Achieving Transparent Coexistence in a Multi-hop Secondary Network Through Distributed Computation

Xu Yuan† Yi Shi† Y. Thomas Hou† Wenjing Lou† Scott F. Midkiff† Sastry Kompella‡

† Virginia Polytechnic Institute and State University, USA

‡ U.S. Naval Research Laboratory, Washington, DC, USA

***Abstract*—Transparent coexistence, also known as underlay, offers much more efficient spectrum sharing than traditional interweave coexistence paradigm. In a previous work, the transparent coexistence for a multi-hop secondary networks is studied. In this paper, we design a distributed solution to achieve this paradigm. In our design, we show how to increase the number of data streams iteratively while meeting constraints in the MIMO interference cancelation (IC) model and achieving transparent coexistence. All steps in our distributed algorithm can be accomplished based on local information exchange among the neighboring nodes. Our simulation results show that the performance of our distributed algorithm is highly competitive when compared to an upper bound solution for the centralized problem.**

I. Introduction

Coexistence of the secondary network with a primary network is the key approach to improve radio spectrum utilization. In [7], a novel coexistence paradigm under the name of *transparent coexistence* is explored for a secondary multi-hop network. Under this paradigm, there is no change on the primary network. A primary network considers itself as the only wireless network that is using the spectrum and its spectrum access behavior is not affected by the secondary network. In this regard, the primary network's behavior is similar to that in the interweave paradigm [2]. The difference is in the behavior for the secondary network. Instead of accessing the spectrum only through spectrum holes in time, frequency, or space, a secondary network under the transparent coexistence paradigm is allowed to access the spectrum in the *same* time, frequency, and location with the primary network, as long as its activities are "invisible" to the primary network. Such transparency is achieved by having the secondary network proactively cancel its interference to the primary network.

Prior to [7], there has been some efforts on underlay coexistence paradigm (see, e.g., [1], [4], [8], [9]). But these efforts have been limited to very simple network settings, e.g., several nodes or link pairs, all for single-hop communications. The transparent coexistence that was studied in [7] focused on multi-hop communications, both for the primary and secondary networks. The results in [7] showed the concept of achieving transparent coexistence for multi-hop primary and secondary networks through a centralized solution. But it is desirable to have a distributed solution to solve this problem. This is the goal of this paper. The main contribution of this paper is the development of a distributed scheduling algorithm to allocate secondary network resource to achieve the transparent coexistence paradigm. For IC, we assume each secondary network is equipped with MIMO, while there is no requirement on the primary nodes. We employ a MIMO interference cancelation (IC) model that was developed in *et al.* [5] to keep track of DoF allocation for IC. It was shown in [5] that this IC model is efficient in DoF allocation while guaranteeing feasibility in the final solution. By feasibility, we mean there exists a feasible precoding and decoding vector for each data stream at the physical layer. However, this model requires to maintain an ordering among the secondary nodes in the network, which poses a challenge in a distributed network environment. We show how such ordering relationship among the neighboring nodes can be established and maintained at each node in a distributed environment. We also show how to adjust node ordering in our distributed algorithm should it become necessary (e.g., remove bottleneck in DoF resource allocation) . Through numerical results, we show that the iterative distributed algorithm that we propose offers competitive performance when compared with an upper bound result.

For correspondence, please contact Prof. Tom Hou (thou@vt.edu).

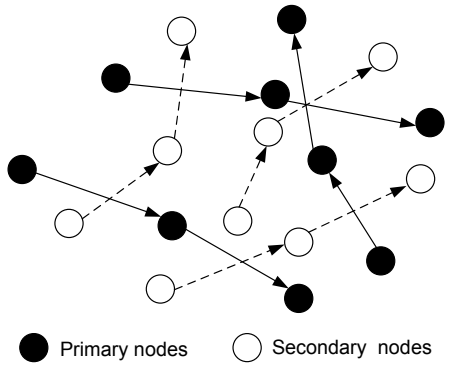

Fig. 1. A multi-hop secondary network overlaid on a multi-hop primary network.

The remainder of this paper is organized as follows. In Section II, we give a centralized problem formulation under transparent coexistence for a multi-hop primary and secondary networks. Section III presents our design of an iterative distributed algorithm to achieve the transparent coexistence for a secondary multi-hop network. Section IV presents simulations results and demonstrates the competitive performance of the proposed distributed algorithm. Section V concludes this paper.

II. A Centralized Problem Formulation of Transparent Coexistence

We consider a multi-hop primary network $\mathcal{P}$, where each node is only equipped with a single antenna. This primary network is co-located with a multi-hop secondary network $\mathcal{S}$ in the same geographical region, as shown in Fig. 1. For the secondary network, we assume that each node is equipped with MIMO, which has the IC capability that is required to achieve transparent coexistence.

Suppose that the primary and secondary networks share the same spectrum channel in the time domain, with T time slots in a frame. For the primary network, each node can access any time slot without any consideration of the secondary network. A secondary node, however, is allowed to use a time slot only if its activity is transparent to the primary network.

Having accurate channel state information (CSI) is critical for the secondary nodes to cancel interference to/from primary nodes. The problem here is: how can a secondary node obtain the CSI between itself and its neighboring primary nodes while remaining transparent to the primary nodes? We propose the following solution to resolve this problem. For each primary node, it typically sends out a pilot sequence (training sequence) to its neighboring primary nodes such that they can estimate the CSI between them for communication. This is the practice for current cellular networks and we assume such a mechanism is available for a primary network. We will exploit this feature for the secondary nodes to estimate CSI. Specifically, the secondary nodes can overhear the pilot sequence signal from the primary node while staying transparent. Suppose the pilot sequence from the primary nodes is publicly available (as in cellular networks) and is known to the secondary nodes. Then the secondary nodes can use this information and the actual received pilot sequence signal from the primary node for channel estimation. Based on the reciprocity property of a wireless channel [6], the estimated CSI can also be used as CSIT (channel state information at transmitter side). Similarly, for each secondary node, it can send out pilot sequence to its neighboring secondary nodes such that those nodes can estimate the CSI. Therefore, a secondary node can obtain complete CSI between itself and its neighboring primary and secondary neighboring nodes.

