Disjoint Path Protection in Multi-Hop Wireless Networks with Interference Constraints

Greg Kuperman
MIT LIDS
Cambridge, MA 02139
gregk@mit.edu

Eytan Modiano
MIT LIDS
Cambridge, MA 02139
modiano@mit.edu

***Abstract*—We consider the problem of providing protection against failures in wireless networks by using disjoint paths. Disjoint path routing is commonly used in wired networks for protection, but due to the interference between transmitting nodes in a wireless setting, this approach has not been previously examined for wireless networks. In this paper, we develop a non-disruptive and resource-efficient disjoint path scheme that guarantees protection in wireless networks by utilizing capacity "recapturing" after a failure. Using our scheme, protection can oftentimes be provided for all demands using *no* additional resources beyond what was required without any protection. We show that the problem of disjoint path protection in wireless networks is not only NP-hard, but in fact remains NP-hard to approximate. We provide an ILP formulation to find an optimal solution, and develop corresponding time-efficient algorithms. Our approach utilizes 87% less protection resources on average than the traditional disjoint path routing scheme. For the case of 2-hop interference, which corresponds to the IEEE 802.11 standard, our protection scheme requires only 8% more resources on average than providing no protection whatsoever.**

I. Introduction

Protecting networks against failure has been the subject of much study over the years. Network infrastructure has traditionally been wired, being either copper or optical fiber. Consequently, network protection schemes have been developed for the parameters of wired networks, with disjoint path protection being one of the most commonly used [1]. Multi-hop wireless networks have emerged as a promising alternative to wired networks for both backbone and last-mile Internet services, particularly in developing nations [2]. Hence, it has become increasingly important to make wireless networks equally resilient to failure as their wired counterparts. Failures in wireless networks can occur due to node failure, obstructions, deep fades, as well as malicious attacks [3]. As opposed to wired networks, wireless networks must also handle the additional complexity of interference, which occurs when two nodes transmit simultaneously using the same frequency channel. Because of these interference constraints, simply applying the disjoint path protection scheme used in wired networks to the wireless setting would not necessarily be effective. Switching to a backup path after some failure may cause interference with already existing paths, disrupting connections that were not affected by that failure. Additionally, the increased demand on a wireless network's already scarce resources may make protection prohibitive. In this work, we consider the problem of providing non-disruptive and resource-efficient protection by using pre-planned disjoint paths for wireless networks that are subject to interference constraints.

Disjoint path protection consists of a primary and failure-disjoint backup path, such that after the failure along the primary path, the flow will be switched to the surviving backup path [4]. Consider the wired network shown in Figure 1: Two primary paths, P_1 and P_2, are shown using solid lines, and a disjoint backup path for P_1 is shown using a dotted line, B_1. After the failure of some edge or node in P_1, flow is switched to B_1, and flow continues on P_2 unaffected.

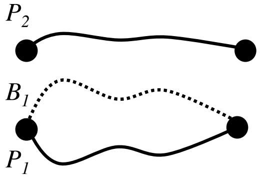

Fig. 1: Disjoint path protection in a wired network

When applying disjoint path protection to wireless networks, interference between nodes must be considered. If two nodes are within close proximity, they cannot transmit simultaneously in the same frequency channel without interfering with one another. In order to avoid interference, transmissions can be separated in time using slotted TDMA, where time is divided into non-overlapping time slots, and transmissions are then scheduled using those time slots [5]. Consider the same set of paths as in Figure 1, except now in a wireless setting. A schedule of link transmissions needs to be determined. Paths P_1 and P_2 must be scheduled so that they do not interfere with one another, and the backup path B_1 must also be scheduled so that it can operate without interference. One possible approach is to schedule B_1 so that it does not interfere with either P_1 or P_2. Such a scheme for disjoint path protection in wireless networks was suggested in [6]. While this solves the issue of interference between the backup and the primary paths, our simulations show that the number of time slots needed for such an approach more than doubles from the case without any protection, leading to a 50% loss in throughput. Other works in wireless network survivability have looked at guaranteeing connectivity after a failure [7, 8], or real-time recovery approaches where new

This work was supported by NSF grants CNS-1116209 and CNS-1017800, and by DTRA grant HDTRA-09-1-005.

