IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2019, accepted December 29, 2019, date of publication January 7, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964565

Cooperative Delay-Constrained Cognitive Radio Networks: Delay-Throughput Trade-Off With Relaying Full-Duplex Capability

ALI GABER MOHAMED ALI[1], EL-SAYED AHMED YOUSSEF[1], MOHAMED R. M. RIZK[1], MOHAMED SALMAN[2], (Student Member, IEEE), AND KARIM G. SEDDIK[3], (Senior Member, IEEE)

[1]Department of Electrical Engineering, Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt
[2]Department of Electrical, Computer, and Energy Engineering, University of Colorado at Boulder, Boulder, CO 80309, USA
[3]ECNG Department, American University in Cairo, Cairo 11835, Egypt

Corresponding author: Ali Gaber Mohamed Ali (aligaber@alexu.edu.eg)

ABSTRACT In this paper, the problem of maximizing the secondary user (SU) throughput under a primary user (PU) quality of service (QoS) delay requirement is studied. Moreover, the impact of having a full-duplex capability at the SU on the network performance is investigated compared to the case of a SU with half-duplex capability. We consider a cooperative cognitive radio (CR) network in which the receiving nodes have multi-packet reception (MPR) capabilities. In our proposed model, the SU not only benefits from the idle time slots (i.e. when PU is idle), but also chooses between sharing the channel or cooperating with the PU in a probabilistic manner. We formulate our optimization problem to maximize the SU throughput under a PU QoS, defined by a delay constraint; the optimization is performed over the transmission modes selection probabilities of the SU. The resultant optimization problem is found to be a non-convex quadratic constrained quadratic programming (QCQP) optimization problem, which is, generally, an NP-hard problem. We devise an efficient approach to solve it and to characterize the network stability region under a delay constraint set on the PU. Numerical results, surprisingly, reveal that the network performance when full-duplex capability exists at the SU is not always better compared to that of a half-duplex SU. In fact, we demonstrate that a full-duplex capability at the SU can, in some cases, adversely influence the network stability performance, especially if the direct channel conditions between the SU and the destinations are worse than that between the PU and the destinations. In addition, we formulate a multi-objective programming (MOP) optimization problem to investigate the trade-off between the PU delay and the SU throughput. Our MOP approach allows for assigning relative weights for our two conflicting performance metrics, i.e., PU delay and SU throughput. Numerical results also demonstrate that our cooperation policy outperforms conventional cooperative and non-cooperative policies presented in previous works.

INDEX TERMS Cognitive radio (CR), throughput, quadratic constrained quadratic programming (QCQP), multi-objective programming (MOP).

I. INTRODUCTION

Cooperative communications have gained a growing interest over the years due to the important key role they can play in enabling efficient exploitation of wireless resources [2]. Furthermore, cognitive radio (CR) communication has emerged as a leading and powerful technology in addressing the insufficiency of available spectrum resources, which are comparatively inefficiently utilized. CR can achieve efficient exploitation of the spectrum while maintaining some quality of service (QoS) for primary users (PUs) [3]. Cooperation based techniques in cognitive radio networks are crucial to achieve the envisioned gains from cognitive networks. Cooperation can be beneficial in all phases of the cognitive radio cycle, starting from cooperative sensing to cooperative communications; the interested reader is referred to [4] and references therein.

In cooperative communications, relay nodes can decode the source node transmitted packets and help by relaying them to the destination node. Recently, cooperative

The associate editor coordinating the review of this manuscript and approving it for publication was Hayder Al-Hraishawi.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

communications have been integrated into cognitive radio networks, e.g., [5]; this can be done by allowing the secondary users (SUs) to act as relay nodes for the PUs. Consequently, the SUs exploit their available transmission resources to transmit their own data as well as relaying the PUs' data. It was demonstrated that cooperation could, in general, improve both the SU and PU throughput. In [6], the authors have shown that a reasonable improvement in the network stable throughput region can be attained due to the existence of a cooperative (relay) node.

In [7] and [8], the stability rate region for a cooperative CR network that consists of one SU (which has two data queues: one for its own packets and one to relay the PU packets), one PU, and a common destination node is derived. In [7], the authors considered a system model in which the SU uses the spectrum only when the PU is not active. A strict priority is assigned in favor of the relaying queue of the PU packets, which can significantly decrease the throughput of the SU, particularly, when the average delay of the PU packet becomes much smaller than the desired delay constraint (i.e., there is a room for improving the SU throughput while not violating the PU delay constraint). To mitigate this problem, the work in [8] introduced a model in which the SU can select between serving its own packets queue or serving the PU relaying queue according to queue service probability (i.e., the SU randomizes the service between the PU relaying queue and its own packets queue whenever it has access to the spectrum). In [7] and [8], the authors optimize the SU throughput and the average packet delay experienced by the SU and PU, respectively, subject to some stability constraints of the network. They showed that cooperation is advantageous only when the channel condition between the PU and the common destination is worse than that between the SU and the destination (same restriction as in [6]). Recently, the scheme presented in [8] was modified by the authors in [9]. The SU is assumed to have a third battery queue to store harvested energy; the stability rate region of the network was derived subject to some SU energy harvesting constraints.

In this paper, we propose a framework for optimizing the SU throughput under a PU delay requirement. Following a queuing-theoretic approach, We formulate our problem as a constrained optimization problem for maximizing the SU throughput under a PU delay constraint. The problem is shown to be non-convex and an approach is devised to find an efficient way to solve the problem. In addition, we present another formulation for our optimization problem in the form of multi-objective programming to investigate the tread-off between our objective functions of maximizing the SU throughput and minimizing the PU delay. An important feature of our proposed model is to consider the case when a full-duplex capability is available at the SU and to characterize its effect on the network stability region. Most of the works that have been previously presented on cooperative CR networks focused on the case of having all the nodes (users) to be half-duplex as full-duplex communications were considered impossible (impractical) in the past due to the effect of self-interference. More recently, several approaches that can achieve significant self-interference suppression were presented in [10]. Exploiting these approaches, it was shown in [11] that the presence of a full-duplex relay can improve the throughput of the users. In [12], the model introduced in [11] was extended to the case of a full-duplex SU that has its own packets' queue in addition to the PU relaying queue; this is unlike [11] where the relay node is assumed not to have packets of its own; the SU throughput was maximized subject to some network stability constraints. It is worth mentioning that recent works presented in the context of cooperative cognitive radio, e.g., [9] and [13] have also considered only the case of half-duplex SUs.

The significance of our presented framework stems from the emergence of various real-time applications that should be supported by cooperative CR networks e.g., gaming, video streaming, and other multimedia applications; these applications require high throughput with firm delay constraints. This aspect is mostly ignored or not investigated in all of the above-cited works [7], [8], [11], [12], [14], which mainly concentrated on improving certain performance measures subject to some network stability and/or energy harvesting constraints, with no explicit delay provisioning. In addition, there exist some other works that have focused on statistical PU delay constraints using the concept of *effective capacity* [15], [16].

It is worth mentioning that optimizing the performance of the network subject to some PU delay provisioning has been considered before in [13] for a simpler network configuration and for a different cooperation policy. More precisely, [13] considered only a half-duplex SU which simplifies the delay and stability analysis of the network. Furthermore, different from our proposed scheme, [13] considered only the case where the SU accesses the spectrum when the PU is sensed not to be active (similar to [7], [8]). We will demonstrate in the numerical simulation section that our proposed scheme outperforms the scheme presented in [13].

Our major contributions in this paper can be summarized as follows:

- We propose a delay-aware scheme aiming at maximizing the SU throughput subject to a PU delay requirement.[1] We prove that our optimization problem is a non-convex quadratic constrained quadratic programming (QCQP) optimization problem which is, in general, an NP-hard problem. We propose an efficient approach to solve it to be able to characterize the network stability region.
- We present a queuing-theoretic analysis of delay and stability for our proposed cooperative scheme. We show that our cooperation scheme, which unlike [7], [8], [13], allows the SU to simultaneously transmit with the PU, achieves better performance than previously proposed

[1] It should be noted that in this paper, we present a queuing-theoretic analysis of the PU delay; other papers, e.g., [15], [16], have focused on the concept of *Effective Capacity* to provide a statistical QoS delay guarantees without going through complex queuing analysis.

cooperative and non-cooperative schemes by enlarging the network stability region.

- We study the effect of having a SU full-duplex capability on the performance of the network based on a queuing-theoretic analysis. The full-duplex capability allows the SU to decode the PU transmissions while simultaneously transmitting some data packet over the channel. We, unexpectedly, demonstrate that having this full-duplex capability at the SU does not always enhance the system performance. It can negatively affect the stability performance of the network and, in some scenarios, result in strictly inferior performance in comparison to the case of a half-duplex SU.
- We present another formulation for our optimization problem in the form of a multi-objective programming (MOP) optimization to investigate the trade-off between the PU delay and the SU throughput. Our MOP approach allows for assigning relative weights (relative importance) to our two conflicting performance metrics, namely, PU delay and SU throughput.

The rest of the paper is organized as follows. In Section II, the system model is presented. The cooperation policy, and the delay and stability analysis are presented in Section III. In Section IV, the problem formulation and the proposed solution approach are presented. In Section V, the numerical results are presented and analyzed. Conclusions are presented in Section VI.

II. SYSTEM MODEL

In this paper, we consider a full-duplex cooperative CR network, as shown in Fig. 1, having one SU (s), one PU (p), and two distinct destinations (the secondary destination d' and the primary destination d). The SU is assumed to be a full-duplex node that can simultaneously send a packet, to d' or d, and receive a packet from the PU. The primary destination (d) is also assumed to have a multi-packet reception (MPR) capability. This means that d can decode multiple simultaneous transmissions received from the PU and the SU.

We divide the time into time slots, where each slot has a fixed duration. For simplicity of presentation, any single packet transmission is assumed to take only one time slot. We assume that the SU has two infinite queues Q_{sp} and Q_s.[2]

[2]It is worth mentioning that, although assuming finite queues at the secondary user is a more realistic assumption, it is shown, in [17], that the delay analysis for finite queues becomes very complicated and intractable. Hence, obtaining closed-form expressions for the delay becomes very difficult. By relaxing the assumption of finite queues, we are able to get expressions for the average delay and use these expressions in our optimization problems. In [18], a SU with finite queues is assumed but for a simpler system where the delay analysis is neglected due to the difficulty of finding a closed-form expression for the packets' average delay. The cooperation policy in [18] was also simpler than ours since it restricted the SU to access the channel only when the PU is idle. Another work that has considered the case of finite queues is the work in [9]; it was shown in this work that the assumption of having infinite queues is good in many cases to analyze systems with finite queues. The reason was in most cases the queue lengths do not grow to infinity and even for short queues the assumption of having infinite queues can provide good approximations for the system performance measures in the case of finite queues.

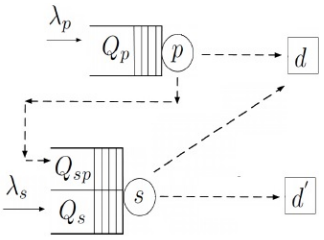

FIGURE 1. The system model.

