Analysis and Algorithms for Partial Protection in Mesh Networks

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

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

Aradhana Narula-Tam
MIT Lincoln Laboratory
Lexington, MA 02420
arad@ll.mit.edu

Abstract—This paper develops a mesh network protection scheme that guarantees a quantifiable minimum grade of service upon a failure within a network. The scheme guarantees that a fraction $q$ of each demand remains after any single link failure. 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 optimal routing and allocation is developed using multiple shortest paths. For $q > \frac{1}{2}$, a heuristic algorithm based on disjoint path routing is developed 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 82% over traditional full protection schemes.

I. INTRODUCTION

Mesh networks supporting data rates of multiple gigabytes per second are being deployed to meet the increasing demands of the telecom industry [1]. As data rates continue to increase, 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 [2]. Currently, there exist few options for protection that offer less than complete restoration after a failure. Due to the cost of providing full protection, many service providers offer no protection whatsoever. By defining varying and quantifiable grades of protection, service providers can protect essential traffic 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.

There are a variety of protection strategies available [3], [4], [5]. The most common in backbone networks today is guaranteed path protection [6], which provides an edge-disjoint backup path for each working path, resulting in 100% service restoration after any link failure. Best effort protection is still loosely defined, but generally it offers no guarantees on the amount of protection provided. A service will be protected, if possible, with any unused spare capacity after fully protecting all guaranteed services [2]. Best effort protection can also be referred to as partial capacity protection, since a service will be restored within existing spare capacity, typically resulting in less than 100% restoration.

Many users may be willing to tolerate short periods of reduced capacity if data rate guarantees can be made at a reduced cost, especially since link failures are relatively uncommon and are on average repaired quickly [2]. 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 [7]. In this 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 0, the service is fully protected, and when $q$ is 1, the service is unprotected. More recently, [8] examines the savings that can be achieved by guaranteeing part of the demand in the event of a link failure, as opposed to full protection. It shows that the amount of protection that can be guaranteed depends on the topology of the network. In [9], the partial protection problem on groomed optical WDM networks is studied, under the assumption that flows must traverse a single path.

In this work, we further expand upon the framework developed in [7] and [8]. We develop a “theory” for partial protection that includes optimal algorithms for capacity allocation, and explicit expressions for the amount of required spare capacity. Routing strategies that allocate working and spare capacity to meet partial protection requirements are derived. Similar to [8], flow bifurcation over multiple paths is allowed. Bifurcation reduces the amount of spare capacity needed to support the QoP requirements. In fact, we show that depending on the value of $q$, it may be possible to provide protection without any spare capacity.

We develop a linear program to find the optimal minimum-cost capacity allocation needed to guarantee partial protection in the event of a link failure. Furthermore, a routing and capacity assignment strategy based on shortest paths is shown to be optimal for $q \leq \frac{1}{2}$. For $q > \frac{1}{2}$, a reduced complexity 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. Simulations over many random network topologies show that this disjoint path routing algorithm performs on average within 1.4% of the minimum-cost solution and leads to as much as 82% savings as compared to traditional full protection schemes.

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 working and backup capacity. In Section IV, a simple path based solution for \( q \leq \frac{1}{2} \) is developed. In Section V, properties of the minimum-cost solution for \( q > \frac{1}{2} \) are determined and used to develop a time-efficient heuristic algorithm. The results of the algorithm are compared to the optimal solution and to traditional protection schemes. All proofs are omitted for brevity and can be found in [10].

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 the demand 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 assume that demands do not share protection capacity with one another. Both working traffic and protection flows (defined as the flow after a failure) can be bifurcated to traverse multiple paths between the source and destination. After the failure of a link, a network management algorithm reroutes the traffic along the allocated protection paths. Without loss of generality, we assume unit demands, unless noted otherwise.

For now assume that link costs are all 1; in the next section we will consider non-uniform link costs. With uniform link costs, the objective is to minimize the total capacity needed to support the flow and the partial protection requirements.

![Fig. 1: Standard protection schemes](image)

One routing strategy for providing this backup capacity is to use a single primary and a single backup path similar to the \( 1+1 \) guaranteed path protection scheme. Consider the network shown in Figure 1. With \( 1+1 \) protection, one unit of capacity is routed on a primary path and one unit of capacity on a backup, as shown in Figure 1a. Upon a link failure, 100% of the service can be restored on 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 naïve 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 Figure 1b. This simple protection scheme will be referred to as \( 1+q \) protection. If the primary path fails, sufficient backup capacity remains to provide service for \( \frac{2}{3} \) of the demand.

