Robust Topology Control in Multi-hop Cognitive Radio Networks

Jing Zhao and Guohong Cao
Department of Computer Science and Engineering
The Pennsylvania State University
E-mail: {juz139,gcao}@cse.psu.edu

***Abstract*—The opening of under-utilized spectrum creates the opportunity of substantial performance improvement through cognitive radio techniques. However, the real network performance may be limited since unlicensed users must vacate and switch to other available spectrum if the current spectrum is reclaimed by the licensed (primary) users. During the spectrum switching time, network partitions may occur since multiple links may be affected if they all operate on the channel reclaimed by the primary users. In this paper, we address this problem through robust topology control, where channels are assigned to minimize channel interference while maintaining network connectivity when primary users appear. To solve this NP-hard problem, we propose both centralized and distributed algorithms. Simulation results show that our solutions outperform existing interference-aware approaches substantially when primary users appear and achieve similar performance at other times.**

I. Introduction

Due to the proliferation of unlicensed wireless devices, unlicensed spectrum (e.g., ISM) is becoming increasingly congested; On the other hand, some licensed spectrum (e.g., UHF) is highly under-utilized. As a result, FCC approved unlicensed use of licensed spectrum [1] through cognitive radio techniques, which enable dynamic configuration of the operating spectrum.

To avoid interference with licensed users, unlicensed users must vacate the spectrum when it is accessed by the licensed users (or called *primary users*). This limits the overall performance of cognitive radio network due to the following reasons. First, the activity region of a primary user is usually larger than that of an unlicensed user [2], and hence the primary user appearance may affect several ongoing transmissions of the unlicensed users. Second, in order to use another available spectrum (channel), unlicensed users have to spend a considerable amount of time for spectrum sensing, neighbor discovery, and channel switching (a few milliseconds) [3]. Third, the channel switching of one unlicensed user may cause a ripple effect [4], i.e., cascaded switching of multiple unlicensed users. Moreover, it is difficult to predict when a primary user will appear in a given spectrum, and hence it is hard to address these problems.

In multi-hop cognitive radio networks, multiple links may be affected if they all operate on the channel used by the primary users. Before these links are switched to other available channels, a network partition may occur, resulting in packet dropping or significant packet delay for the affected users.

In this paper, we aim at controlling the network topology to satisfy the *robustness constraint*, i.e., the network will not be partitioned when primary users appear. Suppose node A wants to send a packet to node B. If link (A, B) is affected by the primary users, with robust topology control, the packet will be re-routed to B through another radio of A along an unaffected path, and thus packet dropping or significant packet delay can be avoided.

In cognitive radio networks, network topology is controlled by spectrum assignment [5]. Thus, we need to carefully assign the operating spectrum for each user to satisfy the robustness constraint. Since the network performance is affected by the interference caused by simultaneous transmissions on the same channel, we should also minimize the channel interference to improve performance. More specifically, we formally define a *Robust Topology Control Problem* in multi-hop multi-radio cognitive networks, where channels are assigned to minimize the channel interference while satisfying the robustness constraint. To solve this NP-hard problem, we first propose a centralized algorithm, and then design a distributed solution. Simulation results show that our solution minimizes the interference and maintains similar performance even when primary users appear.

The remainder of the paper is organized as follows. Section II reviews related work. In Section III, we formally define the robust topology control problem. Section IV and Section V present the centralized and distributed robust topology control algorithms, respectively. Simulation results are shown in Section VI. Section VII concludes the paper.

II. Related Work

Channel assignment (scheduling) in traditional wireless networks is a well-studied problem [6–8]. In the context of cognitive radio networks, both centralized approaches [9–11] and distributed approaches [12, 13] have been proposed. Most of them assume that each node has only one radio which can switch among multiple channels. Since there is only one radio, if the channel used by the radio is reclaimed by the primary user, the ongoing data transmission will be interrupted. In our work, with multiple radios, the transmission can be re-routed through another radio. Compared to existing channel assignment approaches for multi-hop multi-radio networks [14, 15],

This work was supported in part by Army Research Office under MURI grant W911NF-07-1-0318.

the channels are carefully assigned so that the primary user appearance will not partition the network.

There are some solutions to deal with the problems of spectrum handoff when primary users appear. For example, by assuming an accurate primary user behavior model, proactive spectrum handoff is performed prior to the reclamation of the licensed spectrum, to minimize the influence of sudden appearance of primary users [16]. However, the assumption of an accurate primary user behavior model is hard to achieve. To deal with this issue, techniques [3, 9] have been proposed to maintain a backup channel for the unlicensed users in case of primary user appearance, and an algorithm [17] has been proposed for updating such backup channel lists based on the cooperation among neighbors. However, these approaches cannot completely solve the spectrum handoff problem. To continue data transmission during the handoff process, authors in [18] jointly consider spectrum handoff scheduling and routing in which the transmission is re-routed through another unaffected path before handoff occurs. In order to achieve this, they assume the handoff is due to a predictable primary user activity, which is not always possible. In our approach, without assuming a predictable primary user activity, data transmission can be re-routed along an unaffected path during the spectrum switching time, and thus packet dropping or significant packet delay can be avoided.

III. Problem Definition

We consider a network of n unlicensed users, where each user has multiple radios which can be used to access C available channels. The network is modeled as an undirected graph $G(V,E)$ where each node $v \in V$ corresponds to a user, and an edge $e=(u,v) \in E$ represents the link between u and v if they are within the transmission range. Note that G is a connected graph in which any two nodes are connected by either a direct link or a path with multiple nodes.

Our goal is to find a channel assignment $\mathcal{A}$ which assigns a unique channel to each link subject to the *radio constraint*, i.e., the number of different channels assigned to the links connecting to node v should not exceed the number of radios at v. If the number of assigned channels is less than the number of radios, some other channels can be assigned to the extra radios so that the links connecting to v can operate on more than one channel, which makes the network more robust.

