Optimal Primary-Secondary user Cooperation Policies in Cognitive Radio Networks

Nestor D. Chatzidiamantis, *Member, IEEE*, Evangelia Matskani, Leonidas Georgiadis, *Member, IEEE*, Iordanis Koutsopoulos, *Member, IEEE*, and Leandros Tassiulas, *Fellow, IEEE*

Abstract

In cognitive radio networks, secondary users (SUs) may cooperate with the primary user (PU) so that the success probability of PU transmissions are improved, while SUs obtain more transmission opportunities. However, SUs have limited power resources and, therefore, they have to take intelligent decisions on whether to cooperate or not and at which power level, in order to maximize their throughput. Cooperation policies in this framework require the solution of a constrained Markov decision problem with infinite state space. In our work, we restrict attention to the class of stationary policies that take randomized decisions of an SU activation and its transmit power in every time slot based only on spectrum sensing. Assuming infinitely backlogged SUs queues, the proposed class of policies is shown to achieve the maximum throughput for the SUs, while significantly enlarging the stability region of PU queue. The structure of the optimal policies remains the same even if the assumption of infinitely backlogged SU queues is relaxed. Furthermore, the model is extended for the case of imperfect channel sensing. Finally, a lightweight distributed protocol for the implementation of the proposed policies is presented, which is applicable to realistic scenarios.

Part of this paper has been presented in WiOpt 2014.

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

N. D. Chatzidiamantis, E. Matskani and L. Georgiadis are with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece, (e-mails:{nestoras,matskani,leonid}@auth.gr).

I. Koutsopoulos is with Department of Informatics, Athens University of Economics and Bussiness (AUEB), Greece, and the Centre for Research and Technology Hellas (CERTH), Greece, (e-mail: jordan@aueb.gr).

L. Tassiulas is with the Department of Computer and Communications Engineering, University of Thessaly, Greece, and the Centre for Research and Technology Hellas (CERTH), Greece, (e-mail: leandros@uth.gr).

Index Terms

Opportunistic cooperation, resource allocation, imperfect sensing, distributed implementation.

I. Introduction

Cognitive radio networks (CRNs) have received considerable attention due to their potential for improving spectral efficiency [1], [2], [3]. The main idea behind CRNs is to allow unlicensed users, also known as *secondary users* (SU), to identify temporal and/or spatial spectrum “holes”, i.e., vacant portions of licensed spectrum, and transmit opportunistically, thus gaining access to the underutilized shared spectrum while maintaining limited interference to the licensed user, also known as *primary user* (PU). This communication paradigm has been referred to as “Dynamic Spectrum Access” (DSA) in the technical literature [4], [5].

Much prior work on DSA CRNs has been focused on the problem of optimal spectrum assignment to multiple SUs [6], [7], [8]. Several resource allocation algorithms have been proposed, based on either the knowledge of PU transmissions obtained from perfect spectrum sensing mechanisms [6] or from a probabilistic maximum collision constraint with the PUs [7]. Of particular interest is the opportunistic scheduling policy for SUs suggested in [8], which maximizes SUs’ throughput utility while guarantees low number of collisions with the PU, as well. In all these works it is assumed that no interaction between PUs and SUs exists.

Recently, the concept of cooperation between PU and SUs in CRNs emerged, as a means for providing benefits for both types of users. These benefits stem from the fact that, by exploiting the transmit power resources of SUs towards improving the effective transmission rate of the PU, the chances that the PU queue will be empty are increased, and hence the PU channel is free to use more often.

From an information theoretic perspective, cooperation between SUs and PUs at the physical layer has been investigated in many works (see [9] and references therein). Queuing theoretic aspects and spectrum leasing strategies for cooperative CRNs have been investigated in [10], [11], [12], [13], [14]. Specifically, spectrum leasing strategies where the PU leases a portion of its spectrum to SUs in return for cooperative relaying were suggested in [10]. A protocol where a SU relays the PU packets that have not been correctly received by their destination, was suggested and investigated in terms of SU stable throughput in [11], while similar protocols were suggested and compared in [12], considering various physical layer relaying strategies. In [13], the performance of a specific class of PU-SU cooperation policies was investigated in terms of PU and SU stable throughput, assuming that SU is allowed to transmit simultaneously with the PU, even if the PU is busy.

In this work we study optimal cooperative PU-SUs transmission control algorithms with the objective to make as efficient use of the PU channel as possible, namely maximize a function of the transmission rates of the SUs, while guaranteeing unobstructed packet transmission for the PU, and stability of its queue. SUs have limited transmit power resources, therefore intelligent cooperation decisions must be taken. This is the main idea behind the work in [14], where a dynamic decision policy for the SUs activities (i.e., whether to relay PU transmissions and at which power level) is suggested. The proposed policy is proved to be optimal, however, its basic requirement is that the PU packet arrival rates must be lower than a threshold value, which guarantees that the PU queue is stable even when SUs never cooperate. This regime places significant restrictions on the achievable PU stability region, since the sustainable arrival rates of PUs may be much larger than this threshold value.

We present policies that significantly increase the range of PU arrival rates for which PU-SUs cooperation can be beneficial. Specifically, we investigate transmission policies for cooperative CRNs that can be applied even when PU transmission rates are above the threshold set by [14], while still permitting the SUs to utilize the channel for their own transmissions. Since the SU decision options and success probabilities are different during the idle and busy PU periods, while the PU queue size is in turn affected by the cooperation decisions, such policies require in general the solution of a non-trivial constrained Markov decision problem with infinite state space, where the state is the size of the PU queue. The solutions for such Markov decision problems suffer from large convergence times and their implementation in general requires knowledge of the PU queue size [15].

The main contributions of this work are summarized as follows.

1) We introduce a class of stationary policies which take random decisions on SU activities in every time-slot based only on the PU channel spectrum sensing result, i.e., the PU channel being busy or idle. The proposed class of policies is applicable when either SUs are infinitely backlogged or a general SU packet arrival process is assumed. The benefits of our approach are as follows. First, our approach is proven to achieve the same set of

SU rates as the more general policies in which (i) decision may depend on the PU queue size, or (ii) a SU packet may be transmitted instead of a PU packet when the PU queue is non-empty. Hence, the policies in the proposed class of stationary ones are sufficient for optimality with respect to any utility function. Second, compared to other policies, it allows for a significantly larger range of PU traffic arrival rates for which the PU queue is stable, thus increasing the PU throughput. Even more interestingly, the enlargement of the PU stability region still allows the SUs to utilize slots that are unused by the PU, in order to transmit their own traffic. Finally, as long as the system parameters remain the same, the decision variables associated with our policy may be computed offline, through solving a convex optimization problem via efficient interior point methods, and can be used to realize the policy in real-time.

2) Since the proposed policies are based solely on the PU channel state sensing result, we also investigate the effects of imperfect spectrum sensing mechanism in their performance. Considering this case, we incorporate all possible sources of errors and inefficiencies in our model and describe the new performance space of the proposed policies. However, when channel sensing errors are introduced, the determination of the associated control variables requires the solution of a non-convex optimization problem and the optimal solution becomes hard to determine.
3) A distributed implementation of the proposed cooperation policies, applicable to the case of concave SU utility functions, is designed, which is based on a decentralized computation of the problem control variables via the alternating direction method of multipliers. This version offers a robust alternative to the centralized implementation and distributes the computational burden across network nodes without loss in performance.

The remainder of the paper is organized as follows. In Section II, we introduce the system model. In Section III we describe the mode of operation of the proposed restricted class of randomized policies and show their optimality. Exogenous packet arrivals to SU queues and the effects of imperfect spectrum sensing mechanism are investigated in section IV. The distributed implementation of the proposed class of policies is developed in Section V. Section VI presents simulation results and finally, concluding remarks are provided in Section VII.

II. System Model

We consider the system model with one PU and multiple SUs depicted by Fig. 1. Specifically, the PU is the licensed owner of the channel and transmits whenever it has data to send. On the other hand, SUs do not have any licensed spectrum and seek transmission opportunities on the PU channel. We assume that one[1] of the SUs can cooperate with the PU in order to improve the success probability of PU transmissions. This can be achieved by allocating part of the SU power resources towards that purpose. In practice, SU cooperation may be realized with various techniques that span one or more communication layers. For example, the SU may relay PU traffic (e.g. through decode-and-forward, or amplify-and-forward) [14]. Alternatively, this aid by the SU can be provided by means of link layer techniques, such as retransmission of the overheard PU packet by the SU, or even through physical layer techniques (e.g. simultaneous transmission of the PU packet by the SU, in order to improve the signal-to-interference-plus-noise ratio at the PU receiver) [12]. The model is transparent to capture the generality of all these techniques, all of which are factored in the problem in terms of the SU consumed transmit power resources.

Furthermore, after sensing the PU channel, SUs decide on which SU will cooperate so as to transmit PU data and at which power level (if the PU channel is busy), or which SU will transmit its own data and at which power level (if the PU channel is idle). In what follows we describe the parameters of the system model under consideration as well as the available controls.

A. System Model Parameters

We consider the time-slotted model, where time slot $t = 0, 1, ...$ corresponds to time interval $[t, t+1)$; t and $t+1$ are called the "beginning" and "end" of slot t respectively. The PU queue receives new packets in each time slot t according to an independent and identically distributed (i.i.d.) arrival process $A_p\left(t\right)$ with mean rate $\mathbb{E}\left[A_p(t)\right] = \lambda_p$ packets/slot and $\mathbb{E}\left[\left(A_p\left(t\right)\right)^2\right] < \infty$. We assume that the SUs are backlogged so that they *always* have packets to transmit.

We denote by $\mathcal{S}$ the set of SUs. Each SU $s \in \mathcal{S}$ can transmit using one of I_s power levels, $P_s\left(i\right)$, $i = 1, ..., I_s$, where $P_s(i) < P_s(i+1)$. To simplify the description that follows, we set

[1]The presented analysis can be applied in cases where more than one SUs can cooperate with PU, by replacing the selected SU by a subset of SUs.

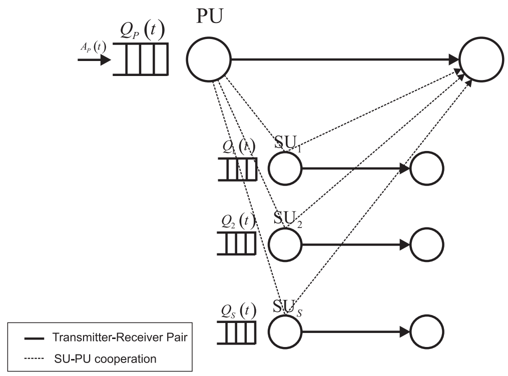

Fig. 1. The system model under consideration.

$P_s(0) = 0$. An SU s may use any of these power levels to either transmit its own data or to assist the PU as discussed above. At each time slot, only a single packet transmission can take place. Furthermore, when transmission of packets from the PU takes place, at most one of the SUs can cooperate. There is a constraint on the long-term average power $\hat{P}_s$ consumed by each $s \in \mathcal{S}$. Hence, for every $s \in \mathcal{S}$, if $i(t)$ is the power level used by s at slot t, it must hold,

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t} \mathbb{E}\left[P_s\left(i\left(\tau\right)\right)\right] \leq \hat{P}_s,\ i\left(\tau\right) \in \mathcal{I}_s^0, \tag{1}$$

where $\mathbb{E}[\cdot]$ denotes expectation, $\mathcal{I}_s = \{1, 2, ..., I_s\}$ and $\mathcal{I}_s^0 = \mathcal{I}_s \cup \{0\}$.