![Fig. 2: Protection using risk distribution](image)

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 Figure 2a, 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 allocation of \( s = \frac{1}{3} \) 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 \( q \) between two nodes \( s \) and \( t \). In particular, the full demand must be met before any failure, and in the event of any link failure, a fraction \( q \) of that demand must remain. The linear program to solve for the optimal routing strategy, denoted \( LP_{PP} \), is defined below.

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

The following values are given:

- \( G = (V,E) \) is the graph with its set of vertices and edges
- \( (s,t) \) is the source and destination, respectively
- \( d \) is the total demand between the source and destination
- \( q \) is the fraction of the demand that must be supported on the event of a link failure
- \( c_{ij} \) is the cost of link \( \{i,j\} \)

The LP solves for the following variables:

- \( w_{ij} \) is the working flow assigned on link \( \{i,j\}, w_{ij} \geq 0 \)
- \( s_{ij} \) is the spare allocation assigned on link \( \{i,j\}, s_{ij} \geq 0 \)
- \( f_{kl}^{ij} \) is the protection flow assigned on link \( \{i,j\} \) after the failure of link \( \{k,l\}, f_{kl}^{ij} \geq 0 \)
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:

- Route working traffic between $s$ and $t$ to meet demand $d$:
  \[
  \sum_{\{i,j\} \in E} w_{ij} - \sum_{\{j,i\} \in E} w_{ji} = \begin{cases} 
    d & \text{if } i = s \\
    -d & \text{if } i = t, \forall i \in V \\
    0 & \text{o.w.}
  \end{cases} 
  \tag{2}
  \]

- Route flow to meet partial protection requirement $q$ after failure of link $\{k,l\}$ between $s$ and $t$:
  \[
  \sum_{\{i,j\} \in E, \{i,j\} \neq \{k,l\}} f_{ij} - \sum_{\{j,i\} \in E, \{j,i\} \neq \{k,l\}} f_{ji} = \begin{cases} 
    dq & \text{if } i = s \\
    -dq & \text{if } i = t , \forall i \in V \\
    0 & \text{o.w.}
  \end{cases} 
  \tag{3}
  \]

- Working and spare capacity assigned on link $\{i,j\}$ meets partial protection requirements after failure of link $\{k,l\}$:
  \[
  f_{ij} \leq w_{ij} + s_{ij}, \forall \{i,j\} \in E, \forall \{k,l\} \in E \tag{4}
  \]

A minimum-cost solution will provide a flow to meet the demand before a link failure and a flow to meet the partial protection requirement after any single-link failure. As we allow bifurcation, each of these 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 working flow is maintained, but the total allocation for working plus spare capacity, given by $(w_{ij} + s_{ij})$ for edge $\{i,j\}$, does not necessarily maintain flow conservation.

**B. Comparison to Standard Protection Schemes**

We compare the optimal solution computed by the above linear program to the standard scheme of $1+1$ protection, as well as $1+q$ protection, on 1000 random graph topologies, each containing 50 nodes with an average node degree of 3.1 and random link costs. Two nodes were randomly chosen from each graph to be the source and destination. The minimum-cost allocation for values of $q$ between 0 and 1 was determined by the linear program using CPLEX. The $1+1$ and $1+q$ protection schemes were solved using the Bhandari algorithm for shortest pair of disjoint paths [11].

The average cost to route the demand and protection capacity using the different routing strategies are plotted in Figure 3 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 using fewer resources than $1+1$, however, the minimum-cost solution produced by the partial protection linear program uses significantly less capacity. A lower bound on the capacity requirement is the minimum-cost routing that provides no protection ($q = 0$), 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 minimum-cost routing with no protection. Partial protection 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.

**IV. Solution for $q \leq \frac{1}{2}$**

In this section we provide insights on the structure of the solution to the minimum-cost partial protection problem on general mesh networks. When $q \leq \frac{1}{2}$, we are able to derive an exact algorithmic solution to the partial protection problem. We show that all minimum-cost solutions for $q \leq \frac{1}{2}$ will never need spare allocation, allowing us to formulate the partial protection problem using standard network flow conservation constraints. A simple path-based algorithmic solution is then derived. The difficulty in obtaining further insights into the optimal solution for $q > \frac{1}{2}$ stems from the fact that, as mentioned in Section II, the total working and spare allocation does not necessarily meet flow conservation requirements at each node. Without this property, most network flow algorithms do not apply [12] and analysis of the linear program becomes difficult.

