On Routing and Channel Selection in Cognitive Radio Mesh Networks

Brendan Mumey, *Member, IEEE*, Jian Tang, *Member, IEEE*, Ivan R. Judson, and David Stevens

***Abstract*—Secondary users in a cognitive radio mesh network may select from a set of available channels, provided that they do not disrupt communications among primary users. This ability can improve the overall network performance but introduces the question of how to best use the channels. This paper first considers the problem of selecting the channels to use given a routing path such that the end-to-end throughput along the path is maximized. We show that a dynamic programming-based approach can optimally solve the problem and, if the path satisfies a natural condition, in time, be linear in the length (hop count) of the path. In addition, the algorithm can easily be implemented in a distributed fashion. We also examine the harder joint problem of finding the best routing path and channel selection that maximizes the end-to-end throughput. We prove that obtaining a $(2/3+\epsilon)$ approximation to the joint problem is NP-hard. We then present a heuristic algorithm for the joint problem and a second heuristic channel-aware routing-only algorithm. Numerical results are provided to demonstrate the effectiveness of the methods on several experimental scenarios.**

***Index Terms*—Channel selection (CS), cognitive radios, interference, routing, wireless mesh networks (WMNs).**

I. Introduction

WIRELESS mesh networks (WMNs) are considered to be an economical method of providing robust high-speed backbone infrastructure and broadband Internet access in large areas [1]. Mesh topology offers the advantages of alternative route selection to ensure throughput and quality-of-service (QoS) requirements under dynamic load conditions. Because the aggregate traffic volume can be substantial on backbone links converging on gateways and mesh routers, considerations of transmission path routing and how to select channels along the path are essential to ensure that a WMN can meet the QoS and throughput requirements of end-users' applications, particularly real-time multimedia applications. Furthermore, range considerations and propagation characteristics demand careful attention to interference. Cognitive radios are desirable for a WMN in which a large volume of traffic is expected to be delivered, because they can more efficiently utilize the available spectrum than conventional static channel assignment methods and, therefore, significantly improve network capacity [2]. However, they introduce additional complexities to resource allocation. With cognitive radios, each node can access a set of available spectrum bands that may span a wide range of frequencies. Each spectrum band may be divided into channels, and the channel bandwidths may vary from band to band. Different channels may support quite different transmission ranges and data rates, both of which have a significant impact on resource allocation and interference effects. Each network link has some subset of channels available due to the activities of primary users and other traffic in the network.

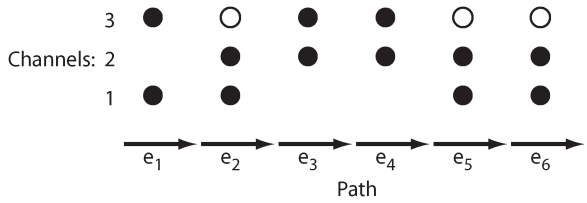

Fig. 1. Example CS problem. In this example, we assume that the channel availability is as shown and that each channel has a capacity of 1. We further assume that two link pairs (e_i, j) and (e_k, j) interfere if and only if $|i-k| \leq 2$. The solid circles indicate an optimal CS for the path. Based on the interference and duplexing constraints, all but two of the selected link pair participate in cliques of size 3 (and, therefore, provide a capacity of 1/3), and $(e_1, 1)$ and $(e_6, 1)$ have a max clique size of 2 (and, therefore, provide a capacity of 1/2). Thus, the end-to-end capacity of the path is 2/3.

In this paper, we study the following two hard resource allocation problems in cognitive radio mesh networks: 1) the *channel selection (CS)* problem, which is to choose a set of available channels on each link in a given routing path to maximize the end-to-end throughput of the path, and 2) the *joint routing and channel selection (JRCS)* problem, where the routing path is not provided as part of the input and must be found along with a CS for each link on the path (see Fig. 1 for an example of the CS problem). We use a general and accurate approach to estimate the end-to-end throughput of a path, based solely on how the selected channels interfere with each other that is independent of any specific scheduling approach and in which channel capacities and availabilities are link specific. Our objective is to design efficient approaches to support emerging wireless applications that demand long-standing connections and high end-to-end throughput, such as real-time streaming video or bulk data transfer.

This paper is different from some previous works on scheduling and spectrum allocation (e.g., [23]) that mainly dealt with the problem of scheduling and allocating channels to links to

Manuscript received November 22, 2011; revised March 19, 2012 and June 12, 2011; accepted August 3, 2012. Date of publication August 15, 2012; date of current version November 6, 2012. The work of J. Tang was supported by the National Science Foundation under grant CNS-1113398. The review of this paper was coordinated by Dr. T. Taleb.

B. Mumey and I. R. Judson are with the Department of Computer Science, Montana State University, Bozeman, MT 59717 USA (e-mail: mumey@cs.montana.edu).

J. Tang is with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: jtang02@syr.edu).

D. Stevens was with Montana State University, Bozeman, MT 59717 USA. He is now with the University of Oregon, Eugene, OR 97403 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2012.2213310

maximize the link-layer throughput. Here, we focus on the end-to-end performance and consider the problem of allocating channels along a multihop routing path, which is a much harder problem due to the constraints related to intraflow interference [28] (links on a common path interfere with each other if assigned the same channels) and because, in general, there are an exponential number of potential paths in a mesh network that connect a pair of nodes and an exponential number of ways of assigning channels along a path. In addition, the algorithms that were proposed for traditional WMNs with homogeneous channels [3], [14], [22] cannot be applied to solve our problems here, which target at a large number of heterogeneous channels that can support different data rates and transmission ranges. In short, routing and CS in cognitive radio mesh networks are very challenging problems. Most existing works [6], [10], [13], [15], [18]–[21] on this topic presented heuristic algorithms that cannot provide any performance guarantees. In this paper, we study the CS problem and the JRCS problem from a theoretical perspective and aim at developing theoretically well-founded and practically useful algorithms to solve them. Our major contributions are summarized as follows.

1) We present a new characterization of optimal CS solutions, which leads to an efficient dynamic programming algorithm that can optimally solve the CS problem and, if the path satisfies certain natural *self-avoiding* criteria (which are defined in Section III), is in time linear in the length (hop count) of the path. In addition, the algorithm can easily be implemented in a distributed fashion, and an optimal solution can be computed in a single pass by the nodes along the path with only local information sharing.
2) We also examine the much harder problem of JRCS, for which the routing path is not provided and must jointly be found along the selection of channels to use along the path. The objective is, again, to maximize the end-to-end path throughput.We show that obtaining a $(2/3 + \epsilon)$ approximation to the JRCS problem is NP-hard for any $\epsilon > 0$, which places the JRCS problem in the complexity class of APX-hard. Despite the theoretical hardness, we present a heuristic algorithm to solve the joint problem and a channel-aware routing-only algorithm that can be combined with the aforementioned optimal CS algorithm.
3) Extensive simulation results show that the proposed joint algorithm outperforms several other methods that first compute a routing path and then determine the CS for that path.

The rest of this paper is organized as follows. We discuss related work in Section II. We formally define the problems to be studied in Section III and examine the computational complexity in Section VI. Then, we present an optimal algorithm for the CS problem in Section IV and some heuristic algorithms for the JRCS problem in Section V. We present the numerical results in Section VII and conclude this paper in Section VIII.

II. Related Work

Cognitive radio networks (CRNs) have recently received extensive attention. Spectrum allocation and access are the most important problems in such networks. There is a large body of work on the link-level optimization of CRNs through CS and scheduling. In [29], the concept of a time-spectrum block was introduced to model spectrum reservation, and a centralized and a distributed protocol were presented to allocate such blocks for cognitive radio users. Tang *et al.* introduced a graph model to characterize the impact of interference and proposed joint scheduling and spectrum allocation algorithms for fair spectrum sharing based on it in [23]. In [15], a linear-programming-based approach is presented for link scheduling and channel assignment.

