Signal Processing 96 (2014) 274–283

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Physical layer metrics estimation for CSMA/CA networks using a Markov modeling and source enumeration

Mohamed Rabie Oularbi [a,*], Saeed Gazor [b], Abdeldjalil Aïssa-El-Bey [a], Sébastien Houcke [a]

[a] Institut Mines-Telecom, Telecom Bretagne, UMR CNRS 3192 Lab-STICC, Technopôle Brest Iroise-CS 83818, 29238 Brest Cedex, France
[b] Department of Electrical and Computer Engineering, Queen's University, Kingston, Canada

ARTICLE INFO

Article history:
Received 14 August 2012
Received in revised form
19 September 2013
Accepted 22 September 2013
Available online 4 October 2013

Keywords:
Cognitive radio
Vertical handover
Physical layer metrics
IEEE 802.11
WiFi
Rank tracking
Markov chains
Multiple antennas

ABSTRACT

The existence of multiple wireless networks with different radio access technologies and protocols makes the radio environment *heterogeneous*. In order to provide the best Quality of Service available from the active networks, and satisfy the concept of always best connected, one can take advantage of this heterogeneity by developing multi-mode terminals able to smartly switch from one interface to another. This switching process, known as *vertical handover* (VHO), requires some relevant metrics to be measured by the terminal in order to decide whether to trigger a VHO or not. Using multiple antennas, we propose to track the number of active sources and employ the results in CSMA/CA networks for VHO. The proposed algorithm is developed using a Markov chain model for sources enumeration at any given time. We also use a three state Markov model for CSMA/CA networks and show how this algorithm can be applied to recursively obtain two informative metrics about the channel state, namely the channel occupancy rate and the collision rate. Numerical simulations confirm that the proposed algorithm performs well for practical SNR values. The proposed algorithm relies on a physical layer sensing and requires no connection to the access point, no synchronization, no signal demodulation and no frame decoding. This particularity ensures a seamless handover with a time/energy economy.

1. Introduction

Wireless communication standards are facing a proliferation leading to the coexistence of different networks belonging to different administrative domains. Some of these networks operate in different frequency bands, like LTE, WiMAX, 3G, etc., while some of them operate in the same band like WiFi and Bluetooth. In parallel to this, there is also an emerging trend to provide ubiquitous wireless access to mobile terminals while maintaining the Quality of Service (QoS) requirements for upper layers applications. In order to make such an operation possible, and to take advantage of this coexistence, the terminal needs to be smart enough to sense the surrounding environment and switch cognitively from one standard to another, e.g., if the first one cannot satisfy the required QoS. This switching process is known as *vertical handover* (VHO). The VHO is possible only if the mobile terminal could sense its environment and accordingly/cognitively reconfigure its communication parameters to better adapt with the channel conditions [1]. This operation includes two stages: sense and decide. This paper deals only with metrics estimation (part of the sensing task). After verifying that the link quality in terms of Signal to Noise Ratio meets its requirements, prior to perform a vertical handover, the terminal has to evaluate some relevant metrics that are

* Corresponding author. Tel.: +33 2 29 00 14 93; fax: +33 2 29 00 10 12.
E-mail address: mohamed.oularbi@telecom-bretagne.eu (M. Rabie Oularbi).

0165-1684/$ - see front matter
http://dx.doi.org/10.1016/j.sigpro.2013.09.012

informative about the available networks' QoS. Indeed, one should note that the QoS metrics are only complementary to the classical link quality based on power strength.

In the context of vertical handover, only the passive estimation is relevant, since the terminal seeks to know a priori if a network satisfies its QoS needs without wasting time and power to get connected to this network [2]. To satisfy those conditions, the algorithm proposed here relies on a physical layer sensing and requires no connection to the access point, no synchronization, no signal demodulation and no frame (data packet) decoding.

Our networks of interest are based on a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol like WiFi (IEEE 802.11). It has been highlighted in [3,4] that the usage of the channel bandwidth in a CSMA/CA system can be approximated by the ratio between the time in which the channel status is busy according to the NAV (Network Allocation Vector) settings and the considered time interval. The higher the traffic, the larger the NAV busy occupation, and vice versa. Once we read a NAV value during a certain time window, the available bandwidth and access delay can be estimated [5]. The main drawback of this method is that it requires the station to be connected to the access point in order to obtain the NAV information from the header, this may increase the decision time if many standards or access points (APs) are detected. An alternative technique has been presented in [6,7] for the estimation of the channel occupancy rate. This technique relies on a physical layer sensing, and thus presents the advantage of time economy since it requires no synchronization and no association to the AP. The authors succeeded in showing the relevance of this physical layer metric by performing experimental measurement in [8] under different scenarios. Unfortunately, this algorithm requires the knowledge of the noise variance and does not give any information about the collision rate which is a complementary metric to the channel occupancy rate. In [9,10] it has been shown that the collision rate depends on the number of users connected to an access point, and in [11] it is stated that the mean of the MAC (Media Access Control) delay associated with a transmission by a particular source is increasing exponentially versus the collision probability. Thus, with a higher collision rate, a lower QoS is available. To the best of our knowledge, the only passive technique for collision detection has been studied in [7,8], in contrast to the MAC layer based technique proposed in [12]. Unfortunately, this method needs the knowledge of the edges of the frame (the start and end points) in order to apply any information theoretic criterion to test if a collision has occurred. Since a cognitive receiver does not know the exact times when the frame starts and ends, this technique cannot be directly applied and requires the edges to be estimated as additional unknown parameters.

From a physical layer perspective, the problem of estimating the channel occupancy rate and the collision rate can be viewed as a source enumeration problem. In fact, the data frames of each user can be viewed as a signal emitted by one source, and the collided frames as a mixture of two or more sources. Rank tracking is a classical model order selection problem that arises in a variety of important statistical signal and array processing, however it is rarely addressed in the literature [13–16]. In this paper, to achieve this recursive estimation, we propose a new algorithm based on a Markov chain model. Assuming that a maximum of M sources can be present at any observation time, we model the system by an $M+1$ states chain (see Fig. 1, M states for the sources plus one for the case that no source is active). Our proposed algorithm estimates the number of sources, recursively, in four steps and is an extension of the proposed one in [17] dedicated to spectrum sensing.