Lemma 1 demonstrates that spare capacity is not needed if and only if the working capacity on an edge is less than or equal to $(1-q)$, because that means that any time a link is lost, at least $q$ remains in the network.

**Lemma 1.** Given a partial protection requirement $q$ between nodes $s$ and $t$, the spare capacity needed to satisfy demand and protection requirements is zero if and only if the working capacity on each link is $w_{ij} \leq (1-q), \forall \{i,j\} \in E$.

**Proof:** If $s_{ij} = 0, \forall \{i,j\} \in E$, then $w_{ij} \leq (1-q), \forall \{i,j\} \in E$. Assume that there exists an edge $\{k,l\}$ such that $w_{kl} > (1-q)$ with $s_{ij} = 0, \forall \{i,j\} \in E$. After the failure
of \( \{k,l\} \), less than \( q \) of flow will remain between the source and destination, which is below the partial protection requirement of \( q \) and implies spare allocation on some edge will be needed.

If \( w_{ij} \leq (1-q), \forall \{i,j\} \in E \), then it is clear that the required spare capacity is zero on all edges: \( s_{ij} = 0, \forall \{i,j\} \in E \). This is because 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.

In Section V, we show routings with zero spare allocation are not necessarily lowest cost for all values of \( q \). However, Lemma 2 shows that when \( q \leq \frac{1}{2} \), the minimum-cost solution will never use spare allocation.

**Lemma 2.** Given a demand between nodes \( s \) and \( t \) with protection requirement \( q \leq \frac{1}{2} \), all minimum-cost solutions use no spare capacity: \( s_{ij} = 0, \forall \{i,j\} \in E \).

*Proof:* Assume there exists a minimum-cost solution with an edge \( \{i,j\} \) that has \( s_{ij} > 0 \). Lemma 1 indicates that there exists an edge \( (k,l) \) with \( w_{kl} = 1 - q + \epsilon, \epsilon > 0 \). In order to provide sufficient protection, the remaining flows among \( s \) and \( t \) must have capacity \( q \) after the failure of edge \( (k,l) \). The amount of working flow remaining after the failure of the edge carrying \((1-q+\epsilon)\) is \((q-\epsilon)\), which means that at least \( \epsilon \) flow of spare allocation will be necessary along some of the protection paths. If instead this spare allocation was used for working traffic, the working flow on \( (k,l) \) would decrease from \( 1-q+\epsilon \) to \( 1-q \), which by Lemma 1 implies that no spare allocation is necessary. The total flow from \( s \) to \( t \) remains at \( 1 \), which meets the demand requirements. Clearly, the total cost of the allocation without spare capacity is less than the cost with spare as the working capacity on \( \{k,l\} \) is reduced.

It also needs to be demonstrated that converting spare flow to working flow will not cause that edge’s working allocation to be greater than \((1-q)\). Assume that in the initial allocation, there exists an edge \( \{i,j\} \) with spare capacity such that \( w_{ij} + s_{ij} > (1-q) \); so \( w_{ij} = 1-q - \delta \), with \( \delta < s_{ij} \leq \epsilon \). After some failure, \( s_{ij} \) is needed to meet the protection requirement of \( q \) flow from \( s \) to \( t \). After a failure, edge \( \{i,j\} \) has greater than \((1-q)\) allocated to it: \( w_{ij} + s_{ij} > 1-q \). With \( q \leq \frac{1}{2} \), we know that \((1-q) \geq q \), which means that a flow of \((1-q)\) will always be sufficient to meet protection requirements. This amount is more than what is needed to meet protection requirements, which means the original allocation was not optimal since spare allocation can be reduced by \( \delta \).

