



HAL
open science

Physical Layer Metrics Estimation for CSMA/CA Networks Using a Markov Modeling and Source Enumeration

Mohamed Rabie Oularbi, Saeed Gazor, Abdeldjalil Aissa El Bey, Sébastien Houcke

► **To cite this version:**

Mohamed Rabie Oularbi, Saeed Gazor, Abdeldjalil Aissa El Bey, Sébastien Houcke. Physical Layer Metrics Estimation for CSMA/CA Networks Using a Markov Modeling and Source Enumeration. Signal Processing, 2014, 96 (Part B), pp.274 - 283. 10.1016/j.sigpro.2013.09.012 . hal-00979190

HAL Id: hal-00979190

<https://hal.science/hal-00979190>

Submitted on 2 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Physical Layer Metrics Estimation for CSMA/CA Networks Using a Markov Modeling and Source Enumeration

Mohamed-Rabie Oularbi, Saeed Gazor, Abdeldjalil Aïssa-El-Bey and Sébastien
Houcke

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 present networks, and to satisfy the concept of always best connected, one can take advantage of this heterogeneity by developing multi-mode terminals allowing 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 vertical handover or not. In this paper, 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 at any given time. We also use a three state Markov model for CSMA/CA networks and show how this algorithm can be applied for recursively obtaining the two most 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. In addition, the proposed algorithm is computationally inexpensive and estimates the noise variance.

I. INTRODUCTION

Wireless communication standards are facing a proliferation leading to coexistence of different networks belonging to different administrative domains which makes the radio environment *heterogeneous*. At the same time, there is an emerging trend to provide ubiquitous wireless access to mobile terminals while the Quality of Service (QoS) needs to be maintained by the upper layers applications. In order to make such an operation possible, and to take advantage of

the heterogeneity the terminal shall be smart to sense the surrounding environment and switch cognitively from a standard to another, e.g., if the first one is not satisfying the required QoS. This switching process is known as *vertical handover* and is possible if the mobile terminal could sense its environment and accordingly/cognitively reconfigure its communication parameters to better adapt to the channel conditions. Thus, the operation of such a cognitive system includes two stages: sense and decide. This paper deals only with the sensing task, prior to perform a vertical handover, where the terminal has to evaluate some relevant metrics that are informative about the QoS of the available networks of interest.

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. To satisfy those conditions, the algorithm here proposed, relies on a physical layer sensing and requires no connection to the access point, no synchronization, no signal demodulation and no frame 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 [1], [2] that the usage of the channel bandwidth in a CSMA/CA system can be approximated as 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. Then, once we read a NAV value during a certain time window, the available bandwidth and access delay can be estimated [3]. The main drawback of this method is that it requires 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 Point (AP) are detected. An alternative technique has been presented in [4], [5], [6] for the estimation of the channel occupancy rate. This technique relies on a physical layer sensing, but requires the knowledge of noise variance and does not give any information about the collision rate which is a complementary metric to the channel occupancy rate. Indeed, in [7] it has been shown that the collision rate depends on the number of users connected to an access point, and in [8] it is stated that the mean of the MAC (Media Access Control) delay associated with a transmission by a particular source is increasing exponentially with the probability of collision. Thus, higher is the collision rate, lower is the available QoS. To the best of our knowledge, the only passive technique for collision detection has been studied in [4], [6]. Unfortunately, this method needs

the knowledge of the edges of the frame (the start and end points) in order to apply information theoretic criterion to test if a collision is occurred. As the receiver doesn't know the exact times when the frame starts and ends, this technique can not 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 a relatively addressed infrequently in the extant literature [9], [10], [11], [12]. 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 a $M + 1$ states chain (see Figure 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 one proposed in [13] which has been proposed for spectrum sensing.

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 [14], [15], [16]. 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 this state. 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 (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.

The remaining of the paper is organized as follows. In Section II, we formulate the problem and present the Markov model for tracking the number of source. In Section III, 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 other applications. In Section IV, we explain how the proposed algorithm is used for a WiFi network. In Section V, 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 VI, concludes the paper.

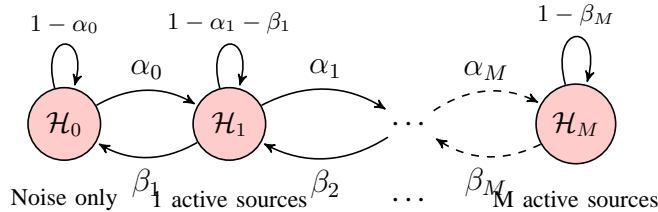


Fig. 1. Markov Chain Model for adaptive source enumeration.