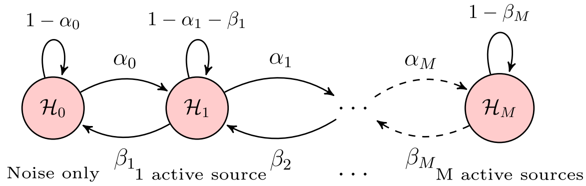

Fig. 1. Markov chain model for adaptive source enumeration.

The application to CSMA/CA based networks is then straightforward. In several papers, such networks have been modeled by a Markov process and their performance have been studied [18–20]. In most of these papers, a two state Markov chain is used (good state and bad state). In our model, we employ a chain with three states: no transmission, one source, and collision. The objective of the channel sensing is to detect these states. After a period of time, we can easily estimate the channel occupancy rate as the ratio of the total time intervals where the channel is declared as busy (the number of sources is not zero) to the total observation time. In addition, the collision rate is estimated as the ratio of the number of frames detected to be involved in a collision to the total number of frames detected during the observation time. One should note that, we do not aim to estimate the total number of users or competing stations [10,21–23], but only to detect if the sensed frame is a result of a collision or not and than deduce the collision rate.

The remainder of the paper is organized as follows: in Section 2, we formulate the problem and present the Markov model for tracking the number of sources. In Section 3, we propose a four step algorithm based on the Markov model for the number of source tracking, the algorithm is presented for a general case and can be used in any other application. In Section 4, we explain how the proposed algorithm is used for a WiFi network. In Section 5.1, we evaluate the performance of the proposed algorithm on the estimation of the channel occupancy rate and the collision rate for a WiFi network. Finally, Section 6 concludes the paper.

Throughout this paper we denote $\mathbb{E}[\cdot]$ for mathematical expectation, $(\cdot)^T$ for matrix transposition, $(\cdot)^H$ for complex conjugate transposition, $\|\cdot\|$ for Frobenius norm, $|\cdot|$ for absolute value, $\odot$ for the element-wise vector product, $\mathbf{I}_N$ for $N\times N$ identity matrix, $\leftarrow$ for overwriting, and diag$(\cdot)$ for diagonal matrix with entries given by elements of $(\cdot)$.

2. Model

Let us consider a receiver equipped with N antennas, and $\mathbf{x}(k)=[x_1(k),\ldots,x_N(k)]^T$ denote the received signal at the time instant k. The signal $\mathbf{x}(k)$ is a mixture of maximum

$K<N$ independent signals as follows:

$$\mathbf{x}(k)=\mathbf{H}(k)\mathbf{s}(k)+\mathbf{w}(k), \tag{1}$$

where $\mathbf{H}(k)\in\mathbb{C}^{N\times K}$ is the channel matrix, $\mathbf{s}(k)\in\mathbb{C}^{K\times 1}$ is the transmitted data vector and $\mathbf{w}(k)\in\mathbb{C}^{N\times 1}$ is a zero mean additive white Gaussian noise with a variance of $\sigma^2(k)$ which is independent of $\mathbf{s}(k)$.

The sources are rising and vanishing upon the time. As a result, the number of active sources K varies with time k and needs to be estimated iteratively. The classical rank estimation methods are computationally expensive. Thus, we here propose a new approach using a Markov model. Since, it is very unlikely that more than one source vanish or appear together at a given time instance, we can use the Markov chain in Fig. 1, where states/hypothesis are defined as follows:

$$\begin{cases}\mathcal{H}_0 & \text{Only noise is observed,}\\ \mathcal{H}_1 & \text{One station is transmitting,}\\ \mathcal{H}_2 & \text{Two stations are transmitting,}\\ \vdots & \vdots\\ \mathcal{H}_M & M \text{ stations are transmitting,}\end{cases} \tag{2}$$

where M is known and denotes the maximum number of simultaneously active sources. We assume that $K\le M<N$. This assumption means that the number of simultaneously transmitting stations M is less than the number of antennas. This is not to assume that the whole number of stations connected to the access point is less than the number of antennas. This assumption is generally realistic for many applications such as in IEEE 802.11 as long as we have more than two antennas. For example in IEEE 802.11, the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol is used to minimize the collisions, i.e., the collision phenomena occur rarely and amongst a very small number of stations, practically two or three but it is very unlikely that more stations are simultaneously involved in one collision. Without a priori knowledge about M, one can set M to $N-1$. In Fig. 1, the transition probabilities α_i for $i=0,\dots,M-1$ represent the probability that the channel is occupied by $i+1$ sources given that it was occupied by i sources in the previous sensing time. We consider the number of collisions as the number of frames/data packet issued from a collision of two or more sources. The probabilities β_i for $i=1,\dots,M$ are the probabilities that $i-1$ sources are present at the present sensing time given that i sources were present in the previous sensing time. We must note that at a given time instant we have multi-hypothesis composite test problem with some a priori estimates for the unknown parameters.

In our model, the mobile stations signals are assumed as zero-mean, white, independent and circularly symmetrical complex Gaussian random variables with variances of $\{\varsigma_i^2(k)\}_{i=1}^K$ at time k. These variances depend on the power of transmitters, the channel gains which are unknown and vary with time. We assume that these variances are almost constant or change smoothly from each sensing time to the next. Thus, the distribution of the random vector process $\mathbf{x}(k)$ is characterized by its covariance matrix at time k, shown by

$$\mathbf{R}(k)=\mathbb{E}[\mathbf{x}(k)\mathbf{x}^H(k)]. \tag{3}$$

Note that it is implicitly assumed that the process $\mathbf{x}(k)$ is non-stationary. The non-stationarity is for two different reasons. The first one is that the number of the components of $\mathbf{s}(k)$ changes with time as sources vanish or arise. The second reason is that the channel responses $\mathbf{H}(k)$ and $\{\varsigma_i^2(k)\}_{i=1}^K$ vary with time. However, we assume that $\mathbf{H}(k)$ and $\{\varsigma_i^2(k)\}_{i=1}^K$ vary very smoothly with time. In other words, our assumption is that only the changes in the number of sources create sudden changes in the eigenstructure of the unknown matrix $\mathbf{R}(k)$. The eigenvalue decomposition (EVD) of the matrix $\mathbf{R}(k)$ can be written as

$$\mathbf{R}(k)=\mathbf{U}(k)\mathbf{\Lambda}(k)\mathbf{U}^H(k), \tag{4}$$

$$\mathbf{U}(k)\mathbf{U}^H(k)=\mathbf{I}_N. \tag{5}$$

where $\mathbf{\Lambda}(k)=\text{diag}[\lambda_1(k),\lambda_2(k),\dots,\lambda_N(k)]$, $\mathbf{U}(k)=[\mathbf{u}_1(k),\mathbf{u}_2(k),\dots,\mathbf{u}_N(k)]$, $\lambda_i(k)$'s are the eigenvalues and $\mathbf{u}_i(k)$'s are the orthonormal eigenvectors. Without loss of generality, it is convenient to assume $\lambda_1(k)\ge\lambda_2(k)\ge\cdots\ge\lambda_N(k)\ge 0$.

