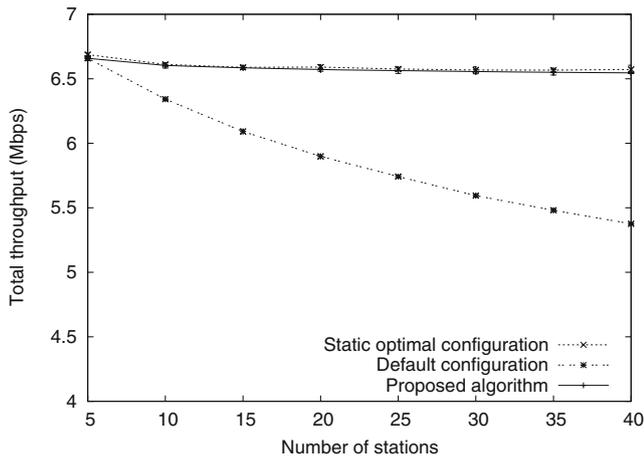
Figure 3 Throughput performance

Figure 5 Unstable configuration

of stations increases. From these results, we conclude that the proposed algorithm maximizes the throughput performance.

5.2 Stability

One of the objectives of the configuration of the PI controller presented in Section 4.4 is guaranteeing a stable behavior of the system. In order to assess this objective, we plot in Fig. 4 the value of the system's control signal (CW_{offset}) every beacon interval, for our $\{K_p, K_i\}$ setting with $n = 20$ stations. We can observe that with the proposed setting, CW_{offset} performs stably with minor deviations around its point of operation. In case that a larger setting for $\{K_p, K_i\}$ was used to improve the speed of reaction to changes, we would have the situation of Fig. 5. For this case, with values for $\{K_p, K_i\}$ 20 times larger, the CW_{offset} shows a
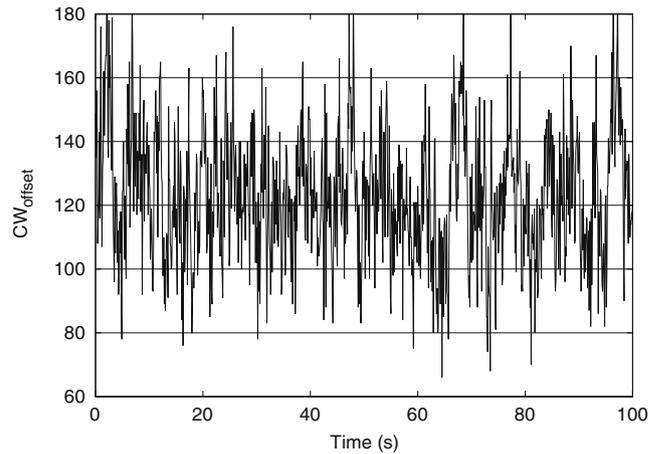
strong unstable behavior with drastic oscillations. We conclude that the proposed configuration achieves the objective of guaranteeing a stable behavior.

5.3 Speed of reaction to changes

In addition to a stable behavior, we also require the PI controller to quickly react to changes on the WLAN. To assess this objective we ran the following experiment. For a WLAN with 15 saturated stations, at $t = 80$ we added 15 more stations. We plot the behavior of CW_{offset} for our $\{K_p, K_i\}$ setting in Fig. 6 (label "Kp, Ki"). The system reacts fast to the changes on the WLAN, as CW_{offset} reaches the new value almost immediately. We have already shown in the previous section that large values for the parameters of the controller lead to unstable behavior. To analyze the impact of small values for these parameters, we plot on the
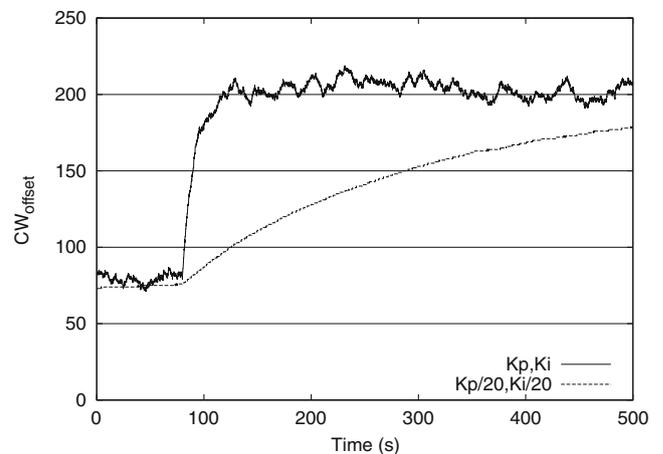

Figure 4 Stable configuration

Figure 6 Speed of reaction to changes

same figure the CW_{offset} evolution for a $\{K_p, K_i\}$ setting 20 times smaller (label "$Kp/20, Ki/20$"). With such setting, although obtaining a minor gain in stability, the system reacts too slow to changes of the conditions on the WLAN.