Suppose that there is a set of sessions $\tilde{\mathcal{F}}$ in the primary network $\mathcal{P}$. The routing for each primary session can be found by standard routing protocol, e.g., OLSR or AODV. Denote $\tilde{\mathcal{L}}$ as the set of active links associated with sessions $\tilde{\mathcal{F}}$. Denote $\tilde{z}^{(\tilde{l})}(t)$ as the number of data streams on link $\tilde{l} \in \tilde{\mathcal{L}}$ in time slot t. Since each primary node has only a single antenna, $\tilde{z}^{(\tilde{l})}(t) = 1$ if link $\tilde{l}$ is active in time slot t and 0 otherwise.

For the secondary network $\mathcal{S}$, suppose that there is a set of sessions $\mathcal{F}$ in $\mathcal{S}$. Again, the routing of each secondary session can be found by standard routing protocol. Denote $\mathcal{L}$ as the set of links associated with sessions $\mathcal{F}$. Denote A_i as the number of antennas on a node $i \in \mathcal{S}$. Denote $z^{(l)}(t)$ as the number of data streams on link $l \in \mathcal{L}$ in time slot t. For each secondary node, it needs to know the neighboring primary link scheduling, which can be obtained by monitoring/sensing its neighboring primary nodes' activities.

Time Slot Scheduling. Denote $x_i(t)$ as a binary variable indicating whether or not secondary node $i \in \mathcal{S}$ is a transmitter in time slot t. That is, $x_i(t) = 1$ if node i is transmitting in t and 0 otherwise. Similarly, denote $y_i(t)$ as a binary variable indicating whether or not secondary node $i \in \mathcal{S}$ is a receiver in time slot t. We assume that the secondary transceiver is half-duplex. Since a node $i \in \mathcal{S}$ cannot transmit and receive in the same time slot, we have:

$$x_i(t) + y_i(t) \leq 1 \qquad (i \in \mathcal{S}, 1 \leq t \leq T). \tag{1}$$

SM and IC at a Secondary Transmitter. If a secondary node is a transmitter in a time slot, it should allocate DoF for SM and IC. For SM, denote $\mathcal{L}_i^{\text{Out}}$ as the set of outgoing links from node i. Then, the number of DoFs at secondary node $i \in \mathcal{S}$ that is used for SM is $\sum_{l \in \mathcal{L}_i^{\text{Out}}} z^{(l)}(t)$. For IC, the secondary transmitter should perform both inter-network IC (i.e., canceling its interference to its neighboring primary receivers) and intra-network IC (i.e., canceling its interference to a "subset" of its neighboring secondary receivers).

(i) For IC to its neighboring primary receivers, denote $\tilde{\mathcal{I}}_i$ as the set of neighboring primary nodes that are located within the interference range of node i. Denote $\tilde{\mathcal{L}}_p^{\text{In}}$ as the set of incoming primary links to node $p \in \tilde{\mathcal{I}}_i$. Then, the number of DoFs required for IC to node i's neighboring primary receivers is $\left(\sum_{p \in \tilde{\mathcal{I}}_i} \sum_{\tilde{l} \in \tilde{\mathcal{L}}_p^{\text{In}}} \tilde{z}^{(\tilde{l})}(t) \right)$.

(ii) For IC to node i's neighboring secondary receivers, we employ the node-ordering scheme proposed in [5]. The idea is to put all the nodes in the network into an ordered node list. An optimal ordering of such a node list is part of the optimization problem. Denote $\pi(t)$ as an ordered list of the secondary nodes in time slot t ($1 \leq t \leq T$) and $\pi_i(t)$ as the position of node $i \in \mathcal{S}$ in $\pi(t)$. Therefore,

$$1 \leq \pi_i(t) \leq S\ , \quad (i \in \mathcal{S}, 1 \leq t \leq T, S = |\mathcal{S}|). \tag{2}$$

To model the relative ordering between any two secondary nodes i and j in $\pi(t)$, we denote a binary variable $\theta_{j,i}(t)$ as the relative position of node i and node j in $\pi(t)$ as follows: $\theta_{j,i}(t) = 1$ if node j is before node i and 0 otherwise. Then, we have

$$\pi_i(t) - S \cdot \theta_{j,i}(t) + 1 \leq \pi_j(t) \leq \pi_i(t) - S \cdot \theta_{j,i}(t) + S - 1\ , \tag{3}$$

where $(i, j \in \mathcal{S}, 1 \leq t \leq T)$. Denote $\mathcal{I}_i$ as the set of neighboring secondary nodes that are located within the interference range of node i, and $\mathcal{L}_j^{\text{In}}$ as the set of incoming secondary links to node $j \in \mathcal{S}$. It was shown in [5] that node i only needs to cancel its interference to these secondary receivers that are before itself in the ordered node list $\pi(t)$. Node i does not need to be concerned with its interference to those secondary receivers that are after itself in $\pi(t)$ as such interference will be canceled by those nodes later when we consider them in the ordered node list. Therefore, the

number of DoFs required for IC to secondary receivers is $\sum_{j\in\mathcal{I}_i}\left(\theta_{j,i}(t)\cdot\sum_{k\in\mathcal{L}_j^{\rm In}}^{{\rm Tx}(k)\neq i} z^{(k)}(t)\right)$.