Under Hypothesis $\mathcal{H}_m$, the autocorrelation matrix expressed in (3) can be written as

$$\mathbf{R}(k)=\mathbf{H}(k)\begin{bmatrix}\varsigma_1^2(k) & 0 & \cdots & 0\\ 0 & \varsigma_2^2(k) & \cdots & 0\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & & \varsigma_m^2(k)\end{bmatrix}\mathbf{H}^H(k)+\sigma^2(k)\mathbf{I}_N. \tag{6}$$

Since the number of sources can change with time, the above EVD structure is also subject to modification and thus needs to be tracked. A simple but computationally exhaustive form of subspace tracking consists of performing the estimation of the all EVD parameters for every new observation $\mathbf{x}(k)$. To reduce computational complexity, we are interested in algorithms that can estimate $\mathbf{\Lambda}(k)$ and $\mathbf{U}(k)$ adaptively using the previous estimates of $\mathbf{\Lambda}(k-1)$ and $\mathbf{U}(k-1)$. These subspace algorithms seek to satisfy the following equation:

$$\widehat{\mathbf{U}}(k)\widehat{\mathbf{\Lambda}}(k)\widehat{\mathbf{U}}^H(k)=(1-\varepsilon)\widehat{\mathbf{U}}(k-1)\widehat{\mathbf{\Lambda}}(k-1)\widehat{\mathbf{U}}^H(k-1)+\varepsilon\mathbf{x}(k)\mathbf{x}^H(k) \tag{7}$$

where the constant $\varepsilon\in(0,1)$ is called the forgetting factor and determines the effective length of the exponential window $(2-\varepsilon)/\varepsilon$ [24]. A larger value for $\varepsilon<1$ results in a better tracking capability in an environment where $\mathbf{H}(k)$ and $\{\varsigma_i^2(k)\}_{i=1}^K$ vary faster with time (e.g., for larger speeds).

3. Proposed algorithm

In our algorithm, to update the EVD, we propose to use the PROTEUS-1 algorithm introduced by Champagne et al. [25]. There are several other alternative algorithms that could be used. This algorithm is computationally efficient using a CORDIC (COordinate Rotation DIgital Computer) processor, as it uses only plane rotations for updating the eigenvectors and directly provides the set of orthonormal eigenvectors, which makes it a well suited subspace tracking algorithm for our multi-hypothesis problem which deals with non-stationary data.

In (6), the number of active sources is the number of eigenvalues of $\mathbf{R}(k)$ which are larger than $\sigma^2(k)$ and is in fact a function of the time k and needs to be determined.

We propose the following four step soft enumeration algorithm to determine the probability of each hypothesis in (2), recursively.

Step 0: *Initialization step*. We define $P^m_{k|k-1}$ as the a priori probability of having m active sources at time k:

$$P^m_{k|k-1} = P[\mathcal{H}_m \text{ at time } k|\Omega(k-1)] \tag{8}$$

where the hypothesis $\mathcal{H}_m$ denotes the event that m sources are active and $\Omega(k-1)$ denotes the available information and observation at time instant $k-1$. At the instant $k=0$, in the absence of any knowledge, we initialize the a priori probabilities as $P^m_{0|-1} = 1/(M+1), \forall m = 0, \ldots, M$. The EVD of the received signal $\widehat{\mathbf{U}}(0)$ and $\widehat{\mathbf{\Lambda}}(0)$ is initialized randomly.

Step 1: *Sense and Update*. Observe $\mathbf{x}(k)$ and perform the preprocessing and normalization steps in PROTEUS-1 [25] as follows:

1. *Projection*: $\mathbf{y}(k) = \widehat{\mathbf{U}}^H(k-1)\mathbf{x}(k)$;
2. *Mapping the pair* $(\mathbf{y}(k), \widehat{\mathbf{U}}(k-1))$:

   $$\mathbf{y}(k) \leftarrow \mathbf{D}^H(k)\mathbf{y}(k), \quad \widehat{\mathbf{U}}(k) \leftarrow \widehat{\mathbf{U}}(k-1)\mathbf{D}(k).$$

   using $\mathbf{D}(k) = \mathrm{diag}(y_1(k)/|y_1(k)|, \cdots, y_1(k)/|y_n(k)|)$ where $y_i(k)$ denotes the ith entry of the vector $\mathbf{y}(k)$. This mapping removes the phase of elements of $\mathbf{y}(k)$ and incorporates them into the corresponding column of $\widehat{\mathbf{U}}(k)$. In this way, the vector $\mathbf{y}(k)$ becomes real and the new pair represents the same EVD.
3. *Reordering/sorting*: Find the permutation matrix $\mathbf{\Pi}$ which reorders the entries of $\mathbf{y}(k)$ in the decreasing order and

   $$\mathbf{y}(k) \leftarrow \mathbf{\Pi}^T\mathbf{y}(k), \quad \widehat{\mathbf{U}}(k) \leftarrow \widehat{\mathbf{U}}(k)\mathbf{\Pi}, \quad \widehat{\mathbf{\Lambda}}(k) \leftarrow \mathbf{\Pi}^T\widehat{\mathbf{\Lambda}}(k-1)\mathbf{\Pi}.$$

5. $\widehat{\lambda}_i(k) \leftarrow (1-\varepsilon)\widehat{\lambda}_i(k) + \varepsilon y_i^2(k)$

where $\mathbf{G}_{ij}(\theta)$ is the well-known plane or Givens rotation matrix defined by

$$\mathbf{G}_{ij}(\theta) = \begin{bmatrix} I_{i-1} & & & & \\ & \cos(\theta) & \cdots & \sin(\theta) & \\ & \vdots & I_{j-i-1} & \vdots & \\ & -\sin(\theta) & \cdots & \cos(\theta) & \\ & & & & I_{N-j} \end{bmatrix}. \tag{9}$$

The computational cost of this step, as stated in [25], is equal to $6N^3 + 15.5N^2 + (\nu + 7.5)N$ flops, where ν is used as a common flop count for the square root operation.