We assume an erasure channel model, i.e., that each transmission (by the PU or one of the SUs) is either received correctly or erased.

- When SU s transmits one of its own packets with ith power level, $i \in \mathcal{I}_s^0$, the probability of success is $r_s(i)$, where $r_s(0) = 0,$ i.e. the success probability is zero if no power is used for transmission.
- When SU s cooperates with the PU, (namely it assists in the transmission of PU packets by transmitting with ith power level), the success probability of the PU transmitted packet is $r_p\left(s, i\right)$. If $i = 0$, the SU "cooperates" with zero transmission power, hence in effect no cooperation takes place; therefore it is natural to assume that $r_p(s, 0) = r_p(0) \geq 0$ for all $s \in \mathcal{S}$, where $r_p(0)$ denotes the probability of successful packet transmission by the PU when the SUs do not cooperate. In addition, we assume that $r_p\left(s, i\right) \leq r_p\left(s, i+1\right)$, i.e., the probability of successful reception is a non-decreasing function of transmission power.

B. Available Controls

In the beginning of time slot t there are various control options, depending of the status of the primary queue $Q_p(t)$. In case $Q_p(t) > 0$ (namely, the PU channel is busy), then the available controls are:

- A packet from the PU queue is transmitted, and transmission of SU packets is excluded. We refer to this constraint as the *PU priority constraint*.
- A SU s is selected for cooperation with the PU in order to assist the transmission of the PU packet.
- The ith power level, $i \in \mathcal{I}_s^0$, is selected, so that s cooperates with the PU using power level $P_s(i)$. When $i(t) = 0$ no cooperation takes place.

On the other hand, when $Q_p(t) = 0$ (namely, the PU channel is idle), the available controls are the following:

- A SU s is selected to transmit its own packet.
- The ith power level, $i \in \mathcal{I}_s^0$, is selected, so that s transmits its own packets using power level $P_s(i)$. If $i = 0$, no transmission takes place in slot t.

C. Admissible Policies, Rate Region, Performance Objective and Extended class of Policies

A control policy is called *admissible* if the following policy constraints are satisfied:

- PU priority constraint is satisfied.
- The PU queue must be mean-rate stable, i.e., the output long-term average rate of the PU queue should be equal to its long term average input rate [16].
- The average power constraints of (1) are satisfied.

Under an admissible policy, each SU $s \in \mathcal{S}$ obtains a long-term average transmission rate equal to

$$\bar{r}_s = \lim_{t\to\infty} \inf \frac{\sum_{\tau=0}^{t-1} \mathbb{E}\left[r_s(P_s(i(\tau))\right]}{t} \tag{2}$$

where $P_s(i(t))$ is the power level at which s transmits in slot t. In the sequel, we denote by $\bar{\mathbf{r}}$ the vector of the long-term average transmission rates of SUs, i.e., $\bar{\mathbf{r}} \triangleq \{\bar{r}_s\}_{s\in\mathcal{S}}$. The *achievable rate region* for the problem under consideration is defined as the set of vectors of SU rates $\bar{\mathbf{r}}$ that can be obtained by all admissible policies.

The selection of an admissible policy depends on the particular optimization objective, which is expressed as a function of the vector of achievable long-term average SU transmission rates $\bar{\mathbf{r}}$. The optimization objective is of the form:

$$\text{maximize: } f\left(\bar{\mathbf{r}}\right) \tag{3}$$

where $\bar{\mathbf{r}}$ belongs to the rate region. In the simplest case, $f\left(\cdot\right)$ is a linear function of $\bar{\mathbf{r}}$, however, fairness considerations may require $f\left(\cdot\right)$ to be a nonlinear (usually separable) function of $\bar{\mathbf{r}}$, [17], [18].

The PU queue size $Q_p(t)$ can be seen as the state of a constrained Markov Decision Process problem [15], where the constraints are imposed by the policy constraints described above. Let $\mathcal{C}_1$ be the class of admissible policies of this Markov Decision Process. This class contains policies that are based on past history actions and includes the class of randomized stationary policies of the following form:

- When $Q_p(t) = m,\ m > 0$, select a SU s to cooperate with the PU at ith power level with a certain probability that depends on m.
- When $Q_p(t) = 0,$ select a SU s to transmit its own packets at ith power level with a certain probability.

Consider a subclass of the policies in $\mathcal{C}_1$, denoted by $\mathcal{C}_0$, which consists of policies whose decisions are based solely on whether the PU queue is zero or not. In each time slot t, a policy in $\mathcal{C}_0$ acts as follows:

- When $Q_p(t) > 0,$ or equivalently the PU channel is sensed busy, select a SU s to cooperate at ith power level with a probability $q\left(s, i\left|b\right.\right)$.
- When $Q_p(t) = 0,$ or equivalently the PU channel is sensed idle, select a SU s to transmit its own data at ith power level with probability $q\left(s, i\left|e\right.\right)$.

Since the policies in $\mathcal{C}_0$ are not based on the actual value of $Q_p\left(t\right)$, but only whether $Q_p\left(t\right)$ is greater than or equal to zero, it follows that $\mathcal{C}_0 \subseteq \mathcal{C}_1$.

For the analysis that follows, it is helpful to introduce the extended class of policies $\mathcal{C}_2$ which follow the policy constraints with the exception the PU priority constraint, i.e., when the PU queue is non-empty at the beginning of a slot, the policy may select to transmit one of the SU packets instead of a PU packet. In this case, the available controls at the beginning of each time slot are of the form $(u, s, i),\ u \in \left\{1, 0\right\},\ s \in \mathcal{S},\ i \in \mathcal{I}_s^0$, where

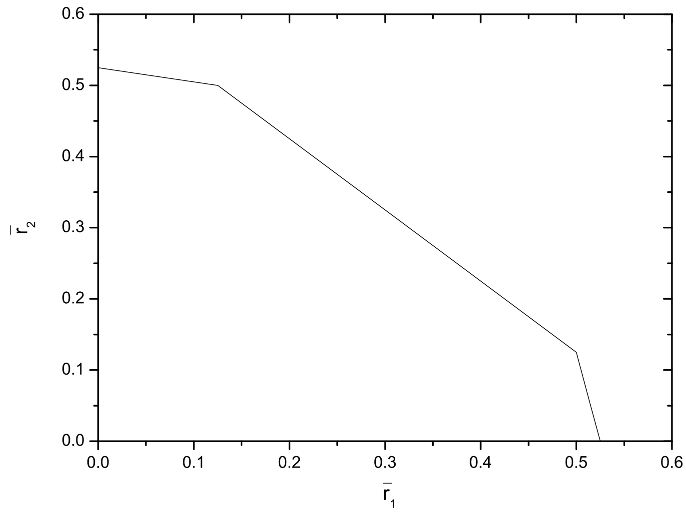

Fig. 2. The rate regions $\mathcal{R}_0$, $\mathcal{R}_1$ and $\mathcal{R}_2$, which coincide, for the system setup scenario with $\mathcal{S} = \{1, 2\}$, $\lambda_p = 0.3$, and $\mathcal{I}_s^0 = \{0, 1, 2, 3, 4\}$, $\mathbf{P}_s = \{0, 0.25, 0.5, 0.75, 1\}$, $r_p(0) = 0.4$, $r_p(s, 1) = 0.5$, $r_p(s, 2) = 0.6$, $r_p(s, 3) = 0.7$, $r_p(s, 4) = 0.8$, $r_s(1) = 0.3$, $r_s(2) = 0.5$, $r_s(3) = 0.8$, $r_s(4) = 1$, $\hat{P}_s = 0.5$, for all $s \in \mathcal{S}$.

- Control $(1, s, i)$, dictates transmission of PU traffic and assigns SU s at ith power level to cooperate with the PU. Note that this control can be assigned even if the PU queue is empty, in which case no packet is transmitted.
- Control $(0, s, i)$, dictates transmission of only SU traffic, and selects SU s to transmit at ith power level.

Since policies in $\mathcal{C}_2$ do not impose the PU priority constraint, and they may include even non-stationary policies, it follows that $\mathcal{C}_1 \subseteq \mathcal{C}_2$. Hence, it holds that $\mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \mathcal{C}_2$ and the corresponding achievable rate regions $\mathcal{R}_0,\ \mathcal{R}_1,\ \mathcal{R}_2$, satisfying the policy constraints under the classes of policies $\mathcal{C}_0,\ \mathcal{C}_1,\ \mathcal{C}_2$, satisfy $\mathcal{R}_0 \subseteq \mathcal{R}_1 \subseteq \mathcal{R}_2$.

It might seem at first glance that a policy in class $\mathcal{C}_0$ with a restricted control space will lead to suboptimal performance. However, this is not the case. In the next section we show that $\mathcal{R}_2 \subseteq \mathcal{R}_0$, thus reaching the interesting key conclusion that $\mathcal{R}_0 = \mathcal{R}_1 = \mathcal{R}_2$. The rate regions $\mathcal{R}_0, \mathcal{R}_1$ and $\mathcal{R}_2$ (which coincide) for a particular system setup scenario with 2 SUs are illustrated in Fig. 2. Hence, under any optimization objective, *it suffices to restrict attention to policies in* $\mathcal{C}_0$ *even if one has the freedom of not adhering to the PU priority constraint.*

III. Characterization of Achievable Rate Regions $\mathcal{R}_0$, $\mathcal{R}_1$, $\mathcal{R}_2$

In this section we substantiate our previous claim. Towards this end, we first determine the achievable rate region of policies in $\mathcal{C}_0$, namely $\mathcal{R}_0$, in subsection (III-A), as well as the stability region of the PU queue when policies in class $\mathcal{C}_0$ are employed. Second, we determine the achievable rate region of policies in $\mathcal{C}_2$, namely $\mathcal{R}_2$, in subsection (III-B), and finally we prove that $\mathcal{R}_0$ coincides with $\mathcal{R}_2$.

