Analysis and Algorithms for Partial Protection in Mesh Networks

Greg Kuperman, Eytan Modiano, and Aradhana Narula-Tam

***Abstract*—This paper develops a novel mesh network protection scheme that guarantees a quantifiable minimum grade of service upon a failure within the network using multipath routing. Typically, networks fully guarantee service after a single-link failure, which is often an over-provisioning of resources to maintain essential traffic for the infrequent event of a failure. Our scheme guarantees that a fraction q of each demand remains after any single-link failure, at a fraction of the price of full protection. A linear program is developed to find the minimum-cost capacity allocation to meet both demand and protection requirements. For $q \leq \frac{1}{2}$, an exact algorithmic solution for the minimum-cost routing and capacity allocation is developed using multiple shortest paths. For $q > \frac{1}{2}$, an algorithm is developed based on disjoint path routing that performs, on average, within 1.4% of optimal, and runs four orders of magnitude faster than the minimum-cost solution achieved via the linear program. Moreover, the partial protection strategies developed achieve reductions of up to 83% over traditional full protection schemes.**

***Index Terms*—Multipath routing; Networks; Network survivability; Network protection and restoration.**

I. Introduction

Mesh networks with ever-increasing data rates are being deployed to meet the increasing demands of the telecom industry. As data rates continue to rise, the failure of a network line element or worse, a fiber cut, can result in severe service disruptions and large data loss, potentially causing millions of dollars in lost revenue [1]. Currently, there exist few options for protection that offer less than complete restoration after a failure. Due to the cost of providing full protection, service providers may offer lower tiers of protection that are best effort, and offer no guarantees on fully restoring a connection [2]. Additionally, the time that any given connection is in a failed state due to a fiber cut is relatively small [2,3]; thus service providers may wish to only support essential traffic after a network failure. By defining varying and quantifiable grades of protection, service providers can protect vital services without incurring the cost of providing full protection, making protection more affordable and better suited to user/application requirements. The protection scheme developed in this paper provides “partial protection” guarantees, at a fraction of the cost of full protection, with each session having its own differentiated protection guarantee.

Guaranteed network protection has been studied extensively [4–9]. The most common in backbone networks today is guaranteed path protection [10], which provides an edge-disjoint backup path for each primary path, resulting in 100% service restoration after any link failure. Best effort protection is still loosely defined, but generally offers no *guarantees* on the amount of protection provided. In best effort protection, a service will be protected, if possible, with any unused capacity after fully protecting all guaranteed services [2,11]. Best effort protection can also be referred to as partial capacity restoration, since a service will be restored within existing unused capacity, typically resulting in less than 100% restoration.

Many users may be willing to tolerate short periods of reduced capacity to protect only essential services if data rate guarantees can be made at a reduced cost. In this paper, we consider an alternate form of guaranteed protection, where a fraction of a demand is guaranteed in the event of a link failure. If provided at a reduced cost, many users may opt for partial protection guarantees during network outages.

A quantitative framework for deterministic partial protection in optical networks was first developed in [12]. In that work, a minimum fraction q of the demand is guaranteed to remain available between the source and destination after any single-link failure, where q is between 0 and 1. When q is equal to 1, the service is fully protected, and when q is 0, the service is unprotected. Partial protection has been considered in a number of areas, with a set of proposed algorithms and heuristics: [13] considers partial protection to protect high-definition video with regular-definition service, and [14] provides a comparison of proposed implementations for IP-over-WDM networks. In [15], the partial protection problem on groomed optical WDM networks is studied, under the assumption that flows must traverse a single path. More recently, [16] shows that the amount of partial protection that can be guaranteed depends on the topology of the network. The work in [16] develops algorithms for partial protection across disjoint paths. The main purpose of our paper is to provide a quantitative framework for how to make optimal allocation decisions across disjoint paths, in order to provide protection

Manuscript received July 23, 2013; revised June 7, 2014; accepted June 14, 2014; published July 31, 2014 (Doc. ID 194297).

G. Kuperman (e-mail: gregk@mit.edu) is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA.

E. Modiano is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA.

A. Narula-Tam is with Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts 02420, USA.

http://dx.doi.org/10.1364/JOCN.6.000730

guarantees. This paper is an extended version of our preliminary work on this topic [17,18]. Ideas developed in our preliminary works have recently been extended to orthogonal frequency-division multiplexing optical networks [19].

In this paper, we build upon the initial framework developed in [12] and develop a "theory" for partial protection using multipath routing. We develop optimal algorithms for capacity allocation and explicit expressions for the amount of required additional backup capacity. Routing strategies that allocate working and backup capacity to meet partial protection requirements are derived. Similar to [16], flow bifurcation over multiple paths is allowed (multipath routing). Bifurcation reduces the amount of additional backup capacity needed to support the protection requirements. In fact, we show that depending on the value of q, it may be possible to provide protection without any additional backup capacity at all.

A linear program is developed to find the optimal minimum-cost capacity allocation needed to guarantee partial protection in the event of a link failure. Without backup capacity sharing, a routing and capacity assignment strategy based on shortest paths is shown to be optimal for $q \le \frac{1}{2}$. For $q > \frac{1}{2}$, an efficient algorithm based on disjoint path routing is shown to have a cost that is at most twice the optimal minimum-cost solution, and in practice only slightly above optimal. We also consider the case in which backup capacity sharing is possible. With backup capacity sharing, demands may share protection resources if at most one demand will use those resources at a time. For the backup sharing case, we show that depending on the value of q, it may be possible to provide protection at minimal allocation cost, i.e., the shortest path routing. We consider two cases for backup capacity sharing: preemptive and nonpreemptive partial protection. For the preemptive case, primary resources available prior to a link failure may be preempted to provide backup for other demands, as long as all protection requirements are met after the failure. For the nonpreemptive case, only demands that are directly affected by the link failure drop to the rates guaranteed under partial protection.

In Section II, the partial protection model is described. In Section III, the partial protection problem is formulated as a linear program with the objective of finding the minimum-cost allocation of primary and backup capacity. In Section IV, solutions for partial protection without the use of backup capacity sharing are developed, including a simple path-based routing for an optimal solution when $q \le \frac{1}{2}$, and when $q > \frac{1}{2}$, properties of an optimal solution for a network of disjoint paths are determined and used to develop a time-efficient algorithm. In Section V, backup capacity sharing is considered, and an algorithm is developed for the case of routing demands one at a time upon their arrival.

II. Partial Protection Model

The objective of partial protection is to find an allocation that ensures that enough capacity exists to support the full demand before a link failure and a fraction q of that demand

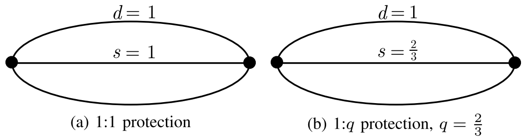

Fig. 1. Standard protection schemes.

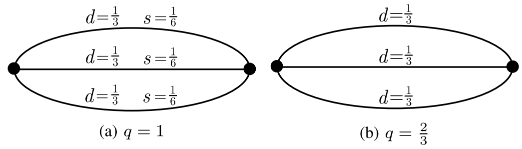

Fig. 2. Protection using risk distribution.

afterward. We assume that the graph G, with a set of vertices V and edges E, is at least two-connected. Each link has a fixed cost of use: c_{ij} for each edge $\{i,j\} \in E$. We consider only single-link failures. We consider both the cases when backup capacity can and cannot be shared. With backup capacity sharing, demands can share protection resources if at most one demand will use those resources at a time. Under the single-link failure model, if two primary flows are disjoint, then at most one can fail at a time, and at most one will need to use its backup resources. Both primary traffic and protection flows (defined as the flow after a failure) can be bifurcated to traverse multiple paths between the source and destination, which is often referred to as multipath routing [20]. Without loss of generality, we assume unit demands, unless noted otherwise.

We now present a motivating example. We assume that link costs are all 1 (in the following section we consider nonuniform link costs), and with uniform link costs, the objective is to minimize the total capacity needed to support the flow and the partial protection requirements.

One routing strategy for providing backup capacity is to use a single primary path and a single backup path similar to the 1:1 guaranteed path protection scheme. Consider the network shown in Fig. 1. With 1:1 protection, one unit of capacity is routed on a primary path and one unit of capacity on a backup [Fig. 1(a)]. Upon a link failure, 100% of the service can be restored via the backup path. Now, consider a partial protection requirement to provide a fraction $q = \frac{2}{3}$ of backup capacity in the event of a link failure. A simple protection scheme similar to 1:1 protection would be to route one unit along the primary path and $\frac{2}{3}$ along a disjoint protection path, as shown in Fig. 1(b). We will refer to this protection scheme as 1:q protection. More formally, 1:q protection routes the full demand along a primary path, and reserves sufficient capacity for at least a fraction q of the full demand on a failure-disjoint backup path. After the failure of the primary path, at least a fraction q of the original demand will be available between the source and destination. In the example from Fig. 1(b), if the primary path fails, sufficient backup capacity remains to provide service for $\frac{2}{3}$ of the demand.