The relayed packets received from the PU are stored in the queue Q_{sp}, whereas the SU own packets are stored in the queue Q_s. We assume that the PU has only one infinite data queue Q_p, which is used to store the PU arriving packets. We assume that the arrivals at the SU and the PU data queues are Bernoulli processes with arrival rates λ_s and λ_p, respectively. In addition, we assume that the arrival processes at the two users are independent.

We model the wireless fading channel between any two nodes (m, n), where $m \in \{s, p\}$ denotes the transmitter and $n \in \{s, d', d\}$ denotes the receiver, where clearly m is different from n, as a Rayleigh flat-fading channel. Let h_{mn} denote the channel gain, where $\mathbb{E}(|h_{mn}|^2) = \rho_{mn}^2$. The channel gains, h_{mn}'s, are modeled to be constant within any given slot but change independently from one slot to the next. All receivers are exposed to complex and independent additive white Gaussian noise with zero-mean and unit-variance. We let P denote the transmitted power from the SU or the PU, assuming that the transmitted power from any node is fixed.

In our proposed model, there will be four different scenarios for packet transmissions; the first scenario is when either the SU or the PU transmits alone (i.e., no concurrent transmissions). In this scenario, the probability of successful decoding at the receiving node can be expressed as follows:

$$f_{mn} = \mathbb{P}\{R < \frac{1}{2}\log_2(1 + P|h_{mn}|^2)\} = \exp\left(-\frac{2^{2R}-1}{P\rho_{mn}^2}\right), \quad (1)$$

where $mn \in \{ps, pd, sd, sd'\}$ and R is the transmission rate.

In the second scenario, the SU sends a packet from Q_s concurrently with a PU packet transmission, i.e., each user is interfering with the other user transmission. In this scenario, the probability that the destination node n successfully decodes the packet sent from the source node m, considering that the interference caused by node I transmission as noise, is denoted by v_{mn}^I, where $(mn, I) \in \{(sd', p), (pd, s)\}$. The derivation of the expression for v_{mn}^I is presented in Appendix A.

In the third scenario, the SU sends a relayed packet from Q_{sp} concurrently with a PU packet transmission, i.e., both users send primary packets to the primary destination node d. In this scenario, d tries to decode both transmissions by exploiting its MPR capability (e.g., using successive interference cancellation). The probability of successful

transmission, in this scenario, can be expressed by:

$$g^I_{mn} = u^I_{mn} + (1 - u^I_{mn})v^I_{mn}, \tag{2}$$

where $(mn, I) \in \{(sd, p), (pd, s)\}$ and u^I_{mn} is the probability that the destination node n successfully decodes both packets sent from source nodes m and I. The derivation of u^I_{mn} is presented in Appendix A.

In the fourth scenario, the SU utilizes its full-duplex capability, i.e., it tries to decode a primary packet while sending either a primary or a secondary packet. Let f^{dup}_{sd} represent the probability that the SU (s) decodes a primary packet from the PU (p) in the full-duplex mode; it can be written as follows:

$$\begin{aligned} f^{dup}_{ps} &= \mathbb{P}\left\{R < \frac{1}{2}\log_2\left(1 + \frac{P|h_{ps}|^2}{1 + Pg}\right)\right\} \\ &= \exp\left(-\frac{(2^{2R} - 1)(1 + Pg)}{P\rho^2_{ps}}\right), \end{aligned} \tag{3}$$

where the effectiveness of self-interference suppression is represented by the scalar gain $g \in [0, 1]$ [11], [19]. If $g = 0$ then self-interference is completely cancelled, and if $g = 1$ then no self-interference suppression is achieved. The details of the approaches that can be used for self-interference suppression are out the scope of this paper (interested readers are referred to [10] and references therein). It should be noted that to consider the case where the SU is only half-duplex, we set $f^{dup}_{sd} = 0$ not $g = 1$, since $g = 1$ corresponds to the case of full-duplex mode with no self-interference suppression which is completely different from the half-duplex mode. For $g = 1$, the SU is still full-duplex but cannot cancel any self-interference.

III. PROPOSED COOPERATIVE FRAMEWORK AND SYSTEM ANALYSIS

In this section, we introduce our proposed cooperation scheme. Then, we provide stability and delay analyses for our proposed scheme. It is worth mentioning that ACK/NACK packet transmission from d', s, and/or d is assumed at the end of each slot duration. We assume that these ACK/NACK packets are correctly received at all nodes in the network. For simplicity of presentation in this subsection, we assume that the SU is able to perfectly sense the absence or the existence of the PU.[3] As a result of the previous assumptions, the cooperation policy can be divided into two main scenarios as follows:

A. Q_P HAS PACKETS (AN ACTIVE PU; A BUSY TIME SLOT)

In this scenario, the SU could choose one of two access decisions as explained below:

- The SU takes a decision to access the channel, therefore causing interference on the PU transmission, by transmitting a packet from Q_{sp} or Q_s with probabilities p^{busy}_{sp} or p^{busy}_{s}, respectively (unlike [7], [8], [13], where the SU always abstains from accessing the channel upon detecting the existence of a PU transmission). Consequently, this should result in better exploitation of the PU channel. In this scenario, the SU can still decode the PU transmitted packet if node d fails to decode it by utilizing its full-duplex capability. It should be noted that for a half-duplex SU, the SU will not be capable of helping the PU in this case.
- The SU takes a decision to abstain from transmitting any packets and only listens to the PU transmission (to help in relaying the PU packet if the primary destination fails to decode it). This happens with probability $1 - p^{busy}_{s} - p^{busy}_{sp}$.

Note that, the PU is the sole owner of the spectrum, and as a result the PU is not expected to set any provisions to the SU; the PU transmits the packet on the top of Q_p whenever it has packets to transmit. Consequently, three possible scenarios would arise:

- If the destination d decodes the received packet successfully, then Q_p drops the packet regardless of the decision taken by the SU.
- If the SU decodes the PU packet successfully, and at the same time the destination d failed to decode it, then Q_p will drop the packet and Q_{sp} will store it.
- If both s and d were not capable of decoding the PU packet, then the PU keeps the packet on the top of Q_p to try to re-transmit it in the next time slot.

B. Q_P IS EMPTY (AN INACTIVE PU; AN IDLE TIME SLOT)

In this scenario, the SU could choose one of two access decisions as explained below:

- The SU sends a packet from Q_{sp} with probability p^{idle}_{sp}. Q_{sp} will drop this packet if the SU receives an ACK from the primary destination d.
- The SU sends a packet from Q_s with probability $p^{idle}_{s} = 1 - p^{idle}_{sp}$. Q_s will drop this packet if an ACK from the secondary destination d' is received at the SU.

Our presented model involves interaction between different queues. The delay and stability analysis involving more than two interacting queues is, in general, a very difficult problem [17]. Consequently, we use the concept of dominant system to decouple the interaction between the system queues. The dominant system approach was previously used in different contexts, e.g., [20], to obtain sufficient stability conditions for a set of interacting queues. In our dominant system, the SU sends dummy packets when it decides to transmit a packet from an empty queue; this decouples the interaction among the queues by decoupling the service rate for any queue in the system from the number of packets in the other queues. It is evident that the stability of the dominant system implies the stability of the original system (as sending dummy packets can only deteriorate the system performance). Moreover, the dominant system packets' average delay represents an upper bound on the original system packet delay.

[3] In the analysis presented later in the paper, we will consider the more practical scenario assuming sensing errors.

Next, we provide the details of our proposed system stability analysis followed by the delay analysis. We apply Loynes' theorem [21] to investigate the queues' stability in the network. Loynes' theorem roughly states that a queue is stable if its average service rate is strictly larger than its average arrival rate, if both the service and the arrival processes are stationary. It is worth mentioning that the packets of the PU can be served, in general, by two queues: Q_{sp} and Q_p. This should be considered while deriving the packet delay of the PU.

According to our cooperative policy and the scenarios explained above in subsection III-A and subsection III-B, the packets departure rate from Q_p depends on the action taken by the SU. Hence, we have two main cases as follows.

1) The SU detects that the PU is active while the PU is actually active (Q_p has packets). Let p_{det} denote the probability of detection of the PU by the SU. As a result, we have the three following scenarios (for the full-duplex case) for any packet that departs Q_p (as in subsection III-A):
   - The SU decides not to send any packets and only listens to the PU transmission to aid relaying it if d will not be able to decode it. A departure from Q_p occurs if d is able to decode the PU packet, and this happens with probability f_{pd}, or if the SU captures it while d could not, and this happens with probability $f_{ps}(1-f_{pd})$.
   - The SU decides to simultaneously transmit a packet, with the PU transmission, by transmitting a packet from Q_s with probability p_s^{busy}. A packet departs from the PU queue if it is successfully captured by either d, and this happens with probability v_{pd}^s, or by the SU using its full-duplex capability, and this happens with probability $(1-v_{pd}^s)f_{ps}^{dup}$ (because this happens if d is not able to decode the PU packet while the SU can).
   - The SU decides to interfere with the PU transmission by sending a packet from Q_{sp} with probability p_{sp}^{busy}. A packet departs Q_p if it is successfully captured by either d, and this happens with probability g_{pd}^s, or by the SU exploiting its full-duplex capability and this happens with probability $(1-g_{pd}^s)f_{ps}^{dup}$ (again, because this happens if d is not able to decode the PU packet while the SU can).
2) The SU detects that the PU is not active, while the PU is actually active. Therefore, the SU thinks that Q_p is empty. Let $(1-p_{det})$ denote the probability of misdetection of PU by the SU. As a result, we have the following two scenarios for any packet that departs Q_p (as in subsection III-B):
   - The SU transmits a packet from Q_s with probability $(1-p_{sp}^{idle})$. A packet departs Q_p if it is successfully captured either by d, and this happens with probability v_{pd}^s, or by the SU using its full-duplex capability, and this happens with probability $(1-v_{pd}^s)f_{ps}^{dup}$) assuming that d was not able to decode the PU packet.[4]
   - The SU sends a packet from Q_{sp} with probability p_{sp}^{idle}. A packet departs from Q_p if it is successfully captured by either d, and this happens with probability g_{pd}^s, or by the SU exploiting its full-duplex capability, and this happens with probability $(1-g_{pd}^s)f_{ps}^{dup}$.

