Adaptive Optimization of IEEE 802.11 DCF Based on Bayesian Estimation of the Number of Competing Terminals

Alberto Lopez Toledo, Tom Vercauteren, and Xiaodong Wang

Abstract—The performance of the Distributed Coordination Function (DCF) of the IEEE 802.11 protocol has been shown to heavily depend on the number of terminals accessing the distributed medium. The DCF uses a carrier sense multiple access scheme with collision avoidance (CSMA/CA), where the backoff parameters are fixed and determined by the standard. While those parameters were chosen to provide a good protocol performance, they fail to provide an optimum utilization of the channel in many scenarios. In particular, under heavy load scenarios, the utilization of the medium can drop tenfold. Most of the optimization mechanisms proposed in the literature are based on adapting the DCF backoff parameters to the estimate of the number of competing terminals in the network. However, existing estimation algorithms are either inaccurate or too complex. In this paper, we propose an enhanced version of the IEEE 802.11 DCF that employs an adaptive estimator of the number of competing terminals based on sequential Monte Carlo methods. The algorithm uses a Bayesian approach, optimizing the backoff parameters of the DCF based on the predictive distribution of the number of competing terminals. We show that our algorithm is simple yet highly accurate even at small time scales. We implement our proposed new DCF in the ns-2 simulator and show that it outperforms existing methods. We also show that its accuracy can be used to improve the results of the protocol even when the terminals are not in saturation mode. Moreover, we show that there exists a Nash equilibrium strategy that prevents rogue terminals from changing their parameters for their own benefit, making the algorithm safely applicable in a complete distributed fashion.

Index Terms— IEEE 802.11 wireless networks, distributed coordination function, sequential Monte Carlo, game theory, Nash equilibrium.

1 Introduction

The IEEE 802.11 protocol [1] has become the predominant technology for wireless local area networks (WLAN). One of the most important elements of the 802.11 in terms of performance is the medium-access control (MAC). The MAC protocol is used to provide arbitrated access to a shared medium, in which several terminals access and compete for the radio spectrum. The design of the MAC protocols is often application-dependent and it is closely linked to the characteristics of the medium in which it operates. It also determines the performance metrics of the network, such as throughput, stability, and delay bounds, that directly affect the quality of service (QoS) offered to the applications. The IEEE 802.11 wireless networks employ the distributed coordination function (DCF) as a primary medium access mechanism. It is based on the carrier-sensing multiple-access with collision avoidance (CSMA/CA) protocol and binary exponential backoff [2], [3]. The performance of an IEEE 802.11 network largely depends on the operation of this backoff mechanism. Several studies of 802.11 have shown that the DCF is very sensitive to the number of competing terminals that access the wireless channel [2], [3], [4], [5], [6], [7]. These works indicate that a way to optimize the network performance is to make the parameters of the backoff window depend on the number of terminals competing for the medium. Several existing works have proposed solutions based on this premise, but their estimation algorithms are either inaccurate or too complex.

In this paper, we propose a dynamic optimization protocol to optimize the operation of the IEEE 802.11 DCF based on an online Bayesian estimator of the number of competing terminals developed by the authors in [8]. The sequential Monte Carlo (SMC) methodology [9] has been shown to be extremely powerful in dealing with filtering problems in non-Gaussian and nonlinear complex dynamic systems, where conventional approaches fail to work. In [8], we developed several SMC-based adaptive estimators for the number of competing terminals in an IEEE 802.11 network that outperform the existing best estimator based on the extended Kalman filter (EKF) [3]. In particular, we developed a maximum a posteriori (MAP) estimator whose computational load and memory requirements are equivalent to those of the well-known Viterbi algorithm. Consequently, our algorithm overcomes both of the problems mentioned above: It is accurate and easy to implement. In this paper, we propose an optimization mechanism that is able to make use of the predictive distribution of the number of competing terminals to maximize the throughput of the IEEE 802.11 DCF. We show that the accuracy of the Bayesian algorithm is particularly good at small time scales, which makes our proposal attractive to optimize the

• *The authors are with the Department of Electrical Engineering, Columbia University, 500 West 120th St., New York, NY 10027.*
E-mail: {alberto, tvercauteren, wangx}@ee.columbia.edu.

Manuscript received 14 Jan. 2005; revised 4 May 2005; accepted 24 Oct. 2005; published online 17 July 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0008-0105.

protocol when the terminals are in a nonsaturation regime, a problem usually not addressed in the literature. Finally, we propose a modification of the optimal window size selection that is a subgame perfect Nash equilibrium strategy, allowing the algorithm to be implemented in a completely distributed scenario without the arbitrage of a central entity such as an access point. A conference version of this work appears in [10].

The remainder of this paper is organized as follows: Section 2 presents the prior art. Section 3 describes the operation of the IEEE 802.11 DCF and discusses its performance. In Section 4, we present the problem of estimating the number of competing terminals in an IEEE 802.11 network, and we summarize the EKF and SMC algorithms that we use for comparison. In Section 5, we give the mathematical formulation for the problem of maximizing throughput of the IEEE 802.11 DCF based on the knowledge of the number of active users. Section 6 employs the game theory tools to analyze the DCF problem and shows that it has a subgame perfect Nash equilibrium. The performance of the estimators is evaluated in Section 7 by using both model-based data and realistic ns-2 simulations. Finally, Section 8 concludes the paper.

2 Existing Works and Motivations

The prior art on the optimization of the IEEE 802.11 DCF can be basically divided in three categories: 1) change the contention resolution algorithm, 2) adapt the parameters of the protocol (e.g., backoff parameters) to a rough estimate of the network status, being the number of competing terminals, the load of the network, the probability of collision, etc. (all these performance metrics are linked in one way or another), and 3) adapt the parameters of the protocol to an *accurate* estimate of the network status, often using advanced filtering mechanisms.

From the first category, a fast collision recovery (FCR) mechanism is proposed in [11] that dynamically distributes the backoff timer among all competing terminals, trying to reduce the unused backoff time. FCR gives more priority to terminals that have transmitted recently, thus reducing the collision penalty. This, however, exacerbates the already existing unfairness of IEEE 802.11 as pointed out in [4]. In [4], a new collision-resolution mechanism named GDCF is introduced. This new protocol implements a "gentle" reduction in the contention window after a successful transmission, in contrast to the "hard" reduction of the window to CW_{min} and, hence, avoiding increasing the collision probability that is likely to appear subsequent to a successful transmission. A similar but simpler approach called "probabilistic" DCF (PDCF) is presented in [12], where the window size is reset to CW_{min} with fixed probability.

The second and third categories are mainly based on the fact that the performance of the IEEE 802.11 DCF is very sensitive to the number of competing terminals accessing the wireless channel [2]. One of the main difficulties in adjusting the backoff parameters to the number of competing terminals in the network is that its value is often not available to the terminals. And, while a terminal could cache the identity of the past senders in the network, the number of competing terminals is the number of terminals that have data to send at any given time, so the list of neighbors is not sufficient. A distributed algorithm called IEEE 802.11+ is proposed in [5] and extended in [13]. It assumes that the backoff interval is sampled from a geometrical distribution with parameter p and, by estimating p, the contention window size is adapted to improve the throughput of the protocol. The parameter p is approximated using the average contention window of a *tagged* terminal. In self-adapt DCF [14], the number of competing terminals is estimated by monitoring the current load of the channel. However, the load of the channel is often noisy and may mislead the estimator under nonaverage transmitting scenarios. Moreover, it requires changes in the RTS/CTS frame format, making it impossible to adopt in current networks. Finally, in [15], a simpler scheme is proposed by estimating only the "range" of number of the terminals (dynamic optimization on range, or DOOR) and not its actual value. It is claimed that this approach is more suitable for implementation than complex filter-based estimations and leads to similar results. However, the range estimation inevitably leads to inaccurate results. Belonging to the last category, an extended Kalman filter (EKF) algorithm is proposed in [3], assuming a constant number of users and making use of a cumulative summary (CUSUM) change detection trigger. The EKF approach implicitly uses a linear Gaussian model departing from the discrete nature of the variables of interest in the 802.11 protocol (number of users, number of busy slots, etc.).

The estimation-based mechanisms have a benefit over their protocol-modification counterparts since they only involve adjusting the contention window parameters (e.g., CW_{min} and CW_{max}), while the rest of the protocol (even the use of the contention window values) remains unchanged. This is particularly useful in practice, as the code running in the access points (AP) does not need to be changed. This allows an easy modification of existing implementations and ensures the coexistence of enhanced versions with legacy ones. However, existing methods based on the estimation of the number of competing terminals exhibit two problems. First, the number of competing terminals is a non-Gaussian nonlinear dynamic system that is difficult to track accurately with conventional filters. Advanced estimators, such as the EKF-based estimator from [3], provide better results, but they are subject to critics due to their complexity [15]. Second, the performance of the IEEE 802.11 DCF is extremely sensitive to the number of competing terminals, particularly in the typical operating point of 1-15 terminals. This makes the simple approximation methods, such as DOOR, yield a suboptimal operation of the protocol compared with the theoretical optimum. In our opinion, there is a need for an accurate estimation algorithm that is able to efficiently track the number of competing terminals in an IEEE 802.11 network, and, at the same time, is easy to implement.

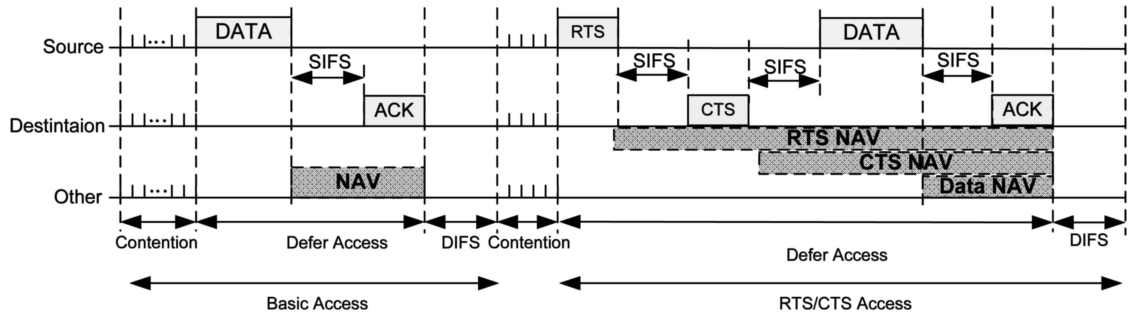


Fig. 1. IEEE 802.11 MAC access mechanisms.

