# Network Estimation in Cognition-Empowered Wireless Networks

Igor Burago and Marco Levorato

*(Invited Paper)*

*Abstract*—**An approach to parametric identification of the transmission processes of the terminals in a wireless network is proposed, presenting a trade-off between accuracy of capturing the temporal dependencies in observations of transmission processes and the time complexity of the estimation procedure. The maximum likelihood estimator is built for an approximation of the true likelihood function for the observed network activity. A complex network where terminals store packets in a finite buffer and implement a backoff-based random channel access protocol is considered. Minimal information is available for observation to the cognitive terminals, in the form of energy readings mapped to the number of transmitting nodes in each time instant. The entanglement of the transmission processes induced by interference and the filtering effect of packet buffering make this task particularly difficult. It is shown how, based on the estimated parameters, the cognitive terminals, operating in the same channel resource, can predict the transmission trajectories of the other nodes and devise smart transmission strategies controlling the interference generated to the network.**

*Index Terms*—**Wireless networks, Cognitive radio, Parameter estimation, Maximum likelihood estimation.**

## I. Introduction

THE cognitive radio paradigm was introduced in [1], where Mitola proposed the implementation of intelligence in wireless terminals. This concept originated a vast research effort exploring techniques for the coexistence of cognitive (unlicensed) terminals with licensed terminals operating in the same frequency bands [2]–[5]. In these works, the objective of the smart terminals is to identify and exploit transmission channel resource left unused by the licensed terminals along the time and frequency dimensions, *i.e.*, the white space approach. The focus on the avoidance of interference to licensed users led to solutions primarily based on channel sensing, where cognition is used to acquire sufficient information to make transmission decisions with constrained sampling capabilities [4].

The white space approach finds relevant applications in wireless networks, but fails to explore the important opportunities that arise from a wider notion of wireless terminals' intelligence. Increasingly, this approach imposes a rigid classification of the wireless nodes in primary and secondary users, that does not fit a large spectrum of network scenarios and applications. Finally, the white space approach confines the operations of the cognitive terminals within unused channel resource, thus neglecting the potential benefits of controlling the unavoidable interactions occurring between terminals through interference and cooperation [6]. Recent work [7]–[9] extends the cognitive radio framework beyond the white space, and explores techniques accounting for the long-term effect of interference on the state trajectory of the network. However, the learning and optimization capabilities of the cognitive terminals are limited to relatively simple network scenarios by high computational complexity and slow convergence rate.

This paper represents a first attempt to incorporate advanced machine learning and artificial intelligence techniques in the real-time operations of wireless networks to control the interactions between the transmission processes of the terminals. A first important step, then, is the development of a framework for the *analysis* of the dynamics of other nodes operating in the network. This is to enable prediction, classification, and adaptation to the surrounding network environment. Prior work proposing the estimation of wireless network parameters targeted at the white space approach uses a binary idle/busy representation of the network state [10]. Learning-based predictive channel access has been proposed in prior contributions [11], [12]. However, the learning algorithms presented in these works are based on models that abstract the network behavior and result in limited analytical capabilities. Importantly, different from existing contributions in the cognitive network area, we assume that the observed terminals implement protocols whose parameters are determined by application-specific requirements. Furthermore, the models proposed in [11], [12] cannot capture the interdependence between transmission decisions and the statistics of the future state of the network that is critical to control the interactions occurring in the network. The learning algorithm proposed herein incorporates structural information on the complex dynamics of the network in the deconstruction of sequences of simple observations, and does not need any supervised training phase. An approach based on Hidden Markov Model theory, such as that used in [11] over a simplified state space, would result in unfeasible computational complexity and generate an over-parametrization. In this paper, we attempt to develop a method for analyzing the dynamics of wireless networks with feasible complexity. We argue that our approach could enable a further level of intelligence in the cognitive network domain, and foster new coexistence frameworks.

We consider a scenario where the wireless nodes store packets in a finite buffer and a random channel access protocol is used to share the wireless resource. Collision events trigger the initiation of a random backoff period [13]. The nodes can use different backoff parameters to optimize the performance of the supported traffic streams [14]. For this complex scenario, we propose an algorithm for the online learning of the statistics

The authors are with the Department of Computer Science, Donald Bren School of Information and Computer Science, University of California, Irvine, CA 92697, USA (e-mail: {iburago, levorato}@uci.edu).

of the transmission trajectories of the wireless terminals and their complex interactions. Minimal information is assumed available for observation, where the smart node analyzing the behavior of the network can only access energy readings that are mapped to the number of terminals transmitting in any given time slot [15], [16]. The traffic regime and the protocol parameters of the individual nodes are unknown. The identity of the transmitting nodes is assumed not to be observable, and the transmitted packets] are indistinguishable. We remark that the entanglement between the transmission processes of the nodes due to their reaction to packet collision, and the filtering effect of packet buffering make our task particularly difficult.

Numerical results show that the learned parameters provide adequate reconstruction of the signal properties, and that the cognitive terminal can use them to build advanced predictive transmission strategies. However, estimation of the individual-node traffic and protocol parameters explaining the network trajectory observed in one particular sample may be difficult in some parameter regions due to low information capacity of the observed channel-access signal, and temporal dependencies between observations in it. We observe that knowledge of individual-node parameters can enable a further refinement of the cognitive terminals' transmission strategy, for instance, to generate application-specific interference matching the Quality of Service (QoS) requirements of the individual flows.

The rest of this paper is organized as follows. In Section II, the network scenario and logical operations of the wireless nodes are described. In Section III, the stochastic process tracking the dynamics of the transmission processes of the wireless terminals is derived. Section IV presents the algorithm for the estimation of the parameters governing the dynamics of the network from a sequence of observations. Section V proposes an algorithm for the control of the access pattern of a cognitive terminal based on the estimated network parameters. Numerical results are presented in Section VI. Section VII discusses future work and directions. Section VIII concludes the paper.

## II. Network Operations

Consider a communication network consisting of $N$ wireless nodes communicating with each other. Slotted time is assumed, where the duration of a time slot $t=1,2,\dots$ fits the transmission of one information packet. Each node stores packets to be delivered in a finite buffer of size $Q$. The queue is managed with a first-in, first-out discipline, where packets exceeding the buffer capacity are discarded. Packet arrival at the queue of each node $k$ is defined as a general stochastic process controlled by the vector of parameters $\tau^{(k)}$. For definiteness, packets arrived in slot $t$ are processed and available for transmission in that same slot $t$. The parameter vector $\tau^{(k)}$ shapes the packet arrival process to match the *traffic regime* induced by the applications running on the node. In general, traffic regimes of different nodes are considered to be independent of each other.

The transmission strategy of each node is governed by a node-specific channel access protocol. Herein, it is assumed that all nodes implement a random access protocol based on a backoff mechanism (such as that defined in the Distributed Coordination Function [13]). The vector of parameters controlling the backoff mechanism of node $k$ is denoted with $\theta^{(k)}$. Each node can have a different set of parameters, independent with respect to those of other nodes. We observe that the proposed framework can be generalized to nodes implementing structurally different protocols, at the price of a larger parameter space.

According to the considered transmission protocol, decision on whether a node will transmit in any particular time slot is determined by the number of the packets in the transmission queue of that node, and its backoff counter: If the queue is not empty and the backoff counter is zero, the node attempts to transmit the first queued packet. If the transmission attempt is successful, the packet is removed from the node's queue, and the backoff counter maintains its zero value. Otherwise, if the transmission fails, the packet remains in the queue for a subsequent transmission attempt, and the backoff counter is set to a nonzero value $\delta$ according to some probability distribution determined by the parameters $\theta^{(k)}$. For the following $\delta$ time slots, the node will not attempt any transmissions; it will only accept arriving packets, and will decrease the backoff counter by one in every slot. Note that a feedback from the receiver is necessary to determine the decision of the individual nodes. We assume that the receivers of successfully decoded packets transmit a binary feedback to the respective transmitter to report successful or failed decoding.

The widely used collision model is adopted in this work. Thus, transmission in a given slot is successful if exactly one node transmits a packet, whereas transmissions overlapping the same slot result in decoding failure with probability one. More refined models including packet capture effect [17], [18] and stochastic processes capturing the variations in the channel gain of wireless links [19], [20] can be incorporated in the proposed framework.

## III. Network State Dynamics

With respect to the considered protocol, the *state* of a node is fully described by the pair $(i,j)$, where $i\in\{0,\dots,Q\}$ is the queue length and $j\in\{0,\dots,B\}$ is the backoff counter. Note that if the queue is empty, the backoff counter is forced to be zero. Let $p_{i,j}^{(k,t)}$ be the probability of the node $k\in\{1,\dots,N\}$ being in the state $(i,j)$ at the end of slot $t$. As we just mentioned, we exclude from the consideration pairs $(i,j)$ for $i=0$ and $j>0$, as they do not correspond to valid states.

In this section, we will show how these probabilities can be computed recursively for any moment in time. For that purpose, let us temporarily assume that probabilities $p_{i,j}^{(k,t-1)}$ for the preceding time slot are known, and let the initial state of each node $k$ be set to $(0,0)$:

$$p_{0,0}^{(k,0)}=1,\quad p_{i,j}^{(k,0)}=0,\quad \text{for } i,j>0. \tag{1}$$

At the start of each of the successive time slots, according to the modeling assumptions, every node first processes incoming packets. Let $s_{i,j}^{(k,t)}$ be the probability of the node $k$ being in the state $(i,j)$ for a queue of $i$ packets and the backoff counter of $j$, after incoming packets are added to the queue during time slot $t$, but before any packet is transmitted by the node. By construction, for $0 < i < Q$, $0 \le j \le B$, and $t > 0$,

$$s_{0,0}^{(k,t)} = p_{0,0}^{(k,t-1)} \alpha^{(k,t)}(0, \tau), \tag{2}$$

$$s_{i,j}^{(k,t)} = \sum_{x=0}^{\min(A,i)} p_{i-x,j}^{(k,t-1)} \, \alpha^{(k,t)}(x, \tau), \tag{3}$$

$$s_{Q,j}^{(k,t)} = \sum_{x=0}^{\min(A,Q)} p_{Q-x,j}^{(k,t-1)} \sum_{y=x}^{A} \alpha^{(k,t)}(y, \tau), \tag{4}$$

where $\alpha^{(k,t)}(x, \tau) = P(X^{(k,t)} = x \mid \tau^{(k)})$ is the probability of the node $k$ receiving exactly $x$ new packets in slot $t$, conditioned on the vector of its parameters, for $x \in \{0, \dots, A\}$. In general, these probabilities may depend on the current state of the node, but they are assumed to be independent from arrival probabilities on other nodes given the state of the node $k$.