Considering the channels assigned to the radios at each node, a channel assignment $\mathcal{A}$ generates a new undirected graph $G_{\mathcal{A}}(V, E_{\mathcal{A}})$, where $E_{\mathcal{A}}$ consists of the edges defined as follows. There is an edge $e=(u,v;c)$ on channel c if $(u,v) \in E$ and $c \in \mathcal{A}(u) \bigcap \mathcal{A}(v)$. Here, $\mathcal{A}(u)$ ($\mathcal{A}(v)$) denotes the set of channels assigned to u (v). Note that multiple edges may exist between two neighboring nodes (u,v) if they share more than one channel, where one edge corresponds to one channel.

Due to the robustness constraint, $G_{\mathcal{A}}$ should not be partitioned by removing the edges related to the channel of the primary user. Here, we assume that the primary user can affect the entire network (e.g. transmission of the TV tower), but only one channel can be reclaimed at one time. In practical scenarios, the primary user may reclaim multiple channels but only affects a limited region. This will be discussed in Section IV-B. We assume the primary user will use the reclaimed channel for some time; otherwise, there is no need to reassign the channels (more details in Section VI-C1). The following theorem gives a sufficient condition for a robust channel assignment.

Theorem 1 If each node has at least two radios, there exists a channel assignment satisfying the robustness constraint.

Proof: One such channel assignment is to assign the same two channels to each node. When a channel is reclaimed, the network remains connected by using the other channel. ■

In this paper, we assume each node is equipped with at least two radios so that there always exists a robust channel assignment. However, the channel interference between different links may limit the network performance. For example, according to the proof of Theorem 1, a plain robust channel assignment is to assign the same two channels to each node, but this solution has severe channel interference. If the channel interference is minimized by assigning different channels for links within the interference range, the channel assignment may not be robust (more details in Section IV-A). Our goal is to find a robust channel assignment which also leads to low channel interference.

Let $\mathcal{A}(e)$ denote the channel assigned to link e by $\mathcal{A}$. Let $D(u)$ denote the set of nodes in the interference range of u. Simultaneous transmissions on two links $e=(u,v)$ and $e'=(u',v')$ interfere with each other if both of the following conditions are satisfied: (i) $\mathcal{A}(e)=\mathcal{A}(e')$, (ii) $u' \in D(e)$ or $v' \in D(e)$, where $D(e)$ denotes the set of nodes in the interference range of link $e=(u,v)$, i.e., $D(e)=D(u) \bigcup D(v)$. Transmissions on different channels can be run in parallel. Thus, to improve the network capacity [19], we should minimize the *network interference*, which is measured by the total number of links that interfere with each other.

The robust topology control problem is formalized as follows.

Definition 1 (Robust Topology Control) Given a graph G, the robust topology control problem seeks a channel assignment $\mathcal{A}$ such that $G_{\mathcal{A}} - \{(u,v;c) | (u,v;c) \in E_{\mathcal{A}}\}$ is connected for any available channel c. The network interference of G should also be minimized.

Theorem 2 The robust topology control problem is NP-hard.

Proof: The proof is by reduction to the NP-hard minimum edge coloring problem in a three-regular graph [20]. The minimum edge coloring problem is to find the minimum number of colors to color all edges in a graph. The constraint is that two edges should be assigned different colors if they share a common end node.

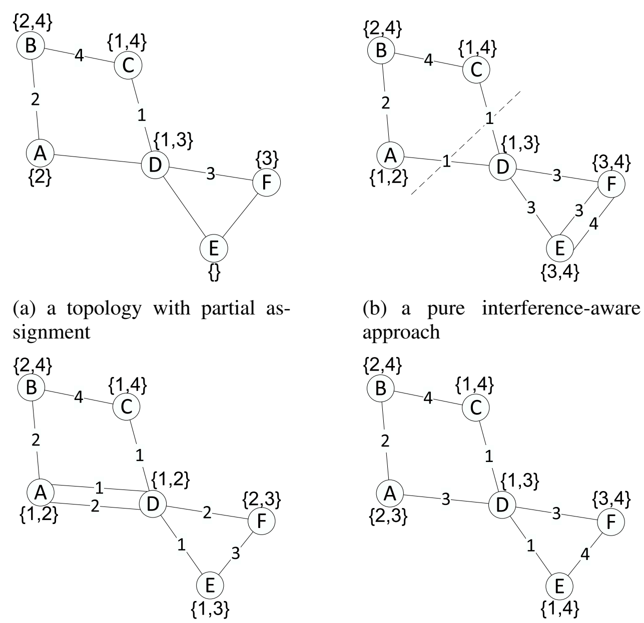
Let G be a three-regular graph in which each node has the same degree of three. Suppose each node is equipped with three radios to access three available channels. In this proof, we use a simple one-hop interference model where two links interfere only if they operate on the same channel and share a common end node. Since the radios of each node operate on the same three channels, when any one channel is reclaimed, the network remains connected by using the other two channels. Since the robustness constraint is satisfied, we only need to assign a proper channel for each link so that the network interference is minimized, making the problem identical to the minimum edge coloring problem. Since the minimum edge coloring problem is NP-hard, our robust topology control problem is NP-hard. ■

In this paper, we use the protocol interference model where all nodes have the same transmission range and the same interference range. We use 802.11 DCF [21] as the MAC layer protocol to coordinate the channel access. Half-duplex is also enforced such that each link can only support transmission in one direction at one time.

IV. Centralized Robust Topology Control Algorithm

In this section, we first introduce the design philosophy of our Centralized Robust Topology Control Algorithm (CRTCA), and then present the detailed description and analysis of CRTCA.

A. Design Philosophy



(a) a topology with partial assignment

(b) a pure interference-aware approach

(c) CRTCA

(d) Enhanced-CRTCA

Fig. 1. Illustration of design philosophy

Channels are assigned by going through the edges of G. In each step, a new channel is assigned for each edge so that the two end nodes of the edge can adjust their radios and communicate on that channel. Existing interference-aware channel assignment approaches mainly assign different channels for links within the interference range. Although these approaches can reduce channel interference, they may not be able to achieve robustness. We use Figure 1 to show the weakness of these approaches on robustness and then explain the underlying principle of CRTCA. In Figure 1, suppose all nodes have two radios, and there are four available channels. The label associated with a node indicates the set of channels assigned to that node, and the label associated with a link indicates the set of channels shared by the two end nodes of that link. Links that are not labeled are not visited yet.