A. Achievable Rate Region of Policies in Class $\mathcal{C}_0$

For a given policy π in class $\mathcal{C}_0$, the average packet service rate of the PU queue is given by

$$\bar{r}_p = \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} r_p(s,i) q(s,i\,|b\,). \tag{4}$$

Standard results from queuing theory show that the stability region of the PU queue under π, that is, the closure of the set of PU arrival rates λ_p for which the PU queue is mean-rate stable [16], is the set of arrival rates that fall in the interval $[0,\bar{r}_p]$. Assume next that $\lambda_p \in [0,\bar{r}_p)$ (so that the PU queue is stable) and let q_b be the steady state probability that the PU queue is busy under π. Viewing the transmitter at the PU as a queuing system holding 0 (if the PU queue is empty) or 1 packets (i.e., the packet whose transmission is attempted if the PU queue is non-empty) and applying Little's formula to this system, we have

$$q_b = \Pr\left\{\text{PU queue is non-empty}\right\} = \frac{\lambda_p}{\bar{r}_p}. \tag{5}$$

Hence, the steady state probability that the PU queue is empty is $q_e = 1 - q_b$. Due to the imposed PU priority constraint, SUs may transmit their own data only when the PU queue is empty. Hence, the average packet transmission rate of SU s traffic is equal to

$$\bar{r}_s = \left(\sum_{i\in\mathcal{I}_s} r_s\left(i\right) q\left(s,i\,|e\,\right)\right) q_e. \tag{6}$$

The average power consumption of SU $s \in \mathcal{S}$ is

$$\bar{P}_s = q_e \sum_{i\in\mathcal{I}_s} P_s\left(i\right) q(s,i\,|e\,) + q_b \sum_{i\in\mathcal{I}_s} P_s\left(i\right) q(s,i\,|b\,) \tag{7}$$

and since $\pi \in \mathcal{C}_0$, it satisfies the power constraints (1), i.e., $\bar{P}_s \le \hat{P}_s$, $s \in \mathcal{S}$. The discussion above shows that the constraints that need to be satisfied by the set of probabilities

$\{q_b, q(s,i|b), q(s,i|e), q_e\}$ $s \in \mathcal{S}$, according to (1), (5), are given by

$$q_b \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} r_p(s,i) q(s,i|b) = \lambda_p \tag{8}$$

$$q_e \sum_{i\in\mathcal{I}_s} P_s(i)\, q(s,i|e) + q_b \sum_{i\in\mathcal{I}_s} P_s(i)\, q(s,i|b) \le \hat{P}_s,\ \ s \in \mathcal{S} \tag{9}$$

$$q_b + q_e = 1 \tag{10}$$

$$\sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} q(s,i|b) = 1 \tag{11}$$

$$\sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} q(s,i|e) = 1 \tag{12}$$

$$q_b \ge 0,\ q_e \ge 0,\ q(s,i|b) \ge 0,\ q(s,i|e) \ge 0,\ \ \ s \in \mathcal{S},\ \ i \in \mathcal{I}_s^0 \tag{13}$$

Conversely, given the set of probabilities $\{q_b, q(s,i|b), q(s,i|e), q_e\}_{s\in\mathcal{S},\ i\in\mathcal{I}_s^0}$ that satisfy the constraints (8)-(13), with $q_b < 1$, an admissible policy in $\mathcal{C}_0$ can be defined. Hence, the performance space of these policies is the set of $\bar{\mathbf{r}}$ defined by (6), where the set of probabilities $\{q_b, q(s,i|b), q(s,i|e), q_e\}_{s\in\mathcal{S},\ i\in\mathcal{I}_s^0}$ satisfy constraints (8)-(13).

While constraints of (8)-(13) are nonlinear with respect to parameters $\{q_b, q(s,i|b), q(s,i|e), q_e\}$, they can be easily transformed into linear ones through the transformation

$$q(b,s,i) = q_b q(s,i|b),\ q(e,s,i) = q_e q(s,i|e). \tag{14}$$

Note that $q(b,s,i)$ is the probability that the PU is busy *and* SU s is selected for cooperation at power level i, while $q(e,s,i)$ is the probability that the PU is idle *and* SU s packets are transmitted in a slot at power level i. With this transformation, the constraints that characterize the achievable rate region of policies in $\mathcal{C}_0$ become,

$$\sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} r_p(s,i)\, q(b,s,i) = \lambda_p \tag{15}$$

$$\sum_{i\in\mathcal{I}_s} P_s(i)\, q(e,s,i) + \sum_{i\in\mathcal{I}_s} P_s(i)\, q(b,s,i) \le \hat{P}_s,\ \ s \in \mathcal{S} \tag{16}$$

$$\sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} q(e,s,i) + \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} q(b,s,i) = 1 \tag{17}$$

$$q(e,s,i) \ge 0\ q(b,s,i) \ge 0,\ \ s \in \mathcal{S},\ \ i \in \mathcal{I}_s^0. \tag{18}$$

In addition, the achievable rate of each SU $s \in \mathcal{S}$, given by (6), can be rewritten as

$$\bar{r}_s = \sum_{i \in \mathcal{I}_s} r_s(i)\, q(e, s, i) \tag{19}$$

In fact, it can shown that (6) and (8)-(13), define the same performance space as (15)-(19). This is described in the following proposition.

Proposition 1. *The performance space of $\{\bar{r}_s\}$ which is defined by Eqs. (6) and (8)-(13) is equivalent with the corresponding performance space defined by Eqs. (15)-(19).*

Proof: Please refer to Appendix A. ■

In the next section, we use the characterization of the achievable rate region of policies in $\mathcal{C}_0$ in terms of constraints (15)-(19) to show that this region coincides with the achievable rate region of policies in $\mathcal{C}_2$.

1) Stability region of PU Queue under the class of policies in $\mathcal{C}_0$: Based on the discussion above, the stability region of the PU queue under the class of policies in $\mathcal{C}_0$ is the set of λ_p for which there exists a set of probabilities $\{q(b, s, i),\ q(e, s, i)\}_{s \in \mathcal{S},\ i \in \mathcal{I}_s^0}$ that satisfy (15)-(19). Based on this observation we have the following corollary.

Corollary 2. *The stability region of the PU queue under the class of policies in $\mathcal{C}_0$ is the interval $[0, \hat{\lambda}]$ where $\hat{\lambda}$ is the resulting value of the objective of the following linear optimization problem in terms of $x(b, s, i)$, for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$.*

$$\text{maximize:} \quad \textstyle\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) x(b, s, i) \tag{20}$$

$$\text{subject to} \quad \textstyle\sum_{i \in \mathcal{I}_s} P_s(i)\, x(b, s, i) \leq \hat{P}_s,\ \ s \in \mathcal{S} \tag{21}$$

$$\textstyle\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} x(b, s, i) \leq 1 \tag{22}$$

$$x(b, s, i) \geq 0,\ \ s \in \mathcal{S},\ \ i \in \mathcal{I}_s^0 \tag{23}$$

Proof: Please refer to Appendix B. ■

Remark 3. It can be easily seen that the value of optimization problem in Corollary 2 does not change if inequality in (22) is replaced by equality. This implies what is intuitively expected, i.e., when $\lambda_p = \hat{\lambda}$, no idle slots are left by PU, i.e., $q_b = 1$ and $q_e = 0$, and *the available power from any SU is allocated only to the cooperation with the PU.*

2) Implementation of policies in class $\mathcal{C}_0$: In order to implement the policies in the proposed restricted class $\mathcal{C}_0$, the probabilities $\{q(e,s,i), q(b,s,i)\}_{i\in\mathcal{I}_s^0,\, s\in\mathcal{S}}$ need to be determined. These probabilities are obtained through solving the following optimization problem OPT0

$$\text{maximize} \quad f(\bar{\mathbf{r}}) \tag{24}$$

$$\text{subject to} \quad \sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s^0} r_p(s,i)\, q(b,s,i) = \lambda_p \tag{25}$$

$$\sum_{i\in\mathcal{I}_s} P_s(i)\, q(e,s,i) + \sum_{i\in\mathcal{I}_s} P_s(i)\, q(b,s,i) \leq \hat{P}_s,\ \ s\in\mathcal{S} \tag{26}$$

$$\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s^0} q(e,s,i) + \sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s^0} q(b,s,i) = 1 \tag{27}$$

$$q(e,s,i)\geq 0\ \ q(b,s,i)\geq 0,\ \ s\in\mathcal{S},\ \ i\in\mathcal{I}_s^0, \tag{28}$$

where $\bar{\mathbf{r}} \triangleq \{\bar{r}_s\}_{s\in\mathcal{S}}$, and $\bar{r}_s = \sum_{i\in\mathcal{I}_s} r_s(i)\, q(e,s,i)$. In problem OPT0 the optimization variables are $\{q(e,s,i), q(b,s,i)\}_{i\in\mathcal{I}_s^0,\, s\in\mathcal{S}}$, whereas $r_p(s,i)$, $r_s(i)$, $P_s(i)$, for all $i\in\mathcal{I}_s^0$, and $s\in\mathcal{S}$, are fixed system model parameters. Specifically, $r_p(s,i)$ denotes the probability of successful transmission of the PU packet when SU s cooperates at ith power level, while $r_s(i)$ denotes the probabilty of successful transmission of SU s packet, when SU s transmits at ith power level. $P_s(i)$ denotes the transmit power that corresponds to level $i\in\mathcal{I}_s^0$ that SU s uses in either case, and $\hat{P}_s$ denotes the maximum average transmit power available for SU s. Constraint (25) ensures that the average packet service rate of the PU queue equals its average input rate, λ_p, and, therefore, guarantees stability of the PU queue. The inequality constraints in (26) are the long-term average power constraints for all SUs. Finally, constraints (27) and (28) are imposed because the optimization variables $\{q(e,s,i), q(b,s,i)\}_{i\in\mathcal{I}_s^0,\, s\in\mathcal{S}}$ represent probabilities. In case where the selected objective function in (24), $f(\cdot)$, is a concave function of $\bar{\mathbf{r}}$, then, problem OPT0 is a convex optimization problem which can be solved efficiently via interior point methods. Once variables $\{q(e,s,i), q(b,s,i)\}_{i\in\mathcal{I}_s^0,\, s\in\mathcal{S}}$ are determined, we can obtain the probabilities $\{q_b, q(s,i\,|b), q(s,i\,|e), q_e\}_{s\in\mathcal{S},\ i\in\mathcal{I}_s^0}$ through the linear transformation in (14). Then, policies in $\mathcal{C}_0$ act as we describe in section II-C.

B. Achievable Rate Region of Policies in Class $\mathcal{C}_2$

Contrary to the available controls when the PU priority constraint is imposed, the set of available controls for policies in $\mathcal{C}_2$ does not obey the PU priority constraint (thus, a slot may be allocated to SU packet transmission, even if the PU queue is nonempty). Hence, this class of policies falls in the framework of policies studied in [16], whose achievable rate region can be characterized again by the achievable rate region of stationary policies. In the latter framework, a stationary policy selects at the beginning of each time slot the control (u, s, i) with probability $p\left(u, s, i\right)$. Under such a policy, the probability of successful transmission of SU s packets is

$$\bar{r}_s = \sum_{i \in \mathcal{I}_s} r_s\left(i\right) p\left(0, s, i\right), \tag{29}$$

while, the probability of successful transmission of PU packets is

$$\bar{r}_p = \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) p(1, s, i), \tag{30}$$

and stability of the PU queue requires that

$$\bar{r}_p \geq \lambda_p. \tag{31}$$

Also, the average power constraint requirement implies that

$$\sum_{i \in \mathcal{I}_s} P_s\left(i\right) p(0, s, i) + \sum_{i \in \mathcal{I}_s} P_s\left(i\right) p(1, s, i) \leq \hat{P}_s,\ s \in \mathcal{S}. \tag{32}$$

Finally, since $p\left(u, s, i\right)$ are probabilities, we must have

$$\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} p\left(0, s, i\right) + \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} p\left(1, s, i\right) = 1 \tag{33}$$

$$p(0, s, i) \geq 0,\ \ p(1, s, i) \geq 0,\ \ \text{s} \in \mathcal{S}, i \in \mathcal{I}_s^0. \tag{34}$$

Constraints (31)-(34) together with (29) define the achievable rate region $\mathcal{R}_2$ of policies in $\mathcal{C}_2$. The similarity of these constraints compared to those in (15)-(19) should be noted. From a math perspective, the only difference is that there exists equality in (15), as opposed to inequality in (31). However, there is difference in the interpretation of these probabilities. Specifically,

- $q\left(b, s, i\right)$ is the probability that PU queue is nonempty *and* SU s is selected for cooperation at ith power level, while $p(1, s, i)$ is the probability that SU s is selected for cooperation at ith power level and dictating PU transmission as well (irrespective of the PU queue size).
- $q\left(e, s, i\right)$ is the probability that PU queue is empty *and* secondary user s packets are transmitted in a slot at ith power level, while $p(0, s, i)$ is the probability of selecting secondary user s packet for transmission at the ith power level, while PU does not transmit (irrespective of the PU queue size).