When incoming packets are added to their queues, nodes start to process outgoing packets. From the definition of channel access protocol, it follows that there are two ways for a node to transition into each state, both of which are determined by the node's state after packet arrival.

Indeed, if the backoff counter of a node is zero and its queue is empty, then either *(a)* the node was in this same state after packet arrival, and remained inactive, or *(b)* it had the same zero backoff counter, but with one packet in the queue, and that packet was transmitted successfully.

Alternatively, if the backoff counter is zero and the queue has at least one packet, then either *(a)* the node had the same queue after packet arrival, but with the backoff counter of one, and, consequently, been idle, or *(b)* it had the same zero backoff counter, but with one more packet in the queue, and that packet was transmitted successfully.

Finally, if both the queue length and the backoff counter are nonzero, then the node retained the same queue as after packet arrival, and either *(a)* had the backoff counter of one or greater, and waited, or *(b)* it had the backoff counter of zero, which required the node to transmit, but that transmission failed due to interference.

Thus, the state probabilities for each node can be obtained through the following recurrent relations. Each equality corresponds to one of the three cases above considered in the same order. The first terms on the right sides correspond to the possibilities outlined under *(a)*, the second terms to the ones under *(b)*. For $1 \le i \le Q$, $1 \le j \le B$, and $t > 0$,

$$p_{0,0}^{(k,t)} = s_{0,0}^{(k,t)} + s_{1,0}^{(k,t)}(1 - c^{(t)}), \tag{5}$$

$$p_{i,0}^{(k,t)} = s_{i,1}^{(k,t)} + s_{i+1,0}^{(k,t)} \, (1 - c^{(t)}), \tag{6}$$

$$p_{i,j}^{(k,t)} = s_{i,j+1}^{(k,t)} + s_{i,0}^{(k,t)} \delta^{(k,t)}(j, \theta) \, c^{(t)}, \tag{7}$$

where, formally, $s_{Q+1,0}^{(k,t)} = s_{i,B+1}^{(k,t)} = 0$. Note that, in each of the above transition scenarios, the change in the queue length of a node can be explained either by the node having that longer queue in the previous time slot, and receiving no packets in the current slot, or by having a shorter a queue and receiving a few new packets in this time slot. Both of these possibilities are already taken into account in probabilities $s_{i,j}^{(k,t)}$ above.

Here $c^{(t)}$ denotes a binary variable indicating the event of collision when multiple nodes attempt to transmit simultaneously in time slot $t$. Additionally, $\delta^{(k,t)}(j, \theta) = P(\Delta^{(k,t)} = j \mid \theta^{(k)})$ stands for the probability of the backoff counter being set to $j$ on the node $k$ after the transmission in the time slot $t$ fails, where $\theta^{(k)}$ is the vector of parameters controlling the backoff distribution.

Using the above node state probabilities, we can obtain the probability $r^{(k,t)}$ of the node $k$ transmitting in the time slot $t$. According to the protocol, it happens if and only if that node has a nonempty queue and a zero backoff counter:

$$r^{(k,t)} = \sum_{i=1}^{Q} s_{i,0}^{(k,t)}. \tag{8}$$

All of the probabilities defined above— $s_{i,j}^{(k,t)}$, $p_{i,j}^{(k,t)}$, $r^{(k,t)}$ — are implicit functions of the backoff and traffic parameters, $\theta = (\theta^{(k)})_{k=1}^{N}$ and $\tau = (\tau^{(k)})_{k=1}^{N}$, respectively, where $\theta^{(k)}$ and $\tau^{(k)}$ are parameter vectors for a node $k$. Furthermore, all these probabilities are also conditional on the vector of transmission collisions $c^{(t)}$.

Thus, by following relations (2) through (8), for any given distributions $\alpha^{(k,t)}(\cdot, \tau)$ and $\delta^{(k,t)}(\cdot, \theta)$, we can compute the transmission probabilities $r^{(k,t)}$ as functions of the collision indicators $c^{(t)}$ and the parameters $\theta$, $\tau$, for all time slots $t$ from 1 up to $T$ in $O(TNQB)$ time and $O(NQB)$ memory.

Notice that by conditioning node state probabilities with the collision indicators $c^{(t)}$, we are able to decouple the state trajectory of each terminal $k$ from that of the other terminals. Therefore, given $c^{(t)}$, recurrent computations shown in (2)–(8) can be done independently for each node $k$.

## IV. Network Estimation

In previous section, we showed how, knowing the parameters controlling the random processes on the terminals, it is possible to get some insight into the expected behavior of the network. Specifically, we constructed state probabilities $p_{i,j}^{(k,t)}$ and transmission probabilities $r^{(k,t)}$ as functions of parameter vectors $\theta$ and $\tau$. Having that, it is meaningful to set for the opposite task of reconstructing node parameters $\theta$ and $\tau$ by observing the network's behavior.

In this paper, we concentrate on one particular kind of observational data: the number of terminals (simultaneously) transmitting in each time slot, of which, from now on, we will formally think as realizations of random variables $\eta_t$ (for slots $t = 1, 2, \dots$). The information on the number of transmitting terminals in many cases can be obtained by means of energy samples [15], [16].

In general, when nodes transmit at different power levels, the problem of sensing the number of active transmitters indeed cannot be solved exactly for arbitrary transmission strategies. For it to be feasible, it is necessary to acquire estimations of transmission power levels of the terminals, the difficulty of estimating which is rooted in the little information known a priori by secondary users: the transmitted data and the transmission channels are usually inaccessible.

There is a range of works dedicated to blind signal source detection methods based on the Neyman–Pearson test in Gaussian channels [21], Rayleigh fading channels [22], multiple antenna channels [23], and large dimensional multi-antenna channels [24], but these successive works are designed to answer a binary hypothesis test on the presence or absence of a signal source.

An alternative methods for separating signal sources and accurately estimating the number of those sources have been proposed in [25], [26]. However, those methods were designed for distributed sensors, where it is assumed that terminals cooperatively observe the network (which is a common assumption for cognitive cooperative channel sensing).

We remark that this is a worst-case assumption compared to a network scenario where the cognitive terminal has access to the packet headers of transmitted packets, that contain transmitter IDs and packet sequence numbers [8]. This further layer of information can, possibly, improve the quality of parameter estimation on shorter samples.

For this limited observational information, we face the following problem. Given a finite sample $n_1, n_2, \ldots, n_T$ from the distributions of the random variables $\eta_t$, it is necessary to estimate parameters $\tau^{(k)}$ and $\theta^{(k)}$ of the probability distributions $\alpha^{(k,t)}(\cdot, \tau)$ and $\delta^{(k,t)}(\cdot, \theta)$ controlling, respectively, traffic regimes and backoff strategies on each node $k$ (for particular parametrizations of the functions $\alpha^{(k,t)}$ and $\delta^{(k,t)}$).

That is, we have a problem of *parametric identification* that, due to limited observational information, we approach as a problem of *criterial optimization*. Namely, we would like to find node parameters that best explain the signal $n_1, n_2, \ldots, n_T$ as a sample of $\eta_t$. For that purpose, later in this section we proceed to construct an approximation of the sample probability $P(\eta_1 = n_1, \ldots, \eta_T = n_T \mid \theta, \tau, c)$ measuring the likelihood of observing $n_1, n_2, \ldots, n_T$ for any given parameters $\theta$, $\tau$, and a sequence of collision events $c^{(t)}$. That likelihood estimation, then, can be optimized with respect to the unknown parameters.

With such setting in mind, it is important to understand that our capabilities for parametric identification in this problem are limited by the nature of the observed signal. The quality of any given vector of parameter values can only be measured to the extent in which those values affect the number of transmitting nodes, as measured in the limited number of observations available to the estimator.

In practice, it means that, when solving this problem, we may encounter the following difficulty that is intrinsic to its setting. For a given sample of observations, likelihood approximation may have multiple local extrema, which do not necessarily coincide with the true parameters standing behind the observed behavior. Moreover, if those local extrema correspond to sample probabilities that are close (or equal), it implies that there are multiple values of parameters that are similar (or identical) in their explanation power for the sample in question.

Such phenomena is present when the behaviors of one or more nodes are underrepresented in the provided sample: As multiple sequence of event may lead to identical observations, there may not be enough information regarding parameters of separate nodes available in the sample. In order to determine whether the length of the sample should be increased, one can use the parameters resulting from maximizing the likelihood of the current sample to generate predictions for the observations in this sample, and measure the accuracy of them. If the prediction accuracy is unsatisfactory, it is reasonable to obtain more data and prolong the sample.

Further layers of information being available for observation, such that contained in packet headers, can possibly reduce the necessary sample size. The information-theoretic study of the minimal information necessary for exact estimation is left to future investigations.