We conclude that, by means of the Ziegler-Nichols rules, we achieve a proper tradeoff between stability and speed of reaction to changes. To further validate this, in Fig. 7 we illustrate the time plot of the instantaneous throughput of one station, averaged over 1 s intervals, for the same previous experiment of Fig. 6. We can see from the figure that the system is able to provide stations with constant throughput (apart from minor oscillations due to the use of CSMA/CA), reacting almost immediately to changes.

5.4 Non-saturated stations

Our approach has been designed to optimize performance both under saturation and non-saturation conditions, in contrast to the static optimal configuration shown previously which is based on the assumption that all stations are saturated. In order to evaluate and compare the performance of the two algorithms when there are non-saturated stations in addition to saturated stations, we performed the following experiment. We had 5 saturated stations and a variable number of non-saturated stations in the WLAN. The non-saturated stations generated CBR traffic at rate of 100 Kbps. The total throughput resulting from this experiment is illustrated in Fig. 8. In this figure, we compare the performance of our approach against the static optimal configuration, resulting from computing the configuration with Eq. 18 and taking as n the total number of
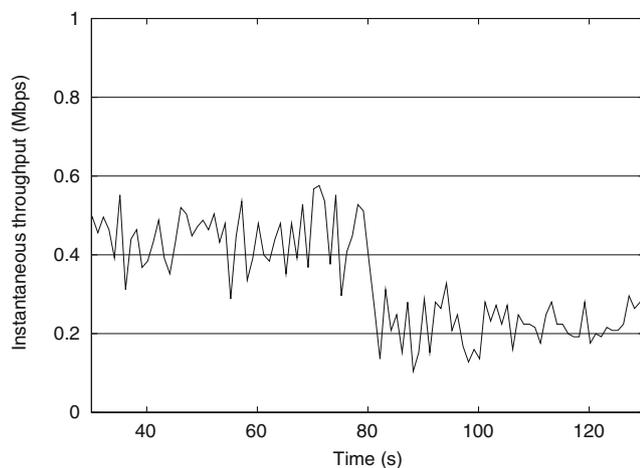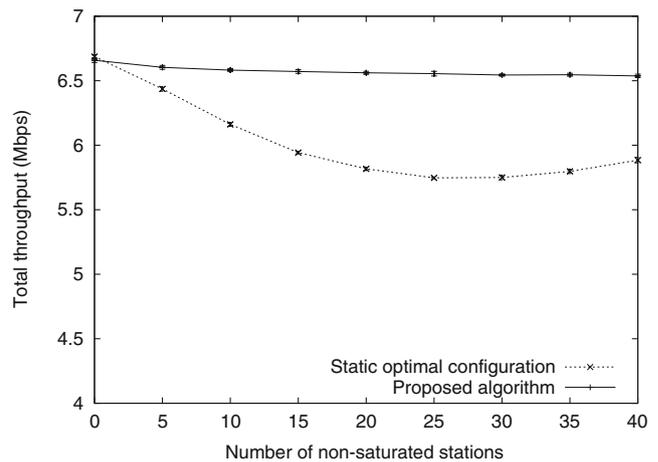

Figure 8 Non-saturated stations

stations present in the WLAN, regardless of whether they are saturated or not.

We observe from Fig. 8 that with our approach, the total throughput remains approximately constant with values similar to the ones obtained for saturation conditions (Fig. 3), independently of the number of non-saturated stations. In contrast, the performance of the static optimal configuration decreases very substantially as the number of non-saturated stations increases. This is due to the fact that the static optimal configuration considers that all stations are continuously sending packets and therefore uses too conservative CW values.