Combining DoF consumption for SM and IC, we have:

$$\sum_{l\in\mathcal{L}_i^{\rm Out}} z^{(l)}(t)+\left(\sum_{p\in\tilde{\mathcal{I}}_i}\sum_{\tilde{l}\in\tilde{\mathcal{L}}_p^{\rm In}}\tilde{z}^{(\tilde{l})}(t)\right)+\sum_{j\in\mathcal{I}_i}\left(\theta_{j,i}(t)\cdot\sum_{k\in\mathcal{L}_j^{\rm In}}^{{\rm Tx}(k)\neq i} z^{(k)}(t)\right)\le A_i\ , \tag{4}$$

which means the total number of DoFs consumed at this transmitter for SM and IC cannot exceed the total number of its DoFs.

SM and IC at a Secondary Receiver. If a secondary node is a receiver in a time slot, it should allocate DoFs both for SM and IC. For SM, The number of DoFs used for SM at a secondary receiver i in time slot t is $\sum_{k\in\mathcal{L}_i^{\rm In}} z^{(k)}(t)$. For IC, the secondary receiver should perform both inter-network IC (i.e., canceling the interference from its neighboring primary transmitters) and intra-network IC (i.e., canceling the interference from a "subset" of its neighboring secondary transmitters). Denote $\tilde{\mathcal{L}}_p^{\rm Out}$ as the set of outgoing primary links from primary node $p\in\mathcal{P}$. Then at secondary receiver i, the number of DoFs used for IC from primary transmitters is $\left(\sum_{p\in\tilde{\mathcal{I}}_i}\sum_{\tilde{l}\in\tilde{\mathcal{L}}_p^{\rm Out}}\tilde{z}^{(\tilde{l})}(t)\right)$. To cancel the interference from its neighboring secondary transmitters, we again employ the node-ordering scheme. Similar to that in the secondary transmitter case that was discussed earlier, node i only needs to cancel the interference from these secondary transmitters that are before itself in the ordered node list $\pi(t)$. Node i does not need to be concerned with the interference from those secondary transmitters that are after itself in $\pi(t)$ as such interference will be canceled by those nodes later when we consider them in the ordered node list. At receive node i, the number of DoFs used to cancel interference from other secondary transmitters is $\sum_{j\in\mathcal{I}_i}\left(\theta_{j,i}(t)\cdot\sum_{l\in\mathcal{L}_j^{\rm Out}}^{{\rm Rx}(l)\neq i} z^{(l)}(t)\right)$. Summing up DoF allocations for SM and IC at a secondary receiver i, we have:

$$\sum_{k\in\mathcal{L}_i^{\rm In}} z^{(k)}(t)+\left(\sum_{p\in\tilde{\mathcal{I}}_i}\sum_{\tilde{l}\in\tilde{\mathcal{L}}_p^{\rm Out}}\tilde{z}^{(\tilde{l})}(t)\right)+\sum_{j\in\mathcal{I}_i}\left(\theta_{j,i}(t)\cdot\sum_{l\in\mathcal{L}_j^{\rm Out}}^{{\rm Rx}(l)\neq i} z^{(l)}(t)\right)\le A_i\ . \tag{5}$$

Link Capacity Constraint. For simplicity, we assume that fixed modulation and coding scheme (MCS) is used for each data stream and that each data stream corresponds to one unit data rate. Denote $r(f)$ as the rate of session $f\in\mathcal{F}$. Then, for each link $l\in\mathcal{L}$, we have the following link capacity constraint:

$$\sum_{f\in\mathcal{F}}^{f\ \text{traversing}\ l} r(f)\le\frac{1}{T}\sum_{t=1}^{T} z^{(l)}(t)\qquad(l\in\mathcal{L}). \tag{6}$$

A. Problem Formulation

For the centralized problem formulation, we consider a throughput optimization problem, with the objective of maximizing the minimum session rate $r_{\min}$ among all secondary sessions. The optimization problem can be written as follows:

max $r_{\min}$
s.t $r_{\min}\le r(f)\qquad(f\in\mathcal{F})$;
Half duplex constraints: (1);
Node ordering constraints: (2), (3);
SM and IC at secondary transmitters: (4);
SM and IC at secondary receivers: (5);
Link capacity constraints: (6).

The above formulation is in the form of mixed integer nonlinear programming (MINLP), but can be reformulated into a mixed integer linear programming (MILP) through *Reformulation-Linearization Technique*(RLT) [3, Chapter 6]. However, a MILP problem is still NP-hard in general. The goal of this paper is to design a distributed algorithm that can offer competitive results to this problem.

III. A Distributed Solution

We propose an efficient distributed scheduling algorithm to achieve transparent coexistence of the multi-hop secondary network with the primary network. The main idea is to increase the number of data streams for each link by 1 (by considering all active links) during an iteration. For each link, we try to increase the number of DoFs for SM on this link by 1 at both the transmitter and the receiver. This increment is successful only if the neighboring interference constraints are satisfied at both transmit and receive nodes of this link. If this increment is not successful for any link during an iteration, we will then try to make a local adjustment of node ordering, with the goal of freeing up some DoFs for SM/IC after the adjustment. Again, a local node ordering adjustment is successful only if the DoF constraints at each node are satisfied after the adjustment. At any iteration when a local node ordering adjustment is not successful (and thus the number of data streams on the associated link cannot be further increased), our algorithm terminates.

A. Step 1: Choosing a Link

The goal of this step is to choose each active link during an iteration (for data stream increment). If a link is traversed by multiple sessions, it is necessary to represent the link multiple times to ensure that each session is considered for data stream increment. There are many ways to achieve this. In our solution, we propose to employ the so-called distributed ranking algorithm by Zaks [10]. This algorithm was designed to solve the problem of sorting and ranking n processors in a distributed system. The input is an initial value for each processor (not necessarily distinct). The output is a ranking of all n processors.