### A. Parameter Optimization Problem

Having the transmission probabilities (8), we can find the probability distribution of random variables $\eta_t$ representing the number of nodes transmitting in the time slot $t$:

$$P(\eta_t = n \mid \theta, \tau, c) = \sum_{K \in C_n} \pi_K \, \overline{\pi}_{C_n \setminus K}, \tag{9}$$

where $C_n$ is the set of all $n$-element subsets of the set $\{1, \ldots, N\}$ of the node ordinal numbers, and

$$\pi_K = \prod_{k \in K} r^{(k,t)}, \qquad \overline{\pi}_K = \prod_{k \in K} (1 - r^{(k,t)}). \tag{10}$$

Collision indicators $c^{(t)}$ that are necessary for computing $r^{(k,t)}$ are trivial functions of observations $n_t$:

$$c^{(t)} = \mathbf{1}[n_t > 1]. \tag{11}$$

With the probability (9) obtained, it becomes possible to approximate the likelihood of a series of observations $n_1, \ldots, n_T$ with the quasi-likelihood function

$$L(n_{1\ldots T} \mid \theta, \tau, c) = \prod_{t=1}^{T} P(\eta_t = n_t \mid \theta, \tau, c), \tag{12}$$

which coincides with the true likelihood $P(\eta_1 = n_1, \ldots, \eta_T = n_T \mid \theta, \tau, c)$ under the simplifying assumption of observations being independent of each other.

Although the function $L$ does not take into account temporal dependencies between observations and, therefore, diverges from the true likelihood, we prefer it to be this way: If we were to compute the true likelihood instead, we would have to pay a significantly higher computational cost.

Indeed, we could consider a Markov model with the state space of the whole network, which consists of all possible combinations of states of separate nodes. Since transitional probabilities for such Markov model can be easily computed, we could then use the forward-backward algorithm on it to compute the true likelihood for a given sample. The time complexity of the forward-backward algorithm is $O(TS^2)$, where $S = (QB)^N$ is the number of network states, so it would take us at least $O(T(QB)^{2N})$ time to compute the true likelihood this way.

However, if we compute the quasi-likelihood (12) instead, the time complexity would be $O(TQBN + TN\binom{N}{N/2})$, where the first term is responsible for the recurrent computations (2)–(8) necessary to obtain transmission probabilities $r^{(k,t)}$, and the second term is responsible for the computation of the function $L$ itself, as shown in (9)–(12). Since $\binom{N}{N/2} = O(N^{-\frac{1}{2}} 2^N)$, the final time complexity is $O(T(QBN + N^{\frac{1}{2}} 2^N))$.

Comparing the time complexities for these two functions, we can quickly see that for most of the practical values of $Q$,

$B$, $N$, and $T$, the latter algorithm is much more preferable. Thus, here we deliberately choose to sacrifice accuracy of the true likelihood for the lower computational cost of $L$.

It is quite natural now to require parameters $\theta, \tau$ to maximize this likelihood function, and state the optimization problem

$$L(n_{1\ldots T} \mid \theta, \tau, c) \to \max_{\theta,\tau}. \tag{13}$$

Note that, when finding the most likely explanation for the sample in this way, we do not map observations to states of the whole network. Instead, we work with expected states of individual nodes, as predicted by the probabilities $p_{i,j}^{(k,t)}$. This allows us to avoid the cost associated with other approaches working in the space of network states, the size of which is growing exponentially with the number of nodes.

As stated now, problem (13) can already be approached with derivative-free optimization methods. However, in this case it is also possible to recursively compute the gradient of the likelihood function (12) in the space of parameters $\theta$, $\tau$. Indeed, from (9) and (10),

$$\frac{\partial}{\partial\theta} P(\eta_t = n \mid \theta, \tau, c) = \sum_{K \in C_n} \left[ \overline{\pi}_{C_n \setminus K} \frac{\partial \pi_K}{\partial\theta} + \pi_K \frac{\partial \overline{\pi}_{C_n \setminus K}}{\partial\theta} \right], \tag{14}$$

and

$$\frac{\partial \pi_K}{\partial\theta} = \sum_{k \in K} \left[ \frac{\partial r^{(k,t)}}{\partial\theta} \prod_{l \in K \setminus \{k\}} r^{(l,t)} \right], \tag{15}$$

$$\frac{\partial \overline{\pi}_K}{\partial\theta} = -\sum_{k \in K} \left[ \frac{\partial r^{(k,t)}}{\partial\theta} \prod_{l \in K \setminus \{k\}} (1 - r^{(l,t)}) \right]. \tag{16}$$

Clearly, the derivatives with respect to $\tau$ have the same form.

Differentiating the right sides of the recurrent relations (2)–(8), we obtain that, for $1 \le i \le Q$, $1 \le j \le B$, and $t > 0$,

$$\frac{\partial p_{0,0}^{(k,t)}}{\partial\theta} = \frac{\partial s_{0,0}^{(k,t)}}{\partial\theta} + (1 - c^{(t)}) \frac{\partial s_{1,0}^{(k,t)}}{\partial\theta}, \tag{17}$$

$$\frac{\partial p_{i,0}^{(k,t)}}{\partial\theta} = \frac{\partial s_{i,1}^{(k,t)}}{\partial\theta} + (1 - c^{(t)}) \frac{\partial s_{i+1,0}^{(k,t)}}{\partial\theta}, \tag{18}$$

$$\frac{\partial p_{i,j}^{(k,t)}}{\partial\theta} = \frac{\partial s_{i,j+1}^{(k,t)}}{\partial\theta} + c^{(t)}\, \delta^{(k,t)}(j,\theta)\, \frac{\partial s_{i,0}^{(k,t)}}{\partial\theta} + c^{(t)}\, s_{i,0}^{(k,t)} \frac{\partial}{\partial\theta} \delta^{(k,t)}(j,\theta), \tag{19}$$

$$\frac{\partial p_{i,j}^{(k,t)}}{\partial\tau} = \frac{\partial s_{i,j+1}^{(k,t)}}{\partial\tau} + c^{(t)}\, \delta^{(k,t)}(j,\theta)\, \frac{\partial s_{i,0}^{(k,t)}}{\partial\tau}. \tag{20}$$

In the same manner, for $0 < i < Q$, $0 \le j \le B$,

$$\frac{\partial s_{0,0}^{(k,t)}}{\partial\theta} = \alpha^{(k,t)}(0,\tau)\, \frac{\partial p_{0,0}^{(k,t-1)}}{\partial\theta}, \tag{21}$$

$$\frac{\partial s_{i,j}^{(k,t)}}{\partial\theta} = \sum_{x=0}^{\min(A,i)} \alpha^{(k,t)}(x,\tau)\, \frac{\partial p_{i-x,j}^{(k,t-1)}}{\partial\theta}, \tag{22}$$

$$\frac{\partial s_{Q,j}^{(k,t)}}{\partial\theta} = \sum_{x=0}^{\min(A,Q)} \frac{\partial p_{Q-x,j}^{(k,t-1)}}{\partial\theta} \sum_{y=x}^{A} \alpha^{(k,t)}(y,\tau), \tag{23}$$

and

$$\frac{\partial s_{0,0}^{(k,t)}}{\partial\tau} = p_{0,0}^{(k,t-1)}\, \frac{\partial}{\partial\tau} \alpha^{(k,t)}(0,\tau) + \alpha^{(k,t)}(0,\tau)\, \frac{\partial p_{0,0}^{(k,t-1)}}{\partial\tau}, \tag{24}$$

$$\frac{\partial s_{i,j}^{(k,t)}}{\partial\tau} = \sum_{x=0}^{\min(A,i)} \left[ p_{i-x,j}^{(k,t-1)}\, \frac{\partial}{\partial\tau} \alpha^{(k,t)}(x,\tau) + \alpha^{(k,t)}(x,\tau)\, \frac{\partial p_{i-x,j}^{(k,t-1)}}{\partial\tau} \right], \tag{25}$$

$$\frac{\partial s_{Q,j}^{(k,t)}}{\partial\tau} = \sum_{x=0}^{\min(A,Q)} \left[ p_{Q-x,j}^{(k,t-1)} \sum_{y=x}^{A} \frac{\partial}{\partial\tau} \alpha^{(k,t)}(y,\tau) + \frac{\partial p_{Q-x,j}^{(k,t-1)}}{\partial\tau} \sum_{y=x}^{A} \alpha^{(k,t)}(y,\tau) \right]. \tag{26}$$

Finally,

$$\frac{\partial r^{(k,t)}}{\partial\theta} = \sum_{i=1}^{Q} \frac{\partial s_{i,0}^{(k,t)}}{\partial\theta}. \tag{27}$$

In the equations (17)–(27) above, vector $\theta$ stands for the parameters $\theta^{(l)}$ of the backoff distributions of any node $l$. Similarly, $\tau$ stands for any $\tau^{(l)}$. Where the derivatives with respect to $\tau$ are not shown, they have the same form as the corresponding derivatives for $\theta$. Trivially, initial derivatives of state probabilities are zero for all nodes, states, and parameters:

$$\frac{\partial p_{i,j}^{(k,0)}}{\partial\theta} = \frac{\partial p_{i,j}^{(k,0)}}{\partial\tau} = 0. \tag{28}$$

### B. Optimization Procedure

For any given sample $n_1, \ldots, n_T$, two alternative strategies can be used to search for the maximum-likelihood parameters optimizing the quasi-likelihood function (12). Firstly, we can directly apply a general-purpose optimization algorithm to $L(n_{1\ldots T} \mid \theta, \tau, c)$ as a function of $\theta$ and $\tau$. In such case, when we compute this function at some point $(\theta, \tau)$ in the parameter space, we start at the initial state distributions (1) and thread our calculations through the recurrence relations (2)–(8) for each of the $T$ successive time slots. With the parameters being fixed to particular values $\theta$ and $\tau$, the distributions $\alpha^{(k,t)}(\cdot,\tau)$ and $\delta^{(k,t)}(\cdot,\theta)$ become known and do not change as we move from one slot to the next.

At that, both derivative-free optimization techniques and gradient-based optimization methods are applicable. In the former case, it is sufficient to have values of the likelihood function (12). In the latter case, in addition to the likelihood itself, we also use its gradient (14) which, similarly to the probabilities (2)–(8), is computed recursively for all time slots up to slot $T$ according to (17)–(27).

Thus, if an iterative method is used for optimization (*e.g.*, the BFGS method [27], [28] or the Nelder-Mead algorithm [29]), we have the following procedure for parameter estimation that is run for a given sample of observations $n_1, \ldots, n_T$.

