Coping with the Resilience - Congestion Tradeoff in Multipath Routing Schemes

Ron Banner and Ariel Orda
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 32000, Israel
{ banner@tx, ariel@ee .technion.ac.il

Abstract

Two major objectives of multipath routing schemes are congestion avoidance and resilience to failures. However, focusing on each objective alone severely deteriorates the quality of the other. More specifically, when multipath routing is employed for congestion avoidance, the traffic is distributed among several different paths, each potentially prone to network failures. Since any path failure results in the failure of the entire transmission, a "naive" use of multipath routing for load balancing may largely increase the vulnerability of the connections to network failures. On the other hand, when multipath routing is employed in order to improve the resilience to failures, most of the corresponding schemes focus on the establishment of pairs of *disjoint* paths. However, in many cases, this restrictive requirement may lead to the selection of poor routing paths that were not designed to transfer large volumes of data; thus, their employment may dramatically increase the congestion state of the network. In this work we establish efficient multipath routing schemes that incorporate the two fundamental objectives of congestion avoidance and resilience to failures. To the best of our knowledge, we are the first to relax the common requirement of disjoint paths that solely considers full (100%) protection to single failures, into a weaker requirement that can accommodate any degree (0%-100%) of protection i.e., any desired probability to survive a network failure. We incorporate this new concept in several polynomial schemes that enhance survivability while considering the resilience-congestion tradeoff. Then, we turn to consider the major weakness of multipath routing schemes that solely balance the network's load, namely, increased vulnerability to failures. Specifically, we establish a multipath routing scheme that minimizes the congestion state of the network while satisfying some essential reliability requirements. As the corresponding problem is shown to be intractable, we establish an efficient ε-optimal approximation scheme.

1. Introduction

Practical routing schemes typically focus on discovering a single "optimal" path for routing, according to some desired metric. Accordingly, traffic is always routed over a single path, which often results in substantial waste of network resources. *Multipath Routing* is an alternative routing scheme that distributes the traffic among multiple "good" paths instead of routing all traffic along a single "best" path.

Multipath routing can significantly reduce congestion in "hot spots", by deviating traffic to unused network resources, thus improving network utilization and providing load balancing [13]. Moreover, congested links usually result in poor performance and high variance. For such circumstances, multipath routing can offer steady and smooth data streams [4].

Multipath routing can also improve the network's survivability i.e., its ability to maintain service continuity in the presence of failures. The main idea is to route duplicates of the original data flow along several (usually two) disjoint paths. In normal operation mode, the receiver picks to use the better path and discards data from the other. Then, upon a failure in the active path, the inferior path is selected in order to restore the damaged traffic. Such a survivability technique, usually referred to as *1+1 protection*, allows fast recovery upon detection of a failure. In a more capacity-efficient approach (albeit with a significant larger recovery time), referred as a *1:1 protection*, data is sent only on the active path; the backup path is activated only if upon a failure in the active path [16].

A major problem that arises in the use of multipath routing for load balancing is increased vulnerability to failures. Indeed, when traffic is splitted among multiple paths, a failure in each routing path results in a failure of the entire transmission. For example, if the traffic is carried by n disjoint paths, each having a success probability p, then the success of the transmission is radically (exponentially) reduced to p^n. Moreover, if these n disjoint paths have different success probabilities, then it follows that the transmission success is dominated by the path that has the smallest success probability. Therefore, it is essential for such multipath schemes to exclusively use paths with high success probabilities.

At the same time, when multipath routing is used in order to enhance survivability to failures, the congestion state of the network may severely increase. Indeed, the restrictive requirement of protection schemes that assign traffic *only* to pairs of *disjoint paths* may be in many cases infeasible (where there are no such paths) and in other cases very limiting [17]. In the latter case, this often leads to the selection of inferior routing paths that were not designed to carry large volumes of data. Moreover, the focus on disjoint paths, used by existing protection schemes, solely considers full (100%) protection to single failures; i.e., only two degrees of survivability are provided, namely 100% or 0% protection. However, in practice, a much "softer" criterion for survivability, which enables to control various important tradeoffs, should be employed [17],[18] .

In this study we investigate ways to efficiently cope with the tradeoff between the degree of survivability and the level of congestion in multipath routing schemes. More specifically, we address two main issues. First, we address the requirement of

multipath routing schemes that balance the network's load to use only sufficiently reliable paths. To that end, we establish a multipath routing scheme that minimizes the congestion while using only paths that satisfy a given end-to-end reliability requirement. Then, we turn to consider protection schemes that allow providing *various degrees of survivability* while considering the network's load. To that end, we introduce a new concept that relaxes the restrictive path disjointedness criterion of current protection schemes. To the best of our knowledge, we are the first to relax this standard path disjointedness criterion (which solely provides 0% and 100% survivability) into a *continuous survivability criterion*, which provides various degrees of survivability.

We adopt the widely used single link failure model [9],[15],[18],[23]. Moreover, as the quality of a protection scheme can only be measured in the presence of failures, we assume that a failure event has taken place. Then, we evaluate the degree of survivability of each connection according to the probability that the connection continuously maintains the service after the failure. More specifically, given a failure event, we determine the connection's degree of survivability according to its probability to survive that failure i.e., the probability that either the active path or the backup path has survived the failure.

Example: Consider the connection described in Fig. 1.1. Assume that the upper and lower paths are the connection's active and backup paths, respectively. In addition, assume that a *failure probability* p_i is associated with every link $e_i \in E$. Given the event of a link failure, we measure the connection's degree of survivability i.e., the probability of the connection to survive that failure.