For both partial and full protection requirements, in many cases capacity savings can be achieved if the risk is distributed by spreading the primary allocation across multiple paths. For example, by spreading the primary allocation across the three available paths, as shown in Fig. 2(a), any single-link failure results in a loss of at most $\frac{1}{3}$ of the demand. To fully protect this demand against any single-link failure (i.e., $q = 1$), additional spare capacity allocation[1] of $s = \frac{1}{6}$ needs to be added to each link. With this strategy, a total of 1.5 units of capacity are required, as opposed to the total of 2 units needed by 1:1 protection. If instead the protection requirement was $q = \frac{2}{3}$, no spare allocation is needed since after any failure $\frac{2}{3}$ units are guaranteed to remain. By spreading the primary and backup allocation across the multiple paths between the source and destination, the risk is effectively distributed and the fraction of primary allocation lost by a link failure is reduced.

III. Minimum-Cost Partial Protection

In this section, a linear program is developed to achieve an optimal minimum-cost solution to the partial protection problem. The objective of the linear program is to find a minimum-cost routing strategy to meet demand d and partial protection requirement q for a set of demands. In particular, a demand's full flow requirement must be routed before any failure, and in the event of any link failure, a fraction q of that flow must remain. Backup capacity sharing is utilized to further reduce the capacity allocation (and cost) needed to meet demand and protection requirements. If two demands' primary paths are edge disjoint, then under a single-link failure model, only one demand can fail at a time. Hence, backup capacity can be shared between the two since at most one demand will need to use it at any given point in time. The linear program to solve for the optimal routing strategy, denoted LP_{PP}, is defined below. We start by considering the case in which only primary demands that are directly affected by a failure are switched to their respective protection flows (no preemption). Afterward, the linear program is modified to allow for all demands' primary capacity to be preempted after a failure to route protection flows, so long as all demands have their protection requirements met.

A. Linear Program to Meet Partial Protection: LP_{PP}

The following values are given:

- $G = (V, E, C)$ is the graph with its set of vertices, edges, and costs.
- d^{st} is the total demand between nodes s and t.
- q^{st} is the fraction of the demand between s and t that must be supported in the event of a link failure.
- c_{ij} is the cost of link $\{i,j\}$.

[1]We define spare capacity allocation to be the capacity that must be allocated in addition to the necessary capacity used to support the primary demand before a link failure.

The LP solves for the following variables:

- x_{ij}^{st} is the primary flow on link $\{i,j\}$ for demand (s,t), $x_{ij}^{st} \geq 0$.
- $f_{ij,kl}^{st}$ is the protection flow on link $\{i,j\}$ after the failure of link $\{k,l\}$ for demand (s,t), $f_{ij,kl}^{st} \geq 0$.
- $y_{ij,kl}^{st}$ is the spare capacity for demand (s,t) on link $\{i,j\}$ for failure of link $\{k,l\}$, $y_{ij,kl}^{st} \geq 0$.
- w_{ij} is the total primary flow on link $\{i,j\}$, $w_{ij} \geq 0$.
- s_{ij} is the total spare allocation on link $\{i,j\}$, $s_{ij} \geq 0$.

The objective of LP_{PP} is to minimize the cost of allocation over all links:

$$\min \sum_{\{i,j\}\in E} c_{ij}(w_{ij} + s_{ij}), \tag{1}$$

subject to the following constraints:

- Flow conservation constraints for primary flow: route primary traffic to meet the set of demands:

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

- Partial protection constraint: route flow to meet partial protection requirement q^{st} after failure of link $\{k,l\}$:

$$\sum_{\substack{\{i,j\}\in E \\ \{i,j\}\neq\{k,l\}}} f_{ij,kl}^{st} - \sum_{\substack{\{j,i\}\in E \\ \{j,i\}\neq\{k,l\}}} f_{ji,kl}^{st} = \begin{cases} d^{st}q^{st} & \text{if } i = s \\ -d^{st}q^{st} & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}, \quad \forall\, i \in V, \quad \forall\, \{k,l\} \in E, \quad \forall\, (s,t) \in (V,V). \tag{3}$$

- Primary capacity on link $\{i,j\}$ must meet all primary flows before a link failure

$$\sum_{(s,t)\in(V,V)} x_{ij}^{st} = w_{ij}, \quad \forall\, \{i,j\} \in E. \tag{4}$$

- Primary and spare capacity on link $\{i,j\}$ for each demand meets partial protection requirements after failure of link $\{k,l\}$:

$$f_{ij,kl}^{st} \leq x_{ij}^{st} + y_{ij,kl}^{st}, \qquad \begin{matrix} \forall\, \{i,j\} \in E, \quad \forall\, \{k,l\} \in E \\ \forall\, (s,t) \in (V,V) \end{matrix}. \tag{5}$$

- Spare capacity on link $\{i,j\}$ satisfies all protection flows after failure of link $\{k,l\}$:

$$\sum_{(s,t)\in(V,V)} y_{ij,kl}^{st} \leq s_{ij}, \qquad \begin{matrix} \forall\, \{i,j\} \in E \\ \forall\, \{k,l\} \in E \end{matrix}. \tag{6}$$

A minimum-cost solution will provide flows to meet all primary demands before a link failure and flows to meet their respective partial protection requirements after any single-link failure. Protection capacity sharing is captured in constraint (6): for all demands that use link $\{i,j\}$ for protection after the failure of link $\{k,l\}$, enough spare capacity

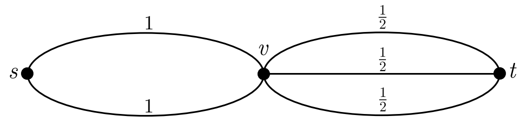

Fig. 3. Example of flow not being conserved at node v.

is allocated in addition to those demands' primary capacity to meet protection flow requirements. The spare capacity allocated to link $\{i,j\}$ will be the maximum needed for all possible link failures and will be shared amongst all the demands. To allow for preemption, constraints (5) and (6) are replaced by constraint (7).

- After failure of link $\{k,l\}$, all protection flows that use link $\{i,j\}$ can use any available primary and spare allocation:

$$\sum_{(s,t)\in(V,V)} f_{ij,kl}^{st} \le w_{ij} + s_{ij}, \qquad \begin{array}{l} \forall\, \{i,j\} \in E \\ \forall\, \{k,l\} \in E \end{array}. \tag{7}$$

With bifurcation, each of the flows may be routed over multiple paths. An interesting characteristic of the optimal solution given by the linear program is that, at each node, flow conservation for the primary flow is maintained, but the total allocation for primary plus spare capacity, given by $(w_{ij} + s_{ij})$ for edge $\{i,j\}$, does not necessarily maintain flow conservation. Consider the example demonstrated in Fig. 3. For $q = 1$ between s and t, each of the two links between nodes s and v will need one unit of allocation, and each of the links between nodes v and t will need $\frac{1}{2}$ unit of allocation. It is easily verified that after any link failure, one unit of flow will always remain between s and t. However, at node v, there is a total of two units of flow going in and 1.5 units going out. Prior to a link failure, the primary path between s and t will use one edge between s and v, and between v and t, two links will be used, each with a capacity allocation of $\frac{1}{2}$. After a link failure, similar allocations will be used to maintain full flow. Hence the total flow to support the demand before and after the link failure is conserved; however, the capacity used to achieve this flow is not conserved at v.

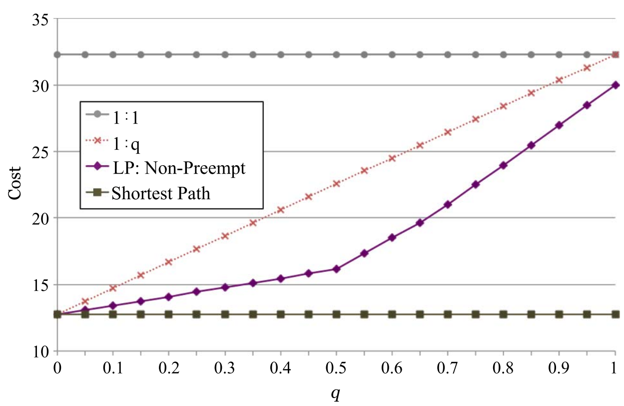

Fig. 4. Without backup capacity sharing: capacity cost versus q.

B. Comparison to Standard Protection Schemes

To compare the optimal solution to alternative protection schemes, two simulations are run: one in which backup capacity sharing is not allowed, and one in which it is. For the case without backup capacity sharing, 1000 random graph topologies are generated, each containing 50 nodes with an average node degree of 3.1, and having random link costs. Two nodes are randomly chosen from each graph to be the source and destination. The minimum-cost partial protection routing, as found by LP_{PP}, is compared to the standard scheme of 1:1 protection, as well as 1:q protection. By not allowing flow to bifurcate, i.e., $x_{ij}^{st} \in \{0,1\}$, $\forall\, \{i,j\} \in E$, the resulting scheme would be 1:q protection (and hence is now a mixed integer linear program). The linear programs are solved by using the CPLEX solver. Suurballe and Tarjan's algorithm [21] for the shortest pair of disjoint paths is used to solve for 1:1 protection.

