Cross-Layer Scheduling for Cooperative Multi-Hop Cognitive Radio Networks

Dongyue Xue and Eylem Ekici

Abstract—**In this paper, a cross-layer optimal scheduling algorithm for cooperative multi-hop Cognitive Radio Networks (CRNs) is presented, where secondary users (SUs) assist primary users' (PUs') multi-hop transmissions and in return gain an immediate time-share of the channel proportional to their assistance. While providing deterministic upper-bounds for PU queue backlogs, the proposed algorithm approaches the optimal PU throughput arbitrarily close, with a tradeoff in the average delay upper-bounds of flows. The analysis is further extended to a model with a more general "long-term reward mechanism", where a *time-averaged* share of the channel is guaranteed for SUs. The proposed algorithm provides order-optimal delay for the primary traffic. Distributed implementation issues have also been investigated. The properties of the proposed algorithm have also been illustrated through simulation studies.**

Index Terms—**Cognitive radio, optimal scheduling, congestion control, multi-hop wireless networks, finite buffers**

I. Introduction

TRADITIONAL fixed spectrum assignment gives rise to spectrum under-utilization as reported by Federal Communication Commission (FCC) in [1]. Cognitive Radio Networks (CRNs) [2] have recently emerged as a technology for secondary users (SUs) to opportunistically utilize the spectrum assigned to incumbent users, referred to as primary users (PUs). The traditional view on CRNs emphasizes point-to-point connections for both PU and SU subsystems, and multi-hop CRNs have only been considered in recent past. In addition to the majority of CRN solutions in the literature that do not provide provable performance levels, throughput and utility optimization for single-hop CRNs have also been investigated [3]-[6]. However, these works are not readily extendable to multi-hop CRNs.

Back-pressure-based scheduling algorithms have been extensively investigated for generic wireless networks [15][16]. In addition to the seminal work [15], distributed and low-complexity algorithms have been proposed in the literature, including [18][19]. This technique has been applied to CRNs in [11]-[14]. Specifically, in [11], an optimal cross-layer scheduling algorithm has been proposed in a single-hop setting to maximize SU throughput subject to PU collision constraints. This single-hop setting is extended in [12], where aggregated utility is maximized subject to PU power constraints. In [13], a cooperative CRN is considered to optimize PU and SU utility, where SUs assist PU transmission in a two-hop relay scenario, which is not readily extendable to generic multi-hop CRNs and does not involve delay analysis. A multi-hop CRN scheduling algorithm is proposed in [14] with estimated end-to-end delay, but there is no cooperation between PUs and SUs. In addition, the algorithms proposed in [11]-[14] are centralized in general. To the best of our knowledge, no throughput/utility-optimal scheduling algorithms have been proposed in the literature for cooperative multi-hop CRNs with investigations on distributed implementation. Furthermore, characterizing delay upper-bounds for PUs in a CRN is challenging, especially with the opportunistic access of SUs to the same channel.

The general cognitive radio networks perform well when the PU traffic activity is low and SUs find ample opportunities to access the licensed spectrum. In such cases, existing algorithms such as [11][12][14] designed for CRNs can be applied with high SU throughput and low PU-SU collision probability. A challenging situation emerges when PU traffic intensity increases to higher levels. In such cases, SUs not only have fewer transmission opportunities, but also run greater risk of collision with PU transmissions. Here, we propose the application of cooperative communication principles to CRNs that relate the SU channel access rights with the services they render for PUs. As such, unintentional infringements of PU rights of channel access (i.e., collisions of SU transmissions with PU transmissions) are eliminated, which arise in non-cooperative CRNs with possibly small but non-zero probabilities. Moreover, SUs access the crowded licensed spectrum with little or no 'spectral opportunities'.

In this paper, we propose a throughput-optimal cross-layer scheduling algorithm for a multi-hop cooperative CRN under a property-rights model [13], where SUs relay data between PU pairs to gain access to the licensed spectrum. An illustrative example is shown in Figure 1, where the cooperative CRN is composed of an SU subnetwork and a PU subnetwork. The SU subnetwork consists of SUs communicating with a secondary base station over a single hop as assumed for IEEE 802.22. In the PU subnetwork, we consider a case where the channel condition is not desirable for the the direct transmission between the PU and the primary base station due to physical separation. Thus, the PU is willing to "lease" a portion of the spectrum access to SUs in return for some form of service. Specifically, PU data is relayed by SUs from the source PU to PU base station, and SUs in return gain an time-share of the channel proportional to their assistance to the PU. The model illustrated in Figure 1 can be considered as a generalization of the overlay CRNs with two-hop relay [8]-[10]. The proposed

Manuscript received 5 January 2012; revised 14 May 2012 and 12 July 2012. This work was supported in part by NSF grant CCF-0914912.

The authors are with the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, Ohio 43210, USA (e-mail: xued, ekici@ece.osu.edu).

Digital Object Identifier 10.1109/JSAC.2013.130318.

algorithm solves the throughput maximization problem using pre-determined routes with a reward mechanism: the SUs are guaranteed a throughput proportional to the PU data they relay.

An optimal opportunistic scheduling scheme has been proposed in [7] to guarantee each user a proportional share of the network resource for a non-cognitive setting, which is extended to a scenario of two-hop relay CRNs in [13]. Different from the approach employed in [7] which is not readily applicable to a general multi-hop setting, we employ Lyapunov optimization tools to develop the throughput-optimal scheduling algorithm in a multi-hop cooperative CRN. Salient contributions of our work with respect to the literature can be listed as follows: (1) The algorithm can achieve a PU throughput arbitrarily close to the optimal values, with a tradeoff between throughput and PU/SU packet queue length. Specifically, the PU achieves a throughput "$(\epsilon + \frac{1}{V_2})$-close" to the optimal value at a tradeoff of $O(\frac{V_2}{\epsilon})$ in average SU queue length and $O(V_2)$ in the deterministic PU buffer size. (2) The algorithm guarantees deterministically upper-bounded finite buffer sizes for PU queues in the CRN. Derived from the previous two features, we show that the algorithm achieves *order optimal delay* [25] for PU traffic, i.e., the delay is upper-bounded by the first order of the number of hops in a route. (3) Distributed implementation and an extended algorithm with a general long-term reward mechanism are discussed. (4) The immediate and long-term reward mechanisms introduced in the paper, along with the proposed algorithms, provide a novel approach to guarantee SUs' access to the opportunistic channel while avoiding unintentional collisions between PUs and SUs.

The rest of the paper is organized as follows: Section II introduces the network model for the cooperative multi-hop CRN. In Section III, we propose and analyze the throughput-optimal algorithm and its performance. In Section IV, we consider a long-term reward mechanism and discuss the corresponding distributed implementation issues. Numerical results are illustrated in Section V. Finally, we conclude our work in Section VI.

II. Network Model

In this section, we first present the overall multi-hop cooperative CRN model, followed by the analysis of the routing and queuing structure in the CRN.

A. Overall Network Elements and Constraints

In this paper, we consider a time-slotted multi-hop cooperative CRN, as illustrated in Figure 1, where SUs relay PU data in return for the right to use the wireless spectrum. The multi-hop cooperative CRN in question can be divided into two subnetworks: a "PU relay subnetwork" and an "SU subnetwork". The PU relay subnetwork is composed of one primary source node (s_P), a corresponding primary destination node (d_P) which is represented as a primary base station in Figure 1, and a set of SUs $\mathcal{S}$ that relay the PU traffic between s_P and d_P over possibly multiple hops, where $\mid \mathcal{S} \mid= N$. This model can be considered as a generalization of the overlay CRNs with two-hop relay [8]-[10]. We assume that s_P and d_P cannot communicate directly. Thus, PU data is

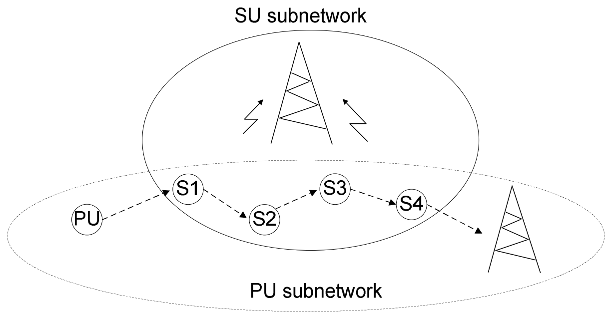


Fig. 1. Cooperative CRN model

relayed solely by SUs. The PU relay subnetwork is represented as ($\mathcal{N}$, $\mathcal{L}$) where $\mathcal{N} = \{s_P, d_P\} \cup \mathcal{S}$ denotes the node set of the PU relay subnetwork and $\mathcal{L}$ denotes the link set for PU data relay, i.e., $\mathcal{L} = \{(m,n) : m,n \in \mathcal{N},$ and there exists a link between nodes m and $n\}$.