Throughout this paper we denote: $\mathbb{E}[\cdot]$ for mathematical expectation, $(\cdot)^T$ for conjugate transposition, $(\cdot)^H$ for complex conjugate transposition, $\|\cdot\|$ for Frobinius 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 $\text{diag}(\cdot)$ for diagonal matrix with entries given by elements of (\cdot) .

II. MODEL

Let consider a receiver equipped with N antennas, and $\mathbf{x}(k) = [x_1(k), \dots, x_N(k)]^T$ denote the received signal at the time instant k . be a mixture of maximum $K < N$ independent signals as follows

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

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 variance of $\sigma^2(k)$ which is independent with $\mathbf{s}(k)$.

The sources are rising transmitting their packets and vanishing upon the time. As a result, the number of active sources K varies with time k and needs to be estimated k iteratively with time k . 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 arise at any given time instance, we can used the Markov chain in Figure 1, where states/hypothesies are defined as follows

$$\left\{ \begin{array}{ll} \mathcal{H}_0 & : \text{ Only noise is observed,} \\ \mathcal{H}_1 & : \text{ One station is transmitting,} \\ \mathcal{H}_2 & : \text{ Two stations are transmitting,} \\ \vdots & \quad \quad \quad \vdots \\ \mathcal{H}_M & : \text{ } M \text{ stations are transmitting,} \end{array} \right. \quad (2)$$

where M is known and denotes the maximum number of sources that may be active (thus, we have $K \leq M < N$). If no knowledge is available about M , one can set M to $N - 1$. In Figure 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. 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 instance we have multi-hypothesis composite test problem with some a priori estimates for the unknown parameters.

In our model, the Mobile stations signal 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 is shown by

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

Note that it is implicitly assumed that the process $\mathbf{x}(k)$ is non-stationary. The non-stationarity has 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 creates 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

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

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

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 $\lambda_1(k) \geq \lambda_2(k) \geq \dots \geq \lambda_N(k) \geq 0$.

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

$$\mathbf{R}(k) = \mathbf{H}(k) \begin{bmatrix} \zeta_1^2(k) & 0 & \dots & 0 \\ 0 & \zeta_2^2(k) & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & & \zeta_m^2(k) \end{bmatrix} \mathbf{H}^H(k) + \sigma^2(k) \mathbf{I}_N. \quad (6)$$

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 consist 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

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

where the constant $\epsilon \in (0, 1)$ is called the forgetting factor and determines the effective length of the exponential window $\frac{2-\epsilon}{\epsilon}$ [17]. A larger value for $\epsilon < 1$ results in a better tracking capability in an environment where $\mathbf{H}(k)$ and $\{\zeta_i^2(k)\}_{i=1}^K$ vary faster with time (e.g., for larger speeds).

III. PROPOSED ALGORITHM

In our algorithm, for updating EVD, we propose to use the PROTEUS-1 algorithm proposed by B. Champagne *et al.* [18]. There are several other alternative algorithms that could be used. This algorithm is computationally efficient using a CORDIC processor as it uses only plane rotations for updating the eigenvectors and directly provides the set of orthonormal eigenvectors which 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 a four steps soft enumeration algorithm to determine the probability of each hypothesis in (2), recurrssively.

STEP 0: Initialization step: We define $P_{k|k-1}^m$ as the a priori probability of having m active sources at time k

$$P_{k|k-1}^m = P[\mathcal{H}_m \text{ at time } k | O(k-1)] \quad (8)$$

where the hypothesis \mathcal{H}_m denotes the event that m sources are active and $O(k-1)$ denotes the available information and observation up to the instant $k-1$. At the instant $k=0$, in the absence of any knowledge, we initialize the a priori probabilities as $P_{0|-1}^m = \frac{1}{M}, \forall m = 1, \dots, M$. The EVD of the received signal, $\hat{\mathbf{U}}(0)$ and $\hat{\mathbf{\Lambda}}(0)$ are initialized randomly.

STEP 1: Sense and Update: Receive $\mathbf{x}(k)$ and perform the preprocessing and normalization steps in PROTEUS-1 [18] as follows

- 1) *Projection* : $\mathbf{y}(k) = \hat{\mathbf{U}}^H(k-1)\mathbf{x}(k)$;
- 2) *Mapping into real vector elements:*

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

using $\mathbf{D}(k) = \text{diag}(\frac{y_1(k)}{|y_1(k)|}, \dots, \frac{y_n(k)}{|y_n(k)|})$ where $y_i(k)$ denotes the i th entry of the vector $\mathbf{y}(k)$.