The average cost to route the demand and protection capacity using the different routing strategies is plotted in Fig. 4 as a function of q. The top line, showing capacity requirements under 1:1 protection, remains constant for all values of q. The next two lines from the top are 1:q and LP_{PP}, respectively. As expected, both meet demand and protection requirements use fewer resources than 1:1; however, the minimum-cost solution produced by the partial protection linear program that allows flow to bifurcate uses significantly less capacity allocation. A lower bound on the capacity requirement is the shortest path routing, which provides no protection (shown in the bottom line of the figure). The cost of providing partial protection q is the difference between the cost of the respective protection strategies and the shortest path routing. Our partial protection scheme achieves reductions in excess resources of 82% at $q = \frac{1}{2}$ to 12% at $q = 1$ over 1:1 protection, and 65% at $q = \frac{1}{2}$ to 12% at $q = 1$ over 1:q protection. Additionally, we note that only two disjoint paths were typically used for a given source/destination pair. Additional disjoint paths, if available, are longer than the initial pair of disjoint paths, and were often too costly to use.

For the case in which backup capacity sharing is allowed, we compare both preemptive and nonpreemptive partial protection with the 1:1 and 1:q protection schemes, which now allow for backup capacity sharing. Because of the difficulty of jointly optimizing all demands when backup capacity can be shared, we consider the case of routing the demands one at a time upon their arrival. A similar method was used in [7,8,22] for the case of backup resource

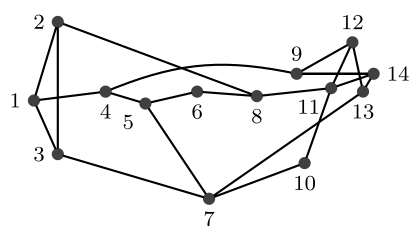

Fig. 5. 14-node NSFNET backbone network.

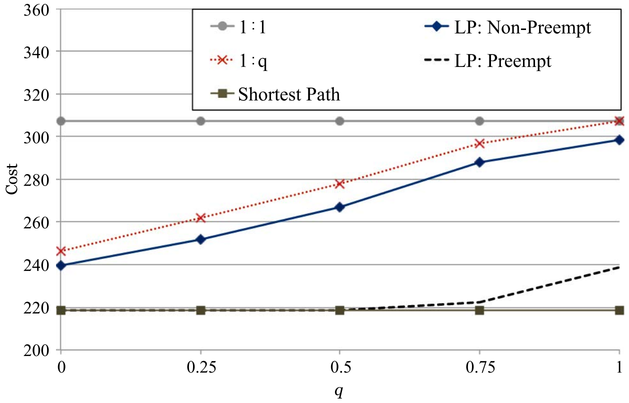

Fig. 6. With backup capacity sharing: capacity cost versus q.

sharing with 1:1 guaranteed protection. In our tests, we assume that once a connection is established, it remains active for the length of the simulation. To compare the various backup capacity schemes, we run a simulation using 100 random unit demands on a given network. It would be infeasible to run such a test on 1000 random graphs, as was done in the case in which protection resources were not shared. Instead of choosing a random topology, we run the simulation on the well-known NSFNET topology, as seen in Fig. 5.

The protection requirement, q, for each demand has a truncated normal distribution with standard deviation $\sigma = \frac{1}{2}$. The mean of q is varied between 0 and 1 for each iteration. The average costs to route the demand and protection capacity using the different routing strategies are plotted in Fig. 6 as a function of the expected value of q. Once again, the shortest path routing without protection considerations is used as a lower bound for the allocation cost. In this simulation, preemptive partial protection is able to meet requirements using only the capacity needed for the shortest path routing for $q \le \frac{1}{2}$, and only an additional increase in total capacity of 2% for $q \le \frac{3}{4}$. When considering savings in excess resources, preemptive partial protection achieves reductions of 83% at $q = 1$ over both 1:1 protection and 1:q protection, which are the same at $q = 1$. Nonpreemptive shared partial protection, at $q = \frac{1}{2}$, achieves reductions in excess resources of 59% over 1:1 shared protection and 19% over 1:q shared protection. An interesting observation is that 1:q and the linear program without preemption are almost parallel. A possible explanation for why this may be is that the LP without preemption, as noted above, will often use two disjoint paths for a given source/destination pair, with the flow split evenly amongst the two paths. For the case of 1:q with backup capacity sharing, the solution also uses two disjoint paths, with the backup path being shared amongst failure-disjoint primary paths.

IV. Solutions Without Backup Capacity Sharing

In this section, we provide insights on the structure of the solution to the minimum-cost partial protection problem when backup capacity sharing cannot be utilized. In Subsection IV.A, we are able to derive an exact algorithmic solution to the partial protection problem for $q \le \frac{1}{2}$, which runs in polynomial time using a simple series of shortest paths. When $q > \frac{1}{2}$, we analyze solutions for simpler two-node networks in Subsection IV.B. Using these insights for a two-node network for $q > \frac{1}{2}$, combined with the exact solution for $q \le \frac{1}{2}$, a time-efficient algorithm is developed in Subsection IV.C for general mesh networks. In Section V, the case in which backup capacity sharing is allowed is considered.

A. Solution for $q \le \frac{1}{2}$

As mentioned in Section II, the total primary and spare allocation coming in and out of any given node for an optimal solution does not necessarily maintain flow conservation. Without this property, most network flow algorithms do not apply [23] and analysis of the linear program becomes difficult. We show that all minimum-cost solutions for $q \le \frac{1}{2}$ will never need spare allocation, hence allowing us to formulate the partial protection problem using standard network flow conservation constraints. This then allows us to derive a simple path-based algorithmic solution. All proofs for this section are provided in Appendix A.

We begin by demonstrating that spare capacity is never needed for an optimal solution if the primary capacity on an edge is less than or equal to $(1-q)$. Hence, any time a link fails, at least q remains in the network.

Lemma 1. *No spare capacity is needed to satisfy the flow and protection requirements if and only if the primary capacity on each link is less than or equal to* $(1-q)$.

In Subsection IV.B, we show routings with zero spare allocation are not necessarily lowest-cost for all values of q. However, Lemma 2 shows that when $q \le \frac{1}{2}$, the minimum-cost solution will never use spare allocation.

Lemma 2. *Given a demand between nodes s and t with a protection requirement of* $q \le \frac{1}{2}$, *all minimum-cost solutions have no spare capacity on any edge:* $s_{ij} = 0,\ \forall \{i,j\} \in E$.

Combining Lemmas 1 and 2, it can be seen that a minimum-cost solution exists that does not use any spare allocation for $q \le \frac{1}{2}$, and that $x_{ij} \le (1-q),\ \forall \{i,j\} \in E$. Since the problem can now be formulated for $q \le \frac{1}{2}$ using no spare allocation, flow conservation at each node is preserved. The linear program can now be written using a standard flow formulation without the use of spare allocation. The modified linear program, referred to as $\text{LP}_{q\le .5}$, routes the flows on the paths in a manner that minimizes total cost and ensures that no edge carries more than $(1-q)$ of flow:

$$\text{LP}_{q\le .5}\text{: min} \sum_{\{i,j\}\in E} c_{ij} x_{ij}, \tag{8}$$

$$\sum_{\{i,j\}\in E} x_{ij} - \sum_{\{j,i\}\in E} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}, \quad \forall\, i \in V, \tag{9}$$

$$x_{ij} \le (1-q), \quad \forall\, \{i,j\} \in E. \tag{10}$$

The above linear program achieves a minimum-cost routing in a network by using only primary allocation to meet the demand. $\mathrm{LP}_{q\leq .5}$ is a network flow problem with directed and capacitated edges, which is recognized as a minimum-cost flow problem [23], for which algorithmic methods exist for finding an optimal solution. In Theorem 1, we show that an optimal solution for $q \leq \frac{1}{2}$ uses at most three paths with allocation q on each of the shortest pairs of disjoint paths and allocation $(1-2q)$ on the shortest path. Consider a directed graph $G=(V,E)$ with a source s and destination t. Let p_0 be the cost of the shortest path, p_1 and p_2 be the costs of the two shortest pairs of disjoint paths, f_0 be the flow on the shortest path, f_1 and f_2 be the flows on each of the two shortest pairs of disjoint paths, respectively, and $\mathcal{T}_{st}(q)$ be the cost of the allocation needed to meet demand and protection requirements between s and t for a value of q.[2]

Theorem 1. *Given a source s and destination t in a two-connected directed network $G=(V,E)$ with $q \leq \frac{1}{2}$, there exists a minimum-cost solution meeting primary and partial protection requirements with $f_0=(1-2q)$ and $f_1=f_2=q$, giving a total cost $\mathcal{T}_{st}(q)=(1-2q)p_0+q(p_1+p_2)$, where path 0 is the shortest path and paths 1 and 2 are the shortest pairs of disjoint paths.*

B. Solutions for $q > \frac{1}{2}$

When $q \leq \frac{1}{2}$, no spare allocation is needed when spare capacity cannot be shared, and the minimum-cost routing to meet the demand and protection requirements can be found using a series of shortest paths. When $q > \frac{1}{2}$, it may be necessary to use spare allocation to meet all requirements. Since the overall allocation of primary plus spare capacity does not necessarily meet flow conservation at any particular node, it may not be possible to provide a simple flow-based description of the optimal solution, as was done when $q \leq \frac{1}{2}$.