Step 2: *Calculation of Log-Likelihood functions*. In this step, we calculate the Log-Likelihood functions using the observed samples at time k. Under hypothesis $\mathcal{H}_m$, we estimate the noise variance by

$$\widehat{\sigma}^2_m(k) = \frac{1}{N-m}\sum_{i=m+1}^{N} \widehat{\lambda}_i(k) \quad \text{under } \mathcal{H}_m. \tag{10}$$

This expression represents the maximum likelihood (ML) estimator of the noise variance only for a rectangular window, whereas we are using an exponential window. Our simulation results reveal that as the number of sources changes with time the estimate of noise variance will vary significantly.[1] Thus, we used the following alternative estimator under $\mathcal{H}_0$:

$$\widehat{\sigma}^2_0(k) = \widehat{\lambda}_N(k) \quad \text{under } \mathcal{H}_0. \tag{11}$$

This estimator is more robust than $\widehat{\sigma}^2_0(k) = (1/N)\sum_{i=1}^{N}\widehat{\lambda}_i(k)$ to the variation of number and power of sources and is more suitable for environments with fast changing dynamics. We must note that $\widehat{\lambda}_i(k)$'s are eigenvalues of (7) and are biased estimators for the true eigenvalues of $\mathbf{R}(k)$. In fact, the bias using the following estimator $\widehat{\sigma}^2_0(k) = \widehat{\lambda}_N(k)$ is less significant than the one when using $\widehat{\sigma}^2_0(k) = (1/N)\sum_{i=1}^{N}\widehat{\lambda}_i(k)$. Obviously some performance gain can be obtained using more elaborate estimators such as the recently proposed ones in [26]. Given these estimates the Log-Likelihood functions for $m \geq 1$ are estimated as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{x}(k)|\mathcal{H}_m) = {} & \log\left(\frac{\varepsilon}{m}\right) - (N-m)\log\left(\pi\widehat{\sigma}^2_m(k)\right) \\ & - \frac{1}{\widehat{\sigma}^2_m(k)}\|\mathbf{x}(k) - \mathbf{V}_m(k)\mathbf{V}_m(k)^H\mathbf{x}(k)\|^2 \\ & - \sum_{j=1}^{m}\left(\log\left(\pi\widehat{\lambda}_j(k)\right) - \frac{1}{\widehat{\lambda}_j(k)}|\mathbf{u}_j^H(k)\mathbf{x}(k)|^2\right), \end{aligned} \tag{12}$$

and under $\mathcal{H}_0$ as

$$\mathcal{L}(\mathbf{x}(k)|\mathcal{H}_0) = -N\log\left(\pi\widehat{\sigma}^2_0(k)\right) - \frac{1}{\widehat{\sigma}^2_0(k)}\|\mathbf{x}(k)\|^2, \tag{13}$$

where $\mathbf{V}_m(k) = [\mathbf{u}_{m+1}(k)\ \mathbf{u}_{m+2}(k) \cdots \mathbf{u}_N(k)]$. The term $\log(\varepsilon/m)$ is an additional penalty function introduced to avoid over-estimation. The reason is that the larger values of m involve more unknown parameters which result in a larger bias in the estimation of the Log-Likelihood functions. The function $\log(\varepsilon/m)$ is obtained in an empirical manner comforted by simulations and may be justified using the asymptotic distribution of the eigenvalues [24]. The overall computational cost of this step is equal to $N^2 + 2N + M(N^2 + N + 4) + 9$.

Step 3: *Updating posterior probabilities*. In this step, we combine the priori probabilities $P^m_{k|k-1}$ defined in (8), and the Log-Likelihood functions in (12) and (13), to obtain the *posterior* probabilities defined as

$$P^m_{k|k} = P[\mathcal{H}_m \text{ at time } k|\Omega(k)]. \tag{14}$$

In fact, $P^m_{k|k}$ and $P^m_{k|k-1}$ denote the estimated probabilities of presence of m sources at the sensing time k, respectively with or without the use of the current available vector of observation $\mathbf{x}(k)$. Let $\mathbf{P}_{k|k} = [P^0_{k|k}, \ldots, P^M_{k|k}]^T$ and $\mathbf{P}_{k|k-1} = [P^0_{k|k-1}, \ldots, P^M_{k|k-1}]^T$ denote the vectors containing the $M+1$

[1] We observed that as the number of sources changes, a leakage from the signal subspace affects the noise subspace and results in increased error in the estimations of the noise power.

posterior and a priori probabilities, respectively, and

$$\mathbf{F}(k) = [\exp(\mathcal{L}(\mathbf{x}(k)|\mathcal{H}_0)), \ldots, \exp(\mathcal{L}(\mathbf{x}(k)|\mathcal{H}_M))]^T \quad (15)$$

be the vector containing the $M+1$ likelihood values where $(m+1)$th element of $\mathbf{F}(k)$ represents an estimate of $f(\mathbf{x}(k)|\mathcal{H}_m)$. By exploiting the Bayes rule, we obtain posterior probability vector as

$$\mathbf{P}_{k|k} = \frac{1}{\mathbf{F}^T(k)\mathbf{P}_{k|k-1}} \mathbf{F}(k) \odot \mathbf{P}_{k|k-1}. \quad (16)$$

A decision can be made on the rank, according to the most probable hypothesis, i.e.

$$\widehat{K} = \arg\max_m \{P^0_{k|k}, \ldots, P^M_{k|k}\}. \quad (17)$$

The computational cost of this step is equal to $3(M+1)$.

Step 4: *Prediction of priori probabilities*. A prediction of the priori probabilities $P^m_{k+1|k}$ for the next sensing time is needed to compute the posterior probabilities expressed in (16). We use the Markov model illustrated in Fig. 1 to predict these probabilities as

$$\mathbf{P}_{k+1|k} = \mathbf{T}\mathbf{P}_{k|k} \quad (18)$$

where the transition matrix of the Markov chain $\mathbf{T}$ as shown in Fig. 1 is given by

$$\mathbf{T} = \begin{bmatrix} 1-\alpha_0 & \alpha_0 & 0 & \cdots & 0 \\ \beta_1 & 1-\alpha_1-\beta_1 & \alpha_1 & \ddots & \vdots \\ 0 & \beta_2 & 1-\alpha_2-\beta_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \alpha_{M-1} \\ 0 & \cdots & 0 & \beta_M & 1-\beta_M \end{bmatrix}. \quad (19)$$

A flow-chart of this algorithm is presented in Fig. 2 and Algorithm 1.