As a result, the PU packets' departure rate from Q_p can be expressed as

$$\begin{aligned}\mu_p = p_{det}\{&p_s^{busy}(v_{pd}^s+(1-v_{pd}^s)f_{ps}^{dup})\\&+p_{sp}^{busy}(g_{pd}^s+(1-g_{pd}^s)f_{ps}^{dup})\\&+(1-p_s^{busy}-p_{sp}^{busy})(f_{pd}+(1-f_{pd})f_{ps})\}\\&+(1-p_{det})\{(1-p_{sp}^{idle})(v_{pd}^s+(1-v_{pd}^s)f_{ps}^{dup})\\&+p_{sp}^{idle}(g_{pd}^s+(1-g_{pd}^s)f_{ps}^{dup})\},\end{aligned} \tag{4}$$

For Q_p to be stable, the arrival rate λ_p has to be less than μ_p, i.e.,

$$\begin{aligned}\lambda_p < p_{det}\{&p_s^{busy}(v_{pd}^s+(1-v_{pd}^s)f_{ps}^{dup})\\&+p_{sp}^{busy}(g_{pd}^s+(1-g_{pd}^s)f_{ps}^{dup})\\&+(1-p_s^{busy}-p_{sp}^{busy})(f_{pd}+(1-f_{pd})f_{ps})\}\\&+(1-p_{det})\{(1-p_{sp}^{idle})(v_{pd}^s+(1-v_{pd}^s)f_{ps}^{dup})\\&+p_{sp}^{idle}(g_{pd}^s+(1-g_{pd}^s)f_{ps}^{dup})\}.\end{aligned} \tag{5}$$

Following a similar approach to the analysis of μ_p, the service rate μ_{sp} of Q_{sp} can be given as

$$\begin{aligned}\mu_{sp} = &p_{det}\frac{\lambda_p}{\mu_p}p_{sp}^{busy}g_{sd}^p+(1-p_{fa})(1-\frac{\lambda_p}{\mu_p})p_{sp}^{idle}f_{sd}\\&+(1-p_{det})\frac{\lambda_p}{\mu_p}p_{sp}^{idle}g_{sd}^p+p_{fa}(1-\frac{\lambda_p}{\mu_p})p_{sp}^{busy}f_{sd},\end{aligned} \tag{6}$$

where $\frac{\lambda_p}{\mu_p}$ is the probability that Q_p is nonempty, and p_{fa} denotes the false alarm probability (i.e., the probability to sense the PU to be active while it is idle). As a result, the arrival rate at Q_{sp}, λ_{sp}, is given by

$$\begin{aligned}\lambda_{sp} = p_{det}\frac{\lambda_p}{\mu_p}\{&p_s^{busy}f_{ps}^{dup}(1-v_{pd}^s)+p_{sp}^{busy}f_{ps}^{dup}(1-g_{pd}^s)\\&+(1-p_s^{busy}-p_{sp}^{busy})f_{ps}(1-f_{pd})\}.\end{aligned} \tag{7}$$

To satisfy the stability requirement of Q_{sp}, λ_{sp} has to be less than μ_{sp}, and this can be readily simplified to the following condition

$$\lambda_p < \frac{C}{A-B+C}\mu_p, \tag{8}$$

where A, B, and C are given, respectively, by

$$\begin{aligned}A = p_{det}\{&p_s^{busy}f_{ps}^{dup}(1-v_{pd}^s)+p_{sp}^{busy}f_{ps}^{dup}(1-g_{pd}^s)\\&+(1-p_s^{busy}-p_{sp}^{busy})f_{ps}(1-f_{pd})\},\end{aligned}$$

[4]Note that we assume that the SU will always enable its full-duplex capability even if the sensing decision was that the PU is inactive. This would enable the secondary user to alleviate some of the sensing decision errors.

$$B = p_{det} p_{sp}^{busy} g_{sd}^{p} + (1 - p_{det}) p_{sp}^{idle} g_{sd}^{p},$$
$$C = (1 - p_{fa}) p_{sp}^{idle} f_{sd} + p_{fa} p_{sp}^{busy} f_{sd}.$$

From (5) and (8), it can be easily demonstrated that the arrival rate of the PU has to satisfy the following condition for the system to be stable.

$$\lambda_p < \min\{\mu_p, \mu_r\}, \tag{9}$$

where μ_p is as given in (4) and μ_r is given by

$$\mu_r = \frac{C}{A - B + C} \mu_p. \tag{10}$$

Finally, for Q_s, the service rate μ_s can be shown to be given by

$$\mu_s = p_{det} \frac{\lambda_p}{\mu_p} p_s^{busy} v_{sd'}^{p} + (1 - p_{fa}) \left(1 - \frac{\lambda_p}{\mu_p}\right) (1 - p_{sp}^{idle}) f_{sd'} + (1 - p_{det}) \frac{\lambda_p}{\mu_p} (1 - p_{sp}^{idle}) v_{sd'}^{p} + p_{fa} \left(1 - \frac{\lambda_p}{\mu_p}\right) p_s^{busy} f_{sd'}. \tag{11}$$

To satisfy the Q_s stability requirement, the arrival rate λ_s should be less than μ_s, i.e.,

$$\lambda_s < p_{det} \frac{\lambda_p}{\mu_p} p_s^{busy} v_{sd'}^{p} + (1 - p_{fa}) \left(1 - \frac{\lambda_p}{\mu_p}\right) (1 - p_{sp}^{idle}) f_{sd'} + (1 - p_{det}) \frac{\lambda_p}{\mu_p} (1 - p_{sp}^{idle}) v_{sd'}^{p} + p_{fa} \left(1 - \frac{\lambda_p}{\mu_p}\right) p_s^{busy} f_{sd'}. \tag{12}$$

This completes our stability region characterization; any pair (λ_p, λ_s) that satisfies (9) and (12) belongs to the stability region of our proposed scheme.

It is worth mentioning that a PU packet might experience two queuing delays; the delay in Q_p and the delay in Q_{sp} if it is relayed via the SU. Consequently, the average delay experienced by the PU packet is expressed as follows:

$$D_p = \tau T_p + (1 - \tau)(T_p + T_{sp}) = T_p + (1 - \tau) T_{sp}, \tag{13}$$

where the average delays at Q_{sp} and Q_p are denoted by T_{sp} and T_p, respectively. And τ is given by

$$\tau = \frac{(1 - p_s^{busy} - p_{sp}^{busy}) f_{pd} + p_s^{busy} v_{pd}^{s} + p_{sp}^{busy} g_{pd}^{s}}{\mu_p}, \tag{14}$$

and it denotes the probability that a PU packet is decoded successfully at the destination d provided that it was dropped from Q_p. The queues Q_p and Q_{sp} are assumed to be discrete time M/M/1 queues with Geometric service processes and Bernoulli arrival processes. As a result, by applying Pollaczek-Khinchine [22] and Little's law, T_p and T_{sp} can be given by:

$$T_p = \frac{1 - \lambda_p}{\mu_p - \lambda_p}, \quad T_{sp} = \frac{1 - \lambda_{sp}}{\mu_{sp} - \lambda_{sp}}. \tag{15}$$

IV. PROBLEM FORMULATION AND THE PROPOSED SOLUTION APPROACH

In this section, we present our optimization problem. Our target is to maximize the throughput of the SU subject to a PU QoS delay requirement on the PU. Therefore, we can write our optimization problem, in an epigraph form [23, Chapter 4], as follows:

$$\begin{aligned} \mathbf{P1}: \max_{p_s^{busy}, p_{sp}^{busy}, p_{sp}^{idle}, \lambda_s} \quad & \lambda_s \\ \text{subject to } & \lambda_s \le \mu_s, \\ & D_p \le \phi, \\ & 0 \le p_s^{busy} + p_{sp}^{busy} \le 1, \\ & 0 \le p_{sp}^{idle} \le 1, \\ & p_i^{busy} \ge 0 \quad i \in \{s, sp\}, \end{aligned} \tag{16}$$

where the first constraint guarantees the stability of Q_s, while the second constraint represents the PU QoS Packet delay requirement. It should be noted that having a constraint on the PU delay is stricter than having a PU queue stability constraint and, hence, guarantees the stability of the primary queue, i.e., the length of Q_p is guaranteed not to grow up to infinity. Therefore, we do not have to have an extra stability constraint for Q_p.

It is worth mentioning that μ_s and D_p, expressed in (11) and (13), are non-convex functions in the optimization parameters. This renders the overall optimization problem to be a non-convex optimization problem.[5] Next, we introduce a number of steps that describe our solution approach for solving this non-convex optimization problem.

First, note that if μ_p is fixed to a certain value then μ_s from (11) turns out to be a linear function of the optimization parameters as proven in Appendix C. Consequently, the first constraint in **P1** in (16) becomes convex. Furthermore, if μ_p is fixed then the delay constraint ($D_p \le \phi$), which is the second constraint in **P1**, turns out to be a quadratic function of the optimization parameters in the form $\boldsymbol{p}^T \boldsymbol{A} \boldsymbol{p} + \boldsymbol{c}^T \boldsymbol{p} + d \le 0$, as proven in Appendix C, where $\boldsymbol{p}$ is a vector containing all the optimization parameters. Unlike the first constraint, the second constraint is, unfortunately, non-convex since the matrix $\boldsymbol{A}$ is not, in general, a positive semi-definite matrix as will be illustrated later. As a result, by fixing μ_p, the first constraint is converted to be linear and the second constraint is converted to be non-convex quadratic, while the remaining constraints are already linear constraints. This form of optimization problems is known as non-convex QCQP optimization problems. To solve this resultant problem, we utilize the feasible point pursuit successive convex approximation (FPP-SCA) algorithm proposed in [24]. This algorithm solves the problem by linearizing the non-convex parts of the quadratic delay constraint. We use it because of its advantages explained in [24], over other methods presented in the same context in [25]–[27] that can be used to deal with QCQP optimization problems.

[5]The non-convexity of D_p is proved in Appendix B

In [25], the authors proposed the reformulation linearization technique (RLT) to solve non-convex QCQP optimization problems. This method consists of two steps; the reformulation step and the linaerization step. In the first step (the reformulation step), non-linear redundant constraints are created. These constraints involve pairwise product combinations of the individual scalar variables, due to multiplying various constraint pairs. In the second step (the linearization step), each distinct product of variables is replaced by a continuous variable. Then, the resultant convex problem is solved and an approximate solution of the non-convex QCQP optimization problem is obtained. The main drawback of this method is that the obtained solution of the linear program for the non-convex QCQP problem is rarely feasible. Moreover, the size of the linear program approximation is much larger than that of the original problem, and this makes it computationally involved. The authors in [26], [27] adopted the successive convex approximation (SCA) method to approximate the non-convex QCQP optimization problem by a convex one. The SCA approach separates each quadratic term into concave and convex parts; then, the concave parts are replaced by convex approximations (which are usually linear) around a feasible point. Then, the resultant convex optimization problem is solved where the obtained solution acts as the approximation point for the next iteration. The main disadvantage of the SCA based algorithms presented in [26], [27] and the RLT method presented in [25] is that they all require to start from a feasible point to converge. It should be noted that obtaining an initial feasible point is, in general, difficult. The FPP-SCA algorithm presented [24] overcomes these drawbacks and gives more accurate results as explained in details in [24]; therefore, we adopt it to solve our non-convex QCQP optimization problem as will be explained next.

To characterize the stability region (λ_p, λ_s), we need to find the maximum achievable λ_s for every λ_p, denoted by $\lambda_s^*(\lambda_p)$. To do this we need to find the optimum μ_p for a given λ_p and this can be done by a numerical search over the values of μ_p; for each value of μ_p, a non-convex QCQP optimization problem is solved. For each value of μ_p, we solve for the maximum stable λ_s given the value of μ_p. Eventually, the maximum value of λ_s over all feasible μ_p's is set to be our solution, i.e., $\lambda_s^*(\lambda_p)$. The values of feasible μ_p's range from λ_p to μ_p^m, where μ_p^m is the maximum feasible value of μ_p and the minimum value of μ_p is obviously λ_p to guarantee the stability of Q_p. The value of μ_p^m can be easily determined by setting $p_s^{busy} = 0$ and $p_{sp}^{idle} = 1$ in (9) and performing the optimization only over p_{sp}^{busy} (this can be easily done via a simple one-dimensional numerical search).