3 IEEE 802.11 Distributed Coordination Function

3.1 The DCF Protocol

In IEEE 802.11 DCF, the access to the medium is controlled by the use of interframe space (IFS) times between transmissions [4]. The IFS interval defines the amount of time a terminal needs to wait before being able to transmit. IEEE 802.11 specifies four IFS intervals: the short IFS (SIFS), the point coordination function IFS (PIFS), the distributed coordination function IFS (DIFS), and the extended IFS (EIFS). The IFS intervals have durations related to the priority of the data being transmitted. The SIFS is usually assigned to high priority data such as acknowledgements (ACK), the PIFS is assigned to the point coordinating function (PCF), the centralized DCF counterpart, in which some delay bounds are defined, the DIFS is assigned to DCF data, and, finally, in order of priority, the EIFS is used for situations in which a terminal has received unknown or incorrect data and needs to report it. For that reason, they are defined such as SIFS $<$ PIFS $<$ DIFS $<$ EIFS. The DCF defines two distinct techniques to access the medium: the *basic access* and the *RTS/CTS* access.

Basic Access: In the basic access, the terminals implement a two-way handshake mechanism (Fig. 1). A terminal senses the channel to be idle before starting a transmission. If the channel is idle for at least a period of distributed interframe space (DIFS), then the terminal is allowed to transmit. If during this sensing time the channel appears to be busy at any time, the terminal defers the transmission and enters into the collision avoidance (CA) mode. In CA mode, the terminal generates a random backoff interval during which it waits before attempting another transmission. This random backoff is used to minimize the probability of collision between terminals accessing the medium. The idle time after waiting for a DIFS interval is slotted and the terminals are only allowed to transmit at the beginning of the slot time. The slot time size σ accounts for the time the signal is propagating and is set equal to the time needed for any terminal to detect the transmission of a packet from any other terminal [2]. If this time was not accounted for, a terminal could assess the channel as idle when the data sent by another terminal has not yet arrived.

The random backoff timer is uniformly chosen between $[0, v)$, where v is called the *contention window*, and it is such that satisfies $v \in [CW_{min}, CW_{max}]$, where CW_{min} and CW_{max} are called the *minimum* and *maximum* contention window, respectively. At the first transmission attempt, the value of the contention window v is set to CW_{min}. The backoff timer is decremented while the channel is idle (i.e., it only counts the *idle time*). If, at any time, the channel is sensed busy, the backoff timer is paused until the channels are sensed idle again *after* the corresponding DIFS time. When the backoff timer reaches 0, the terminal is allowed to transmit. Following the successful reception of the data, the receiving terminal waits a SIFS interval and transmit an ACK to the transmit terminal. As the SIFS interval is shorter than the DIFS interval, the destination terminal has priority in sending the ACK. Such a two-way handshake-based ACK is necessary because the CSMA/CA protocol does not assume the terminals have the capability to detect collisions. Upon reception of the ACK, the backoff stage is reset to 0 and $v = CW_{min}$. This is referred as a "heavy decrease" in [4]. If the source terminal does not receive the ACK after a timeout period (ACK_timeout) or it detects the transmission of any other frame in the channel (collision), the frame is assumed to be lost. After each unsuccessful transmission, the value of v is doubled up to a maximum of $CW_{max} = 2^m CW_{min}$, where m is usually referred to as the *maximum backoff stage* [4]. The values of CW_{min}, CW_{max}, and slot size σ are determined by the characteristics of the physical layer.

As shown in Fig. 1, when another terminal is transmitting, the rest of terminals set up the network-allocation network timer (NAV), which acts as a virtual carrier sense. When hearing a data frame, the rest of the terminals set up the duration of the NAV to the duration specified in the header of the transmitted data frame. The NAV vector includes the SIFS and the duration of the ACK transmission. All the terminals defer their access to the medium until the NAV timer expires.

The *RTS/CTS* access is similar to the *basic access* but makes use of a four-way handshake protocol in which, prior to data transmission, a terminal transmits a special short request-to-send frame (RTS) to try to "reserve" the transmission and reduce the cost of collisions. The receive terminal responds with another short special clear-to-send (CTS) frame as shown in Fig. 1. In this paper, we focus primarily on the *basic access*.

3.2 Analytical Throughput

We consider an IEEE 802.11 network with DCF operating in the basic access mode as described in Section 3.1. We assume that the number of terminals using the network at a given time is finite. Note that, while the IEEE 802.11

protocol does not specify a mechanism for access control, it is safe to assume that the number of users is upper bounded. As the number of users increases, the collisions in the network would also increase, leading to a saturation of the network that would eventually lead the utilization of all the terminals to zero. We also assume that the terminals transmit in a saturation regime, i.e., they always have something to send. In this saturation regime, it is shown in [2] that the normalized throughput of the system can be analytically derived.

From the point of view of a terminal, the time can be slotted into variable length slots. Specifically, in the DCF operation, a time slot will correspond to an idle slot σ, a busy slot which has the duration of a successful transmission $T_s(L)$, or the duration of a collision $T_c(L^*)$, where L is the time length of the packet and L^* is the time length of the largest of the packets involved in the collision. The normalized throughput is then given by

$$S = \frac{\mathbb{E}[L]}{T_s - T_c + \frac{T_c + \sigma(1-P_{tr})}{P_s}}, \tag{1}$$

where P_{tr} is the probability of a terminal transmitting in the slot and $\mathbb{E}[L]$, $T_s = T_s(\mathbb{E}[L])$, $T_c = T_c(\mathbb{E}[L^*])$, and σ are constants denoting, respectively, the average packet payload length, the average time of a busy slot with successful transmission, the average time of a busy slot with collision, and the duration of an empty slot. $\mathbb{E}[L^*]$ is the average length of the longest packet involved in a collision.

Let x_t be the number of competing terminals in the network at time t (discrete). Here, the term *competing* refers to a terminal that is either transmitting or backlogged (i.e., it has data to send). Let q be the probability that a terminal transmits in a given slot. Then, the probability P_{tr} that at least one terminal transmits in a given time slot t is given by

$$P_{tr} = 1 - (1-q)^{x_t}. \tag{2}$$

In [2], it is shown that

$$q = \frac{2(1-2p_c)}{(1-2p_c)(CW_{min}+1) + p_c CW_{min}(1-(2p_c)^m)}, \tag{3}$$

where CW_{min} and m are the minimum contention window and the maximum backoff stage, respectively. Then, the collision probability p_c and the probability of a successful transmission P_s for a terminal are given by

$$p_c = 1 - (1-q)^{x_t - 1}, \qquad P_s = \frac{x_t \cdot q(1-q)^{x_t-1}}{1-(1-q)^{x_t}}. \tag{4}$$

The interesting conclusion of this analysis is that the throughput is a function of the number of competing terminals x_t and the probability of a terminal transmitting q. Given x_t, CW_{min}, and m, (3) and (4) can be solved and a unique solution can be found [2]. Therefore, the normalized saturation throughput only depends on the number of competing terminals and the backoff parameters, i.e.,

$$S = S(x_t, CW_{min}, m). \tag{5}$$

Then, once the number of competing terminals x_t is estimated, the optimization problem involves selecting the other parameters CW_{min} and m to maximize the system throughput.

4 Estimation of the Number of Competing Terminals

It is shown in [2] that, when the terminals are in saturation regime, i.e., they always have a packet to send, and when the system reaches a steady state, then the number of competing terminals x_t can be expressed as a function of the collision probability p_c as

$$x_t = f(p_c) \triangleq 1 + \frac{\log(1-p_c)}{\log\left(1 - \frac{2(1-2p_c)}{(1-2p_c)(CW_{min}+1)+p_c CW_{min}(1-(2p_c)^m)}\right)}. \tag{6}$$

The above function is monotonic increasing in p_c and, hence, an inverse function exists, i.e, $p_c = h(x_t)$, where $h(\cdot) = f^{-1}(\cdot)$. The problem of estimating the number of competing terminals then involves estimating x_t based on a noisy observation of p_c, which each terminal can acquire by monitoring the channel activity.

While there are different ways of estimating the collision probability in the channel, an easy method proposed in [3] consists of counting the proportion of busy slots in a given period. The number of busy slots is an indication of the collision probability because an attempt of transmission in a busy slot would result in a sure collision. This process of counting the slots does not impose any burden to the normal operation of the DCF, as the terminals are always monitoring the network and checking the medium state.[1] So, our observation variable of the collision probability y_t can be defined at each time step t as

$$y_t = \sum_{i=(t-1)B}^{tB-1} C_i, \tag{7}$$

where $C_i = 0$ if the ith time slot is empty or corresponds to a successful transmission (i.e., no collision) and $C_i = 1$ if the ith basic time slot is busy or corresponds to an unsuccessful transmission (i.e., would result in collision); B is the number of slots that compose the *observation slot* for the measurement. It is easy to see that y_t follows a binomial distribution $\mathcal{B}(B, p_c)$ with B trials and probability of success p_c, i.e.,

$$p(y_t = b) = \binom{B}{b} p_c^b (1-p_c)^{B-b}, \quad b = 0, 1, 2, \ldots, B. \tag{8}$$

Therefore, we can see that $\mathrm{Var}[y_t] = Bp_c(1-p_c)$ and $\mathbb{E}[y_t] = Bp_c$. From the above discussion, we can cast our problem into the following state-space representation:

$$x_t \sim \mathcal{M}(\boldsymbol{\theta}), \tag{9}$$

$$y_t \sim \mathcal{B}(B, h(x_t)), \tag{10}$$

1. We are assuming that the terminals are not in power-saving mode.

where $\mathcal{M}(\boldsymbol{\theta})$ denotes a discrete-time Markovian model with some unknown parameters $\boldsymbol{\theta}$; x_t is the state realization of the Markovian model at time instant t.

In this paper, we consider two advanced methods for estimating x_t based in y_t: the extended Kalman Filter with CUSUM approximation developed in [3] and sequential Monte Carlo (Bayesian)-based estimators developed in [8].