Algorithm 1. Adaptive source enumeration.

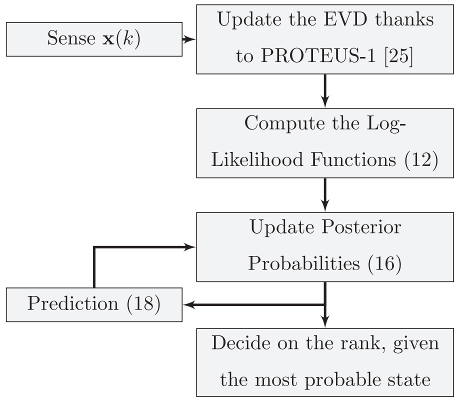

Fig. 2. Flow chart of the proposed algorithm.

The computational cost of this last step is equal to $(M+1)^2$, thus, the overall computational cost of the proposed algorithm is equal to the sum of the CCs of each step, that is $6N^3 + (M+16.5)N^2 + (M+9.5+\nu)N + 7M + 12$.

4. Application to CSMA/CA based wireless networks

The proposed algorithm in the previous section gives an estimate of the number of active sources at each sensing time, recursively. For the context of metric estimation, the soft information in $\mathbf{P}_{k+1|k}$ can be used to more accurately extract other information. In particular, the values $1-P^0_{k|k}$ and $1-P^0_{k|k}-P^1_{k|k}$ represent instantaneous estimates of the probabilities that the system is occupied and has a collision, respectively. Thus, averaging these values over time we can estimate the occupancy and collision rates of samples.

We here apply this proposed algorithm to estimate the channel occupancy rate and the collision rate of a WiFi access point. The IEEE 802.11 (WiFi) communication relies on the protocol Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). CSMA/CA is a multiple access method, any node which wishes to transmit data must first listen to the channel for a predetermined amount of time and determine whether or not the channel is used by another node. Only if the channel is identified to be "idle," then the node is permitted to begin the transmission process. Otherwise if the channel is sensed as "busy," the node defers its transmission and waits for a random period of time called "backoff". There are Inter Frame Spacing (IFS) time intervals between any two consecutive frames during which the observed signal consists of only noise samples. Whereas during data frames, we have signal plus noise. During transmission of data frames, the observed samples can be from one source plus noise or from two (or more) sources in the case of a collision. Thus from a physical layer point of view, the WiFi communication can be modeled as a Markov chain with three states:

$$\begin{cases} \mathcal{H}_0 & \text{Only noise is observed (Interframe spacing);} \\ \mathcal{H}_1 & \text{Only one mobile station is transmitting;} \\ \mathcal{H}_2 & \text{Two or more stations are transmitting (Collision).} \end{cases} \quad (20)$$

The Markov chain associated is illustrated in Fig. 3, where the transition probability $1-\alpha_0 \in (0,1)$ represents the probability of idle channel when the channel was idle in the previous sensing time, $1-\alpha_1-\beta_1 \in (0,1)$ represents the probability of busy channel when the channel was busy in the previous sensing time. Finally $1-\beta_2 \in (0,1)$ represents the probability of a collision when a collision was detected in the previous time. The transition matrix of the chain is

$$\mathbf{T} = \begin{bmatrix} 1-\alpha_0 & \beta_1 & 0 \\ \alpha_0 & 1-\alpha_1-\beta_1 & \beta_2 \\ 0 & \alpha_1 & 1-\beta_2 \end{bmatrix}. \quad (21)$$

We assume that the cognitive device is equipped with more than two antennas $N>2$. In (21) we ignored the transition probability from $\mathcal{H}_0$ to $\mathcal{H}_2$ (the true value is not zero as two stations may incidently start to transmit at the exact same time). However, the matrix in (21) allows to detect changes from $\mathcal{H}_0$ to $\mathcal{H}_2$ in two time steps; the probability vector $\mathbf{P}_{k|k}$ first indicates that the "one active source" state is the most likely state and shortly after moves to the "two active sources". Obviously, using a non-zero probability for that entry in (21) results in improvement of detection of such unlikely events at the expense of degradation for more likely events. Note that the algorithm remains unchanged if we use a non-zero entry

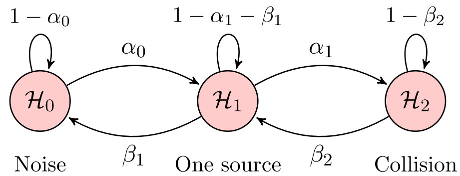

Fig. 3. Markov model for the PHY layer of a WiFi channel.

for the probabilities of transitions between $\mathcal{H}_0$ and $\mathcal{H}_2$. We must also note that once in $\mathcal{H}_2$, the users will backoff for some time and therefore the transition matrix can be modified in order to reflect the behavior of backoff timers.

5. Simulations

5.1. Tracking capability of the proposed algorithm

Simulations have been assessed on WiFi 802.11n signals. WiFi signals are OFDM signals with a total number of 64 sub-carriers and a cyclic prefix of length 16. The receiver is assumed to be equipped with $N=4$ antennas. The channel $\mathbf{H}(k)$ is a set of complex numbers randomly chosen according to a Gaussian law with zero mean and a unitary variance. The channel is also assumed to be a slow fading channel. The SNR is defined by frame and for the ith source it is set as $\mathrm{SNR}_i = \varsigma_i^2/\sigma^2$. The forgetting factor used for the PROTEUS-1 algorithm is $\varepsilon = 0.05$. The moments of the Markov chain illustrated in Fig. 3 are as follows: $\alpha_0 = 10^{-4}$, $\alpha_1 = \alpha_0/2$, $\beta_1 = 10\alpha_0$ and $\beta_2 = 40\beta_1$. These probabilities represent the dynamic of users and have been determined empirically by simulations.