From the above results we conclude that our algorithm achieves optimal performance also when non-saturated stations are present in the WLAN, in contrast to the static optimal configuration which sees its performance severely degraded as the number of non-saturated stations increases.

5.5 Bursty traffic

In order to understand whether bursty traffic can harm the performance of the proposed algorithm, we repeated the experiment reported in the previous section but with the non-saturated stations sending highly bursty traffic instead of CBR. In particular, in our experiment we used ON/OFF sources with exponentially distributed active and idle periods of an average duration of 100 ms each. The results of this experiment are depicted in Fig. 9.

We can see from these results that, similarly to Fig. 8, the proposed algorithm performs optimally independent of the number of bursty stations, and substantially outperforms the static optimal configuration. We conclude that our approach does not only work well
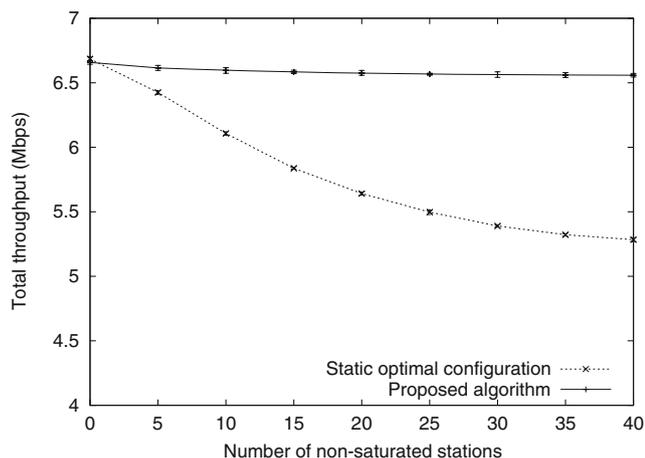

Figure 7 Instantaneous throughput

Figure 9 Bursty traffic

under constant traffic but also under highly variable sources.

5.6 Comparison against other approaches

The Sliding Contention Window (SCW) [14] and the dynamic tuning algorithm of [15] (hereafter referred to as DTA) are, like ours, centralized solutions compatible with the 802.11e standard that do not require hardware modifications. In this section we compare our solution against these centralized mechanisms.

Figure 10 gives the total throughput performance of the different solutions for varying numbers of stations. We observe that the proposed algorithm outperforms significantly both SCW and DTA. The reason is that our algorithm is sustained on the analysis of Section 3, which guarantees optimized performance, in contrast to SCW and DTA which are based on heuristics. In
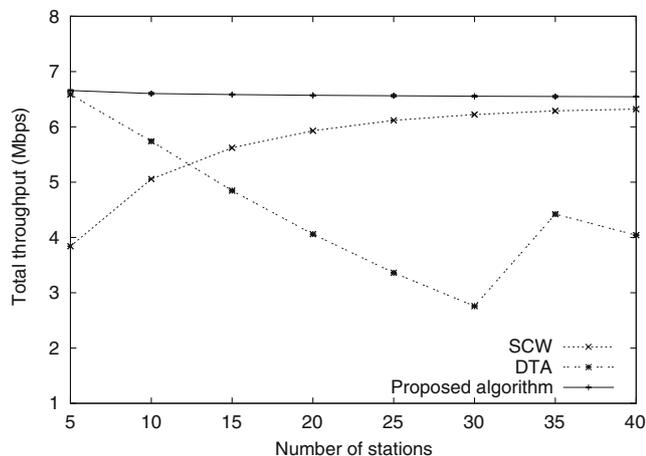
particular, SCW uses an algorithm to adjust CW_{min} that chooses overly large values, thereby degrading the performance. On the other hand, DTA sets the CW_{min} value as an heuristic function of the number of stations yielding overly small values, which results in a degraded performance also for this case.

5.7 Impact of channel errors

Most of the adaptive mechanisms proposed for IEEE 802.11 WLANs do not consider the impact of channel errors [4–15]. However, channel errors may influence these mechanisms since they are wrongly interpreted as collisions, leading to an unnecessary increase of the contention window and therefore to a suboptimal configuration.

