Batch and Sequential Bayesian Estimators of the Number of Active Terminals in an IEEE 802.11 Network

Tom Vercauteren, *Student Member, IEEE*, Alberto Lopez Toledo, *Student Member, IEEE*, and Xiaodong Wang, *Senior Member, IEEE*

***Abstract*—The performance of the IEEE 802.11 protocol based on the distributed coordination function (DCF) has been shown to be dependent on the number of competing terminals and the backoff parameters. Better performance can be expected if the parameters are adapted to the number of active users. In this paper we develop both off-line and online Bayesian signal processing algorithms to estimate the number of competing terminals. The estimation is based on the observed use of the channel and the number of competing terminals is modeled as a Markov chain with unknown transition matrix. The off-line estimator makes use of the Gibbs sampler whereas the first online estimator is based on the sequential Monte Carlo (SMC) technique. A deterministic variant of the SMC estimator is then developed, which is simpler to implement and offers superior performance. Finally a novel approximate maximum *a posteriori* (MAP) algorithm for hidden Markov models (HMM) with unknown transition matrix is proposed. Realistic IEEE 802.11 simulations using the ns-2 network simulator are provided to demonstrate the excellent performance of the proposed estimators.**

***Index Terms*—Gibbs sampler, hidden Markov model (HMM), IEEE 802.11 wireless networks, sequential Monte Carlo, unknown transition matrix.**

I. Introduction

THE performance of the IEEE 802.11 DCF [1] is known to be very sensitive to the number of users competing to access the wireless channel [2], [3]. The problem of estimating the number of competing terminals has been addressed recently in [3]. The approach there is based on an extended Kalman filter assuming a constant number of users, and coupled with a change detection mechanism. Such an algorithm implicitly uses a linear Gaussian dynamic model where the variables (number of competing terminals and number of busy slots) are assumed to be continuous. Such underlying assumptions are unrealistic since the number of users in the network only takes discrete integer values and the observations used for the estimation are also integer values. In this paper, it is assumed that the number of users in the network evolves according to a Markov chain with unknown transition probability matrix. Such an assumption can only be validated through a thorough analysis of real data traffic and is unlikely to be fully satisfied. However it allows us to use the fact that both the state and the observations are discrete and to develop solutions that can be implemented on an IEEE 802.11 wireless card. Our simulations do not strictly make use of the Markovian assumption. Indeed our ns-2 simulator uses an exponential On-Off activation process in continuous time which is not Markovian anymore in the irregular discrete time scale developed in [2] and used in this work. The active nodes are saturating and the departures are done according to a realistic IEEE 802.11 protocol. The estimation is then based on the observed collision probabilities in the shared wireless channel.

Bayesian Monte Carlo signal processing techniques [4], [5] offer a paradigm for tackling challenging signal processing problems for which traditional methods are difficult to apply. Two categories of techniques are available: Markov chain Monte Carlo (MCMC) methods [4] for batch signal processing and sequential Monte Carlo (SMC) methods [5], [6] for adaptive signal processing. Both MCMC and SMC have been applied to solve a number of important and challenging signal processing problems found in wireless communications [7]. In this paper, we develop both off-line and online Bayesian Monte Carlo algorithms for estimating the number of users in an IEEE 802.11 wireless network as well as the unknown parameters. While usual SMC methods are not well suited to parameter estimation, we show that the complete information about the transition matrix can be carried over through some sufficient statistics so that the algorithm developed in [8] can be adapted to our hidden Markov model (HMM) problem.

A deterministic variant of the SMC estimator is also developed, which is simpler to implement and offers superior performance. The idea of using a set of sufficient statistics to represent the information we have about the transition matrix is included in the deterministic sample filter setting proposed in [9]. The use of sufficient statistics is pushed one step further than in [8] because this information about the parameters is now integrated out so that no Monte Carlo approximation needs to be performed. The exponential increase in complexity is avoided by discarding the tails of the posterior distributions. For some applications, this algorithm might still be somewhat too computationally demanding. Inspired from the deterministic sequential sampling algorithm, we develop an approximate MAP algorithm that trades accuracy for computational requirement. Both algorithms can be applied to any HMM with unknown transi-

Manuscript received February 13, 2005; revised March 27, 2006. This work was supported in part by the U.S. National Science Foundation (NSF) by Grant DMS 0244583, and in part by the U.S. Office of Naval Research (ONR) by Grant N00014-03-1-0039. The work of A. Lopez Toledo was supported byFundación "La Caixa." The associate editor coordinating the review of this paper and approving it for publication was Prof. Simon J. Godsill.

T. Vercauteren is with INRIA Sophia Antipolis, Asclepios Research Project, 06902 Sophia Antipolis Cedex, France.

A. Lopez Toledo and X. Wang are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: wangx@ee.columbia.edu).

Digital Object Identifier 10.1109/TSP.2006.885723

tion probabilities (and unknown prior distribution) and these are main contributions of our work. The online algorithms led to an approximation of the probability distribution function (or to a hard estimate for the approximate MAP algorithm) of the number of competing terminals at a specific time step given the entire set of observations.

The remainder of this paper is organized as follows. In Section II, we give the mathematical formulation for the problem of estimating the number of active users in an IEEE 802.11 network. In Sections III and IV, we develop the off-line MCMC and the online SMC estimators, respectively. The deterministic sequential sampling and the approximate MAP algorithms are derived in Section V. The performance of the estimators is evaluated in Section VI by using both model-based data and realistic IEEE 802.11 ns-2 simulations. Finally, Section VII concludes the paper.

II. System Model

In [3], it is shown that the number x of competing terminals in an IEEE 802.11 wireless network can be expressed as a function of the packet collision probability p_c in the shared channel. We briefly review the analysis performed in [2] to provide the necessary background on IEEE 802.11.

With the basic access mechanism, a station with a new packet to send monitors the channel. If the medium is idle for at least a Distributed Inter Frame Space (DIFS), the station transmits. In the other cases (medium sensed busy or packet not new), the station waits until the channel is idle for a DIFS and sets its backoff timer to a discrete value uniformly chosen among $[0,\ldots,CW-1]$ where CW is the contention window size. The transmission starts when the timer reaches zero. The idle time is divided into slots of length σ. For the first transmission attempt, CW is set to the minimum contention window W. After each unsuccessful attempt, the contention window is doubled until $CW = 2^m W$, where $2^m W$ is the maximum contention window size. The backoff timer is decremented whenever the channel is sensed idle and frozen when the channel is busy. The timer is reactivated as soon as the channel is sensed idle for at least a DIFS. Since a station cannot sense the channel while it is transmitting, in case of a collision the transmission is not interrupted before the end of the packet. In case of a successful transmission, the acknowledgment (ACK) is sent after a Short Inter Frame Space (SIFS) which is considered as part of the successful transmission. From the point of view of a station, the time can now be slotted into variable length slots. Specifically, one time slot will either correspond to an idle slot of length σ or a busy slot of length $T_s(P)$ for a successful transmission and $T_c(P^*)$ for a collision

$$\begin{cases} T_s(P) = H + P + \mathrm{SIFS} + \delta_{\mathrm{prop}} + \mathrm{ACK} + \mathrm{DIFS} + \delta_{\mathrm{prop}} \\ T_c(P^*) = H + P^* + \mathrm{DIFS} + \delta_{\mathrm{prop}} \end{cases} \tag{1}$$

where P is the packet payload, P^* is the length of the longest packet involved in a collision, $H = \mathrm{PHY}_{\mathrm{hdr}} + \mathrm{MAC}_{\mathrm{hdr}}$ is the packet header, and δ_{prop} is the propagation delay.

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

$$x = f(p_c) \triangleq 1 + \frac{\log(1-p_c)}{\log\left(1 - \frac{2(1-2p_c)}{(1-2p_c)(W+1)+p_c W(1-(2p_c)^m)}\right)} \tag{2}$$

where W and m are the exponential backoff parameters. Each terminal can measure the packet collision probability by monitoring the channel activity, and can thus estimate the number of competing terminals in the network.

The monitoring procedure for estimating the packet collision probability p_c is as follows. Since in a busy time slot a packet transmission would eventually fail, information about the packet collision probability can be obtained by counting the number of experienced collisions C_{coll}, as well as the number of observed busy slots C_{busy} within one *observation slot* composed of B basic time slots in which the measurements are taken: $y_t = C_{\mathrm{busy}} + C_{\mathrm{coll}}$. More specifically, within each observation time step t, y_t is measured as

$$y_t = \sum_{i=(t-1)B}^{tB-1} C_i \tag{3}$$

where the indicator C_i is given by $C_i = 0$ if the i^{th} basic time slot is empty or corresponds to a successful transmission, and $C_i = 1$ if the i^{th} basic time slot is busy or corresponds to an unsuccessful transmission. We know that $p(C_i = 1) = 1 - p(C_i = 0) = p_c$, therefore, y_t follows a binomial distribution $\mathcal{B}(B, p_c)$

$$p(y_t = b) = \mathcal{B}(b; B, p_c) = \binom{B}{b} p_c^b (1-p_c)^{B-b}, \quad b \in [0, B]. \tag{4}$$

From now on a time slot will refer to an observation time slot (composed of B basic time slots).

The number of competing terminals x_t at time $t = 1, 2, \ldots$, takes value from the set $\mathcal{X}$. In wireless LAN systems, admission control is always performed to maintain certain quality of service (QoS) and thus $\mathcal{X}$ is a finite set. Typically, we have $\mathcal{X} = [1,\ldots,N]$, with N being the maximum number of users, so that we do not need to differentiate between the index of the states and the states themselves. We assume that x_t evolves according to a first-order Markov chain with a 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 $\sum_{j=1}^{N} a_{i,j} = 1$. We denote the initial probability vector as $\boldsymbol{\pi} = [\pi_1,\ldots,\pi_N]$, i.e., $p(x_0 = i) = \pi_i$.

1) The Inference Problem: From the above discussion, we can cast our problem into a hidden Markov model (HMM) with unknown parameters

$$x_t \sim \mathcal{MC}(\boldsymbol{\pi}, \boldsymbol{A}), \quad y_t \sim \mathcal{B}(B, h(x_t)) \tag{5}$$

where $\mathcal{MC}(\boldsymbol{\pi}, \boldsymbol{A})$ denotes a discrete-time Markov chain with the initial probability distribution $\boldsymbol{\pi}$ and the transition probability matrix $\boldsymbol{A}$; x_t is the state realization of the Markov chain at time instant t; $\mathcal{B}(B, p)$ denotes a binomial distribution with B trials and probability of success p; y_t is the observation; $h(\cdot) \triangleq f^{-1}(\cdot)$ where $f(\cdot)$ is given in (2).