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

routes are found after a failure [9, 10]. These approaches to resiliency do not consider routing and scheduling with respect to interference constraints, and assume that there exists some unspecified mechanism to find a route and schedule at any given point in time. Furthermore, there is no guarantee that sufficient capacity will be available to protect against a failure.

In our work, we use the idea of "recapturing" lost capacity that becomes available after a failure, which was first presented in [11]. When path P_1 fails, it no longer supports any flow; hence, the time slots that were assigned to it are no longer necessary to schedule those transmissions. Since the backup path B_1 will only be operational after the failure of P_1, the time slots that P_1 used can be reused on B_1. This capacity recapturing approach significantly reduces the number of time slots needed for protection, but it comes at the cost of additional complexity. If the paths P_1 and P_2 did not interfere with one another, then they could have been scheduled using the same set of time slots. While P_1 did not interfere with P_2, path B_1 might interfere with P_2, and hence, cannot reuse P_1's time slots. Alternatively, path B_1 could only partially interfere with P_2, allowing the reuse of only some of P_1's time slots. Additionally, the time slots for protection on B_1 can be possibly shared between P_1 and P_2 if the two primary paths are failure disjoint of one another. In wired networks, where there is no interference between nodes, shared backup protection was shown to be NP-hard [4]. Adding scheduling to allow interference-free communication adds further complexity. To the best of our knowledge, disjoint path protection in wireless networks subject to interference constraints has not been previously examined.

The novel contributions of our paper is a scheme that provides guaranteed protection using disjoint paths for wireless networks that is both non-disruptive and resource-efficient. We call this problem the Wireless Disjoint Path Protection (WDPP) problem. In Section II, the model and problem description for WDPP is presented. In Section III, we show that finding an optimal solution to WDPP is not only NP-hard, but is in fact NP-hard to approximate. Simulations are run comparing WDPP to the use of wired protection schemes in wireless networks, showing significant reductions in resources needed for protection. In Section IV, time-efficient algorithms are developed. Notably, for the case of 2-hop interference, which corresponds to the IEEE 802.11 standard, our protection scheme requires only 8% more resources on average than providing no protection whatsoever.

II. Model and Problem Description

In this paper, we study the problem of providing non-disruptive and resource-efficient protection by using disjoint paths for wireless networks that are subject to interference constraints. Our goal is to provide disjoint backup path protection in a manner similar to what has been done in the wired setting. Namely, after the failure of some network element, connections that fail switch to their respective backup paths, and connections that did not fail continue to use their primary paths. After any failure, connections must maintain the same amount of flow that they had before the failure. As opposed to wired networks, transmissions in wireless networks must be scheduled so communications can occur without interference. In order to guarantee minimal disruption and to provide rapid recovery, connections that were not affected by a failure will continue to use their primary paths, as well as maintain the same transmission schedule on those paths. In order to meet these requirements, resources are allocated and scheduled in advance for both the primary and backup routes to protect against failures.

The mechanism for disjoint path protection in wireless networks is as follows. Each demand will have a primary and backup path, as well as an interference-free schedule for those paths. After the failure of some network element, a demand whose primary path fails will switch to its disjoint backup path and schedule. If a demand's primary path did not fail, it will continue to use its pre-failure primary path and schedule. If a primary path does fail, the time slots that were used to schedule that path are no longer needed, and can be reused for the protection path.

Our objective is to minimize the length of the schedule needed to route and schedule all demands without interference. Minimizing the length of the schedule will allow each link to communicate for a longer period of time, raising the overall throughput [5]. We call our problem Wireless Disjoint Path Protection (WDPP). We develop a solution to WDPP for general binary interference constraints, which can be extended to SINR interference. Due to length constraints, only the binary interference model is presented in this paper; treatment of the SINR interference model can be found in the technical report [12].

The binary interference model is as follows: for any pair of links, $\{i, j\}$ and $\{k, l\}$, either both links can be active simultaneously, or at most one link can be active [5]. Binary interference is used for the K-hop interference model [13], and the protocol interference model [14]. In K-hop interference, if link $\{k, l\}$ is within K hops of link $\{i, j\}$, the two links will interfere. In the protocol model, link $\{i, j\}$ can be active only if i is within range of j, and no other nodes that are within range of j are transmitting.