In order to asses the impact of channel errors upon our approach we performed the following experiment. We varied the frame error rate (FER) from 0% to 10% for a scenario with $n = 20$ active stations in the WLAN. We compared the performance of our proposal against the static optimal configuration, which does not change the configuration upon failed transmissions and therefore uses always the optimal contention window value. The results of this experiment are illustrated in Fig. 11. We observe that for a realistic range of error probabilities (from 0% to 5%) the impact on throughput performance is negligible. Moreover, even for very large error rates (up to 10%) the performance loss is very small. Note that current WLANs use link adaptation mechanisms which guarantee small error rates by choosing a more robust modulation scheme upon detecting quality variations of the wireless channel [23]. We conclude that with the proposed scheme errors have a minimal impact on the performance.
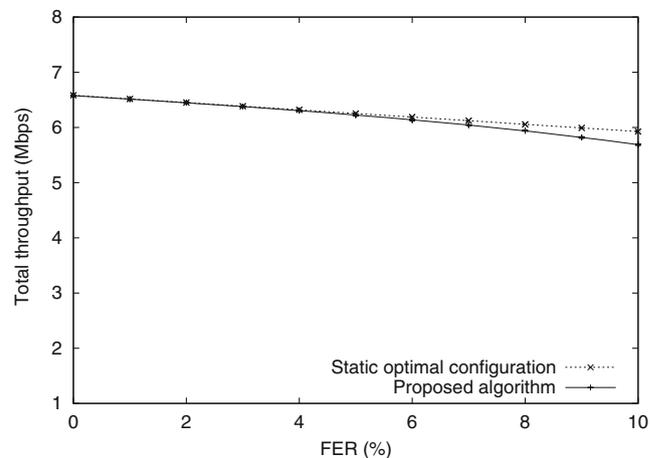

Figure 10 Comparison against other approaches

Figure 11 Impact of channel errors

6 Conclusions

In this paper we have proposed a novel adaptive algorithm for optimizing the performance of a WLAN. The algorithm is sustained on the observation that the collision probability in an optimally configured WLAN is approximately constant, independent of the number of stations. Our proposal only requires to measure this collision probability by monitoring successfully transmitted frames during an inter-beacon period at the AP.

Our algorithm is based on a well established controller from discrete-time control theory, the PI controller. By means of a theoretical analysis of the WLAN and the controller, we have designed our algorithm to maximize the throughput performance. We achieve a proper tradeoff between stability and speed of reaction to changes by applying the Ziegler-Nichols rules. We have shown via simulations that our algorithm drives the WLAN to the optimal point of operation, even for non-saturated and highly bursty traffic, reacting quickly to changes of the conditions in the WLAN.

As opposed to most of the previous proposals, our algorithm is fully compatible with the 802.11e EDCA standard and does not require any modifications neither at a hardware nor at a driver level. We have shown that our proposal substantially outperforms other centralized 802.11e-compatible solutions.

Appendix

Theorem 1 *The system is stable with the proposed K_p and K_i configuration.*

Proof The closed-loop transfer function of our system is

$$S(z) = \frac{-C(z)H(z)}{1 - C(z)H(z)} =$$
$$= \frac{-z(z-1)HK_p - zHK_i}{z^2 + (-HK_p - 1)z + H(K_p - K_i)} \quad (44)$$

where

$$H = -\frac{\tau_{opt} p_{opt} \left(1 + p_{opt} \sum_{i=0}^{m-1} (2p_{opt})^i\right)}{2} \quad (45)$$

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 [24]:

$$a_2 < 1 \quad (46)$$

$$a_1 < a_2 + 1 \quad (47)$$

$$a_1 > -1 - a_2 \quad (48)$$

In the transfer function of Eq. 44 the coefficients of the characteristic polynomial are

$$a_1 = -HK_p - 1 \quad (49)$$

$$a_2 = H(K_p - K_i) \quad (50)$$

From Eqs. 42 and 45 we have

$$HK_p = -0.4 \frac{\tau_{opt}}{p_{opt}} \quad (51)$$

and from Eqs. 43 and 45 we have

$$HK_i = -\frac{0.4}{0.85 \cdot 2} \frac{\tau_{opt}}{p_{opt}} \quad (52)$$

from which

$$a_1 = 0.4 \frac{\tau_{opt}}{p_{opt}} - 1 \quad (53)$$

$$a_2 = -0.16 \frac{\tau_{opt}}{p_{opt}} \quad (54)$$

Given $\tau_{opt} \leq p_{opt}$, it can be easily seen that the above $\{a_1, a_2\}$ satisfy the conditions of Eqs. 46, 47 and 48. The proof follows.