1) Set the parameters to their initial values $\theta_0$ and $\tau_0$.

2) With the current estimations $\theta_{m-1}$ and $\tau_{m-1}$ of the parameters obtained on the previous, $(m-1)$st iteration, compute the transmission probabilities $r^{(k,t)}$ for all nodes $k$ and all time slots from 1 to $T$, according to recurrent equations (2)–(8).
3) If the chosen optimization algorithm is gradient-based, use $\theta_{m-1}$ and $\tau_{m-1}$ to also compute the partial derivatives of the transmission probabilities $\partial r^{(k,t)}/\partial\theta^{(l)}$ and $\partial r^{(k,t)}/\partial\tau^{(l)}$ as shown in recurrent equations (17)–(27).
4) Compute the value of the likelihood function $L(n_{1\dots T} \mid \theta_{m-1}, \tau_{m-1}, c)$ and, if required, its gradient, in accordance with (9)–(12) and (14)–(27).
5) Update the current estimation of parameters $\theta_{m-1}$ and $\tau_{m-1}$ to obtain the next-iteration estimation $\theta_m$ and $\tau_m$, using the computed quasi-likelihood and, if necessary, its gradient.
6) Stop, if the parameters have converged, or the number of iterations allocated for the procedure has been exhausted. Otherwise go to step 2.

Alternatively, problem (13) can be rewritten as

$$\log L(n_{1\dots T} \mid \theta, \tau, c) = \sum_{t=1}^{T} g_t(n_t \mid \theta, \tau, c) \to \max_{\theta,\tau}, \quad (29)$$

where

$$g_t(n \mid \theta, \tau, c) = \log P(\eta_t = n \mid \theta, \tau, c). \quad (30)$$

Then, for problem (29), we can apply the method of stochastic optimization [30], [31] to obtain the iterative algorithm:

$$\theta_m^{(k)} = \theta_{m-1}^{(k)} + \gamma_m \frac{\partial}{\partial\theta^{(k)}} g_{t(m)}(n_{t(m)} \mid \theta_{m-1}, \tau_{m-1}, c), \quad (31)$$

$$\tau_m^{(k)} = \tau_{m-1}^{(k)} + \gamma_m \frac{\partial}{\partial\tau^{(k)}} g_{t(m)}(n_{t(m)} \mid \theta_{m-1}, \tau_{m-1}, c), \quad (32)$$

where $m$ is the number of current iteration, $t(m)$ runs through a sample of observations (potentially repeatedly), and $\gamma_m > 0$ is a decreasing sequence satisfying the properties

$$\sum_{m=1}^{\infty} \gamma_m = \infty, \qquad \sum_{m=1}^{\infty} \gamma_m^2 < \infty. \quad (33)$$

Unlike the first strategy of direct optimization of $L(n_{1\dots T} \mid \theta, \tau, c)$, in the stochastic version of the optimization process, we do not compute the whole sum from (29) (or, equivalently, the product from (13)). Moreover, we can avoid the necessity to evaluate the state and transmission probabilities along all the time slots 1 through $t(m)$ for current parameters $\theta_{m-1}$, $\tau_{m-1}$, although at a price of some accuracy loss. Since each correction step (31) and (32) is usually small, it is likely that parameters $\theta_{m-1}$, $\tau_{m-1}$ and $\theta_m$, $\tau_m$ are close. On that ground, the probabilities $p_{i,j}^{(k,t(m))}(\theta_{m-1}, \tau_{m-1})$ can be approximated through a single application of recurrence relations (2)–(8) to the probabilities $p_{i,j}^{(k,t(m)-1)}(\theta_{m-2}, \tau_{m-2})$ that can be simply recalled from the previous, $(m-1)$st step of the algorithm, as opposed to computing their exact values $p_{i,j}^{(k,t(m)-1)}(\theta_{m-1}, \tau_{m-1})$.

Thus, we define the following stochastic-gradient procedure.

1) Set the parameters to their initial values $\theta_0$ and $\tau_0$.
2) With the current estimation $\theta_{m-1}$ and $\tau_{m-1}$ of the parameters obtained on the previous, $(m-1)$st iteration, compute probabilities $s_{i,j}^{(k,t(m))}$, $p_{i,j}^{(k,t(m))}$, and $r^{(k,t(m))}$ by applying (2)–(8) to the probabilities $p_{i,j}^{(k,t(m)-1)}$ obtained on the previous iteration, or to $p_{i,j}^{(k,0)}$ if $t(m) = 1$, otherwise.
3) Similarly, compute gradients of the above probabilities, as shown in (17)–(27) using the gradients of $p_{i,j}^{(k,t(m)-1)}$ and parameters $\theta_{m-1}$, $\tau_{m-1}$ from the previous iteration.
4) Compute the summand $g_{t(m)}(n_{t(m)} \mid \theta_{m-1}, \tau_{m-1}, c)$ of the logarithm of the quasi-likelihood function by following (30), (9), and (10).
5) Compute the gradient of $g_{t(m)}(n_{t(m)} \mid \theta_{m-1}, \tau_{m-1}, c)$ in accordance with (30) and and its gradient, as it is shown in (14)–(27).
6) Obtain the new estimation of parameters $\theta_m$ and $\tau_m$ by applying (31) and (32) to $\theta_{m-1}$ and $\tau_{m-1}$, respectively.
7) Stop, if the parameters have converged, or the number of iterations allocated for the procedure has been exhausted. Otherwise go to step 2.

When the cost of computing probabilities (2)–(8) for all $T$ time slots on every iteration is acceptable, the previous, non-stochastic version of the procedure is preferable. The stochastic-gradient procedure has a potential advantage of lower computational costs when the convergence is fast enough. It also is more suitable for settings that require updating parameter estimations online, with training data not bounded by a certain number of time slots.

However, due to recurrent nature of transmission probabilities $r^{(k,t)}$ and the form of the likelihood function $L$, it is problematic to obtain bounds on the speed of convergence, which deprives us of the guarantee on whether it is indeed faster in general. This uncertainty is only compounded by the assumption that probabilities $p_{i,j}^{(k,t(m))}(\theta_{m-1}, \tau_{m-1})$ can be approximated with $p_{i,j}^{(k,t(m)-1)}(\theta_{m-2}, \tau_{m-2})$, which, as we mentioned above, would be desirable to make additional computational gains.

The choice between two procedures should be made separately for different parametrizations, based on empirical evaluation of convergence rates.

## V. Transmission Policy Adaptation

To illustrate applicability of the proposed network-estimation framework, we formulate a coexistence problem where a cognitive, adaptive terminal shapes its channel access strategy to avoid collisions with other nodes operating in the network. Compared to prior work, the framework presented in this paper empowers the cognitive terminal with the ability to learn the parameters governing the dynamics of the network, predict of the future network state, and possibly control the interactions triggered by interference in the complex network environment described in Section II. We remark that this scenario only represents an example of the wide spectrum of applications enabled by our network analysis framework.

To enable prediction and control, we need to build the conditional probability of the future channel access trajectory as a function of prior observations and actions of the cognitive terminal. We include, then, in the model a Boolean random variable $y_t$ reflecting the decision of the cognitive node to transmit ($y_t = 1$) or remain silent ($y_t = 0$) in slot $t$. This variable influences the statistics of the future network state and, thus, of the number of transmitting nodes $\eta_t$. Importantly, in order to compute this probability, we will need to map the observation history to the state distribution, at the price of a larger computational complexity compared to our low-complexity estimation algorithm, where we only operated in the observation space.

In accordance with the premise of the adaptive transmission strategy that is constructed later in this section, we assume that an adaptive decision $y_t$ depends only on the number $\eta_{t-1}$ of the non-adaptive nodes transmitting in the preceding time slot, and all of the preceding decisions $y_1, \dots, y_{t-1}$ of the cognitive node. That is, given $\eta_{t-1}$ and $y_1, \dots, y_{t-1}$, random variable $y_t$ is independent of the state of the network or any of its characteristics in the time slots preceding the $(t-1)$st slot.

Additionally, let $z_t$ be the random vector describing the state of all of the non-adaptive nodes in the network right before their transmission in the time slot $t$ (but after the arrival of new packets in that slot). For the considered network, $z_t$ consists of queue lengths and backoff counters for all $N$ of those nodes. For brevity, let $y_1^t$ and $z_1^t$ stand for the lists of variables $y_1, \dots, y_t$ and $z_1, \dots, z_t$, respectively.

From the definition of $z_t$ and the assumed transmission protocol, it immediately follows that the number $\eta_t$ of non-adaptive nodes transmitting in slot $t$ is a deterministic function of $z_t$, and is independent of $y_1^t$ and $z_1^{t-1}$. Hence, probabilities $P(\eta_t \mid z_1^t) = P(\eta_t \mid z_t)$ are simply equal to either 1, when the value of $\eta_t$ agrees with the state $z_t$, or 0 otherwise.

It is easy to see that $P(\eta_{t+1} \,|\, z_1^t, \eta_t, y_1^t) = P(\eta_{t+1} \,|\, z_t, y_1^t)$. Indeed, because of the determinism of $\eta_{t+1}$, the probability on the left is equivalent to the probability of transitioning from state $z_t$ to a state $z_{t+1}$ having $\eta_{t+1}$ non-adaptive nodes transmitting. The uncertainty of transitioning from $z_t$ to $z_{t+1}$ is due to one transmission phase (in slot $t$) and one phase of packet arrivals (in slot $t+1$) that separate the two states. Network state after the transmission phase is predetermined by $z_t$ (fixing behavior of the non-adaptive nodes) and $y_t$ (fixing behavior of the cognitive node). Network state after the arrival phase depends only on the traffic regime parameters $\tau$ that were estimated in the learning phase.