The following network model is used for the remainder of the paper. We are given a graph G with a set of wireless nodes V and edges E. A set of demands $(s_i, d_i) \in D$ must be routed and scheduled, such that there will exist a primary and disjoint backup path from s_i to d_i, $\forall i$. Since we are considering wireless networks in the context of backbone and last-mile services, we assume that the wireless nodes are static. For any node, we assume that its neighbors are fixed; hence, the set of edges E is fixed. For the binary interference model, an interference matrix $\mathcal{I}$ can be defined where $I_{ij}^{kl} \in \mathcal{I}$ is 1 if links $\{i, j\}$ and $\{k, l\}$ can be activated simultaneously (do not interfere with each other), and 0 otherwise. We assume that the network uses a synchronous time slotted system, with equal length time slots, where the set of time slots used is $\mathcal{T}$, and $T = |\mathcal{T}|$.

Our objective is to minimize the total number of time slots needed to route and schedule all demands using disjoint path protection. Solutions are developed for both node and link failures, and similar to the work in wired protection, we use a single failure model, where we assume at most one failure at a time. Our work can be extended to multiple failures by considering additional disjoint paths. All transmissions share a single frequency channel.

III. Minimum Length Schedule for Wireless Disjoint Path Protection

We start by studying the minimum length schedule to route a set of demands with disjoint path protection in a network that is subject to interference constraints. We first demonstrate that finding a minimum length schedule for Wireless Disjoint Path Protection (WDPP) is NP-hard, and that the problem remains NP-hard even to approximate. Subsequently, in Section III-A an integer linear program (ILP) is formulated to find the optimal solution for the minimum length schedule. In Section III-B, a simulation of WDPP is performed, with results compared to the use of a wired disjoint path protection scheme in wireless networks.

Theorem 1. *For a set of demands requiring disjoint path protection in a wireless network subject to binary interference constraints, determining a minimum-length schedule is NP-hard, and remains NP-hard to approximate.*

To prove the NP-hardness and non-approximability of WDPP, we reduce from the problem of determining the chromatic number of a graph [15]. The proof can be found in the technical report [12].

A. Integer Linear Program for WDPP

Since it is NP-hard to find a minimum length schedule, and approximating a solution is also NP-hard, we develop an integer linear program (ILP). The conditions for both link and node-disjoint paths are given, where link-disjoint paths are guaranteed to only survive a link failure, and node-disjoint are guaranteed to survive either a link or node failure. We assume unit demands, and that the links in the network all have unit capacity. These two constraints can be easily modified.

For the ILP, the following values are given:

- $G = (V, E)$ is the graph with a set of vertices and edges
- D is the set of demands
- $\mathcal{T}$ is the set of time slots in the system, $\mathcal{T} \subset \mathbb{Z}^+$
- $\mathcal{I}$ is the interference matrix, where $I_{ij}^{kl} \in \mathcal{I}$ is equal to 1 if links $\{i, j\}$ and $\{k, l\}$ can be activated simultaneously, 0 otherwise

The ILP solves for the following variables:

- x_{ij}^{sd} is equal to 1 is the primary flow assigned for demand (s, d) on link $\{i, j\}$, 0 otherwise
- y_{ij}^{sd} is equal to 1 is the protection flow assigned on link $\{i, j\}$ for demand (s, d), 0 otherwise
- $\lambda_{ij}^{sd,t}$ is a scheduling variable for the primary flow for demand (s, d) and is equal to 1 if link $\{i, j\}$ is activated in time slot t, 0 otherwise
- $\delta_{ij,kl}^{sd,t}$ is a scheduling variable for the flow after the failure of link $\{k, l\}$ for demand (s, d), and is equal to 1 if link $\{i, j\}$ is activated in time slot t, 0 otherwise
- τ_{ij}^{t} is a scheduling variable and is equal to 1 if link $\{i, j\}$ is activated in time slot t, 0 otherwise
- $\pi_{ij,kl}^{t}$ is a scheduling variable, and is 1 if link $\{i, j\}$ is activated in time slot t after link $\{k, l\}$ fails, 0 otherwise
- s_t is equal to 1 if time slot t is used by the primary or protection flow, and 0 otherwise

The objective function is to minimize the number of time slots (the length of the schedule) needed to route all demands with disjoint path protection.

$$\textbf{Objective:} \quad \min \sum_{t \in \mathcal{T}} s_t \tag{1}$$

The following constraints are imposed to find a feasible routing and scheduling.