Based on the previous discussion, and for a given μ_p, we can readily see from (11) that the first constraint is converted to be a linear constraint in the optimization parameters. If a 4-D vector is defined as follows $\boldsymbol{p} = [p_s^{busy}, p_{sp}^{busy}, p_{sp}^{idle}, \lambda_s]^T$, we can now write the second constraint as $\boldsymbol{p}^T\boldsymbol{A}\boldsymbol{p} + \boldsymbol{c}^T\boldsymbol{p} + d \leq 0$ where $\boldsymbol{A}$, $\boldsymbol{c}$, and d can be directly derived from (6), (7), and (13). It can be readily seen that this constraint is a quadratic and non-convex constraint because $\boldsymbol{A}$ is, in general, an indefinite matrix. As mentioned before, we use the FPP-SCA algorithm to linearize the non-convex parts of the delay constraint.

The FPP-SCA algorithm linearizes the non-convex parts of the delay constraint as explained next. Using Eigenvalue decomposition, the matrix $\boldsymbol{A}$, which is, in general, an indefinite matrix, can be expressed as $\boldsymbol{A} = \boldsymbol{A}^+ + \boldsymbol{A}^-$, where $\boldsymbol{A}^+ \succeq 0$ and $\boldsymbol{A}^- \preceq 0$. For any $z \in R^{4\text{x}1}$, we get

$$(\boldsymbol{p} - z)^T\boldsymbol{A}^-(\boldsymbol{p} - z) \leq 0, \tag{17}$$

$$\boldsymbol{p}^T\boldsymbol{A}^-\boldsymbol{p} \leq 2z^T\boldsymbol{A}^-\boldsymbol{p} - z^T\boldsymbol{A}^-z. \tag{18}$$

With the help of the above inequalities, we can replace the quadratic non-convex constraint $(\boldsymbol{p}^T\boldsymbol{A}\boldsymbol{p} + \boldsymbol{c}^T\boldsymbol{p} + d \leq 0)$ by the following convex one:

$$\boldsymbol{p}^T\boldsymbol{A}^+\boldsymbol{p} + 2z^T\boldsymbol{A}^-\boldsymbol{p} + \boldsymbol{c}^T\boldsymbol{p} + d \leq z^T\boldsymbol{A}^-z, \tag{19}$$

for some fixed vector $\mathbf{z}$, which relaxes the non-convex part of the constraint to a linear one. The selection of the vector $\mathbf{z}$ is given in Algorithm **1**.

Algorithm 1

For a given λ_p

For a step size $\delta = \frac{\mu_p^m - \lambda_p}{N}$, where N is the number of points we run the algorithm for.

For $\mu_p = \lambda_p : \delta : \mu_p^m$

Initialization: set $i = 0$, and $\mathbf{z}_0 = \mathbf{0}$.

Repeat

1) solve

$$\begin{aligned}\lambda_s(\lambda_p, \mu_p) = \min_{\boldsymbol{p}}\ & \boldsymbol{x}^T\boldsymbol{p} + \Lambda s \\ \text{s. t. } & \boldsymbol{v}^T\boldsymbol{\alpha} + u = \mu_p, \\ & \boldsymbol{m}^T\boldsymbol{p} + n \leq 0, \\ & \boldsymbol{p}^T\boldsymbol{A}^+\boldsymbol{p} + 2z_i^T\boldsymbol{A}^-\boldsymbol{p} \\ & \quad + \boldsymbol{c}^T\boldsymbol{p} + d \leq z_i^T\boldsymbol{A}^-z_i + s, \\ & 0 \leq \boldsymbol{b}^T\boldsymbol{p} \leq 1, \\ & \mathbf{0} \preceq \boldsymbol{p} \preceq \mathbf{1}, \\ & 0 \leq s. \end{aligned} \tag{20}$$

2) Let $\boldsymbol{p}_k^*$ represent the optimal $\boldsymbol{p}$ obtained at the *i-th* iteration, and set $\boldsymbol{z}_{i+1} = \boldsymbol{p}_i^*$.

3) Set $i = i + 1$.

until convergence

Return $\lambda_s^*(\lambda_p) = \max \lambda_s(\lambda_p, \mu_p)$.

Now, our non-convex problem is converted into a convex one. Lastly, the optimization problem in **P1** can be written as in Algorithm **1**, where $\boldsymbol{x} = [0, 0, 0, -1]^T$, $\boldsymbol{b} = [1, 1, 0, 0]^T$ and u, $\boldsymbol{v}$, n, $\boldsymbol{m}$ can be readily derived from (4).

Algorithm **1** finds the feasible solution that maximizes the SU throughput subject to a PU QoS packet delay constraint. It should be noted that we use a slack variable (s) to secure the feasibility of the approximated problem, and use a penalty (Λ) to guarantee that this slack variable is mildly used.

For the stability region (λ_p versus λ_s) to be drawn with the PU delay constraint, λ_p is varied from zero to λ_p^m and the SU maximum stable throughput is found for each λ_p using Algorithm **1**. Then, the convex hull is taken over the obtained values. Note that it is already known that the stable throughput point $(\lambda_p, \lambda_s) = (0, f_{sd'})$ is in the stable throughput region. The previous point corresponds to the scenario when $\lambda_p = 0$, and consequently, the SU is free to send its own data packets in all time slots achieving a maximum stable throughput of $f_{sd'}$.

A. SOFT-DELAY GUARANTEE SCHEME

In this subsection, we formulate an optimization problem that targets two (contradicting) objectives: to maximize the SU throughput λ_s and to minimize the PU average delay D_p. Unlike the previous formulation, we do not need to have a strict PU delay constraint like $D_p \leq \phi$. Our aim here is to have a multi-objective programming (MOP) optimization formulation. Our optimization problem can be written as follows:

$$\begin{aligned} \mathbf{P2}: \min_{p_s^{busy}, p_{sp}^{busy}, p_{sp}^{idle}, \lambda_s, t} \quad & D_p - k\lambda_s \\ \text{subject to } & \lambda_s \leq \mu_s, \\ & 0 \leq p_s^{busy} + p_{sp}^{busy} \leq 1, \\ & 0 \leq p_{sp}^{idle} \leq 1, \\ & p_i^{busy} \geq 0 \quad i \in \{s, sp\}, \end{aligned} \tag{21}$$

where k is a constant weighting factor that trades-off the two objective functions. By changing the value of k, we can control the importance of the average delay of the PU packets versus the SU throughput. For larger k, the above optimization problem is more oriented towards maximizing λ_s with a finite D_p; while for smaller k, the problem is more oriented towards minimizing D_p. Using the epigraph form [23, Chapter 4], our optimization problem can be rewritten as follows:

$$\begin{aligned} \mathbf{P2}: \min_{p_s^{busy}, p_{sp}^{busy}, p_{sp}^{idle}, \lambda_s, t} \quad & t \\ \text{subject to } & D_p - k\lambda_s \leq t, \\ & \lambda_s \leq \mu_s, \\ & 0 \leq p_s^{busy} + p_{sp}^{busy} \leq 1, \\ & 0 \leq p_{sp}^{idle} \leq 1, \\ & p_i^{busy} \geq 0 \quad i \in \{s, sp\}. \end{aligned} \tag{22}$$

In the last optimization problem the trade-off between λ_s and D_p is captured by minimizing the objective function t which is lower bounded, in the first constraint, by $D_p - k\lambda_s \leq t$. While the rest of the constraints in the last two optimization problems in (21) and (22) are similar to those in the previous formulation **P1**. Similar to **P1**, we can show that D_p is a non convex function in the optimization parameters as shown in Appendix B. However, for fixed μ_p, the first constraint becomes a non convex quadratic constraint, as explained above and the second constraint becomes a linear constraint as shown in Appendix C. Hence, we obtain, for each fixed μ_p, a non-convex QCQP optimization problem which can be solved using the FPP-SCA algorithm presented in [24] as explained above.

It is worth noting that in our proposed multi-objective approach, the stability of the PU queue will always be guaranteed; having the PU delay as part of the objective function will guarantee that the PU queue will always be stable, otherwise the PU delay D_p will grow to infinity which will clearly not result in minimizing our proposed objective function.

V. NUMERICAL RESULTS

In this section, the effect of having a SU full-duplex capability on the stable throughput region is investigated under a PU packet delay constraint. We present and compare the stability region for three different schemes (a) full-duplex SU with complete self-interference suppression (i.e., $g = 0$), (b) full-duplex SU with no self-interference suppression ($g = 1$), and (c) half-duplex SU with $f_{ps}^{dup} = 0$. We carry out this comparison for four different channel scenarios, i.e., different sets of channel gains variances ρ_{sd}^2 and ρ_{pd}^2, while fixing the remaining simulation parameters. The fixed parameters are: $R = 1$, $P = 10$, $\rho_{s,d'}^2 = 0.8$, $\rho_{p,s}^2 = 0.6$, and $\rho_{p,d'}^2 = 0.3$. Furthermore, we show that our cooperation policy outperforms some other previously proposed cooperation policies presented in [7], [8], [13]. We also investigate the trade-off between the PU delay and the SU throughput using our MOP formulation approach. We also investigate the impact of introducing sensing errors at the SU on the system performance.

For the first scenario, high direct channel gains between $p - d$ and $s - d$ are assumed. We choose $\rho_{pd}^2 = 0.8$ and $\rho_{sd}^2 = 0.9$. The results are shown in Fig. 2. Since a good direct channel between the PU and the primary destination d already exists, most of the packets are directly delivered from the PU to d, and therefore, the three configurations achieve almost the same maximum stable throughput (in this scenario, relaying will not be very effective and will not result in significant performance gains). For small λ_p, the SU does not play a crucial role in relaying the PU packets; as a result all the three configurations achieve nearly close performance. While, for high λ_p, the SU full-duplex capability can help achieving faster PU Packets delivery, allowing the SU to utilize more resources to transmit its own data packets. Thus, as clear in Fig 2, the system with full-duplex enabled SU achieves a larger stability region than that of the half-duplex one.

In the second scenario, bad direct channel gain between $p - d$ ($\rho_{pd}^2 = 0.04$) and good direct channel gain between $s - d$ ($\rho_{sd}^2 = 0.9$) are assumed. In this scenario, the SU

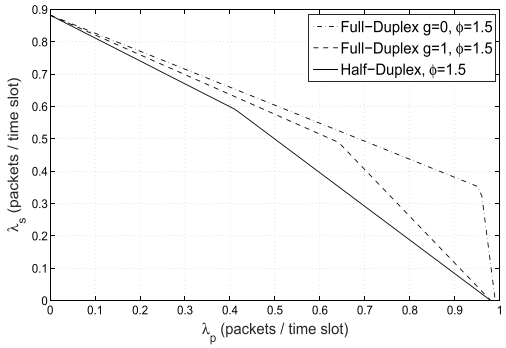

FIGURE 2. Stable throughput region for the case of good direct channel gain *p-d* and good channel gain *s-d*.