Routing and CS have been studied for CRNs. In [27], a novel layered graph was proposed to model spectrum access opportunities and was used to develop joint spectrum allocation and routing algorithms. In [26], the authors presented distributed algorithms for joint spectrum allocation, power control, routing, and congestion control. A mixed-integer nonlinear-programming-based algorithm was presented to solve a joint spectrum allocation, scheduling, and routing problem in [11]. A distributed algorithm was presented in [21] to solve a joint power control, scheduling, and routing problem, with the objective of maximizing data rates for a set of communication sessions.

Spectrum-aware mesh routing (SAMER) [18] is a routing protocol that accounts for long- and short-term spectral availability and seeks to utilize available time-spectrum blocks by routing data traffic over paths with higher spectrum availability, without ignoring instantaneous spectral conditions. Spectrum-aware routing (SPEAR), which was presented in [20], aimed at maximizing the throughput by combining end-to-end optimization with the flexibility of link-based approaches to address spectrum heterogeneity. In [10], Hincapie *et al.* proposed a novel distributed routing protocol that can select a route and allocate channels and timeslots for a connection request to satisfy its end-to-end bandwidth requirement. The proposed protocol is based on dynamic source routing and selects time-spectrum blocks for links using a novel metric to obtain high-capacity and low-interference blocks for links during the route discovery procedure. In [16], Mumey *et al.* considered the problem of finding a transmission schedule and a CS solution for a given path and presented a constant factor approximation algorithm based on graph coloring.

The related problem of flow assignment (in which the communication request can simultaneously be routed along multiple paths) has also been well studied for CRNs. In [24], a polynomial-time approximation scheme is presented for a more general maximum multiflow scheduling problem (maximize the total flow of a set of commodities with no specific routing path), and several constant-factor approximations are given for special cases. This paper also points out some errors in previous work on that problem. In [19], a joint problem of channel capacity assignment and flow assignment is considered and modeled as a convex nonlinear optimization problem, and several heuristic methods are proposed. An interference- and congestion-aware routing protocol is proposed in [6] for a hybrid wireless architecture in which some link channels are fixed, and some may be chosen.

In addition, routing and CS have been studied in the context of traditional WMNs with homogeneous channels [3], [14],

[22]. A constant-bound approximation algorithm was proposed in [3] to jointly compute channel assignment, routing, and scheduling solutions for fair rate allocation. In [14], a similar problem is studied, and derived upper bounds on the achievable throughput are derived using a fast primal-dual algorithm. In [22], Tang *et al.* proposed an interference-aware channel assignment algorithm along with an optimal routing scheme for end-to-end bandwidth guarantees. Khalife *et al.* [13] proposed a routing approach based on the probabilistic estimate of channel availability on network links. Recently, Almasaeid *et al.* [4] have developed an optimal dynamic programming algorithm for a channel assignment and routing problem that minimizes end-to-end packet delay but does not consider interference.

The differences between this paper and these related works are summarized as follows. First, in both of our problems, the objective is to maximize the end-to-end throughput, which is different from works that address link-layer (single-hop) throughput, such as [23] and [29]. Second, we obtain an optimal algorithm for the CS problem that runs in time linear in the length of the given routing path, provided that the path satisfies a natural condition. To the best of our knowledge, this is the first polynomial-time algorithm that obtains an optimal solution to this problem. Many related works (such as [6], [10], [11], [13], [15], [18]–[21], [26], and [27]) only presented heuristic algorithms that cannot provide any performance guarantees. Third, our optimization problems are different from those studied in [10], [11], [16], [18], [20], [21], [26], and [27]. Fourth, as aforementioned, the algorithms proposed for traditional WMNs with homogeneous channels [3], [14], [22] cannot be applied to solve our problems here due to channel heterogeneity. Last, this paper is the first to establish a bound on the complexity of the JRCS problem. We show that it is NP-hard to approximate to within a factor $(2/3+\epsilon)$ and provide effective heuristic algorithms to solve it.

III. Problem Definitions

We consider a wireless mesh backbone network $G=(V,E)$ with static mesh routers, where V is the set of nodes, and E is the set of available communications links between the nodes. Each node is equipped with a cognitive radio. Similar to [10], [11], and [27], a spectrum occupancy map is assumed to be available to network nodes from a centrally maintained spectrum database. This scenario has recently been promoted by the Federal Communications Commission to indicate, over time and space, the channel availabilities in the spectrum below 900 MHz and around 3 GHz [7]. In this case, spectrum (channel) availability between any given node pair is known. We study the problem of determining the optimal route and channel assignment for a communication session between two nodes in the network. Spectrum sensing is out of scope of this paper.

We define our assumptions about the parameters of the CRN. Let m be the number of channels available in the network. In general, each link e will have only a subset of these channels available at any given time. This can be due to interference, with the link distance being greater than the transmission range, or because the channel is already in use on that link. We will also assume that each available channel j on link e has an associated bit rate $b_{e,j}\geq 0$. This bit rate can depend on the link distance and other factors such as interference from other ongoing transmissions in the network. We assume that communication in the network is done using synchronized transmission frames. Let A_e be the set of channels available to link e during the current frame. We assume that channel availability can be determined for the wireless region in question, e.g., by consulting a television whitespace database, and A_e will not change during the transmission frame (if channel availability can change at arbitrary times, then some signaling mechanism needs to be implemented in the protocol so that a new CS can be recomputed for the transmission path).

We adopt the following simple interference model. We assume that there is an interference distance R_j for each channel j such that a link $e=(u,v)$ *interferes* with another link $e'=(u',v')$ on channel j if and only if $|u-v'|\leq R_j$ or $|u'-v|\leq R_j$. We will also consider that the nodes in question are half-duplex, which means that nodes cannot simultaneously transmit and receive. The duplexing and interference constraints impose conditions on which link flows can be active at the same time. We will summarize these conditions in a well-known *conflict graph*, i.e., $G_c=(V_c,E_c)$, where the vertices V_c are the link-channel pairs (e,j), and the edges (undirected) indicate link-channel pairs that cannot be simultaneously operational due to interference or duplexing constraints.

Suppose that we have a routing path p from s to t and a set of active channels $J_e\subset A_e$ has been chosen to be used for each link $e\in p$. Let $G_{c,p}^{\langle J_e\rangle}$ be the conflict graph that is restricted to the link-channel pairs of the form (e,j), where $e\in p$, and $j\in J_e$. Let $t_{e,j}$ be the total amount of time allocated to the link-channel pair (e,j) in the transmission frame (assumed to be of length 1). Each clique C in $G_{c,p}^{\langle J_e\rangle}$ imposes the constraint

$$\sum_{(e,j)\in C} t_{e,j}\ \leq 1. \tag{1}$$

Let $m_{e,j}^{\langle J_e\rangle}$ be the size of the largest clique that contains (e,j) in $G_{c,p}^{\langle J_e\rangle}$. We make a simplifying assumption that the scheduling mechanism creates a *uniform schedule* such that

$$t_{e,j}=1/m_{e,j}^{\langle J_e\rangle}. \tag{2}$$

Observe that each constraint of the form (1) is satisfied by (2). We note that a uniform schedule may not be optimal, but we adopt this assumption for algorithmic convenience and because it may not be possible to alter the scheduling mechanism used in a real network.

Let $G_{c,p}^{j}$ be the subgraph of $G_{c,p}$ that consists of the link-channel pairs that use channel j.

Definition 1: We say that the routing path p is *self-avoiding* if all of the subgraphs $G_{c,p}^{j}$ are interval graphs such that the intervals occur in order of p.