Before a failure:

- Find a primary path for demand (s, d) before any link failure.

$$\sum_{\{i,j\} \in E} x_{ij}^{sd} - \sum_{\{j,i\} \in E} x_{ji}^{sd} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases}, \ {}^{\forall (s,d) \in D}_{\forall i \in V} \tag{2}$$

- Ensure a link is scheduled to support the primary flow for demand (s, d) on edge $\{i, j\}$.

$$x_{ij}^{sd} \le \sum_{t \in \mathcal{T}} \lambda_{ij}^{sd,t}, \quad {}^{\forall \{i,j\} \in E}_{\forall (s,d) \in D} \tag{3}$$

- At most one demand can use edge $\{i, j\}$ during slot t.

$$\sum_{\forall (s,d) \in D} \lambda_{ij}^{sd,t} \le \tau_{ij}, \quad {}^{\forall \{i,j\} \in E}_{\forall t \in \mathcal{T}} \tag{4}$$

- Mark if slot t is used to schedule a demand before a failure.

$$\tau_{ij}^{t} \le s^{t}, \quad {}^{\forall \{i,j\} \in E}_{\forall t \in \mathcal{T}} \tag{5}$$

- Interference constraints: In a time slot, only links that do not interfere can be activated simultaneously.

$$\tau_{ij}^{t} + \tau_{uv}^{t} \le 1 + I_{uv}^{ij}, \quad {}^{\forall \{i,j\} \in E, \ \forall \{u,v\} \in E}_{\{i,j\} \ne \{u,v\}, \ \forall t \in \mathcal{T}} \tag{6}$$

After a failure:

- Find a second path for demand (s, d) to be used as the disjoint protection path.

$$\sum_{\{i,j\} \in E} y_{ij}^{sd} - \sum_{\{j,i\} \in E} y_{ji}^{sd} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases}, \ {}^{\forall (s,d) \in D}_{\forall i \in V} \tag{7}$$

- Enforce path disjointness between the primary and protection path for demand (s, d).
  - *Edge*-disjoint:

$$x_{ij}^{sd} + y_{ij}^{sd} \le 1, \quad {}^{\forall (s,d) \in D}_{\forall \{i,j\} \in E} \tag{8a}$$

- *Node*-disjoint:

$$\sum_{j \in V \setminus (s,d)} x_{ij}^{sd} + \sum_{j \in V \setminus (s,d)} y_{ij}^{sd} \leq 1, \quad \substack{\forall (s,d) \in D \\ \forall i \in V} \tag{8b}$$

- If after the failure of $\{k, l\}$, the primary path for demand (s, d) did *not* fail (i.e. edge $\{k, l\}$ was not part of the primary path), then that primary path must remain active and use the same schedule as from before the failure. In other words, if edge $\{i, j\}$ was part of the primary path, but the failed edge $\{k, l\}$ was not, then force the same time slot assignment on edge $\{i, j\}$ for after the failure of $\{k, l\}$ that $\{i, j\}$ used before the failure, i.e. $\delta_{ij,kl}^{sd,t} = 1$ if $\lambda_{ij}^{sd,t} = 1$ when $x_{ij}^{sd} = 1$ and $x_{kl}^{sd} = 0$.

$$\lambda_{ij}^{sd,t} + [(x_{ij}^{sd} - x_{kl}^{sd}) - 1] \leq \delta_{ij,kl}^{sd,t}, \quad \substack{\forall \{i,j\} \in E, \ \forall \{k,l\} \in E \\ \forall t \in \mathcal{T}, \ \forall (s,d) \in D} \tag{9}$$

- If after the failure of edge $\{k, l\}$, the primary path for demand (s, d) *did* fail (i.e. edge $\{k, l\}$ was part of the primary path), schedule the disjoint backup path.