Denote the observation sequence up to time t as $\boldsymbol{y}_t \triangleq [y_1, y_2, \ldots, y_t]$ and the network state sequence up to time t as $\boldsymbol{x}_t \triangleq [x_1, x_2, \ldots, x_t]$. Let the model parameters be $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}\}$. Given the observations $\boldsymbol{y}_t$ at time t, we are interested in estimating the current state x_t. The computational complexity of the naive solutions to these problems grows exponentially as the time index t increases. The forward-backward procedure [10], [11] provides a recursive algorithm to get a linear complexity growth but is not able to cope with unknown parameters. The expectation-maximization (EM) algorithm deals with such a problem but only converges to a local maximum of the likelihood function which can be quite different from the global maximum [10]. Other techniques for online identification of HMM based on maximum likelihood methods, least square methods, prediction error or ensemble learning can be found in [12]–[16]. In this paper, we resort to the Monte Carlo signal processing techniques to solve the above inference problems. We then show that from the Monte Carlo techniques we are able to derive a deterministic sequential sampling algorithm and an approximate MAP algorithm. In both approaches the integrations are done analytically. This provides two novel and powerful algorithms for the filtering and estimation of HMM with unknown parameters.

III. The Off-Line Estimator

We provide the basis of the Bayesian approach and show how the Gibbs sampler can be used to obtain an off-line approximation of the joint posterior probability distribution function of the state sequence and parameters.

A. The Gibbs Sampler

Let $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_m)$, where ν_i is either a scalar or a vector, be the unknown variables to be estimated; and $\boldsymbol{y}$ the available observations. We are interested in the marginal distribution $p(\nu_i|\boldsymbol{y})$. Directly evaluating the marginal distribution involves integrating out the rest of the parameters from the joint posteriori distribution $p(\boldsymbol{\nu}|\boldsymbol{y})$, which is computationally infeasible especially when the parameter dimension m is large. The basic idea behind the Gibbs sampler is to generate Monte Carlo samples from the joint posterior distribution $p(\boldsymbol{\nu}|\boldsymbol{y})$ and then to estimate any marginal distribution using these samples. Given the samples at iteration $(k-1)$, $\boldsymbol{\nu}^{(k-1)} = \left(\nu_1^{(k-1)}, \ldots, \nu_m^{(k-1)}\right)$, at the k^{th} iteration, the Gibbs sampler algorithm can be implemented as follows.

• For $i = 1, \ldots, m$, draw $\nu_i^{(k)}$ from the conditional distribution

$$p\left(\nu_i|\nu_1^{(k)}, \ldots, \nu_{i-1}^{(k)}, \nu_{i+1}^{(k-1)}, \ldots, \nu_m^{(k-1)}, \boldsymbol{y}\right). \tag{6}$$

It is known that under regularity conditions [17]–[20], the distributions of the samples $\boldsymbol{\nu}^{(k)}$ drawn by the above Gibbs sampler converge geometrically to $p(\boldsymbol{\nu}|\boldsymbol{y})$, as $K \to \infty$. Moreover, we have

$$(1/K)\sum_{k=1}^{K} \varphi\left(\boldsymbol{\nu}^{(k)}\right) \xrightarrow{a.s.} \int \varphi(\boldsymbol{\nu}) p(\boldsymbol{\nu}|\boldsymbol{y}) d\boldsymbol{\nu}$$

as $K \to \infty$, for any integrable function $\varphi(\cdot)$.

B. The Off-Line Batch Estimation Algorithm

In batch processing, we assume that observations corresponding to T time slots $\boldsymbol{y}_T = [y_1, \ldots, y_T]$ are available, based on which we need to estimate the system state $\boldsymbol{x}_T = [x_1, \ldots, x_T]$ corresponding to these time slots, as well as the system parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}\}$. For the Gibbs sampler discussed above, we need to be able to compute and draw samples from the prior and conditional posterior distributions. Modeling realistic priors for a particular application is a difficult task. It is therefore common to choose priors that convey little prior knowledge and/or ease the calculations. A well-known strategy for MCMC computation is to choose the prior distributions with a suitable form so that the posteriors belong to the same functional family as the priors. The priors and posteriors are then said to be conjugate [21]. The choice of the functional family depends on the likelihood. We will use this *conjugate* strategy throughout the paper.

1) Prior Distributions: Denote $\boldsymbol{a}_i \triangleq [a_{i,1}, \ldots, a_{i,N}]$ the i^{th} row of the state transition probability matrix $\boldsymbol{A}$, $i = 1, \ldots, N$. It can be seen here that the discrete states x_t are drawn from multinomial distributions. For this kind of likelihood functions it is well known that the Dirichlet distribution provides conjugate priors. We will, therefore, assume multivariate Dirichlet priors for both the initial probability vector $\boldsymbol{\pi}$ and $\boldsymbol{a}_i$.

The multivariate Dirichlet distribution $\mathcal{D}(\gamma_1, \ldots, \gamma_N)$ with strictly positive shape parameters $\gamma_1, \gamma_2, \ldots, \gamma_N$ has the following density function:

$$p(\boldsymbol{u}) = \left(\Gamma\left(\sum_{i=1}^{N}\gamma_i\right) \Big/ \prod_{i=1}^{N}\Gamma(\gamma_i)\right) \cdot \prod_{i=1}^{N} u_i^{\gamma_i - 1}$$

where $\Gamma(\cdot)$ is the Gamma function and $\boldsymbol{u} = [u_1, \ldots, u_N]$ is such that $u_i \geq 0$ and $\sum_{i=1}^{N} u_i = 1$. It is easy to draw samples from such a distribution by using Gamma distributed samples. The prior distributions for the unknown parameters and network states are, thus, as follows.

1. The prior distribution for the initial probability vector $\boldsymbol{\pi}$ is given by

$$\boldsymbol{\pi} \sim \mathcal{D}(\rho_1, \rho_2, \ldots, \rho_N). \tag{7}$$

2. The prior distribution for the i^{th} row of the transition probability matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{a_i} \sim \mathcal{D}(\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,N}) \quad i = 1, \ldots, N. \tag{8}$$

Note that for large N, it is common to assume that the matrix $\boldsymbol{A}$ is banded; i.e., $a_{i,j} = 0$ for $|i-j| > \delta$. Correspondingly, with some notational abuse, in the prior distribution of $\boldsymbol{a}_i$, we set $\alpha_j = -\infty$, if $|i-j| > \delta$. The resulting Dirichlet distribution then has a reduced dimension, i.e., $(a_{i,i-\delta}, \ldots, a_{i,i}, \ldots, a_{i,i+\delta}) \sim \mathcal{D}(\alpha_{i,i-\delta}, \ldots, \alpha_{i,i}, \ldots, \alpha_{i,i+\delta})$.

3. Finally, the prior distribution for the network state x_t is imposed by our choice of (7)–(8). It can be sampled from its prior distribution by using the samples $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{A}^{(0)}$ of $\boldsymbol{\pi}$ and $\boldsymbol{A}$

$$x_t \sim \mathcal{MC}(\boldsymbol{\pi}^{(0)}, \boldsymbol{A}^{(0)}). \tag{9}$$

2) Conditional Posterior Distributions: Based on our model and our choice of prior distributions, we get the following conditional posterior distributions (using Bayes rules and Markovian assumptions):

1. The conditional posterior distribution of the initial probability vector $\boldsymbol{\pi}$

$$\begin{aligned} &p(\boldsymbol{\pi}|\boldsymbol{y}_T, \boldsymbol{A}, \boldsymbol{x}_T) \\ &= p(\boldsymbol{\pi}|x_0) \propto p(x_0|\boldsymbol{\pi})\, p(\boldsymbol{\pi}) \propto \pi_{x_0} \prod_{j=1}^{N} \pi_j^{\rho_j - 1} \\ &= \prod_{j=1}^{N} \pi_j^{\rho_j + \mathbb{I}(x_0 - j) - 1} \\ &= \mathcal{D}(\boldsymbol{\pi}; \rho_1 + \mathbb{I}(x_0 - 1), \rho_2 + \mathbb{I}(x_0 - 2) \\ &\quad \ldots, \rho_N + \mathbb{I}(x_0 - N)) \end{aligned} \tag{10}$$

where $\mathcal{D}(\cdot; \boldsymbol{\gamma})$ is the probability density function (pdf) of the Dirichlet distribution with parameters $\boldsymbol{\gamma}$.

2. The conditional posterior distribution of the i^{th} row $\boldsymbol{a}_i$ of the transition probability matrix

$$\begin{aligned} &p(\boldsymbol{a}_i|\boldsymbol{y}_T, \boldsymbol{\pi}, \eta, \boldsymbol{x}_T, \boldsymbol{a}_{-i}) \\ &= p(\boldsymbol{a}_i|\boldsymbol{x}_T, \boldsymbol{\pi}) \propto p(\boldsymbol{x}_T|\boldsymbol{a}_i, \boldsymbol{\pi})\, p(\boldsymbol{a}_i) \\ &\propto p(x_0|\boldsymbol{\pi}) \left[\prod_{t=1}^{T} p(x_t|x_{t-1}, \boldsymbol{a}_i)\right] \prod_{j=1}^{N} a_{i,j}^{\alpha_{i,j} - 1} \\ &\propto \pi_{x_0} \prod_{j=1}^{N} a_{i,j}^{n_{i,j}} \prod_{j=1}^{N} a_{i,j}^{\alpha_{i,j} - 1} \propto \prod_{j=1}^{N} a_{i,j}^{\alpha_{i,j} + n_{i,j} - 1} \\ &= \mathcal{D}(\boldsymbol{a}_i; \alpha_1 + n_{i,1}, \alpha_2 + n_{i,2}, \ldots, \alpha_N + n_{i,N}) \end{aligned} \tag{11}$$

where $n_{i,j}$ is the number of state transitions from state i to state j in $\boldsymbol{x}_T$ and $\boldsymbol{a}_{-i} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{i-1}, \boldsymbol{a}_{i+1}, \ldots, \boldsymbol{a}_N)$. Note that if $\boldsymbol{A}$ is banded, then, $n_{i,j} = 0$ for $|i - j| > \delta$.