If we consider N disjoint paths between the source and destination, with the ith path having cost p_i, we see that this is equivalent to a two-node network with N links where the ith link has cost p_i. Hence, we investigate the properties of minimum-cost solutions for two-node networks in order to gain insight on solutions for general networks. These insights are then extended to develop a time-efficient algorithm for general mesh networks in Subsection IV.C.

A two-node network is defined as having a source and destination node with N links between them. Each link has a fixed cost of use, c_i. We first note that a solution that uses no spare allocation is not necessarily a minimum-cost allocation when unequal link costs are considered. Consider the example in Fig. 7 and let $q=\frac{2}{3}$. Allocating a capacity of $\frac{1}{3}$ onto each link does not use any spare capacity and has a total cost of $\frac{1}{3}(1+2+6)=3$. In contrast, consider using the two lowest-cost links with the addition of spare

[2]It is possible that the shortest path is also one of the shortest pairs of disjoint paths.

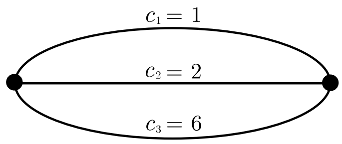

Fig. 7. Two-node network with link costs.

capacity, with each link having an allocation of $\frac{2}{3}$. The protection requirement is met, and the total cost is reduced to $\frac{2}{3}(1+2)=2$, which is less than the cost of the allocation that uses zero spare capacity.

For two-node networks, we order the edges such that $c_1 \leq c_2 \leq \cdots \leq c_N$. Define x_i as the allocation on the ith edge. We note that if $M \leq N$ edges are used for a minimum-cost allocation satisfying requirements in a two-node network, then the M lowest-cost edges are the ones that are used (otherwise, the edge allocations could always be rearranged to produce a lower-cost solution). From our analysis, we are able to define a value K, which will be important for evaluating two-node networks: $K=\mathrm{argmax}_{K=2,\ldots,N}(c_K \leq \frac{1}{K-1}\sum_{i=1}^{K} c_i)$ (demonstrated in the proof for Lemma 4). K is the maximum number of links such that the incremental cost of using an additional link would not improve the solution. We now present the minimum-cost capacity allocation for a two-node network. All proofs for this section are provided in Appendix B.

We develop results on the condition in which spare capacity is needed, which edges are active (i.e., have nonzero allocation) in the minimum-cost solution, and what the capacity allocation is across that set of active edges. Recall that spare capacity is the capacity that is allocated in addition to the capacity needed to route the primary demand.

Lemma 3. *A minimum-cost allocation for a two-node network uses spare capacity allocation if and only if $q > \frac{K-1}{K}$.*

When spare capacity is needed for a minimum-cost solution (i.e., $q > \frac{K-1}{K}$), *exactly* the K lowest-cost edges will be active, which is demonstrated in Lemma 4.

Lemma 4. *When spare allocation is needed, the minimum-cost solution for a two-node network uses exactly the K lowest-cost edges, where $K=\mathrm{argmax}_{K=2,\ldots,N}(c_K \leq \frac{1}{K-1}\sum_{i=1}^{K} c_i)$.*

An interesting result that can be seen from Lemma 4 is that the number of edges used in an optimal solution when spare allocation is needed has no dependence on the partial protection requirement q. This is stated formally in Corollary 1.

Corollary 1. *The set of K edges that are used in a minimum-cost solution when spare capacity is needed is independent of the partial protection requirement q.*

Next, we demonstrate in Lemma 5 that when spare capacity is needed, an even allocation across the K lowest-cost edges is optimal.

Lemma 5. *A minimum-cost allocation when spare capacity is needed will be an even allocation of $q\frac{1}{K-1}$ on the K lowest-cost edges, and no allocation on the remaining edges.*

Lemmas 4 and 5 both assume that spare allocation is needed for a minimum-cost solution to meet demand and protection requirements. We now show that when a solution does not use spare allocation, at most the K lowest-cost edges will be used, where the K edges are those used in a solution for when spare allocation is required.

Lemma 6. *When spare allocation is not needed, a minimum-cost solution will use at most the K lowest-cost edges, where K is the number of edges used when spare allocation is needed.*

When $q > \frac{K-1}{K}$, spare capacity is needed, and an even distribution across K edges meeting the above conditions is the optimal minimum-cost solution. When spare allocation is not needed, an even capacity allocation across the edges is no longer necessarily the optimal solution. When $q \le \frac{1}{2}$, the optimal solution is given by Theorem 1. We now present the optimal solution for when $\frac{1}{2} < q < \frac{K-1}{K}$, which is the case in which no spare capacity is needed.

Theorem 2. *The minimum-cost allocation when* $\frac{1}{2} < q \le \frac{K-1}{K}$ *will be nonzero allocation on edges* 1 *to* J*, where* J *is the integer satisfying* $\frac{J-2}{J-1} < q \le \frac{J-1}{J}$*. Moreover, the minimum-cost allocation when* $q \le \frac{K-1}{K}$ *is* $x_i = (1-q), \forall i = 1, \ldots, (J-1)$*;* $x_J = (J-1)q-(J-2)$*;* $x_i = 0, \forall i = (J+1), \ldots, N$.

C. Time-Efficient Heuristic Algorithm

Consider a mesh network with N disjoint paths between the source and destination, and let p_i be the cost of the ith path. By treating these N disjoint paths as a two-node network with N links, the results from Subsection IV.B can be applied to develop a time-efficient algorithm for general mesh networks for the case of $q > \frac{1}{2}$. Recall that for $q \le \frac{1}{2}$, the optimal minimum-cost solution for general mesh networks was derived in Subsection IV.A.

The algorithm is based on finding the k-shortest edge-disjoint paths for $k = 2$ to $k = N$, where N is the maximum number of edge-disjoint paths and the length of each path is its cost. The set of shortest disjoint paths can be found using Suurballe and Tarjan's algorithm [21]. For each set of k disjoint paths, we look to see if spare allocation is needed, i.e., $q > \frac{K-1}{K}$, and use the minimum-cost allocation given by Lemma 5 and Theorem 2. From the different possible disjoint path routings (from $k = 2$ to $k = N$ disjoint paths, where N is the maximum number of disjoint paths available), the allocation of minimum cost is chosen. We call this algorithm the partial protection disjoint path routing algorithm (PP-DPRA). Theorem 3 gives a bound on PP-DPRA's performance.

Theorem 3. *PP-DPRA produces a routing meeting demand and protection requirements with a cost that is at most twice the optimal minimum cost.*

Proof. The cost to allocate capacity for $q = \frac{1}{2}$ is given by Theorem 1 as $\frac{1}{2}(p_1 + p_2)$, where p_1 and p_2 are the costs of each of the shortest pairs of disjoint paths. Doubling the allocation on each of the shortest pairs of disjoint paths will strictly double the total cost. We note that this allocation is sufficient to provide protection for all $q \le 1$; so the cost for protecting all $q \le 1$ using a pair of disjoint paths is at most double that of $q = \frac{1}{2}$. The minimum cost to provide protection is monotonically nondecreasing with respect to the value of the partial protection requirement q; this can be clearly seen because if there existed a solution for a demand for some q_2 that has a lower cost than that of some q_1, where $q_2 > q_1$, then the solution for q_2 would be used to protect for q_1 as well. Since the optimal solution is monotonically nondecreasing with respect to q, we know that routing $\frac{1}{2}$ unit of flow onto each of the shortest pairs of disjoint paths is a lower bound, and routing one unit of flow onto each of the disjoint paths will be an upper bound. Hence, routing onto the shortest pair of disjoint paths is at most twice the cost of the optimal solution for any $q > \frac{1}{2}$. Using more disjoint paths, if possible, can only lower the total cost needed to meet demand and protection requirements. ■

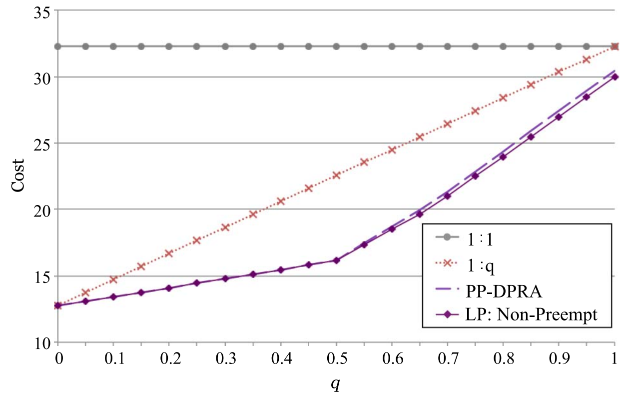

Fig. 8. Algorithm comparison: cost versus q.