To apply the distributed ranking algorithm, we need to assign an initial value for each link. This value will be randomly generated and maintained by the transmitter of the link. After applying the distributed ranking algorithm, the transmitter of each link will maintain this link's rank.

After each link obtains its rank, it will know precisely which time slot it will be chosen for data stream increment. Specifically, for link l with rank(l), it will be chosen in the [kL+rank(l)]-th time slot,[1] where $k = 0, 1, 2, \cdots$, and L is the total number of active secondary links.

B. Step 2: Data Stream Increment

Once a link is chosen (in Step 1), we attempt to increase one data stream (for SM) on the selected link, while satisfying IC and transparency to the primary network. We first present the data structure that is maintained at each secondary node. Then we present the necessary conditions under which one more data stream can be added on the link in a time slot. Finally, we describe how to update state information on the nodes that are involved in this increment.

State Information In our algorithm, a secondary node maintains the following state information:

- $s_i(t)$: The status of node i in time slot t, i.e., $s_i(t) =$ Tx, Rx or Idle represents that node i is a transmitter, receiver or idle in time slot t.
- $\omega_i(t)$: A list that represents a local ordering of nodes within node i's interference range (including node i).
- $\mathcal{B}_i(t)$: The set of nodes that are before node i in $\omega_i(t)$ in time slot t.
- $\mathcal{Y}_i(t)$: The set of nodes that are after node i in $\omega_i(t)$ in time slot t.
- $\lambda_i^{\mathrm{SM}}(t)$: The number of DoFs that node i has allocated for SM in time slot t.
- $\lambda_i^{\mathrm{IC}}(t)$: The number of DoFs that node i has allocated for IC in time slot t.
- $\lambda_i^{\mathrm{RM}}(t)$: The number of remaining DoFs at node $i \in \mathcal{S}$ in time slot t, i.e., $\lambda_i^{\mathrm{RM}}(t) = A_i - \lambda_i^{\mathrm{SM}}(t) - \lambda_i^{\mathrm{IC}}(t)$.
- $\tilde{\alpha}_i(t)$: The total number of data streams transmitted by those primary transmitters that may interfere with node i's reception in time slot t, i.e., $\tilde{\alpha}_i(t) = \sum_{p \in \tilde{\mathcal{I}}_i} \sum_{\tilde{l} \in \tilde{\mathcal{L}}_p^{\mathrm{Out}}} \tilde{z}^{(\tilde{l})}(t)$.
- $\tilde{\beta}_i(t)$: The total number of data streams received by those primary receivers that are within node i's interference range in time slot t, i.e., $\tilde{\beta}_i(t) = \sum_{p \in \tilde{\mathcal{I}}_i} \sum_{\tilde{l} \in \tilde{\mathcal{L}}_p^{\mathrm{In}}} \tilde{z}^{(\tilde{l})}(t)$.
- $\alpha_i(t)$: The total number of data streams transmitted by those secondary transmitters that may interfere with node i's reception in time slot t, i.e., $\alpha_i(t) = \sum_{j \in \mathcal{I}_i} \sum_{k \in \mathcal{L}_j^{\mathrm{Out}}}^{\mathrm{Rx}(k) \neq i} z^{(k)}(t)$.
- $\beta_i(t)$: The total number of data streams received by those secondary receivers that are within node i's interference range in time slot t, i.e., $\beta_i(t) = \sum_{j \in \mathcal{I}_i} \sum_{k \in \mathcal{L}_j^{\mathrm{In}}}^{\mathrm{Tx}(k) \neq i} z^{(k)}(t)$.
- $z_{i,j}(t)$: The number of data streams transmitted from node i to node j.

[1]This time slot refers to scheduling in a control channel, which is different from scheduling for data transmission in the data channel.

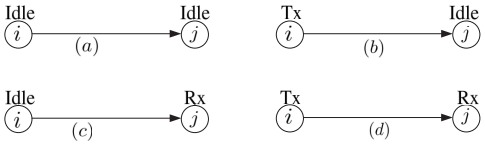

Fig. 2. Four cases of link status.

During the initialization stage, each node is set to Idle, i.e., $s_i(t) =$Idle for $i \in \mathcal{S}, t = 1, 2, \cdots, T$. The initial list $\omega_i(t)$ is set to $\emptyset$ for $i \in \mathcal{S}$, indicating that a local ordering of nodes is null. As a result, $\mathcal{B}_i(t) = \emptyset$ and $\mathcal{Y}_i(t) = \emptyset$ for $i \in \mathcal{S}$. Since the initial DoF allocation for SM and IC at each node is 0, we have $\lambda_i^{\mathrm{SM}}(t) = \lambda_i^{\mathrm{IC}}(t) = 0$, $\lambda^{\mathrm{RM}}(t) = A_i$ and $z_{i,j}(t) = 0$ for $i, j \in \mathcal{S}, t = 1, 2, \cdots, T$. $\tilde{\alpha}_i(t)$ and $\tilde{\beta}_i(t)$ are constants and are calculated based on active sessions in the primary network. On the other hand, the initial values for $\alpha_i(t)$ and $\beta_i(t)$ are 0. Except that $\tilde{\alpha}_i(t)$ and $\tilde{\beta}_i(t)$ are constants, the values for $s_i(t), \omega_i(t), \mathcal{B}_i(t), \mathcal{Y}_i(t), \lambda_i^{\mathrm{SM}}(t), \lambda_i^{\mathrm{IC}}(t), \lambda^{\mathrm{RM}}(t), z_{i,j}(t), \alpha_i(t)$ and $\beta_i(t)$ are variables and will be updated during each iteration of the algorithm.