As discussed earlier, since $\mathcal{C}_0 \subseteq \mathcal{C}_2$, $\mathcal{R}_0 \subseteq \mathcal{R}_2$. The next theorem shows that $\mathcal{R}_2 = \mathcal{R}_0$.

Theorem 4. *It holds $\mathcal{R}_2 \subseteq \mathcal{R}_0$, hence $\mathcal{R}_0 = \mathcal{R}_1 = \mathcal{R}_2$.*

Proof: Please refer to the Appendix C. ■

IV. Extensions to the basic model

In this section, we extend the model that has been investigated so far in two directions. First, we assume exogenous packet arrivals to the SU queues, instead of infinite queue backlogs. Second, imperfect channel sensing effects are taken into account.

A. Incorporating Exogenous Packet arrivals to SU queues

In this part, we investigate the scenario where packets arrive exogenously to SU queues. Specifically, we assume that at the beginning of slot t, $A_s(t)$ packets arrive to the queue of SU. Furthermore, for a given SU s, $A_s(t),\ t = 0, 1...$ are i.i.d random variables with $\mathbb{E}\left[A_s(t)\right] = \lambda_s,\ \mathbb{E}\left[\left(A_s\left(t\right)\right)^2\right] < \infty$ and the arrival processes $\{A_s(t)\}_{t=0}^{\infty},\ s \in \mathcal{S}$ are independent of each other. Regarding the packet arrival process to the PU queue, $A_p\left(t\right)$, we also assume that it consists of i.i.d. random variables and is independent of the arrival processes to the SU queues.

1) Admissible Policies : As in the case where the SU queues were backlogged, an admissible policy should satisfy the constraints described in section II-C. Regarding SU queues, there are no constraints on the rates of their arrival processes. Hence, depending on the arrival rates to these queues, they may be stable or unstable. To deal with the issue of instability, we assume that flow control is applied to each of the SU queues, which has the following form [16]: among the $A_s(t)$ packets that arrive at the queue of SU s, a number $B_s(t) \leq A_s(t)$ is accepted by the system and the rest (if any) are dropped. Thus, the *flow control objective* is that the SU queues with input the $B_s(t)$ packets must be mean rate stable.

In general, the admissible policies in this setup take control actions at time slot t, based on the history of the system up to time t, which includes queue sizes of the PU and SU queues up to time t. We call this class of policies $\tilde{\mathcal{C}}_1$. Similar to the previous analysis, we consider a subclass of policies in $\tilde{\mathcal{C}}_1$, denoted by $\tilde{\mathcal{C}}_0$, which consists of policies whose decisions are based solely on whether the PU queue is empty or not, hence not requiring information about the queue sizes at the PU and SU queues. In each time slot t, a policy in $\tilde{\mathcal{C}}_0$ acts as follows:

- *Flow control action*: Each of the $A_s(t)$ packets that arrive to SU s at time t, is admitted with probability p_s^a. The packet admission events are independent of each other and independent of other processes in the system.
- When $Q_p(t) > 0$, select a SU s to cooperate at ith power level with a probability $q\left(s, i\left|b\right.\right)$.
- When $Q_p(t) = 0$, select a SU s to transmit its own data at ith power level with probability $q\left(s, i\left|e\right.\right)$. If the selected SU has no data to transmit, it loses its transmission opportunity.

For performance comparison, we consider the extended class of policies $\tilde{\mathcal{C}}_2$ which employs flow control at the SU queues and obeys all constraints of policies in $\tilde{C}_1$, except the PU priority constraint. Hence we again have $\tilde{\mathcal{C}}_0 \subseteq \tilde{\mathcal{C}}_1 \subseteq \tilde{\mathcal{C}}_2$. The performance measure of interest in this case is the throughput of SU queues, i.e., the long term average number of packets per slot, R_s, that are delivered to the receiver of SU $s, \ s \in \mathcal{S}$. The set of achievable throughput vectors $\boldsymbol{R} = \{R_s\}_{s\in\mathcal{S}}$ under class of policies $\tilde{\mathcal{C}}_i, \ i = 0, 1, 2$, is denoted by $\tilde{\mathcal{R}}_i$. Since $\tilde{\mathcal{C}}_0 \subseteq \tilde{\mathcal{C}}_1 \subseteq \tilde{\mathcal{C}}_2$ we again have, $\tilde{\mathcal{R}}_0 \subseteq \tilde{\mathcal{R}}_1 \subseteq \tilde{\mathcal{R}}_2$.

2) Throughput Regions of Policies in Classes $\tilde{\mathcal{C}}_0$ and $\tilde{\mathcal{C}}_2$: Similarly to the analysis in Section III-A, it can be shown that $\tilde{\mathcal{R}}_0$ consists of all vectors $\boldsymbol{R} = \{R_s\}_{s\in\mathcal{S}}$ that satisfy

$$R_s \leq \min\left\{\lambda_s, \bar{r}_s\right\}, \ s \in \mathcal{S} \tag{35}$$

where $\bar{r}_s$ is defined by (15)-(18) and (19). Note that in the current setup, $\bar{r}_s$ represents the "offered" service rate to SU s queue, i.e., the probability of successful transmission of an SU s packet. For maximizing the throughput of each SU queue, we must have $R_s = \min\left\{\lambda_s, \bar{r}_s\right\}$. Moreover, since flow control is chosen to stabilize the SU queues, we must have $R_s = \lambda_s p_s^a$, with $p_s^a = \frac{\min\{\lambda_s, \bar{r}_s\}}{\lambda_s}, \ s \in \mathcal{S}$.

On the other hand, for the stationary policies in $\tilde{\mathcal{C}}_2$, it can be shown [16] that $\tilde{\mathcal{R}}_2$ consists of all vectors that satisfy (35) and $\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s^0} r_p\left(s,i\right)q\left(b,s,i\right) \geq \lambda_p$, with $\bar{r}_s$ being defined by (16)-(18) and (19).

Based on the structure of the throughput regions described above, it follows by a similar argument as in section III that $\tilde{\mathcal{R}}_0 = \tilde{\mathcal{R}}_2$, which implies again that policies in $\tilde{\mathcal{C}}_0$ can achieve any throughput vector achievable by the less restrictive policies in $\tilde{\mathcal{C}}_2$.

3) Selecting Optimal Policies in $\tilde{\mathcal{C}}_0$: Consider the problem of selecting a policy in $\tilde{\mathcal{C}}_0$ that maximizes $f(\boldsymbol{R})$, with $\boldsymbol{R} \in \tilde{\mathcal{R}}_0$. Based on the above, it is then easy to see that this optimization problem is equivalent to

$$\text{maximize } f\left(\left\{\min\left(\lambda_s, \bar{r}_s\right)\right\}_{s\in\mathcal{S}}\right), \tag{36}$$

where $\bar{r}_s$ is defined by (19) and (15)-(18).

B. Imperfect Sensing

In this part, we investigate the effects of imperfect sensing on the mode of operation and the performance of policies in $\mathcal{C}_0$. For simplicity we assume that the SUs are infinitely backlogged. The case where packets arrive randomly at the SUs can be handled in a similar fashion as in section IV-A.

We assume that cooperative sensing takes place, so that all SUs make the same decision at each slot as to whether the primary channel is busy or idle. We assume that PU channel sensing events are independent across slots and independent of the transmission choices of the users. We denote the probabilities of detection and false alarm of the sensing mechanism as $\mathcal{P}_D = \Pr\{\text{sense busy}|\text{channel is busy}\}$ and $\mathcal{P}_F = \Pr\{\text{sense busy}|\text{channel is idle}\}$, respectively. Two sources of error and inefficiency may occur in this situation:

- The primary channel is busy but sensed idle (an event occurring with probability $1-\mathcal{P}_D$). We distinguish two subcases:
  - One of the SUs transmits its own packet at the same slot with the PU, an event with probability $1-\sum_{s\in\mathcal{S}} q\left(s,0\left|e\right.\right)$[2]. In this case, collision occurs and both transmissions fail.
  - No SU transmits a packet, an event with probability $\sum_{s\in\mathcal{S}} q\left(s,0\left|e\right.\right)$. In this case the PU transmission is successful with probability $r_p\left(0\right)$.

  The effect of this error on the probability of successful transmission of PU packet is given by

$$\bar{r}_p = \left(1-\mathcal{P}_D\right)\sum_{s\in\mathcal{S}} q\left(s,0\left|e\right.\right) r_p\left(0\right) + \mathcal{P}_D \sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s^0} r_p\left(s,i\right) q\left(s,i\left|b\right.\right) \tag{37}$$

- When the PU channel is idle but it is sensed busy, an SU may be allocated for cooperation with the PU, thus losing the opportunity to transmit its own data. Hence, the probability of successful transmission of SU packets is affected by the probability of the event that the

[2]Recall using power level 0 implies no transmission.

PU channel is idle and sensed idle (equal to $q_e\left(1-\mathcal{P}_F\right)$). For the SU s, this probability becomes

$$\bar{r}_s = q_e\left(1-\mathcal{P}_F\right)\sum_{i\in\mathcal{I}_s^0} r_s\left(s,i\right)q\left(s,i\left|e\right.\right). \tag{38}$$

Regarding the average power consumed by SU s under a policy in $\mathcal{C}_0$, we consider the following events:

1) The event that PU channel is busy and is sensed busy, with probability $q_b\mathcal{P}_D$. Then, SU s consumes an average power of $\sum_{i\in\mathcal{I}_s^0} P_s\left(i\right)q\left(s,i\left|b\right.\right).$
2) The event that PU channel is busy and is sensed idle, with probability $q_b\left(1-\mathcal{P}_D\right).$ Then, SU s consumes an average power of $\sum_{i\in\mathcal{I}_s^0} P_s\left(i\right)q\left(s,i\left|e\right.\right).$
3) The event that PU channel is idle and is sensed idle, with probability $q_e\left(1-\mathcal{P}_F\right)$. Then, SU s consumes an average power of $\sum_{i\in\mathcal{I}_s^0} P_s\left(i\right)q\left(s,i\left|e\right.\right).$
4) The event that PU channel is idle and is sensed busy, with probability $q_e\mathcal{P}_F$. Then, SU s consumes an average power of $\sum_{i\in\mathcal{I}_s^0} P_s\left(i\right)q\left(s,i\left|b\right.\right).$

Based on the above, the new performance space when channel sensing errors are introduced is determined by (10)-(13) and

$$q_b\mathcal{P}_D\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s^0} r_p\left(s,i\right)q\left(s,i\left|b\right.\right) + q_b\left(1-\mathcal{P}_D\right)r_p\left(0\right)\sum_{s\in\mathcal{S}} q\left(s,0\left|e\right.\right) = \lambda_p. \tag{39}$$

$$\left(q_b\mathcal{P}_D + q_e\mathcal{P}_F\right)\sum_{i\in\mathcal{I}_s^0} P_s\left(i\right)q\left(s,i\left|b\right.\right) + \left(1-q_b\mathcal{P}_D-q_e\mathcal{P}_F\right)\sum_{i\in\mathcal{I}_s^0} P_s\left(i\right)q\left(s,i\left|e\right.\right) \le \hat{P}_s,\ s\in\mathcal{S}. \tag{40}$$

We seek transmission policies that achieve the following objective, OPT1:

$$\text{maximize} \quad f\left(\bar{r}_s\right) \tag{41}$$

$$\text{subject to} \quad \text{(10)-(13), (39)-(40)} \tag{42}$$

where $\bar{r}_s$ are given by (38).