Figure 1(a) gives a partial assignment by choosing the least used channel for the link visited. The next link to visit is (A, D). As shown in Figure 1(b), following a pure interference-aware approach, link (A, D) should be assigned one of the least used channel 1. However, this assignment may lead to potential network partitions; i.e., the network can be partitioned into two components $\{A, B, C\}$ and $\{D, E, F\}$ if channel 1 is reclaimed by a primary user.

In CRTCA, to avoid possible network partitions, we assign another least used channel 2 to link (A, D) so that network partitions will not occur when any single channel is reclaimed (as shown in Figure 1(c)). Channel 2 is referred to as the *backup channel* of link (A, D) which already has channel 1.

Assigning a backup channel may lead to channel adjustment of previously assigned links. Comparing Figure 1(c) with Figure 1(a), we can see that assigning a backup channel to link (A, D) causes link (D, F) to be adjusted from channel 3 to channel 2 since D can only have two radios. Since it has used channel 1 and channel 2, it cannot use channel 3. After adjusting channels of the previously assigned links, we have to check again whether any new partition will occur. If so, backup channels may be assigned, which may result in more links to be adjusted and rechecked.

This naive approach can be further enhanced. Comparing Figure 1(c) with Figure 1(b), assigning a backup channel to link (A, D) increases the channel interference since link (D, F) is now tuned to channel 2 which is used by several nodes. In the enhanced approach, as shown in Figure 1(d), link (A, D) will operate on another least used channel 3. However, if some channel is reclaimed, the network in Figure 1(d) may be less connected than that in Figure 1(c). For example, in Figure 1(c), no matter which channel is reclaimed, the packets from A will traverse at most three hops to E. In Figure 1(d), if channel 3 is reclaimed, the packets may have to traverse four hops from A to E. We will further evaluate the performance of these two approaches through extensive simulations.

B. Algorithm Description

Definition 2 (Potential Interference Index) The potential interference index of edge $e \in E$, denoted by $p(e)$, is the cardinality of the set of edges that can potentially interfere with e, i.e., $p(e) = |\{(u, v)|(u, v) \in E, u \text{ or } v \in D(e)\}|$.

Let $Q(v)$ denote the number of radios at node v ($Q(v) \leq C$). The formal description of the algorithm is shown in Algorithm 1. We first sort all edges based on the potential interference index. Then we go through each edge $e = (u, v)$ in the sorted order and assign some channel c to e based on the channel selection rule that will be presented later.

Suppose channel c is assigned to link $e = (u, v)$. The *robustness test* $t(e, c)$ is performed to check whether u and v are connected when channel c is reclaimed. The test is

Algorithm 1 Centralized Robust Topology Control Algorithm

Compute $p(e)$ for each edge $e \in E$
Sort E in the descending order of $p(e)$
for each edge e in the sorted order of E **do**
  Call the *channel selection rule* for e and add e to queue LQ
  while $LQ \neq \Phi$ **do**
    $\tilde{e} \leftarrow LQ.pop()$
    if $t(\tilde{e}, \mathcal{A}(\tilde{e}))$ fails **then**
      Call the *channel selection rule* to assign a backup channel $c \notin \mathcal{A}(\tilde{e})$ for $\tilde{e}$
    end if
  end while
end for
Assign nodes having extra radios with channels that are least used by their neighbors

performed on the network graph. After removing all the links that are only assigned channel c (no backup channel), if u can still reach v through breadth-first search, the test succeeds. If the test fails, a backup channel is added so that the two end nodes share two common channels. This may adjust previously assigned links that require the robustness test to check again whether any new partition occurs. To effectively manage these links, we use a queue LQ. When LQ is empty, the channel assignment moves to another new link.

Channel Selection Rule One key component of Algorithm 1 is the channel selection rule which specifies how the channel is selected for a link e. There are three cases: 1) If the number of assigned channels is less than the number of radios at both node u and node v, the least used channel is selected to minimize the number of links with which e interferes; 2) If the number of channels equals to the number of radios at u but the number of channels is less than the number of radios at v, we have to select the least used channel from the set of channels assigned to u; and 3) both u and v have assigned channels to all their radios, so we have to select the least used channel in $\mathcal{A}(u) \bigcap \mathcal{A}(v)$. If $\mathcal{A}(u) \bigcap \mathcal{A}(v) = \Phi$, we need to adjust the channel assignment for previously assigned links such that e can be assigned a channel (the detail is shown in Procedure 1).

The channel selection rule can be enhanced by selecting the channel from the set of channels that do not partition the network (as shown in Figure 1(d)). If the set is non-empty, we choose the least used channel among all channels in the set. Otherwise, the default channel selection rule is used to select the channel that is least used but partitions the network. A backup channel is then added to achieve robustness. The underlying principle of this enhancement is to help minimize the number of times that backup channels have to be used.

With the enhanced channel selection rule, not least used channels may be selected, which can increase the channel interference. However, we find that in most cases the interference added by this enhanced rule is less than that by assigning backup channels. This is because in order to assign backup channels, multiple previous links may be adjusted to the channels that have already been used by other links.

Discussions In this paper, we focus on maintaining network connectivity when any single channel is reclaimed by the primary user. Reclaiming multiple channels may disconnect the network, but the chance of such event is very small. If some channels are often reclaimed together, these highly correlated channels can be found as shown in [22]. Then, these correlated channels can be treated as a single channel in the robustness test to deal with multiple channel reclaims. If the primary user only affects a limited region, the channels available at one node may be different from the channels available at another node. Despite this difference, the channels available at neighboring nodes are similar [22]. Our algorithm can be easily extended to this scenario. To perform channel assignment for link $e = (u, v)$, the channel will be selected from the set of channels that are available at both u and v.