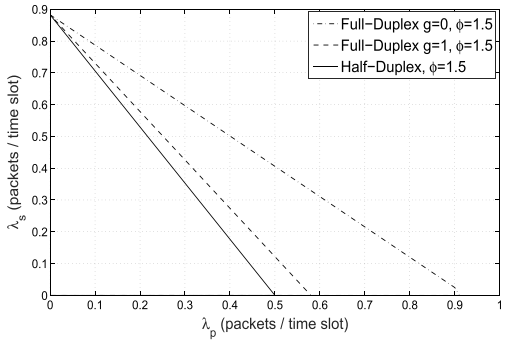

FIGURE 3. Stable throughput region for the case of bad direct channel gain *p-d* and good channel gain *s-d*.

plays a crucial role by serving as a relay to deliver the PU packets due to the bad $p-d$ channel. It is obvious from Fig. 3 that the existence of the full-duplex capability at the SU can significantly improve the stable throughput region compared to that of the half-duplex mode. This is because the full-duplex capability enables the SU to store more PU packets in the relaying queue Q_{sp} to overcome the bad $p-d$ channel. Actually, in this scenario, the PU service rate is dominated by the relaying rate of the SU. On the other hand, in the half-duplex system, λ_s can significantly decrease as a result of the bad $p-d$ channel because the SU has to listen more to the PU transmissions, which decreases the opportunities available for the SU to transmit its own data packets.

In the former two scenarios, a good direct channel between s and d is assumed. This, in turn, means that the SU will be able to easily deliver the PU relayed packets to the primary destination d. As a result, the full-duplex capability at the SU was always beneficial and it improved the stability region. In the next two scenarios, we consider two other possibilities in which the SU suffers from bad channel to the primary destination d.

In the third scenario, good direct channel gain between $p-d$ ($\rho^2_{pd}=0.8$) and bad direct channel gain between $s-d$ ($\rho^2_{sd}=0.04$) are assumed. Fig. 4 demonstrates that almost the same maximum stable λ_p is achieved by the three configurations, since a good direct channel between the PU and d already exists, hence, most of the PU packets are

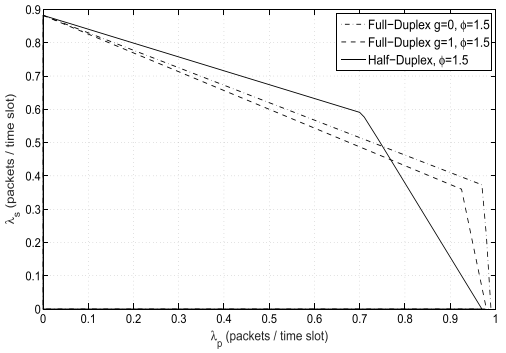

FIGURE 4. Stable throughput region for the case of good direct channel gain *p-d* and bad channel gain *s-d*.

successfully delivered to d on the PU direct channel. On the other hand, none of the three configurations is strictly better over the whole feasible range of λ_p. For small λ_p, the best performance is achieved by the half-duplex configuration, while for large λ_p the full-duplex configuration is the best. The reason behind this is that for small values of the PU arrival rate λ_p, the stability conditions are dominated by Q_{sp}; refer to (9). Consequently, having only half-duplex capability decreases the Q_{sp} arrival rate of (7) which, as a result, preserves the Q_{sp} stability for a wider range of λ_p compared to that of the full-duplex mode. As λ_p increases, it is obvious that the full-duplex system becomes the best configuration. This occurs because, for large values of λ_p, the stability conditions are dominated by Q_p; refer to (9). Therefore, having a full-duplex capability at the SU enlarges Q_p service rate (4), which, as a result, preserves Q_p stability for larger values of λ_p.

In the fourth scenario, bad direct channel gain between $p-d$ ($\rho^2_{pd}=0.04$) and bad direct channel gain between $s-d$ ($\rho^2_{sd}=0.04$) are assumed. Fig. 5 demonstrates that the half-duplex system is always better than the full-duplex system. For this scenario, and when we have a full-duplex SU, most of the PU packets have to be delivered to d via the relying queue Q_{sp} because of the good direct channel condition between p and s and bad direct channel condition between p and d. However, these packets cannot be relayed by the SU due to the bad direct channel condition of the $s-d$ link causing accumulation of PU packets in the relaying queue Q_{sp}. On the other hand, having only half-duplex SU decreases the accumulation of packets at Q_{sp} by decreasing Q_{sp} arrival rate which, in turn, improves the stability region. This obviously shows that having a full-duplex SU is not always advantageous as in many scenarios it would be better for the PU to resend its lost packets and not to relay them through the SU.

To evaluate the effectiveness of the proposed cooperation policy, we compare our system performance (stability region) to the policy where there is no cooperation between the PU and SU and the policies presented in [7], [8], and [13]. In [13], the cooperation policy presented in [7], [8] was adopted (where the SU is assumed to be half-duplex and transmits only when the PU is inactive), and the SU

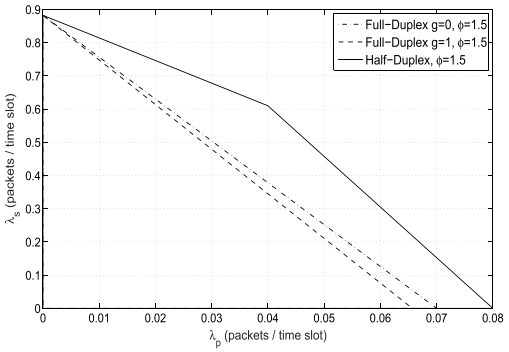

FIGURE 5. Stable throughput region for the case of bad direct channel gain *p-d* and bad channel gain *s-d*.

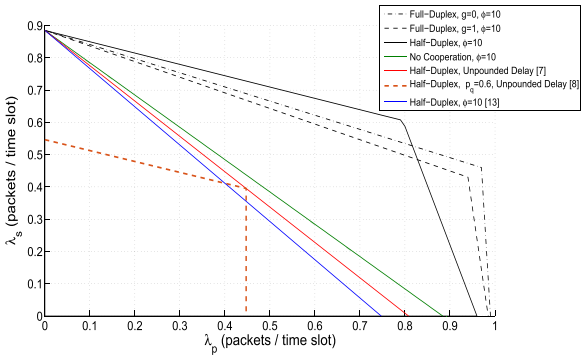

FIGURE 7. Our performance vs traditional schemes performance for good *p-d* and bad *s-d* links.

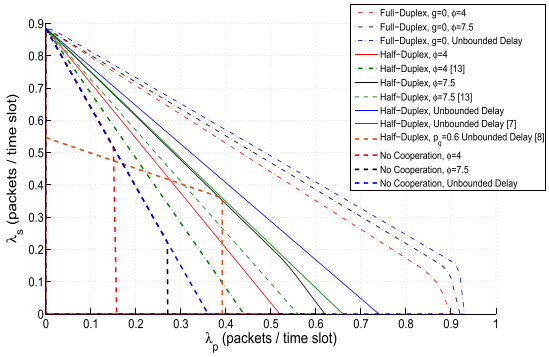

FIGURE 6. Our performance vs traditional schemes performance for bad *p-d* and good *s-d* links. Note that p_q, used in [8], corresponds to p_s^{idle} in our analysis.

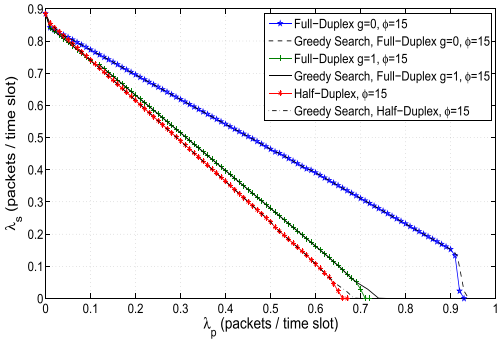

FIGURE 8. Comparison between the FPP-SCA algorithm and the greedy search algorithm.

throughput was maximized subject to a delay constraint on the PU. In Fig. 6, we use the same system parameters adopted in Fig. 3. We choose this scenario because it reflects the crucial role of the SU (that has a good channel to d) in relaying the PU packets when the PU suffers from bad channel to d. Fig. 6 demonstrates that our proposed scheme, for both the full-duplex and the half-duplex configurations, achieves larger stability regions than those of the case where there is no cooperation between the PU and SU for different values of the PU delay constraint. It also shows that our scheme for the half-duplex configuration outperforms that of [7], [13] (which represents the unbounded delay scheme for [13]), and [8]. In Fig. 6, the impact of relaxing the delay requirement of the PU on the system performance is also studied. It shows that as the delay constraint is relaxed by increasing the value of ϕ, we will have a larger feasibility set, and hence, a larger stability region is achieved. It should be noted that if ϕ is increased to infinity (an unbounded delay), the achieved stability region represents the case where the SU throughput is maximized subject to a network stability constraint only, without considering any delay constraints.

In Fig. 7, we use the same system parameters used to generate Fig. 4, which represents the opposite scenario to the one presented in Fig. 6. Fig. 7 demonstrates that even if the SU has a bad channel to d and the PU has a good direct channel to d, the cooperation between the PU and the SU is still beneficial. The authors in [7], [8], [13] stated that cooperation between the PU and the SU, in this scenario, does not make sense and can adversely affect the PU performance, as shown in Fig. 7; this is because in this case the PU packets are accumulating in the relaying queue which already has a bad channel to d. However, in our proposed cooperation policy, the SU is assumed to be able to access the channel when the PU is active and transmit simultaneously with it; hence, the SU does not have to restrict its transmissions to the PU idle time slots as in [7], [8], [13]. This increases the transmission probability and prevents the PU packets from being congested in Q_{sp}. Consequently, a larger stability region is achieved and this is one of the main advantages of our cooperation protocol.

Fig. 8 evaluates the potential accuracy of the FPP-SCA algorithm adopted in this paper, by comparing its results generated by Algorithm **1** to the exact results obtained from the greedy search algorithm. This is performed, without taking the convex hull, for the case where the channel $p-d$ is bad and that between $s-d$ is good. Fig. 8 demonstrates how accurate is the adopted FPP-SCA algorithm, as its generated results are almost identical to the exact results obtained from the greedy search algorithm over the entire range of λ_p.

Fig. 9 plots the stability region of the soft delay guarantee scheme (with our multi-objective programming). We use the same system parameters used to generate Fig. 3, i.e., the PU suffers from bad direct channel to d and the SU has good direct channel to d, because this is the scenario where the SU

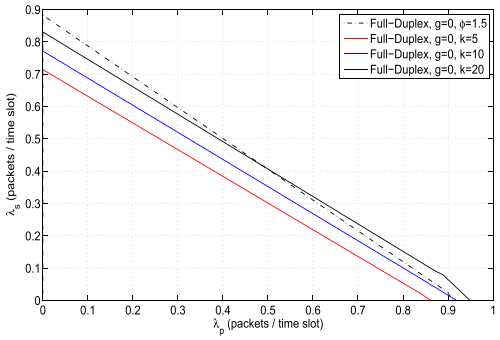

FIGURE 9. Impact of the weighting factor k for optimizing the difference between the PU delay and the SU throughput.