This condition means that the link pairs in p that involve channel j can be placed in order on the real number line R such that two link pairs (e,j) and (e',j) conflict if and only if their corresponding intervals overlap. Interval graphs have a useful property that their cliques can easily be enumerated because a

clique will be represented by a set of consecutive intervals that all mutually overlap. In fact, all maximal cliques on an interval graph with n vertices can be enumerated in $O(n)$ time. We will focus on self-avoiding routing paths in this paper, because it is easy to find the maximal cliques in their interference graphs.

We define the *end-to-end throughput* τ of the path p and selected active channels $\langle J_e \rangle_{e \in p}$ as the minimum over all the links $e \in p$ of the effective throughput on link e. The effective throughput of a link e is the sum of the bit rates on each active channel times the amount of transmission time allocated to each channel, i.e.,

$$\tau\left(p, \langle J_e \rangle\right) = \min_{e \in p} \sum_{j \in J_e} b_{e,j} / m_{e,j}^{\langle J_e \rangle}. \tag{3}$$

We note that this objective function assumes that a uniform transmission schedule will be created so that (2) is satisfied and interfering link-channel pairs are not scheduled at the same time.

We can now formalize the two computational problems considered as follows, and in addition to the source node s and destination node t, we assume that the available channel sets A_e and bit rates $b_{e,j}$ are provided in the input.

CS. Given a routing path p from s to t, determine active channels sets $J_e \subset A_e$ for all $e \in p$ that maximize the end-to-end throughput $\tau(p, \langle J_e \rangle)$.

JRCS. Find a routing path p from s to t and active channels sets $J_e \subset A_e$ for all $e \in p$ that maximize the end-to-end throughput $\tau(p, \langle J_e \rangle)$.

IV. Optimal Channel Selection

In this section, we present an optimal algorithm for the problem of choosing the best set of channels to use for a given routing path (CS). Our proposed CS algorithm is based on a dynamic programming approach in which the solutions to partial problems are used to assemble a solution to the full problem. To help motivate the algorithm, we provide a simple example of CS in Fig. 1.

Let $p = (v_0, v_1, \ldots, v_n)$ be the given routing path from the source node s to the destination node t, where $v_0 = s$, and $v_n = t$. Let $e_i = (v_{i-1}, v_i)$ be the ith link on the path $(1 \le i \le n)$ and let A_i be the set of available channels on e_i. The objective is to find active channel sets $J_i \subset A_i$ for $i = 1, \ldots, n$ that maximize the end-to-end throughput $\tau(p, \langle J_i \rangle)$. For $0 < i \le n$, we define $X_i = \{(e_i, j) | j \in A_i\}$ as the available link pairs that involve e_i. For $0 \le 1 \le n$, let $p_l = (v_0, v_1, \ldots, v_l)$ be the subpath that consists of the first l links of p. For $0 < l < n$, we define the *bridging set* as

$$B_l = \{(e_i, j), (e_{i'}, j') | i < 1, i' > 1, ((e_i, j), (e_{i'}, j')) \in G_{c,p}\}. \tag{4}$$

We also let $B_0 = X_1$, i.e., the set of available link-channel pairs for the first link e_1 in the path. Observe that $X_{l+1} \subset B_l$ for all $0 \le l < n$ due to the half-duplex constraint at each node. For $0 \le l < n$, let $\langle J_i \rangle_{i=1}^{l}$ be a CS for the subpath p_l, and let $B \subset B_l$.

Definition 2: We say that $\langle J_i \rangle_{i=1}^{l}$ is B-*compatible*, written $\langle J_i \rangle_{i=1}^{l} \sim B$, if, for all $1 \le i \le l$, $(e_i, j) \in B \Rightarrow j \in J_i$ and $(e_i, j) \in B_l \setminus B \Rightarrow j \notin J_i$.[1]

Suppose that $\langle J_i \rangle_{i=1}^{l} \sim B$. We are interested in calculating the throughput of this channel assignment for the partial path p_l. Because B may contain link-channel pairs from further along in p, we will include these pairs in the calculation, because these pairs may affect clique sizes. We will refer to this throughput as $\tau^B(p_l, \langle J_i \rangle)$. Next, let

$$\langle J_i^{l,B} \rangle_{i=1}^{l} = \underset{\langle J_i \rangle_{i=1}^{l} \sim B}{\arg\max}\, \tau^B\left(p_l, \langle J_i \rangle\right)$$

be a CS for the subpath p_l that is B-compatible and has the maximum end-to-end throughput along the subpath p_l. The main idea of the algorithm is that $\langle J_i^{l,B} \rangle$ can be computed by dynamic programming. In particular, suppose that $\langle J_i^{l,B} \rangle$ are known for all $B \subset B_l$, for some $0 \le l < n-1$. We will use these CSs to compute the $\langle J_i^{l+1,B'} \rangle$ for all $B' \subset B_{l+1}$. Let $B' \subset B_{l+1}$ and suppose that $\langle J_i^{l+1,B'} \rangle_{i=1}^{l+1}$ be an optimal CS for the subpath p_{l+1} that is compatible with B'. We define

$$B_{l+1}^{prev} = B_{l+1} \cap B_l \tag{5}$$

$$B_{l+1}^{new} = B_{l+1} \setminus B_l. \tag{6}$$

Note that $B_{l+1} = B_{l+1}^{new} \cup B_{l+1}^{prev}$. Let

$$B = \left(\bigcup_{i=1}^{l+1} J_i^{l+1,B'} \cup B' \right) \cap B_l.$$

We observe that $B \subset B_l$ and B agrees with B' on B_{l+1}^{prev} (which means that $B \cap B_{l+1}^{prev} = B' \cap B_{l+1}^{prev}$). Thus

$$\begin{aligned} \tau^{B'}\left(p_l, \left\langle J_i^{l+1,B'} \right\rangle_{i=1}^{l}\right) &= \tau^{B}\left(p_l, \left\langle J_i^{l+1,B'} \right\rangle_{i=1}^{l}\right) \\ &\le \tau^{B}\left(p_l, \left\langle J_i^{l,B} \right\rangle_{i=1}^{l}\right) \end{aligned} \tag{7}$$

because the throughput on all of the links in the subpath p_l is unchanged. We use the notation, $m_{e_{l+1},j}^{\langle J^{l+1,B'} \rangle, B'}$, to refer to the size of the largest clique that contains (e_{l+1}, j) present among the link-channel pairs from $\langle J^{l+1,B'} \rangle$ and B' together. Let

$$\langle J^{opt} \rangle = \langle J^{l,B} \rangle, J_{l+1}^{l+1,B'}$$

Be the channel assignments given by $\langle J^{l,B} \rangle$ for the subpath p_l and by $J_{l+1}^{l+1,B'}$ for link e_{l+1}. A key observation is that $m_{e_{l+1},j}^{\langle J^{l+1,B'} \rangle, B'} = m_{e_{l+1},j}^{\langle J^{opt} \rangle, B'}$ for any $(e_{l+1}, j) \in J_{l+1}^{l+1,B'}$, because

[1] We use the set-difference notation, $A \setminus B = A \cap \bar{B}$.

both B_i and B_{i+1} will contain any link-channel pairs from p_{l+1} that conflict with it. We have

$$\tau^{B'}\left(p_{l+1},\langle J^{l+1,B'}\rangle\right)=\min\left(\tau^{B}\left(p_l,\langle J^{l+1,B'}\rangle_{i=1}^{l}\right)\right.$$
$$\left.\sum_{j\in J_{l+1}^{l+1,B'}} b_{e_{l+1},j}/m_{e_{l+1},j}^{\langle J^{l+1,B'}\rangle,B'}\right)$$
$$\leq\min\left(\tau^{B}\left(p_l,\langle J^{l,B}\rangle_{i=1}^{l}\right)\right.$$
$$\left.\sum_{j\in J_{l+1}^{l+1,B'}} b_{e_{l+1},j}/m_{e_{l+1},j}^{\langle J^{opt}\rangle,B'}\right)$$
$$=\tau^{B'(p_{l+1},\langle J^{opt}\rangle)}. \quad (8)$$