Procedure 1 Channel Selection Rule for (u, v)

if $|\mathcal{A}(u)| < Q(u)$ and $|\mathcal{A}(v)| < Q(v)$ **then**
  $c \leftarrow$ the least used channel among channels in C
  /*In the *Enhanced Channel Selection rule*, c is selected from the channels that do not partition the network if such channels exist*/
  $\mathcal{A}(x) \leftarrow \mathcal{A}(x) \bigcup c$ for $x = u, v$
else if $|\mathcal{A}(u)| < Q(u)$ or $|\mathcal{A}(v)| < Q(v)$ **then**
  $c \leftarrow$ the least used channel among channels in $\mathcal{A}(u)$ {assume $|\mathcal{A}(u)| = Q(u)$}
  $\mathcal{A}(v) \leftarrow \mathcal{A}(v) \bigcup c$
else if $\mathcal{A}(u) \bigcap \mathcal{A}(v) \neq \Phi$ **then**
  $c \leftarrow$ the least used channel among channels in $\mathcal{A}(u) \bigcap \mathcal{A}(v)$
else
  $c \leftarrow$ the least used channel among channels in $\mathcal{A}(u) \bigcup \mathcal{A}(v)$
  $c' \leftarrow$ the most used channel in $\mathcal{A}(v)$ {assume $c \in \mathcal{A}(u)$}
  Adjust c' to c in $\mathcal{A}(v)$
  for each previously assigned link $e' = (v, w)$ of which $c' \in \mathcal{A}(e')$ **do**
    Insert e' into LQ and adjust c' to c in $\mathcal{A}(e')$
    Adjust c' to c in $\mathcal{A}(w)$ if $\mathcal{A}(v) \bigcap \mathcal{A}(w) = \Phi$ {This may cause channel adjustment for links connecting to w, and the adjustment is a recursive process}
  end for
end if

C. Algorithm Analysis

Theorem 3 CRTCA terminates in finite steps and achieves robustness upon termination.

Proof: To prove the first part of the theorem, we show that the while-loop in CRTCA terminates, i.e., the queue LQ that includes the links to go through the robustness test can be eventually cleared. If the link $e = (u, v)$ removed from LQ fails the robustness test, a backup channel is assigned. This may lead to channel adjustment of previously assigned links, which are inserted into LQ again to check whether any new partition occurs. After assigning the backup channel, u and v can be connected by the backup channel if the former channel assigned to e is reclaimed. Since e passes the robustness test in the future, it will not push other links into LQ again.

Since there are finite number of links in the network graph and each link can only push other links into LQ once, LQ will be eventually cleared. Thus, the channel assignment succeeds in finite steps.

Next we show $G - E_c$ ($E_c = \{e | e \in E, c$ is the only channel e can operate on$\}$) is connected for any available channel c.

Suppose G is partitioned when channel c is reclaimed, i.e., $G - E_c$ has at least two components, C_1 and C_2. Let the set of links connecting C_1 and C_2 be E_{C_1,C_2}. In E_{C_1,C_2}, all links operate on channel c. Let e be the link that is most recently checked for robustness in E_{C_1,C_2}. Since the two end nodes of e are disconnected by removing E_{C_1,C_2}, e would fail the robustness test. According to CRTCA, this leads to assigning another backup channel to e. This connects C_1 and C_2 in $G - E_c$, which contradicts with the assumption we made earlier. ■

Theorem 4 The time complexity of CRTCA is $O(m^3)$ where $n = |V|$ and $m = |E|$ ($m > n$). The number of channels and the number of radios are assumed to be constants.

Proof: Line 1 takes $O(m^2)$ time since $O(m)$ time is needed to compute $p(e)$ for each edge e. Line 2 takes $O(m \log m)$ time to sort edges in the descending order of $p(e)$. The for-loop iterates m times in total for each edge in G.

Now we compute the time complexity of the channel selection rule. Since there are at most m links in the interference range, finding the least used channel takes $O(m)$ time. Then $O(n)$ nodes might be involved in the channel adjustment for previously assigned links. Thus, it takes $O(m)$ time for each link in the channel selection rule. In the enhanced rule, we have to perform extra robustness tests for each channel to check which channel does not partition the network. Since the underlying breadth-first search takes $O(m + n)$ time and the number of channels is assumed to be a constant, the overall time complexity is still $O(m)$.

For the while-loop, the link removed from LQ goes through the robustness test which takes $O(m + n)$ time. If the link fails the test, a backup channel is assigned which makes the link pass the robustness test in the future. On the other hand, previously assigned links may be adjusted and are inserted into LQ. Since each link can fail the robustness test at most once, a link may be inserted into LQ $O(m)$ times. To account for m links in the network, all while-loops take $O(m^3)$ time.

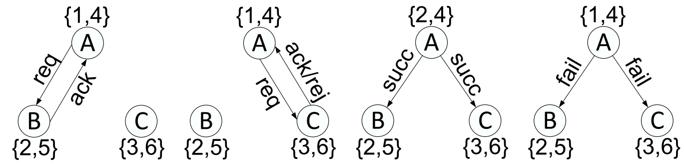
In Line 12, each node attempts to assign channels to extra radios by considering its $O(n)$ neighbors. Thus, Line 12 takes $O(n^2)$ time for all nodes. To summarize, the time complexity is $O(m^3)$. ■

V. Distributed Robust Topology Control Algorithm

In this section, we present the distributed version of our Robust Topology Control Algorithm (DRTCA). We first present the generic framework for topology control, and then propose a local robustness test to reduce the message overhead. Finally, we present the theoretical analysis of DRTCA.

A. Topology Control in a Distributed Environment

1) Neighbor Discovery: Each node scans all channels to detect beacons that contain the IDs and the current radio settings of its neighbors. The ID is a number that is unique to each node. The neighbor information is stored locally, based on which it knows how to reach a neighbor through which channel. After discovering all neighbors, the node switches to some least used channel and notifies its neighbors about its channel setting. It also broadcasts beacons so that it can be detected by other newly joined nodes.