EKF-based estimator: In [3], the EKF approach implicitely simplifies the state-space (9)-(10) into a linear Gaussian model. The state x_t is assumed to fluctuate as $x_t = x_{t-1} + w_t$, where w_t is the *state noise* with variance Q_t and the observation is approximated as $y_t = Bh(x_t) + u_t$, where u_t is the *observation noise* with variance $R_t = Bh(x_t)(1 - h(x_t))$. In order to fit the EKF assumptions, all variables are used as continuous variables and Q_t and R_t are assumed to be known. Therefore, R_t is approximated by

$$\begin{aligned} R_t &\approx B \cdot h(\hat{x}_{t|t-1}) \cdot (1 - h(\hat{x}_{t|t-1})) \\ &= B \cdot h(\hat{x}_{t-1}) \cdot (1 - h(\hat{x}_{t-1})), \end{aligned} \tag{11}$$

where $\hat{x}_{t|t-1}$ and $\hat{x}_t$ are estimates of $\mathbb{E}[x_t|y_1, y_2, \ldots, y_{t-1}]$ and $\mathbb{E}[x_t|y_1, y_2, \ldots, y_t]$, respectively.

The usual EKF approach provides the following update procedure:

$$\hat{x}_t = \hat{x}_{t-1} + K_t z_t, \tag{12}$$

where $z_t = y_t - Bh(\hat{x}_{t|t-1}) = y_t - Bh(\hat{x}_{t-1})$ is the innovation given by the tth measure and the Kalman gain K_t is specified by

$$K_t = \frac{(P_{t-1} + Q_t)g_t}{(P_{t-1} + Q_t)g_t^2 + R_t}, \tag{13}$$

where g_t is the sensitivity of the measurement

$$g_t = B\frac{\partial h(x)}{\partial x}\bigg|_{x=\hat{x}_{t-1}}. \tag{14}$$

It is observed in [3] that Q_t, the variance of the process noise, needs to be adjusted. In [3], the number of competing terminals is assumed to be constant ($Q_t = 0$) and a "change detection" filter based on the cumulative summary (CUSUM) test is implemented to select a large value (determined by simulations, e.g., 5) for Q_t when the number of competing terminals is detected to have noticeably changed. The intuition behind it is that, if the number of competing terminals does not change too abruptly, the network state will never be too far from the analyzed steady state.

SMC-based Estimators: In [8], it is assumed that the number of competing terminals evolves according to a first-order Markov chain with an unknown transition probability matrix $\boldsymbol{A} = [a_{i,j}]$, i.e., $p(x_{t+1} = j|x_t = i) = a_{i,j}$, where $a_{i,j} \geq 0$ and $\sum_{j=1}^{N} a_{i,j} = 1$, N is the maximum number of competing terminals, and initial probability vector $\boldsymbol{\pi} = [\pi_1, \cdots, \pi_N]$, i.e., $p(x_0 = i) = \pi_i$. Because the probability that a large number of users enter or leave the system in two consecutive time slots is small, the transition matrix $\boldsymbol{A}$ is assumed to have a banded structure, i.e., $a_{i,j} = 0, \quad \text{for } |i - j| > \delta$ for some $\delta < N$. Hence, we have the state equation $x_t \sim \mathcal{MC}(\boldsymbol{\pi}, \boldsymbol{A})$, with $\boldsymbol{\pi}$ and $\boldsymbol{A}$ unknown and the observation equation $y_t = \mathcal{B}(B, p_c)$.

Denote the observation sequence up to time t as $\boldsymbol{y}_t \triangleq [y_1, y_2, \cdots, y_t]$ and the network state sequence up to time t as $\boldsymbol{x}_t \triangleq [x_1, x_2, \cdots, x_t]$, and denote the unknown parameters as $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}\}$. We are interested in obtaining a Bayesian estimate of the posterior distributions $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$ and $p(\boldsymbol{\theta}_t|\boldsymbol{y}_t)$. In this paper, we will present two algorithms developed in [8], deterministic SMC and approximate *maximum a posteriori* (MAP). The basic idea behind the sequential Monte Carlo (SMC) approach is to recursively update a set of weighted samples, $\{\boldsymbol{x}_t^{(k)}, \boldsymbol{\theta}_t^{(k)}, \boldsymbol{w}_t^{(k)}\}$, $k = 1, 2, \ldots, K$, representing the distributions of interest. For example, we have:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_t) \approx \sum_{k=1}^{K} w_t^{(k)} \mathbb{I}(\boldsymbol{x}_t - \boldsymbol{x}_t^{(k)}), \tag{15}$$

where $\mathbb{I}(x) = 1$ if $x = 0$ and $\mathbb{I}(x) = 0$ otherwise.

The usual SMC approach is not well-suited for parameter estimation (here, $\boldsymbol{\theta}$) [9] and the key to the approach developed in [8] is to see that the complete information about the transition matrix can be carried over through some sufficient statistics. A well-known strategy for Bayesian inference is to choose the prior distributions with a suitable form so that the posteriors belong to the same functional family as the prior. By assuming that the prior distributions of $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}\}$ are given by multivariate Dirichlet distributions, it is shown in [8] that the posterior distributions of $\boldsymbol{\theta}$ given $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are also multivariate Dirichlet distributions:

$$p(\boldsymbol{\pi}|\boldsymbol{x}_t, \boldsymbol{y}_t) = p(\boldsymbol{\pi}|x_1, y_1) = \mathcal{D}(\boldsymbol{\pi}_i; \rho_1, \rho_2, \cdots, \rho_N), \tag{16}$$

$$p(\boldsymbol{a}_i|\boldsymbol{x}_t, \boldsymbol{y}_t) = \mathcal{D}(\boldsymbol{a}_i; \alpha_{i,1,t}, \alpha_{i,2,t}, \cdots, \alpha_{i,N,t}), \quad i = 1, \cdots, N, \tag{17}$$

where $\mathcal{D}(.;\ldots)$ denotes the Dirichlet probability density function. We get the following update procedure:

$$\alpha_{i,j,t} = \alpha_{i,j,t-1} + \mathbb{I}(x_{t-1} - i)\mathbb{I}(x_t - j). \tag{18}$$

From the Bayes theorem, we have

$$p(\boldsymbol{x}_t|\boldsymbol{y}_t) = p(y_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1})p(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}). \tag{19}$$

Thanks to the sufficient statistics framework, (19) can be updated analytically:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_t) = \mathcal{B}(y_t; B, h(x_t)) \frac{\alpha_{x_{t-1},x_t,t-1}}{\sum_{j=1}^{N} \alpha_{x_{t-1},j,t-1}} p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}). \tag{20}$$

We now have all the tools to derive the deterministic SMC estimator. Suppose a set of weighted samples containing no duplicate and representing $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$ is available at time $(t-1)$, i.e., (15); based on (20), $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$ can be approximated by

$$p^{ext}(\boldsymbol{x}_t|\boldsymbol{y}_t) \propto \sum_{k=1}^{K} \sum_{j=1}^{N} w_t^{(k,i)} \mathbb{I}(x_t - i)\mathbb{I}(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{(k)}), \tag{21}$$

where the weight update procedure is given by

$$w_t^{(k,i)} \propto w_{t-1}^{(k)} \mathcal{B}(y_t; i) \frac{\alpha^{(k)}_{x^{(k)}_{t-1},i,t-1}}{\sum_{j=1}^{N} \alpha^{(k)}_{x^{(k)}_{t-1},j,t-1}}. \tag{22}$$

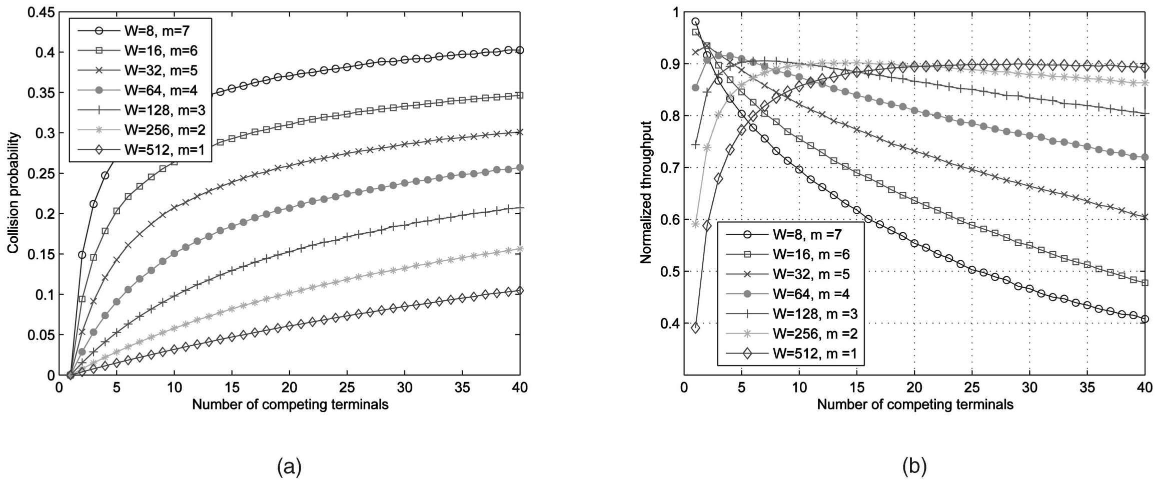


Fig. 2. Collision probabilities and throughput in ns2. (a) Collision probabilities from ns-2 used in the estimator. (b) Saturation throughput of the implementation of IEEE 802.11 in ns-2.

A selection step is then performed to retain a fixed number of samples.

The use of sufficient statistics also leads the authors in [8] to another estimation scheme which is a modification of the well-known Viterbi algorithm to fit the unknown transition matrix scenario. In this approximate MAP approach, the objective is to recursively maximize $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$ with respect to $\boldsymbol{x}_t$. With this goal, the Viterbi algorithm uses

$$\delta_t(i) = \max_{\boldsymbol{x}_{t-1}|x_t=i} p(\boldsymbol{x}_t|\boldsymbol{y}_t) = p(y_t|x_t=i) \max_{x_{t-1}|x_t=i} \max_{\boldsymbol{x}_{t-2}|x_{t-1},x_t=i} [p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})] \tag{23}$$

that can recursively be computed if the transition matrix is known by taking $p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1}) = a_{x_{t-1},x_t}$ out of the inner max. The estimate of x_t at time t is then given by $\max_i \delta_t(i)$. When the transition matrix is unknown, even if the probability of any path can be analytically computed as in (20), such a recursion cannot directly be used because $p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})$ depends on $\boldsymbol{x}_{t-2}$. However, if we make the approximation that $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})$ is maximized when $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$ is maximized, an approximation $\hat{\delta}_t(i)$ of $\delta_t(i)$ can be computed recursively as

$$\hat{\delta}_t(i) = p(y_t|x_t=i) \max_j \left[\hat{\delta}_{t-1}(j) p(x_t=i|\boldsymbol{x}^{(j)}_{t-1},\boldsymbol{y}_{t-1})\right] \tag{24}$$

$$= \mathcal{B}(y_t; B, h(i)) \max_j \left[\hat{\delta}_{t-1}(j) \cdot \frac{\alpha^{(j)}_{j,i,t-1}}{\sum_{k=1}^N \alpha^{(j)}_{j,k,t-1}}\right], \tag{25}$$

where $\boldsymbol{x}^{(j)}_{t-1}$ corresponds to the retained path ending at $x_{t-1}=j$ and $\alpha^{(j)}_{j,\cdot,t-1}$ is the corresponding sufficient statistics updated by (18). For more details of the two SMC-based estimators discussed above, the readers are referred to [8].