With these observations in mind, let us move on to consider conditional probabilities $P(\eta_{t+1} \mid \eta_t, y_1^t)$. By the law of total probability and definition of the conditional probability,

$$\begin{aligned} &P(\eta_{t+1} \mid \eta_t, y_1^t) \\ &\quad = \sum_{z_1^t \in Z^t} P(\eta_{t+1} \mid z_1^t, \eta_t, y_1^t)\, P(z_1^t \mid \eta_t, y_1^t) \\ &\quad = \sum_{z_1^t \in Z^t} P(\eta_{t+1} \mid z_1^t, \eta_t, y_1^t)\, \frac{P(\eta_t \mid z_1^t, y_1^t)\, P(z_1^t, y_1^t)}{P(\eta_t, y_1^t)}, \end{aligned} \tag{34}$$

where each of the variables $z_1, \dots, z_t$ in the sum runs through the set $Z$ of all possible states of the network. Then, from what we have concluded earlier,

$$\begin{aligned} &P(\eta_{t+1} \mid \eta_t, y_1^t) \\ &\quad = \frac{1}{P(\eta_t, y_1^t)} \sum_{z_1^t \in Z} P(\eta_{t+1} \mid z_t, y_t)\, P(\eta_t \mid z_t)\, P(z_1^t, y_1^t) \\ &\quad \propto \sum_{z_t \in Z_t(\eta_t)} \left[ P(\eta_{t+1} \mid z_t, y_t) \sum_{z_1^{t-1} \in Z^{t-1}} P(z_1^t, y_1^t) \right], \end{aligned} \tag{35}$$

for $Z_t(n)$ denoting the set of all possible network states $z_t$ that necessitate $\eta_t(z_t) = n$.

The conditional probability in sum (35) can be expanded to

$$\begin{aligned} &P(\eta_{t+1} \mid z_t, y_t) \\ &\quad = \sum_{z_{t+1} \in Z} P(\eta_{t+1} \mid z_{t+1}, z_t, y_t)\, P(z_{t+1} \mid z_t, y_t) \\ &\quad = \sum_{z_{t+1} \in Z_{t+1}(\eta_{t+1})} P(z_{t+1} \mid z_t, y_t). \end{aligned} \tag{36}$$

The inner sum in (35), taken over different state trajectories $z_1^{t-1}$, can be computed recursively as follows.

$$\begin{aligned} &Q_t(z_t, y_1^t) \\ &\quad = \sum_{z_1^{t-1} \in Z^{t-1}} P(z_1^t, y_1^t) \\ &\quad = \sum_{z_1^{t-1} \in Z^{t-1}} P(z_t, y_t \mid z_1^{t-1}, y_1^{t-1})\, P(z_1^{t-1}, y_1^{t-1}) \\ &\quad = \sum_{z_{t-1} \in Z} \left[ P(z_t, y_t \mid z_{t-1}, y_{t-1}) \sum_{z_1^{t-2} \in Z^{t-2}} P(z_1^{t-1}, y_1^{t-1}) \right] \\ &\quad = \sum_{z_{t-1} \in Z} P(z_t, y_t \mid z_{t-1}, y_{t-1})\, Q_{t-1}(z_{t-1}, y_1^{t-1}), \end{aligned} \tag{37}$$

where $t > 0$, and $Q_0(z_0) = P(z_0)$ is the probability of the network being in the state $z_0$ initially, before the cognitive node has become active. Given the state $z_{t-1}$ and adaptive decision $y_{t-1}$ in the preceding time slot, the current state $z_t$ and decision $y_t$ become independent of each other:

$$P(z_t, y_t \,|\, z_{t-1}, y_{t-1}) = P(y_t \,|\, z_{t-1}, y_{t-1})\, P(z_t \,|\, z_{t-1}, y_{t-1}), \tag{38}$$

where the first probability on the right side of the equation simplifies even more if the decision rule of the cognitive node is deterministic. In any case, both of the probabilities in (38) can be computed immediately from the definition of the assumed transmission protocol used by the non-adaptive nodes.

Now we can use conditional probabilities (35) to construct an adaptive transmission strategy. For simplicity, let us assume that the cognitive node always have a packet for transmission in every time slot (*i.e.*, its queue is never empty), and that the goal of the node is to maximize its throughput, but attempt to minimize, to some extent, its negative effect on other nodes in terms of number of collisions. For this end, we can set the decision rule for the cognitive node to be

$$y_t = \mathbf{1}\big[P(\eta_t = 0 \mid \eta_{t-1}, y_1^{t-1}) > \varepsilon_0\big], \tag{39}$$

where threshold $\varepsilon_0$ is the parameter of the strategy, and $\mathbf{1}[\cdot]$ is the indicator function. Conforming to our assumptions used in the derivation of (35), this decision function is deterministic, given the knowledge of previous decisions $y_1^{t-1}$, and the state of the network $z_{t-1}$ uniquely identifying the number of transmissions $\eta_{t-1}$.

Direct implementation of this method is quite costly computationally due to the necessity of maintaining probabilities

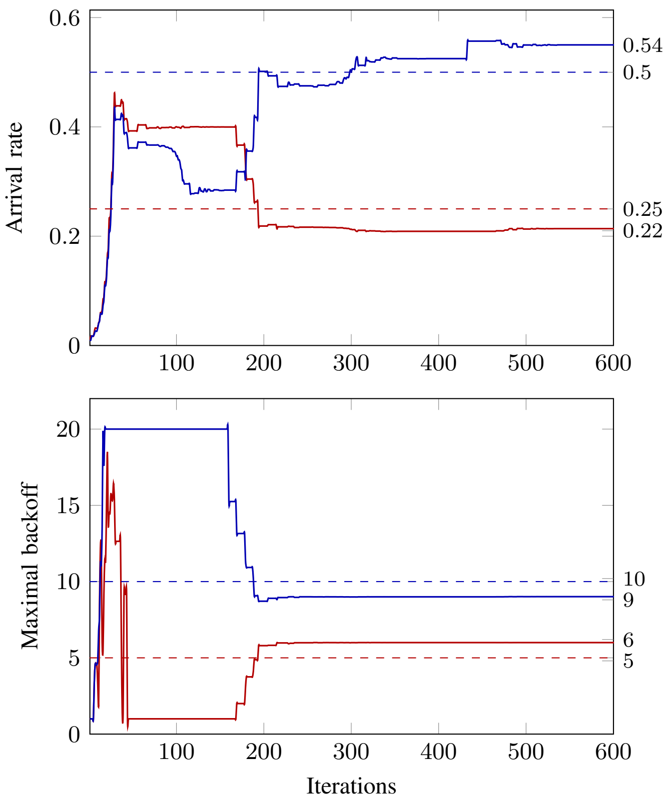


Figure 1. Estimations of the arrival rate and maximal backoff parameters during likelihood optimization for the two-node example problem with true arrival rates of 0.25, 0.5, and maximal backoffs of 5, 10.

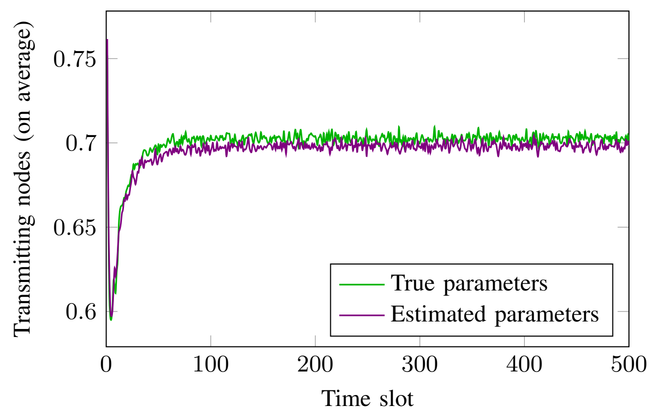


Figure 2. Average number of nodes transmitting in every time slot for the two-node example problem with true parameters 0.25, 0.5, and 5, 10, and estimated parameters 0.22, 0.54, and 6, 9.

$Q_t(z_t, y_1^t)$ that, as it is clear from (37), require iterating over the state space $Z$, the size of which grows exponentially with the number of nodes $N$. Because of this, it is problematic to apply this method to large networks. Research of whether accurate estimations of probabilities (35) can be obtained without this expensive computation is left for future work.

## VI. Numerical Results

### A. Network Detection

For the purposes of illustration of the method proposed in Section IV, let us consider a network where both packet arrival probability distributions $\alpha^{(k,t)}(\cdot, \tau)$ and backoff probability distributions $\delta^{(k,t)}(\cdot, \theta)$ are stationary and controlled by a single parameter on every node $k$. We observe that such assumption is not necessary to carry out the estimation and prediction procedures described earlier.

To simplify presentation, we set $A = 1$, so that, in any given time slot at any node, at most one packet can arrive with probability

$$\alpha^{(k,t)}(x, \tau) = \begin{cases} \exp(-\tau^{(k)}), & \text{if } x = 0; \\ 1 - \exp(-\tau^{(k)}), & \text{if } x = 1; \end{cases} \tag{40}$$

where the parametrization for $\tau^{(k)}$ is chosen to eliminate explicit constraints on the probabilities remaining in the interval $[0, 1]$. Random backoff for collisions is assumed to be distributed uniformly in the integer range from 1 up to a node-specific parameter $\theta^{(k)}$:

$$\delta^{(k,t)}(j, \theta) = \begin{cases} 1/\theta^{(k)}, & \text{if } 1 \le j \le \theta^{(k)}; \\ 0, & \text{otherwise.} \end{cases} \tag{41}$$

Our numerical experiments for illustrative problems presented below were organized as follows. For parameters $\tau^{(k)}$ and $\theta^{(k)}$ being fixed for each node $k$, we simulated the evolution of a network induced by those true parameters, and, thus, obtained a training sample $n_1, n_2, \ldots, n_T$ of the number of transmitting nodes $\eta_t$ for the first $T$ time slots.