To assess PP-DPRA's performance, PP-DPRA is compared to 1:1, 1:q, and LP_{PP}. The simulation is similar to the one run in Subsection III.B for the case in which backup capacity sharing is not possible. PP-DPRA is implemented in C. The average costs to meet demand and protection requirements over all random graphs are plotted in Fig. 8. Simulation results show that for $q \le \frac{1}{2}$, as anticipated, the routing given by Theorem 1 matches the optimal routing produced by LP_{PP}. For $q > \frac{1}{2}$, the average cost is greater than the minimum-cost solution by 1.4% on average. Additionally, on average, the running time for routing a demand with PP-DPRA was 10^{-3} s, while with the linear program LP_{PP} it was 22 s. This reduction in running time of four orders of magnitude makes the algorithm suitable for networks that require rapid setup times for incoming demands.

V. Solutions With Backup Capacity Sharing

In Section III, a linear program that finds the minimum-cost solution for the partial protection problem utilizing backup capacity sharing was presented. A linear program is often not an efficient method for finding a solution, and in Section IV, an efficient algorithm was presented for the case without backup capacity sharing. These results offer a fundamental understanding of the partial protection problem and are useful for networks that do not allow backup capacity sharing. But often, networks do utilize backup sharing, and significant savings can often be achieved. In this section, a time-efficient algorithm for partial protection in general mesh networks using backup capacity sharing is presented.

If two primary flows for two different demands are edge-disjoint from one another, then under a single-link failure model, at most one can be disrupted at any given point in time. Since at most one demand will need to be restored after a failure, two failure-disjoint flows can share backup capacity.

Determining how much backup capacity can be shared for 1:1 guaranteed protection was investigated in [7,8,22]. Because of the difficulty of jointly optimizing all demands when backup capacity can be shared, they consider the case of routing the demands one at a time upon their arrival. We use a similar model for the development of our algorithm.

Conflict sets are used to determine how much backup capacity can be shared by each incoming demand [7,8]. A conflict set indicates how much backup sharing is possible on an edge by examining how much backup capacity it already has to protect against any particular edge failure. If some edge has more backup capacity already assigned to it than is needed to protect against a particular edge failure, then those resources can be used at no additional cost. For example, let some edge $\{i,j\}$ have one unit of backup capacity allocated to it to protect against the failure of $\{k,l\}$, and with edge $\{i,j\}$ not being scheduled to protect against any other link failures. Now consider some new connection with a primary flow that uses some other edge $\{u,v\}$. Edges $\{k,l\}$ and $\{u,v\}$ can never fail simultaneously under a single-link failure model; thus, the new connection can use the backup capacity allocated to $\{i,j\}$ for protecting against the failure of $\{u,v\}$ without incurring additional cost. Further details of protection routing using conflict sets can be found in [7,8]. This model can be extended to partial protection by guaranteeing that any particular demand has its partial flow requirement met after a failure.

For the case of one-at-a-time routing, previous works offer heuristics to jointly optimize the primary and backup paths for each incoming demand, as was done in [7,8,22]. We instead choose a simple strategy of using the shortest path for the primary route. Our simulations show that using the shortest path for the primary route in fact performs better than jointly optimizing the primary and backup paths for each incoming demand. We call our algorithm dynamic shared partial protection (DSPP).

We compare, via simulation, DSPP to 1:1, 1:q, and the nonpreemptive LP (LP_{PP}), each of which jointly optimizes the primary and backup paths for each incoming demand (a "greedy optimal" approach). Demands are served one at a time in the order of their arrival. Once a connection is established, it will remain active for the length of the simulation. The performance of the strategies is compared using the NSFNET topology (Fig. 5) with 100 random unit demands. The protection requirement, q, for each demand has a truncated normal distribution with a standard deviation $\sigma = \frac{1}{2}$. The mean of q is varied between 0 and 1 for each iteration.

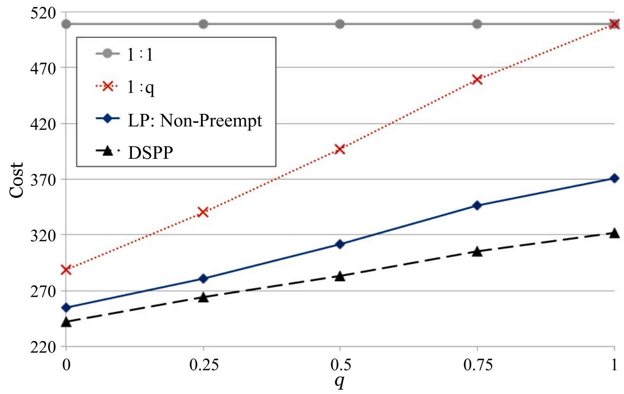

Fig. 9. Sharing algorithm comparison: cost versus q.

The costs to route the demand and protection capacity are plotted in Fig. 9 as a function of the expected value of q. It is seen that when demands are routed one at a time with a greedy optimal approach (i.e., an optimal solution is found with respect to the incoming demand), the partial protection scheme offers significant savings over 1:1 routing over a wide range of q, with LP_{PP} achieving even greater gains than 1:q protection because of its use of flow bifurcation. With routing a demand one at a time upon its arrival, we find that DSPP performs *better* than the greedy schemes that jointly optimize the primary and backup paths for each incoming demand. The greedy optimal approach of jointly optimizing the primary and backup routes will often take a longer primary path for an incoming demand in order to take advantage of backup sharing. The longer primary path makes it more difficult for future demands to find disjoint primary routes, thus lowering their ability to share protection resources. A similar result has been previously observed in [22].

VI. Conclusion

In this paper we developed a mathematical model to provide an alternative form of guaranteed protection in networks: partial protection, which guarantees that a fraction q of a demand remains after a network failure. A linear program was formulated to find a minimum-cost solution for both the cases with and without backup capacity sharing. Simulations show that this LP offers significant savings over the most common protection schemes used today. For the case with backup sharing, both a preemptive and a nonpreemptive scheme are developed, with the preemptive scheme able to offer protection for a wide range of partial protection requirements without the use of any additional resources beyond all the demands' shortest path routings. Algorithms for both the cases with and without backup capacity sharing were developed. Without backup capacity sharing, simulation results show that the algorithm comes within 1.4% of optimal on average and runs four orders of magnitude faster than the linear program. For the case with backup capacity sharing, the algorithm developed actually performs better than jointly optimizing the primary and backup paths for each incoming demand.

Appendix A: Proofs for Subsection IV.A

Proof of Lemma 1: First we show that if there is no spare capacity on any link, i.e., $s_{ij} = 0, \forall \{i,j\} \in E$, and all flow and protection requirements are satisfied, then it must be the case that $x_{ij} \le (1-q), \forall \{i,j\} \in E$. Assume all requirements are satisfied, and that there exists an edge $\{k,l\}$ such that $x_{kl} > (1-q)$ with $s_{ij} = 0, \forall \{i,j\} \in E$. After the failure of $\{k,l\}$, less than q of the flow will remain between the source and destination, which is below the partial protection requirement of q. This implies spare allocation on some edge will be needed to meet all requirements, which contradicts our original assumption.

We now consider the other direction: if $x_{ij} \le (1-q)$, $\forall \{i,j\} \in E$, then the required spare capacity is zero on all edges: $s_{ij} = 0, \forall \{i,j\} \in E$. This is straightforward to see since after the failure of any edge, at most $(1-q)$ of flow between the source and destination is disrupted, leaving at least q, which meets partial protection requirements. ∎

Proof of Lemma 2: For some $q \le \frac{1}{2}$, assume there exists a minimum-cost solution with an edge $\{i,j\}$ that has $s_{ij} > 0$. Since a minimum-cost solution has an edge with spare capacity allocated to it, according to Lemma 1, there must exist some edge $\{k,l\}$ with primary capacity allocation greater than $(1-q)$, i.e., $x_{kl} = 1-q+\epsilon$, $\epsilon > 0$.

To meet protection requirements, after the failure of edge $\{k,l\}$, the remaining flows between s and t must have a total capacity of q. The amount of primary flow remaining after the failure of the edge carrying $(1-q+\epsilon)$ is $(q-\epsilon)$, which means that at least ϵ flow of spare allocation will be necessary along some of the protection paths. If this spare allocation was instead used as primary traffic, the primary flow on $\{k,l\}$ would decrease from $1-q+\epsilon$ to $1-q$, which by Lemma 1 implies that no spare allocation is necessary. This maintains the total flow from s to t at 1; hence, the primary demand and protection requirements are met. Clearly, the total cost of the allocation without spare capacity is less than the cost with spare since the primary capacity on $\{k,l\}$ is reduced and the spare allocation on edge $\{i,j\}$ can be removed. ∎

Proof of Theorem 1: The modified linear program $\mathrm{LP}_{q\le .5}$ seeks to find a minimum-cost routing with capacitated edges. This is recognized to be a *minimum-cost flow* formulation, which is defined as finding a flow of lowest cost between a source and destination in a network that has both edge costs and edge capacities [23]. Algorithms exist for finding optimal solutions to minimum-cost flow problems. One such algorithm is the successive shortest paths (SSP), which successively finds the shortest path and routes the maximum flow possible on that path [23]. This is repeatedly done until the desired flow between the source and destination is routed. After SSP terminates, a set of edge allocations representing the minimum-cost flow will be returned. The paths that these edges represent, and those paths' respective flows, can be found by using the path decomposition algorithm [23].