5 Optimization of IEEE 802.11 DCF

5.1 Utility Function

In this section, we propose a novel optimization algorithm based on the SMC estimators of the number of competing terminals described in the previous section. We are interested in optimizing the throughput of the IEEE 802.11 DCF when the number of competing terminals is less than 40. Our simulations in ns-2 (Fig. 2) show that the effect of CW_{max} greater than 1,024 has no effect on the network performance for $x_t \leq 40$. So, in order to simplify the problem, we impose m to be fixed such as $CW_{max} = 2^m CW_{min} = 1,024$, i.e., $m = \log_2\left(\frac{1,024}{CW_{min}}\right)$ and CW_{min}[2] takes values from a set $\mathcal{W}$. This set can be fixed or it can be constructed, for example, using the method in Section 5.3. Then, assuming m is no longer a variable, a simple formulation of the backoff window choice is given by

$$W^*_{t+1} = \arg\max_{W\in\mathcal{W}} \mathbb{E}_{p(x_{t+1}|\boldsymbol{y}_t)} \left\{\Psi_u\big(S(x_{t+1},W,m) - S(x_{t+1},W_t,m)\big)\right\}, \tag{26}$$

where Ψ_u is a utility function of the difference in throughput and $\mathrm{S}(\cdot)$ is given in (5). Ψ_u will typically be a nondecreasing function and should be convex on the positive part and concave on the negative part.

Considering the choice of the cost function, we are interested in studying the case in which a change in CW_{min} negatively affects the normal operation of the protocol. Let v be the actual window size for an IEEE 802.11 terminal. Then, by the operation of the protocol, we know that $v(t) \in [CW_{min}(t),\ 1{,}024]$ and, in the next observation slot,

$$v(t+1) = \begin{cases} CW_{min}(t), & \text{if success in } t, \\ \min(2\times v(t),\ 1{,}024), & \text{if collision in } t. \end{cases} \tag{27}$$

Note that, because the protocol dictates that

$$v(t) \in [CW_{min}(t), CW_{max}(t)],$$

a change in CW_{min} may produce a "jump" in the value of v if $v(t) < CW_{min}(t+1)$. Fig. 6a shows the average evolution of $v(t)$ with the number of competing terminals for different backoff parameters. As $x \to 20$, the optimal $CW_{min}(t) = 256$, and the average value for $v(t) \approx 330$. If $x(t+1) > 20$, the optimal value of $CW_{min}(t+1)$ becomes 512, forcing $v(t+1)$ to be at least 512 as indicated by the arrow. If the estimate

2. For ease of notation, we would use the term CW_{min} and W interchangeably.

$x(t+1) > 20$ is spurious and $x(t+2) < 20$ (so $CW_{min}(t+2)$ is back to 256 again), it would take a terminal an extra successful transmission to return v to the average correct level (i.e., 256), incurring an average delay of 128 slots. For this reason, the utility function Ψ_u must penalize oscillations of CW_{min}, so the change of window is not made for small differences.

An interesting note is that our utility function in (26) is not based on a hard decision on the number of competing terminals but makes use of its distribution if available. In [15], [16], a similar optimization scheme was introduced, but a hard estimate of the number of terminals was used to make a range estimation. To prevent frequent switching, the authors proposed using overlapping ranges. We believe that our Bayesian criterion is more natural to make a soft decision.

5.2 Predictive Distribution Based on SMC Samples

As shown in our criterion (26), we need to have access to the predictive distribution $p(x_{t+1}|\boldsymbol{y}_t)$ in order to perform an optimal control of the protocol.

Given a set of samples and weights $\left\{\boldsymbol{x}_t^{(k)}, w_t^{(k)}\right\}_{k=1}^K$ representing $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$ at time t, (26) can be approximated as

$$
\begin{aligned}
&\hat{W}_{t+1}^{SMC} \\
&= \arg\max_{W\in\mathcal{W}} \sum_{k=1}^K \sum_{i=1}^N \Psi_u(\Delta S(x_{t+1}=i,W))p(x_{t+1}=i,\boldsymbol{x}_t^{(k)}|\boldsymbol{y}_t) \\
&= \arg\max_{W\in\mathcal{W}} \sum_{k=1}^K \sum_{i=1}^N \Psi_u(\Delta S(i,W))\frac{\alpha_{x_t^{(k)},i,t}}{\sum_{j=1}^N \alpha_{x_t^{(k)},j,t}} w_t^{(k)} \\
&= \arg\max_{W\in\mathcal{W}} \sum_{k=1}^K \frac{w_t^{(k)}}{\sum_{j=1}^N \alpha_{x_t^{(k)},j,t}} \sum_{i=1}^N \Psi_u(\Delta S(i,W))\alpha_{x_t^{(k)},i,t},
\end{aligned}
\tag{28}
$$

where $\Delta S(x_{t+1},W) = S(x_{t+1},W,m) - S(x_{t+1},W_t,m)$.

For the case in which we only have access to a hard estimate of the number of competing terminals, the backoff window choice (26) is simply approximated by

$$
W_{t+1}^{MAP} = \arg\max_{W\in\mathcal{W}} \Psi_u\big(S(\hat{x}_{t+1|t},W,m) - S(\hat{x}_{t+1|t},W_t,m)\big),
\tag{29}
$$

where

$$
\hat{x}_{t+1|t} = \arg\max_{x_{t+1}} p(x_{t+1}|\hat{\boldsymbol{x}}_t,\boldsymbol{y}_t) \approx \arg\max_{x_{t+1}} p(x_{t+1}|\boldsymbol{x}_t,\boldsymbol{y}_t)
$$

is an approximate MAP estimate of x_{t+1} with $\hat{\boldsymbol{x}}_t$ being the current MAP estimate of $\boldsymbol{x}_t$.

For the EKF algorithm, $p(x_t|y_t,y_{t-1},\ldots,y_1)$ is approximated by a Gaussian $p(x_t|y_t,y_{t-1},\ldots,y_1) \approx \mathcal{N}(x_t;h(x_t),P_t)$. This would involve complex numerical integrations, so we use the hard estimate of the number of competing terminals as in (29).

5.3 Choice of Backoff Window Size Set $\mathcal{W}$

Having discussed how to perform an optimal choice of the backoff window within a given set $\mathcal{W}$, we can now give some insight on the choice of this set. It will be chosen such that the optimal throughput can always be approached and such that its cardinality remains low. Indeed, a small number of configurations will allow a more stable system and an easier implementation. Our design criterion can be written as

$$
\begin{aligned}
&\forall i \in [1,\cdots,N], \\
&|\max_{W\in\mathbb{N}^*} S(i,W,m) - \max_{W\in\mathcal{W}} S(i,W,m)| < \Delta S_{max},
\end{aligned}
\tag{30}
$$

where ΔS_{max} is the maximum throughput loss to optimality we allow. ΔS_{max} will typically be chosen small, for instance, 2.5 percent. Within this constraint, we would like to have as few points in $\mathcal{W}$ as possible. Because of the regularity of S(.), such a set can be constructed by performing the following operations:

1. Let $i_{mid} = 1$.
2. Choose the greatest integer j_{ref} such that
$$
S_{opt}(i_{mid}) - S(i_{mid},W_{opt}(j_{ref}),m) < \Delta S_{max},
$$
where
$$
W_{opt}(k) = \arg\max_{W\in\mathbb{N}^*} S(k,W,m)
$$
and $S_{opt}(k) = S(k,W_{opt}(k),m)$. Let $W_{opt}(j_{ref})$ be in $\mathcal{W}$.
3. Find the smallest integer $i_{mid} > j_{ref}$ such that $S_{opt}(i_{mid}) - S(i_{mid},W_{opt}(j_{ref}),m) > \Delta S_{max}$.
4. If $i_{mid} < N$ and $j_{ref} < N$, go back to Step 2.
5. If $i_{mid} \geq N$ and $j_{ref} \geq N$, remove $W_{opt}(j_{ref})$ from $\mathcal{W}$ and let N be in $\mathcal{W}$.

6 Game Theoretical Analysis of Distributed Implementations

So far, we have considered the scenario in which all the terminals agree on the observations and adjust to the window size that optimizes the total throughput of *the network*. This can be easily implemented if the estimation and optimization algorithms are placed in the access point and the optimal backoff parameters are broadcast to all terminals periodically. The access point can, for example, use piggybacking in the ACK frames, and the terminals may use overhearing to adjust the parameters accordingly. While this solution ensures that all terminals would use the same window size and, hence, provides fair results, the problem is that it requires the modification of the protocol, i.e., the access point needs to introduce new messages or new fields in existing frames that have to be understood by the terminals. A nice feature of the estimation and optimization algorithms described in this paper is that they are distributed by nature and they do require minimal changes in the IEEE 802.11 DCF (namely, the ability to adaptively change the values of the backoff parameters). In this section, we consider such a distributed scenario in which every terminal estimates the number of competing terminals and optimizes its own network utilization without sharing any information. While we explore this approach in the context of our optimization algorithm, the following analysis can be applied to any optimization scheme.

In such a distributed approach, allowing a terminal to select its own window may introduce unfairness. Moreover, a rogue terminal may want to change the backoff parameter for its own benefit. This kind of economic behavior can effectively be modeled and analyzed using game theory concepts and tools [17]. We will show in the following analysis that these situations can effectively be avoided and all the terminals can be enforced to select the appropriate windows parameters in a totally distributed manner. We will also show that this strategy is, in fact, the best possible, i.e., there is no incentive for any terminal to lie in their estimations for their own benefit, resulting in a stable optimal operation.