We consider an SU subnetwork similar to the architecture presented in IEEE 802.22: The SU subnetwork is composed of a set of SUs $\mathcal{S}$ and a secondary base station d_S as their one-hop destination. Then, the SU subnetwork can be represented by $(\mathcal{S} \cup \{d_S\}, \mathcal{L}')$, where $\mathcal{L}' = \{(l, d_S)\colon l \in \mathcal{S}\}$ is the set of uplinks in the SU subnetwork. Note that our analysis can readily be extended to cases where downlinks are also considered. [1]

Let $\mathcal{V} = \mathcal{L} \cup \mathcal{L}'$. We represent the CRN interference model by an interference graph $G = (\mathcal{V}, \mathcal{E})$ (a.k.a. conflict graph). A pair of links in $\mathcal{V}$ is in $\mathcal{E}$ if the links interfere with each other when scheduled simultaneously. Furthermore, let μ_{mn} be the scheduled link rate for PU data over link $(m,n) \in \mathcal{L}$, and the scheduled SU link rate denoted as s_l over link $(l, d_S) \in \mathcal{L}'$. For analytical simplicity, we assume a scheduled link rate takes a value from $\{0,1\}$ in units of packets per time slot. A link schedule represented by a vector $((\mu_{mn})_{(m,n)\in\mathcal{L}}, (s_l)_{l\in\mathcal{S}}) \in \{0,1\}^{|\mathcal{L}|+N}$ is said to be *feasible* iff any pair of scheduled links does not belong to the interference edge set $\mathcal{E}$. With a time slot system, a feasible link scheduler chooses a feasible link schedule $((\mu_{mn}(t))_{(m,n)\in\mathcal{L}}, (s_l(t))_{l\in\mathcal{S}}) \in \mathcal{I}$ for each time slot t, where $\mathcal{I}$ is the set of all feasible link schedules. We also assume that a node is equipped with one transceiver capable of communicating with only one neighbor in a time slot. Specifically, $\forall n \in \mathcal{N}\backslash\{s_P\}$, the following inequality holds:

$$\sum_{j:(j,n)\in\mathcal{L}} \mu_{jn}(t) + \sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}(t) + \mathbf{1}_{\{n\in\mathcal{S}\}} s_n(t) \leq 1, \ \forall t, \quad (1)$$

where $\mathbf{1}_{\{x\}}$ is the indicator function for event x. In addition, since s_P is the sender of the PU pair, we must have $\sum_{n\in\mathcal{S}} \mu_{ns_P}(t) = 0, \ \forall t$.

In the following subsection, we build a routing and queuing model for the CRN for a fixed-routing scenario.

[1]If downlinks are also available for the SU network, the SU base station d_S can participate in the PU data relay. Specifically, we can update the PU relay network as $(\bar{\mathcal{N}}, \bar{\mathcal{L}})$, where $\bar{\mathcal{N}}$ denotes the node set of PU relay subnetwork including the SU base station, i.e., $\bar{\mathcal{N}} = \mathcal{N} \cup \{d_S\}$, and $\bar{\mathcal{L}}$ denotes the link set for PU data relay including SU uplinks and downlinks, i.e., $\bar{\mathcal{L}} = \mathcal{L} \cup \mathcal{L}' \cup \{(d_S, l)\colon l \in \mathcal{S}\}$. Then the following discussion still holds for the updated PU relay network $(\bar{\mathcal{N}}, \bar{\mathcal{L}})$.

B. Routing and Queueing Structure for the CRN Model

In the cooperative CRN, SUs are rewarded with a throughput proportional to PU data they relay. Thus, we consider a fixed multi-path routing scenario, where the PU data transmission has K loop-free pre-determined routes. We denote the path for the k-th route as $P_k = (v_k^0, v_k^1, \cdots, v_k^{H_k}, v_k^{H_k+1})$, where (H_k+1) is the total number of hops in the PU relay subnetwork for route k, and $v_k^m \in \mathcal{N}$, $\forall m \in \{0, 1, \cdots, H_k+1\}$, $\forall k \in \{1, \cdots, K\}$. Without loss of generality, we assume that each SU $l \in \mathcal{S}$ is in at least one of the K routes, that is: $\forall l \in \mathcal{S}, \exists k, m$ s.t. $v_k^m = l$. Note that we always have $v_k^0 = s_P$ and $v_k^{H_k+1} = d_P, \forall k \in \{1, \cdots, K\}$. According to this routing structure, we construct PU packet queues $U_m^k(t)$ in the nodes along the k-th route, where $0 \le m \le H_k+1$ and $1 \le k \le K$. Note that, since $v_k^{H_k+1} = d_P$, we have $U_{H_k+1}^k(t) = 0$, $\forall t$, $\forall k \in \{1, \cdots, K\}$.

In the cross-layer CRN model, PU and SU packets are generated (by specific applications) at the transport layer and admitted to the network (link) layer. For analytical simplicity, we assume that PU and SU traffics are constantly backlogged at the transport layer. Let $\mu_{-1,0}^k(t)$ be the admitted arrival rate from the PU transport layer to the source PU (s_P) that is scheduled to pass through the k-th route, where we also assume the sum of admitted PU arrival rates over K routes is upper-bounded by μ_M, i.e, $\sum_{k=1}^K \mu_{-1,0}^k(t) \le \mu_M$, $\forall t$, and λ_k, $k = \{1, \cdots, K\}$, is the time-average of $\mu_{-1,0}^k(t)$.

To characterize the route-based queue evolution, let $\mu_{m,m+1}^k(t)$, $0 \le m \le H_k$, be the scheduled rate for the hop (v_k^m, v_k^{m+1}) along the k-th path. Note that a hop schedule $(\mu_{m,m+1}^k(t))_{m,k}$ corresponds to a PU link schedule:

$$\sum_{m,k} \mu_{m,m+1}^k(t) \mathbf{1}_{\{(v_k^m, v_k^{m+1})=(l,n)\}} = \mu_{ln}(t), \forall (l,n) \in \mathcal{L}. \quad (2)$$

Then, the queue $U_m^k(t)$ evolves as follows, $0 \le m \le H_k$,

$$U_m^k(t+1) \le [U_m^k(t) - \mu_{m,m+1}^k(t)]^+ + \mu_{m-1,m}^k(t), \quad (3)$$

where the inequality holds if $\mu_{m-1,m}^k(t) = 1$ and $U_{m-1}^k(t) = 0$, $1 \le m \le H_k$, i.e, when the transmission over the m-th hop is scheduled but not performed due to lack of PU packets at queue $U_{m-1}^k(t)$. Note that a link $(m,n) \in \mathcal{L}$ can be a hop in multiple routes, and hence we can only schedule the hop with rate 1 on at most one such route in any time slot.

We consider an ***immediate reward mechanism*** for SUs in the CRN model, i.e., when a PU packet is admitted to the network, an appropriate number of SU packets are also admitted in all SUs along the path relaying the admitted PU packet. Specifically, let ρ_k be the rate of reward for SUs when a PU packet is admitted to route k, i.e., $\rho_k \mu_{-1,0}^k(t)$ packets from the SU transport layer will be admitted simultaneously to the SU queues corresponding to the nodes v_k^m, $1 \le m \le H_k$. Here, we assume that $\rho_k \mu_{-1,0}^k(t)$ takes integer values. We extend our analysis to fractional-valued $\rho_k \mu_{-1,0}^k(t)$ in Section IV, where we provide the discussion of a long-term reward mechanism.

Let $Q_l(t)$ be the SU packet queue backlog at node $l \in \mathcal{S}$. With an abuse of notation, we denote $l \in P_k$ if the SU l is on the route P_k, i.e., $\exists m$: $v_k^m = l$. Then, the SU queue dynamics for $Q_l(t)$ can be expressed as follows:

$$\begin{aligned} Q_l(t+1) &= [Q_l(t) - s_l(t)]^+ + \sum_{k=1}^K \sum_{m=1}^{H_k} \rho_k \mu_{-1,0}^k(t) \mathbf{1}_{\{v_k^m = l\}} \\ &= [Q_l(t) - s_l(t)]^+ + \sum_{k=1}^K \rho_k \mu_{-1,0}^k(t) \mathbf{1}_{\{l \in P_k\}}, \end{aligned} \quad (4)$$

where the second equality holds since each route is loop-free.