Due to (39)-(40), OPT1 is a non-convex optimization problem and therefore it is difficult to be solved optimally. One way to solve OPT1 numerically, is to fix q_b, in which case the constraints become linear and the problem can be easily solved. Let $g(q_b)$ be the maximum value of the objective of OPT1 for $q_b \in [0,1]$ (for some values of q_b the problem may be infeasible). We can then solve the one-dimensional problem:

$$\text{maximize } g(q_b) \tag{43}$$

where $0 \leq q_b \leq 1$ and the maximum can be specified through exhaustive linear search methods. However, based on the following remark, we can restrict the region of possible q_b values, where linear search is performed.

Proposition 5. *The probability of PU being busy when imperfect sensing takes place, varies within*

$$\frac{\lambda_p}{\mathcal{P}_D r_{p,max} + (1-\mathcal{P}_D)\, r_p(0)} \leq q_b \leq \min\left\{\frac{\lambda_p}{\mathcal{P}_D r_p(0)}, 1\right\} \tag{44}$$

where $r_{p,max} = \max_{s,i}\{r_p(s,i)\}$.

Proof: The proof follows straightforwardly based on (39) and is given in Appendix D. ■

Solving the one-dimensional problem (43) by exhaustive search may be computationally expensive. As will be seen in section VI, a large number of numerical investigations suggest that $g(q_b)$ is a concave function of q_b. We have not been able to prove rigorously that this property holds. However, if it is indeed true, binary search methods can be used instead for the solution of (43), thus reducing the computational complexity from $\mathcal{M}$ to $\log_2 \mathcal{M}$, where $\mathcal{M}$ stands for the number of values of q_b investigated in the space given by (44).

V. Distributed Implementation

In this section, we assume perfect PU channel sensing and infinitely backlogged SUs, and focus on approaches based on policies in $\mathcal{C}_0$ that do not rely on central coordination in order to achieve the following objective, OPT2:

$$\begin{aligned} &\text{maximize} && \textstyle\sum_{s\in\mathcal{S}} f_s(\bar{r}_s) \\ &\text{subject to} && \text{(15), (16), (17), (18) and (19)} \end{aligned} \tag{45}$$

Functions $\{f_s(\cdot)\}_{s\in\mathcal{S}}$ are usually selected so that certain fairness criteria for SU rate allocation are satisfied, see [17] and [18], and they are assumed to be concave with respect to $\bar{r}_s$. Thus, due to the fact that for all $s \in \mathcal{S}$, $\bar{r}_s$ is a linear function of variables $\left\{\{q(e,s,i)\}_{i\in\mathcal{I}_s^0}, \{q(b,s,i)\}_{i\in\mathcal{I}_s^0}\right\}$, $f_s(\bar{r}_s)$ is also a concave function of these variables. Hence, OPT2 is a convex optimization problem and can be solved efficiently via interior point methods.

In an operational environment where parameters may change with time, problem OPT2 will have to be solved whenever significant changes to such parameters occur. A centralized solution requires a single node to be responsible for gathering instantaneous parameter values, for the solution of OPT2 and for determining the appropriate scheduling of packet transmissions. While such a solution may be acceptable in certain environments, it creates a "single point of failure". Moreover the central node must be continually informing the SUs as to which one will cooperate or transmit in each time slot and at which power level. There may also be a scalability issue with this approach since the number of variables is of the order $2\left|\mathcal{S}\right|I$, where I is the maximum number of power levels of SU nodes ($\sum_{i\in\mathcal{S}}\left|\mathcal{I}_s^0\right|$ parameters $\left\{q\left(b,s,i\right)\right\}_{s\in\mathcal{S},\ i\in\mathcal{I}_s^0}$ plus $\sum_{i\in\mathcal{S}}\left|\mathcal{I}_s^0\right|$ parameters $\left\{q\left(e,s,i\right)\right\}_{s\in\mathcal{S},\ i\in\mathcal{I}_s^0}$). Hence, depending on the computing power and memory availability at the central node, solving problem OPT2 in a centralized location may become prohibitive for larger number of SUs.

1) Advantages of the Distributed Approach: In this section, we derive a solution to OPT2 in a distributed fashion. The main features of our approach are the following.

a) The PU involvement in the algorithm is only to announce its arrival rate λ_p at the beginning of the algorithm - no further participation is required.

b) A SU node does not need to know the parameters (i.e., $r_s\left(i\right)$, $r_p\left(s,i\right)$, $i\in\mathcal{I}_s$) of other SU nodes.

c) The distributed solution requires each SU node $s\in\mathcal{S}$ to solve optimization problems with $\left|\mathcal{I}_s^0\right|$ variables, hence the computational complexity per node does not increase with the number of SU nodes.

d) Two messages are broadcasted by each SU node per iteration of the distributed algorithm. The number of iterations for convergence depends on the number of SU nodes, but this is tolerable for the algorithm execution in a real-time setting.

e) Once convergence of the algorithm is reached for a given arrival rate, the SUs need only observe the state of the PU channel (busy or idle); they can decide autonomously which SU node is scheduled to either cooperate with the PU, or to transmit its own traffic, without the need of a scheduler, or the exchange of control messages.

We assume that there is a separate low-rate channel which is used by the SUs for control message exchanges [19]. In particular we assume that control messages may be broadcasted among the SUs, either because the low-rate channel is broadcast in nature, or through the establishment of Broadcast Trees that usually are employed in ad-hoc networks [20], [21].

2) Implementation of the Distributed Optimization Algorithm: Towards a distributed solution to problem OPT2 we would ideally like to decompose the global problem into $\left|\mathcal{S}\right|$ parallel subproblems, each one involving only local variables and parameters of node s. Among all alternatives we tried towards this end, the best algorithm in terms of convergence was the one built upon the *Alternating Direction Method of Multipliers* (ADMoM), which has superior convergence properties over the traditional dual ascent method [22], [23], [24]. To apply ADMoM to OPT2, we first turned the average power inequality constraints (16) into equalities, by introducing auxiliary variables $\{y_s\}_{s\in\mathcal{S}}$, where y_s is associated with the respective s^{th} constraint, and is positive-valued. Also, for notational simplicity, we equivalently rewrite problem OPT2 as OPT3 given by

$$\text{minimize} \quad -\sum_{s\in\mathcal{S}} f_s\left(\phi_s\left(\boldsymbol{x}_s\right)\right) \tag{46}$$

$$\text{subject to} \quad \sum_{s\in\mathcal{S}} g_{1s}\left(\boldsymbol{z}_s\right) = \lambda_p \tag{47}$$

$$h_s\left(\boldsymbol{x}_s, \boldsymbol{z}_s, y_s\right) = \hat{P}_s,\ s\in\mathcal{S} \tag{48}$$

$$\sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{x}_s\right) + \sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{z}_s\right) = 1 \tag{49}$$

$$\boldsymbol{x}_s \geq 0, \boldsymbol{z}_s \geq 0, y_s \geq 0,\ s\in\mathcal{S} \tag{50}$$

where we use the variables $\boldsymbol{x}_s \triangleq \{q\left(e,s,i\right)\}_{i\in\mathcal{I}_s^0}$, $\boldsymbol{z}_s \triangleq \{q\left(b,s,i\right)\}_{i\in\mathcal{I}_s^0}$, and we also define the following functions: $\phi_s\left(\boldsymbol{x}_s\right) \triangleq \sum_{i\in\mathcal{I}_s} r_s\left(i\right) q\left(e,s,i\right)$, $g_{1s}\left(\boldsymbol{z}_s\right) \triangleq \sum_{i\in\mathcal{I}_s^0} r_p\left(s,i\right) q\left(b,s,i\right)$, $g_{2s}\left(\boldsymbol{x}_s\right) \triangleq \mathbf{1}^T\boldsymbol{x}_s = \sum_{i\in\mathcal{I}_s^0} q\left(e,s,i\right)$, $g_{2s}\left(\boldsymbol{z}_s\right) \triangleq \mathbf{1}^T\boldsymbol{z}_s = \sum_{i\in\mathcal{I}_s^0} q\left(b,s,i\right)$, and

$$h_s\left(\boldsymbol{x}_s, \boldsymbol{z}_s, y_s\right) \triangleq \sum_{i\in\mathcal{I}_s} P_s(i) q(e,s,i) + \sum_{i\in\mathcal{I}_s} P_s(i) q(b,s,i) + y_s,\ s\in\mathcal{S}. \tag{51}$$

Let ν and ξ denote the dual variables associated with the constraints of (47) and (49) respectively, and μ_s the dual variable associated with the s^{th} constraint of (48). Then, the augmented Lagrange function corresponding to OPT3 used by ADMoM, parametrized by the penalty parameter $\rho > 0$, is given by [22], [23]

$$\begin{aligned} L_p &= \sum_{s\in\mathcal{S}} L_s - \nu\lambda_p - \xi + \frac{\rho}{2}\left\{\left(\sum_{s\in\mathcal{S}} g_{1s}\left(\boldsymbol{z}_s\right) - \lambda_p\right)^2\right. \\ &+ \left.\sum_{s\in\mathcal{S}}\left(h_s\left(\boldsymbol{x}_s, \boldsymbol{z}_s, y_s\right) - \hat{P}_s\right)^2 + \left(\sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{x}_s\right) + \sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{z}_s\right) - 1\right)^2\right\} \end{aligned} \tag{52}$$

with

$$L_s \triangleq -f_s\left(\phi_s\left(\boldsymbol{x}_s\right)\right) + \nu g_{1s}\left(\boldsymbol{z}_s\right) + \mu_s\left(h_s\left(\boldsymbol{x}_s, \boldsymbol{z}_s, y_s\right) - \hat{P}_s\right) + \xi g_{2s}\left(\boldsymbol{x}_s\right) + \xi g_{2s}\left(\boldsymbol{z}_s\right),\ s\in\mathcal{S}. \tag{53}$$