Then, we ran the optimization procedure based on the Nelder-Mead optimization algorithm [29], [32], [33], as described in Section IV. Taking into consideration the discrete nature of the backoff parameters $\theta^{(k)}$, we added the following regularization term to the optimized likelihood function:

$$\begin{aligned} \widetilde{L}(n_{1\ldots T} \mid \theta, \tau, c) &= L(n_{1\ldots T} \mid \theta, \tau, c) \\ &\quad - \gamma_m \sum_{k=1}^{N} \left|\theta^{(k)} - [\theta^{(k)}]\right|, \end{aligned} \tag{42}$$

where $[\cdot]$ denotes the nearest integer function, and $\gamma_m > 0$ is a weight that decreases with the increasing iteration number. In all cases, we started optimization with initial parameters for all nodes $k$ being set to the smallest possible maximal backoff $\theta_0^{(k)} = 1$ and arrival probabilities defined by $\tau^{(k)} = -\log(0.999)$ corresponding to the arrival rates of 0.001.

Figure 1 shows the process of likelihood optimization for sample of $T = 500$ observations in the two-node problem ($N = 2$) with true parameters corresponding to arrival rates of 0.25, 0.5, and maximal backoffs of 5, 10, respectively. The top plot depicts estimated arrival rates as they evolved according to the parameters $\tau$ during optimization. Similarly, the bottom plot depicts estimated maximal backoff parameters. In both of the plots, red lines are used for characteristics of the first node, and blue lines are used for those of the second node. Dashed lines show the values corresponding to the true parameters with which the training sample used in optimization was generated.

Figure 2 depicts the sampling mean of the number $\eta_t$ of the nodes transmitting in each time slot. The green line corresponds

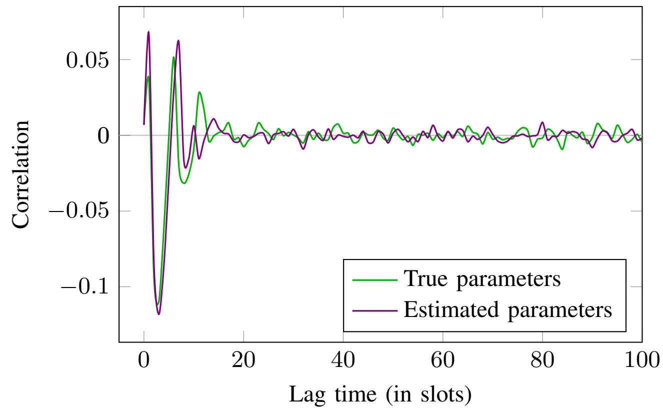


Figure 3. Autocorrelation functions for the number of transmitting nodes generated for the two-node example problem with true parameters 0.25, 0.5, and 5, 10, and estimated parameters 0.22, 0.54, and 6, 9.

Table I
FREQUENCIES OF THE NUMBER OF TRANSMITTING NODES FOR THE TWO-NODE EXAMPLE PROBLEM

| | True parameters (0.25, 0.5, and 5, 10) | Estimated parameters (0.22, 0.54, and 6, 9) |
|---|---|---|
| 0 | 0.422 | 0.433 |
| 1 | 0.456 | 0.437 |
| 2 | 0.122 | 0.131 |

to the random process induced by the true parameters, the purple line to the process controlled with the estimated parameters shown on the right edges of the plots in Figure 1. Multiple samples were generated for both parameter vectors, and the average was taken separately for each time slot.

As expected, the processes resulting from the two parameter vectors are very similar. It is confirmed, for instance, by the correlation structure of the signals shown in Figure 3, which depicts the autocorrelation functions for two samples of 100 000 time slots generated with the true and estimated parameters. Histograms of the two samples are shown in Table I.

However, as we suggested in Section IV, it is not always possible to obtain exact parametric identification of all nodes due to the two factors: low information capacity of the observed signal, and difference between the true likelihood function and its approximation. Effects of both factors vary for different parametric regions.

The former factor plays a major role when the network under consideration operates in extreme regimes or those close to extreme. When the traffic arrival is rare and transmissions are scarce, the consequent frequency of collisions is low and, therefore, the amount of information about the maximal backoff parameters in a given sample is also low. Conversely, when the traffic arrival is so frequent that packet buffers on transmitters rarely become empty or, even worse, most of the time are filled up to their limit, there is almost no information on the arrival rate parameter present in any sample.

The other factor is present due to the quasi-likelihood function (12) that we chose to approximate the true likelihood of a sample. In it, for the benefit of acquiring performance gains explained in Section IV, we purposefully omit the influence of temporal dependencies between observations. Although it is difficult to predict the negative effect of this omission on the quality of parametric estimation, it is reasonable to expect that the deviation from the true likelihood generally grows with the frequency of collisions increasing.

In order to illustrate how the combination of the above two factors affects the quality of resulting estimations, let us consider parametric identification for a network of three identical nodes, all parametrized with the same arrival probability and maximal backoff limit. Figure 4 depicts the absolute error in estimations of both parameters obtained from the maximum of the quasi-likelihood (12). Each point on all four plots corresponds to a single problem with corresponding true values of the arrival probability and the maximal backoff. The plots *(a)* and *(b)* in the left column show the absolute value of the difference between the true arrival probability (x-coordinate) and its estimation. Similarly, the plots *(c)* and *(d)* in the right column show the absolute value of the difference between the true maximal backoff (y-coordinate) and its estimation. In each plot, the white-hatched area depicts the region of congestion, where the average rate of packet arrival exceeds the average rate of successful packet transmission.

As we can see in Figure 4, if we limit our consideration to the problems outside of the congestion region, the quality of estimation is sufficiently accurate. For the arrival probability, there is an emerging region of small errors in the vertical stripe in the range from approximately 0.09 to 0.16, that is visible in Figure 4*(b)*. We conjecture that, in this region, the estimated arrival probability start to diverge from its true value due to proximity to the boundary of the congestion region, where the probability of collisions increases and, consequently, the effects of temporal dependencies in the observations grows.

As it is visible in Figure 4*(d)*, for the maximal backoff parameter, most inaccuracies are distributed in the region of maximal backoff being 10 or greater, along the y-axis where arrival probability is roughly 0.05 or less. The divergence of the estimated maximal backoff in this area can be explained by the specifics of the network regimes in this region. When packet arrival probability is low, collisions are rare, and when, in addition to that, retransmission pressure is also low because of high maximal backoff limits, information density per observation for the backoff parameter is small, requiring longer samples.

### B. Policy Adaptation

Let us now illustrate the policy adaptation procedure proposed in Section V. The setting here is the same as in the network estimation examples shown above, but with additional, cognitive node that does not follow the same random backoff protocol as the other two nodes. Instead, the cognitive node makes a decision on whether to transmit based on the rule (39). For simplicity of illustration, we assume that it has the arrival rate of one, and, therefore, always has a packet for transmission.

Numerical experiments were set as follows. In each of the $T$ time slots of an experiment, first, conditional probabilities (35) were calculated using the number $n_{t-1}$ of the non-adaptive nodes that transmitted in the preceding time slot $t-1$, and all

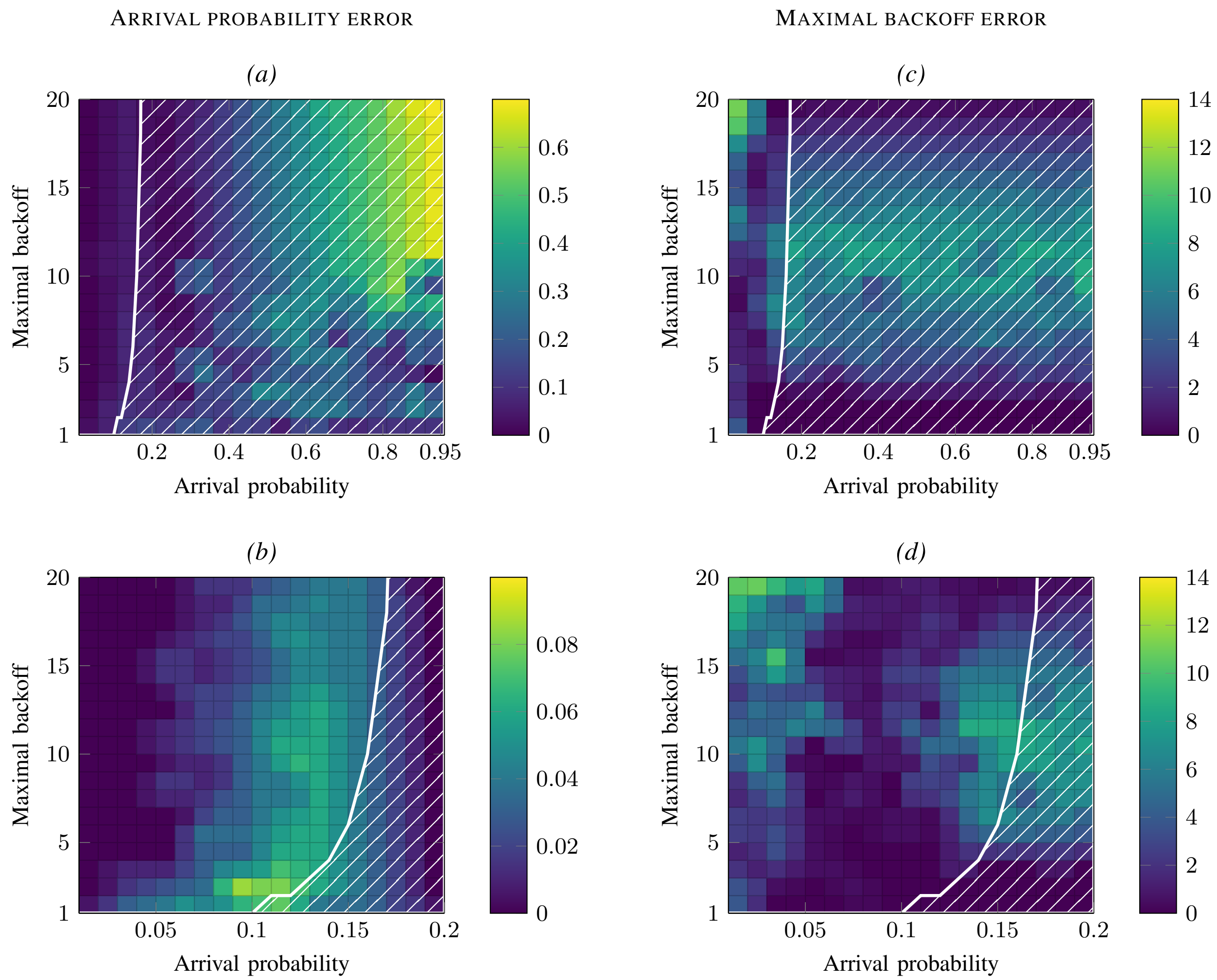


Figure 4. Absolute error in parametric identification of arrival rate (*a*, *b*) and maximal backoff (*c*, *d*) estimated on samples of 1000 time slots for different problems with three identical nodes. The white-hatched area depicts the congestion region.

of the previous decisions $y_1, y_2, \ldots, y_{t-1}$ of the cognitive node. Then, the transmission decision $y_t$ of the cognitive node for the current slot $t$ was computed using (39). Finally, activity of the non-adaptive nodes was simulated according to the assumed random-backoff protocol, and the number of transmitted nodes was recorded as $n_t$. Throughput of every node (including the cognitive one) was measured as the ratio of the number of packets arrived for transmission and the number of packets successfully transmitted without interference in each time slot.