With the above introduced routing and queuing structures, we say the network is *stable* if queues $U_m^k(t)$ and $Q_l(t)$ are stable $\forall m, k$ $\forall l$ simultaneously. A generic queue with backlog $X(t)$ is said to be *stable* if $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{X(t)\} < \infty$. Furthermore, we can define the capacity region Λ_E of the CRN as the closure of all feasible arrival rate vectors each stabilizable by some scheduler. Note that a feasible arrival rate vector is in the form of $((\lambda_k)_{k\in\{1,\cdots,K\}}, (\sum_{k=1}^K \rho_k \lambda_k \mathbf{1}_{\{l\in P_k\}})_{l\in\mathcal{S}})$, where $(\lambda_k)_{k\in\{1,\cdots,K\}}$ represents the PU arrival rates per route and $(\sum_{k=1}^K \rho_k \lambda_k \mathbf{1}_{\{l\in P_k\}})_{l\in\mathcal{S}}$ represents the SU arrival rates according to the reward mechanism. To assist the analysis, we let $(\lambda_{k,\epsilon}^*)_{k\in\{1,\cdots,K\}}$ be a solution to the following optimization problem:

$$\begin{aligned} &\max_{(\lambda_k)_{k\in\{1,\cdots,K\}}} \sum_{k=1}^K \lambda_k \\ &\text{s.t. } (\lambda_k) : ((\lambda_k + \epsilon), (\sum_{k=1}^K \rho_k(\lambda_k + \epsilon)\mathbf{1}_{\{l\in P_k\}})) \in \Lambda_E \end{aligned} \quad (5)$$

where $\epsilon > 0$ can be chosen arbitrarily small. According to [17], we have: $\lim_{\epsilon\to 0^+} \sum_{k=1}^K \lambda_{k,\epsilon}^* = \sum_{k=1}^K \lambda_k^*$, where $(\lambda_k^*)_{k\in\{1,\cdots,K\}}$ is a solution to the following optimization:

$$\begin{aligned} &\max_{(\lambda_k)_{k\in\{1,\cdots,K\}}} \sum_{k=1}^K \lambda_k \\ &\text{s.t. } (\lambda_k) : ((\lambda_k), (\sum_{k=1}^K \rho_k \lambda_k \mathbf{1}_{\{l\in P_k\}})) \in \Lambda_E \end{aligned} \quad (6)$$

Note that $\sum_{k=1}^K \lambda_{k,\epsilon}^*$ can be regarded as the PU throughput ϵ-close to the optimality $\sum_{k=1}^K \lambda_k^*$.

In Section III, we will propose a cross-layer scheduling algorithm that ensures finite PU buffer size and achieves a PU throughput arbitrarily close to the optimal value $\sum_{k=1}^K \lambda_k^*$, with a tradeoff between the PU throughput and average PU/SU delay upper-bound. We note that delay period of a packet starts when the packet is admitted to the source node from the transport layer and ends when it reaches its destination.

III. Proposed Algorithm for the CRN

In this section, we design the throughput-optimal scheduling algorithm with the immediate reward mechanism. The algorithm is composed of two parts, namely, a congestion controller and a hop/link scheduler. The congestion controller generates and admits PU packets into the PU relay subnetwork, and a corresponding fraction of SU packets are admitted to their sources according to the immediate reward mechanism. The hop/link scheduler regulates the link transmission rates of the cooperative CRN. The formalized algorithm description

Step 1. Congestion Controller:
$k^* = \arg\min_k \left(\rho_k \sum_{l\in P_k} Q_l(t) + U_0^k(t)\right)$;
for $k = 1, ..., K$
  if $k = k^*$ **do:**
    if $\rho_{k^*} \sum_{l\in P_{k^*}} Q_l(t) + U_0^{k^*}(t) \le V_2$ **do:**
      $\mu_{-1,0}^k(t) = \mu_M$;
      for $m \in \{1, ..., H_k\}$
        do: admit $\rho_k\mu_M$ packets to SU v_k^m;
      end for
    else do: $\mu_{-1,0}^k(t) = 0$;
    end if
  else do: $\mu_{-1,0}^k(t) = 0$;
  end if
end for
Step 2. Hop/Link Scheduler:
  Find $\{(\mu_{mn}(t))_{(m,n)\in\mathcal{L}}, (s_l(t))_{l\in\mathcal{S}}\} \in \mathcal{I}$ that maximizes:
  $\sum_{k=1}^{K}\sum_{m=0}^{H_k} \mu_{m,m+1}^k(t)(U_m^k(t) - U_{m+1}^k(t))$
  $+ \sum_{l\in\mathcal{S}} Q_l(t)s_l(t)$;
Update $(U_m^k(t+1))$ and $(Q_l(t+1))$ according to (3)(4).

Fig. 2. Optimal algorithm with immediate reward mechanism in time slot t

is provided in Figure 2. Note that, according to the fixed-routing structure in PU relay subnetwork introduced in Section II-B, when developing the scheduler, we focus on the hop/link schedule $((\mu_{m,m+1}^k(t))_{m,k}, (s_l(t))_{l\in\mathcal{S}})$ which is composed of a PU hop schedule and an SU link schedule.

1) Congestion Controller:

$$\begin{aligned}&\min \sum_{k=1}^{K} \mu_{-1,0}^k(t)(\rho_k \sum_{l\in P_k} Q_l(t) + U_0^k(t) - V_2)\\ &\text{s.t. } \sum_{k=1}^{K} \mu_{-1,0}^k(t) \le \mu_M.\end{aligned} \tag{7}$$

The congestion controller (7) is a threshold-based optimization problem, with the control parameter V_2 as the threshold. The congestion controller (7) is developed to deterministically upper-bound the PU buffer size. Specifically, we will show later that V_2 determines the finite PU buffer size (in Proposition 1) and tradeoffs between the throughput optimality and delay performance (in Theorem 1). For time slot t, we define $k^* \triangleq \arg\min_k(\rho_k \sum_{l\in P_k} Q_l(t) + U_0^k(t))$. Then, to solve (7), we set

$$\mu_{-1,0}^{k^*}(t) = \begin{cases} \mu_M, & \text{if } \rho_{k^*} \sum_{l\in P_{k^*}} Q_l(t) + U_0^{k^*}(t) \le V_2, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

For $k \ne k^*$, we set $\mu_{-1,0}^k(t) = 0$. With a centralized control, $\rho_k\mu_{-1,0}^k(t)$ packets are admitted to SUs v_k^m, $m = 1, \cdots, H_k$.

2) Hop/Link Scheduler:

$$\begin{aligned}&\max\{\sum_{k=1}^{K}\sum_{m=0}^{H_k} \mu_{m,m+1}^k(t)(U_m^k(t) - U_{m+1}^k(t))\\ &\quad + \sum_{l\in\mathcal{S}} Q_l(t)s_l(t)\},\\ &\text{s.t. } \{(\mu_{mn}(t))_{(m,n)\in\mathcal{L}}, (s_l(t))_{l\in\mathcal{S}}\} \in \mathcal{I},\end{aligned} \tag{9}$$

where the optimization is taken over all feasible $((\mu_{m,m+1}^k(t))_{m,k}$,$(s_l(t))_{l\in\mathcal{S}})$ and we recall from (2) that each hop schedule $(\mu_{m,m+1}^k(t))_{m,k}$ corresponds to a PU link schedule $(\mu_{mn}(t))_{(m,n)\in\mathcal{L}}$. In the hop/link scheduler (9), each SU link rate is weighted by the SU queue backlog and each PU hop rate is weighted by a "hop back-pressure", i.e., the difference between the PU queue backlogs across a hop. The structure of the hop/link scheduler favors hops/links with higher weights for resource allocation, where we note that a higher weight implies a higher congestion level for a hop/link. When the hop back-pressure $U_m^k(t) - U_{m+1}^k(t) \le 0$, $m \in \{0, \cdots, H_k\}$, we set $\mu_{m,m+1}^k(t) = 0$, without loss of optimality.

The above proposed algorithm has the following finite buffer property:

Proposition 1: $\forall m \in \{0, \cdots, H_k + 1\}$, $\forall k \in \{1, \cdots, K\}$, the following inequality holds:

$$U_m^k(t) \le U_M \triangleq \mu_M + V_2, \tag{10}$$

where U_M can be regarded to as the finite PU buffer size for PU queue backlogs $U_m^k(t)$.