Computational complexity: The optimization steps and variables updates that need to be carried out at each SU node $s \in \mathcal{S}$, according to ADMoM, are given by

$$\begin{aligned} \boldsymbol{x}_s^{k+1} &= \arg\min_{\boldsymbol{x}_s \geq 0} L_s\left(\boldsymbol{x}_s, \boldsymbol{z}_s^k, y_s^k, v^k, \xi^k, \mu_s^k\right) + \frac{\rho}{2}\left(h_s\left(\boldsymbol{x}_s, \boldsymbol{z}_s^k, y_s^k\right) - \hat{P}_s\right)^2 \\ &+ \frac{\rho}{2}\left(\sum_{m=1}^{s-1} g_{2m}\left(\boldsymbol{x}_m^{k+1}\right) + \sum_{m=s+1}^{|\mathcal{S}|} g_{2m}\left(\boldsymbol{x}_m^k\right) + g_{2s}\left(\boldsymbol{x}_s\right) + \sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{z}_s^k\right) - 1\right)^2, \end{aligned} \tag{54}$$

$$\begin{aligned} \boldsymbol{z}_s^{k+1} &= \arg\min_{\boldsymbol{z}_s \geq 0} L_s\left(\boldsymbol{x}_s^{k+1}, \boldsymbol{z}_s, y_s^k, v^k, \xi^k, \mu_s^k\right) + \frac{\rho}{2}\left(h_s\left(\boldsymbol{x}_s^{k+1}, \boldsymbol{z}_s, y_s^k\right) - \hat{P}_s\right)^2 \\ &+ \frac{\rho}{2}\left(\sum_{m=1}^{s-1} g_{1m}\left(\boldsymbol{z}_m^{k+1}\right) + \sum_{m=s+1}^{|\mathcal{S}|} g_{1m}\left(\boldsymbol{z}_m^k\right) + g_{1s}\left(\boldsymbol{z}_s\right) - \lambda_p\right)^2 \\ &+ \frac{\rho}{2}\left(\sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{x}_s^{k+1}\right) + \sum_{m=1}^{s-1} g_{2m}\left(\boldsymbol{z}_m^{k+1}\right) + \sum_{m=s+1}^{|\mathcal{S}|} g_{2m}\left(\boldsymbol{z}_m^k\right) + g_{2s}\left(\boldsymbol{z}_s\right) - 1\right)^2, \end{aligned} \tag{55}$$

$$y_s^{k+1} = \arg\min_{y_s \geq 0} L_s\left(\boldsymbol{x}_s^{k+1}, \boldsymbol{z}_s^{k+1}, y_s, v^k, \xi^k, \mu_s^k\right) + \frac{\rho}{2}\left(h_s\left(\boldsymbol{x}_s^{k+1}, \boldsymbol{z}_s^{k+1}, y_s\right) - \hat{P}_s\right)^2, \tag{56}$$

$$\xi^{k+1} = \xi^k + \rho\left(\sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{x}_s^{k+1}\right) + \sum_{s\in\mathcal{S}} g_{2s}\left(\boldsymbol{z}_s^{k+1}\right) - 1\right), \tag{57}$$

$$\nu^{k+1} = \nu^k + \rho\left(\sum_{s\in\mathcal{S}} g_{1s}\left(\boldsymbol{z}_s^{k+1}\right) - \lambda_p\right), \tag{58}$$

$$\mu_s^{k+1} = \mu_s^k + \rho\left(h_s\left(\boldsymbol{x}_s^{k+1}, \boldsymbol{z}_s^{k+1}, y_s^{k+1}\right) - \hat{P}_s\right), \tag{59}$$

where k denotes the iteration index. Note that the computational burden is distributed across SU nodes; the computational complexity at each node depends primarily on the two quadratic optimization problems in (54) and (55), each of which has $|\mathcal{I}_s^0|$ variables, and can be efficiently solved via interior point methods, or standard methods such as Newton Method. All the following steps involve a single variable and are straightforward.

Communication overhead: Each node s, in order to perform the steps in (54) and (55), needs to know information concerning the updated local variables of other nodes. This can be accomplished through message broadcasts by each SU node via the control channel in the following manner. The nodes update their local variables and broadcast the messages required sequentially, in a prespecified order. Specifically, for the step in (54), each node $s \in \mathcal{S}$ updates its primal variable $\boldsymbol{x}_s^{k+1}$ and broadcasts message $g_{2s}\left(\boldsymbol{x}_s^{k+1}\right)$. Similarly, for the step in (55), each SU node updates its variable $\boldsymbol{z}_s^{k+1}$and broadcasts $g_{1s}\left(\boldsymbol{z}_s^{k+1}\right)$ and $g_{2s}\left(\boldsymbol{z}_s^{k+1}\right)$ in one message, according to the prespecified order. Steps dictated by (56)-(59), for each node s, require only its local variables and information that is already acquired by s from the previous message broadcasts and thus can be implemented in parallel by all nodes. Each iteration of the distributed algorithm consists of one round of these update steps by all $|\mathcal{S}|$ nodes. Consequently, the communication overhead of the algorithm is $2\,|\mathcal{S}|$ message broadcasts per iteration.

Convergence: For the convergence of the algorithm in decentralized manner, each SU keeps track of a local metric and determines local convergence with respect to it, within a prespecified accuracy. This local metric for each node $s \in \mathcal{S}$ may be the the successive differences of its local objective function under optimization, i.e., $f_s\left(\boldsymbol{x}_s\right)$. Once this local metric drops under the prespecified accuracy, local convergence is declared, and node s announces it via the control channel. As soon as all SU nodes reach convergence, the algorithm terminates.

Real-time implementation: We assume that the PU broadcasts its average arrival rate λ_p at the beginning of the algorithm. Once convergence of the algorithm for a given λ_p is reached, all SUs have knowledge of the sums of probabilities $g_{2s}\left(\boldsymbol{x}_s^{opt}\right), g_{2s}\left(\boldsymbol{z}_s^{opt}\right), \forall s \in \mathcal{S}$. Thus, if the SUs use the same randomization algorithm and common seed, as long as they observe the state of the PU channel, they can all independently produce the same result as to who SU is scheduled to cooperate with the PU or transmit its own data in every time slot. Then, the scheduled SU determines its power level for its transmission based on its own probability parameters. The system evolves without the need for further coordination among network nodes.

The algorithm runs again only when some of the parameters of the operational environment change significantly. Thus, when the arrival rate changes within a pre-specified percentage of its previous value, the PU informs the SUs about the new value of λ_p. Also, in case wireless channel gains change for some SU within a certain percentage, the corresponding SU may announce the rerun of the algorithm. In such cases the algorithm can adapt to changes in the operational environment; the problem is not solved from scratch, but the algorithm is initialized at the optimal point of the previous system state. This speeds up its convergence and reduces the overall communication overhead, as will be shown in the simulation results that follow.

Exogenous Packet arrivals to SU queues: In case of this scenario, we seek a decentralized solution to the optimization problem (36) according to subsection IV-A. However, if $f(\boldsymbol{R})$ is separable, i.e., $f(\boldsymbol{R}) = \sum_{s\in\mathcal{S}} f_s\left(R_S\right)$, then problem in (36) is essentially identical to the one in (45) where we replace $f_s(\bar{r}_s)$ with $f_s\left(\min\{\lambda_s, \bar{r}_s\}\right)$. We can therefore employ ADMoM using the same techniques as previously to provide a distributed implementation of the current optimization problem. Note that the fact that in the distributed implementation only SU s needs to know $f_s\left(\min\left\{\lambda_s, r_s\right\}\right)$, implies that each SU needs to know only its arrival rate in order to implement the distributed algorithm.

VI. Simulation and Numerical Results

In this section, we confirm the optimality claims in terms of performance for the proposed class of policies through several simulation experiments for different scenarios. First, we assume that SUs are infinitely backlogged and spectrum sensing is perfect. In this scenario, the performance of an optimal policy in $\mathcal{C}_0$ is compared to the transmission algorithm presented in [14] and an optimal dynamic policy from $\mathcal{C}_2$, constructed through the Lyapunov optimization techniques [16]. Furthermore, the convergence of the distributed algorithm, as well as its ability to adapt to changing parameters is studied. Next, we consider exogenous packet arrivals to SUs queues and the performance of an optimal policy in the proposed class $\tilde{\mathcal{C}}_0$ is presented. Finally, imperfect spectrum sensing is assumed and the convexity of the resulting optimization problem is investigated. In all the above scenarios, we consider a system model with one PU and several SUs, and as objective optimization function $f\left(\cdot\right)$ the sum of transmission rates of the SUs, i.e., $f\left(\bar{\boldsymbol{r}}\right) = \sum_{s\in\mathbb{S}} \bar{r}_s$.

Assuming perfect sensing and infinitely backlogged SUs, the performance of a setup which consists of 5 SUs and a set of 5 available transmit power levels is investigated in Figs. 3-4, in terms of $f\left(\bar{\mathbf{r}}\right)$ and average backlog of PU queue. Specifically, we assume for this setup that $\mathcal{I}_s^0 = \{0,1,2,3,4\}$, $\mathbf{P}_s = \{0, 0.25, 0.5, 0.75, 1\}$, $r_p\left(0\right) = 0.4$, $r_p\left(s,1\right) = 0.5$, $r_p\left(s,2\right) = 0.6$, $r_p\left(s,3\right) = 0.7$, $r_p\left(s,4\right) = 0.8$, $r_s\left(1\right) = 0.3$, $r_s\left(2\right) = 0.5$, $r_s\left(3\right) = 0.8$, $r_s\left(4\right) = 1$, and the average power constraint is $\hat{P}_s = 0.15$, for all $s \in \mathcal{S}$. It can be seen in Fig. 3 that the sum rate achieved by SUs that employ an optimal policy from the restricted class of policies $\mathcal{C}_0$ is identical to the sum rate achieved under the optimal policy in $\mathcal{C}_2$. This is in accordance with the main result of Theorem 1. Additionally, as it is illustrated by Fig. 4, the average backlog of the PU queue remains very low under the optimal policy in $\mathcal{C}_0$.

On the contrary, the dynamic policy from $\mathcal{C}_2$ induces large sizes to PU queue even for small

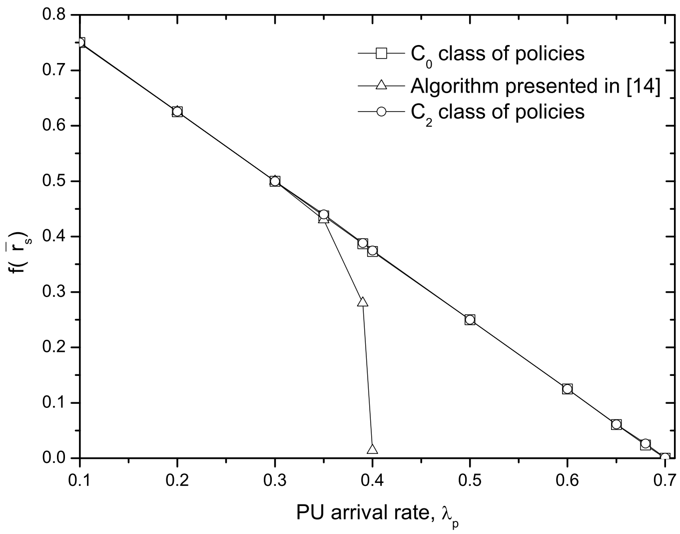

Fig. 3. The SU throughput utility function.

arrival rates. Moreover, when compared to the control algorithm presented in [14], the class $\mathcal{C}_0$ of policies extends the range of λ_p that can be supported by the system, *providing mutual benefits to both PU and SUs out of their cooperation*. In particular, transmission rates higher than the PU queue service rate without SU cooperation can be supported for the PU through the class of policies $\mathcal{C}_0$, while transmission opportunities are provided to SUs to transmit their own data. It should be noted that the policy in [14] was shown to be optimal for $\lambda_p < 0.4$, and this is confirmed in Fig. 3, where it is shown that all three policies achieve the same sum-rate for $\lambda_p < 0.4$. However, the policy in [14] renders the PU queue unstable for $\lambda_p > 0.4$ and reduces the SU sum rates to zero. The reason is the following. In [14], decisions are taken at the end of busy periods of the PU queue. If $\lambda_p > 0.4$, whenever a decision not to cooperate is taken, there is a nonzero probability that the primary queue never becomes empty, and hence there is no possibility for the SUs to take corrective actions.