Combining Lemmas 1 and 2, it can be seen that an optimal solution exists that does not use any spare allocation for \( q \leq \frac{1}{2} \) with \( w_{ij} \leq (1-q), \forall \{i,j\} \in E \). For the case of \( q \leq \frac{1}{2} \), no spare allocation is needed and flow conservation constraints are met. Therefore, the linear program can be written using a standard flow formulation without using spare allocation, maintaining flow conservation at each node. The modified linear program, referred to as \( LP_{q \leq 1/2} \), routes the flows on the paths in a manner that minimizes total cost and ensures that no edge carries more than \((1-q)\).

\[
LP_{q \leq 0.5} : \min \sum_{\{i,j\} \in E} c_{ij} w_{ij} \quad (5)
\]
\[
\sum_{\{i,j\} \in E} w_{ij} - \sum_{\{j,i\} \in E} w_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t, \forall i \in V \\ 0 & \text{o.w.} \end{cases} \quad (6)
\]
\[
w_{ij} \leq (1-q), \forall \{i,j\} \in E \quad (7)
\]

The above linear program achieves a minimum-cost routing in a network by using only working allocation to meet the demand. \( LP_{q \leq 0.5} \) is a network flow problem with directed and capacitated edges, which is recognized as a minimum-cost flow problem [12], 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 pair of disjoint paths and allocation \((1-2q)\) on the shortest path.\(^1\)

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 pair 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 pair of disjoint paths, respectively, and \( T_{st}(q) \) be the cost of the allocation needed to meet demand and protection requirements between \( s \) and \( t \) for a value of \( q \).

**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 an optimal solution meeting working and partial protection requirements with \( f_0 = (1-2q) \) and \( f_1 = f_2 = q \); giving a total cost \( T_{st}(q) = (1-2q)p_0 + q(p_1 + p_2) \), where \( p_0 \) is the shortest path and paths 1 and 2 are the shortest pair of disjoint paths.

*Proof:* The modified linear program \( LP_{q \leq 0.5} \) seeks to find a minimum-cost routing with capacitated edges. This is a classic minimum-cost flow problem [12] that has known algorithms for finding the optimal solution. We use the shortest successive paths algorithm [12] to solve \( LP_{q \leq 0.5} \). The successive shortest paths algorithm will find the shortest path in the network between the specified source and destination, and route flow equal to the lowest capacity edge. Next, a residual graph is created from the routed flow. For edge \( \{i,j\} \) in a residual graph with edge capacities \( u_{ij} \), flow \( x_{ij} \), and cost \( c_{ij} \), the residual edge capacities are \( r_{ij} = u_{ij} - x_{ij} \) and \( r_{ji} = x_{ij} \). The latter is a “backwards” edge, and if flow is routed on this edge, it actually is removing flow from the forward direction. The cost of the backwards edge \( \{j,i\} \) is \(-c_{ij} \).

The cost of the shortest path in \( G \) is \( p_0 \), and it has capacity \((1-q)\), which is the case because all edges have equal capacity. The residual capacities are 0 for all edges in \( p_0 \) and \( q \) for all backwards edges created in the residual graph. Since all edges

\(^1\)It is possible that the shortest path is one of the pair of disjoint paths, in which case \( f_0 = (1-q) \) and \( f_2 = q \).
that can have flow routed on them have capacity \((1 - q)\) and \(q\), and since \(q \leq \frac{1}{2}\), the remaining flow of \(q\) can be routed with one more iteration of the successive shortest path algorithm. The next shortest path is found in the residual graph, and \(q\) is routed onto it. If that shortest path does not use any backwards edges, then it is disjoint from the first shortest path found and the allocations are \((1 - q)\) on the shortest path and \(q\) on the next path. Otherwise, the next shortest path uses backwards edges. Since the capacity of the backwards edges is \(q\), and we are routing \(q\), the backwards edges will be removed and the original “forward” edge will have \(q\) removed from it (with total allocation of \((1 - 2q)\)). Now the successive shortest paths algorithm finishes with a minimum-cost flow. To recover the paths, we use flow decomposition [12], which is to find a path from source to destination, and subtract the flow equivalent to the minimum edge capacity (almost a successive shortest paths in reverse). We first find the shortest path, which will have a minimum of \((1 - 2q)\) flow in either case of backwards edges used or not. We are now left with two disjoint paths of \(q\) flow each. These disjoint paths are by definition the minimum-cost pair of disjoint paths, since if there existed a lower cost pair of disjoint paths we can produce a lower cost flow by routing \((1 - 2q)\) onto the shortest path and \(q\) onto each of the lower cost pair of disjoint paths, which would give a lower cost routing. This is not possible since the successive shortest paths algorithm found the minimum-cost solution. Denote the cost of each of the shortest pair of disjoint paths as \(p_1\) and \(p_2\) respectively. We are left with a minimum-cost solution of \((1 - 2q)p_0 + q(p_1 + p_2)\).