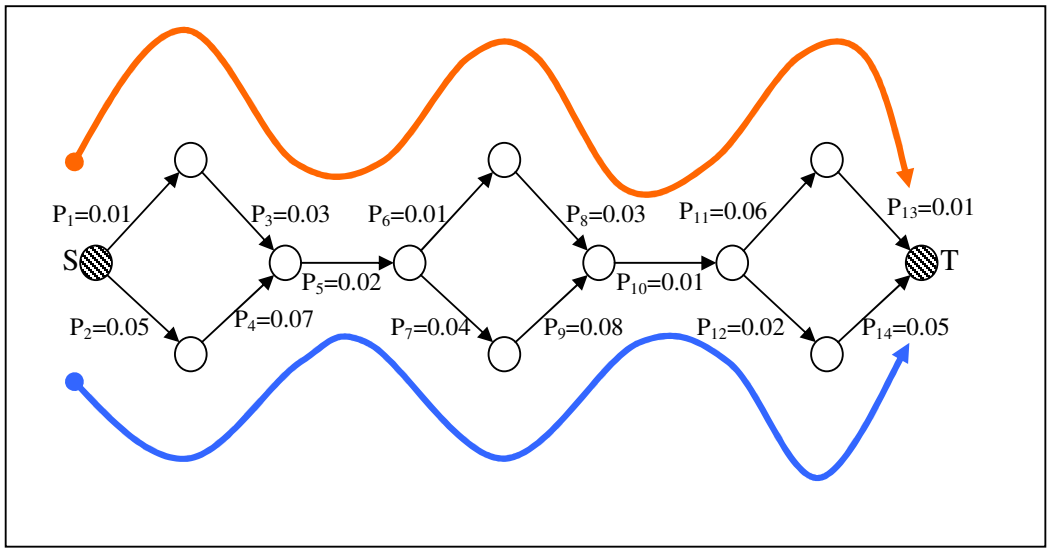

Fig.1.1: A survivable connection

To that end, we first determine the failure probability of each link in the presence of a failure i.e., the failure probability of each link given that a failure event has already taken place in the network. For the single link failure model, this conditional probability is:

$\widetilde{p_e} \triangleq p\left(\text{link } e \text{ has failed} \middle| \text{a failure event has taken place in the network}\right) = \frac{p_e}{1-\prod_{e' \in E}\left(1-p_{e'}\right)}$.

Next, we calculate the probability that the connection fails under the assumption of a network failure. To that end, we observe that, if the failure occurred at one of the links in the set $\{e_1, e_2, e_3, e_4, e_6, e_7, e_8, e_9, e_{11}, e_{12}, e_{13}, e_{14}\}$, then the connection has survived the failure. However, if the failure occurred at either link e_5 or link e_{10}, then the connection is broken. Thus, the probability that the given connection survives a failure is equal to the probability that both links e_5 and e_{10} survives it, namely: $\left(1-\widetilde{p_5}\right)\cdot\left(1-\widetilde{p_{10}}\right) = 0.95 \times 0.97 = 0.92$. Thus, we conclude that, upon a link failure, the connection in this example survives the failure with a probability of at least 0.92.

This example illustrates that, unlike the *inflexible* requirement of path disjointedness, the idea of continuous survivability provides a *flexible* criterion for choosing the desired degree of survivability (e.g. 92%-survivability); consequently, it enables to consider *tradeoffs* for practical protection schemes (e.g., resilience vs. congestion, the resilience vs. delay). In addition, it *precisely expresses* the preferred level of survivability and, in contrast to the standard survivability technique of path disjointedness, it enables to provide *accurate control* on the survivability restrictions.

We employ the concept of continuous survivability in order to design protection schemes that consider the resilience-congestion tradeoff. More specifically, we design polynomial protection schemes for 1+1 protection and 1:1 protection such that, for any $0 \le p \le 1$, establish p-survivable connections that minimize the network's load. Then, we show that the concept of continuous survivability leads to an efficient third protection architecture, which is an hybrid between 1:1 protection and 1+1 protection; this new architecture is shown to admit an efficient polynomial solution. Finally, we show that all proposed schemes can be enhanced in order to consider QoS requirements. In addition, we prove that, under the single link failure model, we cannot improve the quality of our solutions if we admit more than two paths for each survivable connection; thus, our solutions are optimal with respect to more general protection frameworks that admit *any* (≥ 2) number of paths.

The rest of this paper is organized as follows. In section 2, we introduce some terminology and definitions, and formulate the main problems considered in this study. In section 3, we present some background and survey related previous work. In section 4, we establish routing schemes that incorporate the continuous survivability concept. In section 5, we consider multipath routing schemes with end-to-end reliability constraints. Finally Section 6 summarizes the results and discusses future research directions.

2. Model and Problems Formulation

This section formulates the general model and main problems addressed in this study. We begin with the following general definitions.

A *network* is represented by a directed graph $G(V,E)$, where V is the set of nodes and E is the set of links. Let $N=|V|$ and $M=|E|$. A *path* is a finite sequence of nodes $p=(v_0,v_1,\cdots,v_h)$, such that, for $0\le n\le h-1$, $(v_n,v_{n+1})\in E$. A path is *simple* if all its nodes are distinct. A *cycle* is a path $p=(v_0,v_1,\cdots,v_h)$ together with the link $(v_h,v_0)\in E$ i.e., $(v_0,v_1,\cdots,v_h,v_0)$. Denote the set of all cycles in a network G by $T(G)$.

A *commodity* is a pair of nodes $(i,j)\in V\times V$ that is assigned with a non-negative demand $\gamma^{(i,j)}$. Let β be the set of all commodities with *positive* demand $\beta=\left\{(i,j)\middle|(i,j)\in V\times V,\gamma^{(i,j)}>0\right\}$. Given a commodity $(i,j)\in V\times V$, we say that node i is the *source node* of the given commodity and node j is its *target node*. If $|\beta|\le 1$, we say that the network has a *single commodity* flow demand. Otherwise, we say that the network has a *multi-commodity* flow demand.

The set $P^{(i,j)}$ is the collection of all directed paths from the source i to the destination j in the network. In addition, let $P\triangleq\bigcup_{(i,j)\in V\times V}P^{(i,j)}$, and let $P_{simple}^{(i,j)}\subseteq P^{(i,j)}$ represent the set of *simple paths* from i to j in the network. Finally, for each path $p\in P^{(s,t)}$ and link $e\in E$, $\Delta_e(p)$ counts the number of occurrences of the link e in the path p. For example, given a non-simple path $p=(v_0,v_1,v_2,v_3,v_1,v_2,v_4)$ and a link $e=(v_1,v_2)$, we have $\Delta_e(p)=2$.