3. The conditional posterior distribution of network state x_t

$$\begin{aligned} &p(x_t = i|\boldsymbol{y}_T, \boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{x}_{-t}) \propto p(y_t|x_t = i) \\ &\cdot p(x_t = i, \boldsymbol{x}_{-t}, \boldsymbol{\pi}, \boldsymbol{A}) \\ &\propto \begin{cases} \mathcal{B}(y_t; B, h(i)) \cdot a_{x_{t-1}, i} \cdot a_{i, x_{t+1}}, & \text{if } t \in [2, T-1] \\ \mathcal{B}(y_t; B, h(i)) \cdot \pi_i \cdot a_{i, x_{t+1}}, & \text{if } t = 1 \\ \mathcal{B}(y_t; B, h(i)) \cdot a_{x_{t-1}, i}, & \text{if } t = T \end{cases} \end{aligned} \tag{12}$$

where $\boldsymbol{x}_{-t} \triangleq [x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T]$, and $\mathcal{B}(\cdot; B, p)$ denotes a binomial pdf with B trials and probability p.

We can now outline the batch algorithm based on the Gibbs sampler.

Algorithm 1 [Gibbs sampler batch estimator]

- *Draw the initial samples* $\left\{\boldsymbol{\pi}^{(0)}, \boldsymbol{A}^{(0)}, \boldsymbol{x}_T^{(0)}\right\}$ *from their prior distributions, given by* (7)–(9).
- *For* $k = 1, 2, \ldots, K_0 + K$,
  - *Draw a sample* $\boldsymbol{\pi}^{(k)}$ *from its conditional posterior distribution* $p\left(\boldsymbol{\pi}|\boldsymbol{y}_T, \boldsymbol{A}^{(k-1)}, \boldsymbol{x}_T^{(k-1)}\right)$ *given by* (10).
  - *For* $i = 1, 2, \ldots, N$, *draw a sample* $\boldsymbol{a}_i^{(k)} \triangleq [a_{i,1}^{(k)}, \ldots, a_{i,N}^{(k)}]$ *from its conditional posterior distribution* $p\left(\boldsymbol{a}_i|\boldsymbol{y}_T, \boldsymbol{\pi}^{(k)}, \boldsymbol{x}_T^{(k-1)}, \boldsymbol{a}_1^{(k)}, \ldots, \boldsymbol{a}_{i-1}^{(k)}, \boldsymbol{a}_{i+1}^{(k-1)}, \ldots, \boldsymbol{a}_N^{(k-1)}\right)$ *given by* (11).
  - *For* $t = 1, 2, \ldots, T$, *draw a sample* $x_t^{(k)}$ *from the conditional posterior distribution* $p\left(x_t|\boldsymbol{y}_T, \boldsymbol{\pi}^{(k)}, \boldsymbol{A}^{(k)}, x_1^{(k)}, \ldots, x_{t-1}^{(k)}, x_{t+1}^{(k-1)}, \ldots, x_T^{(k-1)}\right)$ *given by* (12).

This Gibbs iteration is usually carried out for $(K_0 + K)$ iterations with the first K_0 as the burn-in period. Only the samples from the last K iterations are used to make the inference. The *a posteriori* probabilities of the network states are computed from the K random samples of x_t as

$$\hat{p}(x_t = i|\boldsymbol{y}_T) = \frac{1}{K} \sum_{k=K_0+1}^{K_0+K} \mathbb{I}\left(x_t^{(k)} - i\right), \quad i = 1, \ldots, N, \quad t = 1, \ldots, T \tag{13}$$

where $\mathbb{I}(\cdot)$ is an indicator such that $\mathbb{I}(x) = 1$ if $x = 0$, and $\mathbb{I}(x) = 0$ if $x \neq 0$. Moreover, the model parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{A})$ can be estimated by

$$\widehat{\boldsymbol{\theta}} = \frac{1}{K} \sum_{k=K_0+1}^{K_0+K} \boldsymbol{\theta}^{(k)}. \tag{14}$$

IV. The Online SMC Estimator

In this section we will first briefly review the sequential Monte Carlo methodology and then show how sufficient statistics can be employed within this framework to deal with static parameters. We then provide the online SMC estimator of the number of competing terminals.

A. Background on Sequential Monte Carlo

We consider a generic dynamic model described by

$$\begin{aligned} \text{initial eqn} &: p_{\boldsymbol{\theta}}(x_0); \\ \text{state eqn} &: p_{\boldsymbol{\theta}}(x_t|x_{t-1}),\ \forall t \geq 1; \\ \text{measurement eqn} &: p_{\boldsymbol{\theta}}(y_t|x_t),\ \forall t \geq 1 \end{aligned} \tag{15}$$

where x_t and y_t are, respectively, the state and the observation at time t; $p_{\boldsymbol{\theta}}(\cdot)$ are some probability density functions depending on some static parameters $\boldsymbol{\theta}$ assumed known for the moment. Suppose we want to make online inference of the unobserved states $\boldsymbol{x}_t = (x_0, x_1, \ldots, x_t)$. That is, at current time t

we wish to make an estimate of a function of the state variable , say $\psi(\boldsymbol{x}_t)$, based on the currently available observations, $\boldsymbol{y}_t = (y_0, y_1, \ldots, y_t)$. The optimal solution (in terms of any common criterion) only depends on the conditional pdf $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$ (e.g., the minimum mean squared error estimator is the conditional mean).

Monte Carlo methods provide an approximation of this pdf based on K random samples

$$\left\{\boldsymbol{x}_t^{(k)}\right\}_{k=1}^{K}$$

from the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$. Since sampling directly from $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$ is often not feasible or computationally too expensive, the idea of importance sampling can be used to approximate $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$ by employing some trial sampling density $q_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$ from which we can easily draw samples from. Suppose a set of random samples

$$\left\{\boldsymbol{x}_t^{(k)}\right\}_{k=1}^{K}$$

has been drawn according to $q_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$. By associating the *importance weight* $w_t^{(k)} = p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(k)}|\boldsymbol{y}_t)/q_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(k)}|\boldsymbol{y}_t)$ to the sample $\boldsymbol{x}_t^{(k)}$, the posterior distribution of interest is approximated as

$$\hat{p}_\theta(\boldsymbol{x}_t|\boldsymbol{y}_t) = \frac{1}{W_t}\sum_{k=1}^{K} w_t^{(k)} \mathbb{I}(\boldsymbol{x}_t - \boldsymbol{x}_t^{(k)}) \tag{16}$$

where $W_t = \sum_{k=1}^{K} w_t^{(k)}$, and the set

$$\left\{\boldsymbol{x}_t^{(k)}, w_t^{(k)}\right\}_{k=1}^{K}$$

is called a set of *properly weighted samples* with respect to the target distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$ [5].

Suppose a set of properly weighted samples

$$\left\{\boldsymbol{x}_{t-1}^{(k)}, w_{t-1}^{(k)}\right\}_{k=1}^{K}$$

with respect to $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$ is available at time $(t-1)$. The SMC procedure generates a new set of samples and weights

$$\left\{\boldsymbol{x}_t^{(k)}, w_t^{(k)}\right\}_{k=1}^{K}$$

properly weighted with respect to $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$, from the previous set. In particular, if we choose the optimal trial distribution $q_{\boldsymbol{\theta}}(x_t|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t) = p_{\boldsymbol{\theta}}(x_t|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t)$ and if x_t can only take values from a finite set say $\mathcal{X} = \{b_1, b_2, \ldots, b_{|\mathcal{X}|}\}$, assuming $\boldsymbol{\theta}$ is known, then the SMC procedure is as follows: [5].

- For $i = 1, \ldots, \mathcal{X}$, compute $q_{\boldsymbol{\theta}}\left(x_t = b_i|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t\right) = p_{\boldsymbol{\theta}}\left(x_t = b_i|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t\right)$ up to a normalizing constant

$$q_{\boldsymbol{\theta}}(x_t = b_i|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t) \propto p_{\boldsymbol{\theta}}(y_t|x_t = b_i) p_{\boldsymbol{\theta}}(x_t = b_i|x_{t-1}^{(k)}). \tag{17}$$

- Normalize these values: $\sum_{b\in\mathcal{X}} q_{\boldsymbol{\theta}}(b|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t) = 1$.
- Draw a sample $x_t^{(k)}$ from the trial distribution $q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t)$ and let $\boldsymbol{x}_t^{(k)} = \left(\boldsymbol{x}_{t-1}^{(k)}, x_t^{(k)}\right)$.
- Update the importance weight

$$\begin{aligned} w_t^{(k)} &\propto w_{t-1}^{(k)} p_{\boldsymbol{\theta}}\left(y_t|x_{t-1}^{(k)}\right) \\ &\propto w_{t-1}^{(k)} \sum_{b\in\mathcal{X}} p_{\boldsymbol{\theta}}\left(y_t|x_t = b\right) p_{\boldsymbol{\theta}}\left(x_t = b|x_{t-1}^{(k)}\right). \end{aligned} \tag{18}$$

- Normalize the importance weights for them to sum up to one.

A common problem with the SMC algorithm is known as the degeneracy phenomenon. In [5] it is shown that the variance of the importance weights can only increase over time which makes the degeneracy problem ineluctable. After a few iterations, some samples will have very small weights. Such samples are said to be ineffective. If there are too many ineffective samples, the Monte Carlo procedure becomes inefficient. The *resampling* scheme is a useful method for reducing ineffective samples and enhancing effective ones. One simple resampling scheme, which we use in this paper, can be described as follows (cf. [6] for other schemes).

- Draw K sample streams $\{\overline{\boldsymbol{x}}_t^{(j)}\}_{J=1}^{K}$ from $\{\boldsymbol{x}_t^{(k)}\}_{k=1}^{K}$ with probabilities proportional to the weights $\{w_t^{(k)}\}_{k=1}^{K}$.
- Assign equal weights to each stream, $\overline{w}_t^{(k)} = K^{-1}$.

It is shown in [22] that samples drawn by the above resampling procedure are indeed properly weighted with respect to $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t)$.

The degeneracy of the samples can be measured by the effective sample size which is defined and approximated, respectively, by [23]:

$$\begin{aligned} K_{\text{eff}} &\triangleq K\left[1 + \text{Var}\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(k)}|\boldsymbol{y}_t)}{q_{\boldsymbol{\theta}}(x_t^{(k)}|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t)}\right)\right]^{-1} \\ \widehat{K_{\text{eff}}} &= \left(\sum_{k=1}^{K} (w_t^{(k)})^2\right)^{-1}. \end{aligned} \tag{19}$$

Heuristically, $\widehat{K_{\text{eff}}}$ reflects the equivalent size of a set of i.i.d. samples for the set of K weighted ones. It is suggested in [23], [6] that resampling should be performed whenever the effective sample size becomes small, e.g., $\widehat{K_{\text{eff}}} \leq K/10$.