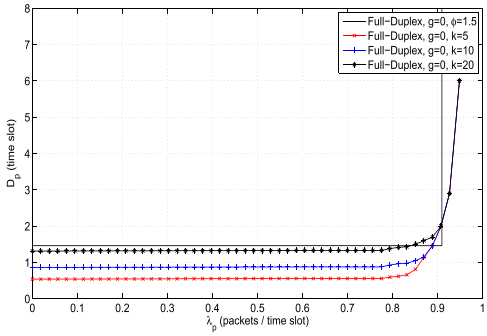

FIGURE 10. Delay performance for different values of k.

role is more crucial. Fig. 9 demonstrates that as we increase the value of k, the weight factor that controls the trade-off between D_p and λ_s, we become more interested in maximizing λ_s than minimizing D_p. As a result, a larger stability region is achieved; on the other hand, the PU delay performance degrades as will be shown in the following figure.

Fig. 10 plots the delay performance of the soft delay guarantee scheme and compares it with the one shown in Fig. 3. Fig. 10 shows that the PU delay decreases as we increase the value of k as expected. It is clear from this figure that the soft-delay guarantee scheme can achieve much higher maximum stable throughput for the PU at the expense of the delay. Therefore, it can be concluded that none of the proposed schemes is better in general over the range of PU arrival rates. Each one can outperform the other schemes under certain conditions. Choosing between them depends on the application and the QoS requirements of the PU. For example, when the PU traffic is delay-sensitive, with a strict delay requirement, such as gaming and video streaming, it is preferable to use the scheme with a strict PU delay constraint. On the other hand, if the PU uses delay insensitive traffic, such as web browsing, it could be better to use the soft-delay guarantee scheme as it supports high maximum stable λ_p with a finite PU delay performance.

In the following four figures, we illustrate the effect of sensing errors on the overall performance of the network; more specifically, the stability region and the maximum stable λ_p where $\lambda_s = 0$. Obviously, due to false alarms, the SU can sense the existence of the PU while in fact the PU is

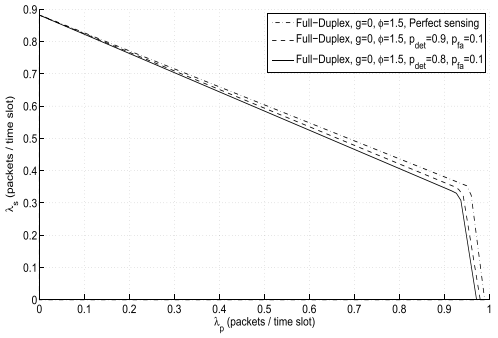

FIGURE 11. SU sensing error impact for good direct channel gain p-d and good direct channel gain s-d.

idle with probability p_{fa}, which is the false alarm probability. Hence, this causes the SU to use p_s^{busy} and p_{sp}^{busy} to make access decisions instead of p_s^{idle} and p_{sp}^{idle}. On the other hand, the SU can miss-detect the presence of the PU (i.e., the PU is sensed to be idle) while it is transmitting a packet. Consequently, the SU makes its access decisions using p_s^{idle} and p_{sp}^{idle} instead of p_s^{busy} and p_{sp}^{busy}. Hence, sensing errors can, in general, result in notable degradation in the stability region.

In Fig. 11, we consider the case of good channel for both the $p-d$ and $s-d$ links. For this scenario, most of the packets are delivered directly from p to d without the cooperation from s. Hence, we achieve the maximum stable λ_p with the optimized parameters $p_{sp}^{\text{idle}} = 1$, $p_s^{\text{busy}} = 0$ and $p_{sp}^{\text{busy}} = 0.94$ with perfect sensing. Since, the values of p_{sp}^{idle} and p_{sp}^{busy} are close to each other, the introduction of sensing error (p_{fa} and p_{det}) at the SU does not significantly reduce the maximum stable λ_p as shown in the figure. In fact, we found that in the case of $p_{\text{fa}} = 0.1$ and $p_{\text{det}} = 0.9$, we achieve the maximum stable λ_p with the optimized parameters $p_{sp}^{\text{idle}} = 1$, $p_s^{\text{busy}} = 0$ and $p_{sp}^{\text{busy}} = 0.91$ where the decrease in the value of p_{sp}^{busy} compared to that in the perfect sensing case is due to sensing errors at the SU.

Not only the maximum stable λ_p is almost the same for the perfect or non-perfect sensing cases, for the scenarios that we have simulated, but also the overall stability region is almost the same for both cases. The rationale behind this is that the optimal values of p_s^{busy} and p_{sp}^{busy} are close to the optimal values of p_s^{idle} and p_{sp}^{idle}, respectively, for the perfect sensing case. Consequently, introducing sensing errors in the model does not significantly change the access decisions taken by the SU, and hence, almost the same performance is achieved.

For $\lambda_p = 0.9$, perfect sensing and imperfect sensing (with $p_{\text{fa}} = 0.1$ and $p_{\text{det}} = 0.9$) achieve stable $\lambda_s = 0.39$ and $\lambda_s = 0.37$ with parameters $p_{sp}^{\text{idle}} = 0.14$, $p_s^{\text{busy}} = 0.84$, $p_{sp}^{\text{busy}} = 0.1$, $p_{sp}^{\text{idle}} = 0.19$, $p_s^{\text{busy}} = 0.8$ and $p_{sp}^{\text{busy}} = 0.15$. Compared to the perfect sensing case, the SU dedicated slightly more resources for the PU to maintain a stable $\lambda_p = 0.9$ which cause a slight degradation in the SU stable throughput as shown in the figure.

In Fig. 12, we consider the case where we have low channel gain for $p - d$ and high channel gain for $s - d$,

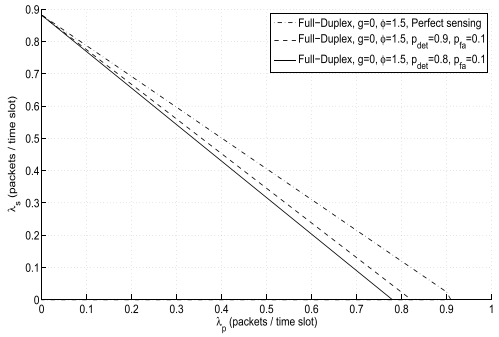

FIGURE 12. SU sensing error impact for bad direct channel gain *p-d* and good direct channel gain *s-d*.

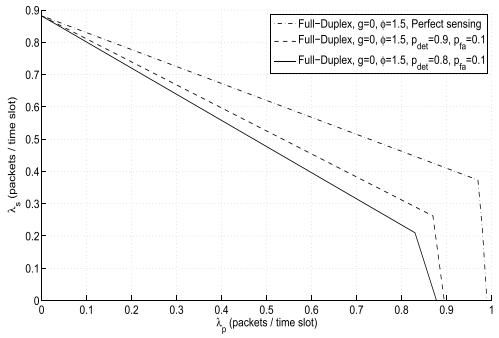

FIGURE 13. SU sensing error impact for high direct channel gain *p-d* and low direct channel gain *s-d*.

i.e., the cooperation of the SU is very important to the PU as explained before. The degradation in the stability region, shown in Fig. 12, for the non-perfect sensing case compared to the perfect sensing case is mainly due to the reduction in the maximum stable λ_p. Here, the sensing error causes a slight reduction in the maximum stable λ_p, unlike the previous case in Fig. 11. The rationale behind this is that the optimal parameters that achieve the maximum stable λ_p for the perfect sensing is $p_{sp}^{\text{idle}} = 1$, $p_s^{\text{busy}} = 0$ and $p_{sp}^{\text{busy}} = 0.89$. Due to the slight difference between p_{sp}^{idle} and p_{sp}^{busy}, adding sensing error at the SU causes slight changes in the decisions taken by the SU which, in turn, reduces the maximum stable λ_p. It can be concluded that for the previous two scenarios our cooperation policy gives good performance in the presence of sensing errors, and shows a reasonable resistance to the sensing errors.

For Fig. 13, we consider the case where we have a high channel gain for $p-d$ and a low channel gain for $s-d$. For this case we have a moderate degradation in the maximum stable λ_p due to sensing errors. This is because the maximum stable λ_p for the perfect sensing case is obtained with $p_{sp}^{\text{idle}} = 1$, $p_s^{\text{busy}} = 0$ and $p_{sp}^{\text{busy}} = 0.83$. Due to the difference between p_{sp}^{idle} and p_{sp}^{busy}, only a moderate degradation in the maximum stable λ_p occurs due sensing errors. We achieve the corner point ($\lambda_p = 0.97, \lambda_s = 0.37$) in the stability region of perfect sensing with $p_{sp}^{\text{idle}} = 0.09$, $p_s^{\text{busy}} = 0.73$ and $p_{sp}^{\text{busy}} = 0.15$. Due to the moderate differences between $p_{sp}^{\text{idle}}, p_{sp}^{\text{busy}}$ and $p_s^{\text{idle}}, p_s^{\text{busy}}$, a notable degradation in the stability region

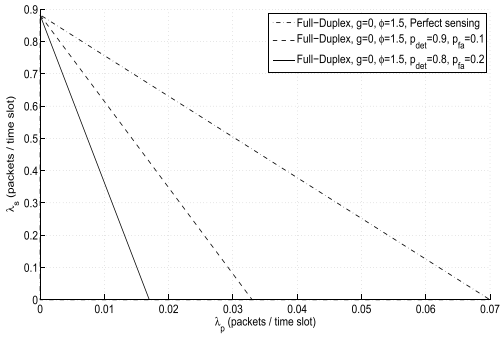

FIGURE 14. SU sensing error impact for low direct channel gain *p-d* and low direct channel gain *s-d*.

occurred when sensing errors are introduced. Introducing sensing errors in the model, in this scenario, makes the SU take wrong access decisions and as a result the overall performance degrades.

In Fig. 14, we consider the case where we have two low channel gains over the links $p-d$ and $s-d$. Clearly, from the figure, the system performance has degraded significantly. This is expected because from Fig. 5 even if the sensing errors do not exist, the low channel gains between $p-d$ and $s-d$ result in a bad performance.

VI. CONCLUSION

In this paper, a cooperative CR network has been studied with the objective of maximizing the SU throughput subject to a PU delay constraint. The impact of having a full-duplex capability at the SU on the network performance has been also investigated compared to the case of having a half-duplex SU. We formulate an optimization problem to maximize the throughput of the SU subject to a PU delay constraint, which was demonstrated to be non-convex. We propose an efficient approach to solve the optimization problem by iterating over a set of non-convex QCQP optimization problems; we use the FPP-SCA algorithm to solve the problem in each iteration. Unexpectedly, our numerical results have shown that having a full-duplex capability at the SU is not always beneficial and it can negatively affect the network stability region in some scenarios. Moreover, our proposed policy has been demonstrated to outperform the no-cooperation and other cooperation policies that have been previously presented in terms of the achieved stability region. In addition, we have formulated a multi-objective programming optimization problem to investigate the trade-off between the PU delay and the SU throughput.