The *normal form* representation of an n-players game $G = \{E_1, \dots, E_n; u_1, \dots, u_n\}$ specifies the players' strategy spaces $\{E_i\}$ and their payoff functions $\{u_i\}$. In the context of our problem, we define the game as the following x-player game:

Definition 1. *The normal-form definition of the optimization of the IEEE 802.11 DCF (DCF game) is denoted by $G = \{\mathcal{W}_1, \dots, \mathcal{W}_x, S_1, \dots, S_x\}$, where x is the number of competing terminals, $\mathcal{W}_i$ is the set of values for CW_{min} for terminal i, and S_i is the throughput of terminal i.*

Note that the throughput S_i of terminal i is a function of x and the strategies of all the x competing terminals (i.e., $S_i = S_i(W_1, \dots, W_x)$). A powerful concept in game theory is the *Nash Equilibrium* (NE). In the x-player formal game, we say that a set of strategies is NE if it is the player's best response to the strategies selected by the rest of players.

Definition 2. *In the DCF game $G = \{\mathcal{W}_1, \dots, \mathcal{W}_x, S_1, \dots, S_x\}$, the strategies $(W_1^*, \dots, W_x^*)$ are* ***Nash Equilibrium*** *if*

$$\begin{aligned} &S_i(W_1^*, \dots, W_{i-1}^*, W_i^*, W_{i+1}^*, \dots, W_x^*) \\ &\geq S_i(W_1^*, \dots, W_{i-1}^*, W_i, W_{i+1}^*, \dots, W_x^*), \quad \forall W_i \in \mathcal{W}_i, \end{aligned} \tag{31}$$

$$\text{or} \quad W_i^* = \arg\max_{W_i \in \mathcal{W}_i} S_i(W_1^*, \dots, W_{i-1}^*, W_i, W_{i+1}^*, \dots, W_x^*). \tag{32}$$

The NE is of interest because it is a self-enforcing point from which the player has no incentive to deviate, and, hence, the system is stable.

In a real scenario, the terminals face two problems: there is no global information of the system available (in particular the number of competing terminals) and the estimates of the number of competing terminals are noisy. So, a terminal may attempt to reduce its window size to increase its transmission probability. This, in turn, will make the other terminals estimate a higher probability of collision, hence, increasing their estimate of the terminals competing in the network which, in turn, increases their CW_{min} and reduces their probability of transmission (they spend more time backlogged). The overall result is that the rogue terminal would capture most of the network throughput at the other terminals' cost. As a terminal has an incentive to deviate from the optimal strategy for its own benefit, the conclusion is that, *with partial information, the strategy in (26) is not an NE point in the distributed DCF game.* We need to extend the definition of our game model to an infinitely repeated game in which the varying strategies that the terminals take in time are taken into account.

Definition 3. *The infinitely repeated DCF game $G^r(\mu)$ is a staged game in which, at any stage t, each player decides which strategy to use and receives the corresponding payoff. In this sense, each stage of the game is another game as described in Definition 1, denoted by $G(t)$. The payoff for player i up to stage t is defined as*

$$S_i^r(t) = S_i(1) + \mu S_i(2) + \mu^2 S_i(3) + \dots = \sum_{j=1}^{t} \mu^{j-1} S_i(j), \tag{33}$$

where $S_i^r(t)$ is the payoff of the repeated game after t stages for player i, $S_i(j)$ is the payoff of the jth stage game for player i, and $\mu \in (0,1)$ is the discount factor.

Infinitely repeated games are often used to analyze games that will end with probability one, but there is uncertainty as to when this will happen. So, it is reasonable to try to maximize $S_i^r(\infty)$ instead of a particular (set of) $S_i(t)$. This concept of an infinitely repeated game describes the DCF problem more accurately. The set of strategies at every stage of the game depend on the observed behaviors of the rest of the players in previous stages. This also allows for the players to implement strategies that enforce some behavior in other players by threatening with a punishment strategy that would lower their payoff. The worst punishment strategy for terminal i would be the rest of the terminals implementing the following strategy

$$W_j^i = \arg\min_{W_j} \max_{W_i} S_i(t), \quad \forall j \neq i, \tag{34}$$

i.e., the strategy that minimizes the maximum payoff of the terminal i. In our DCF case, the worst punishment is given when the rest of the terminals select their strategies to be $W_j^i = \min\{\mathcal{W}\}$, i.e., the minimum of the set of available window sizes (this would correspond to the "all cheat" data in Fig. 3b). We now recall a well-known theorem in game theory called the *Folk Theorem*, which states that any combined strategy that gives each player a better payoff than the worst punishment can be implemented as an NE [17] as long as the players stick to it. The rationale behind the theorem is that, when a terminal is detected to be deviating from the agreed strategy (to have a temporal better payoff), the rest of terminals may implement a punishment strategy long enough so that the rogue terminal will eventually obtain less payoff over time than if it would have not deviated from the agreed strategy in the first place. It is obvious that the strategy we want to agree on with all the terminals is the one given in (26).

To make *Folk Theorem* applicable, though, the terminals must be able to tell when a rogue terminal is deviating from the agreed strategy. We argue that the detection of a rogue terminal is still possible with some probability. For a given number of terminals, and assuming no hidden terminal or capture effects, it is easy to see that all the terminals will have, on average, an equal share of the channel[3] and that share is known by the terminals as they use it for their optimal decisions. However, if a terminal is cheating, the expected throughput of the noncheating terminals will be reduced and the noncheating terminals can identify the presence of rogue terminals in the network. This detection is not straightforward and it is indeed very difficult to guarantee with total accuracy because, if a terminal reduces the window size, it will inject more packets into the network, affecting, in turn, the estimation of the number of competing terminals. However, as Fig. 3a shows, the effect of one cheater terminal does not match the expected behavior even

3. IEEE 802.11 DCF is known to be locally unfair, but we can assume that, on average, the throughput per terminal is equal.

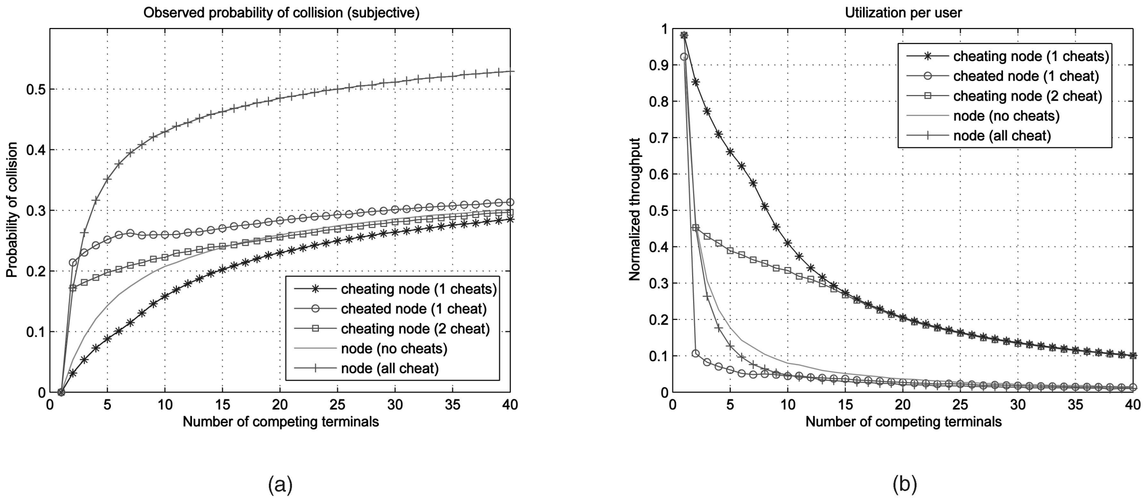


Fig. 3. Effect of cheating in IEEE 802.11 DCF. (a) Observed probability of collision when cheating. (b) Terminal utilization under cheating.

for the new observed "artificial" collision probability. In the figure, the rogue terminal is using the $CW_{min} = CW_{max} = 8$ window instead of the regular 32-1,024 window. If we assume that the estimate of the number of terminals is within some error, we can also estimate the deviation from the expected utilization and identify the existence of rogue terminals (especially for small number of terminals). This technique, however, does not work if the rogue terminal reduces the window size one step at a time, so the overall contribution of this decrement in the collision probability is below the estimation noise. On the other hand, if more than one terminal reduces the window size at the same time, then the probability of being detected increases. If we assume the above is true, then the we can say that there is a nonzero probability for the detection of a rogue terminal and we can make use of the *punishment strategy* to achieve the NE equilibrium.

Another problem with the punishment strategy is that it usually has a cost for the terminals that implement it in response to a rogue terminal. Fig. 3b shows that the utilization of all the terminals is reduced from the regular operation point ("no cheats") to the point in which everyone is implementing the punishment ("all cheat"). The idea of punishment strategy is to act as a "threat" to a potential cheating, but nothing prevents the rogue terminal from cheating if the punishment is not credible. To avoid this concept of *incredible threat*, it is important to show that a strategy is not only NE, but *subgame perfect Nash equilibrium*, in which the players' strategies in every subgame (subset of stages in the game) are also an NE. This means that not only is the agreed strategy NE, but the punishment strategy is also NE (the best that a cheated terminal can do). We claim, however, that, in the DCF game, the punishment strategy is credible. It is reasonable to assume that the relation between the throughput of each strategy as shown in Fig. 3b holds for small deviations. Then, we can get the final result in the following theorem:

Theorem 1. *Assume that the relation from Fig. 3b holds for all variations in window size, i.e., $S_r > S_o > S_a > S_c$, where S_r and S_c are the payoff vector (throughput) for the rogue terminal and the cheated terminals, respectively, when the rogue terminal is cheating; S_o is the payoff vector for all the terminals when none of them is cheating, and S_a is the payoff vector for all the terminals when everyone implements the punishment strategy. Then, the following punishment strategy W^* is a subgame perfect Nash equilibrium for the DCF game for an infinite-lived network:*
At stage t,