$$y_{ij}^{sd} - (1 - x_{kl}^{sd}) \leq \sum_{t \in \mathcal{T}} \delta_{ij,kl}^{sd,t}, \quad \substack{\forall \{i,j\} \in E, \ \forall \{k,l\} \in E \\ \forall (s,d) \in D} \tag{10}$$

- At most one demand can use edge $\{i, j\}$ during slot t after the failure of $\{k, l\}$.

$$\sum_{\forall (s,d) \in D} \delta_{ij,kl}^{sd,t} \leq \pi_{ij,kl}^{t}, \quad \substack{\forall \{i,j\} \in E, \ \forall \{k,l\} \in E \\ t \in \mathcal{T}} \tag{11}$$

- Mark if slot t is used to schedule any demand's disjoint protection path after the failure of $\{k, l\}$.

$$\pi_{ij,kl}^{t} \leq s^{t}, \quad \substack{\forall \{i,j\} \in E, \ \forall \{k,l\} \in E \\ \forall t \in \mathcal{T}} \tag{12}$$

- Interference constraints: In any given time slot, after the failure of link $\{k, l\}$, only links that do not interfere with one another can be activated simultaneously.

$$\pi_{ij,kl}^{t} + \pi_{uv,kl}^{t} \leq 1 + I_{uv}^{ij}, \quad \substack{\forall \{i,j\} \in E, \ \forall \{u,v\} \in E \\ \forall \{k,l\} \in E, \ \forall t \in \mathcal{T} \\ \{i,j\} \neq \{u,v\} \neq \{k,l\}} \tag{13}$$

B. Simulation Results for WDPP

The Wireless Disjoint Path Protection scheme is compared to the traditional $1 + 1$ protection scheme used in wireless networks, as was suggested in [6]. For this scheme, a disjoint primary and backup path are identified, and a schedule is found such that the two paths do not interfere with one another. We call this approach wireless $1 + 1$. The number of time slots to route and schedule the demands without any protection is a lower bound for any solution that includes protection for the same set of demands. Hence, we compare the number of additional time slots needed for protection beyond those that were needed for the case without any protection.

Due to its complexity, an integer linear program can take a long time to run. Because of this, it may not always be possible to obtain an optimal solution, even for small networks; we found this to be the case for WDPP. The ILP developed in Section III-A jointly optimizes the schedule for before and after a failure. To allow our ILP to run in a reasonable amount of time, we separate the before and after phase, and use a two-step approach: First, we find the routes and schedules for all of the demands "before a failure", then we find the same for "after a failure". While this approach is sub-optimal, our simulations show that the routes and schedules found required only minimal additional time slots for protection beyond the solution without any protection. The "before a failure" phase is the minimum number of time slots to route and schedule the demands without any protection. The minimum number of time slots for the wireless $1+1$ scheme is found using an ILP.

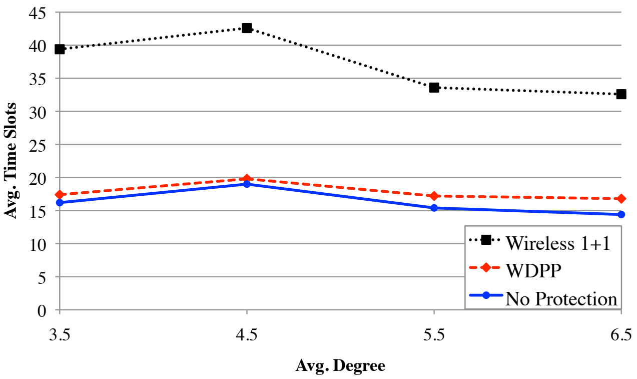

Fig. 2: Binary interference simulation results

The 2-hop binary interference model is used, which corresponds to the IEEE 802.11 standard [13]. Fifty random graphs were generated with twenty nodes each. Nodes that are within a certain transmission range of one another have a link, and the transmission range is varied to give different desired average node degrees (i.e. average number of neighbors that a node has). The node degree is varied from 3.5 to 6.5, and for each graph, twelve source/destination pairs are randomly chosen to be routed concurrently. We simulate only the edge-disjoint case. The results are plotted in Figure 2. On average, WDPP used 94% fewer time slots to provide the same level of resiliency as that of wireless 1+1. In fact, for 50% of the cases tested, WDPP needed no additional time slots beyond what was required to route and schedule the demands without protection. On average, WDPP needed only 8% more time slots beyond that of the no protection case, while wireless $1 + 1$ needed 128% additional time slots. For the most part, the same time slots that were used to schedule the primary paths can be reused to schedule the disjoint backup paths.