There are two sources of information that significantly affect behavior of the cognitive node operating in accordance with the decision rule (39). One is the parameters $\theta$ and $\tau$ of the non-adaptive nodes. If they are assumed to be unknown, and it is their estimations that are used to determine the adaptive strategy, performance of the cognitive node becomes dependent of the quality of estimation. How and to what extent does that dependence influence performance of the cognitive node is the question of future research.

The other significant value is the proper parameter $\varepsilon_0$ of the adaptive decision (39). If the value of $\varepsilon_0$ is too high, there is a risk of the adaptive node transmitting too often and damaging performance of other nodes and, in some network regimes, its own performance. Conversely, if it is too low, transmission attempts of the cognitive node are too rare for its satisfactory performance.

Figure 5*(a)* depicts throughputs of all nodes for different values of $\varepsilon_0$, for the problem where both non-adaptive nodes have packet arrival probability of $0.1$ and maximal backoff of $4$. Similarly, Figure 5*(b)* illustrates the problem, again, with two identical non-adaptive nodes with arrival rate of $0.4$ and maximal backoff of $10$. Finally, Figure 5*(c)* corresponds to the problem with two different non-adaptive nodes with arrival rates of $0.2$ and $0.4$, and maximal backoffs of $5$ and $10$. Each of the curves is obtained from a sample of multiple experiments, and depicts the median throughput of all three nodes for different thresholds $\varepsilon_0$. The areas highlighted in the background of the curves show the respective interdecile ranges.

In the first problem, arrival rates on the non-adaptive nodes are low enough for the damage of the cognitive node on them to remain insignificant, so the threshold $\varepsilon_0$ can be selected to simply maximize the throughput of the adaptive node. In the second and third problem, however, it becomes clear that the choice of $\varepsilon_0$ should be guided by the trade-off between performance of the adaptive node and the non-adaptive nodes.

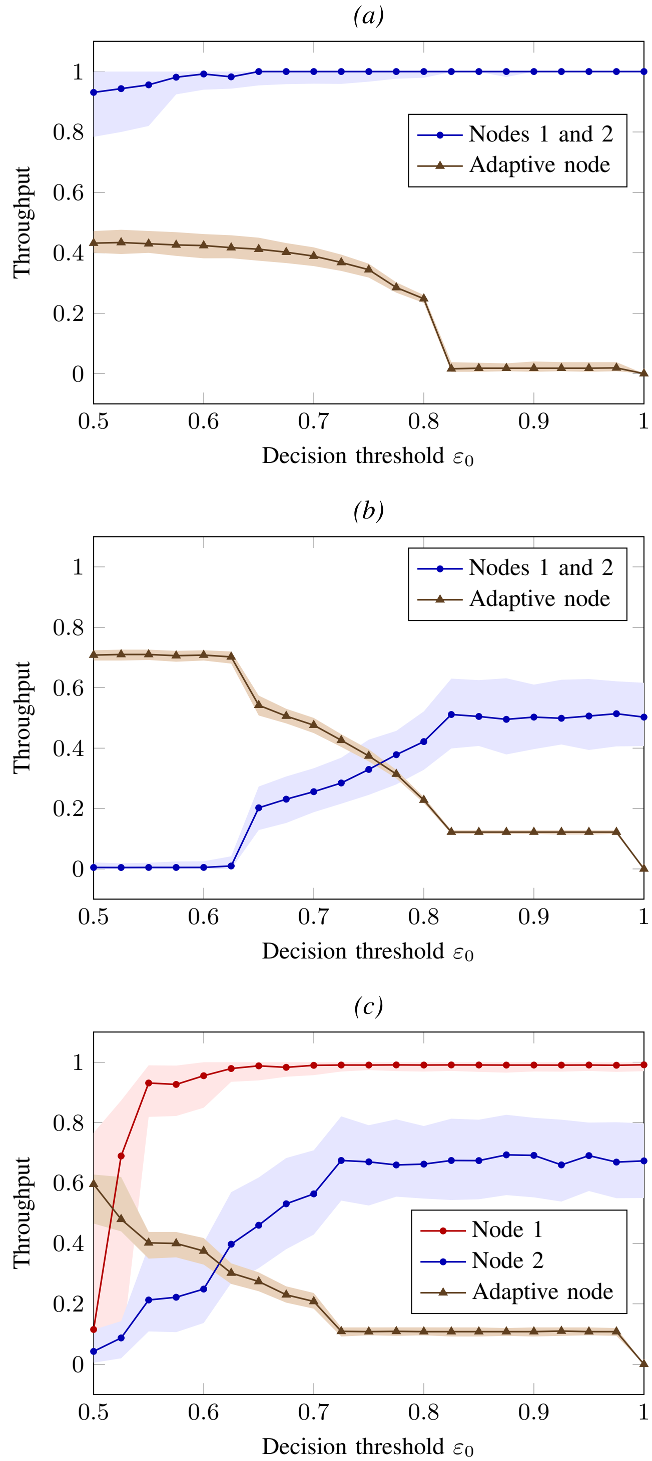


Figure 5. Average efficiency of the adaptive node for different values of parameter $\varepsilon_0$ compared to that of non-adaptive nodes with *(a)* arrival rate of 0.1 and maximal backoff of 4, *(b)* arrival rate of 0.4 and maximal backoff of 10, *(c)* arrival rates of 0.2, 0.4 and maximal backoffs of 5, 10.

For instance, the following simple rule can be used. Given lower bounds $\beta_k$ on the average throughput of all non-adaptive nodes, find the intervals of $\varepsilon_0$ in which the throughput $v_k(\varepsilon_0)$ of every non-adaptive node $k$ is greater than or equal to $\beta_k$. Among the points satisfying this condition, select one with the highest throughput $v_{\mathrm{A}}(\varepsilon_0)$ of the cognitive node:

$$\varepsilon_0^* = \underset{\substack{0<\varepsilon_0<1 \\ \forall k\ v_k(\varepsilon_0)\geq\beta_k}}{\arg\max}\, v_{\mathrm{A}}(\varepsilon_0). \tag{43}$$

This selection rule can be used both for throughputs estimated on one particular experiment, or throughputs averaged out over multiple experiments done for the same parameters of the non-adaptive nodes.

Figure 6 shows a typical fragment of the behavior of the cognitive node when $\varepsilon_0 = 0.6$, and both non-adaptive nodes have parameters of 0.1 and 4, as in the problem illustrated by Figure 5*(a)*. Blue bars depict the number $\eta_t$ of transmissions of the non-adaptive nodes, whereas brown bars correspond to the moments of transmission of the cognitive node ($y_t = 1$). Solid black line reflects changes in the conditional probability $P(\eta_t = 0 \mid \eta_{t-1}, y_1^{t-1})$ of the non-adaptive nodes being silent, as estimated by the cognitive node during the experiment. Dashed line signifies the value of the threshold $\varepsilon_0$.

As it is visible in Figure 6, probability $P(\eta_t = 0 \mid \eta_{t-1}, y_1^{t-1})$ jumps to values close to 1 in the time slots immediately following the ones with the two non-adaptive nodes colliding ($\eta_{t-1} = 2$). As we move away from a collision, the probability decreases until, after some activity of the non-adaptive nodes, it starts to rise again, adaptively reacting to their behavior.

## VII. Future Work

This paper addresses the estimation of the traffic and protocol parameters driving the dynamics of a wireless network where multiple terminals share the channel resource. The exploratory results presented herein open to a new class of cognitive networks, and emphasize the limitations of direct estimation, both due to the capacity of the observation channel and the unavoidable approximations imposed by complexity. Important components and algorithms will need to be built to enable a leap forward in cognitive technologies.

*1) Information Channel:* The information conveyed by observations should be thoroughly studied in relation with the implemented networking protocols and parameters. This study will enable the design of scenario-specific functionalities and estimator classes.

*2) Heterogeneous Protocols:* In the next generation of networks, the nodes may dynamically adapt their transmission protocols to the specific application being served. Although the proposed methodology applies to such scenario, the number of terminals implementing a protocol from each family needs to be known to run the corresponding dynamic programming. Therefore, algorithms for detecting the protocol family actively used in the network shall be developed to provide this information to the algorithm estimating the parameters.