Sufficient Conditions for Data Stream Increment. We now discuss when the number of data streams on a chosen link can be incremented by 1 in a given time slot. Suppose link (i, j) is the link. Then both nodes i and j first check their current status ("Tx", "Rx", or "Idle"). Some cases can be clearly ruled out for consideration, i.e., $s_i(t) =$ Rx or $s_j(t) =$ Tx. In this case, link (i, j) cannot be considered for data stream increment in t and we move to the next time slot $(t + 1)$ immediately. Otherwise, if link (i, j) is suitable for stream increment, there are four possible states as shown in Figure 2. The sufficient conditions for data stream increment on link (i, j) are as follows:

Case (a): $s_i(t) =$ Idle and $s_j(t) =$ Idle.

- $s_i(t) =$ Idle: Since node i is idle, its local ordering list $\omega_i(t)$ is empty. We need to establish a new $\omega_i(t)$. But the local ordering of node i's active neighboring nodes are not yet established. To get around this issue, we can consider putting node i either as the first or the last node in $\omega_i(t)$, by treating the relative ordering of the other nodes in $\omega_i(t)$ as a blackbox (i.e., without explicit knowledge of its details).
  - If node i is at the beginning of $\omega_i(t)$, then the following two conditions must be satisfied: (i) the total number of DoFs at node i should be greater than the total number of data streams received by its neighboring primary receivers, i.e., $A_i > \tilde{\beta}_i(t)$, (ii) all secondary receivers that are after node i in $\omega_i(t)$ must have at least one remaining DoF to cancel one more interference stream from node i.
  - If node i is at the end of $\omega_i(t)$, the following condition must be satisfied: the total number of DoFs

at node i is more than the sum of data streams received by both neighboring primary and secondary receivers, i.e., $A_i > \tilde{\beta}_i(t) + \beta_i(t)$.

- $s_j(t) =$ Idle: Similar to node i, we consider putting receive node j either as the first or the last node in $\omega_j(t)$ by treating the relative ordering of the other nodes in $\omega_j(t)$ as a blackbox. The sufficient conditions for j as the first or the last node are similar as i, we omit its discussion here.

If the conditions for $s_i(t) =$ Idle and $s_j(t) =$ Idle are both satisfied, we proceed with this increment and update state information at nodes i and j and their neighboring nodes according to Figure 3 and Figure 4.

State update at idle node i and neighboring receiver k

1. If $s_i(t) =$ Idle:
2. Update $s_i(t) =$ Tx; $\lambda_i^{\mathrm{SM}}(t) \leftarrow \lambda_i^{\mathrm{SM}}(t) + 1$; $\lambda_i^{\mathrm{RM}}(t) \leftarrow \lambda_i^{\mathrm{RM}}(t) - 1$; $z_{i,j}(t) \leftarrow z_{i,j}(t) + 1$.
3. If node i is put at the beginning of $\omega_i(t)$:
4. $\mathcal{Y}_i(t) \leftarrow$ {Neighboring active secondary receivers.}
5. Update $\lambda_i^{\mathrm{IC}}(t) \leftarrow \tilde{\beta}_i(t)$; $\lambda_i^{\mathrm{RM}}(t) \leftarrow \lambda_i^{\mathrm{RM}}(t) - \lambda_i^{\mathrm{IC}}(t)$;
6. For each receive node $k \in \mathcal{Y}_i(t)$:
7. $\mathcal{B}_k(t) \leftarrow \mathcal{B}_k(t) \cup \{i\}$; $\lambda_k^{\mathrm{IC}}(t) \leftarrow \lambda_k^{\mathrm{IC}}(t) + 1$; $\lambda_k^{\mathrm{RM}}(t) \leftarrow \lambda_k^{\mathrm{RM}}(t) - 1$.
8. Else if node i is put at the end of $\omega_i(t)$:
9. $\mathcal{B}_i(t) \leftarrow$ {Neighboring active secondary receivers.}
10. Update $\lambda_i^{\mathrm{IC}}(t) \leftarrow \tilde{\beta}_i(t) + \beta_i(t)$; $\lambda^{\mathrm{RM}}(t) \leftarrow \lambda^{\mathrm{RM}}(t) - \lambda_i^{\mathrm{IC}}(t)$.
11. For each node $k \in \mathcal{B}_i(t)$:
12. $\mathcal{Y}_k(t) = \mathcal{Y}_k(t) \cup \{i\}$.

Fig. 3. Pseudocode to update state information when $s_i(t) =$ Idle.

State update at idle node j and neighboring transmitter k

1. If $s_j(t) =$ Idle:
2. Update $s_j(t) =$ Rx; $\lambda_j^{\mathrm{SM}}(t) \leftarrow \lambda_j^{\mathrm{SM}}(t) + 1$; $\lambda_j^{\mathrm{RM}}(t) \leftarrow \lambda_j^{\mathrm{RM}}(t) - 1$; $z_{i,j}(t) \leftarrow z_{i,j}(t) + 1$.
3. If node j is put at the beginning of $\omega_j(t)$:
4. $\mathcal{Y}_j(t) \leftarrow$ {Neighboring active secondary transmitters.}
5. Update $\lambda_j^{\mathrm{IC}}(t) \leftarrow \tilde{\beta}_j(t)$; $\lambda_j^{\mathrm{RM}}(t) \leftarrow \lambda_j^{\mathrm{RM}}(t) - \lambda_j^{\mathrm{IC}}(t)$
6. For each transmit node $k \in \mathcal{Y}_j(t)$:
7. $\mathcal{B}_k(t) \leftarrow \mathcal{B}_k(t) \cup \{j\}$; $\lambda_k^{\mathrm{IC}}(t) = \lambda_k^{\mathrm{IC}}(t) + 1$; $\lambda_k^{\mathrm{RM}}(t) = \lambda_k^{\mathrm{RM}}(t) - 1$.
8. Else if node j is put at the end of $\omega_j(t)$:
9. $\mathcal{B}_j(t) \leftarrow$ {Neighboring active secondary transmitters.}
10. Update $\lambda_j^{\mathrm{IC}}(t) \leftarrow \tilde{\alpha}_j(t) + \alpha_j(t)$; $\lambda^{\mathrm{RM}}(t) \leftarrow \lambda_j^{\mathrm{RM}}(t) - \lambda_j^{\mathrm{IC}}(t)$.
11. For each $k \in \mathcal{B}_j(t)$:
12. $\mathcal{Y}_k(t) \leftarrow \mathcal{Y}_k(t) \cup \{j\}$.