IV. Time-Efficient Algorithms for WDPP

In the previous section, an integer linear program was presented to find the minimum length schedule for Wireless Disjoint Path Protection (WDPP). An ILP is not a computationally efficient method of finding a solution; in fact, the ILP in Section III needed to be split into two parts to allow it to run in a reasonable amount of time. In this section, we develop a time-efficient algorithm for WDPP. As was demonstrated in Section III, an optimal solution to WDPP is NP-hard, even to approximate. To solve WDPP, we utilize a dynamic approach that will route and schedule each demand one-at-a-time, where each demand is scheduled such that it does not interfere with previously scheduled connections. In Theorem 2, we show that even when demands are routed one-at-a-time, finding the minimum number of time slots to route and schedule any individual demand is NP-hard.

Since it is NP-hard to determine the minimum number of time slots to route and schedule any individual demand, the algorithm will work in the following fashion: For each demand, a route and schedule is first found for the primary path, and then a route and schedule is found for the disjoint protection path. In Section IV-A, an algorithm to find an interference-free path is presented. This path is found as follows: We first find a path that is of low-interference (a metric that we define later), and then determine an interference-free schedule for that path. This path algorithm is then used as a subroutine to efficiently solve WDPP, which is presented in Section IV-B.

Theorem 2. *When demands are routed and scheduled one-at-a-time with disjoint path protection under binary interference constraints, the minimum number of time slots for any individual demand is NP-hard to determine when accounting for the time slots that are currently in use.*

To prove Theorem 2, a reduction from the Dynamic Shared-Path-Protected Lightpath-Provisioning Problem (DSPLP) [4] is performed. The proof can be found in the technical report [12].

A. Interference-Free Path with a Minimal Length Schedule

In this section, an algorithm is developed to find an interference-free path, which will then be used to construct the disjoint path protection algorithm in the following section. We assume connections already exist, and are scheduled using the set of $\mathcal{T}$ time slots. We desire to set up a new path from s to d. We take a two-step approach: First, find a path that is of "low-interference", and then find a minimal length schedule for this path. We call this algorithm `feasible_path`. The details for SINR interference can be found in [12].

1) Low-Interference Path: Given that a set of connections already exist using the set of $\mathcal{T}$ time slots across the set of edges E, each edge can be assigned a value according to its general "interference load", which we define to be the set of time slots that cannot be used for that particular edge. These time slots may not be available because either that edge uses them, or some interfering edge uses them. For edge $\{i, j\}$, we label the set of unavailable time slots τ_{ij}. If an edge that is heavily loaded (few available time slots) is used for a connection, then that may prevent some future connection from being able to find an interference-free path without the use of additional time slots. To find paths that are of low-interference, we assign a cost to each edge that is equal to its interference load: $c_{ij} = |\tau_{ij}|$. We then find a shortest path from s to d with respect to these edge-costs, giving preference to edges that are not heavily loaded.

We build the set τ_{ij} for edge $\{i, j\}$ in the following manner. Define the set of time slots that are currently assigned to edge $\{i, j\}$ as t_{ij}. For binary interference, label the set of edges that $\{i, j\}$ interferes with as γ_{ij}. The set of time slots not available for use on $\{i, j\}$ are the ones currently assigned to $\{i, j\}$ and to the set of edges γ_{ij}: $\tau_{ij} = t_{ij} \cup_{\{k,l\} \in \gamma_{ij}} t_{kl}$.

2) Minimal Length Schedule for a Path: Once a low-interference path has been found, we want to find a minimal length schedule for it. We construct a conflict graph G^c, which is built as follows: A node is added for each edge in the original graph, and an edge is added between two nodes in G^c if the edges associated with those nodes interfere with one another [5]. Any independent set[1] of G^c are a set of edges in the original graph that can be activated simultaneously. Any feasible coloring[2] of the nodes of G^c is a feasible schedule of link activations. Label the set of edges of the path as P. We construct G^c using only the set of edges P: Add node v_{ij} to G^c for each edge in P, and add an edge between v_{ij} and v_{kl} if edges $\{i, j\}$ and $\{k, l\}$ cannot be active simultaneously.