Each link $e\in E$ is assigned a *weight* $w_e\in\mathbb{Z}^+$, a *capacity* $c_e\in\mathbb{Z}^+$ and a *failure probability* $p_e\in[0,1]$. We consider a *link state* routing environment, where each source node has a (precise) image of the entire network.

Definition 2.1 Given a (non-empty) path p, its weight $W(p)$ is defined as the sum of weights of its links, namely, $W(p)=\sum_{e\in p}w_e$.

Definition 2.2 Given a (non-empty) path p, its capacity $C(p)$ is defined as the capacity of its bottleneck link, namely, $C(p)=\underset{e\in E}{Min}\{c_e\}$.

Definition 2.3 Given a (non-empty) path p, its reliability $\Pi(p)$ is defined as the multiplication of the success probabilities of its links, namely, $\Pi(p)=\prod_{e\in p}(1-p_e)$.

Definition 2.4 Let $G(V,E)$ be a network. A *path flow* is a real-valued function $f:P\to\mathbb{R}^+\cup\{0\}$ that satisfies the following two properties:

Capacity constraints: For each $e \in E$, $\sum_{p \in P} \Delta_e(p) \cdot f(p) \le c_e$.

Flow demand: For each commodity $(i,j) \in V \times V$, $\sum_{p \in P^{(i,j)}} f(p) \ge \gamma^{(i,j)}$.

Definition 2.5 Given is a path flow $f : P \to \mathbb{R}^+ \cup \{0\}$ over a network $G(V,E)$. A *link flow of a commodity* $(i,j) \in V \times V$ is a real-valued function $f : E \times V \times V \to \mathbb{R}^+ \cup \{0\}$ that satisfies, for each link $e \in E$: $f_e^{(i,j)} \triangleq \sum_{p \in P} \Delta_e(p) \cdot f(p)$.

Denote $f_e \triangleq \sum_{(i,j) \in V \times V} f_e^{(i,j)}$.

Definition 2.6 Let $G(V,E)$ be a network. A *cycle flow of a commodity* $(i,j) \in V \times V$ is a real-valued function $f : T(G) \times V \times V \to \mathbb{R}^+ \cup \{0\}$.

Note that, for a given link flow of a commodity $(i,j) \in V \times V$, the path flow $f : P^{(i,j)} \to \mathbb{R}^+ \cup \{0\}$ is not necessarily unique. In addition, the path flow representation (of size $O(|P|)$) may have exponential size (with respect to the network representation). However, it follows from the flow decomposition theorem [1] that any path flow assigned to a commodity $(i,j) \in V \times V$ (i.e. the collection of pairs $(p, f(p))$ for each $p \in P^{(i,j)}$) has a corresponding path flow with at most M paths and cycles with a positive flow that share the same link flow representation.

Definition 2.7 Given a network $G(V,E)$ and a link flow $\{f_e\}$, the value $\frac{f_e}{c_e}$ is the *link congestion factor*.

Definition 2.8 Given a network $G(V,E)$ and a link flow $\{f_e\}$, the *network congestion factor* is the largest link congestion factor in the network, i.e., $\max_{e \in E} \left\{ \frac{f_e}{c_e} \right\}$.

As noted in [2],[13],[22], the network congestion factor provides a good indication of congestion.

Achieving a minimum network congestion factor minimizes the utilization of the most utilized link in the network. Observe that minimizing this value enables to successfully transfer the largest flow demand without violating the capacity constraint of any link. Hence, this problem is equivalent to maximizing the total throughput and can be efficiently solved using well-known solutions to the Maximum Flow Problem [1]. In [3], we formally establish this equivalence between these two problems.

Finally, we formalize some survivability concepts. To that end, we first establish the following elementary terminology. A link is classified as *faulty* upon its failure; it

remains faulty until it is *repaired*. We say that a link $e \in E$ is *operational* if it is not faulty. Likewise, we say that a path *p* is *operational* if it has no faulty link i.e., for each $e \in p$, link *e* is operational.

Definition 2.9 Given a network $G(V,E)$ and a pair of nodes *s* and *t*, a *survivable connection* is a pair of paths $(p_1, p_2) \in P^{(s,t)} \times P^{(s,t)}$.

Definition 2.10 Given are a network $G(V,E)$ and a survivable connection (p_1, p_2). We say that the connection (p_1, p_2) is *operational* if either p_1 or p_2 are operational.

The following definition quantifies the quality of a survivable connection. Since protection schemes focus on the capability to *recover* from network failures, the quality of survivable connections is determined according to their probability to remain operational *upon* network failures. Since we assume a single link failure model, we determine the quality of a survivable connection as the connection's probability to remain operational *following a single failure event*. This is formalized as follows.

Definition 2.11 Given are a network $G(V,E)$, a failure probability $p_e \geq 0$ for each link $e \in E$, and a survivable connection (p_1, p_2). We say that (p_1, p_2) is *p-survivable connection* if, upon a link failure, it remains operational with a probability of at least *p*.

We are now ready to formulate the main problems considered in this study, which minimize the network congestion factor subject to different considerations. We saw that, in the single commodity case, minimizing the network congestion factor is equivalent to maximizing the total throughput. Therefore, all these problems remain equivalent for the single commodity case if we consider a different objective criterion, namely that of maximizing the throughput. However, as opposed to the multi-commodity case, minimizing the network congestion factor is well defined also for the multi-commodity cases.

We proceed to present the first problem. We are given a network and a flow demand to be transferred through a survivable connection. The goal is to route two duplicates of the original flow demand along two paths that constitute the most survivable connection, while keeping the congestion of the network below some specified level. This is formulated as follows.