V. Solution for \(q > \frac{1}{2}\)

When \(q \leq \frac{1}{2}\), no spare allocation is needed and the minimum-cost routing to meet partial protection requirements can be found for any general mesh network. When \(q > \frac{1}{2}\), it may be necessary to use spare allocation to meet protection requirements. Since the overall allocation of working plus spare 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 on general mesh networks. If we consider \(N\) disjoint paths between the source and destination, with the \(i^{th}\) path having cost \(p_i\), we see that this is equivalent to a two-node network with \(N\) links where the \(i^{th}\) link has cost \(p_i\). Hence, we start by investigating the properties of minimum-cost solutions for two-node networks in order to gain insight on solutions for general networks. Using these insights, a heuristic algorithm is developed in Section V-B for general mesh networks.

A. Results for Two-Node Networks

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 Figure 4 and let \(q = \frac{2}{3}\). Allocating a capacity of \(\frac{1}{3}\) onto each link does not use any spare capacity and has total cost of \(\frac{2}{3}(1 + 2 + 6) = 3\). In contrast, consider using the two lowest cost links, each with allocation \(\frac{2}{3}\). Clearly, 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 zero spare capacity allocation.

Fig. 4: Two-node network with link costs

For two-node networks, order the edges such that \(c_1 \leq c_2 \leq ... \leq c_N\). Define \(x_{ii}\) as the allocation on the \(i^{th}\) edge. From our analysis, we are able to define a value \(K\), which will be important for evaluating two-node networks: 
\[
K = \arg \max_{K=2...N} \left( \frac{1}{c_k} \sum_{i=1}^{K} c_i \right) \quad \text{demonstrated in 4}\)
\[
K \leq \frac{1}{K-1} 
\]

Proof: For a given set of edges \(E\) with edge cost \(c_i\) for the \(i^{th}\) edge, the linear program for a two node network (denoted as \(LP_2\)) can be written as follows:

\[
LP_2: \quad \min c'_N x_N \\
\text{s.t.} \quad A_N x_N \geq q e_N \\
\sum_{i \in E} x_i \geq 1 \\
x_i \geq 0, \quad \forall i \in E
\]

\[
A_N \text{ is the } N \times N \text{ matrix of } 1's \text{ with the identity matrix subtracted from it (an all 1's matrix with a diagonal of zeros). } e_N \text{ and } x_N \text{ are the cost and edge allocation vectors for } N \text{ edges, respectively. } e_N \text{ is a vector of } N \text{ 1's.}
\]

The dual to the above primal, denoted by \(LP_{2d}\), is as follows with \(p_i\) being the \(i^{th}\) dual variable:

\[
LP_{2d}: \quad \max \sum_{i=1}^{N} q p_i + p_{N+1}\]

\[
\text{s.t. } p'_{N+1} [A_N e_N] \leq c'_N \\
p_i \geq 0, \quad \forall i \in 1..(N+1)
\]

If spare allocation is needed, then the total flow is by definition greater than 1. In that case, equation 23 is not tight and can be disregarded. By complementary slackness, the dual variable corresponding to that constraint, \(p_{N+1}\), has to be zero. We will assume that spare allocation is needed for our preliminary analysis; we will come back to the case when spare allocation is not needed later.

Solving the primal, we know generally that \(K \leq N\) variables have will be greater than zero, and \(N - K\) variables will be zero.
For all variables that are zero, by complementary slackness their associated dual constraints are not tight. We can arrange the tight constraints of the dual as follows:

\[
LP_{2dK} : \text{max } \sum_{i=1..K} qp_i \\
\text{s.t. } p_K[A_K e_K] = c'_K \\
p_i \geq 0, \quad \forall i = 1..K
\]  

(15)  

(16)  

(17)

Solving the dual with \(K\) linearly independent constraints, we have \(K\) dual variables that are greater than zero, which means \(N - K\) constraints in the primal are not tight. We rewrite the primal as follows:

\[
LP_{2K} : \text{min } c'_K x_K \\
\text{s.t. } A_K x_K \geq q e'_K \\
x_i \geq 0, \quad \forall i = 1..K
\]  

(18)  

(19)  

(20)

If all inequalities in equation 19 are set to equality, we have \(K\) linearly independent equations and \(K\) variables. We can find a unique solution with even distribution of \(x_i\)'s for \(i = 1..K\). A quick check using complementary slackness verifies that this solution is in fact optimal. It is also straightforward to see that these \(K\) variables, corresponding to \(K\) edges, will in fact be the \(K\) lowest cost edges.