- 3) *Reordering/sorting:* Find the permutation matrix $\mathbf{\Pi}$ which reorder the entries of $\mathbf{y}(k)$ the decreasing order and

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

- 4) Update the EVD using as follows [18]:

for $i = 1 : N - 1$ **do**

for $j = i + 1 : N$ **do**

$$\theta \leftarrow y_i(k)y_j(k)/(\hat{\lambda}_i(k) - \hat{\lambda}_j(k))$$

$$\hat{\mathbf{U}}(k) \leftarrow \hat{\mathbf{U}}(k)\mathbf{G}_{ij}(\theta)$$

end for

end for

- 5) $\hat{\lambda}_i(k) \leftarrow (1 - \epsilon)\hat{\lambda}_i(k) + \epsilon y_i^2(k)$

and for under \mathcal{H}_0 as

$$\mathcal{L}(\mathbf{x}(k)|\mathcal{H}_0) = -N \log(\pi \hat{\sigma}_0^2(k)) - \frac{1}{\hat{\sigma}_0^2(k)} \|\mathbf{x}(k)\|^2 \quad (13)$$

where $\mathbf{V}_m(k) = [\mathbf{u}_{m+1}(k) \ \mathbf{u}_{m+2}(k) \ \dots \ \mathbf{u}_N(k)]$. The term $\ln(\frac{\epsilon}{m})$ is an additional penalty function introduced to avoid overestimation. The reason is that the larger values of m involves more unknown parameters which result in a larger bias in the estimation of the Log-Likelihood functions. The function $\ln(\frac{\epsilon}{m})$ is obtained in an empirical manner comforted by simulations and may be justified using the asymptotic distribution of the eigenvalues [17]. 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 $P_{k|k-1}^m$ defined in (8), and the Log-Likelihood functions in (12) and (13), to obtain the *posterior* probabilities defined as

$$P_{k|k}^m = P[\mathcal{H}_m \text{ at time } k | O(k)]. \quad (14)$$

In fact, $P_{k|k}^m$ and $P_{k|k-1}^m$ 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 $\mathbf{P}_{k|k} = [P_{k|k}^0, \dots, P_{k|k}^M]^T$ and $\mathbf{P}_{k|k-1} = [P_{k|k-1}^0, \dots, P_{k|k-1}^M]^T$ denote the vectors containing the $M + 1$ posterior and a priori probabilities, respectively, and

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

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

$$\hat{K} = \arg \max_m \{P_{k|k}^0, \dots, P_{k|k}^M\}. \quad (17)$$

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

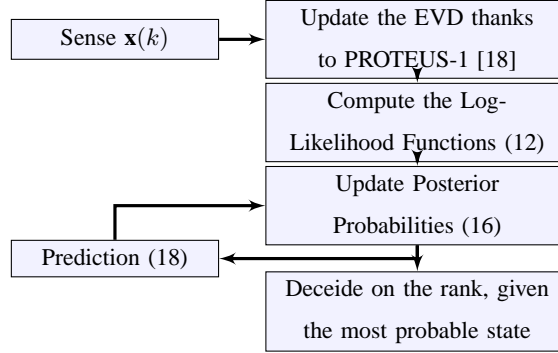


Fig. 2. Flow chart of the proposed algorithm

STEP 4: Prediction of priori probabilities: A prediction of the priori probabilities $P_{k+1|k}^m$ for the next sensing time is needed to compute the posterior probabilities expressed in (16). We use the Markov model illustrated in Figure 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 Figure 1 is given by

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

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

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 $6N^3 + (M + 16.5)N^2 + (M + 9.5 + \nu)N + 7M + 12$.

IV. 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. In addition 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 $1 - P_{k|k}^0$ and $1 - P_{k|k}^0 - P_{k|k}^1$ represent instantaneous estimates of the probabilities that