Because $\langle J^{l+1,B'}\rangle$ was assumed optimal, we must have equality in (8), and therefore, we can take

$$\langle J^{l+1,B'}\rangle=\langle J^{opt}\rangle. \quad (9)$$

Equation (8) shows that the optimal CS for p_{l+1} and B' can be expressed in terms of an optimal CS for p_l and B, a smaller problem. This means that we can use dynamic programming to compute $\langle J^{l+1,B'}\rangle$. Algorithm 1 uses this approach. The idea is to take each $\langle J^{l,B}\rangle$ found for p_l and extend to channel assignment for p_{l+1} for all $B'\subset B_{l+1}$ such that B and B' agree on B_{l+1}^{prev}. Because B fixes which link pairs from B_{l+1}^{prev} are included in B', we can simply enumerate all subsets $B''\subset B_{l+1}^{new}$, and for each, we form $B'=(B\cap B_{l+1}^{prev})\cup B''$. In addition, because $X_{i+1}\subset B_i$, we also let B determine all channel assignments for link e_{l+1}, i.e.,

$$\langle J^{test}\rangle=\langle J^{l,B}\rangle,(B\cap X_{i+1}). \quad (10)$$

We then evaluate $\tau^{B'}(p_{l+1},\langle J^{test}\rangle)$ to see if J^{test} provides better channel assignment for p_{n+1} that is compatible with B', and if yes, we keep it. This evaluation is done similar to (8), i.e.,

$$\tau^{B'}\left(p_{l+1},\langle J^{test}\rangle\right)=\min\left(\tau^{B}\left(p_l,\langle J^{l,B}\rangle_{i=1}^{l}\right)\right.$$
$$\left.\sum_{j\in B\cap X_{i+1}} b_{e_{l+1},j}/m_{e_{l+1},j}^{\langle J^{test}\rangle,B'}\right). \quad (11)$$

To evaluate (11), we need to calculate the second min term, because the first min term is already known. This approach is done by determining the largest clique that (e_{l+1},j) participates among link-channel pairs from $\langle J^{test}\rangle,B'$. If we make the assumption that the routing path p is self-avoiding, then any clique in involving (e_{l+1},j) is either a consecutive list of link-channel pairs in $G_{c,p}^{j}$ (which all mutually overlap in the interval graph representation) or a clique that involves (e_{l+1},j) and a link-channel pair from X_l, X_{l+2}, or both, using a channel other than j. Let Δ be the maximum degree of any node in any of the $G_{c,p}^{j}$, and suppose that at most m_p channels are available on any link in p. All of the clique possibilities can be checked in $O(\Delta m_p)$ time, and therefore, this is the time required to evaluate (11). We can now state the entire DP-ChannelSelect algorithm as follows.

Algorithm 1: DP-ChannelSelect.

INPUT: A self-avoiding path $p=(v_0,v_1,\ldots v_n)$ from s to t.
Step 1 **for** $l=0$ to $n-1$
  Compute B_l, B_l^{prev}, and B_l^{new} using (4), (5) and (6).
 endfor
Step 2 **for** $l=1$ to $n-1$
  forall $B\subset B_l$
    forall $B''\subset B_{l+1}^{new}$
      Construct $B'=(B\cap B_{l+1}^{prev})\cup B''$ and $\langle J^{test}\rangle$ using (10).
      Calculate $\tau^{test}=\tau^{B'}(p_{l+1},\langle J^{test}\rangle)$ using (11).
      if $\tau^{test}>\tau^{B'}(p_{l+1},\langle J^{l+1,B'}\rangle)$
        set $\langle J^{l+1,B'}\rangle=\langle J^{test}\rangle$
      endif
    endforall
  endforall
 endfor
Step 3 Compute
  $B^*=\arg\max_{B\subset B_{n-1}}\tau^{B}(p_{n-1},\langle J^{n-1,B}\rangle,B\cap X_n)$
Step 4 **return** $\langle J^{n-1,B^*}\rangle,B^*\cap X_n$

Next, we analyze the time complexity of Algorithm 1. The running time is dominated by step 2, which must examine every subset of B_{l+1}^{new} for every subset of B_l for $l=1,\ldots,n-1$. The number of subset combinations examined for each l is $2^{|B_l||B_{l+1}^{new}|}$. If p is self-avoiding, then for any l, B_l will comprise consecutive lists of link-channel pairs in $G_{c,p}^{j}$ for each channel j. For each j, the list will be at most $\Delta+1$ in length, because every other link-channel pair in the list will conflict with the $(e_{l'},j)$ in the list with maximum $l'\leq l$. Let m_p be the total number of channels available along p. There can be at most $m_p(\Delta+1)$ link-channel pairs in B_l. Because $B_{l+1}^{new}\subset B_{l+1}$, the same bound applies to it. Thus, the number of subset combinations examined is at most $2^{m_p^2(\Delta+1)^2}$. The time required for evaluating each subset combination is $O(\Delta m_p)$ time, as stated previously. This implies that the overall running time of the algorithm is $O(2^{m_p^2(\Delta+1)^2}\Delta m_p n)$ time, provided that the input routing path is self-avoiding. In practice, the running time can be much faster, because typically, each B_l will be smaller than $m_p(\Delta+1)$ in size, and B_{l+1}^{new} will still be smaller. We remark that this bound places CS in the class of *fixed-parameter tractable* problems [8], where the parameters are Δ and m_p. The aforementioned discussion leads to the following result.

Theorem 1: Algorithm 1 computes an optimal end-to-end channel assignment in $O(2^{m_p^2(\Delta+1)^2}\Delta m_p n)$ time, where p is a self-avoiding routing path of length n, m_p is the maximum number of available channels in any link in p, and Δ is the maximum degree of any node in the path conflict graph $G_{c,p}$.

We observe that the effectiveness of Algorithm 1 heavily depends on the parameters m_p and Δ. Provided that these values are not very large, the algorithm quickly runs in practice; most of the optimal CSs found in our experiments were computed in a few seconds on a laptop computer. However, the performance of the algorithm will degrade and perhaps become impractical if these parameters are very large. We observe that Algorithm 1 can be implemented in a distributed fashion and carried out by the nodes themselves along the routing path p, because the algorithm makes a single pass over the length of p, and the sets of intermediate path/channel assignments computed at each node along the path depend only on those from the previous node and relatively local interference information (as determined by the B_l sets).

V. Channel-Aware Routing

In this section, we present two heuristic algorithms for JRCS.

A. Path Extension Algorithm

The first proposed algorithm is based on the idea of simultaneously finding good path/channel assignments from the source node s to all other nodes in the network, including the destination node t. The approach is similar to the Bellman–Ford shortest path algorithm, because new path/channel assignments are generated by "relaxing" all the links in the network in a series of phases that continues until no better paths are discovered. Each node $u \neq s$ will store a list P_u of path/channel assignments of the form $(p_u, \langle J_e \rangle_{e \in p_u})$, where p_u is a path from s to u. When the link (u, v) is relaxed, the current path/channel assignments stored at u will be augmented by the link (u, v); for each subset $J' \subset A_{(u,v)}$, we will create a new path/channel assignment $(p'_v, \langle J_e, J' \rangle)$, where p'_v is the path p_u, followed by the link (u, v). To limit the search, each node keeps a maximum of d best path/channel assignments (we chose $d = 100$). We also note that it is only necessary to augment path/channel assignments that were created in the previous phase; we will refer to these path/channel assignments as *new* in the algorithm. Clearly, a loop in a path can never improve the end-to-end throughput of the path, and therefore, we restrict attention to *simple* paths that never repeat a vertex. We also require that the path is self-avoiding. The complete algorithm is given in Algorithm 2. (Algorithm 3 is a subroutine used by Algorithm 2.)