It is also worth considering the case when the assumption on uniformity of transmission protocols is relaxed. The proposed approach can be adapted to this setting by formulating the problem so that, first, each of the protocols in question satisfies Bellman's principle of optimality for the terminals' state probabilities, to make the node transmission probability computable (for a fixed sequence of collisions) through dynamic programming, as in equations (2)–(8); and second, the

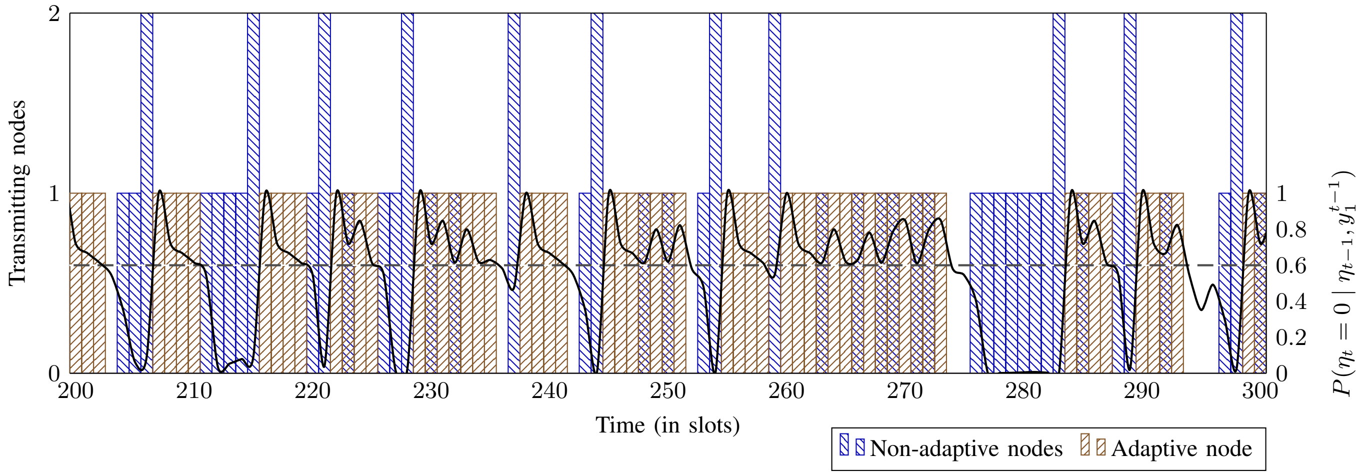


Figure 6. Example of the adaptive behavior of the cognitive node for $\varepsilon_0 = 0.6$ for the problem with two non-adaptive nodes with arrival rates of 0.2, 0.4 and maximal backoffs of 5, 10.

parametrizations of those protocols are merged into a common parameter space (for instance, by concatenating them together and adding new categorical parameters specifying which of the protocols is in use for every node). New parametrization may prevent usage of gradient-based algorithms, and may require adopting techniques from combinatorial optimization.

*3) Time-Varying Network Parameters:* The estimation algorithms should address a scenario where the traffic and protocol parameters change over time in reaction to environmental or application variations. The proposed algorithm can be adapted to this case by adding additional state variables tracking the underlying application and network state. The parameters modeling the dynamics of these variables can be estimated on a larger time scale. A mismatch between the predicted and observed network trajectory could be used to identify large scale state changes.

*4) Alternatives to Complete Parametric Identification:* For the problem of transmission policy adaptation, it is of interest to explore approaches that do not require estimations of packet arrival and transmission processes on all nodes, thus avoiding complete parametric identification. Such approaches may be more robust, because they would require estimating fewer parameters which, furthermore, may be of higher relevance to adaptation decisions. Complete parametric identification may, however, be adequate in other problems, e.g., those occurring in adversarial settings, where detection of particular combinations of arrival-transmission processes is the objective in itself. Furthermore, parametric identification can inform the definition of cost functions designed to customize interference for specific applications, and impose fairness.

## VIII. Conclusions

A framework for the analysis of the network operations in cognitive networks was proposed. The algorithm estimates the traffic and protocol parameters controlling the transmission processes of the observed terminals in a complex network scenario with packet buffering and random channel access. Minimal information is assumed observable to the cognitive terminal. For this challenging scenario, the proposed algorithm can accurately reconstruct the properties of the observed signal, and estimate the generating parameters in some parameter regions, with feasible computational complexity. The estimated parameters grant cognitive terminals the ability to predict the dynamics of the networks and control the interactions triggered by interference.

## References


[1] J. Mitola, “Cognitive radio: an integrated agent architecture for software-defined radio,” Doctor of Technology, Royal Inst. Technol. (KTH), Stockholm, Sweden, 2000.

[2] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, “Breaking spectrum gridlock with cognitive radios: An information theoretic perspective,” *Proceedings of the IEEE*, vol. 97, no. 5, pp. 894–914, May 2009.

[3] S. Haykin, “Cognitive radio: brain–empowered wireless communications,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[4] S. Geirhofer and L. Tong and B. M. Sadler, “Dynamic Spectrum access in the time domain: modeling and exploiting white space,” *IEEE Commun. Mag.*, vol. 45, no. 5, pp. 66–87, May 2007.

[5] Y. Chen and Q. Zhao and A. Swami, “Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors,” *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 2053–2071, May 2008.

[6] M. Levorato, O. Simeone, U. Mitra, and M. Zorzi, “Cooperation and coordination in cognitive networks with packet retransmission,” in *IEEE Information Theory Workshop*, 2009, pp. 495–499.

[7] M. Levorato, U. Mitra, and M. Zorzi, “Cognitive interference management in retransmission-based wireless networks,” *IEEE Trans. on Information Theory*, vol. 58, no. 5, pp. 3023–3046, 2012.

[8] M. Levorato, S. Firouzabadi, and A. Goldsmith, “A learning framework for cognitive interference networks with partial and noisy observations,” *IEEE Trans. on Wireless Communications*, vol. 11, no. 9, pp. 3101–3111, 2012.

[9] N. Michelusi, P. Popovski, O. Simeone, M. Levorato, and M. Zorzi, “Cognitive access policies under a primary arq process via forward-backward interference cancellation,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2374–2386, 2013.

[10] Q. Liang, M. Liu, and D. Yuan, “Channel estimation for opportunistic spectrum sensing: Uniform and random sensing,” in *Information Theory and Applications Workshop*, 2010, pp. 1–10.

[11] T. Clancy and B. Walker, “Predictive dynamic spectrum access,” in *Proc. SDR Forum Technical Conference*, 2006.

[12] S. Yarkan and H. Arslan, “Binary time series approach to spectrum prediction for cognitive radio,” in *IEEE 66th Vehicular Technology Conference*, 2007, pp. 1563–1567.

[13] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[14] Y. Wang, A. Ahmed, B. Krishnamachari, and K. Psounis, "IEEE 802.11p performance evaluation and protocol enhancement," in *IEEE International Conference on Vehicular Electronics and Safety*, 2008, pp. 317–322.

[15] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov 2004, pp. 772–776.

[16] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[17] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler, "Exploiting the capture effect for collision detection and recovery," in *Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*, 2005, pp. 45–52.

[18] J. K. Lee, W. Kim, S. J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi, "An experimental study on the capture effect in 802.11a networks," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. ACM, 2007, pp. 19–26.

[19] C. C. Tan and N. C. Beaulieu, "On first-order Markov modeling for the Rayleigh fading channel," *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2032–2040, 2000.

[20] L. Badia, M. Levorato, and M. Zorzi, "A channel representation method for the study of hybrid retransmission-based error control," *IEEE Trans. on Communications*, vol. 57, no. 7, pp. 1959–1971, 2009.

[21] H. Urkowitz, "Energy Detection of Unknown Deterministic Signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, 1967.

[22] V. I. Kostylev, "Energy Detection of a Signal With Random Amplitude," in *IEEE International Conference on Communications*, 2002, pp. 1606–1610.

[23] R. Couillet and M. Debbah, "A Bayesian Framework for Collaborative Multi-Source Signal Sensing," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5186–5195, 2010.

[24] P. Bianchi, J. Najim, M. Maida, and M. Debbah, "Performance Analysis of Some Eigen-Based Hypothesis Tests for Collaborative Sensing," in *IEEE/SP 15th Workshop on Statistical Signal Processing*. IEEE, 2009, pp. 5–8.

[25] P.-J. Chung, J. F. Böhme, C. F. Mecklenbrauker, and A. O. Hero, "Detection of the Number of Signals Using the Benjamini-Hochberg Procedure," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2497–2508, 2007.

[26] A. Krohn, "Superimposed Radio Signals for Wireless Sensor Networks," Ph.D. dissertation, Carl Friedrich Gauß School, Technical University of Braunschweig, Braunschweig, Germany, 2007.

[27] J. Nocedal, "Updating Quasi-Newton Matrices with Limited Storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[28] D. C. Liu and J. Nocedal, "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[29] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[30] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[31] J. Kiefer, J. Wolfowitz *et al.*, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[32] M. Box, "A new method of constrained optimization and a comparison with other methods," *The Computer Journal*, vol. 8, no. 1, pp. 42–52, 1965.

[33] J. A. Richardson and J. Kuester, "Algorithm 454: the complex method for constrained optimization [e4]," *Communications of the ACM*, vol. 16, no. 8, pp. 487–489, 1973.


**Igor Burago** obtained the B.S. degree in Applied Mathematics and Informatics summa cum laude from the Far Eastern Federal University, Russia, in 2011, and the M.S. degree in Computer and Information Science from the University of Oregon in 2014. He is currently a second-year Ph.D. student in the Department of Computer Science at the University of California, Irvine.

**Marco Levorato** obtained the B.S. and M.S. in Electrical Engineering summa cum laude from the University of Ferrara, Italy in 2005 and 2003, respectively, and the Ph.D. degree in Electrical Engineering from the University of Padova, Italy, in 2009. He is currently an Assistant Professor in Computer Science at the University of California, Irvine. Dr. Levorato has had post-doctoral appointments at Stanford University, the University of Southern California and the Royal Institute of Technology in Stockholm, Sweden. He is the recipient of a Best Paper Award from IEEE Globecom 2012 and has been twice nominated for the Best Young Researcher Award, Department of Information Engineering, University of Padova, Italy.