Problem MSC (Most Survivable Connection Problem) Given are a network $G(V,E)$, a pair of nodes *s* and *t*, a congestion constraint $\alpha \geq 0$, a demand $\gamma \geq 0$, and, for each link $e \in E$, a capacity $c_e \geq 0$ and a failure probability $p_e \geq 0$. Find a survivable connection $(p_1, p_2) \in P^{(s,t)} \times P^{(s,t)}$ with the maximum probability to remain operational upon a link failure, such that assigning the demand γ to the paths p_1 and p_2 produces a network congestion factor of at most α.

We shall establish an efficient polynomial solution for Problem MSC. We shall also show that this solution can be employed in order to define a pair of *maximally link disjoint paths* [19],[20] i.e., a pair of paths that have a minimum number of common links. Finally, note that, in cases where no pair of disjoint paths exists in the network, it is hard to identify the paths that together form the most survivable connection. In such cases, the solution to Problem MSC can determine the most "resilient" pair of paths, by setting the congestion constraint to infinity i.e., $\alpha \leftarrow \infty$.

Although Problem MSC establishes the most survivable connection subject to a congestion constraint, in practice it is often preferable to have a reasonable degree of survivability and a minimum level of congestion. Accordingly, we formulate the following problem.

Problem LCSC (Least Congested Survivable Connection Problem) Given are a network $G(V,E)$, a pair of nodes *s* and *t*, a survivability constraint $p \geq 0$, a demand $\gamma \geq 0$, and, for each link $e \in E$, a capacity $c_e \geq 0$ and a failure probability $p_e \geq 0$. Find a *p*-survivable connection $(p_1, p_2) \in P^{(s,t)} \times P^{(s,t)}$ such that assigning the demand γ to the paths p_1 and p_2 produces the minimal network congestion factor.

We shall later establish a polynomial solution for Problem LCSC.

As can be seen both problems MSC and LCSC focus on the 1+1 protection architecture. However, in Section 4, we reformulate all these problems to correspond to the 1:1 protection as well.

Problems MSC and LCSC focus on survivability, and their solutions enable the use of multipath routing in order to increase the tolerance to network failures. However, as was already explained in the Introduction, multipath routing bares another major benefit, namely load balancing. Accordingly, we turn to consider the increased vulnerability to failures when multipath routing is employed in order to balance the network's load. As was already explained in the Introduction, the use of multipath routing for such purposes must be made through highly reliable paths. Specifically, we present the following problem that seeks a path flow that minimizes the congestion under end-to-end reliability requirements.

Problem ReMP (Reliable Multipath) Given are a network $G(V,E)$, for each link $e \in E$, a failure probability $p_e > 0$ and a capacity $c_e > 0$, and, for each commodity $(i,j) \in V \times V$, a demand $\gamma^{(i,j)} > 0$ and a success probability restriction $\Pi^{(i,j)}$. Find a path flow $f : P \to \mathbb{R}^+ \cup \{0\}$ that minimizes the network congestion factor such that, if $p \in P^{(i,j)}$ and $f(p) > 0$, then $\prod_{e \in p} (1 - p_e) \geq \Pi^{(i,j)}$.

Note that, whereas Problems MSC and LCSC focused on the ability to *recover from network failures*, Problem ReMP considers how to *avoid network failures* by routing only over reliable paths. In Section 5, we show that Problem ReMP is computationally intractable and establish an efficient $\varepsilon-$optimal approximation scheme.

3. Background and Related Work

In recent years, transmission capabilities have increased to rates of 10 Gbit/s and behind [17]. With this increase, any failure may lead to a vast amount of data loss. To deal with such situations, pre-planning a protection path with a sufficient bandwidth for each working path has been widely accepted as an efficient effective strategy [12].

Protection schemes that enhance network's survivability have been extensively studied in recent years [5],[8],[11],[15]. Many studies have proposed end-to-end protection against single link failures by employing the restrictive path disjointedness survivability criterion, which enables a quite limited selection (if any) of routing paths. Other studies suggested a "*local protection*" strategy. The latter approach divides the working path into several segments and protects each such segment using a protection path segment. While this approach increases the range of feasible solutions, it takes extra signaling effort and imposes high requirements on the hardware responsiveness [12].

The idea to provide various degrees of survivability (as opposed to either 0% or 100%) was already considered in [9],[10],[18]. However, in contrast to our study they still require that paths be disjoint. The additional flexibility is obtained there by allowing a restricted number of connections to have unavailable backup paths when a fault occurs. Thus, a restricted (controlled) portion of the bandwidth of the backup paths is saved for other purposes. While this is a potentially useful alternative for providing a continuous survivability, no systematic approach with proven guarantees for general topologies has been considered. Rather, the latter model was mainly considered for ring networks and attained mostly heuristic solutions. In addition, it entirely corresponds to 1:1 protection architectures, and cannot be employed in architectures of 1+1 protection.

Some studies did relax the requirement of path disjointedness [19],[20] into the related concept of *maximally disjoint paths*. Specifically, a pair of paths from a source to a destination is said to be *maximally link disjoint* if the number of links common to both is minimum. This was applied by [19],[20] in order to improve the blocking rate of connection-oriented networks that support calls with multiple QoS requirements. We will later show that our approach, of establishing *p*-survivable connections, can be used in order to establish such pairs of paths.

4. Solution to Problems MSC and LCSC

In this section we present polynomial solutions to the problems that consider the tradeoff between the degree of survivability and the level of congestion. To that end, we first explore some properties of survivable connections.

4.1 Properties of Survivable Connections

In section 2, we defined a survivable connection as a pair of paths $(p_1, p_2) \in P^{(s,t)} \times P^{(s,t)}$. Also, we defined such a connection to be operational if either p_1 or p_2 are operational paths i.e., at least one of the paths is functioning correctly.

Therefore, since we consider the single link failure model, it follows that a link that is not common to both paths can never cause a survivable connection to break. On the other hand, if there is a common link that is not operational, then the entire connection is not operational. We summarize this in the following corollary.