Fig. 4. Pseudocode to update state information when $s_j(t) =$ Idle.

Case (b): $s_i(t) =$ Tx and $s_j(t) =$ Idle.

- $s_i(t) =$ Tx : In this case, the following conditions must be satisfied if node i wants to increase one more data stream on link (i, j): (i) node i has at least one remaining DoF for SM, i.e., $\lambda_i^{\mathrm{RM}}(t) \geq 1$; (ii) each receive node k in $\omega_i(t)$ that is after node i, i.e., $k \in \mathcal{Y}_i(t)$, has at least one remaining DoF to cancel the new interference from node i.
- $s_j(t) =$ Idle : This case has been discussed in *Case (a)*.

If the conditions for $s_i(t) =$ Tx and $s_j(t) =$ Idle are both satisfied, we proceed with this increment and update state information at nodes i, j and their neighboring nodes according to Figure 5 and Figure 4.

State update at transmit node i and neighboring receiver k

1. If $s_i(t) =$ Tx:
2. Update $\lambda_i^{\mathrm{SM}}(t) \leftarrow \lambda_i^{\mathrm{SM}}(t) + 1$; $\lambda_i^{\mathrm{RM}}(t) \leftarrow \lambda_i^{\mathrm{RM}}(t) - 1$; $z_{i,j}(t) = z_{i,j}(t) + 1$.
3. For each receive node $k \in \mathcal{Y}_i(t)$:
4. Update $\lambda_k^{\mathrm{IC}}(t) \leftarrow \lambda_k^{\mathrm{IC}}(t) + 1$; $\lambda_k^{\mathrm{RM}}(t) \leftarrow \lambda_k^{\mathrm{RM}}(t) - 1$.

Fig. 5. Pseudocode to update state information when $s_i(t) =$ Tx.

Case (c): $s_i(t) =$ Idle and $s_j(t) =$ Rx.

- $s_i(t) =$ Idle: This case has been discussed in Case (a).
- $s_j(t) =$ Rx: In this case, the following conditions must be satisfied if node j wants to increase one more data stream on link (i, j): (i) node j has at least one remaining DoF for SM, i.e., $\lambda_j^{\mathrm{RM}}(t) \geq 1$; (ii) each transmit node k in $\omega_j(t)$ that is after node j, i.e., $k \in \mathcal{Y}_j(t)$ has at least one remaining DoF to cancel its interference to node j.

If the conditions for $s_i(t) =$ Idle and $s_j(t) =$ Rx are both satisfied, we proceed with this increment and update state information at nodes i, j and their neighboring nodes according to Figure 3 and Figure 6.

State update at receive node j and neighboring transmitter k

1. If $s_j(t) =$ Rx:
2. Update $\lambda_j^{\mathrm{SM}}(t) \leftarrow \lambda_j^{\mathrm{SM}}(t) + 1$; $\lambda_j^{\mathrm{RM}}(t) \leftarrow \lambda_j^{\mathrm{RM}}(t) - 1$; $z_{i,j}(t) = z_{i,j}(t) + 1$.
3. For each transmit node $k \in \mathcal{Y}_j(t)$:
4. Update $\lambda_k^{\mathrm{IC}}(t) \leftarrow \lambda_k^{\mathrm{IC}}(t) + 1$; $\lambda_k^{\mathrm{RM}}(t) \leftarrow \lambda_k^{\mathrm{RM}}(t) - 1$.

Fig. 6. Pseudocode to update state information when $s_j(t) =$ Rx.

Case (d): $s_i(t) =$ Tx and $s_j(t) =$ Rx. The case for $s_i(t) =$ Tx has been discussed in *Case (b)* and $s_j(t) =$ Rx has been discussed in *Case (c)*.

If the conditions for $s_i(t) =$ Tx and $s_j(t) =$ Rx are both satisfied, we proceed with this increment and update state information at nodes i, j and their neighboring nodes according to Figure 5 and Figure 6.

Recall that there are T time slots in a time frame. If the data stream increment operation described above fails in the first time slot, we try it again in the second time slot and so forth, until a data stream increment is successful in a time slot or fails after all T time slots.

C. Step 3: Adjusting Local Node Ordering

If the sufficient conditions at either node i or node j cannot be satisfied, we move on to this step. The only reason why link (i, j) fails to increase one data stream in step 2 is the lack of DoF resources at some nodes. Since the local ordering of a node directly affects the node's DoF consumption for IC (see (4) and (5)), we will try to adjust a node's local ordering and thus change its DoF consumption for IC. This can be done by swapping the position of the bottleneck node with

some other node ahead of itself in the local node ordering, thereby transferring the IC responsibility from itself to the other node. Since some new DoF resources for the bottleneck become available, a new data stream increment on link (i, j) may be possible.