Algorithm 2: RCS-PathExtend.

INPUT: A connection request from s to t, $G = (V, E)$.
Step 1 **repeat**
  forall $e \in E$
    Link-relax(e)
  endforall
until no list P_u changes
Step 2 **return** $\arg\max_{(p,\langle J \rangle) \in P_t} \tau(p, \langle J \rangle)$

Algorithm 3: Link-Relax.

INPUT: A link $e = (u, v)$.
forall new $(p_u, \langle J_e \rangle_{e \in p_u}) \in P_u$
  Let $p_v = p_u + (u, v)$
  if p_v is a simple self-avoiding path
    forall $J_{(u,v)} \subset A_{(u,v)}$
      Construct $\langle J' \rangle = \langle J_e, J_{(u,v)} \rangle$
      Compute $\tau' = \tau(p_v, \langle J' \rangle)$
      if ($|P_v| < d$ or $\tau' > \min_{(p,\langle J \rangle) \in P_v} \tau(p, \langle J \rangle)$)
        Insert $(p_v, \langle J' \rangle)$ into P_v
        if $|P_v| > d$
          remove $\arg\min_{(p,\langle J \rangle) \in P_v} \tau(p, \langle J \rangle)$ from P_v
        endif
      endif
    endforall
  endif
endforall

B. Bottleneck Routing

The second approach attempts to find a single path whose links all have a high useful capacity. We make this precise as follows. We define the link capacity $c(e)$ of a link e as $c(e) = \sum_{j \in A_e} b_{e,j}$. The link capacity provides an upper bound on the bit flow rate achievable by link e, ignoring intrapath interference. In addition to link capacity, we also take into account how close the link $e = (u, v)$ to the source s and destination t is.

For each link $e = (u, v)$, we define the source–destination distance as $d(e) = \|u - s\| + \|u - t\| + \|v - s\| + \|v - t\|$. Let $d_{max} = \max_{e \in E} d(e)$ and $d_{min} = \min_{e \in E} d(e)$ be the maximum and minimum source–destination distances, respectively. For a heuristic additional weighting factor on the link capacity to better estimate the *usefulness* of a link e for an s-t path, we define

$$u(e) = \left(1 + \frac{d_{max} - d(e)}{d_{max} - d_{min}}\right) c(e). \tag{12}$$

Note that $1 \leq u(e) \leq 2$, with $u(e) = 1$ when $d(e) = d_{max}$ and $u(e) = 2$ when $d(e) = d_{min}$. Let the *useful bottleneck capacity* of a path p be defined as

$$c(p) = \min_{e \in p} u(e).$$

Our goal is to find a path p that maximizes $c(p)$. This is a well-known problem that can efficiently be solved by computing a minimum spanning tree T on the network graph using a link weight function $w(e) = -u(e)$. The unique path in T from s to t will have maximum useful bottleneck capacity. We call this the *bottleneck route*. It can easily be computed using Algorithm 4. Once the bottleneck route has been found, an optimal channel section for the route can be computed using Algorithm 1.

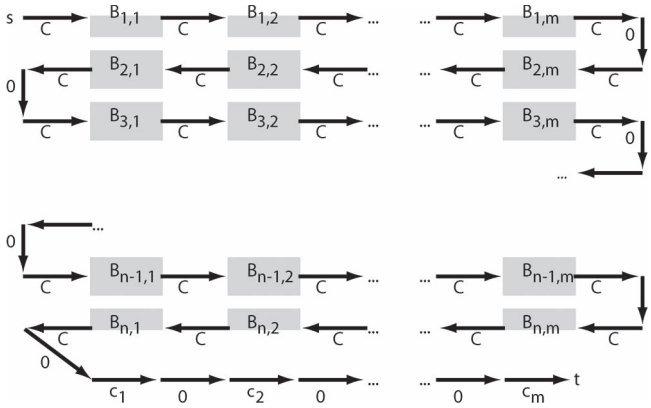

Fig. 2. Layout of the JRCS instance that corresponds to an instance of exact cover.

Algorithm 4: Bottleneck-Route.

INPUT: A connection request from s to t, $G = (V, E)$.
Step 1 Assign edge weights $-u(e)$ to each link $e = (u, v)$ using (12).
Step 2 Compute a minimum weight spanning tree T.
Step 3 Compute the route p from s to t in T.
Step 4 **return** p.

VI. Computational Complexity of Joint Routing and Channel Selection

In this section, we show that the JRCS problem is NP-hard to approximate to within a factor of 2/3 + ϵ for any $\epsilon > 0$. This is done through a reduction from the Exact Cover problem [9]. An instance of this problem consists of a set $U = \{u_1, \ldots u_m\}$ and a collection of subsets of U, $F = \{S_1, \ldots S_n\}$. The problem is to determine if there exists a subcollection $F' \subset F$ such that, for each $u_j \in U$, there is a unique $S_i \in F'$ such that $u_j \in S_i$. Fig. 2 provides a sketch of the network for which we show that it is computationally difficult (NP-hard) to find the best routing and CS. In each of the shaded subblocks in the figure, there are several options for the path to go (as shown in Fig. 3). Because of interference, the choice of paths (and channels) in one subblock, the choices for the neighboring subblocks directly above and below lead to the following theorem.

Theorem 3: The JRCS problem is NP-hard to approximate to within a factor of 2/3 + ϵ for any $\epsilon > 0$.

Proof: We can reduce an instance of the Exact Cover problem to an instance of the JRCS problem as follows. The JRCS instance will consist of a layout of n rows by m columns of *blocks* $B_{i,j}$, as shown in Fig. 2 (the layout for odd n is shown; for even n, the final path to t will go from right to left instead). There are two basic types of blocks, depending on whether $u_j \in S_i$ and they can be oriented left to right or right to left (see Fig. 3). The blocks in the first (last) row are modified by removing the upper (lower) path in the construction diagram. We also use a specific sequence of four links, called a *C-chain*, which is also shown in the figure. We assume the following interference structure: Two links that both use channels 1 or 2 will interfere with each other if there is at most one link that

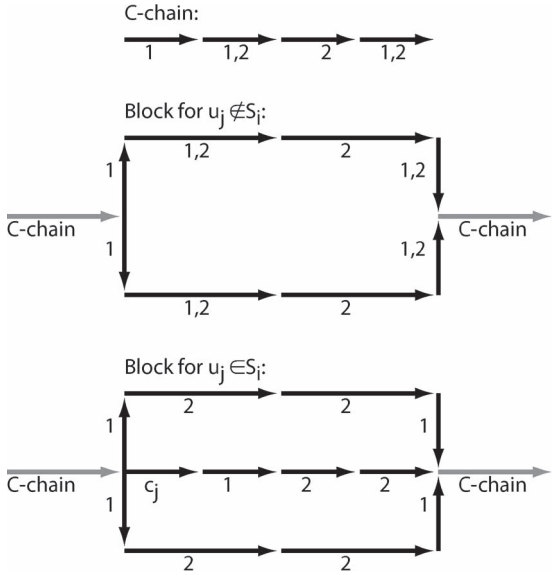

Fig. 3. C-chain construction and block types, depending on whether $u_j \in S_i$. Left-to-right versions are shown.