Corollary 4.1 A survivable connection (p_1, p_2) is operational *iff* for each $e \in p_1 \cap p_2$ it holds that e is an operational link i.e., all the links that are common to both paths are operational.

An important consequence of Corollary 4.1 is that the probability that a survivable connection remains operational upon a link failure is equal to the probability that all its common links are operational upon that failure. Therefore, since we assume that link failure probabilities are independent, it follows that the probability of a survivable connection to survive a failure is equal to the product of all conditional success probabilities of the links that are common to both paths.

Corollary 4.2 Given are a survivable connection (p_1, p_2) and, for each $e \in E$, its failure probability $\widetilde{p_e}$ under the condition of a failure event. The probability that (p_1, p_2) is operational upon a failure is equal to $\prod_{e \in p_1 \cap p_2} \left(1 - \widetilde{p_e}\right)$.

Given the conditional failure probability $\widetilde{p_e}$ under an event of a failure, Corollary 4.2 provides a quality measure for survivable connections. However, standard statistics regarding failure probabilities usually consist of merely records of *unconditional* probabilities. Therefore, we express the value of all conditional failure probabilities $\left\{\widetilde{p_e}\right\}$ (defined in Corollary 4.2) in terms of the available unconditional failure probabilities $\{p_e\}$, namely $\widetilde{p_e} = \dfrac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}$ for each $e \in E$.

The following corollary summarizes the above discussion and provides a simple way to determine the quality of survivable connections.

Corollary 4.3 Given are a survivable connection (p_1, p_2), and for each $e \in E$, a failure probability p_e. The probability that (p_1, p_2) is operational within an event of a failure is equal to $\prod_{e \in p_1 \cap p_2} \left(1 - \dfrac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right)$.

4.2 Problem MSC

In this section we present a polynomial solution to Problem MSC i.e., the problem that seeks the most survivable connection while producing a congestion of at most α when a duplicate of the original demand is assigned to each of the connection's path.

Before we introduce the formal description of the algorithm, we explain its main idea. The algorithm is given a network $G(V,E)$, a pair of nodes *s* and *t*, a congestion constraint $\alpha \geq 0$, a demand $\gamma \geq 0$, and, for each link $e \in E$, a capacity $c_e \geq 0$ and a failure probability $p_e \geq 0$. The algorithm reduces this instance of Problem MSC into an instance of the Min Cost Flow problem. In essence, the reduction is based on a network transformation that considers three different cases, as illustrated in Fig 4.1. In the case of a link $e \in E$ with a capacity c_e that satisfies $\alpha \cdot c_e < \gamma$, it is easy to see that link *e* cannot be used by any path that transfers γ flow units without violating the congestion constraint α i.e., without having link utilization larger than α. Therefore, this link can be discarded from the network without any influence on the optimal solution. On the other hand, each link $e \in E$ that satisfies $\alpha \cdot c_e \geq 2 \cdot \gamma$ can be used by both of the connection's paths in order to transfer γ flow units over each path without violating the congestion constraint α. In that case, the corresponding link is transformed into two parallel links, each with a capacity γ; however, whereas the first is assigned with a zero weight the other link is assigned with a weight that is a function of the link's failure probability. The reason for that stems from corollary 4.3 that shows that the degree of survivability of each connection is solely determined by its common links. More specifically, only when both of the connection's paths share the same link *e*, the link's failure probability p_e should be considered. Indeed, a Min Cost Flow over the constructed network ensures that, when a single path traverses link *e*, the incurred cost is zero, whereas when both paths traverse through *e*, the cost is $f(p_e)$. The third case corresponds to links that satisfy $\gamma \leq \alpha \cdot c_e < 2 \cdot \gamma$. In that case, at most one path can traverse through such a link without violating the congestion constraint α. Thus, these links can be transformed into links that have a capacity γ without any effect on the optimal solution. In addition, since these links can be used by at most one path, their failure probability should not be considered and therefore the transformed links are assigned zero weight.