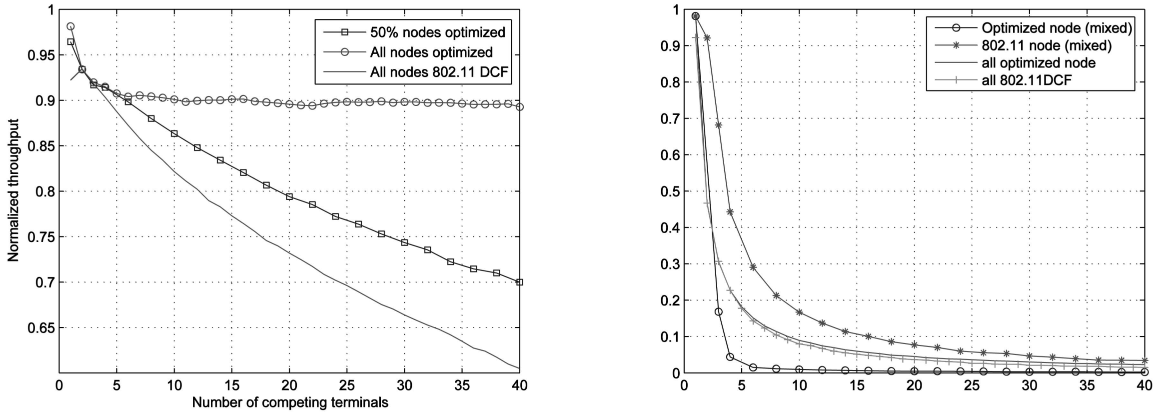
1. *select $W^* = W_t^{opt}$ in (26) if every terminal selects their optimal W^{opt}, i.e., there are no rogue terminals detected;*
2. *otherwise, select the punishment strategy $W^* = \min\{\mathcal{W}\}$ forever.*

Proof. The punishment strategy from the cheated terminals produces a reduction in the payoff vector for the rogue terminal ($S_a < S_r$) that at the same time is worse than the payoff vector for all the terminals if none of them had cheated ($S_a < S_o$). Then, by the Folk Theorem, there exists a discount factor μ, such that the strategy W^* is Nash equilibrium for $G^\infty(\mu)$. Moreover, the punishment strategy is the best strategy for a cheated terminal as $S_c < S_a$. And, once the punishment strategy is in place, there is no incentive for any terminal to modify the window selection, as any other $W \neq \min\{\mathcal{W}\}$ would produce a payoff vector S' such that $S' \leq S_a$. So, the punishment strategy is also NE and W^* is subgame perfect Nash equilibrium for the infinity repeated DCF game. □

The interesting result here is that the strategy defined in Theorem 1 is subgame perfect Nash equilibrium, meaning that for an infinite-lived network, there is no incentive for a rogue terminal to cheat, assuming that there exists a nonzero probability of a rogue terminal being detected. Moreover it also shows that the punishment strategy is not only a credible threat, but it is the best response a cheated terminal can have if a rogue terminal is detected. Finally, we can show that the punishment strategy described above is also optimal in terms of throughput.

Corollary 1. *The strategy W^* defined in Theorem 1 is optimal.*

Proof. Because W^* is subgame perfect Nash equilibrium, the optimal strategy for all the terminals is to follow Step 1 of the strategy and avoid entering into the punishment mode. As Step 1 selects an optimal W, the payoff vector for all terminals is also optimal. □



(a) (b)

Fig. 4. Coexistence of vanilla IEEE 802.11 with optimized terminals (50 percent versus 50 percent). (a) Total throughput of the network. (b) Terminal utilization.

Coexistence of regular IEEE 802.11 DCF terminals with optimized terminals implementing the punishment strategy. The previous result applies to scenarios in which it is assumed that the terminals can decide the value of CW_{min}. The coexistence regular IEEE 802.11 DCF terminals with terminals applying the optimization algorithm is worth discussing. Fig. 4a shows a scenario in which 50 percent of the terminals are regular 802.11 and 50 percent are using the optimization algorithm. The presence of optimized terminals always benefit the overall network throughput. The optimized terminals will observe a higher probability of collision and will increase their window sizes, reducing the overall collision probability by reducing their transmission probabilities (Fig. 4b). This creates an interesting situation from the Nash Equilibrium point of view: If the optimized terminals do not implement the punishment strategy, they "sacrifice" themselves at the cost of a better network throughput. This is not fair for the optimized terminals. However, if they implement the punishment strategy, they would reduce the performance of the regular terminals considerably, which, in turn, is not fair for the regular terminals. The optimized terminals would see the regular terminals as "cheaters," when, in fact, they can't select any window size at all. The Nash Equilibrium in this scenario occurs when all the terminals have the same window selection, i.e., optimized terminals revert to a standard IEEE 802.11 DCF operation, without optimizations. But it is impossible by simple observation of the medium to distinguish a vanilla IEEE 802.11 DCF terminal from a rogue node pretending to be one. A possible solution is to inform all the terminals of the presence of regular IEEE 802.11 DCF terminals, so all of them can fall into regular IEEE 802.11 DCF operation. This "backward compatibility mode" can be informed either by the BS or by identification of the protocol (for example, by a field in the frames).

7 Simulation Results

7.1 Simulation Setup

For the simulations, we use the ns-2 network simulator version 2.27 [18]. We modified the 802.11 implementation so the terminals measure the observation slots as in (7) for the estimates of the collision probability. The parameters used in the simulation are classical for a 1 Mbps WLAN and are taken after [3] for a fair comparison. No packet fragmentation occur, and the terminals are located close to each other to avoid capture or hidden terminal problems. The propagation delay is 1 μs. The packet size is fixed with a payload of 1,024 bytes. The MAC and PHY headers use respectively 272 and 128 bits. The PHY preamble takes 144 bits. The ACK length is 112 bits with an ACK timeout of 300 μs. The Rx/Tx turnaround time is 20 μs and the busy detect time 29 μs. The short retry limit and long retry limit are set to 7 and 4, respectively. Finally, the slot time is 50 μs, the SIFS is 28 μs, and the DIFS is 130 μs. The RTS/CTS threshold was increased so that only the basic access was used.

For the cases in which an analytical model is not available, empirical models can also be used. Fig. 2a shows the collision probability versus the number of competing terminals obtained empirically in the ns-2 simulator. Each point was obtained by simulating a fixed number of terminals transmitting under saturation conditions and measuring the total probability of collision. The simulation time for this empirical measurement lasted 3,000 seconds to provide better accuracy. To avoid including ARP packets in the measurement, an initial 20 seconds transmission was used to ensure all the terminals had updated ARQ tables. Finally, an additional 100 seconds transmission was added before measurements to allow the system to reach the steady state.

We assume that 40 is a reasonable upper limit for the number of competing terminals. To select the appropriate set $\mathcal{W}$ of backoff parameters, we measured the utilization of the IEEE 802.11 for different values of x_t. Our simulations showed that there is almost no impact in performance for $CW_{max} > 1{,}024$, so we fixed m such that $m = \log_2(1{,}024/W_{min})$. As the number of parameters need to be finite, we selected for CW_{min} the powers of 2 lower than 1,024, i.e.,

$$\mathcal{W} = \{(8,7),(16,6),(32,5),(64,4),(128,3),(256,2),(512,1)\}.$$

For the EKF estimator, we used the parameters suggested in [3]: The state variance Q_t is set to 0 except when a

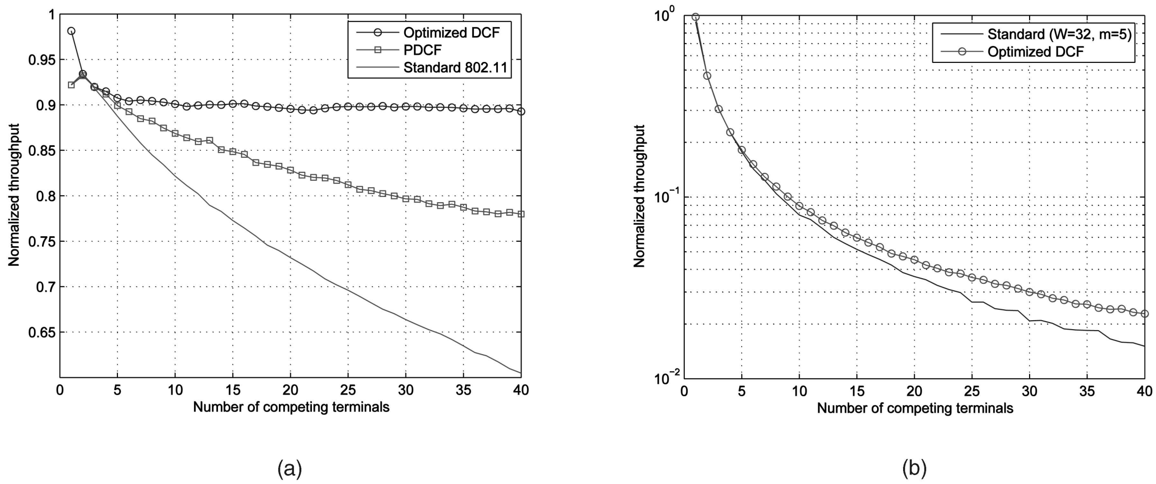


Fig. 5. Performance of the optimized algorithm with respect to the standard DCF. (a) Total throughput of the network. (b) Observed probability of collision.

change is detected where Q_t is set to $Q_{max} = 10$. The initial error variance $P_0 = 100$. R_t is known and given by the observation model. For the change detection filter, we used $v = 0.5$ and $h = 10$. These parameters were used by the authors in [3] for both the saturation and nonsaturation schemes and, thus, we used the same values in our simulations.

Our simulation scenario is composed of a variable number of competing terminals x_t transmitting in saturation conditions. Each ns-2 simulation run lasts between 300 and 1,000 seconds. The arrival and departure of competing terminals to the network (to attach to the corresponding access point) follows an on-off exponential process in continuous time.

7.2 Effect of the Adaptive Choice of Parameters on the DCF Optimization

As discussed in Section 1, the optimization of the IEEE 802.11 DCF based on the estimation of the number of competing terminals often trades off accuracy for complexity. However, we believe that given the sensitivity of the throughput to the number of competing terminals, as shown in Fig. 2b, especially in the 1-15 range, both the speed and accuracy of the estimator are crucial in order to rapidly select the optimal parameters to increase the network utilization. In this section, we evaluate the effect on the performance of the IEEE 802.11 DCF when the optimization scheme described in Section 5 is used. Fig. 2b shows the saturation throughput for our backoff parameters set $\mathcal{W}$.