APPENDIXES

APPENDIX A

DERIVATION OF U_{MN}^{I} AND V_{MN}^{I}

Let W and Z be two independent exponentially distributed random variables characterized by the parameters β_1 and β_2, respectively; their probability density functions (PDFs) are denoted by $p(w)$ and $p(z)$. We assume that W and Z denote the squared absolute channel gains for a multiple access

channel (MAC). We assume that the two transmitters are transmitting using the same rate R and this allows us to define the following no-outage event.

$$\begin{aligned}\Re(\alpha_1,\alpha_2) &= \{(w,z): w>\alpha_1 \cap z>\alpha_1 \cap z+w>\alpha_2\}\\ &= \{(w,z): w>\alpha_1 \cap z>\max[\alpha_2-w,\alpha_1]\}\\ &= \{(w,z): w>\alpha_2-\alpha_1 \cap z>\alpha_1\}\\ &\cup \{(w,z): \alpha_2-\alpha_1 \geq w \geq \alpha_1 \cap z>\alpha_2-w\}, \quad (23)\end{aligned}$$

where α_1 and α_2 are two deterministic functions of the communication rate. The term u^I_{mn} denotes the probability that node n successfully decodes both packets transmitted from nodes m and I, and can be found by integrating the joint PDF of W and Z over the previously defined set in (23) as follows:

$$\begin{aligned}&U(\alpha_1,\alpha_2,\beta_1,\beta_2)\\ &= \iint_{\Re(\alpha_1,\alpha_2)} p(w,z)dwdz\\ &= \int_{\alpha_2-\alpha_1}^{\infty} p(w)\int_{\alpha_1}^{\infty} p(z)\ dzdw\\ &\quad + \int_{\alpha_1}^{\alpha_2-\alpha_1} p(w)\int_{\alpha_2-w}^{\infty} p(z)\ dzdw\\ &= \frac{\beta_1 e^{\beta_2\alpha_2}}{\beta_1-\beta_2}\{\exp\left(-\alpha_1(\beta_1-\beta_2)\right)-\exp(-(\alpha_2-\alpha_1)(\beta_1-\beta_2))\}\\ &\quad + \exp(\alpha_1(\beta_2-\beta_1)+\alpha_2\beta_1).\end{aligned}$$

Hence, u^I_{mn} can be expressed as follows:

$$u^I_{mn} = U\left(\frac{2^R-1}{P}, \frac{2^{2R}-1}{P}, \frac{1}{\rho^2_{m,n}}, \frac{1}{\rho^2_{I,n}}\right). \quad (24)$$

Next we derive an expression for v^I_{mn} which denotes the probability that node n successfully decodes the packet transmitted from node m by considering the interference caused by the transmission of node I as noise. We first define the $V(\cdot,\cdot,\cdot)$ as follows

$$\begin{aligned}V(\alpha_1,\beta_1,\beta_2) &= \mathbb{P}\left\{R<\frac{1}{2}\log(1+\frac{PW}{PZ+1})\right\}\\ &= \mathbb{P}\left\{\frac{(2^{2R}-1)(PZ+1)}{P}<W\right\}\\ &= \int_0^{\infty} p(z)\int_{\frac{(2^{2R}-1)(Pz+1)}{P}}^{\infty} p(w)\ dwdz.\end{aligned}$$

Note that this term $\int_{\frac{(2^{2R}-1)(Pz+1)}{P}}^{\infty} p(w)\ dw$ represents the complementary cumulative distribution function (CCDF) of W at $\frac{(2^{2R}-1)(Pz+1)}{P}$. Since W is assumed to be an exponential random variable with parameter β_1, the CCDF of W at $\frac{(2^{2R}-1)(Pz+1)}{P}$ can be expressed as $\exp\left(-\frac{(2^{2R}-1)(Pz+1)\beta_1}{P}\right)$. As a result, we have:

$$\begin{aligned}V(\alpha_1,\beta_1,\beta_2) &= \int_0^{\infty} p(z)\exp\left(-\frac{(2^{2R}-1)(Pz+1)\beta_1}{P}\right)dz\\ &= \frac{\beta_2\exp(-\alpha_1\beta_1)}{\beta_2+P\alpha_1\beta_1}.\end{aligned}$$

Consequently, v^I_{mn} can be expressed as

$$v^I_{mn} = V\left(\frac{2^{2R}-1}{P}, \frac{1}{\rho^2_{m,n}}, \frac{1}{\rho^2_{I,n}}\right). \quad (25)$$

APPENDIX B
NON-CONVEXITY PROOF OF D_p

As given in (13), D_p is given by

$$\begin{aligned}D_p &= \frac{1-\lambda_p}{\mu_p-\lambda_p} + \left(1-\frac{\boldsymbol{a_1}^T\boldsymbol{\alpha}+c_1}{\mu_p}\right)\left(\frac{1-\lambda_{sp}}{\mu_{sp}-\lambda_{sp}}\right) \quad (26)\\ &= \frac{1-\lambda_p}{\mu_p-\lambda_p} + \left(1-\frac{\boldsymbol{a_1}^T\boldsymbol{\alpha}+c_1}{\mu_p}\right).\\ &\quad \left(\frac{\mu_p-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_2}{\boldsymbol{a}_3^T\boldsymbol{\alpha}+\left(\mu_p-\lambda_p\right)\boldsymbol{a_4}^T\boldsymbol{\alpha}-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_3}\right) \quad (27)\end{aligned}$$

where $\boldsymbol{\alpha} = [p_s^{\text{busy}}, p_{sp}^{\text{busy}}, p_{sp}^{\text{idle}}]$ is a 3−dimensional column vector and all the $\boldsymbol{a_i}^T$'s are 3-dimensional column vectors that are deterministic functions of the channel parameters, p_{det} and p_{fa}. Moreover, all the c_i's are constants that are functions of the channel parameters, p_{det} and p_{fa}. Note that all $\boldsymbol{a_i}^T$'s and c_i's can be directly obtained from (6), (7), and (13). As given in (13), D_p consists of two parts as follows. The first part is T_p which is a linear fractional function of a linear function of the optimization parameters (as μ_p is a linear function of the optimization parameters). The second part is $(1-\tau)T_{sp}$, where $1-\tau$ is linear fractional function and T_{sp} is a linear function divided by a quadratic function of the optimization parameters (which is not convex in general [23]). Hence, D_p is, in general, a non-convex function of the optimization parameters.

APPENDIX C
THE OPTIMIZATION PROBLEM FOR A FIXED μ_p

In this Appendix, We analyze the effect of fixing μ_p on our two optimization problems in **P1** and **P2**; more specifically, the first and second constraints $\lambda_s \leq \mu_s$ and $D_p \leq \phi$ in **P1**, and the first constraint, $D_p - k\lambda_s \leq t$, in **P2**.

Note that from (11), μ_s can be written as

$$\begin{aligned}\mu_s &= \frac{\boldsymbol{a_5}^T\boldsymbol{\alpha}}{\mu_p} + \left(\frac{\mu_p-\lambda_p}{\mu_p}\right)(\boldsymbol{a_6}^T\boldsymbol{\alpha}+c_4)\\ &\quad + \frac{\boldsymbol{a_7}^T\boldsymbol{\alpha}+c_5}{\mu_p} + \left(\frac{\mu_p-\lambda_p}{\mu_p}\right)(\boldsymbol{a_8}^T\boldsymbol{\alpha}). \quad (28)\\ &= \frac{1}{\mu_p}\Big(\left((\boldsymbol{a_5}+\boldsymbol{a_7})^T\boldsymbol{\alpha}+c_5\right)\\ &\quad + (\mu_p-\lambda_p)\left((\boldsymbol{a_6}+\boldsymbol{a_8})^T\boldsymbol{\alpha}+c_4\right)\Big), \quad (29)\end{aligned}$$

where $\boldsymbol{\alpha} = [p_s^{\text{busy}}, p_{sp}^{\text{busy}}, p_{sp}^{\text{idle}}]$ is a 3−dimensional column vector and all the $\boldsymbol{a_i}^T$'s are 3-dimensional column vectors that are deterministic functions of the channel parameters, p_{det} and p_{fa}. Moreover, all the c_i's are constants that are functions of the channel parameters, p_{det} and p_{fa}. Note that all $\boldsymbol{a_i}^T$'s and c_i's can be directly obtained from (6), (7), and (13) (as explained above).

Clearly, for a constant μ_p, μ_s becomes a linear function of $\boldsymbol{\alpha}$.

On the other hand, from (27), the constraint $D_p \leq \phi$ can be written as

$$\frac{1-\lambda_p}{\mu_p-\lambda_p}+\left(1-\frac{\boldsymbol{a_1}^T\boldsymbol{\alpha}+c_1}{\mu_p}\right).\left(\frac{\mu_p-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_2}{\boldsymbol{a_3}^T\boldsymbol{\alpha}+\left(\mu_p-\lambda_p\right)\boldsymbol{a_4}^T\boldsymbol{\alpha}-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_3}\right)\leq\phi. \quad (30)$$

By multiplying both sides by $\mu_p(\mu_p-\lambda_p)\left(\boldsymbol{a}_3^T\boldsymbol{\alpha}+\left(\mu_p-\lambda_p\right)\boldsymbol{a_4}^T\boldsymbol{\alpha}-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_3\right)$, we get

$$\begin{aligned}&(1-\lambda_p)\mu_p\left(\boldsymbol{a}_3^T\boldsymbol{\alpha}+\left(\mu_p-\lambda_p\right)\boldsymbol{a_4}^T\boldsymbol{\alpha}-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_3\right)\\&\quad+(\mu_p-\boldsymbol{a_1}^T\boldsymbol{\alpha}-c_1)(\mu_p-\lambda_p).\\&\left(\boldsymbol{a}_3^T\boldsymbol{\alpha}+\left(\mu_p-\lambda_p\right)\boldsymbol{a_4}^T\boldsymbol{\alpha}-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_3\right)\\&\quad+\mu_p(\mu_p-\lambda_p)(\mu_p-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_2)\leq\\&\phi\mu_p(\mu_p-\lambda_p)\left(\boldsymbol{a}_3^T\boldsymbol{\alpha}+\left(\mu_p-\lambda_p\right)\boldsymbol{a_4}^T\boldsymbol{\alpha}-\boldsymbol{a_2}^T\boldsymbol{\alpha}+c_3\right). \quad (31)\end{aligned}$$

For a fixed value of μ_p, we can see that the second term becomes a quadratic term in $\boldsymbol{\alpha}$ while the other terms are linear; hence, this constraint can be written as a quadratic constraint as

$$\boldsymbol{\alpha}\boldsymbol{A}\boldsymbol{\alpha}^T+\boldsymbol{c}^T\boldsymbol{\alpha}+t\leq 0 \quad (32)$$

where $\boldsymbol{A}$ is a 3×3 matrix, $\boldsymbol{c}$ is 3- dimensional column vector, and t is a constant value. The values of $\boldsymbol{A}, \boldsymbol{c}$ and t can be readily derived from (31).