The main idea of this step is as follows. For each time slot t, we identify the set of bottleneck nodes (denoted as $\mathcal{D}_{(i,j)}(t)$), which do not have enough remaining DoF resources should one more data stream is added onto link (i, j). For each node $k \in \mathcal{D}_{(i,j)}(t)$, we perform local ordering adjustment for k by swapping its position with some other node before k in its local node ordering. To ensure feasibility, only a subset of nodes (denoted as $\bar{\mathcal{B}}_k(t)$), $\bar{\mathcal{B}}_k(t) \subseteq \mathcal{B}_k(t)$, is eligible for swapping with k. After identifying $\bar{\mathcal{B}}_k(t)$ for k, we consider nodes in $\bar{\mathcal{B}}_k(t)$ in the order of non-increasing remaining DoFs, i.e., starting with the one that has the maximum remaining DoF (denoted as node a) after it is swapped with node k. If this swap is infeasible, then local node ordering adjustment fails in this time slot and we move on to the next time slot. Otherwise, we swap k and a and update their state information. In its new position, if a new data stream can be added on link (i, j), we are done. Otherwise, we continue by considering swapping k with the next node in $\bar{\mathcal{B}}_k(t)$ that has the maximum remaining DoF (denoted as node b) following the same process. The algorithm terminates upon a new data stream can be successfully added on link (i, j) or all nodes in $\mathcal{D}_{(i,j)}(t)$ are considered for all time slots in a frame.

IV. Simulation Results

We present simulation results to demonstrate the performance of the proposed distributed algorithm. Since the centralized problem formulation is MILP, which is NP-hard in general, we cannot obtain the optimal solution for comparison. Instead, we will compare the performance of our algorithm to an upper bound of the objective for the centralized problem. Such an upper bound can be obtained by running CPLEX for a given termination time (e.g., 8 hours in our study). Such a comparison approach is very conservation. This is because the optimal objective value (not obtainable) to the centralized problem lies between the upper bound and the feasible solution obtained by our distributed algorithm. Therefore, if the feasible solution from our distributed algorithm is close to the upper bound by CPLEX, then we can claim that our solution (objective) is even closer to the optimal objective and thus is competitive.

We consider a secondary CR network co-locates with a primary network within an area of 100×100. For generality, we normalize the units for distance, bandwidth, and data rate with appropriate dimensions. Each node (both primary and secondary) is randomly deployed inside the 100×100 area. The primary nodes are traditional single-antenna node while the secondary nodes are equipped with MIMO, with four antennas on each node. We assume that each node's transmission range and interference range are 30 and 50, respectively. We assume a time frame is divided into $T = 10$ time slots.

We show results for one network instance, with 20 primary nodes and 30 secondary nodes. The locations of each node are shown in Figure 7.We assume there are three primary sessions and four secondary sessions, with each session's source and destination nodes shown in this figure. We assume that minimum-hop routing is used for each primary and secondary session. Figure 7 shows the routing in primary and secondary network (where the solid arrows represent primary links and the dashed arrows represent the secondary links). Scheduling for the primary and secondary links is also given in this figure, where numbers in the box represents the time slots used by either a primary or secondary link. Note that scheduling for the primary links is decided by the primary network, while scheduling for each secondary link is found by our distributed algorithm.

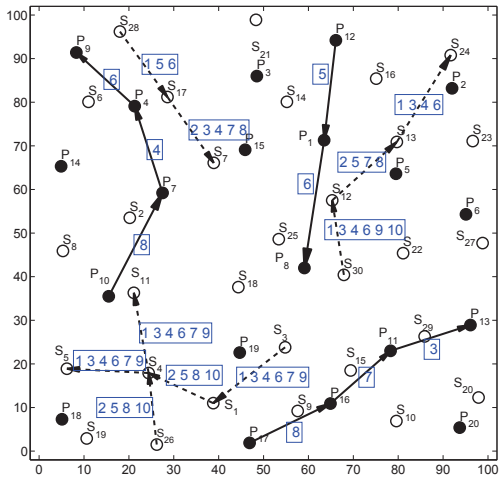

Fig. 7. Routing for each session and scheduling on each link for both primary and secondary networks. The numbers in the box next to a link show the time slots when the link is active.

The objective value obtained from our distributed algorithm is 0.6 (in less than a second computation time). On the other hand, the upper bound obtained by CPLEX is 0.7 (with a cut-off time of 8 hours). As discussed, since the optimal solution lies between 0.6 and 0.7, our objective value (0.6) should be very close to the unknown optimal.

To show the transparent coexistence between primary and secondary networks, we focus on one time slot, say 6. Figure 8 shows the set of active links in time slot 6 for both networks. In this time slot, transparent coexistence is achieved on secondary links $S_{28} \to S_{17}, S_{13} \to S_{24}, S_{30} \to S_{12}, S_3 \to S_1, S_4 \to S_{11}$ and $S_4 \to S_5$ We first consider inter-network IC:

- For secondary link $S_{28} \to S_{17}$, its interference to P_9 on primary link $P_4 \to P_9$ is canceled by S_{28} with 1 DoF, while the interference from P_4 and P_1 to S_{17} is canceled by S_{17}, each with 1 DoF.
- For secondary links $S_3 \to S_1, S_{30} \to S_{12}, S_{13} \to S_{24}, S_4 \to S_{11}$ and $S_4 \to S_5$, the interference from their transmitters $(S_3, S_{30}, S_{13}, S_4)$ to receiver P_8 on primary link $P_1 \to P_8$ is canceled by S_3, S_{30}, S_{13} and S_4, each with 1 DoF. The interference from P_1 to S_{12} and S_{24} is

canceled by S_{12} and S_{24} with 1 DoF, respectively, and the interference from P_4 to S_{11} is canceled by S_{11} with 1 DoF.