Proof: We prove Proposition 1 by induction on time slot. Initially when $t = 0$, $U_m^k(0) = 0\ \forall m, \forall k$. Now assume in time slot t we have $U_m^k(t) \le U_M$, $\forall m, \forall k$. In the induction step, for any given $k \in \{1, \cdots, K\}$ and any given $m \in \{0, \cdots, H_k\}$, we consider two cases:
Case 1: $m = 0$ for a route k. If $U_0^k(t) \le V_2$, then $U_0^k(t + 1) \le U_0^k(t) + \mu_M \le U_M$ according to queue dynamics (3), where we recall that $\mu_{-1,0}^k(t) \le \mu_M$ from the constraint in congestion controller (7). Otherwise, $V_2 < U_0^k(t) \le U_M$, and hence

$$\rho_k \sum_{l\in\mathcal{S}} Q_l(t)\mathbf{1}_{\{l\in P_k\}} + U_0^k(t) > V_2,$$

which yields $\mu_{-1,0}^k(t) = 0$ according to the congestion controller (7), and it follows that $U_0^k(t+1) \le U_0^k(t) \le U_M$ by the queue dynamics (3).
Case 2: $m \in \{1, \cdots, H_k\}$ for a route k. If $U_m^k(t) \le U_M - 1$, then we have $U_m^k(t+1) \le U_m^k(t) + 1 \le U_M$ according to queue dynamics (3). Otherwise, $U_m^k(t) = U_M \ge U_{m-1}^k(t)$, and according to the hop/link scheduler we have $\mu_{m-1,m}^k(t) = 0$, from which we obtain $U_m^k(t+1) \le U_m^k(t) = U_M$ by the queue dynamics (3).

Since the above analysis holds for any given m and k, $U_m^k(t+1) \le U_M\ \forall m \in \{0, \cdots, H_k\}$, $\forall k \in \{1, \cdots, K\}$, i.e., the induction step holds.

Completing Proposition 1, we recall that $U_{H_k+1}^k(t) = 0$, $\forall t$, holds for any route k. Thus, all PU packet queues are *deterministically* upper-bounded by U_M. ■

Now we present the main results of the above proposed algorithm in Theorem 1, with further explanations and discussions followed in Remark 1 and Remark 2.

Theorem 1: Let $\epsilon > 0$ be chosen arbitrarily small. The algorithm ensures the following inequality on SU queue backlogs:

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\sum_{l\in\mathcal{S}} Q_l(t)\} \le \frac{B_2 + V_2 B_R}{\delta_2}, \tag{11}$$

where $B_2 \triangleq \frac{1}{2}K(N+2) + \frac{1}{2}N + \frac{1}{2}N\mu_M^2 \max_k \rho_k^2$, $\delta_2 \triangleq$

$\epsilon \min_k \rho_k$, and B_R is defined as:

$$B_R \triangleq \limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\{\sum_{k=1}^{K} \mu_{-1,0}^k(t)\} - \sum_{k=1}^{K} \lambda_{k,\epsilon}^*.$$

Furthermore, the algorithm achieves:

$$\liminf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{k=1}^{K} \mathbb{E}\{\mu_{-1,0}^k(t)\} \geq \sum_{k=1}^{K} \lambda_{k,\epsilon}^* - \frac{B_2}{V_2}. \quad (12)$$

Proof: The proof is provided in Appendix A. ■

Remark 1 (Stability): The inequalities (10) from Proposition 1 and (11) from Theorem 1 indicate that PU and SU queues are all stable, and hence is the CRN. In addition, Proposition 1 ensures that PU queues maintained along each route are *deterministically* bounded by the finite buffer size U_M.

Remark 2 (Optimal Throughput and Tradeoff with Delay): The inequality (12) gives the lower-bound of the PU throughput the algorithm can achieve. Since the constant B_2 is independent of the control parameter V_2, the algorithm can achieve a PU throughput arbitrarily close to the optimal value $\sum_{k=1}^{K} \lambda_k^*$ as ϵ can be chosen arbitrarily small and V_2 arbitrarily large, with the following tradeoffs in PU and SU delay:

- The PU buffer size U_M is of order $O(V_2)$ as shown in (10). By Little's Theorem, the PU's average end-to-end delay over any given route k is of order $O((H_k+1)V_2)$ which is bounded by the first order of the number of hops (H_k+1), i.e., the algorithm achieves *order-optimal delay* per route [25], which is also confirmed by the numerical results provided in Section V-C. As a comparison, the traditional back-pressure algorithm [15] experiences a delay with an order characterized by the second-order term $(H_k+1)^2$ [24].
- From (11), the average SU buffer occupancy is of order $O(\frac{N+V_2}{\epsilon})$. And so is the average SU delay by Little's Theorem. Note that the average SU delay upper-bound has an extra term $\frac{1}{\epsilon}$ in order compared with the average PU delay.

The complexity of the proposed algorithm is dominated by that of the hop/link scheduler (9), essentially a centralized MWM problem [15], which maybe NP hard depending on the underlying interference model [20]. To reduce complexity of the hop/link scheduler, suboptimal algorithms can be developed to at least achieve a fraction γ of the optimal utility. These suboptimal algorithms include the well-studied Greedy Maximal Matching (GMM) [19] algorithm with $\gamma = \frac{1}{2}$ and the maximum weighted independent set (MWIS) problem such as GWMAX and GWMIN proposed in [23] with $\gamma = \frac{1}{\Delta}$, where Δ is the maximum degree of the CRN topology. The above suboptimal algorithms can be implemented with polynomial complexity in the number of nodes. Still, these algorithms are generally centralized and may have to enumerate all the feasible link schedules in the set $\mathcal{I}$. In Section IV, we propose an algorithm with a long-term reward mechanism that lends itself to a distributed implementation without losing optimality.

Before we conclude this section, we recall that the immediate rewards $\rho_k\mu_{-1,0}^k(t)$ for any route k are assumed to take integer values. This restriction can be eliminated by admitting $\lfloor \rho_k\mu_{-1,0}^k(t)\rfloor$ packets and adding the excess $\rho_k\mu_{-1,0}^k(t) - \lfloor \rho_k\mu_{-1,0}^k(t)\rfloor$ packets to future arrivals. Also note that the analysis can be extended to delayed rewards, i.e., a reward rate $\rho_k\mu_{-1,0}^k(t)$ is admitted to SU queues in time slot $(t+\tau')$, where τ' is the delay in time slots.

IV. Further Discussions

With the immediate reward mechanism, the optimal back-pressure-based algorithm proposed in Section III requires simultaneous admission of both PU and SU packets. This requirement of simultaneous admission can be relaxed when we propose an optimal algorithm with a long-term reward mechanism in Section IV-A. In addition, the long-term reward based algorithm can lead to a distributed solution under the cooperative CRN model, with the distributed implementation issues discussed in Section IV-B.

A. Proposed Algorithm with A Long-Term Reward Mechanism

In the original algorithm proposed in Section III with the immediate reward mechanism, SUs are assigned a channel share *proportional to* the relayed PU data, i.e., there may exist additional unutilized channel opportunities left by the PU. In addition, the congestion controller (7) is centralized to simultaneously admit both PU and SU packets. In this section, we extend our analysis to a CRN model with a more general ***long-term reward mechanism***. The objective of this extension is to allow SUs to better exploit the channel opportunities and to assist a fully distributed implementation. More specifically, the *time-average* rate of admitted SU packets is guaranteed to be *at least* ρ_k times the time-average rate of PU packets the SU relays. Let $A_l(t)$ denote the admitted SU arrival rates with SU uplink $(l, d_S) \in \mathcal{L}'$ in time slot t and upper-bound $A_l(t) \leq A_M$, $\forall l \in \mathcal{S}$. The long-term reward mechanism guarantees

$$\liminf_{T\to\infty}\Big(\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{A_l(t)\} - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{\sum_{k=1}^{K}\mu_{-1,0}^k(t)\mathbf{1}_{\{l\in P_k\}}\rho_k\}\Big) \geq 0, \forall l \in \mathcal{S}, \quad (13)$$

where the routing scenario and PU data admission follow the analysis in Section II-B and reward parameter ρ_k for k-th route takes *fractional values*. The aim of the following modified algorithm is to achieve a PU throughput arbitrarily close to the optimal value $\sum_{k=1}^{K} \lambda_k^*$ with the long-term reward guarantees (13), where we recall that (λ_k^*) is a solution to the optimization (6). With an SU admitted arrival rate, the SU packet queue backlogs will be updated as follows:

$$Q_l(t+1) = [Q_l(t) - s_l(t)]^+ + A_l(t), \forall l \in \mathcal{S}. \quad (14)$$

To guarantee the long-term rewards (13), we construct a virtual queue $D_l(t)$ at each SU $l \in \mathcal{S}$, with virtual queue dynamics:

$$D_l(t+1) = [D_l(t) - A_l(t)]^+ + \sum_{k=1}^{K}\rho_k\mu_{-1,0}^k(t)\mathbf{1}_{\{l\in P_k\}}. \quad (15)$$