(a) request B (b) request C (c) success (d) failure

Fig. 2. A distributed scenario

2) Distributed Channel Assignment: After neighbor discovery, nodes assign channels. As shown in Figure 2, A, B and C are within each other's interference range. The labels associated with a node indicate which channels the two radios of the node are currently tuned to.

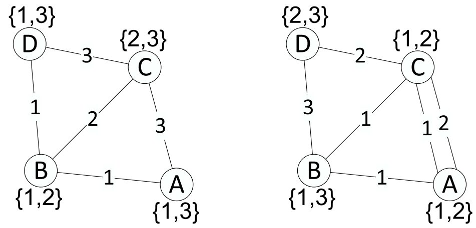
The node with a higher ID is responsible for channel assignment of a link. Suppose A has higher ID than B, then A performs channel assignment for link (A, B). Based on its collected neighbor channel information, the least used channel 2 is selected and used by one of A's radios. To ensure that no other node within its interference range switches to channel 2 at the same time, A has to verify with all these neighbors. Since A's channels are now 1 and 4 which are different from B's channels and C's channels, A needs to switch one radio to one of B's (C's) channels to communicate with B (C). For example, A sends the request message through channel 2 (3), so that B (C) can receive the request. For B and C, after receiving the request, they acknowledge the requester only if they do not want to switch to channel 2. Otherwise, the request is rejected. After A receives acknowledgments from all its neighbors, A switches one radio to channel 2 and sends out the success message.

If A has another neighbor D who only shares channel 1 with A, the above channel adjustment from 1 to 2 will make D and A not share any common channel. Thus, A has to request D to switch the channel to 2. If D has links connecting to its neighbors on channel 1, D has to initiate channel adjustments for nodes within its interference range. D acknowledges A only after D collects all the required acknowledgments. Note that the number of channel adjustments is not large. Since the channel assignment algorithm always selects the least used channel, few links in the interference range are on the same channel (channel 1), leading to few channel adjustments.

If C replies A with a rejection, A sends out a failure message, and all nodes within the interference range (such as B) discard the request.

After this process, each node has to update its neighbor information to reflect the up-to-date radio settings of all nodes within the interference range. Eventually, each node shares at least one channel with any neighboring node for direct communication.

3) Node-based Channel Assignment Rule: In CRTCA, channel assignment is performed for each link e in the descending order of the potential interference index $p(e)$, which



(a) (b)

Fig. 3. Illustration of the local robustness test

gives higher priority to the link that is likely to interfere with more links. Such link-based channel assignment is difficult to be performed in a distributed environment. This is because a node only knows nodes in its interference range, but we need to count all links in the interference range of the two end nodes to obtain $p(e)$.

Alternatively, since the node with larger degree (the number of neighboring nodes) is likely to interfere with more nodes, it is given higher priority to assign channels. The node-based channel assignment rule is as follows.

- A node can only generate its own request when all other nodes with larger node degrees within the interference range have finished their channel assignment.

With this rule, the number of wasted requests is reduced since a smaller number of nodes sharing the same node degree contend with each other. Compared with the potential interference index, the node degree does not precisely measure the interference among data transmissions in different links. As a result, applying this node-based channel assignment may not achieve the same performance as the link-based channel assignment. However, in a distributed environment, it is impossible to implement the link-based channel assignment, and thus we use node-based channel assignment.

B. Local Robustness Test

In CRTCA, the robustness test for link $e = (u, v)$ ($\mathcal{A}(e) = c$) can be performed through breadth-first search. In a distributed environment where nodes do not have the entire network information, breadth-first search has to be performed by sending probe messages to detect whether u is reachable from v without traversing links on c. It involves huge message overhead since the robustness test is needed for each link. To address this issue, we propose a *local robustness test*, where the robustness test is performed locally using only neighbor information without sending probe messages.

Suppose the robustness of link (A, B) is tested at node A in Figure 3(a). Based on its collected neighbor channel information, A checks whether it has a link with one of its neighbors on some channel different from channel 1 and such neighbor also has a link with B on some channel other than channel 1. A finds out that links (A, C) and (B, C) satisfy the requirement, and then (A, B) passes the local robustness test. Moreover, based on its collected neighbor channel information, A can re-route the packet for B through C if channel 1 is reclaimed by a primary user.

In general, since the search space of the local robustness test is limited, unnecessary backup channels may be assigned and thus increase the channel interference. In Figure 3(b), (A, B) fails the local robustness test since B is unreachable from A through A's neighbors without using channel 1. Thus, a backup channel is assigned to link (A, B) so that A and B share two common channels. However, the backup channel is unnecessary since B is actually reachable from A through the path $P' = (AC, CD, DB)$.

Passing the local robustness test guarantees that each flow can be re-routed locally when primary users appear. This prevents the flow from deviating too much from the original path and thus reduces the chance of interfering with other flows due to re-routing.

C. Algorithm Analysis

Theorem 5 DRTCA terminates in finite steps and achieves robustness upon termination.

Proof: DRTCA mainly differs from CRTCA in the channel assignment order for the links. Since the correctness proof of CRTCA does not depend on the link assignment order, the same procedure can be applied to demonstrate the correctness of DRTCA. ■

Theorem 6 The message complexity of DRTCA is $O(mn^2)$, where $n = |V|$ and $m = |E|$.

Proof: In this proof, we assume requests can be generated in order so that there is no contention between any two requests.

The number of messages can be computed based on the number of requests generated. In DRTCA, a node performs channel assignments for the links between itself and its neighbors. It may also adjust its channels for $O(m)$ times due to the channel assignments of all other links. Since each channel assignment leads to one request, there are $O(m)$ requests per node.

The number of messages per request is computed as follows. Let $d(v)$ be the node degree of v whose request results in acknowledgments or rejections from neighbors. A success or failure message is sent in the end, leading to $3d(v)$ messages in total for each request.