Algorithm 1 Adaptive Source Enumeration

- 1: Initialize : $\widehat{\mathbf{U}}(0)$, $\widehat{\mathbf{\Lambda}}(0)$ and $\mathbf{P}_{0|-1}$;
 - 2: Sense $\mathbf{x}(k)$
 - 3: **for** $k = 1 : N_s$ **do**
 - 4: $\mathbf{y}(k) = \widehat{\mathbf{U}}^H(k-1)\mathbf{x}(k)$;
 - 5: $\mathbf{D}(k) = \text{diag}(y_i(k)/|y_i(k)|)$;
 - 6: $\mathbf{y}(k) \leftarrow \mathbf{D}^H(k)\mathbf{y}(k)$;
 - 7: $\widehat{\mathbf{U}}(k) \leftarrow \widehat{\mathbf{U}}(k-1)\mathbf{D}(k)$;
 - 8: $\mathbf{y}(k) \leftarrow \mathbf{\Pi}^T \mathbf{y}(k)$;
 - 9: $\widehat{\mathbf{U}}(k) \leftarrow \widehat{\mathbf{U}}(k)\mathbf{\Pi}$;
 - 10: $\widehat{\mathbf{\Lambda}}(k) \leftarrow \mathbf{\Pi}^T \widehat{\mathbf{\Lambda}}(k-1)\mathbf{\Pi}$;
 - 11: **for** $i = 1 : N-1$ **do**
 - 12: **for** $j = i+1 : N$ **do**
 - 13: $\theta \leftarrow y_i(k)y_j(k)/(\widehat{\lambda}_i(k) - \widehat{\lambda}_j(k))$;
 - 14: $\widehat{\mathbf{U}}(k) \leftarrow \widehat{\mathbf{U}}(k)\mathbf{G}_{ij}(\theta)$;
 - 15: **end for**
 - 16: **end for**
 - 17: $\widehat{\mathbf{\Lambda}}(k) \leftarrow (1 - \epsilon)\widehat{\mathbf{\Lambda}}(k) + \epsilon \text{diag}(y_i^2(k))$;
 - 18: **for** $m = 0 : M$ **do**
 - 19: Compute $\sigma^2(k)$ according to (11);
 - 20: Compute $\mathcal{L}(\mathbf{x}(k)|\mathcal{H}_m)$ thanks to (12);
 - 21: Deduce $f(\mathbf{x}(k)|\mathcal{H}_m) = \exp(\mathcal{L}(\mathbf{x}(k)|\mathcal{H}_m))$;
 - 22: **end for**
 - 23: $\mathbf{P}_{k|k} = \frac{1}{\mathbf{F}^T(k)\mathbf{P}_{k|k-1}}\mathbf{F}(k) \odot \mathbf{P}_{k|k-1}$;
 - 24: Decide on the rank according to the most probable hypothesis;
 - 25: $\mathbf{P}_{k+1|k} = \mathbf{T}\mathbf{P}_{k|k}$
 - 26: **end for**
-

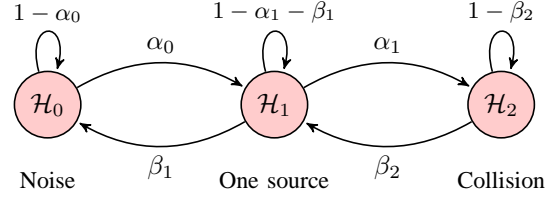


Fig. 3. Markov Model for the PHY layer of a WiFi Channel

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 CSMA/CA. Carrier sense multiple access with collision avoidance (CSMA/CA) is a multiple access method, in computer networking. Any node wishing 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 wait for a random period of time called backoff. Thus, there are Inter Frame Spacing (IFS) time intervals between any two consecutive frames during which the observed signal consist 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 as illustrated in Figure 3 which means $M = 2$.

The three states of the chain are

$$\left\{ \begin{array}{l} \mathcal{H}_0 : \text{ Only noise is observed, (Interframe spacing);} \\ \mathcal{H}_1 : \text{ Only one mobile station is transmitting;} \\ \mathcal{H}_2 : \text{ Two stations are transmitting (Collision).} \end{array} \right. \quad (20)$$

where $1 - \alpha_0 \in (0, 1)$ 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 $N > 2$.

V. SIMULATIONS

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 of $N = 4$ 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 constant during a frame transmission or slowly varying. The forgetting factor used for the PROTEUS-1 algorithm is $\epsilon = 0.05$. The moments of the markov chain illustrated in Figure 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 have been determined empirically by simulations.

The matrices $\hat{\mathbf{U}}(0)$ and $\hat{\mathbf{\Lambda}}(0)$ are initialized by first observing 10 samples on the channel of interest and computing the true EVD of the set of observation. In Figure 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. The top figure represents the posterior probabilities versus time. the decision on the rank (illustrated in the bottom) is made according to the hypothesis presenting the maximum posterior probability. We observe that under these scenario, our algorithm has a good tracking capability. The overestimation that appear 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 $\frac{1}{\epsilon} = \frac{1}{0.05} = 20$.

Note that the delays from the state "two sources" to "one source" and vis versa have no impact on the performance of the algorithm. Indeed, the length of the collision does not matter