From the above queue evolution, the stability of the virtual queues $D_l(t)$ implies the reward guarantees. Employing the

Step 1. PU Congestion Controller:
$k^* = \arg\min_k \left(\rho_k \sum_{l\in P_k} D_l(t) + U_0^k(t)\right)$;
for $k = 1, ..., K$
  if $k = k^*$ **do:**
    if $\rho_{k^*} \sum_{l\in P_{k^*}} D_l(t) + U_0^{k^*}(t) \le V_2$
      do: $\mu_{-1,0}^k(t) = \mu_M$;
    else do: $\mu_{-1,0}^k(t) = 0$;
    end if
  else do: $\mu_{-1,0}^k(t) = 0$;
  end if
end for
Step 2. SU Congestion Controller:
for SU $l \in \mathcal{S}$ **do:**
  if $Q_l(t) \le D_l(t)$
    do: $A_l(t) = A_M$;
  else do: $A_l(t) = 0$;
  end if
end for
Step 3. Hop/Link Scheduler:
  Find $\{(\mu_{mn}(t))_{(m,n)\in\mathcal{L}}, (s_l(t))_{l\in\mathcal{S}}\} \in \mathcal{I}$ that maximizes:
  $\sum_{k=1}^K \sum_{m=0}^{H_k} \mu_{m,m+1}^k(t)(U_m^k(t) - U_{m+1}^k(t))$
  $+ \sum_{l\in\mathcal{S}} Q_l(t)s_l(t)$;
Update $(U_m^k(t+1))$ and $(Q_l(t+1))$ according to (3)(14).

Fig. 3. Optimal algorithm with long-term reward mechanism in time slot t

virtual queues $D_l(t)$, we develop the following modified algorithm composed of a PU congestion controller, an SU congestion controller, and a hop/link scheduler. The formalized algorithm description is provided in Figure 3.

1) PU Congestion Controller: Redefining $k^* \triangleq \arg\min_k(\rho_k \sum_{l\in P_k} D_l(t) + U_0^k(t))$, we admit the PU packets on the k^*-th route as follows

$$\mu_{-1,0}^{k^*}(t) = \begin{cases} \mu_M, \text{ if } \rho_{k^*} \sum_{l\in P_{k^*}} D_l(t) + U_0^{k^*}(t) \le V_2, \\ 0, \quad \text{otherwise,} \end{cases} \tag{16}$$

where V_2 is the same control parameter as in Section III. For route $k \ne k^*$, we set $\mu_{-1,0}^k(t) = 0$. Compared to the original congestion controller (7), we utilize the virtual queue $D_l(t)$ instead of the actual SU queue backlog $Q_l(t)$.

2) SU Congestion Controller: For each SU $l \in \mathcal{S}$,

$$A_l(t) = \begin{cases} A_M, \text{ if } Q_l(t) \le D_l(t), \\ 0, \quad \text{otherwise.} \end{cases} \tag{17}$$

The threshold-based SU congestion controller has a time-varying threshold, i.e, the virtual queue $D_l(t)$.

3) Hop/Link Scheduler: The hop/link scheduler remains the same as (9).

It is not difficult to check that Proposition 1 holds (i.e., the PU packet queues are deterministically upper-bounded) and we present the main results of the modified algorithm in Theorem 2, followed by further explanations in Remark 3.

Theorem 2: For some arbitrarily small $\epsilon > 0$, the SU packet queues and virtual queues are stable:

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{l\in\mathcal{S}} \mathbb{E}\{Q_l(t) + D_l(t)\} \le \frac{2B_5 + 2V_2 B_R}{\delta_2},$$

where $B_5 = B_2 + NA_M^2$ and B_2, B_R, δ_2 are defined in Theorem 1. In addition, the PU throughput is lower-bounded:

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{k=0}^{K} \mathbb{E}\{\mu_{-1,0}^k(t)\} \ge \sum_{k=1}^{K} \lambda_{k,\epsilon}^* - \frac{B_5}{V_2},$$

where we recall that $(\lambda_{k,\epsilon}^*)$ is defined in (5) whose sum is ϵ-close to the optimal PU throughput $\sum_{k=1}^K \lambda_k^*$.

Proof: The proof sketch for Theorem 2 is provided in Appendix B. ■

Remark 3 (Algorithm Performance): Finite buffer property still holds for PU packet queue backlogs since Proposition 1 holds. Remark 1-2 also hold which guarantee network stability and optimality with a delay tradeoff. In addition, since the virtual queues $D_l(t)$ are stable, the long-term reward (13) for SUs is satisfied, i.e., the SU channel exploitation is *at least* ρ_k times the relayed PU data on k-th route, which guarantees that the long-term SU throughput is no less than the SU throughput achieved by the algorithm for the immediate reward mechanism proposed in Section III.

B. Distributed Implementation Issues

In this section, we discuss the distributed implementation issues concerning the algorithm proposed in Section IV-A. Specifically, we discuss approaches to the distributed implementation of the PU/SU congestion controller and the hop/link scheduler, respectively.

When relaying PU data, we append the information of $(\mu_{-1,0}^k(t))$ to PU data packets for SU relay, and a delayed information of $(\mu_{-1,0}^k(t-\tau))$ will be available at SUs, where τ is an integer larger than the maximum propagation delay between any SU and the source PU. To implement the SU congestion controller (17) in a distributed manner at each SU node $l \in \mathcal{S}$, we can replace $(D_l(t))$ in (17) by $(D_l(t-\tau+1))$ which is updated with $(\mu_{-1,0}^k(t-\tau))$ according to the virtual queue dynamics (15). Similarly, when relaying PU data, the information of admitted SU arrival rates $(A_l(t))$ can be piggybacked on each ACK SUs transmit, and a delayed information of $(A_l(t-\tau))$ will be available at the PU source node s_P. To implement the PU congestion controller (16) in a distributed manner at the PU source node s_P, we can replace $(D_l(t))$ in (16) by $(D_l(t-\tau+1))$ which is updated with $(A_l(t-\tau))$. With the employment of delayed queue backlog information, the PU throughput optimality should be maintained with a slower convergence of the system, which can be proved with similar proof techniques in [21][22].

The distributed implementation of hop/link scheduler can be developed similar to [18] to achieve *a fraction* of the optimal throughput. A throughput arbitrarily close to the optimum can be achieved in a distributed manner through extensions of the scheduler by employing random access techniques [27][28]. To be more specific, we can replace the scheduler (9) in Section III by the distributed scheduler in [27][28] with fugacities [29][2] chosen as $\alpha[U_m^k(t) - U_{m+1}^k(t)]^+$ for hop $(v_k^m, v_k^{m+1}) \in P_k$ [3] and $\alpha Q_l(t)$ for an SU link $l \in \mathcal{S}$, where

[2]Fugacities [29] are employed to determine the transmission probabilities of a communication link in a CSMA framework.

[3]If the current PU queue backlog information is not available at an SU, we can instead use the most recent available queue backlog information without loss of optimality.

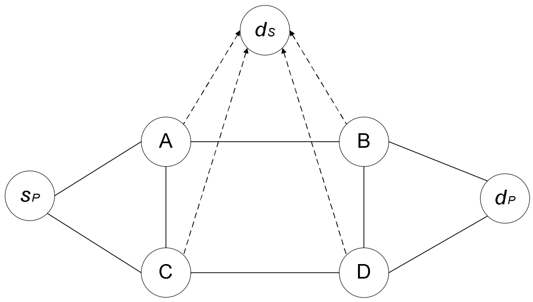


Fig. 4. Cooperative CRN topology for simulation

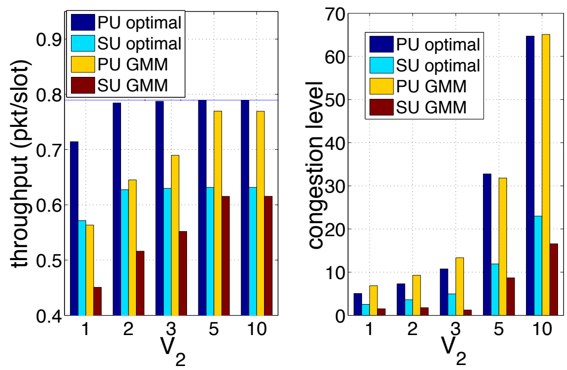


Fig. 5. Algorithm performance with varying V_2, $\rho_1 = \rho_2 = 0.4$

α is a positive weight. It can be shown that the distributed hop/link scheduler can achieve a performance arbitrarily close to that of the centralized scheduler (9) with high probability when the PU queue backlog differences $[U_m^k(t) - U_{m+1}^k(t)]^+$ and SU queue backlogs $Q_l(t)$ are large enough (see Theorem 1 in [27] for a similar proof under a time-scale separation assumption [26]), which leads to the throughput optimality of the modified algorithm.