To complete the proof, we first assume that spare allocation is needed and \(q < \frac{K-1}{K}\). Then, \(q \frac{K}{K-1} < 1\), but we originally had the constraint \(\sum_{i \in E} x_i \geq 1\) in the primal. This constraint will become tight, and then the allocation across the edges will be exactly 1, which means no spare allocation is used. Now assume that spare allocation is not needed and \(q > \frac{K-1}{K}\). \(K\) is the number of non-zero variables when spare allocation is needed. So, with spare allocation not needed, we add back the \((K+1)^{th}\) constraint \(\sum_{i=1..K} x_i \geq 1\). This will in fact be equality since spare allocation is not needed. Summing the first \(K\) constraints in the modified primal \(LP_{2K}\) gives the following inequality: \(\sum_{i=1..K} x_i \geq q \frac{K}{K-1}\). Clearly, if \(q > \frac{K-1}{K}\), then \(q \frac{K}{K-1} > 1\), so the added constraint is not with equality and spare allocation must be used.

We now also need to show \(K = \arg\max_{K=2..N}(c_K \leq \frac{1}{K-1} \sum_{i=1}^K c_i)\), where \(K\) is the maximum number of links such that the incremental cost of using an additional link would not improve the solution.

**Lemma 4.** When \(q > \frac{K-1}{K}\), edge \(j\) is used if and only if \(c_j \leq \frac{1}{K-1} \sum_{i=1}^K c_i\).

**Proof:** We first show edge \(j\) is used if \(c_j \leq \frac{1}{K-1} \sum_{i=1}^K c_i\). Assume otherwise such that edge \(j\) is used and \(c_j > \frac{1}{K-1} \sum_{i=1}^K c_i\). From the proof for Lemma 3, we know the dual will have \(K\) tight constraints corresponding to an allocation on the \(K\) lowest cost edges. Solving the dual for variable \(p_j \geq 0\), we get \(p_j = \frac{1}{K-1} \sum_{i=1}^K c_i - c_j\). But from assumption, \(c_j > \frac{1}{K-1} \sum_{i=1}^K c_i\), which is a contradiction.

Next we show if \(c_j \leq \frac{1}{K-1} \sum_{i=1}^K c_i\), then edge \(j\) is used. Assume otherwise: \(c_j \leq \frac{1}{K-1} \sum_{i=1}^K c_i\) and edge \(j\) is not used. From assumption, \(x_j\) is not active and edges 1 to \(K\) are \((x_i > 0, \forall i = 1..K)\). Additionally, since the \(K\) lowest cost edges are used, \(c_j \geq c_i, \forall i = 1..K\). We know from the proof for Lemma 3 that the dual variables corresponding to the \(K\) active edges are \(p_i = 0, \forall i = 1..K\). Since edge \(j\) is not used, the associated dual constraint: \(\sum_{i=1}^K p_i + 0p_j < c_j\) is not tight. Summing the \(K\) tight constraints of the dual, we get \(c_j > \frac{1}{K-1} \sum_{i=1}^K c_i\), which is a contradiction.

Next, the exact edge allocations for a minimum-cost solution to meet partial protection requirements on a two-node network are defined. Lemma 5 states that when spare allocation is needed, edges 1 to \(K\) will have an equal allocation, and edges \(K+1\) to \(N\) will have no allocation. Lemma 6 shows that when spare allocation is not needed, the solution will use \(J \leq K\) edges. Lemma 6 also provides the allocations across the \(J\) edges.

**Lemma 5.** A minimum-cost allocation when \(q > \frac{K-1}{K}\) will be an even allocation of \(\frac{1}{K-1}\) on the \(K\) lowest cost edges, and no allocation on the remaining edges.