Before we specify the details of SSP, we first define a residual graph, which is commonly used in maximum flow algorithms [23]. If edge $\{i,j\}$ has a capacity and cost of (u_{ij}, c_{ij}) in a graph G with a flow of $x_{ij} \le u_{ij}$ on it, then the residual graph G^r will have two edges $\{i,j\}^r$ and $\{j,i\}^r$ with respective costs and capacities $(u_{ij}-x_{ij},\ c_{ij})$, and $(x_{ij},\ -c_{ij})$. Any flow in a residual graph preserves all node conservation constraints in the original graph [23].

The SSP algorithm is as follows: for a total demand of d needing to be routed from node s to t, first find the shortest path and route flow equal to the lowest capacity edge in that path. If the limiting edge has capacity u^*, then $d-u^*$ remains to be routed (assuming that $d > u^*$). Next, a residual graph is created from the allocation routed on the shortest path. To route the remaining $d-u^*$ of flow, the shortest path in the newly formed residual graph is found. If $d-u^*$ is less than the lowest capacity edge in that path, the algorithm is completed by routing the remaining $d-u^*$ flow from s to t. Otherwise, flow equal to the lowest capacity edge in the shortest path from s to t is routed, the residual graph is updated, and the process is repeated until all of the required d flow has been routed.

In our original network G, every edge has capacity $(1-q)$; hence, the lowest capacity edge in any path is $(1-q)$. We find the shortest path in G, with a set of edges E_0 and having a total cost p_0; $(1-q)$ of flow is routed on the set of edges E_0. In the residual graph, each edge $\{i,j\} \in E_0$ no longer has capacity and is removed, and a new set of edges $E_0^r = \{\{j,i\}^r : \{i,j\} \in E_0\}$ is added, with each edge having capacity $(1-q)$. All other edges that were not part of the shortest path remain in the residual graph with a capacity of $(1-q)$. Since $(1-q)$ was routed on the shortest path, a flow of q remains to be routed from s to t in the residual graph. All of the edges in the residual graph have a capacity of $(1-q)$, and since we assume $q \le \frac{1}{2}$, we know that $q \le (1-q)$. Hence, the next shortest path from s to t in the residual graph, with the set of edges E_1 and having a total cost of p_1, has sufficient capacity to satisfy the q amount of remaining flow that needs to be routed.

Two cases are possible: 1) if the second path does not use edges created in the residual graph by the initial shortest path, i.e., $E_1 \cap E_0^r = \emptyset$, and 2) if the second path does use those edges, i.e., $E_1 \cap E_0^r \ne \emptyset$.

For case 1, the first and second paths do not overlap, and hence happen to be the shortest pairs of disjoint paths in the network between s and t. Since $(1-q)$ was routed onto the first shortest path, and q was routed onto the next shortest disjoint path, we have the same flow as if routing $(1-2q)$ onto the shortest path, and q onto each of the shortest pairs of disjoint paths. This yields a total allocation cost of $p_0(1-q) + p_1 q$.

For case 2, the set of edges $\{j,i\}^r \in E_1 \cap E_0^r$ are those edges where the second shortest path in the residual graph overlaps with the initial shortest path that was found in the original graph. Any allocation on $\{j,i\}^r \in E_1 \cap E_0^r$ "cancels" the original flow allocated on edge $\{i,j\}$ from the first shortest path found; i.e., if one unit of flow is routed on edge $\{i,j\}$ in the original graph, and $\frac{1}{2}$ unit of flow is allocated on $\{j,i\}^r$ in the residual graph, then the flow in the original graph on edge $\{i,j\}$ is $\frac{1}{2}$. The flow on the edges where the two paths overlapped is $x_{ij} = (1-q) - q = (1-2q)$,

$\forall \{j,i\} \in E_1 \cap E_0^r$, which is nonnegative since $q \le (1-q)$. The remaining edges that did not overlap maintain their original flow values: if $\{i,j\}$ belonged to E_0, then its value is $x_{ij} = (1-q)$, and if $\{i,j\}$ belonged to E_1, then its value is $x_{ij} = q$.

To recover the paths, we use flow decomposition [23], which is to repeatedly find a path from source to destination, and subtract the flow equivalent to the minimum edge capacity until all flow from the network has been assigned to some path (almost a SSP in reverse). We first find the edges of the shortest path, which now has a maximum flow of $(1-2q)$. After removing this flow from the network, we are left with two disjoint paths of q flow each, and costs of p_1 and p_2, respectively. These disjoint paths are by definition the minimum-cost pair of disjoint paths: if there existed a lower-cost pair of disjoint paths, then we could produce a lower-cost flow by routing $(1-2q)$ onto the shortest path and q onto each of the lower-cost pairs of disjoint paths, which is a feasible flow and would give a lower-cost routing, which is not possible since the SSP algorithm found the minimum-cost solution. Hence, we are left with a minimum-cost solution having a cost of $(1-2q)p_0 + q(p_1+p_2)$. ■

Appendix B: Proofs of Subsection IV.B

The following is used throughout all proofs in this section: $\mathbf{A_N}$ is the $N \times N$ matrix of 1's with the identity matrix subtracted from it (an all 1's matrix with a diagonal of zeros), $\mathbf{c_N}$ and $\mathbf{x_N}$ are the cost and edge allocation row vectors for N edges, respectively, and $\mathbf{e_N}$ is a column vector of N 1's. The expression $C = [\mathbf{AB}]$ denotes a concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$. Throughout these proofs, results from optimization theory are used, with pertinent details being found in [24].

We note that the proofs for Lemmas 4, 5, and 6 are given before the proof for Lemma 3.

Proof of Lemma 4: For a given set of edges E with the ith edge having a cost of c_i, the linear program for a two-node network needing to route a unit demand with a partial protection requirement of q can be written as follows:

$$\mathrm{LP}_2\!: \min\ \mathbf{x}'_\mathbf{N}\mathbf{c_N}, \tag{B1}$$

$$\text{s.t.}\quad \sum_{i\in E} x_i \ge 1, \tag{B2}$$

$$\mathbf{A_N}\mathbf{x_N} \ge q\mathbf{e_N}, \tag{B3}$$

$$x_i \ge 0, \quad \forall\, i \in E. \tag{B4}$$

Constraint (B2) specifies that at least one unit of flow must be routed across the set of links, and constraint (B3) indicates that after any particular link fails, at least q flow must remain across the remaining set of links.

Since we assume that spare allocation is needed, the total allocation across the set of edges is strictly greater than 1. Hence, constraint (B2), which indicates that flow must be at least 1, is not tight, and can be disregarded.

When solving the primal, LP_2, we know generally that some $K \le N$ edges are active, and $N-K$ edges are not; i.e., if edge i is active, then $x_i > 0$, and vise versa. We note again that the solution will clearly use the K lowest-cost edges; otherwise capacity can be shifted from higher- to lower-cost edges, yielding a lower-cost solution. With the K lowest-cost variables being active, and the $N-K$ highest-cost variables being zero, constraints $K+1$ through N all have the form $\sum_{i=1}^{K} x_i \ge q$. Summing the first K constraints, we find $\sum_{i=1}^{K} x_i = q\frac{K}{K-1}$; constraints $K+1$ through N are all clearly no longer active (and they also no longer linearly independent), and they can be disregarded. By removing the constraints from LP_2 that are not tight, we are left with K variables and K constraints. To solve for K variables, all K constraints must be used and can be set to equality. The primal is rewritten as follows:

$$\mathrm{LP}_{2K}\!: \min\ \mathbf{x}'_\mathbf{K}\mathbf{c_K}, \tag{B5}$$

$$\text{s.t.}\quad \mathbf{A_K}\mathbf{x_K} = q\mathbf{e_K}, \tag{B6}$$

$$x_i \ge 0, \quad \forall\, i = 1, \ldots, K. \tag{B7}$$

This solution is straightforward to find, and is an even distribution of $x_i = q\frac{1}{K-1}$, $\forall\, i = 1, \ldots, K$.