The matrices $\widehat{\mathbf{U}}(0)$ and $\widehat{\mathbf{\Lambda}}(0)$ are initialized by first observing 10 samples on the channel of interest and computing the true EVD of the set of observation. In Fig. 4, we plot an example of a sensed communication, the observation is made of two frames: the first frame is a product of a collision between a frame with an observed SNR=15 dB at the cognitive observer, the second frame is emitted by a single source observed with a SNR=10 dB. Fig. 4a represents the magnitude of the observed signal. Fig. 4b represents the posteriori probabilities of being under each hypothesis. Finally, Fig. 4c represents the true and estimated rank using

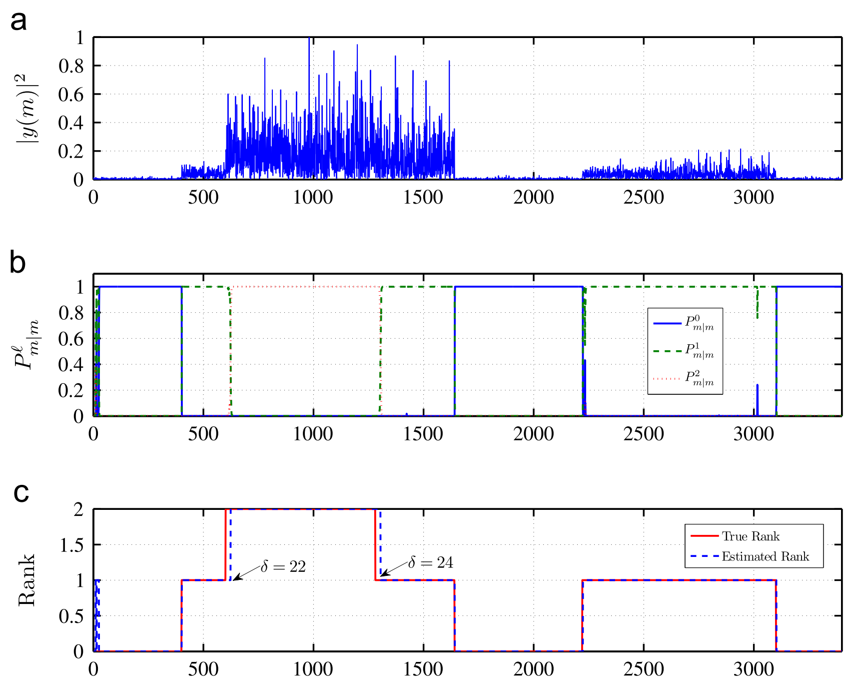

Fig. 4. Example for tracking capacity of the algorithm: (a) magnitude of the observed signal, (b) *a posteriori* probabilities and (c) real and estimated rank.

the proposed algorithm. The decision on the rank (illustrated in the bottom) is made according to the hypothesis presenting the maximum posteriori probability. We observe that under these scenarios, our algorithm has a good tracking capability. The overestimation that appears in the beginning is negligible and may be due to the fact that the algorithm needs time to converge. Some delays during the transition from a rank to another appear, for example, the transition from the rank one to two in the figure occurs with a delay of 22 samples, and from the rank two to one occurs after 24 samples. These delays are proportional to the eigenvalue tracking algorithm exponential window $1/\varepsilon = \frac{1}{0.05} = 20$. Thus $1/\varepsilon$ must be smaller than the coherence time of the channel.

Note that the delays from the state "*two sources*" to "*one source*" and vice versa have no impact on the performance of the proposed estimator in Section 5.2. Indeed, the length of the collision does not matter to us, it is the number of collision that is important in our case. However, the delays occurring when transiting from the states "*one source*" and "*noise*" are important because they determine the length of the frame, parameter that we are going to use when computing the channel occupancy rate.

In Fig. 5, we have conducted simulation in the same scenario as in Fig. 4 but using a transition probability α_0, 100 times greater. We remark that the tracking capacity of the proposed algorithm is affected by this action, and that the algorithm is sensitive to the choice of the transition matrix **T**.

Finally, an extended Markov chain of four states is used to estimate a number of sources up to three. The Markov chain associated to this simulation is detailed in Fig. 6. The receiver has 4 antennas. The transition probabilities are chosen as follows: $\alpha_0 = 10^{-4}$, $\alpha_1 = \alpha_0/2$, $\alpha_2 = \alpha_1/2$, $\beta_1 = 10\alpha_0$, $\beta_2 = 40\beta_1$, $\beta_3 = 10\beta_2$. The associated performance of the algorithm is plotted in Fig. 7. One can clearly see that the algorithm still behaves well as for the three state Markov model associated to the two sources collision case, using the same number of antennas. In conclusion, the proposed algorithm can be used whatever is the length of the Markov chain.

5.2. Application to CSMA/CA based networks

As stated previously the channel occupancy rate noted as C_{or} is defined as being the ratio between the amount of time where the channel is considered as being busy and the length of the observation window. According to our model illustrated in Fig. 3, the C_{or} is processed as follows:

$$C_{or} = \frac{1}{N_s}\sum_{k=1}^{N_s}\left(P^1_{k|k} + P^2_{k|k}\right) = 1 - \frac{1}{N_s}\sum_{k=1}^{N_s} P^0_{k|k}, \tag{22}$$

where N_s is the length of the observation window.

The collision rate is defined as the number of frame issued of a collision (rank > 1) divided by the total number of frames on the observation window, that is

$$R_{\text{col}} = \frac{\text{Number of collided frames}}{\text{Total number of frames}}. \tag{23}$$

Note that the temporal average of $P^2_{k|k}$ gives the collided rate of signal samples. However, the reason that we use (23) instead of the average is that $P^2_{k|k}$ in the MAC layer, the required metric of interest is the collision rate of the frames and not that of samples.