To summarize, the message complexity of DRTCA is $O(mnD)$ where $D = \max(d(v))$ for any node $v \in V$. Since $D = O(n)$, we can rewrite the complexity as $O(mn^2)$. ■

VI. Performance Evaluations

In this section, we evaluate the performance of our solutions by comparing them to existing approaches through extensive simulations.

A. Simulation Setup

In the simulation, 25 nodes are randomly placed in a $900 \times 900\ m^2$ area. Each node has the same number of radios (Q), which can be used to access C licensed channels. The transmission range and the interference range for each node are set to 250 m and 500 m, respectively [4].

We compare the performance of several channel assignment approaches. For our approach, there are four versions:

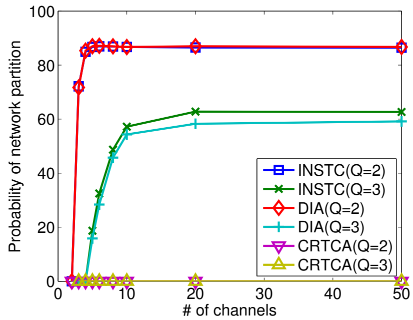


Fig. 4. Probability of network partition vs. # of channels

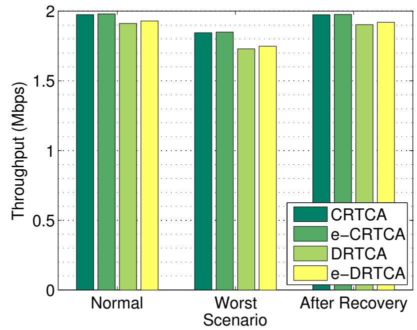


Fig. 5. The throughput of our approaches ($C = 20$ channels, $Q = 2$ radios, $F = 2$ flows, $B = 2$ Mbps)

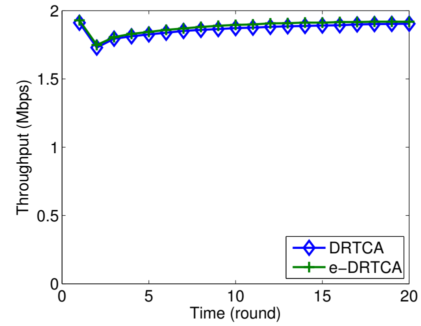


Fig. 6. The throughput of the distributed approaches ($C = 20$, $Q = 2$, $F = 2$, $B = 2$)

e-CRTCA, CRTCA, e-DRTCA and DRTCA. CRTCA and DRTCA use the default channel selection rule, while e-CRTCA and e-DRTCA use the enhanced channel selection rule. For comparison, we implement two interference-aware approaches, INSTC [14] and DIA. INSTC is a centralized interference-aware approach where the channels are assigned in the descending order of the potential interference index. DIA is a distributed interference-aware approach obtained by disabling the robustness test in DRTCA, so that the assigned channel leads to minimum interference. Both approaches can be slightly modified to satisfy the robustness constraint, i.e., all nodes reserve one channel as a backup channel and then find the interference-minimum assignment using the remaining channels. The robust versions of INSTC and DIA are denoted by INSTC-backup and DIA-backup, respectively.

B. Simulation Results on Robustness

To measure robustness, we show the *probability of network partition*; i.e., the probability that the channel assignment is not robust (the network is partitioned when a channel is reclaimed by the primary user). Figure 4 compares CRTCA with other two approaches INSTC and DIA by averaging over $10,000$ randomly generated topologies, as the number of channels and the number of radios ($Q = 2$ or $Q = 3$) change. As shown in the figure, the probability of network partition in CRTCA is 0 since our algorithm is designed to satisfy the robustness constraint.

For INSTC and DIA, when each node has two radios ($Q = 2$), there is no network partition if the number of channels is two. This is because each node tunes its radios to the same two channels, and thus each pair of neighboring nodes can still communicate when either of the two channels is reclaimed by the primary user. If the number of channels is more than two, the probability of network partition increases sharply, but stays flat as the number of channels is more than 20. This is because both INSTC and DIA are unaware of the robustness constraint. Each node can only use two channels at one time and it has to match the channels used by its neighbors. Then, many channels are not used, and providing more available channels does not help improve the robustness.

Using three radios ($Q = 3$) can reduce the probability of network partition compared to using two radios ($Q = 2$). With more radios, each node can use more channels at the same time. This reduces the probability of network partition when a primary user appears. However, the probability of network partition is still very high compared to CRTCA which has no network partition.

From this experiment, we can see that having more available channels does not increase the robustness. Although adding more radios can increase the robustness, it has high cost due to the hardware cost of extra radios.

Note that other robust approaches such as DRTCA, INSTC-backup, and DIA-backup, also do not have network partition if only one channel is reclaimed by the primary user. We omit their plots in Figure 4 due to space limit.

C. Simulation Results on Network Performance

To measure the network performance, we inject several constant bit rate flows (F) into the network. The source and the destination are picked randomly. Each flow follows the shortest path from the source to the destination. If a node has multiple radios connected to the next node in the path, it dynamically forwards the packet using the channel that is least interfered by other transmissions.

To measure the network performance, we measure the *normal throughput* and the *worst throughput*. The normal throughput denotes the throughput when all channels are available; i.e., no primary user appearance. The worst throughput is the least among different cases when a primary user appears at different channels. In the following experiments, each flow is generated at rate 1Mbps. The bandwidth (the maximum transmission rate per channel) is BMbps.

1) Evaluation of Our Approaches: Figure 5 compares four versions of our approaches, CRTCA, e-CRTCA, DRTCA and e-DRTCA, in terms of the normal throughput, the worst throughput and the throughput after recovery from the worst scenario. The worst scenario corresponds to the scenario when the worst throughput is achieved. Suppose the channel that is reclaimed by the primary user in the worst scenario is c. For recovery, centralized approaches can simply re-run the channel assignment process with reduced number of available channels except c. In distributed approaches, only the link operating on channel c is assigned another available channel following the channel selection rule. The assignment should also satisfy the robustness constraint.