References

1. IEEE 802.11 WG (1999) 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, August. IEEE, Piscataway
2. Bianchi G (2000) Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J Sel Areas Commun 18(3):535–547, March

3. Banchs A, Pérez-Costa X, Qiao D (2003) Providing throughput guarantees in IEEE 802.11e wireless LANs. In: Proceedings of the 18th international teletraffic congress (ITC18), Berlin, September 2003

4. Yang Y, Wang JJ, Kravets R (2007) Distributed optimal contention window control for elastic traffic in single-cell wireless LANs. IEEE/ACM Trans Netw 15(6):1373–1386, December

5. Bianchi G, Fratta LL, Oliveri M (1996) Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In: Proceedings of the seventh IEEE international symposium on personal, indoor and mobile radio communications (PIMRC96), Taipei, October 1996

6. Xia Q, Hamdi M (2006) Contention window adjustment for IEEE 802.11 WLANs: a control-theoretic approach. In: Proceedings of IEEE ICC 2006, Istanbul, June 2006

7. Ni Q, Aad I, Barakat C, Turletti T (2003) Modeling and analysis of slow CW decrease for IEEE 802.1 1 WLAN. In: Proceedings of the seventh IEEE international symposium on personal, indoor and mobile radio communications (PIMRC 2003), Beijing, September 2003

8. Pang Q, Liew SC, Lee JYB, Leung VCM (2004) Performance evaluation of an adaptive backoff scheme for WLAN. Wirel Commun Mob Comput 4(8):867–879, December

9. Song N-O, Kwak B-J, Song J, Miller ME (2003) Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm. In: Proceedings of the 57th IEEE semiannual vehicular technology conference (VTC 2003-Spring), Jeju, April 2003

10. Banchs A, Perez X (2006) Distributed fair queuing in IEEE 802.11 wireless LAN. In: Proceedings of IEEE ICC 2002, New York, April 2006

11. Heusse M, Rousseau F, Guillier R, Duda A (2005) Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. In: SIGCOMM '05: Proceedings of the 2005 conference on applications, technologies, architectures, and protocols for computer communications. ACM, New York, pp 121–132

12. Calì F, Conti M, Gregori E (2000) Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit. IEEE/ACM Trans Netw 8(6):785–799

13. Bononi L, Conti M, Gregori E (2004) Runtime optimization of ieee 802.11 wireless lans performance. IEEE Trans Parallel Distrib Syst 15(1):66–80

14. Nafaa A, Ksentini A, Ahmed Mehaoua A, Ishibashi B, Iraqi Y, Boutaba R (2005) Sliding contention window (SCW): towards backoff range-based service differentiation over IEEE 802.11 wireless LAN networks. IEEE Network 19(4):45–51, July

15. Freitag J, da Fonseca NLS, de Rezende JF (2006) Tuning of 802.11e network parameters. IEEE Commun Lett 10(8):611–613, August

16. IEEE 802.11 WG (2005) 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

17. IEEE 802.11 WG (2006) 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-REVma/D9.0, Revision of Std. 802.11–1999

18. Banchs A, Vollero L (2006) Throughput analysis and optimal configuration of 802.11e EDCA. Comput Networks 50(11):1749–1768, August

19. Hollot CV, Misra V, Towsley D, Gong W-B (2001) A control theoretic analysis of RED. In: Proceedings of IEEE INFOCOM 2001, Anchorage, April 2001

20. Franklin GF, Powell JD, Workman ML (1990) Digital control of dynamic systems, 2nd edn. Addison-Wesley, Reading

21. Banchs A, Serrano P, Oliver H (2007) Proportional fair throughput allocation in multirate 802.11e EDCA wireless LANs. Wirel Netw 13(5), October

22. IEEE 802.11 WG (1999) 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: high-speed physical layer extension in the 2.4 GHz band, Supplement to IEEE 802.11 standard, September

23. Aguiar A, Wolisz A (2007) Channel prediction heuristics for adaptive modulation in WLAN. IEEE 65th Vehicular Technology Conference, 2007. VTC2007-Spring, April 2007, Dublin

24. Aström K, Wittenmark B (1990) Computer-controlled systems, theory and design. Prentice Hall International Editions, 2nd edn. Prentice Hall, Englewood Cliffs