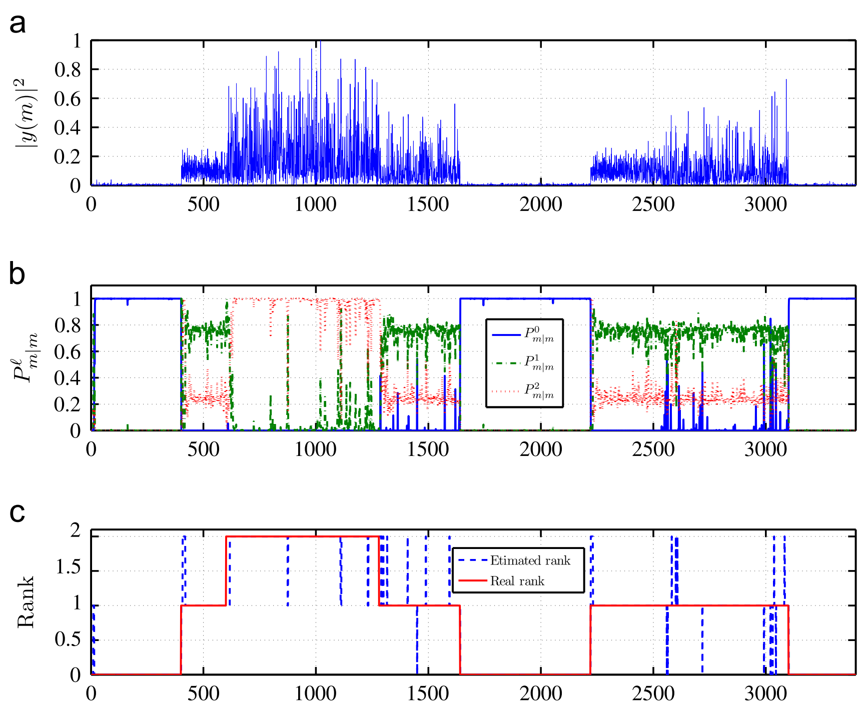

Fig. 5. Example for tracking capacity of the algorithm when changing the transition matrix elements such that α_0 is 100 times greater than the scenario of Fig. 4. (a) magnitude of the observed signal, (b) *a posteriori* probabilities and (c) true and estimated rank.

As the algorithm proposed in [8], the proposed approach suffers in some cases of fluctuations. These fluctuations mainly appear during the transition from a hypothesis to another one, and their duration is proportional to the forgetting factor ε. Specially at high SNRs, as the contrast between the eigenvalues become bigger, we noticed that these fluctuations are mainly an overestimation that rapidly vanishes with time. To overcome this problem, we utilize the smoothing algorithm proposed in [8] which has been tested experimentally on real WiFi signals and shown its efficiency. This smoothing algorithm relies on the fact that in a CSMA/CA algorithm the smallest silence period is an SIFS (Short InterFrame Spacing) and no other frame has a length smaller than it. Hence, if our algorithm meets a frame of size less than the size of an SIFS, it will automatically affect the number of sources of the frame that comes after it. Once, this smoothing operated the channel occupancy rate and the collision rate that are computed thanks to (22) and (23).

To evaluate the performance of the proposed method versus the SNR, we realized simulations under the following scenario: a WiFi communication is intercepted, the true channel occupancy rate is equal to 64.97% and the collision rate is equal to 40%. The observation window contains 7880 samples, and is constituted of 5 frames with two of them randomly issued from a collision.

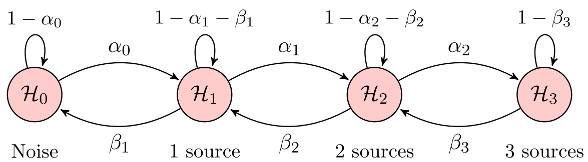

Fig. 6. Four states Markov model for detecting a three sources collision.

Fig. 8 illustrates the NMSE (Normalized Mean Square Error) of the estimation of the channel occupancy rate defined as

$$\mathrm{NMSE} = 10\,\log_{10}\left|\frac{\hat{R}-R}{R}\right|^2, \tag{24}$$

where $\hat{R}$ and R are the estimated and the true channel occupancy rates, respectively. The NMSE is estimated over 1000 Monte-Carlo runs. In this figure, we compare the performance of the proposed algorithm to the one proposed in [6] which requires the knowledge of the noise power σ^2. We observe that for Signal to Noise Ratios (SNRs) below 12 dB, the two techniques have similar performance. The Markov approach is outperformed by the approach proposed in [6] for higher SNRs. The performance of the Markov

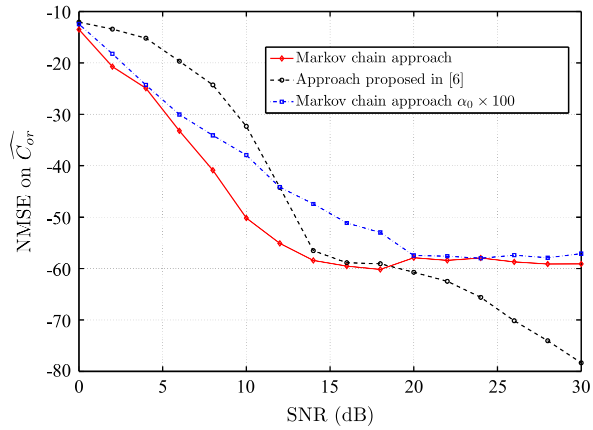

Fig. 8. NMSE on the estimation of the channel occupancy rate.

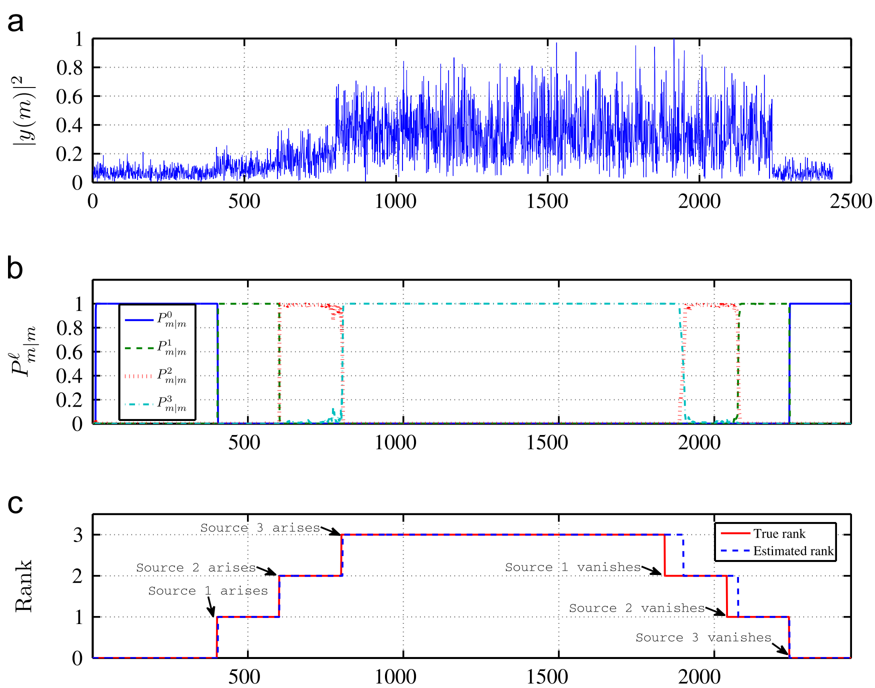

Fig. 7. Performance of the four Markov chain model. (a) magnitude of the observed signal, (b) *a posteriori* probabilities and (c) real and estimated rank.