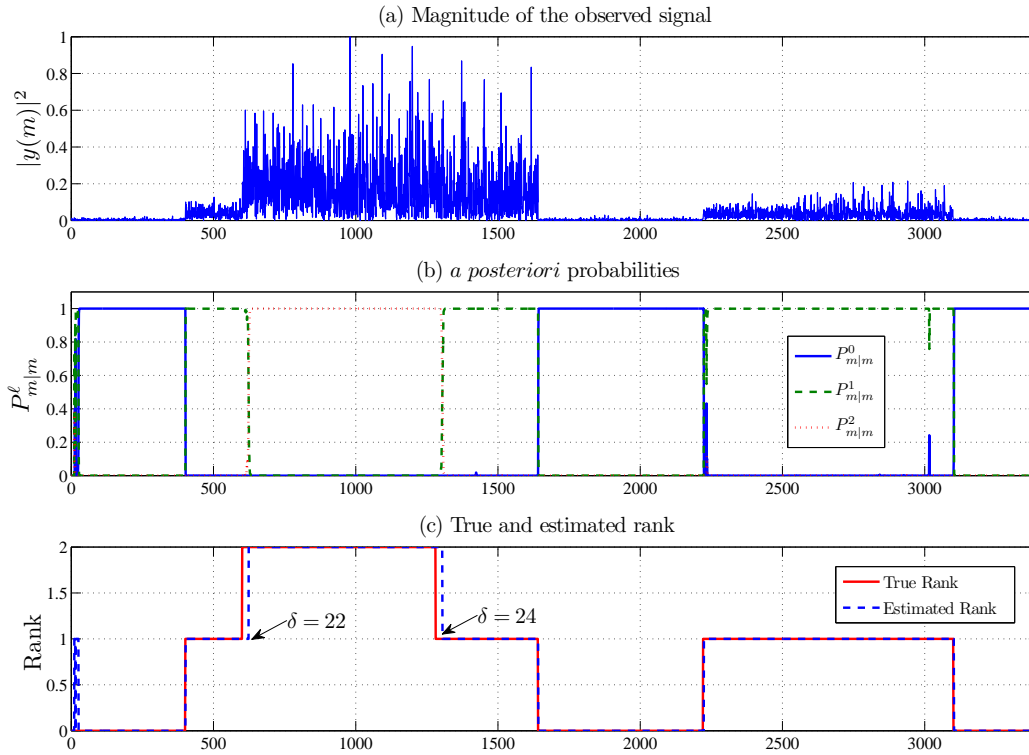


Fig. 4. Example for tracking capacity of the algorithm : (a) Magnitude of the observed signal, (b) a posteriori probabilities and (c) Real and estimated rank.

to us, it is the number of collision that is important in our case. In the other side, 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 Figure 5, we have conducted simulation in the same scenario as in Figure 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 \mathbf{T} .

As stated previously the channel occupancy rate noted 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 Figure 3, the C_{or} is processed

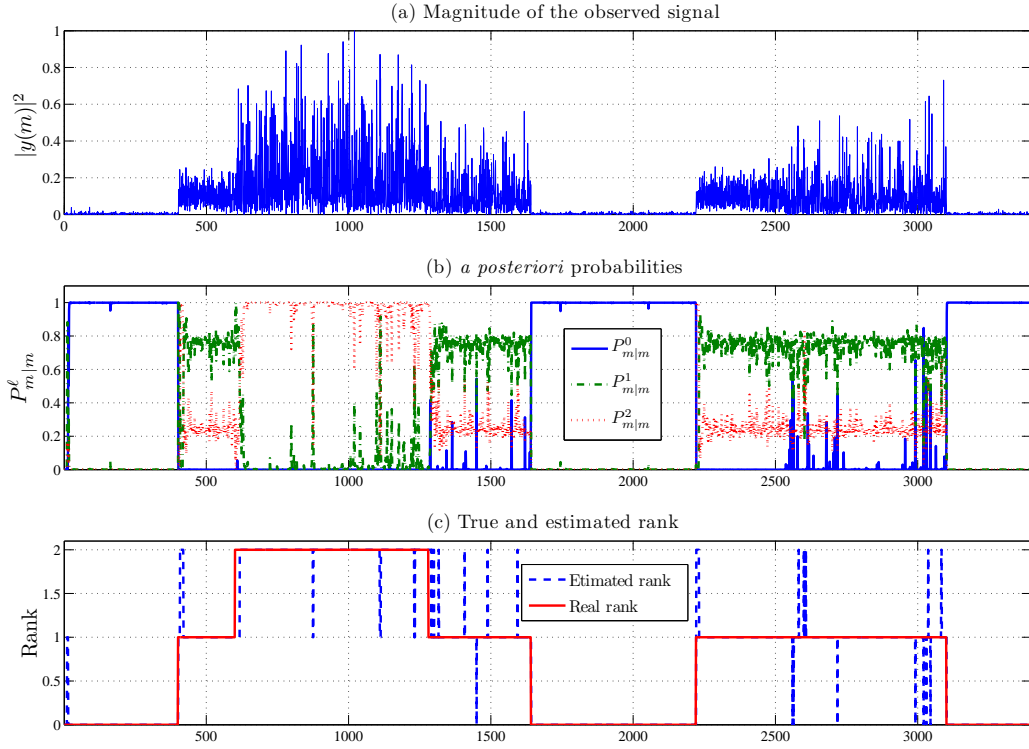


Fig. 5. Example for tracking capacity of the algorithm when changing the transition matrix elements such that α_0 is 100 times greater than in the scenario of Figure 4.

as follows

$$C_{or} = \frac{1}{N_s} \sum_{k=1}^{N_s} (P_{k|k}^1 + P_{k|k}^2) = 1 - \frac{1}{N_s} \sum_{k=1}^{N_s} P_{k|k}^0. \quad (22)$$

where N_s is the length of the observation window.