B. Sequential Monte Carlo With Unknown Static Parameters

If the parameters $\boldsymbol{\theta}$ are unknown, the usual approach is to include the parameters into the state vector. Because of the static evolution of the parameters, the space of parameters is only explored during initialization which is obviously inefficient. Several works have proposed better algorithms for dealing with static parameters within an SMC framework [24], [25]. Gilks and Berzuini [26] proposed to use MCMC moves within the SMC framework. Such a procedure avoids the usual sample depletion problem but requires the storage of the complete path of the particles and increases the computational load. In [27], the author showed that sufficient statistics can be used to perform

these MCMC steps without the growing storage requirement. In this work, we make use of the approach developed in [8] which also uses sufficient statistics. In the model under consideration, the posterior distribution of $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}\}$ given $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ has been shown in (10)–(11) to be defined by Dirichlet distributions. It therefore depends on some sufficient statistics $\boldsymbol{T}_t = \boldsymbol{T}_t(\boldsymbol{x}_t, \boldsymbol{y}_t)$ that can easily be updated.

We consider a generic case where the parameters of the probability density function depend on some sufficient statistics, $p(\boldsymbol{\theta}|\boldsymbol{x}_t, \boldsymbol{y}_t) = p(\boldsymbol{\theta}|\boldsymbol{T}_t(\boldsymbol{x}_t, \boldsymbol{y}_t))$. Since $p(\boldsymbol{\theta}|\boldsymbol{x}_t, \boldsymbol{y}_t)$ is easily updated we are interested in having a Monte Carlo approximation of $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$. Suppose a set of properly weighted samples

$$\left\{\boldsymbol{x}_{t-1}^{(k)}, w_{t-1}^{(k)}\right\}_{k=1}^{K}$$

with respect to $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$ is available at time $(t-1)$

$$\hat{p}(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) = \frac{1}{W_{t-1}} \sum_{k=1}^{K} w_{t-1}^{(k)} \mathbb{I}\left(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{(k)}\right). \quad (20)$$

The main idea is to get a Monte Carlo approximation of $p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{y}_t)$ from (20) and the set of sufficient statistics

$$\left\{\boldsymbol{T}_t^{(k)}\right\}_{k=1}^{K} = \left\{\boldsymbol{T}_t(\boldsymbol{x}_t^{(k)}, \boldsymbol{y}_t)\right\}_{k=1}^{K}.$$

The approximation of the marginal distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$ is then simply obtained by discarding the samples $\boldsymbol{\theta}^{(k)}$. Therefore, only the samples $\boldsymbol{x}_t^{(k)}$ and the corresponding sufficient statistics $\boldsymbol{T}_t^{(k)}$ are stored, but samples for $\boldsymbol{\theta}$ are drawn jointly to $x_t^{(k)}$ to simplify the computations. Specifically this approach is based on the following identity:

$$\begin{aligned} &p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{y}_t) \\ &\quad \propto p(\boldsymbol{x}_t, \boldsymbol{\theta}, y_t|\boldsymbol{y}_{t-1}) \propto p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) p(\boldsymbol{\theta}|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) \\ &\qquad \cdot p(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{\theta}) p(y_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}, \boldsymbol{\theta}) \\ &\quad \propto p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) p(\boldsymbol{\theta}|\boldsymbol{T}_{t-1}) p(x_t|x_{t-1}, \boldsymbol{\theta}) p(y_t|x_t, \boldsymbol{\theta}). \end{aligned} \quad (21)$$

Based on the importance sampling paradigm, a Monte Carlo approximation of (21) can be obtained by keeping the past simulated streams

$$\left\{\boldsymbol{x}_{t-1}^{(k)}, w_{t-1}^{(k)}\right\}_{k=1}^{K}$$

unmodified and drawing $(\boldsymbol{\theta}^{(k)}, x_t^{(k)})$ from a proposal distribution $q(\boldsymbol{\theta}, x_t|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t) = q_1(\boldsymbol{\theta}|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t) \cdot q_2(x_t|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t, \boldsymbol{\theta})$. The weights are updated according to the usual rule

$$w_t^{(k)} \propto w_{t-1}^{(k)} \frac{p\left(\boldsymbol{\theta}^{(k)}|\boldsymbol{T}_{t-1}^{(k)}\right) p\left(x_t^{(k)}|x_{t-1}^{(k)}, \boldsymbol{\theta}^{(k)}\right) p\left(y_t|x_t^{(k)}, \boldsymbol{\theta}^{(k)}\right)}{q_1\left(\boldsymbol{\theta}^{(k)}|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t\right) q_2\left(x_t^{(k)}|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t, \boldsymbol{\theta}^{(k)}\right)}. \quad (22)$$

The sufficient statistics are then updated and the samples for $\boldsymbol{\theta}$ discarded. Estimation of $\boldsymbol{\theta}$ is done through Rao-Blackwellization as follows [28]:

$$\mathbb{E}\{\boldsymbol{\theta}|\boldsymbol{y}_t\} = \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{y}_t}\left\{\mathbb{E}\left\{\boldsymbol{\theta}|\boldsymbol{y}_t, \boldsymbol{x}_t\right\}\right\} \approx \frac{1}{W_t} \sum_{k=1}^{K} w_t^{(k)} \mathbb{E}\left\{\boldsymbol{\theta}|\boldsymbol{T}_t^{(k)}\right\}. \quad (23)$$

Resampling can be performed as usual.

C. The SMC Estimator

We next outline the SMC algorithm for online estimation of the number of users when the system parameters $\boldsymbol{\theta} = [\boldsymbol{\pi}, \boldsymbol{A}]$ are unknown. The prior distributions (7)–(8) for the parameters will be used hereafter. At time $(t-1)$, the posterior distribution of $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}\}$ given $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ has been shown in (10)–(11) to be given by some Dirichlet distributions. Let's denote $\rho_{i,t-1}$ and $\alpha_{i,j,t-1}$ the parameters of these distributions

$$\begin{aligned} p(\boldsymbol{\pi}|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) &= \mathcal{D}\left(\boldsymbol{\pi}; \rho_{1,t-1}, \rho_{2,t-1}, \ldots, \rho_{N,t-1}\right), \\ p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) &= \mathcal{D}\left(\boldsymbol{a}_i; \alpha_{i,1,t-1}, \ldots, \alpha_{i,N,t-1}\right). \end{aligned} \quad (24)$$

The posterior distribution of the parameters therefore only depends on the sufficient statistics

$$\boldsymbol{T}_{t-1} = \{\alpha_{i,j,t-1}, \rho_{m,t-1}\}_{(i,j,m)\in[1,N]^3}.$$

Furthermore, we have

$$p(\boldsymbol{\pi}|\boldsymbol{x}_t, \boldsymbol{y}_t) = p(\boldsymbol{\pi}|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}), \quad (25)$$

$$\begin{aligned} p(\boldsymbol{a}_i|\boldsymbol{x}_t, \boldsymbol{y}_t) &= p(\boldsymbol{a}_i|\boldsymbol{x}_t) \propto p(x_t|x_{t-1}, \boldsymbol{a}_i) p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1}) \\ &= \mathcal{D}\Big(\boldsymbol{a}_i; \underbrace{\alpha_{i,1,t-1} + \mathbb{I}(x_{t-1} - i)\mathbb{I}(x_t - 1)}_{\alpha_{i,1,t}} \\ &\qquad \ldots, \underbrace{\alpha_{i,N,t-1} + \mathbb{I}(x_{t-1} - i)\mathbb{I}(x_t - N)}_{\alpha_{i,N,t}}\Big) \end{aligned} \quad (26)$$

so that $\boldsymbol{T}_t$ is easily updated. It can also be seen that the trial distribution $q_2(x_t|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t, \boldsymbol{\theta})$ has no reason to depend on $\boldsymbol{\pi}$ since x_{t-1} is given. Therefore we only need to consider the transition matrix $\boldsymbol{A}$ in the parameters. The initialization step is derived in a straightforward manner from the present discussion.

We consider the optimal proposal distribution for the number of terminals

$$\begin{aligned} q_2(x_t = i|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t, \boldsymbol{\theta}) &= p(x_t = i|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t, \boldsymbol{\theta}) \\ &\propto \mathcal{B}(y_t; B, h(i)) \cdot a_{x_{t-1}^{(k)}, i}. \end{aligned} \quad (27)$$

Sampling $\boldsymbol{A}$ is a little bit more involved if we want to include the latest observation in the proposal distribution. It is shown in the Appendix that the posterior distribution of $\boldsymbol{a}_i$ given $\boldsymbol{x}_{t-1}$ and $\boldsymbol{y}_t$ is a mixture of Dirichlet distributions which we use as a proposal distribution: see (28) at the bottom of the next page, where $\beta_{i,j,t} = \mathcal{B}(y_t; B, h(j))\alpha_{i,j,t-1}^{\mathbb{I}(x_{t-1}-i)}$. The weight update formula

(22) can now be computed and its derivation can be found in the Appendix

$$w_t \propto w_{t-1} \frac{\sum_{i=1}^{N} \mathcal{B}(y_t; B, h(i)) \alpha_{x_{t-1},i,t-1}}{\sum_{i=1}^{N} \alpha_{x_{t-1},i,t-1}}. \tag{29}$$

We can see a very interesting feature coming out here. The weight update formula does not depend on the actual values sampled at time t. It is therefore possible to compute the weights before sampling. This is very attractive since we are now able to perform resampling before sampling which lowers the loss of diversity occurring during the usual resampling scheme. The same idea appears in the auxiliary sample filter [29] where a part of the weight can be computed before sampling and the rest is roughly estimated. The approximate weights are then used to perform the resampling as a prior step. The major difference here is that the complete weights can be precomputed and that we advocate to use resampling only if necessary, e.g., whenever $\widehat{K_{\text{eff}}} \leq K/10$. The complete SMC estimator with unknown parameters is summarized hereafter.

Algorithm 2 [Online SMC estimator]

- *Initialization: Draw the initial samples of prior distributions $\boldsymbol{\pi}^{(k)}$ according to $p(\boldsymbol{\pi}|y_1)$, and $x_1^{(k)}$ according to $p(x_1|y_1, \boldsymbol{\pi}^{(k)})$. The corresponding weights are all equal.*
- *Importance sampling: For $t = 1, 2, \ldots$*
  - *Compute the new weights according to* (29).
  - *Compute $\widehat{K_{\text{eff}}}$ according to* **(19)**. *If $\widehat{K_{\text{eff}}} \leq K/10$ perform resampling.*
  - *For $k = 1, \ldots, K$*
    a) *Sample $\boldsymbol{A}^{(k)}$ from* (28).
    b) *Sample $x_t^{(k)}$ from* (27).
    c) *Update the sufficient statistics $\boldsymbol{T}_t^{(k)} = \boldsymbol{T}_t(\boldsymbol{x}_t^{(k)}, \boldsymbol{y}_t)$.*
  - *If necessary, estimate the posterior probability distribution of x_t and compute an estimate of $\boldsymbol{A}$ according to* (23).