**Proof:** See proof for Lemma 3.

**Lemma 6.** The minimum-cost allocation when \(q \leq \frac{K-1}{K}\) will have a non-zero allocation on edges 1 to \(J\), where \(J\) is the integer satisfying \(\frac{1}{\frac{1}{N}} < q \leq \frac{1}{\frac{1}{J}}\). Moreover, the minimum-cost allocation when \(q \leq \frac{K-1}{K}\) is: \(x_i = (1-q), \forall i = 1..(J-1)\); \(x_j = (J-1)q - (J-2)\); \(x_i = 0, \forall i = (J+1)..N\).

**Proof:**

With spare allocation not needed, the total allocation across the edges is 1. The \(N + 1\) constraint in the primal needs to be reintroduced:

\[
LP_2 : \text{min } c'_N x_N \\
\text{s.t. } A_N x_N \geq q e'_N \\
x_i \geq 1 \\
x_i \geq 0, \quad \forall i \in E
\]  

(21)  

(22)  

(23)  

(24)

With the following dual:

\[
LP_{2d} : \text{max } \sum_{i=1..N} qp_i + p_{N+1} \\
\text{s.t. } p_{N+1}[A_N e_N] \leq c'_N \\
p_i \geq 0, \quad \forall i = 1..(N + 1)
\]  

(25)  

(26)  

(27)

We introduced back the variable \(p_{N+1}\) to the dual. Our previous solution, which has \(p_i \geq 0, \forall i = 1..K\) and \(p_i = 0, \forall i = (K+1)..N\) is still dual feasible if we set \(p_{N+1} = 0\). We wish to see if can increase the objective value of the dual when \(q < \frac{K-1}{K}\). So the initial feasible solution has \(K\) constraints tight. Each of these constraints has variables \(p_{K+1}\) to \(p_{N+1}\).
These are all zero in our initial feasible solution. So, to increase the new variable \( p_{N+1} \) by some value \( \delta \), we must decrease \( p_1 \) to \( p_K \) by at least \( \delta \). Let \( \pi_j \) be the amount to decrease \( p_j \) to increase \( p_i \) by \( \delta \). Since \( p_1 \) to \( p_K \) need to decrease by at least \( \delta \), we get \( \sum_{j=1}^{K} \pi_j \geq \delta \). Now consider the objective function of the dual with \( p_{N+1} \) increased by \( \delta \): \( \max q(\sum_{i=1}^{K} (p_i - \pi_i) + p_{N+1}) + \delta \).

Since it is a maximization, for a fixed value of \( \delta \), we want to minimize \( \sum_{i=1}^{K} \pi_i \). Any individual tight constraint \( j \) of the dual is \( \sum_{i=1}^{K} p_i - p_j \leq \epsilon_j \). These constraints are tight, so accounting for the decrease of \( p_1 \) to \( p_K \) to get \( p_{N+1} \) to be \( \delta \), we get the following linear program:

\[
\begin{align*}
\min \sum_{i=1}^{K} \pi_i \\
\text{s.t.} \ A_K \pi_K &\geq \delta \epsilon_K \\
\pi_i &\geq 0, \quad \forall i = 1..K
\end{align*}
\]

Here, \( \pi_K \) is the vector of variables \( \pi_i \) to \( \pi_K \). We do not write the condition \( \sum_{i=1}^{K} \pi_i = \delta \), since it is not active. We know the solution to this from the proof for Lemma 3, and that is \( \sum_{i=1}^{K} \pi_i = \delta \frac{K}{K-1} \). Looking at the objective function again, we see that in order for the variable \( p_{N+1} \) to increase (and be included in the basis), we get the condition \( \delta > \delta \frac{K}{K-1} \), which then becomes \( q < \frac{K}{K-1} \), which is exactly the condition when spare allocation is not needed.