In ns-2, we implemented the optimization algorithm described in Section 5 using the estimation algorithms described in Section 4. For comparison purposes, we also implemented an optimized version of PDCF [12], where the reset of the window to CW_{min} after successful transmission is 0.5 (on average, the best option for the 1-40 terminals range). Fig. 5a shows the normalized throughput of the optimization DCF with respect to the standard DCF of $W = 32, m = 5$ and the PDCF implementation. As the figure shows, the increase in efficiency is dramatic, the benefit of the optimized algorithm with respect to the regular IEEE 802.11 is as high as 40 percent for large values of x. The PDCF version is an example of a protocol that falls into the DCF-modification category discussed in Section 2. While a nonadaptive protocol can never outperform an adaptive one, it is interesting to see how simple modification of the existing vanilla DCF protocol results in a considerable throughput benefit. The actual average evolution of the actual window size v is shown in Fig. 6a and the normalized throughput of the network wasted in collisions is shown in Fig. 6b.

7.3 Instantaneous Network Utilization

In this section, we compare the performance of the estimation algorithms described in Section 4. Fig. 7a shows the instantaneous network utilization of the optimization protocols when the terminals follow the step arrival shown in Fig. 7b. Fig. 7b also shows the actual estimates for both the approximate MAP and the EKF algorithms. The estimation window size B is 100. Note that the accuracy of the estimate of the probability of collision is directly related to the value of B: A large B means more accuracy but also greater delay in the estimation. A smaller B provides a noisy measurement of y_t but a faster reactive estimation. The speed of the estimator may be crucial when the number of competing terminals oscillates in the 1–15 range, as the decision regions for the optimal contention window size are narrower, and the estimator may miss the optimal points. In the step case, the algorithms have time to detect and estimate the number of competing terminals. The expected results is a flat line of maximum throughput as the one shown in the perfect estimator, where the algorithm is fed with the actual number of competing terminals and not an estimate. The nonadaptive algorithms fall in throughput after the increment in the number of terminals. The approximate MAP algorithm outperforms the EKF algorithm in the estimates and, hence, the positive effect in the network performance.

On the other hand, Fig. 8a shows the instantaneous utilization of the protocols when the terminals follow an exponential on-off activation with parameter 10 seconds. A terminal is active for an exponential time with $\lambda = 10$ seconds and then deactivates for an exponential time of $\lambda = 10$ seconds. The evolution of the number of

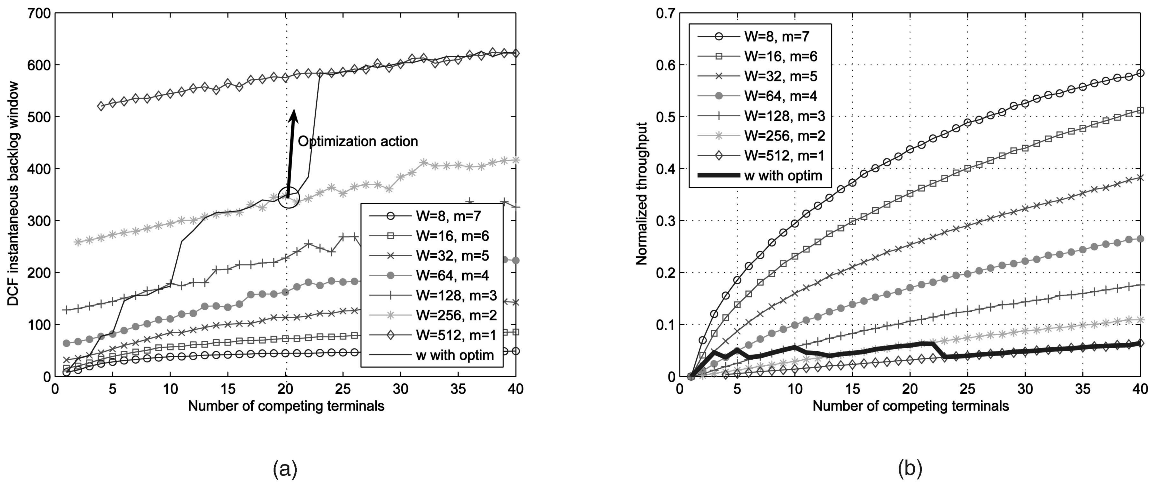


Fig. 6. Performance of the optimized algorithm. (a) Evolution of the instantaneous backoff window versus number of computing terminals for fixed backoff parameters. (b) Normalized throughput wasted in collisions.

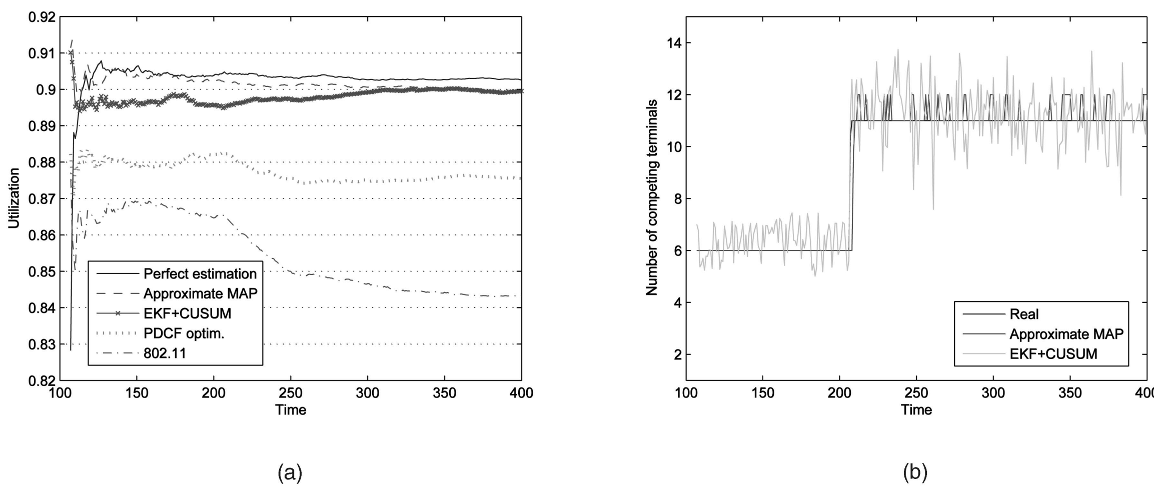


Fig. 7. Instantaneous utilization when terminals arrival has a step form. (a) Instantaneous utilization. (b) Evolution of the number of competing terminals.

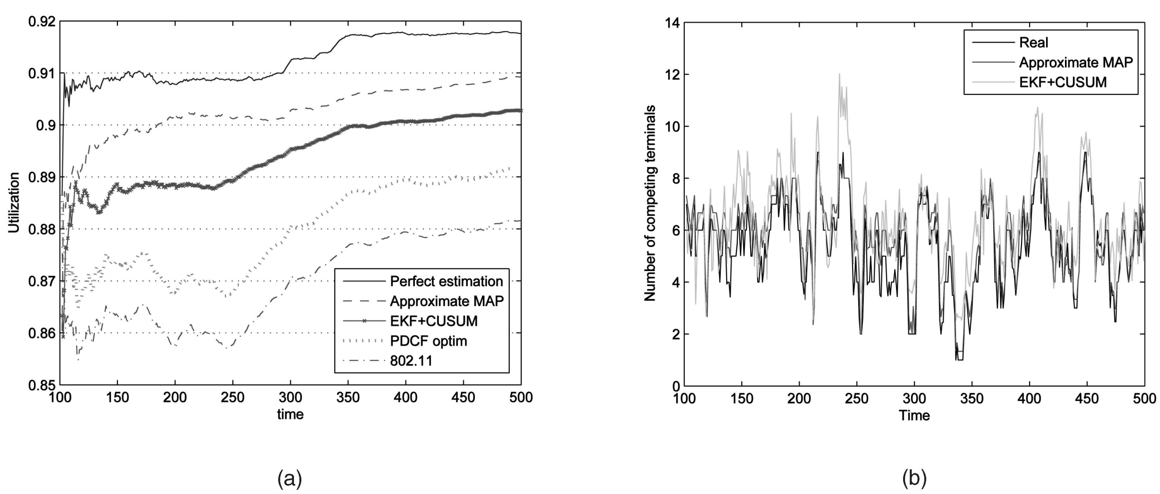


Fig. 8. Instantaneous utilization when terminals arrive exponentially. (a) Instantaneous utilization. (b) Evolution of the number of competing terminals.

competing terminals and the estimates of both the approximate MAP and EKF algorithms are shown in Fig. 8b. The fast and accurate tracking capabilities of the approximate MAP are evident and its MSE is 14.1483 while the MSE of the EKF algorithm is 28.3253. We want to compare the effect of the estimation in time for $B = 50$, to keep the estimation within the granularity of the change in the number of terminals. We used an optimized version of the PDCF protocol for the range of 1-10 terminals (reset probability is 0.9). Note that the

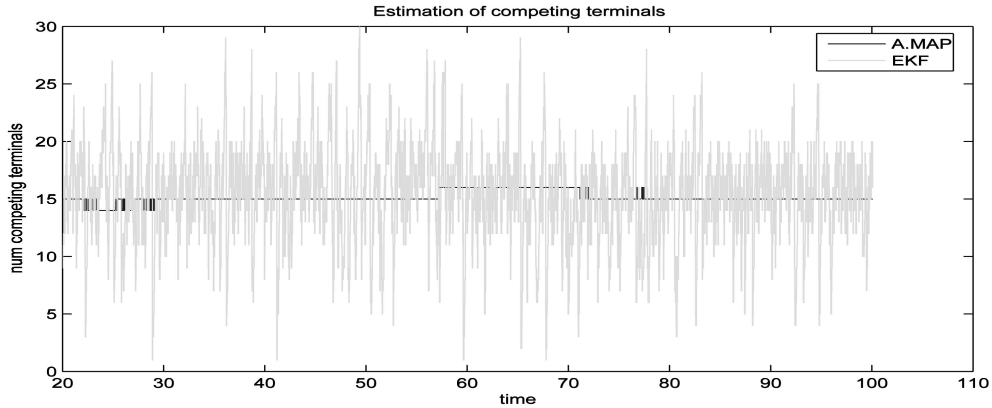


Fig. 9. Accuracy of the estimation algorithms for the extreme case of very noisy measurements ($B = 10$). The number of competing terminals is 15.