By applying our approaches, the worst throughput is similar to the normal throughput; after recovery, it almost reaches the normal throughput. The centralized versions slightly outperform distributed versions due to the difference in the order of channel assignment and the restriction on robustness test which affects the channel interference as discussed in Section V-B. Each enhanced version, either centralized or distributed, slightly outperforms the counterpart.

Figure 6 compares DRTCA and e-DRTCA during the recovery process from which we can see that both approaches recover from the worst throughput quickly and e-DRTCA slightly outperforms DRTCA.

2) Overall Comparisons with Existing Approaches: Figures 7, 8, 9 compare the normal throughput and the worst throughput of three centralized approaches INSTC, INSTC-backup and e-CRTCA, whereas Figures 10, 11, 12 compare three distributed approaches DIA, DIA-backup and e-DRTCA.

Generally speaking, INSTC (DIA) has similar normal throughput with e-CRTCA (e-DRTCA). However, the worst throughput of INSTC and DIA is much less than their normal throughput, whereas the worst throughput of e-CRTCA and e-DRTCA is similar to their normal throughput. For their robust versions, the normal throughput of INSTC-backup (DIA-backup) is less than that of INSTC (DIA). Although INSTC (DIA) and INSTC-backup (DIA-backup) both aim at minimizing the channel interference, INSTC-backup (DIA-backup) has one more restriction that all nodes have to tune one radio to the same backup channel. As a result, INSTC-backup (DIA-backup) has more channel interference and thus lower normal throughput. On the other hand, since the robustness constraint is satisfied, INSTC-backup (DIA-backup) has higher worst throughput than that of INSTC (DIA). Note that they still underperform e-CRTCA (e-DRTCA).

3) Effect of the Number of Channels: Figures 7 and 10 show the effects of the number of channels on throughput. For all approaches except INSTC-backup (DIA-backup), since the normal throughput already reaches the maximum 3MBps when there are 5 channels, increasing the number of channels does not change the normal throughput.

As for the worst throughput, in INSTC and DIA, increasing the number of channels to more than 5 reduces the worst throughput. This is because increasing the number of channels also increases the probability of network partition as shown in Figure 4, which leads to lower throughput.

In e-CRTCA (e-DRTCA), the worst throughput is only a little bit less than the normal throughput, but much higher than that of INSTC (DIA) and higher than that of INSTC-bakcup (DIA-backup). Also, the worst throughput of e-CRTCA and e-DRTCA increases slightly as the number of channels increases, since the channel interference is reduced. Further increasing the number of channels beyond 5 does not change the worst throughput, since the worst throughput already reaches the maximum 3MBps.

In INSTC-backup and DIA-backup, since there are only three radios at each node and one radio is used for backup, only two radios are left for channel assignment, which further limits the number of channels that can be used in the network (as mentioned in Section VI-B). Thus, providing more available channels does not help improve the normal throughput or the worst throughput.

4) Effect of Bandwidth: Figures 8 and 11 show the effects of bandwidth (the maximum transmission rate per channel) on throughput. For all approaches, the normal throughput reaches the maximum 2Mbps with bandwidth increase.

As for the worst throughput, in INSTC and DIA, increasing the bandwidth does not change the worst throughput too much. This is because increasing bandwidth does not affect the probability of network partition, which affects the worst throughput.

In e-CRTCA (e-DRTCA), the worst throughput is higher than that of INSTC (DIA) but is less than the normal throughput if the bandwidth is less than 2Mbps. This is because the bandwidth limitation further reduces the worst throughput when data transmission can only use channels that are not reclaimed by the primary user. When bandwidth is reduced, it may take a long time for a packet to be transmitted from source to destination due to collisions with other packets. If such transmission latency is comparable or larger than that of spectrum handoff, the spectrum handoff problem appears relatively less serious; our approach will not have too much advantage over existing approaches. When the bandwidth is more than 2Mbps, the worst throughput of e-CRTCA (e-DRTCA) is similar to the normal throughput since bandwidth is not an issue and the robustness constraint is satisfied.

The worst throughput of INSTC-backup (DIA-backup) has the same trend as e-CRTCA (e-DRTCA) with bandwidth increase. Since INSTC-backup and DIA-backup use fewer channels (as mentioned in Section VI-C3), they underperform our approaches.

5) Effect of the Number of Flows: Figures 9 and 12 show the effects of the number of flows on performance. For all approaches, increasing the number of flows increases both normal throughput and worst throughput.

In INSTC and DIA, the difference between worst throughput and normal throughput enlarges with the increase of the number of flows. This is because INSTC and DIA are unaware of the robustness constraint. When primary users appear, more throughput will be lost if more flows are injected.

However, in our approaches (e-CRTCA and e-DRTCA), the worst throughput is similar to the normal throughput when the number of flows is less than four. When the number of flows is more than four, the worst throughput is a little bit less than the normal throughput. Since a channel is reclaimed by the primary user, fewer channels can be used for data transmission. Increasing the number of flows increases the chances that many flows use the same channel and interfere with each other, leading to much lower throughput.