We now increase \( \delta \) by increments, until one of the dual variables goes to zero. As shown above, all dual variables decrease by the same quantity. Solving for the dual variables when \( p_{N+1} = 0 \), we get \( p_j = \sum_{i=1}^{K} i = 1^K \epsilon_i - 2^K - 3 \epsilon_j \). So the dual variable with the lowest value will go to zero first. This is the dual corresponding to the edge with the highest cost. Currently the first \( K \) constraints are still tight, meaning the first \( K \) edges still have allocation, but \( p_K \) is now zero. So \( p_1 \) to \( p_{K-1} \) are greater than zero, and \( p_{N+1} = 0 \). We now want to see if we can further increase \( p_{N+1} \). We now consider the following objective function: \( \max q(\sum_{i=1}^{K} (p_i - \pi_i) + p_{N+1}) + \delta \). Clearly, using a similar process as above, we get the condition that we will increase \( \delta \) (which is \( p_{N+1} \)) only if \( q < \frac{K-2}{K-1} \). When this happens, the \( K^{th} \) constraint, that was previously still active when \( p_K \) went to zero, will now become inactive. This means that the \( K^{th} \) edge now has no allocation by complementary slackness. Inductively, we stop increasing \( p_{N+1} \) when we have \( J \) active dual variables such that \( \frac{2J+1}{J+1} < q \leq \frac{2J-1}{J} \). By complementary slackness, this means we have \( J \) active primal variables, which is \( J \) active edges. Additionally, we have \( J \) active dual variables, and this corresponds to \( J \) active constraints in the primal. We get the following set of \( J \) independent equations: \( x_j^p [A_{J-1} \epsilon_N] = [q \epsilon_N 1] \). We can solve this and obtain the results in Lemma 6.

B. 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 \( i^{th} \) path. As discussed in the beginning of the section, by treating the \( N \) paths in the general mesh network as a two-node network with \( N \) links, the results from Section V-A can be applied to develop a heuristic solution for general mesh networks for the case of \( q > \frac{1}{2} \). Recall that for \( q \leq \frac{1}{2} \), the optimal minimum-cost solution for general mesh networks was derived in Section IV.

The heuristic 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. These paths can be found using the Bhandari algorithm [11]. For each set of \( k \) disjoint paths, we look to see if spare allocation is needed, i.e. \( q > \frac{2k-1}{k} \), and use the minimum-cost allocation given by Lemmas 5 and 6. From the different possible disjoint path routings, the allocation of minimum-cost is chosen. We call this algorithm the Partial Protection Disjoint Path Routing Algorithm (PP-DPRA). Theorem 2 gives a bound on PP-DPRA’s performance.

Theorem 2. PP-DPRA produces a routing 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 \( \frac{1}{2}(p_1 + p_2) \). Doubling the allocation on each of the shortest pair of disjoint paths doubles the total cost to \( (p_1 + p_2) \), where \( p_1 \) and \( p_2 \) are the cost of using each of the shortest pair of disjoint paths. We note that this allocation is sufficient to provide protection for all \( q \leq 1 \); so the cost for protecting all \( q \leq 1 \) using a pair of disjoint paths is at most double that of \( q = \frac{1}{2} \). 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.

C. Comparison of PP-DPRA to the Minimum-Cost Solution

The PP-DPRA solution is compared to \( 1 + 1 \), \( 1 + q \), and \( LP_{DP} \). The simulation is identical to the one run in Section III-B, with PP-DPRA being implemented in C. The average costs to meet demand and protection requirements over all random graphs are plotted in Figure 5. Simulation results show that for \( q \leq \frac{1}{2} \), the routing as given by Theorem 1 matches the optimal routing produced by \( LP_{DP} \), and for \( q > \frac{1}{2} \), the average cost is greater than optimal by 1.4% on average. Additionally, on average, the running time for routing a demand with PP-DPRA was 0.001 seconds, while with the linear program \( LP_{DP} \), it was 22 seconds. This reduction in running time of four orders of magnitude makes the algorithm suitable for networks that require rapid setup times for incoming demands.

We note that [8] also developed an algorithm for meeting partial protection requirements by spreading capacity across disjoint paths. However, the algorithm in [8] was designed to minimize capacity over multiple connections, whereas the algorithms in this paper were designed to minimize costs for one connection at a time, making a direct comparison of the algorithms difficult.
VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper we have developed a mathematical model to provide deterministic partial protection for a single commodity. A linear program was formulated to find a minimum-cost solution in general mesh networks. Simulations show this LP offers significant savings over the most common protection schemes used today. A heuristic algorithm, PP-DPRA, was developed. Simulation results show that this algorithm comes within 1.4% of optimal on average and runs four orders of magnitude faster than the linear program.

An important direction for future research will be to consider the additional savings in protection capacity that can be achieved by resource sharing between demands. Currently, resources that could be used to protect multiple demands are potentially being underutilized by being dedicated to only one demand. Preliminary results show that significant savings can be achieved with protection resource sharing. A better understanding of how resources are shared will help develop more efficient algorithms for the partial protection problem.

REFERENCES