We now use an inductive approach to show that all edges that satisfy the requirement $c_j < \frac{1}{K-1}\sum_{i=1}^{K} c_i$ are in fact part of the minimum-cost solution, where $K = \mathrm{argmax}_{K=2,\ldots,N}(c_K \le \frac{1}{K-1}\sum_{i=1}^{K} c_i)$. Assume that a minimum-cost solution uses the $J-1$ lowest-cost edges, and that the Jth edge has cost $c_J < \frac{1}{J-1}\sum_{i=1}^{J} c_i$. If $J-1$ edges are used, then as shown above, the solution will be an even distribution across those $J-1$ edges of $x_i = q\frac{1}{J-2}$, $\forall\, i = 1, \ldots, (J-1)$. The total cost of the assumed optimal solution is $q\frac{1}{J-2}\sum_{i=1}^{J-1} c_i$. We now consider the solution that uses edge J, which was previously excluded. With J edges being used, each edge will have an even allocation of $x_i = q\frac{1}{J-1}$, $\forall\, i = 1, \ldots, J$, and the total cost across the J edges will be $q\frac{1}{J-1}\sum_{i=1}^{J} c_i$. Using some algebraic manipulation, we see that the solution using J edges will have higher cost only if $c_J > \frac{1}{J-1}\sum_{i=1}^{J} c_i$, but we assumed otherwise. Hence, a lower-cost solution can be obtained if all J edges are used. Inductively, we see that this approach can be continued until we find the K lowest-cost edges such that $K = \mathrm{argmax}_{K=2,\ldots,N}(c_K \le \frac{1}{K-1}\sum_{i=1}^{K} c_i)$. ■

Proof of Corollary 1: The set of K edges that are active in a minimum-cost solution when spare capacity is needed is given by the result in Lemma 4: $K = \mathrm{argmax}_{K=2,\ldots,N}$ $(c_K \frac{1}{K-1}\sum_{i=1}^{K} c_i)$, which does not depend on q. ■

Proof of Lemma 5: In the proof for Lemma 4, a solution that optimally solved the two-node network was found for the primal formulation LP_{2K}, which uses K edges, by setting the constraints to equality. This solution was an even distribution of $x_i = q\frac{1}{K-1}$, $\forall\, i = 1, \ldots, K$. We can perform an additional check that our solution is optimal by verifying complementary slackness conditions (not done here for brevity). ∎

Proof of Lemma 6: When spare allocation is not needed, the total allocation across all of the edges is 1:

$$\text{LP}_2\text{: min } \mathbf{c}'_\mathbf{N}\mathbf{x}_\mathbf{N}, \tag{B8}$$

$$\text{s.t.} \quad \mathbf{A}_\mathbf{N}\mathbf{x}_\mathbf{N} \geq q\mathbf{e}'_\mathbf{N}, \tag{B9}$$

$$\sum_{i=1}^{N} x_i = 1, \tag{B10}$$

$$x_i \geq 0, \quad \forall\, i \in E. \tag{B11}$$

The corresponding dual is as follows:

$$\text{LP}_{2d}\text{: max} \sum_{i=1,\ldots,N} qp_i + p_{N+1}, \tag{B12}$$

$$\text{s.t.} \quad \mathbf{p}'_{\mathbf{N+1}}[\mathbf{A}_\mathbf{N}\mathbf{e}_\mathbf{N}] \leq \mathbf{c}'_\mathbf{N}, \tag{B13}$$

$$p_i \geq 0, \quad \forall\, i = 1, \ldots, (N+1). \tag{B14}$$

We note that the dual variable, p_{N+1}, corresponding to the primal constraint $\sum_{i=1,\ldots,N} x_i = 1$, is no longer necessarily zero, as it would be if $\sum_{i=1,\ldots,N} x_i > 1$.

We initially find an optimal solution to the problem as if it requires spare capacity, which assumes constraint (B10) is not active. When constraint (B10) is set to equality, then the current solution is still dual feasible if p_{N+1} is set to zero. The solution with spare capacity (given in Lemma 5) uses an even distribution of $q\frac{1}{K-1}$ on each of the K lowest-cost edges, where $K = \text{argmax}_{K=2,\ldots,N}(c_K \leq \frac{1}{K-1}\sum_{i=1}^{K} c_i)$. Since the K lowest-cost edges are used, the first K constraints will be active in the dual. Additionally, since the first K constraints are tight in LP_2, the first K dual variables will be used to solve for an optimal dual solution. Hence, the solution to the dual will have $p_i > 0$, $\forall\, i = 1, \ldots, K$ and $p_i = 0$, $\forall\, i = (K+1), \ldots, N$.

We now consider the case in which spare capacity is not used. The solution for when spare capacity *is* used remains dual feasible when the spare allocation is *not* used if p_{N+1} is initially set to 0. This solution, while being feasible, is not necessarily optimal. We will use this initial dual feasible solution as our starting point. The initial dual feasible solution has the first K constraints tight (at equality); each of these constraints contain the variables p_{K+1} to p_{N+1}, which are all initially equal to zero.

We wish to find a lower-cost solution to the primal, which means finding a higher value for the objective of the dual. Due to the structure of the problem's linear program and its subsequent dual, if any dual variable p_i, $i = 1, \ldots, N$, has value greater than zero, then the corresponding edge i (primal variable x_i) will be nonzero. We wish to show that when a solution does not use spare capacity, at most the K lowest-cost edges will be used, where the K edges are those used for the minimum-cost allocation when spare allocation is needed. Hence, we want to show that raising the value of any dual variable p_{K+1} through p_N will not raise the objective function. In the current dual feasible solution, where p_{N+1} is set to zero, all constraints 1 through K are tight. To increase the value of any dual variable that is currently at zero (p_{K+1} through p_{N+1}), the sum of the dual variables that currently have value must be decreased.

Assume we wish to raise p_{K+1} through p_{N+1} by some amount δ, i.e., $\sum_{i=K+1}^{N+1} p_i = \delta$. Since the first K constraints are tight, we must decrease p_1 to p_K by at least δ. Consider the jth tight constraint for some $j \leq K$. The jth constraint has the following form: $p_1 + p_2 + \cdots + 0p_j + \cdots + p_K + \sum_{i=K+1}^{N+1} p_i = c_j$, where p_j is multiplied by zero to show that it does not appear in the jth constraint. To raise $\sum_{i=K+1}^{N+1} p_i$ by δ, the $K-1$ dual variables that are greater than zero in the jth constraint must each be reduced by $\frac{1}{K-1}\delta$. When we consider all K tight constraints, it can be easily shown that the only feasible solution to raise $\sum_{i=K+1}^{N+1} p_i$ by δ is to lower each dual variable p_1 through p_K by $\frac{1}{K-1}\delta$. Hence, to achieve the increase of δ across the variables p_{K+1} through p_{N+1}, the total decrease across the first K dual variables is $\frac{K}{K-1}\delta$, which we note is greater than δ.

Looking at the objective function [Eq. (B12)], it can be seen that dual variables p_1 through p_N all have the same cost of q. Any increase in the dual variables p_{K+1} through p_N will result in a larger decrease of the dual variables p_1 through p_K, which will bring down the total value of the objective. Hence, raising the value of p_{K+1} through p_N will not find a new maximum for the objective. The cost of the dual variable p_{N+1} in the objective is 1, and $q \leq 1$. So, it may be possible to raise p_{N+1} while decreasing p_1 through p_K, and also increase the objective function. The dual variable p_{N+1} appears in each of the first K tight constraints. Again, consider the jth tight constraint for some $j \leq K$, and exclude the dual variables p_{K+1} through p_N. The jth constraint has the following form: $p_1 + p_2 + \cdots + 0p_j + \cdots + p_K + p_{N+1} = c_j$. Raising p_{N+1} will simply result in a strict decrease across the first K dual variables. Furthermore, since p_{K+1} through p_N also appear in each of the first K tight constraints, there is never a reason to raise some p_i, $(K+1) \leq i \leq N$, since a larger increase in the objective can be achieved by shifting any allocation that would go from p_i to p_{N+1} instead. Hence, if p_{N+1} were raised, at most the original dual variables p_1 through p_K will be nonzero, which will yield a solution using at most the K lowest-cost edges. ∎

Proof of Lemma 3: First, assume there exists a minimum-cost solution that uses spare capacity when $q \leq \frac{K-1}{K}$. Since we assume that spare capacity is used for a minimum-cost solution, we know that the results from Lemmas 4 and 5 hold. The total capacity allocated for a minimum-cost solution across the K links is $q\frac{K}{K-1}$. Since we assumed that $q \leq \frac{K-1}{K}$, the total allocation across all of the links is $q\frac{K}{K-1} \le \frac{K-1}{K}\frac{K}{K-1} = 1$, which means that the total allocation either uses no spare allocation or is insufficient to meet the primary demand. Hence, we have a contradiction.

Next, assume there exists a minimum-cost solution using no spare capacity if $q > \frac{K-1}{K}$. Continuing the proof from Lemma 5, we consider the objective function of the dual, $\max \sum_{i=1,\ldots,N} qp_i + p_{N+1}$. If the dual variable p_{N+1} is greater than zero, then its corresponding primal constraint [constraint (B10)] must be tight (by way of complementary slackness). If constraint (B10) is tight, then no spare allocation is used. In the proof for Lemma 5, a dual feasible solution was considered, and the conditions for increasing the objective were found. To increase the dual variable p_{N+1} by δ, the first K dual variables must be decreased by a total of $\delta\frac{K}{K-1}$. The cost of each of the first K dual variables is q, while the cost of p_{N+1} is 1. The dual objective can only be raised if the decrease in cost from lowering the first K dual variables is offset by an even larger increase in the objective by raising p_{N+1}. A δ increase in p_{N+1} increases the objective by δ, but decreases the first K dual variables by $\delta\frac{K}{K-1}$, which lowers the objective by $q\delta\frac{K}{K-1}$. We assume that $q > \frac{K-1}{K}$. The decrease in the objective from the first K dual variables is $q\delta\frac{K}{K-1} > \delta\frac{K-1}{K}\frac{K}{K-1} > \delta$. Hence, when $q > \frac{K-1}{K}$, a better solution can be found by having p_{N+1} be greater than zero, which means that no spare allocation is used, thus contradicting our original assumption. ∎

Proof of Theorem 2: We continue the proof from Lemma 3. As was shown, when $q \le \frac{K-1}{K}$, the objective of the dual function can be raised by increasing p_{N+1}. The initial dual feasible solution when $q < \frac{K-1}{K}$ has $p - N + 1$ set to zero. Solving for the dual variables in the set of linear equations of our initial solution (while p_{N+1} is at 0), we get $p_j = \sum_{i=1}^{K} c_i - \frac{2K-3}{K-1}c_j$. By definition, $c_1 \le c_2 \le \cdots \le c_N$, so the solutions to the dual variables are $p_K \le p_{K-1} \le \cdots \le p_1$. To increase p_{N+1} by some value δ, p_1 to p_K each decrease by $\delta\frac{1}{K-1}$, for a total decrease of $\delta\frac{K}{K-1}$.