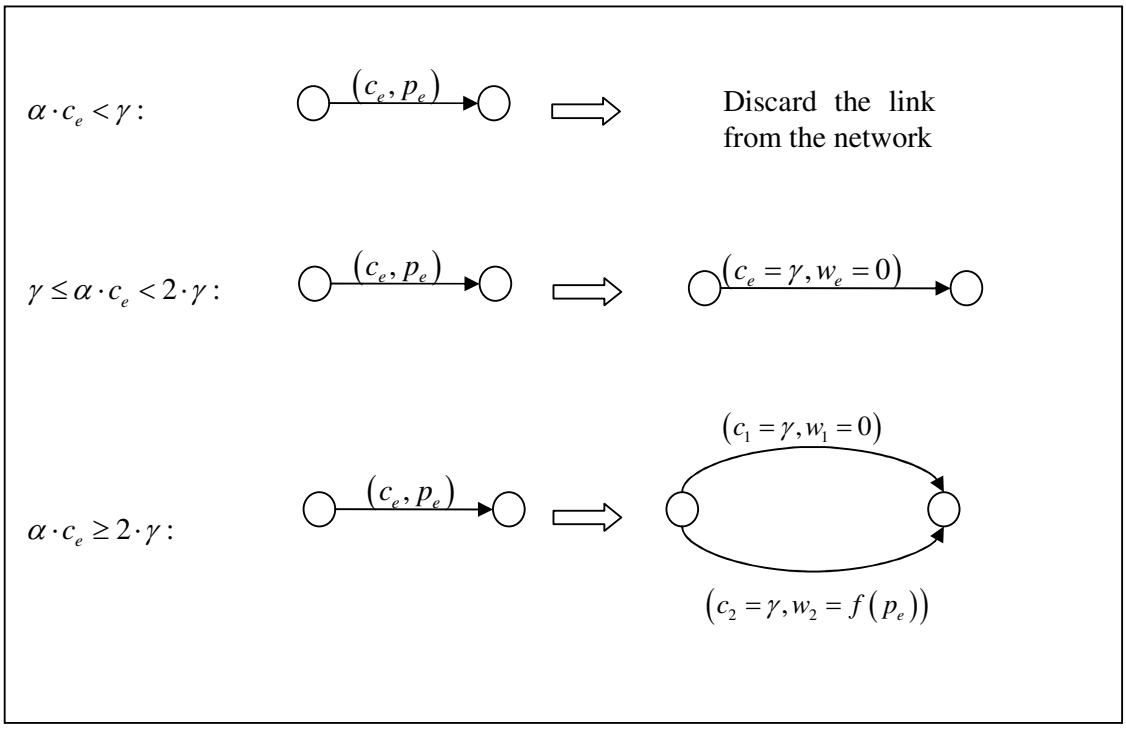

Fig 4.1: Reducing Problem MSC to the Min Cost Flow Problem

Denote the transformed network as $\widetilde{G}\left(\widetilde{V},\widetilde{E}\right)$. The algorithm identifies a min cost flow with a flow demand of $2\cdot\gamma$ flow units over the network $\widetilde{G}\left(\widetilde{V},\widetilde{E}\right)$ by employing any standard Min Cost Flow algorithm that returns an integral link flow when all capacities are integral. Since all link capacities in $\widetilde{G}\left(\widetilde{V},\widetilde{E}\right)$ are integral in γ, the returned link flow is γ-integral. Therefore, since the total traffic equals to $2\cdot\gamma$ flow units, the *flow decomposition algorithm* [1] can be applied in order to decompose the γ-integral link flow into a flow over two paths p_1, p_2 such that each carry γ flow units from *s* to *t*. Moreover, since the flow has minimum cost, it follows that $\sum_{e\in\widetilde{E}} f_e \cdot w_e = \sum_{e\in p_1\cap p_2} \gamma\cdot f\left(p_e\right) = \gamma\cdot \sum_{e\in p_1\cap p_2} f\left(p_e\right)$ has minimum value. Thus, $\sum_{e\in p_1\cap p_2} f\left(p_e\right)$ has minimum value. Finally, if we define $f\left(p_e\right) \triangleq -\ln\left(1-\frac{p_e}{1-\prod_{e'\in E}\left(1-p_{e'}\right)}\right)$ for each $e\in E$, the algorithm defines a pair of paths p_1, p_2 that minimizes $-\sum_{e\in p_1\cap p_2} \ln\left(1-\frac{p_e}{1-\prod_{e'\in E}\left(1-p_{e'}\right)}\right) = -\ln\prod_{e\in p_1\cap p_2}\left(1-\frac{p_e}{1-\prod_{e'\in E}\left(1-p_{e'}\right)}\right)$ and therefore maximizes $\ln\prod_{e\in p_1\cap p_2}\left(1-\frac{p_e}{1-\prod_{e'\in E}\left(1-p_{e'}\right)}\right)$. Thus, the connection $\left(p_1, p_2\right)$ maximizes

$\prod_{e \in p_1 \cap p_2} \left(1 - \frac{p_e}{1 - \prod_{e' \in E} (1 - p_{e'})} \right)$ which, according to Corollary 4.3, equals to the probability that the connection (p_1, p_2) is operational upon a failure. The formal description of the algorithm appears in Fig. 4.2.

Algorithm MSC $\left(G,(s,t),\{c_e\},\{p_e\},\alpha\right)$

Parameters:

$G(V,E)$ – network

(s,t) – commodity

$\{c_e\}$ – capacities

$\{p_e\}$ – failure probabilities

α – congestion constraint

Variables:

$\widetilde{G}\left(\widetilde{V},\widetilde{E}\right)$ – network

$\left\{c_{\tilde{e}}\right\}$ – capacities

$\left\{w_{\tilde{e}}\right\}$ – weights

$\left\{f_{\tilde{e}}\right\}$ – link flow

$\widetilde{\gamma}$ – demand

1. Construct an instance $\left\langle \widetilde{G}\left(\widetilde{V},\widetilde{E}\right),\left(\tilde{s},\tilde{t}\right),\left\{c_{\tilde{e}}\right\},\left\{w_{\tilde{e}}\right\},\widetilde{\gamma}\right\rangle$ of the Min Cost Flow Problem as follows:
   a. $\widetilde{V} \leftarrow V$.
   b. For each link $e: u \to v \in E,\ \alpha \cdot c_e \in [\gamma, 2\cdot\gamma)$ construct a link $\tilde{e}$ between $\tilde{u}$ and $\tilde{v}$. Assign it with a capacity γ and a zero weight.
   c. For each link $e: u \to v \in E,\ \alpha \cdot c_e \geq 2\cdot\gamma$:
      - Construct a *cheap* link $\widetilde{e_1}$ between $\tilde{u}$ and $\tilde{v}$. Assign it with a capacity γ and a zero weight.
      - Construct an *expensive* link $\widetilde{e_2}$ between $\tilde{u}$ and $\tilde{v}$. Assign it with a capacity γ and a weight of $-\ln\left(1-\dfrac{p_e}{1-\prod_{e'\in E}(1-p_{e'})}\right)$.
   d. $\tilde{\gamma} \leftarrow 2\cdot\gamma$.
2. Solve instance $\left\langle \widetilde{G}\left(\widetilde{V},\widetilde{E}\right),\left(\tilde{s},\tilde{t}\right),\left\{c_{\tilde{e}}\right\},\left\{w_{\tilde{e}}\right\},\tilde{\gamma}\right\rangle$ of the Min Cost Flow Problem using the *Cycle Canceling Algorithm* [1].
3. If there is no feasible solution for the instance, then return Fail. Otherwise let $\left\{f_{\tilde{e}}\right\}$ represent the solution.
4. Construct a link flow $f: E \to \{0,\gamma,2\cdot\gamma\}$ as follows:
   - For each link $e: u \to v \in E,\ \alpha \cdot c_e \in [\gamma, 2\cdot\gamma)$ set $f_e \leftarrow f_{\tilde{e}}$.
   - For each link $e: u \to v \in E,\ \alpha \cdot c_e \geq 2\cdot\gamma$ set $f_e \leftarrow f_{\tilde{e}_1} + f_{\tilde{e}_2}$.
5. Employ the *Flow Decomposition Algorithm* [1] over link flow $\{f_e\}$, in order to obtain a path flow $f: P \to \{0,\gamma\}$. Let paths p_1, p_2 represent the pair of paths that carry γ flow units.
6. Return connection (p_1, p_2).