For the same scenario and system setup, we also evaluate the performance of the proposed distributed algorithm. Regarding the distributed implementation parameters, we set the desired accuracy for convergence equal to $\epsilon = 10^{-5}$, while the penalty parameter is taken to be $\rho = 0.1$. For the arbitrary initialization of the algorithm, we used $\left\{q\left(e,s,i\right)^0\right\}_{i\in\mathcal{I}_s^0} = 0.01, \forall s \in \mathcal{S}$, $\left\{q\left(b,s,i\right)^0\right\}_{i\in\mathcal{I}_s^0} = 0.03, \forall s \in \mathcal{S}$, $\{\mu_s^0\}_{s\in\mathcal{S}} = 1$, $\xi^0 = 1$, $\nu^0 = 1$. The distributed algorithm was tested against the centralized solution to problem OPT2, in terms of the value of the objective, and

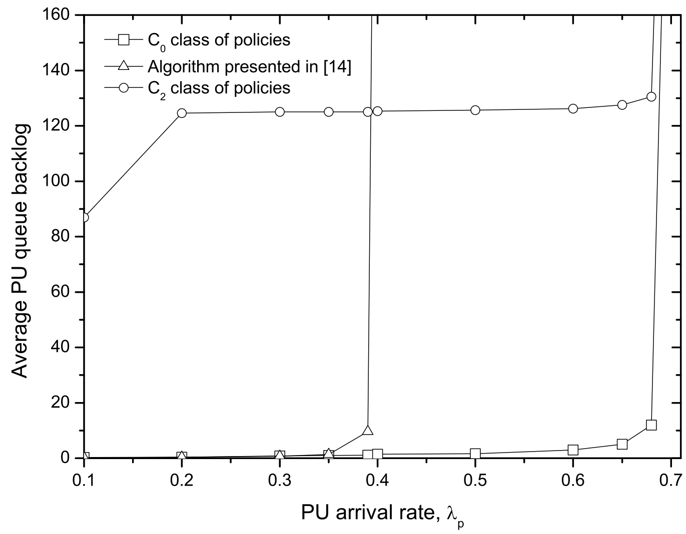

Fig. 4. The average backlog of the PU queue.

for various values of the PU arrival rate λ_p. It was observed that the numerical results obtained from both centralized and distributed implementations were identical (equal with those provided by Fig. 3); this shows that our proposed algorithm keeps up with its centralized counterpart, which can be justified by the convergence properties of ADMoM. Regarding the convergence speed, the number of iterations required for convergence within the given accuracy are given in Table I, when the arrival rate λ_p is varied inside the stability region and the proposed algorithm begins from scratch (arbitrary initialization). Obviously the algorithm is efficient enough, since it converges within a tolerable number of iterations for low PU transmission rates, while the convergence is even faster at higher ones. This can be explained by the fact that as λ_p increases, the constraints in (47)-(50) get tighter, restricting the feasibility set of the problem variables $\{\boldsymbol{x}_s, \boldsymbol{z}_s, y_s\}_{s \in \mathcal{S}}$. Consequently, since the distributed algorithm searches for the optimal solution within the feasibility set in each case of λ_p, it needs more iterations to converge when searching within a wider set than when searching within a narrower set. Finally, the adaptivity of the distributed algorithm to changes in the arrival rate λ_p, is investigated in Table II. In particular, we begin with an initial rate equal to $\lambda_p^0 = 0.5$, and run the algorithm from scratch, as described above. For all values of λ_p different from λ_p^0, we use as initialization for the algorithm the optimal point found at λ_p^0, and write down the number of iterations required for convergence within the given accuracy. Clearly, there is a remarkable reduction in the total number of iterations required

TABLE I

NUMBER OF ITERATIONS FOR THE DISTRIBUTED ALGORITHM.

λ_p	0.2	0.3	0.4	0.5	0.6	0.7
$\sharp$ of iterations	263	172	129	119	105	74

TABLE II

NUMBER OF ITERATIONS AS PU RATE CHANGES FROM $\lambda_p^0 = 0.5$ TO λ_p.

λ_p	0.35	0.4	0.45	0.52	0.55	0.6	0.7
$\sharp$ of iterations	44	34	39	29	39	45	16

for convergence compared with the arbitrary initialization.

Next, we additionally consider exogenous SU packet arrivals to the to the system setup described above. For this scenario, the throughput performance of the optimal policies in class $\tilde{\mathcal{C}}_0$ is investigated for the cases where $\sum_{s\in\mathcal{S}} \lambda_s$ is either well within or outside the achievable rate region $\mathcal{R}_0$, for both centralized and distributed implementations. Specifically, we initially fix $\sum_{s\in\mathcal{S}} \lambda_s$ inside the achievable rate region for each case of λ_p considered; λ_p varies in the range $[0.2, \ldots, 0.7]$, while $\sum_{s\in\mathcal{S}} \lambda_s$ is fixed equal to 0.05, where $\lambda_s = 0.01$, for all $s \in \mathcal{S}$. It was observed that the optimization objective values attained from both implementations are identical and equal to $\sum_{s\in\mathcal{S}} \lambda_s$, for each value of λ_p. Secondly, we consider $\sum_{s\in\mathcal{S}} \lambda_s$ outside the achievable rate region $\mathcal{R}_0$ for each value of the PU arrival rate λ_p; λ_p varies in the range $[0.2, \ldots, 0.7]$, while $\sum_{s\in\mathcal{S}} \lambda_s$ is fixed and equal to 1, where $\lambda_s = 0.2$, for all $s \in \mathcal{S}$. It was observed that the respective throughput utility that results from both centralized and distributed implementations coincide and are equal with the corresponding results when the SU queues are infinitely backlogged (provided by Fig. 3). Hence, the optimal policies in class $\tilde{\mathcal{C}}_0$ achieve the maximum possible value for the SU throughput utility function. The number of iterations required for the convergence of the distributed algorithm is shown in Tables III and IV. For the derivation of these results, an accuracy of $\epsilon = 10^{-5}$ is assumed and the distributed algorithm runs from scratch for each value of λ_p considered, while using the same initialization values for its variables as those used in the simulation experiments concerning the first scenario. The distributed algorithm converges again within a tolerable number of iterations.

TABLE III

NUMBER OF ITERATIONS FOR THE DISTRIBUTED ALGORITHM, $\sum_{s\in\mathcal{S}} \lambda_s = 0.05$.

λ_p	0.2	0.3	0.4	0.5	0.6	0.7
$\sharp$ of iterations	93	89	95	137	301	227

TABLE IV

NUMBER OF ITERATIONS FOR THE DISTRIBUTED ALGORITHM, $\sum_{s\in\mathcal{S}} \lambda_s = 1$.

λ_p	0.2	0.3	0.4	0.5	0.6	0.7
$\sharp$ of iterations	268	127	136	116	103	72

Finally, the effects of imperfect spectrum sensing are investigated in Fig. 5. Specifically, assuming the same system setup and $\lambda_p = 0.3$, we solve numerically OPT1, by fixing q_b and calculating the maximum value of the objective of OPT1 $g(q_b)$ when $q_b \in [0, 1]$, for various values of $\mathcal{P}_D$ and $\mathcal{P}_F$. It can be observed that q_b takes values only on the interval specified by the proposition 5, for all values of $\mathcal{P}_D$ and $\mathcal{P}_F$ considered; thus, restricting the region of q_b where exhaustive linear search methods have to search. Furthermore, when investigating the concavity of $g(q_b)$, simulation results indicate that $g(q_b)$ is concave with respect to q_b, irrespective of the values of $\mathcal{P}_D$ and $\mathcal{P}_F$ considered. As discussed in section IV-B, if this property is true in general, then the computational complexity of the centralized solution, as well as the computational complexity and overhead of a potential distributed implementation, can be significantly reduced.

VII. CONCLUSIONS

In this work we propose and investigate novel primary-secondary user cooperation policies for cognitive radio networks that orchestrate a PU and co-existing SUs in a wireless channel. The key idea is that SUs increase the service rate of the PU queue and therefore they increase the range of arrival rate of the PU for which its queue is stable. At the same time, the PU queue empties more often, and therefore the channel becomes idle more often, thus giving to SUs more transmission opportunities. Our major contribution to the state of the art is the proposition of policies that require only the state of PU channel (busy or empty) for their implementation, yet: 1) they achieve substantial augmentation of the stability region of the PU queue, and 2) they can obtain

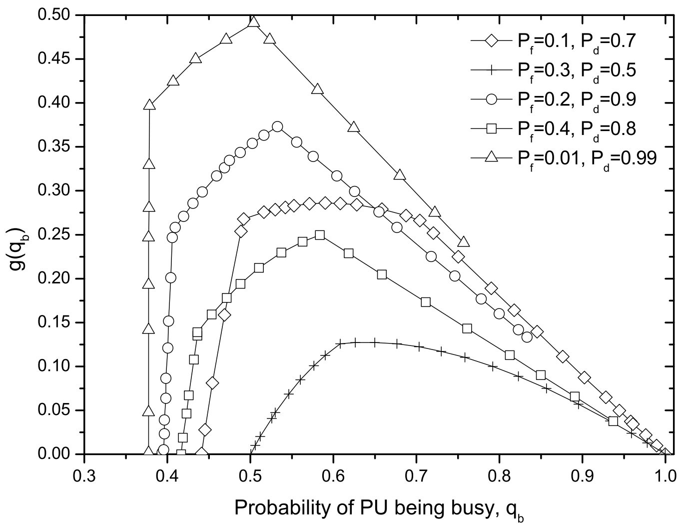

Fig. 5. Imperfect sensing effects.

any long term SU rates achievable by policies for which the restriction of always giving priority to PU traffic is removed. The mode of operation, the performance space and the optimality of the proposed policies is investigated in models where SUs are either infinitely backlogged, or finite exogenous packet arrivals to SU queues occur. An important feature of the proposed transmission algorithm is that the optimal transmit probabilities can be computed offline, through solving a convex optimization problem, and can be communicated to users. A centralized and a distributed version of the algorithm are presented, both of which are applicable depending on the setup. Simulation results verify the benefits of our approach, as well as the consistency of the proposed distributed algorithm with its centralized counterpart performance-wise. A possible extension to this work is the design of a dynamic, online version of the proposed algorithm. Furthermore, the uncoordinated interaction of multiple PUs and SUs gives rise to game-theoretic models that warrant further investigation.

APPENDIX A
PROOF OF PROPOSITION 1

Let us define as $\mathcal{R}_0$ the performance space of $\bar{r}_s$ defined by (6) where $q_b,\ q_e,\ \{q\left(s,i\left|e\right.\right)\}$, $\{q\left(s,i\left|b\right.\right)\}$ satisfy (8)-(13) and $\hat{\mathcal{R}}_0$ the performance space of $\bar{r}_s$ defined by (19) where $\{q\left(e,s,i\right)\}$, $\{q\left(b,s,i\right)\}$ satisfy (15)-(18). Due to the transformation, it holds that any $\bar{r}_s \in \mathcal{R}_0$ is also in $\hat{\mathcal{R}}_0$, i.e., $\mathcal{R}_0 \subseteq \hat{\mathcal{R}}_0$.

Conversely, we consider any $\bar{r}_s \in \hat{\mathcal{R}}_0$. Assuming that $q_e \neq 0$ and $q_b \neq 0$, we make the transformation $q_e = \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} q\left(e,s,i\right)$, $q_b = \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} q\left(b,s,i\right)$, $q\left(s,i\left|e\right.\right) = \frac{q(e,s,i)}{q_e}$ and $q(s,i\left|b\right.) = \frac{q(b,s,i)}{q_b}$. Since the parameters $\{q\left(e,s,i\right)\}$ and $\{q\left(b,s,i\right)\}$ satisfy (15)-(18), it can be shown after some basic algebraic manipulations that q_b, q_e, $\{q\left(s,i\left|e\right.\right)\}$ and $\{q(s,i\left|b\right.)\}$ satisfy (8)-(13). Hence, $\bar{r}_s \in \mathcal{R}_0$, i.e., $\hat{\mathcal{R}}_0 \subseteq \mathcal{R}_0$.