V. Online Deterministic Estimators

The previous sequential Monte Carlo algorithm provided the basis for online Bayesian estimation. The discrete characteristic of the number of terminals and the fact that a set of sufficient statistics for the posterior probability of the unknown transition matrix can easily be updated allowed us to significantly improve on the basic SMC procedure. In this section we will show that this can be pushed one step further and we derive a deterministic sequential sampling algorithm that outperforms the SMC estimator both in terms of computational load, robustness and accuracy. Depending on the specific application in mind even this estimator can seem somewhat computationally intensive. We therefore propose a novel approximate maximum a posteriori algorithm that trades the accuracy of the deterministic sampling algorithm for computational load. Both algorithms are designed for any HMM with unknown transition matrix and do not specifically depend on the DCF scenario under consideration.

A. Deterministic Sequential Sampling

The online SMC estimator discussed in Section IV-A randomly generates samples according to $p(x_t|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t)$ for $x_t \in \mathcal{X}$. Consequently, some information is somehow distorted if the number of Monte Carlo samples is not sufficiently large. Furthermore, the use of the optimal sampling distribution implicitly led us to consider all possible extensions of a sample. The mixture densities indeed arose from this fact. This was possible because x_t can only take values from the finite set $\mathcal{X}$. An alternative deterministic approach, developed in [30], [9], and extended here to the case of unknown parameters and Markov state processes, consists of explicitly considering all K_{ext} possible extensions of the K samples and then perform a selection step so as to avoid the exponential increase of the number of samples and keep a constant number K of them. Another idea in this context is that there is no point in keeping different samples representing the same path. Therefore, the selection step should not rely on the usual resampling scheme. In this paper, we rely on the simplest (but effective) idea which is to select the K most likely samples at each time step. Strictly speaking, we then loose the properly weighted characteristic by cutting out the tails of the pdf during the selection step. To avoid this, a more sophisticated scheme is developed in [31] and [32]. Some of the most likely samples are kept and resampling without replacement is performed on the remaining ones.

We consider again the state-space model (15) where the parameters $\boldsymbol{\theta}$ are assumed known. Suppose a set of weighted samples $\left\{\boldsymbol{x}_{t-1}^{(k)}, w_{t-1}^{(k)}\right\}_{k=1}^{K}$ representing $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$ is available at time $(t-1)$. We assume that this set does not contain any

$$\begin{aligned} q_1\left(\boldsymbol{A}|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t\right) &= \prod_{i=1}^{N} p\left(\boldsymbol{a}_i|\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t\right) \propto \prod_{i=1}^{N} \left[\sum_{j=1}^{N} \beta_{i,j,t} p\left(\boldsymbol{a}_i|\boldsymbol{T}_t(x_t = j, \boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t)\right)\right] \\ &\propto \sum_{j=1}^{N} \beta_{x_{t-1}^{(k)},j,t} p\left(\boldsymbol{a}_{x_{t-1}^{(k)}}|\boldsymbol{T}_t(x_t = j, \boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_t)\right) \prod_{i \neq j} p\left(\boldsymbol{a}_j|\boldsymbol{T}_t(\boldsymbol{x}_{t-1}^{(k)}, \boldsymbol{y}_{t-1})\right) \end{aligned} \tag{28}$$

duplicate samples. The posterior distribution of $\boldsymbol{x}_{t-1}$ is approximated as

$$\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) = \frac{1}{W_{t-1}} \sum_{k=1}^{K} w_{t-1}^{(k)} \mathbb{I}\left(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{(k)}\right) \tag{30}$$

where $W_{t-1} = \sum_{k=1}^{K} w_{t-1}^{(k)}$. From Bayes theorem we have

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t) &\propto p_{\boldsymbol{\theta}}(y_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_{t-1}) \\ &\propto p_{\boldsymbol{\theta}}(y_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}) p_{\boldsymbol{\theta}}(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) \end{aligned} \tag{31}$$

and the state transition distribution can be written as

$$p_{\boldsymbol{\theta}}(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) = p_{\boldsymbol{\theta}}(x_t|x_{t-1}) = \sum_{i=1}^{N} a_{x_{t-1},i} \mathbb{I}(x_t - i). \tag{32}$$

The posterior distribution of $\boldsymbol{x}_t$ can be approximated by

$$\hat{p}_{\boldsymbol{\theta}}^{\text{ext}}(\boldsymbol{x}_t|\boldsymbol{y}_t) = \frac{1}{W_t^{\text{ext}}} \sum_{k=1}^{K} \sum_{i=1}^{N} w_t^{(k,i)} \mathbb{I}\left(\boldsymbol{x}_t - \left[\boldsymbol{x}_{t-1}^{(k)}, i\right]\right). \tag{33}$$

where $W_t^{\text{ext}} = \sum_{k,i} w_t^{(k,i)}$ and

$$w_t^{(k,i)} \propto w_{t-1}^{(k)} p_{\boldsymbol{\theta}}(y_t|x_t = i) p_{\boldsymbol{\theta}}(x_t = i|x_{t-1}^{(k)}). \tag{34}$$

The initialization steps of the algorithm proceed exactly as stated earlier except that no selection needs to be done until to total number of samples exceeds the maximum number allowed N.

We now extend this approach to the case where the system parameters $\boldsymbol{\theta}$ are unknown but their posterior distribution, given $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, only depends on a set of sufficient statistics that can easily be updated, such as considered in Section IV-B. Similarly to (31) we have

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_t) &\propto p(y_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}) p(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) \\ &\propto p(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) \int p_{\boldsymbol{\theta}}(y_t|x_t) p(\boldsymbol{\theta}|x_t, \boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) \mathrm{d}\boldsymbol{\theta} \\ &\quad \cdot \int p_{\boldsymbol{\theta}}(x_t|x_{t-1}) p(\boldsymbol{\theta}|\boldsymbol{T}_{t-1}) \mathrm{d}\boldsymbol{\theta}. \end{aligned} \tag{35}$$

Depending on the specific state-space (15) under consideration, evaluating (35) can be an easy or very difficult task. If no analytical form is available, it is possible to approximate these integrals. Monte Carlo sampling or the unscented transform [33] are some of the options but one could also simply evaluate $p_{\overline{\boldsymbol{\theta}}}(\cdot)$ where $\overline{\boldsymbol{\theta}}$ can be the mean, mode or any other likely value of $\boldsymbol{\theta}$ under $p(\boldsymbol{\theta}|x_t = i, \boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})$ and $p(\boldsymbol{\theta}|\boldsymbol{T}_{t-1}^{(k)})$, respectively. Such an approximation is for example used in the auxiliary particle filter [29] during the auxiliary weights computation. This rough approximation should often be sufficient when $p_{\boldsymbol{\theta}}(\cdot)$ is smooth with respect to $\boldsymbol{\theta}$.

In our case, the emission probabilities $p_{\boldsymbol{\theta}}(y_t|x_t = i) = \mathcal{B}(y_t; B, h(i))$ do not depend on the parameters $\boldsymbol{\theta}$. Thanks to the Dirichlet prior, the integral with respect to $p(\boldsymbol{\theta}|\boldsymbol{T}_{t-1})$ can be computed analytically

$$\begin{aligned} p(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) &= \int p_{\boldsymbol{\theta}}(x_t|x_{t-1}) p(\boldsymbol{\theta}|\boldsymbol{T}_{t-1}) \mathrm{d}\boldsymbol{\theta} \\ &= \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{T}_{t-1})}\left\{a_{x_{t-1},x_t}\right\} = \frac{\alpha_{x_{t-1},x_t,t-1}}{\sum_{j=1}^{N} \alpha_{x_{t-1},j,t-1}}. \end{aligned} \tag{36}$$

The recursion (35) can thus be computed analytically. If, at time $(t-1)$ $\left\{\boldsymbol{x}_{t-1}^{(k)}, w_{t-1}^{(k)}\right\}_{k=1}^{K}$ represents $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$ then, as in (33), $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$ can be approximated by

$$\hat{p}^{\text{ext}}(\boldsymbol{x}_t|\boldsymbol{y}_t) = \frac{1}{W_t^{\text{ext}}} \sum_{k=1}^{K} \sum_{i=1}^{N} w_t^{(k,i)} \mathbb{I}\left(\boldsymbol{x}_t - \left[\boldsymbol{x}_{t-1}^{(k)}, i\right]\right) \tag{37}$$

where the weight update is given by

$$w_t^{(k,i)} \propto w_{t-1}^{(k)} \mathcal{B}(y_t; B, h(i)) \frac{\alpha_{x_{t-1}^{(k)},i,t-1}}{\sum_{j=1}^{N} \alpha_{x_{t-1}^{(k)},j,t-1}}. \tag{38}$$

As one can see, such a procedure has many benefits since the parameters are analytically integrated out, no random sampling has to be performed and no computation needs to be done twice. The complete algorithm can now be summarized as follows.

Algorithm 3 [Online deterministic estimator]

- *Initialization: Enumerate the N possible samples and compute their weights.*
- *Update: For $t = 2, 3, \ldots$*
  - *For $k = 1, 2, \ldots$*
    a) *Enumerate all possible sample extensions.* $\boldsymbol{x}_t^{(k,i)} = \left[\boldsymbol{x}_{t-1}^{(k)}, i\right]$
    b) *$\forall i$, compute the weights $w_t^{(k,i)}$ according to* (38)
  - *If necessary, estimate the posterior probability distribution of x_t.*
  - *Select and preserve N distinct sample streams $\left\{\boldsymbol{x}_t^{(k)}\right\}_{k=1}^{N}$ with the highest importance weights $\left\{w_t^{(k)}\right\}_{k=1}^{N}$ from the set $\left\{\boldsymbol{x}_t^{(k,i)}, w_t^{(k,i)}\right\}_{k,i}$.*
  - *$\forall k$, update the sufficient statistics $\boldsymbol{T}_t^{(k)} = \boldsymbol{T}_t(\boldsymbol{x}_t^{(k)}, \boldsymbol{y}_t)$.*
  - *If necessary, compute an estimate of $\boldsymbol{A}$ according to (23).*

A more accurate estimate of $\boldsymbol{A}$ can easily be obtained by updating the sufficient statistics and estimating $\boldsymbol{A}$ before the selection step. However this would induce a heavier computational load.