Now, for the first constraint in **P2**, given by $D_p - k\lambda_s \leq t$, let's define $\boldsymbol{\zeta}$ to be a 5-dimensional column vector as $\boldsymbol{\zeta} = [\boldsymbol{\alpha}, \lambda_s, t]$, where $\boldsymbol{\alpha}$ is as defined before in Appendix B. Let's also define the row vector $\boldsymbol{q}$ as $\boldsymbol{q} = [0, 0, 0, k, 1]$. From (27), the constraint $D_p \leq t + k\lambda_s$ can be written as follows:

$$\frac{1-\lambda_p}{\mu_p-\lambda_p}+\left(1-\frac{\boldsymbol{b_1}^T\boldsymbol{\zeta}+j_1}{\mu_p}\right).\left(\frac{\mu_p-\boldsymbol{b_2}^T\boldsymbol{\zeta}+j_2}{\boldsymbol{b_3}^T\boldsymbol{\zeta}+\left(\mu_p-\lambda_p\right)\boldsymbol{b_4}^T\boldsymbol{\zeta}-\boldsymbol{b_2}^T\boldsymbol{\zeta}+j_3}\right)\leq\boldsymbol{q}^T\boldsymbol{\zeta} \quad (33)$$

where all $\boldsymbol{b_i}^T$'s are 5-dimensional column vectors with each element in them defined as a function of the channel parameters, p_{det} and p_{fa}; all j_i's are scalar functions of the channel parameters, p_{det} and p_{fa}. The terms $\boldsymbol{b_i}^T$'s and j_i's can be readily derived from (6), (7), and (13). Note that the term $\boldsymbol{q}^T\boldsymbol{\zeta}$, which is the right hand side in the above inequality, is the vector representation of the term $t + k\lambda_s$ in the original constraint $D_p \leq t + k\lambda_s$.

By multiplying both sides by $\mu_p(\mu_p-\lambda_p)\left(\boldsymbol{b}_3^T\boldsymbol{\zeta}+\left(\mu_p-\lambda_p\right)\boldsymbol{b_4}^T\boldsymbol{\zeta}-\boldsymbol{b_2}^T\boldsymbol{\zeta}+j_3\right)$, (33) can be written as follows:

$$\begin{aligned}&(1-\lambda_p)\mu_p\left(\boldsymbol{b}_3^T\boldsymbol{\zeta}+\left(\mu_p-\lambda_p\right)\boldsymbol{b_4}^T\boldsymbol{\zeta}-\boldsymbol{b_2}^T\boldsymbol{\zeta}+j_3\right)\\&\quad+(\mu_p-\boldsymbol{b_1}^T\boldsymbol{\zeta}-j_1)(\mu_p-\lambda_p).\\&\left(\boldsymbol{b}_3^T\boldsymbol{\zeta}+\left(\mu_p-\lambda_p\right)\boldsymbol{b_4}^T\boldsymbol{\zeta}-\boldsymbol{b_2}^T\boldsymbol{\zeta}+j_3\right)\\&\quad+\mu_p(\mu_p-\lambda_p)(\mu_p-\boldsymbol{b_2}^T\boldsymbol{\zeta}+j_2)\leq\\&(\boldsymbol{q}^T\boldsymbol{\zeta})\mu_p(\mu_p-\lambda_p)\left(\boldsymbol{b_3}^T\boldsymbol{\zeta}+\left(\mu_p-\lambda_p\right)\boldsymbol{b_4}^T\boldsymbol{\zeta}-\boldsymbol{b_2}^T\boldsymbol{\zeta}+j_3\right). \quad (34)\end{aligned}$$

Now, we can see that, for a fixed μ_p, the right hand side and the second term in the left hand side in (34) become quadratic functions in ζ (which contains all the optimization parameters), while the other terms are linear. As a result, **P2** turns out to be a QCQP optimization problem.

ACKNOWLEDGMENT

This article was presented in part at the 14th International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM 2019), Poznan, Poland, June 2019 [1].

REFERENCES

[1] A. Gaber, E.-S. Youssef, M. R. Rizk, M. Salman, and K. G. Seddik, "Cooperative delay-constrained cognitive radio networks: Throughput maximization with full-duplex capability impact," in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw.* Cham, Switzerland: Springer, 2019, pp. 180–194.
[2] X. Tao, X. Xu, and Q. Cui, "An overview of cooperative communications," *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 65–71, Jun. 2012.
[3] B. Wang and K. J. R. Liu, "Advances in cognitive radio networks: A survey," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 5–23, Feb. 2011.
[4] M. Ibnkahla, *Cooperative Cognitive Radio Networks: The Complete Spectrum Cycle*. Boca Raton, FL, USA: CRC Press, 2014.
[5] K. B. Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," *Proc. IEEE*, vol. 97, no. 5, pp. 878–893, May 2009.
[6] A. K. Sadek, K. R. Liu, and A. Ephremides, "Cognitive multiple access via cooperation: Protocol design and performance analysis," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3677–3696, Oct. 2007.
[7] B. Rong and A. Ephremides, "Cooperative access in wireless networks: Stable throughput and delay," *IEEE Trans. Inf. Theory*, vol. 58, no. 9, pp. 5890–5907, Sep. 2012.
[8] M. Ashour, A. A. El-Sherif, T. Elbatt, and A. Mohamed, "Cognitive radio networks with probabilistic relaying: Stable throughput and delay trade-offs," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4002–4014, Nov. 2015.
[9] M. A. Abd-Elmagid, T. Elbatt, K. G. Seddik, and O. Ercetin, "Stable throughput of cooperative cognitive networks with energy harvesting: Finite relay buffer and finite battery capacity," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 704–718, Dec. 2018.
[10] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wichman, "In-band full-duplex wireless: Challenges and opportunities," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 9, pp. 1637–1652, Sep. 2014.
[11] N. Pappas, A. Ephremides, and A. Traganitis, "Stability and performance issues of a relay assisted multiple access scheme with MPR capabilities," *Comput. Commun.*, vol. 42, pp. 70–76, Apr. 2014.

[12] S. Elazzouni, O. Ercetin, A. El-Keyi, T. Elbatt, and M. Nafie, "Full-duplex cooperative cognitive radio networks," in *Proc. 13th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2015, pp. 475–482.
[13] A. M. Elmahdy, A. El-Keyi, T. Elbatt, and K. G. Seddik, "Optimizing cooperative cognitive radio networks performance with primary QoS provisioning," *IEEE Trans. Commun.*, vol. 65, no. 4, pp. 1451–1463, Apr. 2017.
[14] M. Salman, A. El-Keyi, M. Nafie, and M. Hasna, "Novel cooperative policy for cognitive radio networks: Stability region and delay analysis," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2016, pp. 1–7.
[15] A. H. Anwar, K. G. Seddik, T. Elbatt, and A. H. Zahran, "Effective capacity of delay-constrained cognitive radio links exploiting primary feedback," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7334–7348, Sep. 2016.
[16] M. A. Abdel-Malek, K. G. Seddik, T. ElBatt, and Y. Mohasseb, "Effective capacity optimization for cognitive radio networks under primary QoS provisioning," *Wireless Netw.*, pp. 1–20, 2019.
[17] B. S. Tsybakov and V. A. Mikhailov, "Ergodicity of a slotted ALOHA system," *Problemy Peredachi Informatsii*, vol. 15, no. 4, p. 73–87, 1979.
[18] A. M. Elmahdy, A. El-Keyi, T. Elbatt, and K. G. Seddik, "On the stable throughput of cooperative cognitive radio networks with finite relaying buffer," in *Proc. IEEE 25th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2014, pp. 942–946.
[19] D. Ramirez and B. Aazhang, "Optimal routing and power allocation for wireless networks with imperfect full-duplex nodes," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4692–4704, Sep. 2013.
[20] I. Krikidis, N. Devroye, and J. Thompson, "Stability analysis for cognitive radio with multi-access primary transmission," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 72–77, Jan. 2010.
[21] R. M. Loynes, "The stability of a queue with non-independent inter-arrival and service times," *Math. Cambridge Philos. Soc.*, vol. 58, no. 3, pp. 497–520, 1962.
[22] J. R. Jackson, "Jobshop-like queueing systems," *Manage. Sci.*, vol. 50, no. 12, pp. 1796–1802, Dec. 2004.
[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[24] O. Mehanna, K. Huang, B. Gopalakrishnan, A. Konar, and N. D. Sidiropoulos, "Feasible point pursuit and successive approximation of non-convex QCQPs," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 804–808, Jul. 2015.
[25] K. M. Anstreicher, "On convex relaxations for quadratically constrained quadratic programming," *Math. Program.*, vol. 136, no. 2, pp. 233–251, Dec. 2012.
[26] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *J. Global Optim.*, vol. 47, no. 1, pp. 29–51, May 2010.
[27] L.-N. Tran, M. F. Hanif, and M. Juntti, "A conic quadratic programming approach to physical layer multicasting for large-scale antenna arrays," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 114–117, Jan. 2014.

ALI GABER MOHAMED ALI received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering from Alexandria University, Egypt, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree. From 2012 to 2014, he worked as a Research Assistant with American University, Cairo, Egypt. From 2015 to 2017, he was a Graduate Research and Teaching Assistant with Virginia Tech, VA, USA. Since 2017, he has been with the Department of Electrical Engineering, Alexandria University, where he is currently an Assistant Lecturer. His current research areas include smart grid communications, localization, cognitive radio, signal processing, and communication systems.

EL-SAYED AHMED YOUSSEF received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Egypt, in 1964 and 1967, respectively, and the Ph.D. degree in electrical engineering from the University of Alberta, Canada, in 1972. He is currently working as a Professor with the Department of Electrical Engineering, Alexandria University. His research interests include digital communications, signal processing, image, and video processing.

MOHAMED R. M. RIZK received the B.Sc. degree in electrical engineering from Alexandria University, Egypt, in 1971, and the M.Sc. and Ph.D. degrees in electrical engineering from McMaster University, Canada, in 1975 and 1979, respectively. He worked as an Assistant Professor with McMaster University. He was a Visiting and an Associate Professor with Sultan Qaboos University, Oman, Beirut Arab University, Lebanon, and the Arab Academy for Science and Technology. He was an Adjunct Professor with Virginia Polytechnic and State University, VA, USA. He is currently an Associate Professor with the Department of Electrical Engineering, Alexandria University. He has authored or coauthored over 30 journals and approximately 70 conference papers. His research interests include communication systems, computer aided design, wireless communications, encryption, fuzzy logic, image processing, and computer networks.

MOHAMED SALMAN (Student Member, IEEE) received the B.Sc. degree (Hons.) in electrical engineering from Alexandria University, Alexandria, Egypt, in 2012, and the M.Sc. degree in wireless technologies from Nile University, Giza, Egypt, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical, Computer, and Energy Engineering, University of Colorado at Boulder, Boulder, CO, USA. His research interests include information theory, network optimization, and wireless communications.

KARIM G. SEDDIK (Senior Member, IEEE) received the B.S. (Hons.) and M.S. degrees in electrical engineering from Alexandria University, Alexandria, Egypt, in 2001 and 2004, respectively, and the Ph.D. degree from the Electrical and Computer Engineering Department, University of Maryland, College Park, in 2008.

He is currently a Professor with the Electronics and Communications Engineering Department, American University in Cairo (AUC), Egypt. Before joining AUC, he was an Assistant Professor with the Electrical Engineering Department, Alexandria University. His research interests include applications of machine learning in communication networks, cognitive radio communications, and layered channel coding. He was a recipient of the State Encouragement Award, in 2016, and the Certificate of Honor from the Egyptian President for being ranked first among all departments in the College of Engineering, Alexandria University, in 2002, the Graduate School Fellowship from the University of Maryland, in 2004 and 2005, and the Future Faculty Program Fellowship from the University of Maryland, in 2007. He has served on the technical program committees for numerous IEEE conferences in the areas of wireless networks and mobile computing.