separates them on the selected path. The links (on channel 2) of the bottom path of block $B_{i,j}$ interfere with the links on the top path of block $B_{i+1,j}$ for $i = 1, \ldots n-1$. Links that use channel 0 do not interfere with each other. A separate channel c_j is available for each $j = 1, \ldots m$, and all links using channel c_j interfere with each other. Furthermore, we will assume that $b_{e,j} = 1$ for all e and j. We claim that a solution to the Exact Cover instance exists if and only if there exists a routing path p from s to t in the JRCS instance with an end-to-end throughput of 1/2. For links e with a single channel available j, this implies that the size of the largest cliques to which the link-channel pair (e, j) can belong is 2. We observe that this forces the links with two channels available in each C-chain to select only one channel; otherwise, the third link will belong to a clique of size 3. This also requires that, if the selected path traverses the bottom path of block $B_{i,j}$, then the path cannot go through the top path of block $B_{i+1,j}$ (otherwise, a clique of size 4 is formed). To prevent this from happening, observe that, in each column j, at least one path must use the center path in the block that corresponds to $u_j \in S_i$. Furthermore, no more than one center path can be used in column j, because c_j is used along each such path and the final portion of the path to t. These links all mutually interfere, and the maximal clique size cannot exceed 2. Finally, we observe that the channel that is chosen for the second link of the first C-chain traversed in the given row determines that this channel must be chosen for all C-chains in the row. We also note that, if channel 2 is chosen for the second link, then the fourth link must choose channel 1; this means that the center path in any block in the row cannot be entered, because this will create a clique of size 3 with the c_j link in the block. Conversely, if channel 1 is chosen for the second link in the first C-chain, then the fourth link must use channel 2, and the center path must always be chosen if a $u_j \in S_i$ block is encountered (because taking the top or bottom path creates a clique of size 3). It follows that choosing channel 1 for the second link in the first C-chain encountered in row i corresponds to adding S_i to the cover

TABLE I
MAXIMUM TRANSMISSION DISTANCES BY FREQUENCY AND DATA RATE

Transmission rate	700 Mhz	2400 Mhz	5800 Mhz
45 Mbps	15.4 km	4.5 km	1.8 km
40 Mbps	18.4 km	5.3 km	2.2 km
30 Mbps	30 km	8.6 km	3.6 km
20 Mbps	41 km	11.8 km	4.9 km
10 Mbps	68 km	20 km	8.2 km

TABLE II
INTERFERENCE RANGES BY FREQUENCY

Frequency	Interference range
700 Mhz	30.8 km
2400 Mhz	9 km
5800 Mhz	3.6 km

F' and choosing channel 2 corresponds to omitting S_i from F'. Observe that each $u_j \in U$ must be covered (there are n rows and only $n-1$ top/bottom block paths; therefore, some row must use center paths exactly once (if two center paths are used for some column j, this creates a clique of size 3 on links that use channel j). Hence, an end-to-end throughput of 1/2 is achievable if and only if an exact cover F' exists. The next lowest potential end-to-end throughput for this JRCS instance is 1/3 (at least one single-channel link is involved in a clique of size 3). If an approximation algorithm for JRCS found a solution at least $2/3+\epsilon$ of optimal, it would find a solution with end-to-end throughput at least $(2/3+\epsilon)1/2 > 1/3$; therefore, it would necessarily find the optimal solution and could be used to solve exact cover. It follows that JRCS is NP-hard to approximate to within a factor of $2/3+\epsilon$. We remark that this also places JRCS in the complexity class APX-hard. ■

VII. NUMERICAL RESULTS

To test our algorithms for CS (see Algorithm 1) and JRCS (see Algorithms 2 and 4), we assumed that three widely spaced frequency bands were available for licensed and unlicensed operations and that the link throughput for each channel was the maximum available, given the link distance and frequency used. The bands exhibit widely ranging propagation, transmission range, and usage characteristics, highlighting the potential value of cognition in transmission scheduling. Tables I and II summarize our assumptions about the transmission rates and interference ranges of each frequency. The table values are reflective of a combination of manufacturer data sheets and our own experience in using these radios in the field. The values in Table I are based on a scenario where each node transmits at 1 W with a 2-dBi antenna and where the receiving antenna has a gain of 2 dBi. The channel bandwidth is 10 MHz, the receiver noise figure is 5 dB, and implementation losses of 3 dB are assumed for each link. The values in Table II are reflective of the widely used rule of thumb that the interference range is about a factor of two greater than the transmission range.

Path loss is calculated using line-of-sight and free-space characteristics. Typical IEEE 802.16 adaptive modulation and coding parameters performance parameters were used to estimate the throughput achievable as a function of the carrier-to-noise ratio and were then translated into the allowable path-loss threshold. The maximum channel transmission rate is a function of distance and frequency (at lower frequency, the maximum distance for a given transmission rate will be greater). We assumed that each channel was available on each link with independent probability 0.3, unless otherwise noted. Primary users were placed at random locations and assigned a random channel. This channel was then made unavailable to any links of cognitive radios within the interference range of the primary user. The number of primary users was set to be half the number of channels.

We compared the routing paths found by Algorithms 2 and 4 with the shortest path routing (the edge weights in this case were the physical link distances). As a benchmark, we also implemented a simple greedy strategy for CS. The algorithm sequentially selects channels for each link e in the given path P based on the available channel sets $\{A_e\}$. We initialize $J_1 = A_1$. For $i = 2, \ldots n$, we use the following simple rule:

$$J_i = \begin{cases} A_i \setminus J_{i-1}, & \text{if } A_i \setminus J_{i-1} \neq \emptyset \\ A_i, & \text{otherwise.} \end{cases}$$

The idea of the algorithm is to select new channels, where possible, on successive links in the routing path, but if that is not possible, then all available channels are (greedily) chosen. This leads to the following six routing and CS algorithm combinations.

1) SP-Gdy: shortest path routing, followed by greedy CS;
2) SP-DPCS: shortest path routing, followed by the optimal CS (see Algorithm 1);
3) Btl-Gdy: Bottleneck routing (see Algorithm 4), followed by greedy CS;
4) Btl-DPCS: Bottleneck routing (see Algorithm 4), followed by the optimal CS (see Algorithm 1);
5) RCS: JRCS (see Algorithm 2);
6) RCS-DPCS: RCS routing, followed by the optimal CS (see Algorithm 1).

For all runs of the RCS algorithm, the parameter d that controls the number of partial paths kept at each node was set to 10.

We tested each of these methods on the following three different simulation scenarios: 1) The number of available channels in the network was varied; 2) the network region size was varied with constant node density; and 3) the node density was varied within a constant region size. For each combination of parameters, 50 random network configuration instances were created. The average throughput over these instances was computed, and the results are shown in Figs. 4–6.

A. Scenario 1: Varying the Number of Channels Available

In this scenario, the number of channels available to secondary users was varied from 3 to 15, with a step size of 3 (equally chosen from each frequency band). Random problem instances were generated for a 50×50 km^2 network with 25 nodes. The average end-to-end transmission rate (throughput) for all tested algorithms is reported. The results, as shown in Fig. 4, indicate that an almost-linear improvement is gained by adding additional channels to the network in terms of additional throughput. The path and channel assignments found

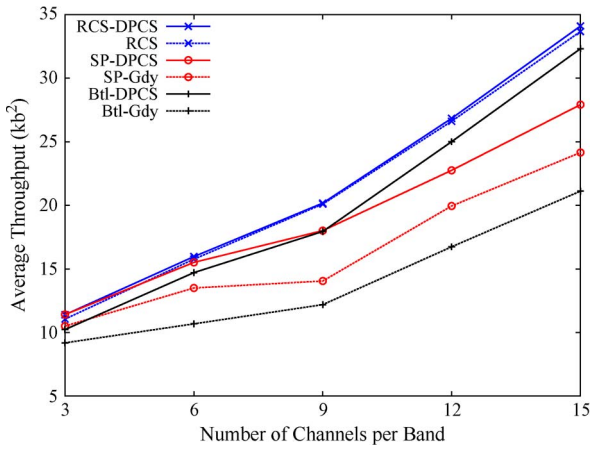

Fig. 4. Average path end-to-end throughput versus the number of available channels.