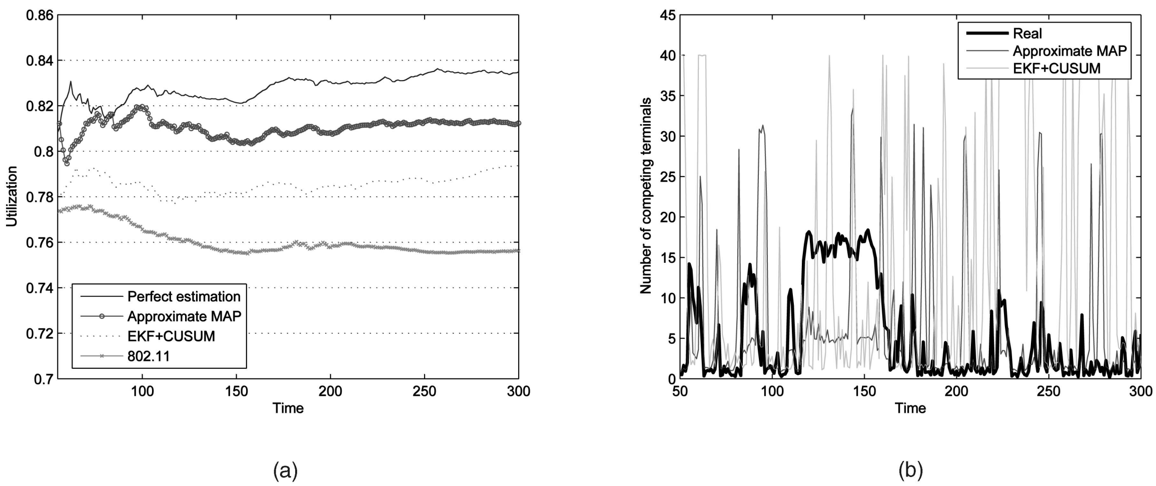


Fig. 10. Instantaneous utilization. (a) Instantaneous utilization when terminals do not saturate. (b) Evolution of the number of competing terminals.

estimation algorithms never outperform the perfect estimation at any point. This is an indication of the benefit of the accurate estimation when optimizing the DCF. A protocol that simplifies the estimations will necessarily fall apart from the perfect curve. Moreover, the estimation protocols take some time to converge to their optimal operation. At time $t = 100$s, the optimization algorithms start their operation and take between 10 and 20 seconds to converge. Note that the approximate MAP algorithm outperforms both the EKF and the modified PDCF algorithm at all times, which is an indication of the benefits of its accuracy in the IEEE 802.11 operation.

7.4 Results under Nonsaturated Network Conditions

A common problem of the estimation mechanisms described in Section 2 is that they base their estimations on the fact that the network is in saturation mode, i.e., at any given time, the terminals always have something to transmit. As [3] shows, the number of competing terminals fluctuates heavily under nonsaturation conditions. As a rude approximation, and intuitively, we can think of n terminals in nonsaturation regime as a process of $x(t)$ saturating terminals (those that have something to transmit in the allowed slots) that fluctuates very fast. In this scenario, the effect of a highly accurate and fast estimate of the number of competing terminals may be crucial to the optimal operation of the protocol. We tested the accuracy of both the EKF and our approximate MAP estimator in a very simple scenario: The number of competing terminals is fixed to 15, all of them saturating, and we reduced the observation slot $B = 10$. Note that $B = 10$ means that the average time for which the terminals measure the channel before estimation averages less than 300ms. Fig. 9 shows that our estimator is not only more accurate than the EKF estimator at very low time scales, it is also potentially able to better track fast fluctuations. Figs. 10a and 10b show the instantaneous utilization and the evolution of the number of competing terminals when 20 terminals are not in saturation regime. Each terminal randomly picked a throughput between 70-100 percent of 1/20th of the network saturation throughput. As we see in Fig. 10b, both estimators have problems in tracking the small fluctuations in the number of competing terminals. However, our estimator clearly does a better job, with an MSE of 154.4874 against an MSE value of 322.4615 for the EKF estimator. This difference makes our algorithm clearly superior in the nonsaturation regime.

8 Conclusions

In this paper, we have proposed a new scheme for optimizing the operation of the IEEE 802.11 DCF by adjusting the contention window parameters based on estimating the number of competing terminals in the network. We have employed a powerful yet computationally simple online algorithm to estimate the number of competing terminals based on the sequential Monte Carlo method. Moreover, its low computational requirements make it a good candidate for its introduction in an actual IEEE 802.11 network. We have provided extensive ns-2 simulation results and have shown that the proposed technique outperforms existing state-of-the-art approaches in all cases. Finally, we have shown a subgame perfect Nash equilibrium strategy for the completely distributed version of the protocol that prevents rogue terminals from changing their parameters for their own benefit.

As a main result, we have shown that the accuracy of the estimation of the number of competing terminals in an 802.11 network has a significant impact on the network performance: in terms of overall network utilization and in terms of observed delay due to collisions. This accuracy is shown to be extremely important when the number of competing terminals fluctuates heavily in small time scales as in the case when the network is in a nonsaturation regime. Consequently, a fast and accurate estimation of the number of competing terminals offers a great benefit toward optimizing the operation of an IEEE 802.11 DCF by adjusting the contention window parameters to the existing network conditions.

Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) under grant CCR-0207550 and in part by the US Office of Naval Research (ONR) under grant N00014-03-1-0039. Alberto Lopez Toledo was supported by Fundación "La Caixa."

References

[1] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *ANSI/IEEE Std 802.11, 1999 Edition (R2003),* pp. i-503, 2003.

[2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 3, pp. 535-547, Mar. 2000.

[3] G. Bianchi and I. Tinnirello, "Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 Network," *Proc. Infocom 2003,* vol. 2, pp. 844-852, Mar. 2003.

[4] C. Wang, B. Li, and L. Li, "A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF," *IEEE Trans. Vehicular Technolgy,* vol. 53, no. 4, pp. 1235-1246, 2004.

[5] F. Cali, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Networking,* vol. 8, no. 6, pp. 785-799, 2000.

[6] M. Natkaniec and A.R. Pach, "An Analysis of the Backoff Mechanism Used in IEEE 802.11 Networks," *Proc. IEEE Symp. Computers and Comm.,* vol. 1, pp. 444-449, 2000.

[7] Z.J. Haas and J. Deng, "On Optimizing the Backoff Interval for Random Access Schemes," vol. 51, no. 12, pp. 2081-2090, Dec. 2003.

[8] T. Vercauteren, A. Lopez Toledo, and X. Wang, "Online Bayesian Estimation of Hidden Markov Models with Unknown Transition Matrix and Applications to IEEE 802.11 Networks," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '05),* vol. 4, pp. 13-16, Mar. 2005.

[9] *Sequential Monte Carlo Methods in Practice,* A. Doucet, N. de Freitas, and N. Gordon, eds. Springer-Verlag, 2001.

[10] A. Lopez Toledo, T. Vercauteren, and X. Wang, "Optimizing IEEE 802.11 DCF Using Bayesian Estimators of the Network State," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '05) (Invited),* vol. 5, pp. 933-936, Mar. 2005.

[11] Y. Kwon, Y. Fang, and H. Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," *Proc. Infocom 2003,* vol. 2, pp. 853-862, Apr. 2003.

[12] C. Wang and W. Tang, "A Probability-Based Algorithm to Adjust Contention Window in IEEE 802.11 DCF," *ICCCAS: Comm., Circuits and Systems,* vol. 1, pp. 418-422, 2004.

[13] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 9, pp. 1774-1786, 2000.

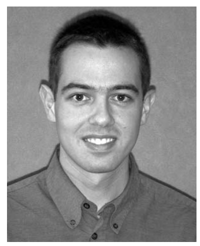
[14] Y. Kwon, Y. Fang, and H. Latchman, "Design of MAC Protocols with Fast Collision Resolution for Wireless Local Area Networks," *IEEE Trans. Wireless Comm.,* vol. 3, no. 3, pp. 793-807, May 2004.

[15] H. Ma, X. Li, H. Li, P. Zhang, S. Luo, and C. Yuan, "Dynamic Optimization of IEEE 802.11 CSMA/CA Based on the Number of Competing Stations," *Proc. IEEE Int'l Conf. Comm., (ICC '04),* June 2004.

[16] H. Ma, X. Li, H. Li, P. Zhang, S. Luo, and C. Yuan, "Range Estimation and Performance Optimization for IEEE 802.11 Based on Filter," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '04),* vol. 3, pp. 1469-1475, Mar. 2004.

[17] R. Gibbons, *Game Theory for Applied Economists.* Princeton Univ. Press, 1992.

[18] Network Simulator 2, http://www.isi.edu/nsnam/ns, 2006.



Alberto Lopez Toledo graduated in computer engineering (with highest honors) from the University of Murcia (UMU), Spain, in 1999, and received the MPhil degree in computer science from the same university in 2002. He received the MS degree in electrical engineering from Columbia University, New York, in 2003, where he is currently a PhD candidate in electrical engineering. From September 1999 to August 2002, he was with the Department of Telematic Systems Engineering at the Technical University of Madrid (UPM), Spain. His research interests are in the areas of wireless networking and cross-layer design. He received Spain's National Academic Excellence Award, the Edwin Howard Armstrong Memorial Award, and the La Caixa and Rafael del Pino Foundation fellowships.

Tom Vercauteren graduated from École Polytechnique, Paris, France, in 2003 and received the MS degree from the Department of Electrical Engineering, Columbia University, New York, in 2004. He is now working toward the PhD degree at INRIA, Sophia-Antipolis, France. His research interests are in the area of statistical signal processing.

Xiaodong Wang (S'98-M'98-SM'04) received the BS degree from Shanghai Jiao Tong University, Shanghai, China, the MS degree from Purdue University, and the PhD degree from Princeton University, all in electrical engineering. From July 1998 to December 2001, he was on the faculty of the Department of Electrical Engineering, Texas A&M University. Since January 2002, he has been with the Department of Electrical Engineering, Columbia University. Dr. Wang's research interests fall in the general areas of computing, signal processing, and communications, and he has published extensively in these areas. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Prentice Hall, 2003). His current research interests include wireless communications, statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 US National Science Foundation CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an associate editor for the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Signal Processing*, and the *IEEE Transactions on Information Theory*.