Although the proposed algorithm can be implemented without loss of throughput optimality in a distributed fashion by directly employing the above introduced random access techniques [27][28], as shown in the simulation results in [27], the distributed version of the algorithm can suffer from a much larger delay compared to the centralized counterpart.

V. Numerical Results

In this section, we present a simulation-based performance evaluation for the algorithm proposed in Section III. Simulation results are obtained using the topology shown in Figure 4, which consists of a PU source (s_P) and a PU destination (d_P). The maximum admitted arrival rate for the PU is $\mu_M = 2$. The primary traffic is relayed by SUs A, B, C and D with their own one-hop secondary traffic destined to the secondary destination (d_S). We employ the node-exclusive model as the underlying interference model for the cooperative CRN and consider two predetermined routes $P_1 = (s_P, A, B, d_P)$ and $P_2 = (s_P, C, D, d_P)$ for PU data relay. Both PU and SU traffics are assumed to be constantly backlogged at the sources. The results reflect averages obtained over 50000 time slots for each run.

A. Effects of V_2 on the Algorithm Performance

By fixing the route-specific reward parameters $\rho_1 = \rho_2 = 0.4$, we present the algorithm performance in Figure 5 by varying the control parameter V_2 in the congestion controller which determines the buffer size according to (10). The throughput is measured in packets per time slot with SU throughput represented by the *sum throughput* of all SU flows. The congestion level is illustrated as the average number of sum of PU/SU packets in the CRN in a time slot. The optimal algorithm is compared with a GMM algorithm which employs the same congestion controller (7) but solves the hop/link scheduler (9) in a suboptimal approach of greedy maximal matching.

In Figure 5, which conforms with the statement in Remark 2, the control parameter V_2 trades off between the throughput and the PU/SU congestion level. Specifically, under the proposed optimal algorithm, the PU throughput converges to its optimal value 0.7895 packet/slot when V_2 increases, and $V_2 = 10$ is sufficiently large such that there is hardly any gain in the PU/SU throughput by further increasing V_2 while the PU and SU congestion levels increase almost linearly with V_2. The sum SU throughput corresponding to the optimal PU throughput is 0.6316 packet/slot. We note that the PU congestion level is higher than that of SUs since the PU traffic is multi-hop and the PU throughput is much larger than that of SU when $\rho_1 = \rho_2 = 0.4$. We also note that the PU traffic is split even in the two fixed routes, P_1 and P_2, due to the symmetry of the topology.

As illustrated in Figure 5, given a same V_2, the suboptimal GMM algorithm experiences similar PU congestion levels as the optimal algorithm. However, the PU/SU throughput under GMM is much worse than that under the optimal algorithm. Taking the case $V_2 = 3$ for instance, compared to the optimal algorithm, the PU congestion level is 23.98% higher while the PU throughput is 12.41% less under GMM. Thus, given a same V_2, suboptimal algorithms such as GMM achieve worse throughput performance than the optimal algorithm, while the PU delay performance is not improved. With a same V_2 value, we observe that the SU congestion level under GMM is lower than that of the optimal algorithm due to a lower SU throughput.

We note that for comparable throughput levels, the optimal algorithm far outperforms the suboptimal GMM algorithm in terms of congestion level. For instance, consider the performance of the optimal algorithm with $V_2 = 2$ and the GMM algorithm with $V_2 = 10$ in Figure 5. The optimal algorithm achieves much lower PU/SU congestion levels than the GMM algorithm, while still achieving a slightly larger PU/SU throughput. However, the superior performance of the optimal algorithm comes at the price of high complexity: As discussed in Section III, the complexity of the optimal algorithm can grow exponentially in the number of nodes whereas the GMM algorithm can be implemented with polynomial complexity in the number of nodes.

B. Algorithm Performance with Different Values of ρ_k

In Figure 6, by fixing $V_2 = 10$, we illustrate the throughput and congestion level performance of the algorithm against the

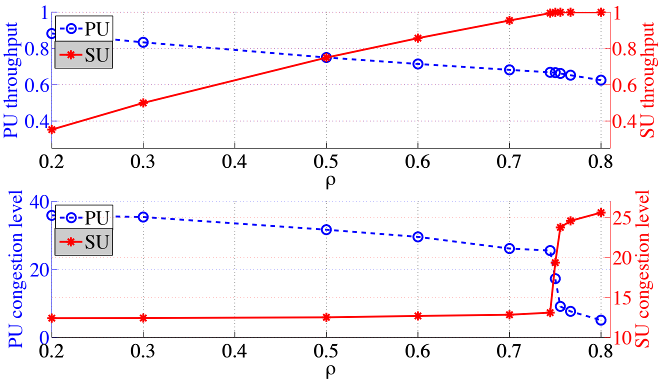


Fig. 6. Algorithm performance with varying ρ, $V_2 = 10$

route-specific reward parameters $\rho_1 = \rho_2 = \rho$, where we recall that the number of admitted secondary packets for each SU is ρ times the admitted PU packets and note that SU throughput is the sum for all SUs. According to the topology and the immediate reward mechanism, we must have the following relation between PU and SU throughput:

$$\text{SU throughput} = 2\rho \times (\text{PU throughput}), \quad (18)$$

noting that there are 2 SUs along each pre-determined route.

We observe initially that the PU throughput decreases and SU throughput increases linearly when ρ increases, satisfying (18). When ρ further increases to around 0.75, SU throughput reaches and stays at its allowed maximum (1 packet/slot with the node-exclusive interference model), hence the LHS of (18) becomes constant, which leads to a faster decrease in PU throughput linearly with respect to ρ to guarantee (18).

With an increasing ρ, the allocated share of SU increases, which results in a corresponding small linear increase in SU congestion levels. When ρ further increases to a level that allows SUs to reach their capacity (which is 1 packet/slot), the SU congestion level increases significantly, which is necessary for the SUs to approach their allowed maximum throughput.

An increasing ρ reduces the throughput of the PU as more capacity is allocated to SUs. A decreased PU throughput level requires smaller queue sizes, which is reflected in the initial linear decrease observed in the PU congestion level. When ρ further increases to around 0.75 (where SU throughput reaches its allowed maximum), PU congestion level drops significantly. This observed drop in PU congestion level can be interpreted as follows. SU traffic becomes congested when SU throughput reaches the capacity, which results in a large increase in the term $\rho_k \sum_{l \in P_k} Q_l(t)$, $k = 1, 2$, in the congestion controller (7). With a fixed threshold, i.e., a fixed control parameter V_2, the threshold-based congestion controller (7) keeps the PU queue backlogs $(U_0^k(t))_{k=1,2}$ at the source PU at a low level. The succeeding PU queue backlogs along the routes will be shaped accordingly by the hop-back-pressure-based hop/link scheduler (9), which leads to a significant decrease in the PU congestion level.

C. Initial Results on Order-Optimal Delay of PU

To verify the order-optimal delay property of the algorithm stated in Remark 2, we consider the node-exclusive interference model with a different CRN topology shown in Figure 7 where there is only one pre-determined route with N SUs.

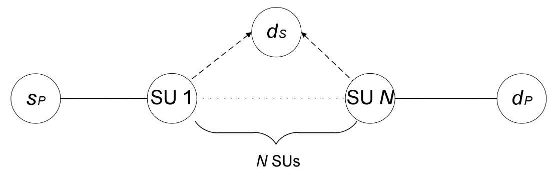


Fig. 7. Cooperative CRN topology for with varying number of SUs

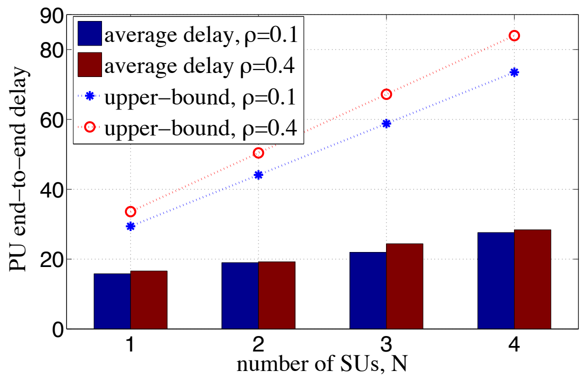


Fig. 8. PU delay performance for topology in Figure 7

With fixed $\mu_M = 2$ and $V_2 = 5$, Figure 8 illustrates that the PU average end-to-end delay, measured in time slots, grows almost linearly with the number of SUs N (note that the number of hops for PU data relay is $N + 1$). In Figure 8, the linear delay upper-bounds are derived from the finite buffer property (10) and the Little's Theorem, where we note that, for $N = 1, 2, 3, 4$, the PU throughput is 0.4762 packet/slot for $\rho = 0.1$ and 0.4167 packet/slot for $\rho = 0.4$. Thus, Figure 8 confirms that the PU traffic experiences an order-optimal delay, i.e., the average end-to-end delay is of first order of the number of hops.