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

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

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

As the algorithm proposed in [4], 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 [4] which have been tested experimentally on real WiFi signals and shown its efficiency. This smoothing algorithm rely 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 meet a frame of size less that the size of an SIFS, it will automatically affect to it the number of source of the frame that comes after it. Once, this smoothing operated the channel occupancy rate and the collision rate 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 frame with two of them issued from a collision as illustrated in Figure 6.

Figures 7 illustrates the NMSE (Normalized Mean Square Error) of the estimation of the channel occupancy rate defined as

$$\text{NMSE} = 10 \log_{10} \left| \frac{\hat{R} - R}{R} \right|^2 \quad (24)$$

where \hat{R} and R are the estimated and the true channel occupancy rates, respectively. In this figure, we compare the performance of the proposed algorithm to the one proposed in [4] which requires the knowledge of the noise power σ_w^2 . We observe that for Signal to Noise Ratios (SNR) below than 12 dB, the two techniques have similar performance when the Markov approach is out performed by the approach proposed in [4] for higher SNRs. The performance of the Markov approach are very attractive since it achieve NMSE very close to the one in [4] without the knowledge of the noise power. In the same figure, is plotted the performance of the proposed algorithm 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 capabilities of the channel occupancy rate.

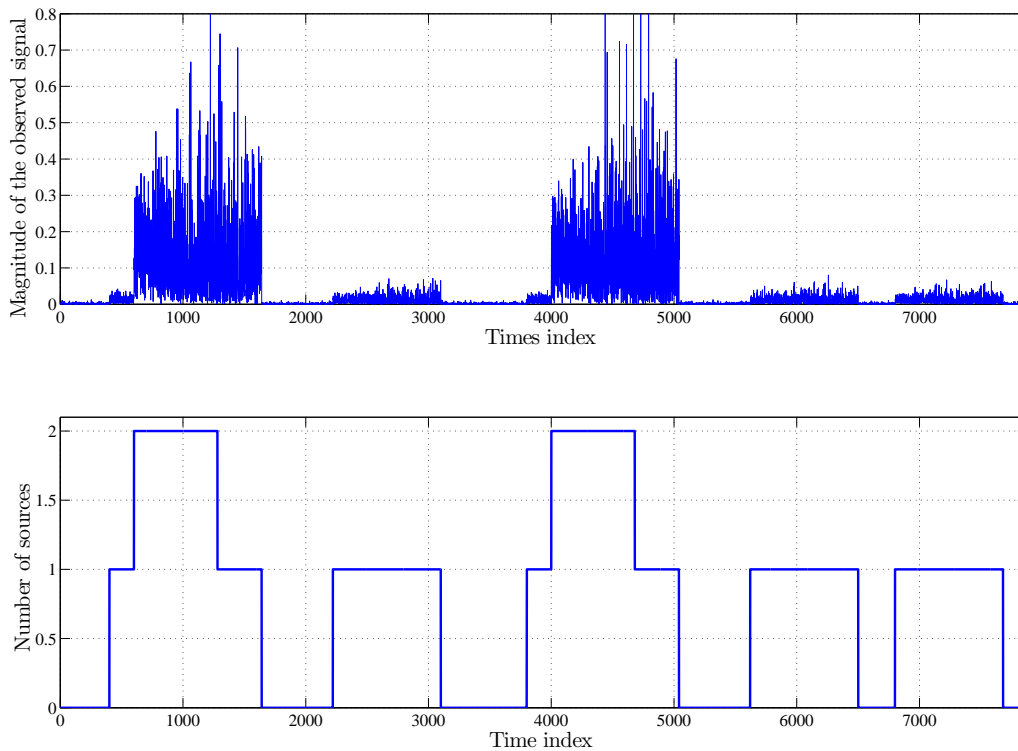


Fig. 6. Used scenario for the evaluation of the algorithm performance.

Figure 8 compares the performance of our algorithm to the ones proposed in [6], [4]. We observe that the proposed approach outperforms both 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, whereas in the other hand for [6], [4] we first need to estimate the number of frames using [5] and then extract the ones suffering from a collision. Thus, two independent sources of error are possible in that case: one in the numerator and one in the denominator. However, conversely 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 in the fact that it does not require any algorithm to detect the frame edges, whereas the approach proposed in [6], [4] 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.