B. Approximate MAP Algorithm

For HMM with known parameters, the Viterbi algorithm provides a recursive solution to get the best state sequence estimation in terms of the maximum *a posteriori* (MAP) [10]. When the parameters are unknown, the most common procedure is to use an EM algorithm which only converges to some local maximum of the *a posteriori* density, but above all it is a batch procedure and thus can not be used in our setting. Online estimation of the HMM parameters has been studied in [12]–[16]. In this section, another approach based on the use of sufficient statistics developed here is taken. An approximate MAP algorithm is presented whose computational load and memory need are equivalent to a usual Viterbi algorithm.

We are interested in recursively maximizing $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$ with respect to $\boldsymbol{x}_t$. In order to do that, the Viterbi algorithm makes use of the quantity

$$\delta_t(i) = \max_{\boldsymbol{x}_{t-1}|x_t=i} p(\boldsymbol{x}_t|\boldsymbol{y}_t). \tag{39}$$

From (35) we have

$$\delta_t(i) = p(y_t|x_t=i) \cdot \max_{x_{t-1}|x_t=i} \max_{\boldsymbol{x}_{t-2}|x_{t-1},x_t=i} [p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})] \tag{40}$$

that can recursively be computed by taking $p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1}) = a_{x_{t-1},x_t}$ out of the inner max so that $\delta_t(i) = p(y_t|x_t = i)\max_j[\delta_{t-1}(j)\cdot a_{j,i}]$. The estimate $\tilde{\boldsymbol{x}}_t$ of $\boldsymbol{x}_t$ 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, 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 only $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$ is maximized, we then get

$$\max_{\boldsymbol{x}_{t-2}|x_t=i,x_{t-1}=j} [p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1}) p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})] = p(\boldsymbol{x}^*_{t-2}, x_{t-1}=j|\boldsymbol{y}_{t-1}) p(x_t=i|\boldsymbol{x}^*_{t-2}, x_{t-1},\boldsymbol{y}_{t-1}) \tag{41}$$

where $\boldsymbol{x}^*_{t-2} = \arg\max_{\boldsymbol{x}_{t-2}|x_t=i,x_{t-1}=j} p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{t-1})$. This allows us to derive an approximate MAP algorithm. The rationale behind the assumption above is that, as time goes, our estimation of the transition matrix will stabilize. The impact of the transition probability should therefore be lower than that of the observation probability. Our simulations showed that (41) provides rather good results given the low complexity of the resulting algorithm. Our approximation $\hat{\delta}_t(i)$ of $\delta_t(i)$ can thus be recursively computed by keeping, for every possible value j, only the best path ending at this particular value j together with the corresponding set of sufficient statistics $\boldsymbol{T}_t^{(j)} = \left\{\alpha^{(j)}_{i,k,t}, \rho^{(j)}_{m,t}\right\}_{(i,k,m)\in[1,N]^3}$. The recursion is then given by

$$\hat{\delta}_t(i) = p(y_t|x_t=i)\max_j \left[\hat{\delta}_{t-1}(j)\cdot \frac{\alpha^{(j)}_{j,i,t-1}}{\sum_{k=1}^{N}\alpha^{(j)}_{j,k,t-1}}\right]. \tag{42}$$

Let $\hat{\boldsymbol{x}}_t$ be our estimate of $\boldsymbol{x}_t$ at time t. The approximate MAP algorithm is now summarized.

Algorithm 4 [Approximate MAP Algorithm]

- *Initialization: For $i = 1,\ldots,N$, set the weight of each point to $\hat{\delta}_1(i) = p(x_1 = i|y_1) \propto \mathcal{B}(y_1;B,h(i))\rho_i/\sum_j \rho_j$.*
- *Update: For $t = 2,3,\ldots$*
  - *For $i = 1,\ldots,N$*
    a) *Set*
    $$\hat{\delta}_t(i) = \mathcal{B}(y_t;B,h(i)) \max_j \left[\hat{\delta}_{t-1}(j)\cdot \frac{\alpha^{(j)}_{j,i,t-1}}{\sum_{k=1}^{N}\alpha^{(j)}_{j,k,t-1}}\right].$$
    b) *Set* $\hat{\psi}_t(i) = \arg\max_j \left[\hat{\delta}_{t-1}(j)\cdot \alpha^{(j)}_{j,i,t-1}/\sum_{k=1}^{N}\alpha^{(j)}_{j,k,t-1}\right]$.
    c) *Set* $\boldsymbol{x}_t^{(i)} = \left[\boldsymbol{x}_{t-1}^{(\hat{\psi}_t(i))}, i\right]$.
    d) *Update the sufficient statistics:* $\boldsymbol{T}_t^{(i)} = \boldsymbol{T}_t(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t)$.
  - *If necessary, get the approximate ML estimate $\hat{\boldsymbol{x}}_t$ of $\boldsymbol{x}_t$ by using the sequence that maximizes $\delta_t(\cdot)$.*
  - *If necessary, get an estimate of $\boldsymbol{A}$ from $\hat{\boldsymbol{A}} = E[\boldsymbol{A}|\boldsymbol{T}_t^{(\hat{x}_t)}]$*

VI. Simulation Results

A. Model Data

In this section, we first evaluate the performance of the estimators with data extracted from the model. We assume an 802.11 network as modeled in [2], where the relation between the number of competing terminals and the probability of collision is given by (2). Our scenario is composed of a variable number of competing stations x transmitting in saturation conditions. As in [2], only DCF basic access is considered, with no capture or hidden terminals. The arrival and departure of competing terminals from the network follow a random Markov chain as in (5). The exponential backoff parameters are $CW_{\min} = 16$, 32, and 64 with $m = 4$, 5, and 6, respectively, i.e., $CW_{\max} = 2^m CW_{\min} = 1024$.

We generate noisy observations from (5), and each station monitors the medium and estimates the probability of collision by counting the number of busy slots as indicated in (3) with $B = 100$. For each estimator that provide an approximation of the filtering density, we first make a hard estimate by taking the mode of the output distribution. The different estimators are then compared by using this hard estimate. Fig. 1(a)–(d) shows the performance of our proposed estimators compared to the extended Kalman filter with cumulative summary change detection (EKF_CUMSUM) proposed in [3] for the case $CW_{\min} = 32$, $m = 5$. The effectiveness of the proposed estimators for all the parameters is summarized in Table I. Several observations are in order. First, all our proposed algorithms substantially outperform the CUSUM-EKF algorithm. Second, as expected, the off-line Gibbs sampler achieves the best performance among all estimators. For the on-line estimators, both the deterministic

TABLE I
AVERAGE MSE OF THE HARD ESTIMATION OBTAINED FROM 100 DATA SETS

Estimator	model-based			ns-2 (Markov)		ns-2 (On-Off exponential)		
CW_{min}, m	16,6	32,5	64,4	32,5	64,4	16,6	32,5	64,4
Gibbs Sampler	0.4723	0.4012	0.4604	1.0318	0.9761	1.4176	1.0416	0.8990
SMC	0.4635	0.4351	0.5420	1.2232	1.0213	1.6042	1.1111	0.9132
Deterministic	0.6533	0.5961	0.8056	1.1663	1.0733	1.4797	1.0614	1.0181
EKF+CUMSUM	1.1528	1.1820	1.2133	1.5236	2.8760	1.8903	2.3663	2.5957
Approximate MAP	0.6079	0.5180	0.6557	1.0680	0.9108	1.5338	1.0842	0.9100

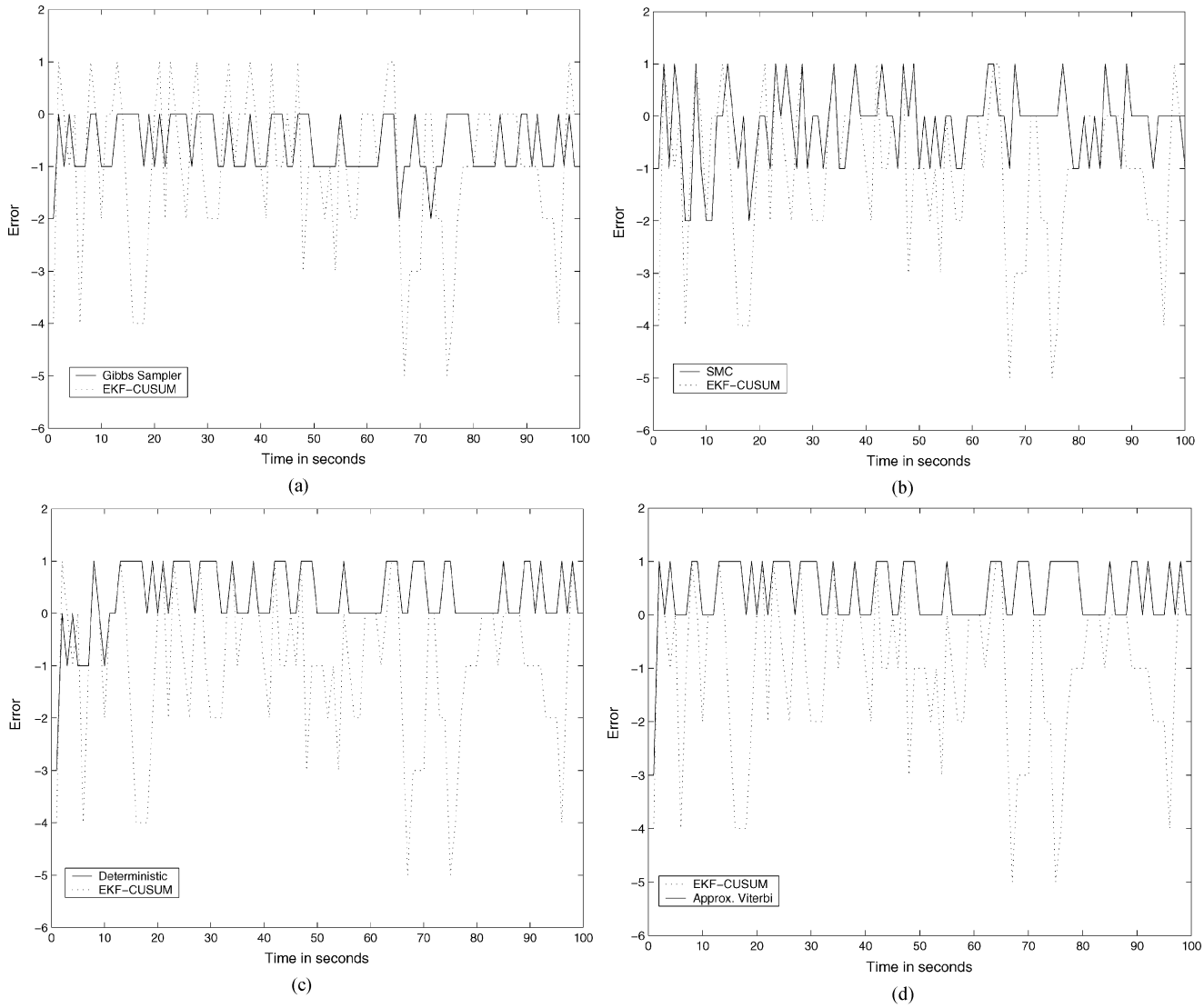


Fig. 1. Estimation error: model-based simulation, $CW_{\min} = 32, m = 5$. (a) Gibbs sampler. (b) SMC. (c) Deterministic sampling. (d) Approximate MAP.

algorithm and the approximate MAP algorithm performs similar than the SMC estimator, approaching the performance of the Gibbs sampler. Finally, the approximate MAP algorithm appears as an excellent option for real-time implementation given its lower complexity and excellent performance.