VI. Conclusions and Future Works

In this paper, we introduced a cross-layer scheduling algorithm for multi-hop cooperative cognitive radio networks. The algorithm can achieve a PU throughput arbitrarily close to the optimum, with a tradeoff in the deterministically upper-bounded PU buffer sizes. The algorithm is then scrutinized with respect to its feasibility for distributed implementation. We have proposed extensions to facilitate distributed implementations. The theoretically proved properties of the algorithms are then illustrated via a simulation study. In our future work, we will investigate methods of relaxing the fixed route assumption and the interference graph model. We will also develop a proof-of-concept implementation of the proposed distributed algorithm with the long-term reward mechanism introduced in Section IV.

Appendix A
Proof of Theorem 1

Before we proceed, we present Lemma 1 as follows to assist us in proving Theorem 1.

Lemma 1: For any feasible rate vector $((\lambda_k)_{k \in \{1,\cdots,K\}}, (\sum_{k=1}^{K} \rho_k \lambda_k \mathbf{1}_{\{l \in P_k\}})_{l \in \mathcal{S}}) \in \Lambda_E$, there exists a stationary randomized algorithm *SE* that stabilizes the network with PU admitted arrival rates $\mu^{k,SE}_{-1,0}(t)$ such that $\mathbb{E}\{\mu^{k,SE}_{-1,0}(t)\} = \lambda_k$, $\forall t\ \forall k \in \{1, \cdots, K\}$, and a hop/link schedule $((\mu^{k,SE}_{m,m+1}(t))_{m,k}, (s^{SE}_l(t))_{l\in\mathcal{S}})$ independent of queue backlogs satisfying:

$$\mathbb{E}\{\mu^{k,SE}_{m-1,m}(t) - \mu^{k,SE}_{m,m+1}(t)\} = 0,\ \forall t,\ \forall m, k; \tag{19}$$

$$\mathbb{E}\{s^{SE}_l(t)\} = \sum_{k=1}^{K} \rho_k \lambda_k \mathbf{1}_{\{l\in P_k\}},\ \forall t,\ \forall l \in \mathcal{S}. \tag{20}$$

Similar formulations of stationary randomized algorithms and existence proofs have been presented in [12][15]-[17], so we omit the proof of Lemma 1 for brevity. Note that $(\lambda_k)_{k\in\{1,\cdots,K\}}$ can take values as $(\lambda^*_{k,\epsilon})_k$ and $(\lambda^*_{k,\epsilon} + \epsilon)_k$.

We denote the queue vector $\mathbf{Q}_E(t) = ((U^k_m(t))_{m,k}, (Q_l(t))_{l\in\mathcal{S}})$ and define the Lyapunov function $L_E(\mathbf{Q}_E(t))$ as

$$L_E(\mathbf{Q}_E(t)) \triangleq \frac{1}{2}\{\sum_{k=1}^{K}\sum_{m=0}^{H_k}(U^k_m(t))^2 + \sum_{l\in\mathcal{S}} Q_l(t)^2\},$$

with the corresponding Lyapunov drift defined as $\Delta_E(t) \triangleq \mathbb{E}\{L_E(\mathbf{Q}_E(t+1)) - L_E(\mathbf{Q}_E(t))|\mathbf{Q}_E(t)\}$.

By squaring both sides of the queue dynamics (3)(4) and through algebra, we have:

$$\begin{aligned}
&\Delta_E(t) - V_2\mathbb{E}\{\sum_{k=1}^{K}\mu^k_{-1,0}(t)|\mathbf{Q}_E(t)\} \\
\le& B_2 - V_2\mathbb{E}\{\sum_{k=1}^{K}\mu^k_{-1,0}(t)|\mathbf{Q}_E(t)\} \\
&- \sum_{k=1}^{K}\sum_{m=0}^{H_k}\mathbb{E}\{U^k_m(t)(\mu^k_{m,m+1}(t) - \mu^k_{m-1,m}(t))|\mathbf{Q}_E(t)\} \\
&- \sum_{l\in\mathcal{L}}\mathbb{E}\{Q_l(t)(s_l(t) - \sum_{k=1}^{K}\rho_k\mu^k_{-1,0}(t)\mathbf{1}_{\{l\in P_k\}})|\mathbf{Q}_E(t)\}.
\end{aligned} \tag{21}$$

Employing the following fact that, $\forall k \in \{1, \cdots, K\}$,

$$\begin{aligned}
&\sum_{m=0}^{H_k} U^k_m(t)(\mu^k_{m,m+1}(t) - \mu^k_{m-1,m}(t)) \\
=& \sum_{m=0}^{H_k}\mu^k_{m,m+1}(t)(U^k_m(t) - U^k_{m+1}(t)) - \mu^k_{-1,0}(t)U^k_0(t),
\end{aligned}$$

we find the equivalence of (21):

$$\begin{aligned}
&\Delta_E(t) - V_2\mathbb{E}\{\sum_{k=1}^{K}\mu^k_{-1,0}(t)|\mathbf{Q}_E(t)\} \\
\le& \sum_{k=1}^{K}\mathbb{E}\{\mu^k_{-1,0}(t) \times (\rho_k\sum_{l\in P_k}Q_l(t) + U^k_0(t) - V_2)|\mathbf{Q}_E(t)\} \\
&+ B_2 - \mathbb{E}\{\sum_{k=1}^{K}\sum_{m=0}^{H_k}\mu^k_{m,m+1}(t)(U^k_m(t) - U^k_{m+1}(t)) \\
&\qquad + \sum_{l\in\mathcal{S}}Q_l(t)s_l(t)|\mathbf{Q}_E(t)\}.
\end{aligned} \tag{22}$$

Note that the first and third terms of the RHS of (22) are minimized by the congestion controller (7) and the hop/link scheduler (9), respectively, over a set of feasible algorithms including the stationary randomized algorithm SE introduced in Lemma 1. Then, we substitute into the first term of the RHS of (22) a stationary randomized SE with admitted PU arrival rate vector $(\lambda^*_{k,\epsilon})_{k\in\{1,\cdots,K\}}$ and into the third term the SE with admitted PU arrival rate vector $(\lambda^*_{k,\epsilon} + \epsilon)_{k\in\{1,\cdots,K\}}$. After the above substitutions, we obtain:

$$\begin{aligned}
&\Delta_E(t) - V_2\mathbb{E}\{\sum_{k=1}^{K}\mu^k_{-1,0}(t)|\mathbf{Q}_E(t)\} \\
\le& B_2 - V_2\sum_{k=1}^{K}\lambda^*_{k,\epsilon} - \epsilon\sum_{l\in\mathcal{S}}Q_l(t)\sum_{k=1}^{K}\rho_k\mathbf{1}_{\{l\in P_k\}} \\
\le& B_2 - V_2\sum_{k=1}^{K}\lambda^*_{k,\epsilon} - \delta_2\sum_{l\in\mathcal{S}}Q_l(t).
\end{aligned} \tag{23}$$

We take the expectation of both sides of (23) over $\mathbf{Q}_E(t)$ and take the time average on $t = 0, 1, \cdots, T-1$, which yields

$$\begin{aligned}
&\frac{\delta_2}{T}\sum_{t=0}^{T-1}\mathbb{E}\{\sum_{l\in\mathcal{S}}Q_l(t)\} \\
\le& B_2 + \frac{V_2}{T}\sum_{t=0}^{T-1}\mathbb{E}\{\sum_{k=1}^{K}\mu^k_{-1,0}(t)\} - V_2\sum_{k=1}^{K}\lambda^*_{k,\epsilon}.
\end{aligned} \tag{24}$$

By taking limsup of T on both sides of (24), we can prove (11). By taking the liminf of T on both sides of (24), we can prove (12). Therefore, Theorem 1 is proved.