For INSTC-backup and DIA-backup, all nodes have three radios ($Q = 3$) but have to tune one radio to the same backup channel. Then, only two radios are left for channel assignment. The final channel interference is much larger than that of other channel assignment approaches which are based on three

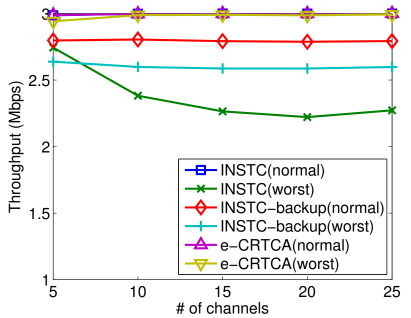


Fig. 7. # of channels vs. throughput ($Q = 3$, $F = 3$, $B = 2.5$)

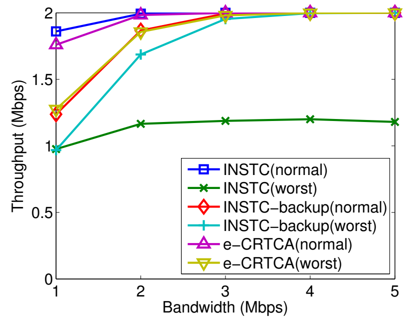


Fig. 8. bandwidth vs. throughput ($C = 20$, $Q = 2$, $F = 2$)

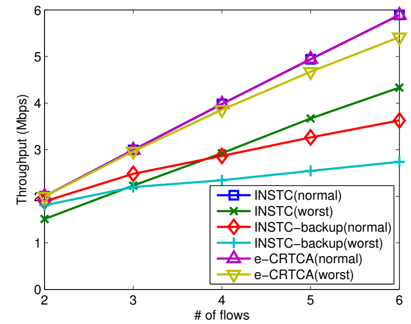


Fig. 9. # of flows vs. throughput ($C = 20$, $Q = 3$, $B = 2$)

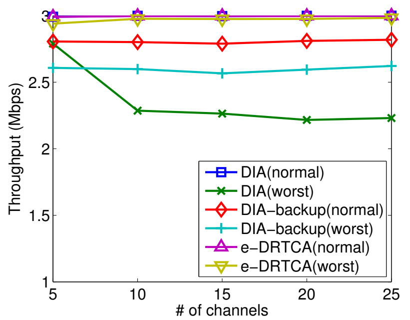


Fig. 10. # of channels vs. throughput ($Q = 3$, $F = 3$, $B = 2.5$)

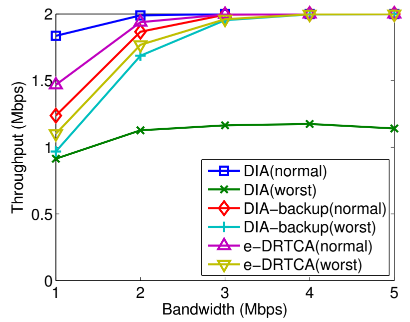


Fig. 11. bandwidth vs. throughput ($C = 20$, $Q = 2$, $F = 2$)

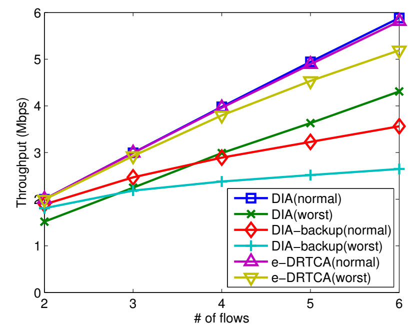


Fig. 12. # of flows vs. throughput ($C = 20$, $Q = 3$, $B = 2$)

radios. This together with the interflow interference result in very poor performance in both normal throughput and worst throughput.

VII. Conclusions

This paper studied the channel assignment problem in multi-hop multi-radio cognitive networks. The problem was formally defined as a robust topology control problem which is NP-hard. To solve the problem, we proposed centralized and distributed algorithms which jointly consider network robustness and channel interference. Extensive simulations demonstrate that our solutions outperform existing interference-aware approaches when primary users appear, and achieve similar performance at other times.

References

[1] FCC, "FCC Adopts Rules for Unlicensed Use of Television White Spaces," November 2008.

[2] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810 – 836, 2009.

[3] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with wi-fi like connectivity," in *ACM SIGCOMM, 2009*.

[4] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *IEEE INFOCOM, 2005*.

[5] P. Marshall, "Extending the reach of cognitive radio," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 612 –625, 2009.

[6] E. Rozner, Y. Mehta, A. Akella, and L. Qiu, "Traffic-Aware Channel Assignment in Enterprise Wireless LANs," in *IEEE ICNP, 2007*.

[7] R. Vedantham, S. Kakamanu, S. Lakshmanan, and R. Sivakumar, "Component Based Channel Assignment in Single Radio, Multi-channel Ad Hoc Networks," in *ACM MobiCom, 2006*.

[8] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein, "Distributed Channel Assignment in Multi-Radio 802.11 Mesh Networks," in *IEEE Wireless Communications & Networking Conference (WCNC), 2007*.

[9] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," in *IEEE DySPAN, 2005*.

[10] V. Brik, E. Rozner, S. Banerjee, and P. Bahl, "DSAP: a protocol for coordinated spectrum access," in *IEEE DySPAN, 2005*.

[11] M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "DIMSUMnet: new directions in wireless networking using coordinated dynamic spectrum," in *IEEE Symposium on World of Wireless Mobile and Multimedia Networks (WoWMoM), 2005*.

[12] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating dynamic time-spectrum blocks in cognitive radio networks," in *ACM MobiHoc, 2007*.

[13] L. Cao and H. Zheng, "Distributed spectrum allocation via local bargaining," in *IEEE SECON, 2005*.

[14] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," in *ACM MobiHoc, 2005*.

[15] A. Subramanian, H. Gupta, and S. Das, "Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks," in *IEEE SECON, 2007*.

[16] L. Yang, L. Cao, and H. Zheng, "Proactive channel access in dynamic spectrum network," in *Proceedings of the Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), 2007*.

[17] H. Kim and K. G. Shin, "Fast discovery of spectrum opportunities in cognitive radio networks," in *IEEE DySPAN, 2008*.

[18] W. Feng, J. Cao, C. Zhang, and C. Liu, "Joint optimization of spectrum handoff scheduling and routing in multi-hop multi-radio cognitive networks," in *IEEE ICDCS, 2009*.

[19] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388 –404, Mar. 2000.

[20] I. Holyer, "The NP-Completeness of Edge-Colouring," *SIAM Journal on Computing*, vol. 10, no. 4, pp. 718 – 720, 1981.

[21] IEEE 802.11 Working Group, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1997.

[22] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: a large-scale spectrum measurement study," in *ACM MobiCom, 2009*.