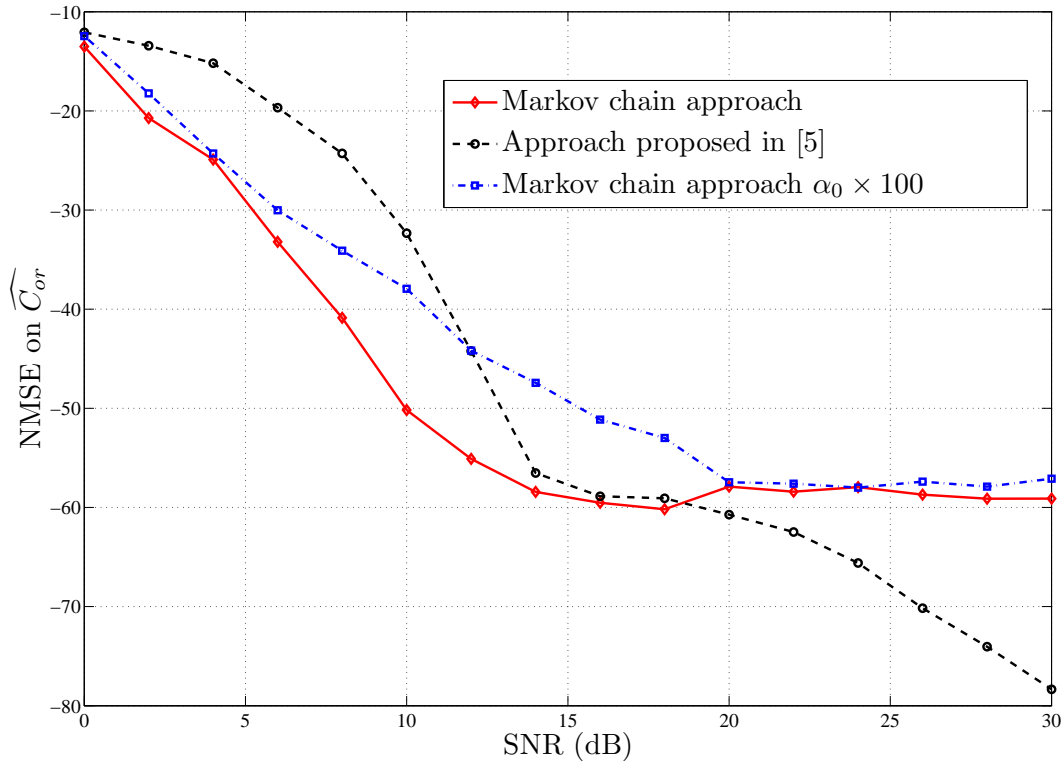


Fig. 7. NMSE on the estimation of the channel occupancy rate.

VI. CONCLUSION

In this paper, we presented a new method for joint estimation of the channel occupancy rate and collision rate of a CSMA/CA based network. The proposed four steps algorithm is based on a Markov chain modeling, and can be extended for any application that requires number of sources tracking. Numerical simulation show that our proposed algorithm well behaves for the practical SNR operating range of wireless networks. Our algorithm is computationally cheap and performs noise variance estimation.

REFERENCES

- [1] Z. Dai, R. Fracchia, J. Gosteau, P. Pellati, and G. Vivier, "Vertical handover criteria and algorithm in IEEE 802.11 and 802.16 hybrid networks," *IEEE International Conference on Communications*, pp. 2480–2484, 2008.
- [2] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, "A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 834–848, 2004.