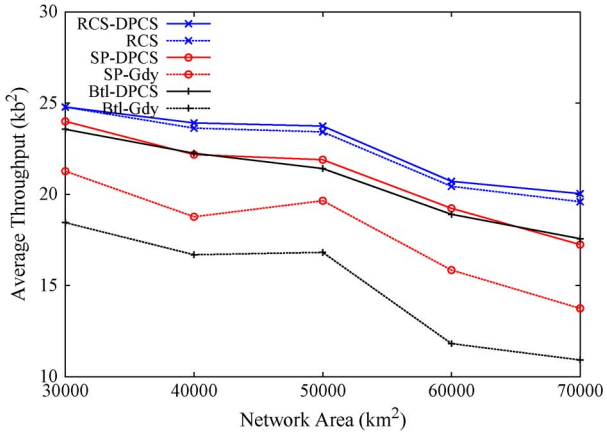

Fig. 5. Average path end-to-end throughput versus network size. The node density was held constant at 0.01 nodes/km^2.

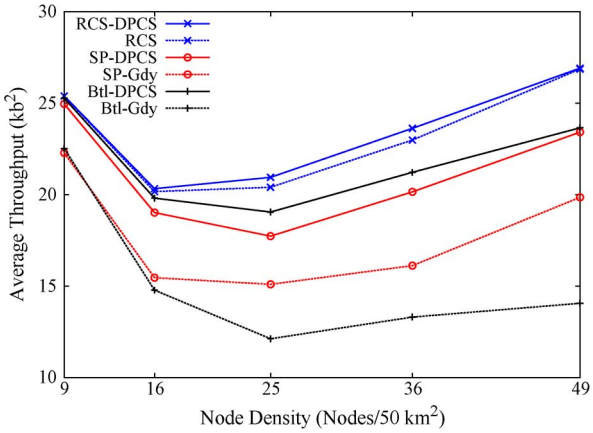

Fig. 6. Average path end-to-end throughput versus node density. The region size was fixed to 50×50 km^2.

by the best performing algorithm, i.e., RCS-DPCS, were, on average, 52.2% better than those found by the worst performing algorithm, Btl-Gdy. The RCS algorithm gave almost-identical results to the RCS-DPCS algorithm and had a 27.6% improvement over SP-Gdy on the average. On average, using DPCS for CS provided a 28.0% improvement versus using the CS found by Gdy.

B. Scenario 2: Varying the Region Size

In this scenario, the physical region size was increased, but the node density was held constant. Random problem instances were generated on a network with three channels per frequency band and a constant density of 0.01 nodes/km^2. The average end-to-end transmission rate (throughput) for all tested algorithms is reported. As the region size grows, paths tend to get longer. The results, as shown in Fig. 5, indicate that the RCS-DPCS and RCS algorithms continue to outperform the other routing and channel assignment methods and that the gap slightly increases as the region size increases. The average throughput somewhat decreases as the size of the simulated network increases. The slight staircase nature of the graph suggests some interplay between the diminished transmission capacity as distances increase versus potentially less intraflow interference along the transmission path. The path and channel assignments found by the best performing algorithm, RCS-DPCS, were, on the average, 55.6% better than those found by the worst performing algorithm, Btl-Gdy. The RCS algorithm gave almost-identical results to the RCS-DPCS algorithm and had a 26.6% improvement over SP-Gdy on the average. On the average, using DPCS for CS provided a 29.9% improvement versus using the CS found by Gdy.

C. Scenario 3: Varying the Node Density

In this scenario, the physical region size was held constant at 50×50 km^2, and the number of nodes was increased. Again, random problem instances were generated on a network with three channels per frequency band. As the number of nodes size grows, the node density increases, and the average number of links with which a given link interferes on a given channel increases. The effect is to increase the average vertex degree in the conflict graph G_c. The results, as shown in Fig. 6, indicate that, again, the RCS-DPCS and RCS algorithms outperform the remaining approaches across the range of node densities considered. In going from nine nodes to 16 or 25 nodes, all of the algorithms suffer a loss of throughput performance. This case may be explained by increased interference that is not offset by the additional transmission capacity offered by potentially shorter links. We note that the relative performance gaps among the algorithms increase as the number of nodes increase. The path and channel assignments found by the best performing algorithm, i.e., RCS-DPCS, were, on average, 58.4% better than those found by the worst performing algorithm Btl-Gdy. The RCS algorithm gave almost-identical results to the RCS-DPCS algorithm and had a 31.5% improvement over SP-Gdy on the average. On the average, using DPCS for CS provided a 32.7% improvement versus using the CS found by Gdy.

D. Scenario 4: Varying the Channel Availability Probability

In this scenario, the physical region size was held constant at 50×50 km^2, and the number of nodes was set to 25, with three channels per frequency band. The (independent) probability p that each channel was available on a link was varied in the range $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The results, as shown in Fig. 7, indicate

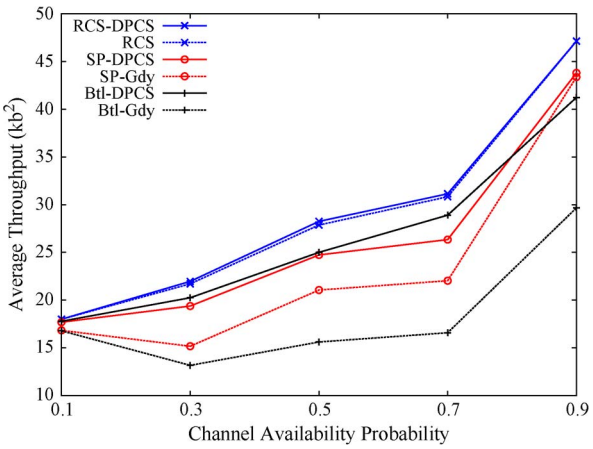

Fig. 7. Average path end-to-end throughput versus channel availability probability.

that, again, the RCS-DPCS and RCS algorithms outperform the remaining approaches across the range of probabilities considered. In going from $p = 0.1$ to $p = 0.3$, we note that Btl-Gdy and SP-Gdy suffer some loss in average performance. This is indicative that the Gdy CS strategy is suboptimal when the channel availability is low. Interestingly, the SP-Gdy substantially improves when the channel availability probability increases to $p = 0.9$. The path and channel assignments found by the best performing algorithm, i.e., RCS-DPCS, were, on average, 60.4% better than those found by the worst performing algorithm, i.e., Btl-Gdy. The RCS algorithm gave almost-identical results to the RCS-DPCS algorithm and had a 26.2% improvement over SP-Gdy on average. On the average, using DPCS for CS provided a 30.4% improvement versus using the CS found by Gdy.

We conducted a further experiment to measure the effect of variable channel availability probability versus each channel being available with the same probability p. In this case, we compared each channel with availability probability $p = 0.5$ versus the three channels in each band being available, with probabilities 0.25, 0.5, and 0.75, respectively. We note that this change does not affect the expected number of available channels per band. We found that, across all algorithms, there was an average 11.9% performance gain in the asymmetric scenario. One explanation for this result is that links are more likely to have a channel that best suits their distance; the higher frequencies are more suited to shorter distances, and the lower frequencies are better for longer distances.

It is interesting to note that, in all four scenarios, in nearly all cases, the RCS algorithm found an optimal CS for its computed path. Only in about 2% of the test cases did RCS-DPCS provide a slightly better CS. The RCS algorithm has the advantage of being simple to implement and, with low-degree polynomial time complexity, is fast in practice.