Fig. 4.2 Algorithm MSC

Definition 4.1 We say that Algorithm MSC *succeeds* whenever it does not return Fail.

Theorem 4.1 If Algorithm MSC succeeds for the input $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$, then the returned connection is an optimal solution for the instance $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$ of Problem MSC.

Proof Since Algorithm MSC succeeds for the input $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$, it follows by construction that there is a feasible solution for the Min Cost Flow instance $\langle \widetilde{G}(\widetilde{V},\widetilde{E}),(\tilde{s},\tilde{t}),\{c_{\tilde{e}}\},\{w_{\tilde{e}}\},\widetilde{\gamma}\rangle$ of Step 1. Therefore, since the capacities and the demand of the min cost flow instance are integral in γ, it follows that the output of the Cycle Canceling Algorithm (link flow $\{f_{\tilde{e}}\}$), executed in step 2, is integral in γ (see [1]). Moreover, since all links in the network $\widetilde{G}(\widetilde{V},\widetilde{E})$ have a capacity of γ, it follows that, with the link flow $\{f_{\tilde{e}}\}$ each link transfers either zero or γ flow units. Thus, as $\widetilde{\gamma}=2\cdot\gamma$, it follows by the flow decomposition theorem [1], that link flow $\{f_{\tilde{e}}\}$ can be decomposed into a pair of paths in $\widetilde{G}(\widetilde{V},\widetilde{E})$ that corresponds to a pair of paths $p_1,p_2\in P^{(s,t)}$ in $G(V,E)$, such that each transfers γ flow units. Finally, by construction, it is easy to see that the flow over the paths $p_1,p_2\in P^{(s,t)}$ does not violate the congestion constraint α. We only need to show that (p_1,p_2) has the maximum probability to remain operational upon a link failure with respect to all connections that satisfy the congestion constraint α for a demand of γ flow units.

To that end, we employ the fact that the total cost of the link flow $\{f_{\tilde{e}}\}$ is minimal with respect to all link flows that transfer $2\cdot\gamma$ flow units from $\tilde{s}$ to $\tilde{t}$ in network $\widetilde{G}(\widetilde{V},\widetilde{E})$. Therefore, since we have shown that $f_{\tilde{e}}\in\{0,\gamma\}$ for each $\widetilde{e}\in\widetilde{E}$, it follows that $\sum_{\widetilde{e}\in\widetilde{E}} f_{\tilde{e}}\cdot w_{\tilde{e}}=\gamma\cdot\sum_{f_{\tilde{e}}\neq 0} w_{\tilde{e}}$ is minimal with respect to link flows that transfer $2\cdot\gamma$ flow units from $\tilde{s}$ to $\tilde{t}$; hence, $\sum_{f_{\tilde{e}}\neq 0} w_{\tilde{e}}$ is also minimal with respect to link flows that transfer $2\cdot\gamma$ flow units from $\tilde{s}$ to $\tilde{t}$.

Consider now an expensive link $\widetilde{e_x}:\tilde{u}\to\tilde{v}\in\widetilde{E}$ (as defined in Step 1.c), where $f_{\widetilde{e_x}}\neq 0$. It follows by construction that there exists a parallel cheap link $\widetilde{e_y}:\tilde{u}\to\tilde{v}\in\widetilde{E}$ such that $w_{\widetilde{e_y}}=0$. Therefore, since $\{f_{\tilde{e}}\}$ is a min-cost flow, it follows that $f_{\widetilde{e_y}}=\gamma$. Thus, it is easy to see that there exists a link $e_{xy}\in E$ that corresponds to the links $\widetilde{e_x}$ and $\widetilde{e_y}$ in $\widetilde{G}(\widetilde{V},\widetilde{E})$ such that $e_{xy}\in p_1\cap p_2$. Conversely, since p_1 and p_2 are

decomposed out of the link flow that corresponds to $\left\{f_{\tilde{e}}\right\}$, it follows that, if $e_{xy} \in p_1 \cap p_2$, then there exists a corresponding pair of parallel links $\widetilde{e_x}, \widetilde{e_y} \in \widetilde{E}$ such that $f_{\widetilde{e_x}} \neq 0, f_{\widetilde{e_y}} \neq 0$. Thus, it holds that $e_{xy} \in p_1 \cap p_2$ in network $G(V,E)$ *iff* the corresponding expensive link $\widetilde{e_x}$ in network $\widetilde{G}(\widetilde{V},\widetilde{E})$ satisfies $f_{\widetilde{e_x}} \neq 0$. Therefore, since the expensive links are the only non-zero cost links in $\widetilde{E}$, and since each expensive link $\tilde{e} \in \widetilde{E}$ that corresponds to the link $e \in E$ has a cost $w_{\tilde{e}} = -\ln\left(1 - \frac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right)$, it follows that $\sum_{f_{\tilde{e}} \neq 0} w_{\tilde{e}} = \sum_{e \in p_1 \cap p_2} -\ln\left(1 - \frac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right)$. Thus, since we have shown that $\sum_{f_{\tilde{e}} \neq 0} w_{\tilde{e}}$ has the minimal value with respect to all γ-integral link flows that transfer $2 \cdot \gamma$ flow units from $\tilde{s}$ to $\tilde{t}$ in $\widetilde{G}(\widetilde{V},\widetilde{E})$, it follows by construction that the decomposed path flow that constitutes the pair of paths p_1, p_2 has the minimum value for $\sum_{e \in p_1 \cap p_2} -\ln\left(1 - \frac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right)$ among all γ-integral path flows that transfer $2 \cdot \gamma$ flow units from s to t without violating the congestion constraint α in network $G(V,E)$. Thus, we conclude that $\sum_{e \in p_1 \cap p_2} -\ln\left(1 - \frac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right) = -\ln \prod_{e \in p_1 \cap p_2}\left(1 - \frac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right)$ is minimal and therefore $\ln \prod_{e \in p_1 \cap p_2}\left(1 - \frac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right)$ is maximal with respect to all the pairs of paths that each transfer γ flow units from s to t without violating the congestion constraint α. Thus, $\prod_{e \in p_1 \cap p_2}\left(1 - \frac{p_e}{1 - \prod_{e' \in E}(1 - p_{e'})}\right)$ is maximal and, by corollary 4.3, the theorem follows. ■

Theorem 4.2 If Algorithm MSC fails for the input $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$, then there is no solution for the instance $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$ of Problem MSC.