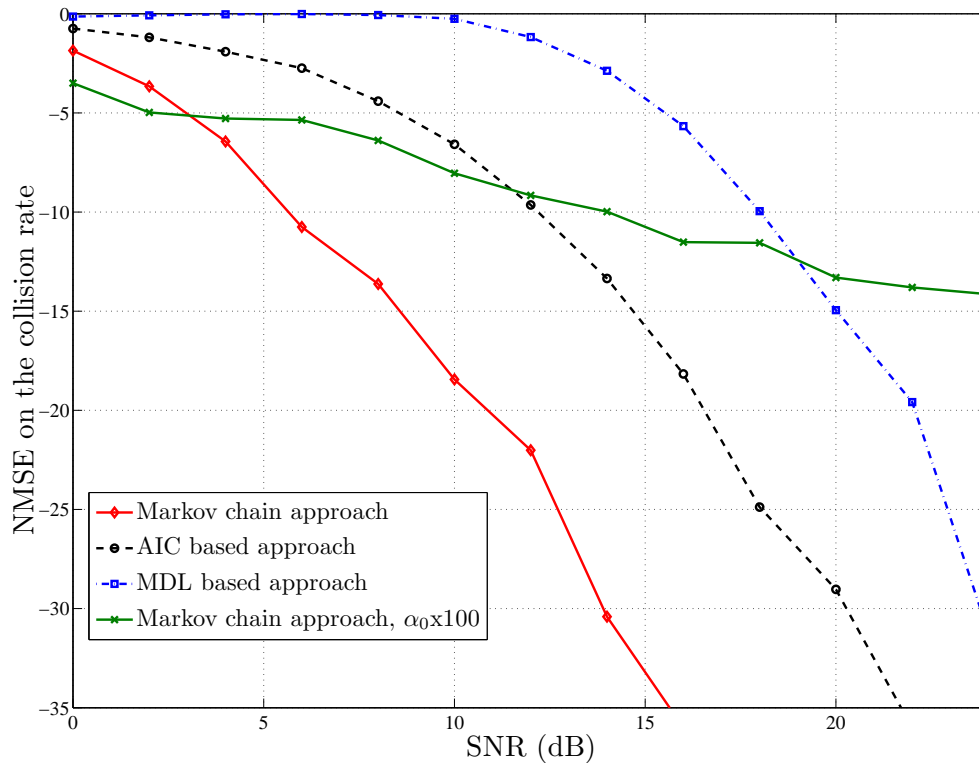


Fig. 8. NMSE on the estimation of the collision rate.

- [3] Q. Zhang, C. Guo, Z. Guo, and W. Zhu, "Efficient Mobility Management for Vertical Handoff between WWAN and WLAN," *Communications Magazine, IEEE*, vol. 41, pp. 102–108, Nov 2003.
- [4] M-R. Oularbi, F-X. Socheleau, S. Houcke, and A. Aissa-El-Bey, "Vertical Handover Metrics Estimation for OFDM Based Systems," *EURASIP Journal On Wireless Communication and Networking*, vol. 2011:93, no. doi:10.1186/1687-1499-2011-93, 2011.
- [5] M-R. Oularbi, S. Houcke, and A. Aissa-El-bey, "Physical layer ieee 802.11 channel occupancy rate estimation," *I/V Communications and Mobile Network (ISVC), 2010 5th International Symposium on*, Sept 2010.
- [6] M-R. Oularbi, S. Houcke, and A. Aissa-El-bey, "Two physical layer metrics for vertical handover towards an ieee 802.11n network," *COgnitive systems with Interactive Sensors, Paris-France*, Nov 2009.
- [7] K. Duffy, D.W. Malone, and D.J. Leith, "Modelling 802.11 wireless links," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, 2005, pp. 6952 – 6957.
- [8] P. Raptis, V. Vitsas, and K. Paparrizos, "Packet delay metrics for ieee 802.11 distributed coordination function," *Mobile Networks and Applications*, vol. 14, pp. 772–781, 2009, 10.1007/s11036-008-0124-7.
- [9] A. Kavcic and B. Yang, "Adaptive rank estimation for spherical subspace trackers," *Signal Processing, IEEE Transactions on*, vol. 44, no. 6, pp. 1573 –1579, June 1996.
- [10] D.J. Rabideau, "Fast, rank adaptive subspace tracking and applications," *Signal Processing, IEEE Transactions on*, vol.

- 44, no. 9, pp. 2229 –2244, Sept. 1996.
- [11] E.C. Real, D.W. Tufts, and J.W. Cooley, “Two algorithms for fast approximate subspace tracking,” *Signal Processing, IEEE Transactions on*, vol. 47, no. 7, pp. 1936 –1945, July 1999.
- [12] P.O. Perry and P.J. Wolfe, “Minimax rank estimation for subspace tracking,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 3, pp. 504 –513, 2010.
- [13] A. Taherpour, S. Gazor, and A. Taherpour, “Adaptive spectrum sensing and learning in cognitive radio networks,” in *18th European Signal Processing Conference (EUSIPCO-2010)*, August 2010, vol. 4, pp. 860–864.
- [14] G. Bianchi, “Performance analysis of the iee 802.11 distributed coordination function,” *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535 –547, Mar. 2000.
- [15] J. McDougall and S. Miller, “Sensitivity of wireless network simulations to a two-state markov model channel approximation,” in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, 2003, vol. 2, pp. 697 – 701 Vol.2.
- [16] T. Vercauteren, A.L. Toledo, and X. Wang, “Online bayesian estimation of hidden markov models with unknown transition matrix and applications to iee 802.11 networks,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, 2005, vol. 4, pp. iv/13 – iv/16 Vol. 4.
- [17] E. Yazdian, S. Gazor, and H. Bastani, “On the spectral distribution of the windowed sample covariance matrix,” *submitted to IEEE Trans. on Signal Processing*.
- [18] B. Champagne and Q.G. Liu, “Plane rotation-based evd updating schemes for efficient subspace tracking,” *Signal Processing, IEEE Transactions on*, vol. 46, no. 7, pp. 1886 –1900, July 1998.
- [19] E. Yazdian, S. Gazor, and H. Bastani, “Eigenvalue estimation of the exponentially windowed sample covariance matrices,” *submitted to IEEE Trans. on Signal Processing*.