B. NS-2 Data

The estimators assume that there exist a model that express the number of terminals as a function of the collision probability, but it does not specify the nature of the model. For the cases in

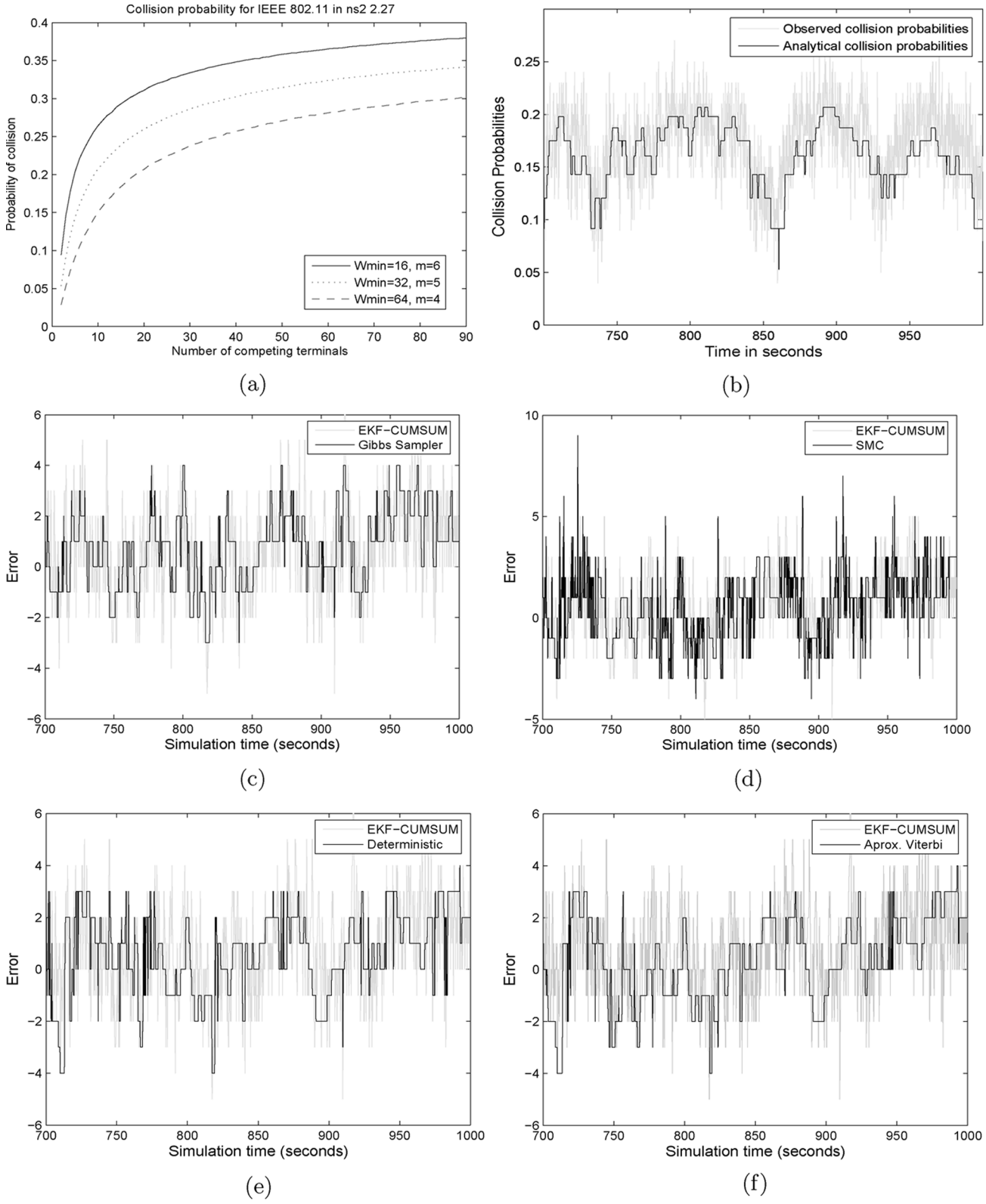


Fig. 2. Estimation error: ns-2 simulation, exponential on-off activation, $CW_{\min} = 32$, $m = 5$. (a) Number of stations versus collision probability in ns-2. (b) Observed probability of collision. (c) Gibbs sampler. (d) SMC. (e) Deterministic sampling. (f) Approximate MAP.

which an analytical model is not available, empirical models can also be used. For the real data simulations, we use the ns-2 network simulator version 2.27 [34]. We modified the 802.11 implementation so that the nodes can monitor the channel and make an estimation of the collision probability based on 3. The parameters used in the simulation are classical for a 1 Mbps WLAN. No packet fragmentation occurs, and the nodes 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 1024 bytes. The MAC and PHY headers use, respectively, 272 and 128 bits. The PHY preamble takes 144 bits. The ACK length is 112 bits. 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 28 μs and the DIFS 130 μs. The RTS/CTS threshold was increased so that only the basic access was used.

Fig. 2 shows the collision probability versus the number of competing stations obtained empirically in the ns-2 simulator. Each point was obtained simulating a fixed number of stations transmitting under saturation conditions and measuring the total probability of collision. The simulation time for this empirical measurement lasted 3000 s to provide better accuracy. To avoid including ARP packets in the measurement an initial 20 s transmission was used to ensure all the nodes had updated ARQ tables. Finally, an additional 100 s transmission was added before measurements to allow the system to reach the steady state.

For testing the estimators in a realistic scenario, we used the curves obtained in Fig. 2 instead of the analytical model in (2). Our simulation scenario is composed of a variable number of competing stations x transmitting in saturation conditions. The terminals use $B = 100$ for estimating the collision probability. Note that increasing B would result in better estimates but, on the other side, would increase the delay as the nodes need to wait more slots to update their estimates. Each ns-2 simulation run lasts 200 s. 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. In our irregular time grid, this is not Markovian anymore.

Fig. 2(c)–(f) shows the performance of our proposed estimators, again compared to the extended Kalman filter with cumulative summary change detection (EKF_CUMSUM) for $CW_{\min} = 32$ and $m = 5$. Fig. 2(b) shows the probability of collision observed by the nodes versus the actual probability of collision. It is interesting to note that the observed probability of collision is considerably noisy. Table I shows the average mean-squared error (mse) for the proposed estimators from a total of 100 data sets. For comparative purposes, we also included an ns-2 simulation in which the arrival of the nodes is according to a Markov chain, as assumed by our model.

Note that for the noisy realistic 802.11 data, the proposed Bayesian estimators offer superior performance. Again, as expected, the off-line Gibbs sampler estimator is better, and the deterministic SMC algorithm offers a better performance than the SMC estimator, despite being simpler. Finally, our novel approximate MAP algorithm perform similarly than the SMC algorithm, and still may be considered as the best option from a practical implementation point of view in terms of the computational complexity and power consumption.

VII. Conclusion

In this paper, we proposed several algorithms for the problem of estimating the number of competing terminals in an IEEE 802.11 wireless network under the framework of Bayesian Monte Carlo signal processing. In particular, we developed a Gibbs sampler algorithm for off-line estimation. We also developed sequential Monte Carlo-based online algorithms, including a simpler deterministic variant. Finally, we developed a novel approximate MAP algorithm that trades accuracy for computational complexity. Our online estimators can be applied to any hidden Markov chain with unknown transition probabilities and unknown prior distributions, which makes them appropriate for an 802.11 protocol where, from the terminal point of view, there is very little knowledge of the state of the system.

Simulation results show that these algorithms achieve better performance compared with the EKF-CUSUM-based estimator under both an artificial hidden Markov model and a real system based on IEEE 802.11 simulations performed with ns-2. Furthermore our approximate MAP proposal offers a similar performance with considerably less computational requirement (and, hence, lower power consumption), making it the preferred candidate for an actual implementation of the estimator on an IEEE 802.11 card.

Appendix

Derivation of (28): See the equation at the top of the next page. The posterior distribution of $\boldsymbol{a}_i$ given $\boldsymbol{x}_{t-1}$ and $\boldsymbol{y}_t$ is, thus, a mixture of Dirichlet distributions that can be rewritten as

$$p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1},\boldsymbol{y}_t)=\frac{1}{\sum_{m=1}^{N}\beta_{i,m,t}}\sum_{j=1}^{N}\beta_{i,j,t}p(\boldsymbol{a}_i|\boldsymbol{T}_t(x_t=j,\boldsymbol{x}_{t-1},\boldsymbol{y}_t))$$

where $\beta_{i,j,t}=\mathcal{B}(y_t;B,h(j))\alpha_{i,j,t-1}^{\mathbb{I}(x_{t-1}-i)}$.