VIII. Conclusion

In this paper, we have examined two important problems for maximizing the end-to-end throughput for communication flows in CRNs. For the CS problem on a known routing path, we presented an optimal algorithm and showed that, for self-avoiding paths, it runs in time linear in the length of the path. Furthermore, the algorithm only needs to propagate local information from the source to the destination and can be implemented in a distributed fashion. We also considered the JRCS problem and showed that it was NP-hard to approximate within a factor of $(2/3 + \epsilon)$. In addition, we presented a heuristic algorithm for the joint problem, the RCS algorithm, and another heuristic approach for routing based on the bottleneck capacity. In our experiments, RCS provided better performance than methods that first computed a routing path and then found a CS for that path. In addition, the CS that was provided by RCS for the routing path that it found was nearly always optimal. This result, coupled with the simplicity and speed of RCS, suggests using it in practice.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *J. Comput. Netw.*, vol. 47, no. 4, pp. 445–487, Mar. 2005.
[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next-generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *J. Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.
[3] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," in *Proc. ACM MobiCom*, 2005, pp. 58–72.
[4] H. Almasaeid, T. Jawadwala, and A. Kamal, "On-demand multicast routing in cognitive radio mesh networks," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–5.
[5] L. Cao and H. Zheng, "Distributed spectrum allocation via local bargaining," in *Proc. IEEE SECON*, 2005, pp. 475–486.
[6] Y. Ding, K. Pongaliur, and L. Xiao, "Channel allocation and routing in hybrid multi-channel multi-radio wireless mesh networks," *IEEE Trans. Mobile Comput.*, Dec. 8, 2011. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TMC.2011.261
[7] "Unlicensed operation in the TV broadcast bands," Fed. Commun. Comm., Washington, DC, FCC 08-260, 2008.
[8] R. Downey and M. Fellows, *Parameterized Complexity*. New York: Springer-Verlag, 1999.
[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1990.
[10] R. Hincapie, J. Tang, G. Xue, and R. Bustamante, "QoS routing in wireless mesh networks with cognitive radios," in *Proc. IEEE GLOBECOM*, 2008, pp. 1–5.
[11] Y. T. Hou, Y. Shi, and H. D. Sherali, "Optimal spectrum sharing for multihop software-defined radio networks," in *Proc. IEEE INFOCOM*, 2007, pp. 1–9.
[12] J. Huang, R. A. Berry, and M. L. Honig, "Spectrum sharing with distributed interference compensation," in *Proc. IEEE DySPAN*, 2005, pp. 88–93.
[13] H. Khalife, S. Ahuja, N. Malouch, and M. Krunz, "Probabilistic path selection in opportunistic cognitive radio networks," in *Proc. IEEE GLOBECOMM*, 2008, pp. 1–5.
[14] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multiradio multichannel wireless mesh networks," in *Proc. ACM MobiCom*, 2005, pp. 73–87.
[15] N. Kumar, M. Kumar, and R. Patel, "Capacity- and interference-aware link scheduling with channel assignment in wireless mesh networks," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 30–38, Jan. 2011.
[16] B. Mumey, X. Zhao, J. Tang, and R. Wolff, "Transmission scheduling for routing paths in cognitive radio mesh networks," in *Proc. IEEE SECON*, 2010, pp. 1–8.
[17] S. Olariu, "An optimal greedy heuristic to color interval graphs," *Inf. Process. Lett.*, vol. 37, no. 1, pp. 21–25, Jan. 1991.
[18] I. Pefkianakis, S. Wong, and S. Lu, "SAMER: Spectrum-aware mesh routing in cognitive radio networks," in *Proc. IEEE DySPAN*, 2008, pp. 1–5.
[19] V. Ramamurthi, A. Reaz, D. Ghosal, S. Dixit, and B. Mukherjee, "Channel, capacity, and flow assignment in wireless mesh networks," *Comput. Netw.*, vol. 55, no. 9, pp. 2241–2258, Jun. 2011.
[20] A. Sampath, L. Yang, L. Cao, H. Zheng, and B. Y. Zhao, "High-throughput spectrum-aware routing for cognitive-radio-based ad hoc networks," in *Proc. CrownCom*, Singapore, 2008.

[21] Y. Shi and Y. T. Hou, "A distributed optimization algorithm for multihop cognitive radio networks," in *Proc. IEEE INFOCOM*, 2008, pp. 1292–1300.
[22] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multichannel wireless mesh networks," in *Proc. ACM MobiHoc*, 2005, pp. 68–77.
[23] J. Tang, S. Misra, and G. Xue, "Joint spectrum allocation and scheduling for fair spectrum sharing in cognitive radio wireless networks," *Comput. Netw. J.*, vol. 52, no. 11, pp. 2148–2158, Aug. 2008.
[24] P. Wan, "Multiflows in multihop wireless networks," in *Proc. MobiHoc*, 2009, pp. 85–94.
[25] F. Wang, M. Krunz, and S. Cui, "Spectrum sharing in cognitive radio networks," in *Proc. IEEE INFOCOM*, 2008, pp. 1885–1893.
[26] Y. Xi and E. M. Yeh, "Distributed algorithms for spectrum allocation, power control, routing, and congestion control in wireless networks," in *Proc. ACM MobiHoc*, 2007, pp. 180–189.
[27] C. Xin, B. Xie, and C.-C. Shen, "A novel layered graph model for topology formation and routing in dynamic spectrum access networks," in *Proc. IEEE DySPAN*, 2005, pp. 308–317.
[28] Y. Yang and R. Kravets, "Contention-aware admission control for ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 4, pp. 363–377, Jul./Aug. 2005.
[29] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating dynamic time-spectrum blocks in cognitive radio networks," in *Proc. ACM MobiHoc*, 2007, pp. 130–139.
[30] Q. Zhao, L. Tong, and A. Swami, "Decentralized cognitive MAC for dynamic spectrum access," in *Proc. IEEE DySPAN*, 2005, pp. 224–232.
[31] H. Zheng and C. Peng, "Collaboration and fairness in opportunistic spectrum access," in *Proc. IEEE ICC*, 2005, pp. 3132–3136.

Brendan Mumey (M'07) received the B.Sc. degree in mathematics from the University of Alberta, Edmonton, AB, Canada, in 1990, the M.Sc. degree in computer science from the University of British Columbia, Vancouver, BC, Canada, in 1992, and the Ph.D. degree in computer science from the University of Washington, Seattle, in 1997.

In 2011, he was a Visiting Fulbright Distinguished Chair with Aalto University, Aalto, Finland. He is currently an Associate Professor of computer science with the Department of Computer Science, Montana State University, Bozeman. His research interests include algorithms, networks, and computational biology. He has published more than 50 conference proceedings and journal papers.

Dr. Mumey has served on various technical program committees in networking, including the IEEE Global Telecommunications Conference and the IEEE International Conference on Communications.

Jian Tang (M'06) received the Ph.D. degree in computer science from Arizona State University, Tempe, in 2006.

He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY. His research interests include wireless networking and cloud computing.

Dr. Tang received the National Science Foundation Faculty Early Career Development Award in 2009. He is currently an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He has served or is serving as a Cochair of the 2011 IEEE Global Telecommunications Conference (GLOBECOM) Wireless Networking Symposium, the 2012 IEEE International Conference on Computing, Networking, and Communications (ICNC) Wireless Networks Symposium, and the 2013 ICNC Wireless Networks Symposium. He has also served on the Technical Program Committee of many IEEE and Association for Computing Machinery (ACM) conferences, such as the 2010–2013 IEEE International Conference on Computer Communications, the 2006–2012 IEEE International Conference on Communications, the IEEE 2006–2012 GLOBECOM, and the 2008–2009 ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems.

Ivan R. Judson is currently working toward the Ph.D. degree with the Department of Computer Science, Montana State University, Bozeman, studying television whitespace and rural networking.

He has been with Montana State University and Argonne National Laboratory, Argonne, IL, where he developed software and built software development teams to address large-scale data management, collaboration, and high-performance computing challenges. He is currently with WebFilings, working on eXtensible Business Reporting Language data warehousing.

David Stevens received the B.S. degree in computer science from Montana State University, Bozeman. He is currently working toward the Ph.D. degree in computer science with the University of Oregon, Eugene.