We now increase δ by increments, until one of the dual variables goes to zero. Since each dual variable decreases by the same amount $\delta\frac{1}{K-1}$, the dual variable with the lowest value will go to zero first. The dual variable p_K will be the first to go to zero, and the corresponding Kth constraint is no longer active, giving a solution in which the Kth edge has zero allocation; this was the most expensive edge that was in use, so it matches intuition that it would be the first to go to zero. Currently, p_1 to p_{K-1} are greater than zero, and p_{N+1} is also greater than zero. We wish to see whether we can further increase p_{N+1} and continue raising the objective value. Without p_K, which is now zero, we have the following objective function: $\max \sum_{i=1}^{K-1} qp_i + p_{N+1}$. Using a similar process as above, we get the condition that we will increase δ (which is p_{N+1}) only if $q < \frac{K-2}{K-1}$. This process can be repeated until raising the value of p_{N+1} further does not increase the objective. Inductively, we stop increasing p_{N+1} when we have J active dual variables such that $\frac{J-1}{J-2} < q \le \frac{J}{J-1}$, where J is an integer. By complementary slackness, there are J active constraints in the primal, which yields a solution using J variables, which will be the J lowest-cost edges. We get the following set of J independent equations: $\mathbf{x}'_\mathbf{J}[\mathbf{A}_{\mathbf{J-1}}\mathbf{e}_\mathbf{J}] = [q\mathbf{e}_\mathbf{J}1]'$. We can solve this set of linear equations, and obtain the results in Theorem 2. ∎

Acknowledgments

This work was supported by NSF grants CNS-1116209 and CNS-0830961, by DTRA grants HDTRA1-07-1-0004 and HDTRA-09-1-005, and by the Department of the Air Force under Air Force contract no. FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government. A preliminary version of this work appeared in IEEE INFOCOM 2011 and OFC/NFOEC 2012.

References

[1] B. Mukherjee, "WDM optical communication networks: Progress and challenges," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1810–1824, 2000.

[2] W. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice-Hall, 2004.

[3] T. H. Shake, B. Hazzard, and D. Marquis, "Assessing network infrastructure vulnerabilities to physical layer attacks," in *22nd National Information Systems Security Conf.*, Arlington, VA, Oct. 18–21, 1999.

[4] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightwave Technol.*, vol. 21, no. 4, pp. 870–883, 2003.

[5] R. Iraschko, M. MacGregor, and W. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 325–336, 1998.

[6] W. Yao and B. Ramamurthy, "Survivable traffic grooming with path protection at the connection level in WDM mesh networks," *J. Lightwave Technol.*, vol. 23, no. 10, p. 2846, 2005.

[7] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, "New and improved approaches for shared-path protection in WDM mesh networks," *J. Lightwave Technol.*, vol. 22, no. 5, pp. 1223–1232, 2004.

[8] G. Mohan, S. Murthy, and A. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," *IEEE/ACM Trans. Netw.*, vol. 9, no. 5, pp. 553–566, 2002.

[9] A. Fumagalli, I. Cerutti, M. Tacca, F. Masetti, R. Jagannathan, and S. Alagar, "Survivable networks based on optimal routing and WDM self-healing rings," in *18th Annu. Joint Conf. of the IEEE Computer and Communications Societies (IEEE INFOCOM)*, 1999, pp. 726–733.

[10] A. Saleh and J. Simmons, "Evolution toward the next-generation core optical network," *J. Lightwave Technol.*, vol. 24, no. 9, pp. 3303–3321, 2006.

[11] W. Grover and M. Clouqueur, "Span-restorable mesh networks with multiple quality of protection (QoP) service classes," *Photon. Netw. Commun.*, vol. 9, no. 1, pp. 19–34, 2005.

[12] O. Gerstel and G. Sasaki, "Quality of protection (QoP): A quantitative unifying paradigm to protection service grades," *Opt. Netw. Mag.*, vol. 3, no. 3, pp. 40–49, 2002.

[13] O. O. Gerstel and G. H. Sasaki, "A new protection paradigm for digital video distribution networks," in *IEEE Int. Conf. on Communications (ICC)*, 2006, pp. 2518–2523.

[14] G. H. Sasaki, K. Wang, and Y. Wang, "Fractional survivable network bandwidth: A comparison of IP over WDM schemes," in *Optical Fiber Communication Conf.*, 2010, paper OWH5.

[15] J. Fang and A. Somani, "On partial protection in groomed optical WDM mesh networks," in *Proc. of the 2005 Int. Conf. on Dependable Systems and Networks*, 2005, pp. 228–237.

[16] A. Das, C. Martel, and B. Mukherjee, "A partial-protection approach using multipath provisioning," in *IEEE Int. Conf. on Communications (ICC)*, 2009, pp. 1–5.

[17] G. Kuperman, E. Modiano, and A. Narula-Tam, "Analysis and algorithms for partial protection in mesh networks," in *Proc. IEEE INFOCOM*, 2011, pp. 516–520.

[18] G. Kuperman, E. Modiano, and A. Narula-Tam, "Partial protection in networks with backup capacity sharing," in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, 2012, paper NW3K.4.

[19] L. Ruan and N. Xiao, "Survivable multipath routing and spectrum allocation in OFDM-based flexible optical networks," *J. Opt. Commun. Netw.*, vol. 5, no. 3, pp. 172–182, 2013.

[20] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 885–896, 1999.

[21] J. Suurballe and R. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325–336, 1984.

[22] H. Wang, E. Modiano, and M. Medard, "Partial path protection for WDM networks: End-to-end recovery using local failure information," in *7th Int. Symp. on Computers and Communications (ISCC)*, 2002, pp. 719–725.

[23] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. New Jersey: Prentice-Hall, 1993.

[24] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, Massachusetts: Athena Scientific, 1997.

Greg Kuperman received his Ph.D. in communications and networking from the Massachusetts Institute of Technology (MIT) in 2013, and he received his B.S.E in computer and telecommunication engineering and M.S.E. in electrical engineering from the University of Pennsylvania in 2005. He is currently a post-doctoral researcher at MIT in the Laboratory of Information and Decision Systems (LIDS).

Eytan Modiano received his B.S. degree in electrical engineering and computer science from the University of Connecticut at Storrs in 1986 and his M.S. and Ph.D. degrees, both in electrical engineering, from the University of Maryland, College Park, MD, in 1989 and 1992, respectively. He was a Naval Research Laboratory Fellow between 1987 and 1992 and a National Research Council Post Doctoral Fellow during 1992–1993. Between 1993 and 1999 he was with MIT Lincoln Laboratory where he was a project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. Since 1999 he has been on the faculty at MIT, where he is a Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks.

He is currently an Associate Editor for *IEEE/ACM Transactions on Networking*. He has served as Associate Editor for *IEEE Transactions on Information Theory*, and as a guest editor for the *IEEE Journal on Selected Areas in Communications* special issue on WDM network architectures, the *Computer Networks* journal special issue on broadband Internet access, the *Journal of Communications and Networks* special issue on wireless ad-hoc networks, and the IEEE *Journal of Lightwave Technology* special issue on optical networks. He was the Technical Program co-chair for IEEE WiOpt 2006, IEEE INFOCOM 2007, and ACM MobiHoc 2007. He is a Fellow of the IEEE and an Associate Fellow of the AIAA.

Aradhana Narula-Tam is an Assistant Leader of the Airborne Networks Group at MIT Lincoln Laboratory. She joined the laboratory in 1998 and has worked in a variety of areas including optical networking, dynamic resource allocation for satellite communication systems, and IP quality of service. She was the Principal Investigator for MIT Lincoln Laboratory's efforts on the DARPA Core Optical Networking program prior to taking a temporary position as an Air Force civilian to lead the Modem Processor Group product team for the Family of Advanced Beyond Line-of-Sight Terminal (FAB-T) program during 2009–2010. Upon returning to the laboratory, she has focused on research and design of line-of-sight communication technologies and network architectures for highly mobile platforms in tactical environments. Dr. Narula-Tam holds a B.S.E. degree from the University of Pennsylvania and S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, all in electrical engineering. Her doctoral thesis analyzed theoretical performance limits of antenna diversity methods for mobile communications. She has served as session and track chairs for IEEE MILCOM and co-authored over 30 papers in the areas of wireless communications, satellite communication systems and networking, and optical networking.