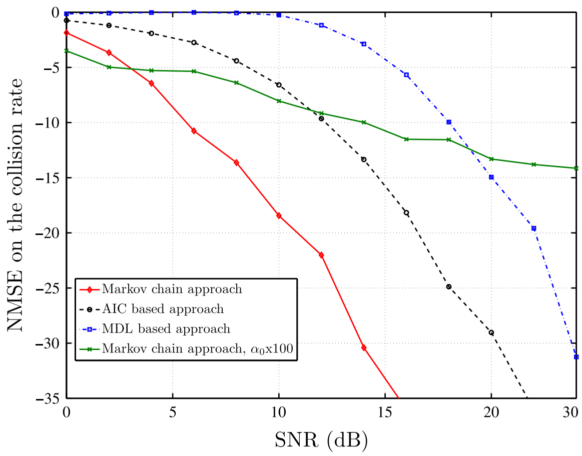

Fig. 9. NMSE on the estimation of the collision rate.

approach is very attractive since it achieves NMSE very close to the one in [6] without the knowledge of the noise power. In the same figure, the performance of the proposed algorithm is plotted when α_0 is chosen 100 times bigger. We observe that even with this bad choice of the transition probabilities the proposed algorithm still has a good estimation capability of the channel occupancy rate. Interestingly, the performance first monotonically improves as the SNR increases (typically up to SNR=15 dB) however the improvement becomes saturated for very high SNRs. This saturation is due to the fact that our estimator has a "bias", i.e., as the SNR increases the error does not tend to zero because of the bias.

Fig. 9 compares the performance of our algorithm to the ones proposed in [7,8]. We observe that the proposed approach outperforms both the techniques based on Akaike Information Criterion (AIC) and Minimum Description Length (MDL). This is mainly due to the fact that the estimation is done jointly in our approach, when on the other hand for [7,8] we first need to estimate the number of frames using [6] than extract the ones suffering of a collision. Thus, two independent sources of error are possible in that case: one in the numerator and one in the denominator. However, converse to the C_{or} estimation when α_0 became 100 times bigger the algorithm loses its accuracy and our approach is outperformed when estimating the collision rate. The main advantage of the proposed approach lies on the fact that it does not require any algorithm to detect the frame edges, when the approach proposed in [7,8] needs to know perfectly the edges of the frame to perform AIC and MDL on it and then decide whether a collision occurred or not.

Concerning the required precision. According to the experimental tests conducted by the authors in [8], the variance on the estimation of the metrics is a function of three parameters, namely the number of competing stations, the throughput and the length of the observation window. A conclusion has been made that knowing the metrics with a very high precision (up to 10^{-3}) is not required because for example a C_{or} of 60% or 60.001% will not change anything to the decision or to the Quality of Service that can be achieved. Thus a precision of 10^{-1} is largely sufficient in the PHY layer. According to the NMSE figures presented in the paper this precision is largely achieved specially in the SNR operating range of a WiFi system. According to [27], a WiFi link is classified according to the SNR as follows:

- SNR > 40 dB, excellent quality (5 bars); always associated;
- 25 < SNR < 40 dB, very good quality (3–4 bars); always associated;
- 15 < SNR < 25 dB, low quality (2 bars); always associated;
- 10 < SNR < 15 dB, very low quality (1 bar); mostly associated;
- 5 < SNR < 10 dB, no signal; not associated.

Finally, we studied the effects of the number of antennas on the performance of our algorithm, under the same scenario cited above. Figs. 10 and 11 show the NMSE on the estimation of the channel occupancy rate and the collision rate respectively. One can see that the estimator of the C_{or} behaves better for higher number of antennas specially for low SNR values. Starting from 10 dB the NMSE is constant and wonders around −50 dB, any change around this value is insignificant

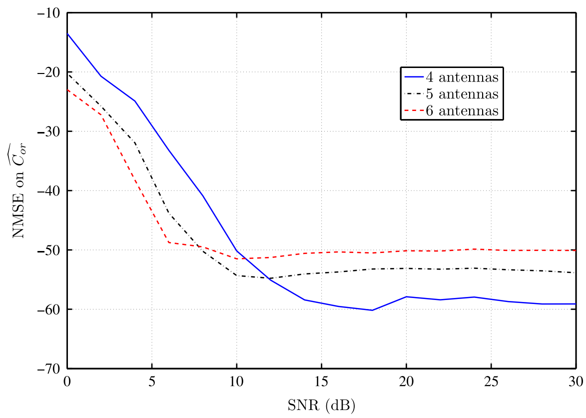

Fig. 10. Effect of the number of antennas on the estimation of the channel occupancy rate.

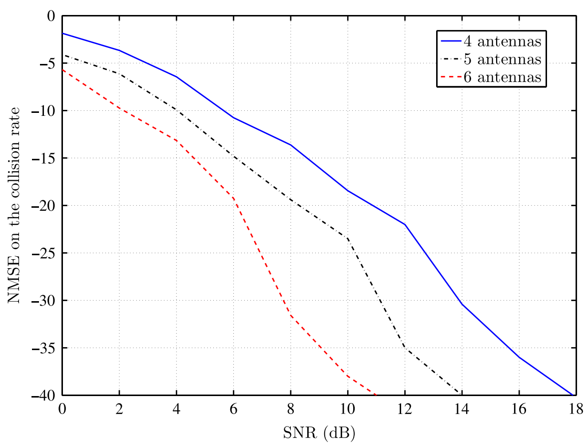

Fig. 11. Effect of the number of antennas on the estimation of the collision rate.

since a very high precision is already achieved. Concerning the collision rate, one can obviously observe from Fig. 11 that better performance is achieved for higher number of antennas. We should note that for complexity purposes a receiver with low number of antennas is always preferred, those simulations have been carried out for comparison purposes only.