We wish to find a minimum node-coloring of G^c, which will be a minimum-length schedule for P. The minimum node-coloring problem is NP-hard to solve [15]. For our problem, we have a restriction that not all colors are available for all nodes: The set of colors not available for node v_{ij} is the set of time slots that edge $\{i, j\}$ cannot use: τ_{ij}. We note that this restricted node-coloring problem remains NP-hard; a valid instance of the restricted problem is to have $\tau_{ij} = \emptyset$, $\forall \{i, j\}$, which is simply the original NP-hard node-coloring problem. To find a solution, we use the Welsh-Powell algorithm that colors the nodes (assigns time slots) in a greedy fashion, starting with nodes that have highest degree [16].

B. Wireless Disjoint Path Protection

In Section IV-A, an algorithm `feasible_path` was presented that finds a path and schedule between two nodes that takes into account other scheduled connections in the network. We use `feasible_path` as a subroutine to construct an algorithm for WDPP. We label the algorithm presented in this section `WDPP_alg`. We present the algorithm for the edge-disjoint protection case, but it can be easily modified for the node-disjoint case as well. Since the subroutine `feasible_path` finds a path with respect to interference constraints, `WDPP_alg` is agnostic to the interference constraints used.

The mechanism for disjoint path protection in wireless networks is as follows. Each demand will have a primary and backup path, as well as an interference-free schedule for those paths, and after the failure of some edge, a demand whose primary path fails will switch to its disjoint backup path and schedule. To minimize network disruption after a failure, if a demand's primary path did not fail, it will continue to use its pre-failure primary path and schedule. If a primary path does fail, the time slots that were used to schedule that path are no longer needed, and can be reused for the protection path.

We consider some incoming demand between nodes s and d, with the network already having some set of scheduled connections using the set of time slots $\mathcal{T}$. As defined in Section IV-A1, τ_{ij} is the set of time slots that cannot be used to schedule edge $\{i, j\}$, which we called the "interference load". We call

[1] An independent set is a set of nodes where no two nodes are the end points of the same edge.

[2] Each node is assigned a color such that all nodes of one color form an independent set.

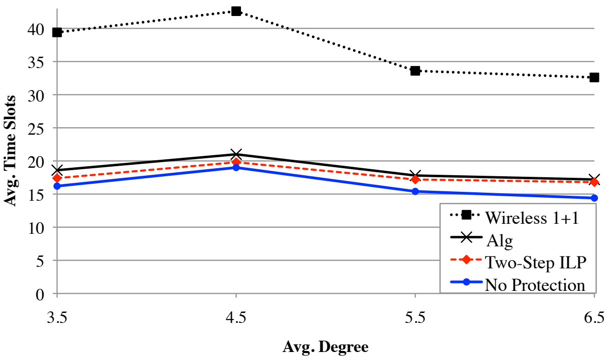

Fig. 3: Binary interference algorithm simulation

the set of interference loads for each edge the *interference set*, and we label it $\Gamma = \{\tau_{ij} | \{i,j\} \in E\}$. Because of the different sets of paths used, the interference set can be different before a failure and after any particular failure. For the existing scheduled connections, the interference set before any failure (only the primary paths) is labeled Γ, and is labeled Γ_{kl} for after the failure of edge $\{k,l\}$. The interference set Γ_{kl} reflects the schedules of all the paths that are currently used in the event of the failure of $\{k,l\}$, which includes the backup paths for demands that fail, as well as the primary paths for the demands that did not fail.

The algorithm for wireless disjoint path protection (`WDPP_alg`) is as follows. First, using the interference set Γ, a path and its corresponding schedule is found between s and d using `feasible_path`. This will be the primary path, and we label its set of edges as P. Next, we find the disjoint backup path. We construct a new graph G^F that does not have the set of edges P; any path between s and d in G^F will be disjoint to P. We consider the possible failure of any edge in the primary path. Upon the failure of edge $\{k,l\} \in P$, demands that did not fail must continue to use their pre-failure path and schedule, and demands that did fail switch to their backup path and schedule. After the failure of an edge in P, the edges of that path no longer supports any flow, and the time slots used on those edges become available for protection. We form a new interference set that contains the information of all the possible paths used after the failure of any edge in the primary path: $\Gamma^F = \cup_{\{k,l\} \in P} \Gamma_{kl}$. Since Γ^F does not contain any scheduling information regarding P, the time slots used to schedule P can be reused to schedule the disjoint backup path. Using graph G^F and interference set Γ^F, a path is found between s and d using `feasible_path`, which is the disjoint backup path.