In case that $q_b = 0$, we define $q(s,i\left|b\right.) = 0$ for $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$. Again after some basic algebraic manipulations, it can be shown that $\hat{\mathcal{R}}_0 \subseteq \mathcal{R}_0$. Similarly, when $q_e = 0$, we define $q(s,i\left|e\right.) = 0$ for $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$ and it can be shown that $\hat{\mathcal{R}}_0 \subseteq \mathcal{R}_0$.

Based on the above, it can be concluded that $\mathcal{R}_0 = \hat{\mathcal{R}}_0$.

APPENDIX B
PROOF OF CORROLARY 2

The optimization problem defined in the corollary has always a feasible solution, which can be obtained through setting $x\left(b,s,i\right) = 0$ for $s \in \mathcal{S}$, $i \in \mathcal{I}_s$ and selecting arbitrarily $x(b,s,0) \geq 0$, so that $\sum_s x(b,s,0) = 1$, resulting to $\sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} r_p(s,i)x(b,s,i) = r_p(0)$. Since $\hat{\lambda}$ is the optimal value of its objective, it follows that $r_p\left(0\right) \leq \hat{\lambda}$ as expected. Physically, this choice of parameters, corresponds to the case where SUs never cooperate.

If λ_p belongs to the stability region of the system, then (15)-(18) are satisfied. But then, Eqs. (20)-(23) are also satisfied by choosing $x(b,s,i) = q(b,s,i)$, which implies that $\lambda_p \leq \hat{\lambda}$.

Conversely, given any $\lambda_p \leq \hat{\lambda}$, the choice of $q(b,s,i) = \left(\lambda_p/\hat{\lambda}\right)\hat{x}\left(b,s,i\right)$ for $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$, $q(e,s,i) = 0$ for $s \in \mathcal{S}$ and $i \in \mathcal{I}_s$, and $q(e,s,0) \geq 0$ arbitrarily chosen so that $\sum_{s\in\mathcal{S}} q(e,s,0) = 1 - \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} q(b,s,i)$ satisfies (16)-(18). In addition, $\sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} r_p(s,i)q(b,s,i) = \lambda_p$, proving that the λ_p belongs to the stability region of the PU queue. This concludes the proof.

APPENDIX C
PROOF OF THEOREM 4

Let $\bar{\mathbf{r}} \in \mathcal{R}_2$. If $\lambda_p = \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} r_p(s,i)p(1,s,i)$, then clearly $\bar{\mathbf{r}} \in \mathcal{R}_0$. Assume next that $\lambda_p < \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} r_p(s,i)p(1,s,i)$. We distinguish the following cases:

Case 1. $\lambda_p \geq r_p\left(0\right)p(1)$, where $p\left(1\right) \triangleq \sum_{s\in\mathcal{S}} \sum_{i\in\mathcal{I}_s^0} p(1,s,i)$ denotes the total probability that PU transmits, summed over all SU s and transmit power levels.

Note that since $r_p\left(0\right)p\left(1\right) \leq \lambda_p < \sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s^0} r_p(s,i)p(1,s,i)$, for each λ_p in the interval above, there exists a parameter α, with $0 \leq \alpha < 1$, such that it holds

$$\lambda_p = \alpha\left(\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s} r_p(s,i)p(1,s,i)\right) + (1-\alpha)\,r_p\left(0\right)p\left(1\right). \tag{60}$$

We define now the new set of parameters $q\left(b,s,i\right)$ and $q\left(e,s,i\right))$ by setting $q\left(e,s,i\right) = p\left(0,s,i\right)$ for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$ and

$$q(b,s,i) = \begin{cases} \alpha p(1,s,i) & \text{if } i \in \mathcal{I}_s \\ \alpha p(1,s,0) + (1-\alpha)p\left(1,s\right) & \text{if } i = 0, \end{cases} \tag{61}$$

for all $s \in \mathcal{S}$, where $p\left(1,s\right) \triangleq \sum\limits_{j\in\mathcal{I}_s^0} p(1,s,j)$. Since $0 \leq \alpha < 1$, parameters $q\left(e,s,i\right)$ and $q\left(b,s,i\right)$, for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$, are non-negative. Furthermore, note that $\sum_{i\in\mathcal{I}_s^0} q(b,s,i) = \sum_{i\in\mathcal{I}_s^0} p(1,s,i)$. Hence the new set of parameters satisfies (17). Also, since $P_s\left(0\right) = 0$, it can be shown that the new set of parameters satisfy (32). Finally, due to (60), it follows that (15) is satisfied. Hence the new set of parameters satisfy (15)-(18). Also since the SU rates computed according to (19) (where $q\left(e,s,i\right) = p\left(0,s,i\right)$ for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$) are the same as the ones given by (29), it follows that $\bar{\mathbf{r}} \in \mathcal{R}_0$.

Case 2. $\lambda_p < r_p\left(0\right)p(1)$. Define the new set of parameters as follows

$$q\left(b,s,i\right) = \begin{cases} 0 & \text{if } i \in \mathcal{I}_s \\ \frac{\lambda_p}{r_{p(0)}p(1)} p\left(1,s\right) & i = 0, \end{cases} \tag{62}$$

and

$$q\left(e,s,i\right) = \begin{cases} p\left(0,s,i\right) & \text{if } i \in \mathcal{I}_s \\ \beta \sum\limits_{i\in\mathcal{I}_s^0} p\left(0,s,i\right) + p\left(0,s,0\right) & \text{if } i = 0, \end{cases} \tag{63}$$

for all $s \in \mathcal{S}$, where $\beta = \frac{1-\frac{\lambda_p}{r_p(0)}}{1-p(1)} - 1$. Since $\lambda_p < r_p\left(0\right)p(1)$, and $p\left(1\right) \leq 1$, it follows that $\beta > 0$, hence, all the defined parameters are non-negative. Also, due to (33), (17) is satisfied. Next, it can be easily shown that (15) is satisfied. Furthermore, due to (32), (16) is also satisfied. Finally, since $P_s\left(0\right) = 0$, it follows that the SU rates computed according to (19) and (63), are the same as the ones given by (29). Hence we conclude that $\bar{\mathbf{r}} \in \mathcal{R}_0$.

APPENDIX D
PROOF OF PROPOSITION 5

We assume first that there exist q_b, $\{q\left(s,i\left|b\right.\right)\}$ and $\{q\left(s,i\left|e\right.\right)\}$ that satisfy the constraints of OPT1. In this case, due to (39), it follows that

$$q_b r_p(0)\mathcal{P}_D \leq \lambda_p \leq r_{p,max} q_b \mathcal{P}_D + q_b\left(1-\mathcal{P}_D\right) r_p\left(0\right),$$

and, consequently,

$$\frac{\lambda_p}{\mathcal{P}_D r_{p,max} + \left(1-\mathcal{P}_D\right) r_p\left(0\right)} \leq q_b \leq \frac{\lambda_p}{\mathcal{P}_D r_p\left(0\right)}.$$

Taking into account that $q_b \leq 1$, (44) follows. Conversely, it is assumed that (44) holds. By choosing the vectors

$$q^1(1,0\left|b\right.) = 1,\ \ q^1\left(s,i\left|b\right.\right) = 0 \text{ otherwise},$$

and

$$\sum_{s\in\mathcal{S}} q^1(s,0\left|e\right.) = 0,\ \ \sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s} q^1\left(s,i\left|e\right.\right) = 1,$$

Eq. (39) results to $\lambda_p^1 = q_b\mathcal{P}_D r_p(0)$. Similarly, if (s^*,i^*) satisfies $r_p\left(s^*,i^*\right) = \max_{s,i}\left\{r_p\left(s,i\right)\right\}$, by choosing the vectors

$$q^2(s^*,i^*\left|b\right.) = 1,\ \ q^2\left(s,i\left|b\right.\right) = 0 \text{ otherwise}$$

and

$$\sum_{s\in\mathcal{S}} q^2(s,0\left|e\right.) = 1,\ \ \sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{I}_s} q^2\left(s,i\left|e\right.\right) = 0$$

results to

$$\lambda_p^2 = q_b\mathcal{P}_D r_{p,max} + q_b\left(1-\mathcal{P}_D\right) r_p\left(0\right).$$

Since by (39) it holds $\lambda_p^1 \leq \lambda_p \leq \lambda_p^2$ there is an α such that $\alpha\lambda_p^1 + (1-\alpha)\lambda_p^2 = \lambda_p$ with $0 \leq \alpha \leq 1$. Hence, the vectors

$$q(s,i\left|b\right.) = \alpha q^1\left(s,i\left|b\right.\right) + (1-\alpha) q^2(s^*,i^*\left|b\right.)$$

and

$$q(s,i\left|e\right.) = \alpha q^1\left(s,i\left|e\right.\right) + (1-\alpha) q^2(s,i\left|e\right.)$$

satisfy the constraints of OPT1.

REFERENCES

[1] "Report of the spectrum efficiency working group," FCC Spectrum Policy Task Force, Tech. Rep. 02-135, 2002.

[2] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, KTH, Stockholm, Sweden, 2000.

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

[4] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey." *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, Sept. 2006.

[5] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, May 2007.

[6] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and fairness in spectrum assignment for opportunistic spectrum access," *ACM/Springer MONET*, vol. 11, no. 4, Aug. 2006.

[7] Y. Chen, 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, pp. 2053–2071, Jan 2008.

[8] R. Urgaonkar and M. J. Neely, "Opportunistic scheduling with reliability guarantees in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 8, pp. 766–777, Jan 2009.

[9] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," in *Proc. IEEE*, vol. 97, Jan 2009, pp. 894–914.

[10] O. Simeone, I. Stanojev, S. Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz, "Spectrum leasing to cooperating secondary ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 203–213, Jan 2008.

[11] O. Simeone, Y. Bar-Ness, and U. Spagnolini, "Stable throughput of cognitive radios with and without relaying capability," *IEEE Trans. Commun.*, vol. 55, pp. 2351–2360, Jan 2007.

[12] I. Krikidis, J. Laneman, J. Thompson, and S. Mclaughlin, "Protocol design and throughput analysis for multi-user cognitive cooperative systems," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 4740–4751, Jan 2009.

[13] S. Kompella, G. D. Nguyen, J. Wieselthier, and A. Ephremides, "Stable throughput tradeoffs in cognitive shared channels with cooperative relaying," in *Proc. IEEE INFOCOM*, 2011, pp. 1961–1969.

[14] R. Urgaonkar and M. Neely, "Opportunistic cooperation in cognitive femtocell networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 607 –616, April 2012.

[15] E. Altman, *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1999.

[16] M. J. Neely, *Stochastic Network Optimization with Application to Communication & Queueing Systems*. Morgan & Claypool, Aug. 2010.

[17] R. Srikant and L. Ying, *Communication Networks: An Optimization, Control and Stochastic Networks Perspective*. Cambridge University Press, 2013.

[18] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, 2000.

[19] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Ph. Commun.*, vol. 4, no. 1, pp. 26–39, 2011.

[20] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mob. Networks and Appl.*, vol. 7, pp. 481–492, 2002.

[21] I. Papadimitriou and L. Georgiadis, “Minimum energy broadcasting in multihop wireless networks using a single broadcast tree,” *Mob. Networks and Appl.*, vol. 11, pp. 361–375, 2006.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[23] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. 2nd ed. Belmont, MA: Athena Scientific, 1996.

[24] D. Bertsekas and N. J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. 2nd ed. Belmont, MA: Athena Scientific, 1999.