Appendix B
Proof Sketch for Theorem 2

Similar to the proof for Theorem 1 in Appendix A, we redefine the Lyapunov function $L_E(\mathbf{Q}_E(t))$ as follows:

$$L_E(\mathbf{Q}_E(t)) \triangleq \frac{1}{2}\{\sum_{k=1}^{K}\sum_{m=0}^{H_k}(U^k_m(t))^2 + \sum_{l\in\mathcal{S}}(Q_l(t)^2 + D_l(t)^2)\}$$

with $\mathbf{Q}_E(t) = ((U^k_m(t))_{m,k}, (Q_l(t))_{l\in\mathcal{S}}, (D_l(t))_{l\in\mathcal{S}})$. Then, similar to the analysis in deriving (22) in Appendix A, we arrive at the following inequality on the corresponding Lyapunov drift $\Delta_E(t)$:

$$\begin{aligned}
&\Delta_E(t) - V_2\mathbb{E}\{\sum_{k=1}^{K}\mu^k_{-1,0}(t)|\mathbf{Q}_E(t)\} \\
\le& \sum_{k=1}^{K}\mathbb{E}\{\mu^k_{-1,0}(t) \times (\rho_k\sum_{l\in P_k}D_l(t) + U^k_0(t) - V_2)|\mathbf{Q}_E(t)\} \\
&+ B_5 + \sum_{l\in\mathcal{S}}\mathbb{E}\{A_l(t)(Q_l(t) - D_l(t))|\mathbf{Q}_E(t)\} \\
&- \mathbb{E}\{\sum_{k=1}^{K}\sum_{m=0}^{H_k}\mu^k_{m,m+1}(t)(U^k_m(t) - U^k_{m+1}(t)) \\
&\qquad + \sum_{l\in\mathcal{S}}Q_l(t)s_l(t)|\mathbf{Q}_E(t)\},
\end{aligned} \tag{25}$$

Note that the first, third, and fourth terms of the RHS of (25) are minimized by the PU congestion controller (16), the SU congestion controller (17) and the hop/link scheduler (9), respectively, over a set of feasible algorithms including the stationary randomized algorithm SE introduced in

Lemma 1 with the additional properties: $\mathbb{E}\{A_l^{STAT}(t)\} = \sum_{k=1}^{K} \rho_k \lambda_k \mathbf{1}_{\{l \in P_k\}}$, $\forall l \in \mathcal{L}$ $\forall t$. Then, we substitute into the first term of the RHS of (25) a stationary randomized SE with admitted PU arrival rate vector $(\lambda_{k,\epsilon}^*)_{k\in\{1,\cdots,K\}}$, into the third term the SE with $(\lambda_{k,\epsilon}^* + \frac{1}{2}\epsilon)_{k\in\{1,\cdots,K\}}$ and last term the SE with $(\lambda_{k,\epsilon}^* + \epsilon)_{k\in\{1,\cdots,K\}}$. Following the proof in Appendix A, we can prove Theorem 2.

References

[1] "FCC Spectrum Policy Task Force: Report of the spectrum efficiency working group", November 2002.

[2] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey, Computer Networks", in *Computer Networks J. (Elsevier)*, vol. 50, no. 13, pp. 2127-2159, 2006.

[3] L. Lai, H. Gamal, H. Jiang and H. Poor, "Cognitive Medium Access: Exploration, Exploitation and Competition", in *IEEE Trans. Mobile Computing*, vol. 10, no. 2, pp. 239-253, Feb. 2010.

[4] Q. Zhao, L. Tong, A. Swami and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: a POMDP framework", in *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 589-600, April 2007.

[5] H. Su and X. Zhang, "Cross-layer based opportunistic MAC protocols for QoS provisioning over cognitive radio wireless networks", in *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 118-129, January 2008.

[6] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks", in *IEEE Trans. Mobile Computing*, vol. 7, no. 5,pp. 533-545, May 2008.

[7] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks", in *Computer Networks*, vol. 41, no. 4, pp. 451-474, Mar. 2003.

[8] S. Bayat, R. Louie, Y. Li, and B. Vucetic, "Cognitive radio relay networks with multiple primary and secondary users: distributed stable matching algorithms for spectrum access", in *Proc. IEEE International Conference on Communications (ICC) 2011*, pp. 1-6, June 2011.

[9] L. Giupponi and C. Ibars, "Distributed cooperation among cognitive radios with complete and incomplete information", in *EURASIP J. Advances in Signal Processing*, vol. 2009, No. 3, pp. 1-13, March 2009.

[10] Y. Han, A. Pandharipande, and S. H. Ting, "Cooperative decode-andforward relaying for secondary spectrum access", in *IEEE Trans. Wireless Commun.*, vol. 8, no. 10, pp. 4945-4950, Oct. 2009.

[11] R. Urgaonkar and M. Neely, "Opporunistic scheduling with reliability guarantees in cognitive radio networks", in *IEEE Trans. Mobile Computing*, vol. 8, no. 6, pp. 766-777, June 2009.

[12] M. Lotfinezhad, B. Liang, and E. Sousa, "Optimal control of constrained cognitive radio networks with dynamic population size", in *Proc. IEEE INFOCOM'10*, pp. 1-9, March 2010.

[13] K. Khalil, M. Karaca, O. Ercetin, and E. Ekici, "Optimal scheduling in cooperate-to-join cognitive radio networks", *in Proc. IEEE INFOCOM'11*, pp. 3002-3010, April 2011.

[14] D. Xue and E. Ekici, "Guaranteed opportunistic scheduling in multi-hop cognitive radio networks", *in IEEE INFOCOM'11*, pp. 2984-2992, April 2011.

[15] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[16] L. Georgiadis, M. Neely, and L.Tassiulas, "Resource allocation and cross-Layer control in wireless networks", in *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, April 2006.

[17] M. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varing channels", Ph.D. dissertation, Mass. Inst. Technol. (MIT), Cambridge, MA, 2003.

[18] X. Wu, R. Srikant, and J. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks", in *IEEE Trans. Mobile Computing*, vol. 6, no. 6, pp. 595-605, June 2007.

[19] C. Joo, X. Lin, and N. Shroff, "Greedy maximal matching: performance limits for arbitrary network graphs under the node-exclusive interference model", in *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2734-2744, December 2009.

[20] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless neworks", in *Proc. 12th Annual International Conference on Mobile Computing and Networking (MobiCom'06)*, pp. 227-238, 2006.

[21] L. Le, E. Modiano, and N. Shroff, "Optimal control of wireless networks with finite buffers", in *Proc. IEEE INFOCOM'10*, pp. 1-9, March 2010.

[22] L. Ying, R. Srikant and D. Towsley, "Cluster-based back-pressure routing algorithm", in *Proc. IEEE INFOCOM08*, pp. 484-492, April 2008.

[23] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem", in *Discrete Applied Mathematics*, vol. 126, No. 2-3, pp. 313-322, March 2003.

[24] L. Bui, R. Srikant and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routin", in *Proc. IEEE INFOCOM09 Mini-Conference*, pp. 2936-2940, April 2009.

[25] P. Huang, X. Lin, and C. Wang, "A low-complexity congestion control and scheduling algorithm for multi-hop wireless networks with order-optimal per-flow delay", in *Proc. IEEE INFOCOM'11*, pp. 2588-2596, April 2011.

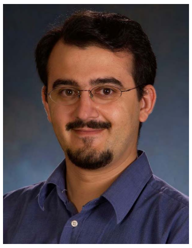
[26] L. Jiang and J. Walrand, "A distributed algorithm for throughput and utility maximization in wireless networks", in *IEEE/ACM Trans. Netw.*, vol. 18, no.3, pp. 960-972, June 2010.

[27] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks", in *Proc. IEEE INFOCOM'10 Mini-Conference*, 271-275, March 2010.

[28] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution", in *Proc. ACM SIGMETRICS'09*, pp. 133-144, June 2009.

[29] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling", in *IEEE INFOCOM'11 Mini-Conference*, pp. 371-375, April 2011.

Dongyue Xue (S'11) received the B.S. degree in Information Engineering from Shanghai Jiaotong University, Shanghai, China, in 2009, and is currently pursuing the Ph.D. degree in electrical and computer engineering at the Ohio State University, Columbus, OH. His research interests include cross-layer scheduling in wireless networks and dynamic resource allocation in cognitive radio networks.



Eylem Ekici (S'99-M'02-SM'11) received his BS and MS degrees in Computer Engineering from Bogazici University, Istanbul, Turkey, in 1997 and 1998, respectively, and the PhD degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA, in 2002. Currently, he is an Associate Professor with the Department of Electrical and Computer Engineering, The Ohio State University. His current research interests include wireless networks, vehicular communication systems, cognitive radio networks, nano communication systems, with an emphasis on resource management, and analysis of network architectures and protocols. He is an Associate Editor of Computer Networks Journal (Elsevier) and ACM Mobile Computing and Communications Review, and IEEE/ACM Transactions on Networking.