To demonstrate the performance of `WDPP_alg`, we simulate the algorithm using the same parameters as the simulation for the ILP in Section III. For `WDPP_alg`, the demands are randomly ordered, and a route and schedule is found for each demand one-at-a-time. The algorithm is compared to the wireless $1+1$ scheme and the two-step ILP, both of which were described in Section III. Figure 3 shows the simulation results for binary interference. On average, `WDPP_alg` performed within 4% of the two-step ILP, and required 88% fewer protection time slots than wireless $1+1$.

V. Conclusion

In this paper, the problem of Wireless Disjoint Path Protection (WDPP) for networks subject to interference constraints was examined. The motivation is to provide protection that is similar to that of wired networks, but designed for wireless networks. Our protection scheme takes advantage of the interference in wireless networks for greater resource efficiency: Resources that are freed after a failure in the network can be reused for protection from that failure. We demonstrated that WDPP is NP-hard, even to approximate. We formulated an ILP, giving solutions using 87% fewer protection resources on average than the wired disjoint path scheme in wireless networks. For the case of 2-hop interference, WDPP only 8% more resources on average than providing no protection whatsoever. We then developed a time-efficient algorithm that performed almost as well as the two-step ILP on average. A future direction is to adapt our work for a distributed setting.

References

[1] A. Saleh and J. Simmons, “Evolution Toward the Next-Generation Core Optical Network,” *Journal of Lightwave Technology*, vol. 24, 2006.

[2] M. D. Williams, *Broadband for Africa: developing backbone communications networks*. World Bank Publications, 2010.

[3] J. P. Sterbenz, R. Krishnan, R. R. Hain, A. W. Jackson, D. Levin, R. Ramanathan, and J. Zao, “Survivable mobile wireless networks: issues, challenges, and research directions,” in *Proceedings of the 1st ACM workshop on Wireless security*. ACM, 2002, pp. 31–40.

[4] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, “New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks,” *Journal of Lightwave Technology*, vol. 22, no. 5, 2004.

[5] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, “Impact of interference on multi-hop wireless network performance,” *Wireless networks*, vol. 11, no. 4, pp. 471–487, 2005.

[6] M. Kodialam and T. Nandagopal, “Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem,” in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM, 2003, pp. 42–54.

[7] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, “Highly-resilient, energy-efficient multipath routing in wireless sensor networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[8] N. Li and J. C. Hou, “Flss: a fault-tolerant topology control algorithm for wireless networks,” in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 275–286.

[9] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, “An adaptive management architecture for ad hoc networks,” *Communications Magazine, IEEE*, vol. 41, no. 2, pp. 108–115, 2003.

[10] A. Raniwala and T.-c. Chiueh, “Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network,” in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. IEEE, 2005, pp. 2223–2234.

[11] G. Kuperman and E. Modiano, “Providing protection in multi-hop wireless networks.” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013.

[12] ——, “Disjoint Path Protection in Multi-Hop Wireless Networks with Interference Constraints,” MIT, Tech. Rep., 2014. [Online]. Available: http://web.mit.edu/gregk/www/papers/DPPMHWNIC.pdf

[13] G. Sharma, R. Mazumdar, and N. Shroff, “On the complexity of scheduling in wireless networks,” in *Proceedings of the 12th annual int’l conf. on Mobile computing and networking*. ACM, 2006, pp. 227–238.

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

[15] D. Zuckerman, “Linear degree extractors and the inapproximability of max clique and chromatic number,” in *Thirty-eighth annual ACM symposium on Theory of computing*. ACM, 2006, pp. 681–690.

[16] D. J. Welsh and M. B. Powell, “An upper bound for the chromatic number of a graph and its application to timetabling problems,” *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.