For intra-network IC within the secondary network, our solution shows that:

- S_{11} is canceling interference from S_3 and S_4, each with 1 DoF.
- S_5 is canceling interference from S_3 and S_4, each with 1 DoF.
- The interference from S_4 to S_1 is canceled by S_1 with 1 DoF.
- The interference from S_3 to S_{12} is canceled by S_{12} with 1 DoF.
- The interference from S_{13} to S_{12} is canceled by S_{13} with 2 DoFs.
- The interference from S_{30} to S_1 and S_{11} is canceled by S_{30}, each with 1 DoF.

The details of DoF allocation for SM and IC at each active secondary node in time slot 6 are shown in Table I. In this table, the second and third columns represent the set of secondary nodes that are before and after this node in its local ordered list, respectively. The fourth column represents the number of DoFs allocated for SM. The fifth column represents the number of DoFs that are allocated for IC to/from primary network. The last column represents the number of DoFs allocated for IC for the set of secondary nodes before itself in the local ordered list within the secondary network.

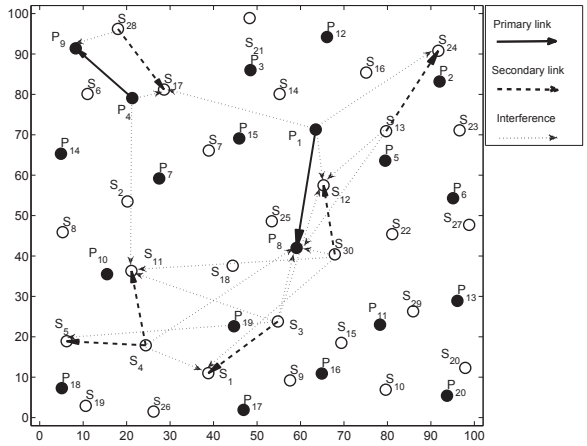

Fig. 8. Active links in time slot 6 in both primary and secondary networks.

V. Conclusions

Transparent coexistence paradigm is attracting attention as it offers much improved performance than traditional interweave paradigm. In this paper, we studied how to design a distributed optimization algorithm to achieve transparent coexistence for multi-hop primary and secondary networks. By employing MIMO at each secondary node for IC, we showed that a distributed algorithm can be designed to ensure inter-network interference is canceled by the secondary nodes while undesirable intra-network interference can be canceled within the secondary network. We showed that each step in the distributed algorithm can be accomplished through local computation based on information exchange among the neighboring nodes. Through simulation study and comparing our results to an upper bound result, we conclude that the distributed algorithm offers competitive throughput performance when achieving transparent coexistence.

TABLE I
DoF allocation for SM and IC at each active secondary node in time slot 6.

Node i	$\mathcal{B}_i(t)$	$\mathcal{Y}_i(t)$	DoF for SM	IC to/from primary	DoF for IC within secondary network
1	$\{S_4\}$	$\{S_{30}\}$	1	0	2
3		$\{S_5, S_{11}, S_{12}\}$	1	1	0
4		$\{S_1, S_{11}\}$	2	1	0
5	$\{S_3, S_4\}$		1	0	2
11	$\{S_3, S_4\}$	$\{S_{30}\}$	1	1	2
12	$\{S_3\}$	$\{S_{13}\}$	2	1	1
13	$\{S_{12}\}$		1	1	2
17			1	2	0
24			1	1	0
28			1	1	0
30	$\{S_1, S_{11}\}$		1	1	2

Acknowledgments

This work was supported in part by the NSF, ONR, and Virginia Tech Institute for Critical Technology and Applied Science. The work of Dr. S. Kompella was supported in part by the ONR. Part of Prof. W. Lou's work was completed while she was serving as a Program Director at the NSF. Any opinion, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not reflect the views of the U.S. government.

References

[1] F. Gao, R. Zhang, Y.-C. Liang, and X. Wang, "Design of learning-based MIMO cognitive radio systems," *IEEE Transactions on Vehicular Technology,* vol. 59, no. 4, pp. 1707–1720, May 2010.

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

[3] Y.T. Hou, Y. Shi, and H.D. Sherali, *Applied Optimization Methods for Wireless Networks,* Cambridge University Press, 2014, ISBN-13: 978-1107018808.

[4] S.-J. Kim and G.B. Giannakis, "Optimal resource allocation for MIMO ad hoc cognitive radio networks," *IEEE Transactions on Information Theory,* vol. 57, no. 5, pp. 3117–3131, May 2011.

[5] Y. Shi, J. Liu, C. Jiang, C. Gao, and Y.T. Hou, "A DoF-based link layer model for multi-hop mino networks," *IEEE Trans. on Mobile Computing,* vol. 12, issue 7, pp. 1395–1408, July 2014.

[6] G.S. Smith, "A direct derivation of a single-antenna reciprocity relation for the time domain." in *IEEE Trans. on Antennas and Propagation,* vol. 52, no. 6, pp. 1568–1577, June 2004.

[7] X. Yuan, C. Jiang, Y. Shi, Y.T. Hou, W. Lou, and S. Kompella, "Beyond interference avoidance: On transparent coexistence for multi-hop secondary CR networks," *Proc. IEEE SECON,* pp. 398–405, New Orleans, LA, June 24–27, 2013.

[8] R. Zhang and Y.-C. Liang, "Exploiting multi-antennas for opportunistic spectrum sharing in cognitive radio networks," *IEEE Journal of Selected Topics in Signal Processing,* vol. 2, no. 1, pp. 88–102, February 2008.

[9] Y.J. Zhang and A.M.-C. So, "Optimal spectrum sharing in MIMO cognitive radio networks via semidefinite programming," *IEEE Journal on Selected Areas in Communications,* vol. 29, no. 2, pp. 362–373, February 2011.

[10] S. Zaks, "Optimal distributed algorithms for sorting and ranking," *IEEE Trans. on Computers,* vol. 5, no. 1, pp. 376–379, April 1985.