Derivation of (29): From (7), we can see that

$$\begin{aligned}
&\frac{p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1},\boldsymbol{y}_t)}{p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})}\\
&=\frac{1}{\sum_{m=1}^{N}\beta_{i,m,t}}\sum_{j=1}^{N}\beta_{i,j,t}\frac{\left(\sum_{m=1}^{N}\alpha_{i,m,t-1}\right)^{\mathbb{I}(x_{t-1}-i)}}{\alpha_{i,j,t-1}^{\mathbb{I}(x_{t-1}-i)}}a_{i,j}^{\mathbb{I}(x_{t-1}-i)}\\
&=\begin{cases}1, & \text{if } x_{t-1}\neq i\\ \dfrac{\sum_{m=1}^{N}\alpha_{i,m,t-1}}{\sum_{m=1}^{N}\beta_{i,m,t-1}}\sum_{j=1}^{N}\beta_{i,j,t-1}\frac{a_{i,j}}{\alpha_{i,j,t-1}}, & \text{if } x_{t-1}=i\end{cases}\\
&=\begin{cases}1, & \text{if } x_{t-1}\neq i\\ \dfrac{\sum_{j=1}^{N}\alpha_{i,j,t-1}}{\sum_{j=1}^{N}\beta_{i,j,t-1}}\sum_{j=1}^{N}\mathcal{B}(y_t;B,h(j))a_{i,j}, & \text{if } x_{t-1}=i\end{cases}.
\end{aligned}$$

Therefore, we get

$$\frac{p(\boldsymbol{A}|\boldsymbol{x}_{t-1},\boldsymbol{y}_t)}{p(\boldsymbol{A}|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})}=\frac{\sum_{i=1}^{N}\alpha_{x_{t-1},i,t-1}}{\sum_{i=1}^{N}\beta_{x_{t-1},i,t-1}}\sum_{i=1}^{N}\mathcal{B}(y_t;B,h(i))\cdot a_{x_{t-1},i}.$$

$$
\begin{aligned}
p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1},\boldsymbol{y}_t) &\propto p(y_t|\boldsymbol{a}_i,\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1})p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1}) \\
&\propto \sum_{j=1}^{N} p(y_t|x_t=j)p(x_t|x_{t-1},\boldsymbol{a}_i)p(\boldsymbol{a}_i|\boldsymbol{x}_{t-1},\boldsymbol{y}_{t-1}) \propto \sum_{j=1}^{N} \mathcal{B}(y_t;B,h(j))a_{i,j}^{\mathbb{I}(x_{t-1}-i)} \prod_{m=1}^{N} a_{i,m}^{\alpha_{i,m,t-1}-1} \\
&\propto \sum_{j=1}^{N} \mathcal{B}(y_t;B,h(j)) \frac{\prod_{m=1}^{N}\Gamma(\alpha_{i,m,t-1}+\mathbb{I}(x_{t-1}-i)\mathbb{I}(j-m))}{\Gamma\left(\mathbb{I}(x_{t-1}-i)+\sum_{m=1}^{N}\alpha_{i,m,t-1}\right)} \\
&\quad \times \mathcal{D}(\boldsymbol{a}_i;\alpha_{i,1,t-1}+\mathbb{I}(x_{t-1}-i)\mathbb{I}(j-1),\ldots,\alpha_{i,N,t-1}+\mathbb{I}(x_{t-1}-i)\mathbb{I}(j-N)) \\
&\propto \sum_{j=1}^{N} \mathcal{B}(y_t;B,h(j)) \frac{\Gamma(\alpha_{i,j,t-1}+\mathbb{I}(x_{t-1}-i))}{\Gamma(\alpha_{i,j,t-1})} \\
&\quad \times \mathcal{D}(\boldsymbol{a}_i;\alpha_{i,1,t-1}+\mathbb{I}(x_{t-1}-i)\mathbb{I}(j-1),\ldots,\alpha_{i,N,t-1}+\mathbb{I}(x_{t-1}-i)\mathbb{I}(j-N)) \\
&\propto \sum_{j=1}^{N} \mathcal{B}(y_t;B,h(j))\alpha_{i,j,t-1}^{\mathbb{I}(x_{t-1}-i)}\mathcal{D}(\boldsymbol{a}_i;\alpha_{i,1,t-1}+\mathbb{I}(x_{t-1}-i)\mathbb{I}(j-1),\ldots,\alpha_{i,N,t-1}+\mathbb{I}(x_{t-1}-i)\mathbb{I}(j-N)).
\end{aligned}
$$

The rest of the weight update follows from the usual formula

$$
\begin{aligned}
&\frac{p(x_t|x_{t-1},\boldsymbol{\theta})p(y_t|x_t,\boldsymbol{\theta})}{p(x_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_t,\boldsymbol{\theta})} \\
&\qquad = p(y_t|x_{t-1},\boldsymbol{\theta}) = \sum_{i=1}^{N} \mathcal{B}(y_t;B,h(i)) \cdot a_{x_{t-1},i}.
\end{aligned}
$$

The weight update is thus performed by

$$
w_t \propto w_{t-1} \frac{\sum_{i=1}^{N} \beta_{x_{t-1},i,t-1}}{\sum_{i=1}^{N} \alpha_{x_{t-1},i,t-1}}.
$$

REFERENCES

[1] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 116–126, Sep. 1998.

[2] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 535–547, Mar. 2000.

[3] G. Bianchi and I. Tinnirello, "Kalman fitler estimation of the number of competing termnals in an IEEE 802.11 network," in *Proc. Infocom 2003*, San Francisco, CA, Mar. 2003, vol. 2, pp. 844–852.

[4] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London, U.K.: Chapman and Hall, 1996.

[5] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[6] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.

[7] X. Wang, R. Chen, and J. S. Liu, "Monte Carlo signal processing for wireless communications," *J. VLSI Signal Process.*, vol. 30, no. 1–3, pp. 89–105, Jan./Mar. 2002.

[8] G. Storvik, "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Trans. Signal Process.*, vol. 50, pp. 281–289, Feb. 2002.

[9] E. Punskaya, "Sequential Monte Carlo methods for digital communications," Ph.D, Univ. Cambridge, Cambridge, U.K., 2003.

[10] L. R. Rabiner, "A tutorial on hidden Markov models and selected application in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, Feb. 1989.

[11] S. L. Scott, "Bayesian methods for hidden Markov models: Recursive computing in the 21st century," *J. Amer. Statist. Assoc.*, vol. 97, no. 457, pp. 337–351, 2002.

[12] V. Krishnamurthy and J. B. Moore, "Online estimation of hidden Markov model parameters based on the Kullback–Leibler information measure," *IEEE Trans. Signal Process.*, vol. 41, pp. 2557–2573, Aug. 1993.

[13] V. Krishnamurthy and G. G. Yin, "Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime," *IEEE Trans. Inf. Theory*, vol. 48, pp. 458–476, Feb. 2002.

[14] J. J. Ford and J. B. Moore, "Adaptive estimation of HMM transition probabilities," *IEEE Trans. Signal Process.*, vol. 46, pp. 1374–1385, May 1998.

[15] F. LeGland and L. Mevel, "Recursive estimation in hidden Markov models," in *Proc. Conf. Decision Contr.*, 1997, vol. 4.

[16] T. Rydén, "On recursive estimation for hidden Markov models," *Stoch. Proc. Applicat.*, vol. 66, pp. 79–96, 1997.

[17] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Beyesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[18] K. S. Chan, "Asymptotic behavior of the Gibbs sampler," *J. Amer. Statist. Assoc.*, vol. 88, pp. 320–326, 1993.

[19] J. S. Liu, W. H. Wong, and A. Hong, "Covariance structure and convergence rate of the Gibbs sampler with various scan," *J. R. Statist. Soc. (B)*, vol. 57, pp. 157–169, 1995.

[20] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[21] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. New York: Wiley, 1994.

[22] D. Crisan and A. Doucet, "Convergenge of sequential Monte Carlo methods," CUED-F-INFENG, 2000, Tech. Rep. 381.

[23] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, Feb. 2002.

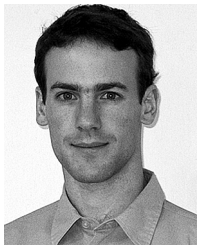
[24] C. Andrieu and A. Doucet, "Online expectation-maximization type algorithms for parameter estimation in general state space models," *in Proc. Acoust., Speech, Signal Process., ICASSP'03*, vol. 6, pp. 69–72, Apr. 2003.

[25] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer-Verlag, 2001.

[26] W. R. Gilks and C. Berzuini, "Following a moving target – Monte Carlo inference for dynamic Bayesian models," *J. R. Statist. Soc. (B)*, vol. 63, no. 1, pp. 127–146, 2001.

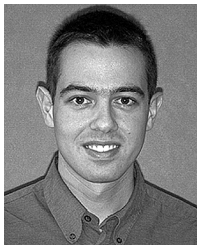
[27] P. Fearnhead, "MCMC, sufficient statistics and particle filters," *J. Computat. Graph. Statist.*, vol. 11, pp. 848–862, 2002.

[28] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.
[29] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.
[30] J. Tugnait and A. Haddad, "A detection-estimation scheme for state estimation in switching environment," *Automatica*, vol. 15, pp. 477–481, 1979.
[31] P. Fearnhead, Sequential Monte Carlo methods in filter theory Univ. Oxford, 1998, Ph.D..
[32] P. Fearnhead and P. Clifford, "On-line inference for hidden Markov models via particle filters," *J. R. Statist. Soc. (B)*, vol. 65, no. 4, pp. 887–899, 2003.
[33] S. J. Julier and J. K. Uhlmann, "The scaled unscented transformation," in *Proc. IEEE Amer. Contr. Conf.'02*, May 2002, pp. 4555–4559.
[34] *N. Simulator 2*, [Online]. Available: http://www.isi.edu/nsnam/ns



Tom Vercauteren (S'05) received the degree from Ecole Polytechnique, Paris, France, in 2003 and the M.S. degree from the Department of Electrical Engineering, Columbia University, New York, in 2004. He is currently working towards the Ph.D. degree in INRIA, Sophia-Antipolis, France.

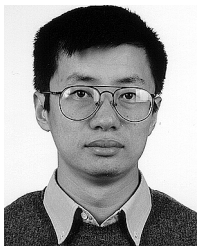
His research interests are in the area of statistical signal processing.



Alberto Lopez Toledo (S'05) received the degree in computer engineering (with highest honors) and the M.Phil. degree in computer science from the University of Murcia (UMU), Spain, in 1999 and 2002, respectively, and the M.S. degree in electrical engineering from Columbia University, New York, in 2003. He is currently working towards the Ph.D. degree in electrical engineering at Columbia University.

His research interests are in the area of wireless networking and cross-layer design.

Mr. Toledo received Spain's National Academic Excellence Award, the Edwin Howard Armstrong Memorial Award, and the La Caixa Foundation and Rafael del Pino Foundation fellowship.



Xiaodong Wang (S'98–M'98–SM'04) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China; the M.S. degree from Purdue University, West Lafayette, IN; and the Ph.D. degree from Princeton University, Princeton, NJ, all in electrical engineering.

From July 1998 to December 2001, he was with the Department of Electrical Engineering, Texas A&M University. Since January 2002, he has been with the Department of Electrical Engineering, Columbia University, New York. His research interests fall in the general areas of computing, signal processing and communications, and has published extensively in these areas. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Englewood Cliffs, NJ: Prentice-Hall, 2003). His current research interests include wireless communications, statistical signal processing, and genomic signal processing.

Dr. Wang received the 1999 NSF 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.