Proof Since Algorithm MSC fails for the input $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$, it follows by construction that there is no feasible solution for the Min Cost Flow instance $\langle \widetilde{G}(\widetilde{V},\widetilde{E}),(\tilde{s},\tilde{t}),\{c_{\tilde{e}}\},\{w_{\tilde{e}}\},\widetilde{\gamma}\rangle$ of Step 1. Thus, it is impossible to transfer $2 \cdot \gamma$ flow units from $\tilde{s}$ to $\tilde{t}$ in the network $\widetilde{G}(\widetilde{V},\widetilde{E})$. Hence, by construction, it is impossible to

transfer $2\cdot\gamma$ flow units from *s* to *t* over the network $G(V,E)$ without violating the congestion constraint α. In particular, it is impossible to transfer $2\cdot\gamma$ flow units from *s* to *t* over a pair of paths that carry γ flow units each without violating the congestion constraint α. ■

Theorems 4.1 and 4.2 establish that Algorithm MSC solves Problem MSC. Note that the time complexity of the solution is determined by the min cost flow algorithm of Step 2. Thus, Algorithm MSC can achieve the same time complexity as any standard solution of the Min Cost Flow problem.

4.3 Problem LCSC

In this section we present a polynomial solution to Problem LCSC i.e., the problem that seeks *p*-survivable connections with minimum congestion when a duplicate of the original demand is assigned to each of the connection's paths. To that end, we employ the solution to Problem MSC as follows. Given an instance $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,p\rangle$ of Problem LCSC, we search for the smallest α such that the solution to the instance $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$ of Problem MSC consists of a *p*-survivable connection. Obviously, in order to perform this within polynomial complexity, we must solve a polynomial number of instances of Problem MSC. In the following, we present two important observations that enable us to consider at most $\log(2\cdot M+1)=O(\log N)$ instances of Problem MSC.

Our first observation states that, for any instance of Problem LCSC, the optimal network congestion factor, denoted as α^*, belongs to a given set of at most $2\cdot M+1$ elements. More specifically, we claim that each link $e\in E$ in any solution to Problem LCSC can admit only three possible link congestion factor values: the first is zero, which corresponds to cases where the connection is not using the link at all; the second is $\frac{\gamma}{c_e}$, which corresponds to cases where exactly one of the connection's path employs the link e; the third value equals to $\frac{2\cdot\gamma}{c_e}$ and corresponds to cases where both of the connection's paths employ the link e. Thus, the optimal network congestion factor α^* must be a member of the following set $\left\{\frac{i\cdot\gamma}{c_e}\middle| e\in E, i=0,1,2\right\}$. Note that this set consists of at most $2\cdot M+1$ members.

Our second observation enables to speed up the search process by employing a binary search. To that end, note that, for each $\alpha\geq\alpha^*$, the solution to the instance $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$ of Problem MSC produces a connection that is survivable upon a link failure with a probability of at least *p*; on the other hand, for each $\alpha<\alpha^*$, the solution to the instance $\langle G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha\rangle$ of Problem MSC, produces a connection that is survivable upon a link failure with a probability that is less than *p*. Thus, the above strategy to search for a *p*-survivable connection that minimizes the

network congestion factor α, can employ a binary search over $\left\{\frac{i\cdot\gamma}{c_e}\middle| e\in E, i=0,1,2\right\}$ and therefore consider at most $\log(2\cdot M+1)=O(\log N)$ instances of Problem MSC. In Fig. 4.3 we provide the formal description of the algorithm.

Algorithm LCSC$(G,\{s,t\},\{c_e\},\{p_e\},\gamma,p)$

Parameters:

G – network

$\{s,t\}$ – commodity

$\{c_e\}$ – capacities

$\{p_e\}$ – failure probabilities

γ – demand

p – survivability constraint

Variables:

α – network congestion factor

(p_1,p_2) – survivable connection

1. Perform a binary search over the set $\overline{\alpha}=\left\{\frac{i\cdot\gamma}{c_e}\middle| e\in E, i=0,1,2\right\}$ in order to find the smallest $\alpha\in\overline{\alpha}$ that satisfies $\prod_{e\in p_1\cap p_2}\left(1-\frac{p_e}{1-\prod_{e'\in E}(1-p_{e'})}\right)\geq p$ where (p_1,p_2) are obtained by $(p_1,p_2)\leftarrow$ Algorithm_MSC$(G,(s,t),\{c_e\},\{p_e\},\gamma,\alpha)$.

2. **If** the search failed

Return Fail.

Else

Return (p_1,p_2).

Fig. 4.3: Algorithm LCSC

Finally, we consider the complexity of Algorithm LCSC. The algorithm conducts a binary search over the set $\overline{\alpha}$, hence it considers $\log(2 \cdot M + 1) = O(\log N)$ values. For each considered value, it executes Algorithm MSC that solves a single instance of the min-cost flow problem. Therefore, the complexity of Algorithm LCSC is $O(T(N,M) \cdot \log N)$ where $T(N,M)$ is the running time of any standard min-cost flow algorithm for an *N*-node *M*-link network.

4.4 QoS Extensions

For an instance of either Problem MSC or Problem LCSC, there may be several optimal solutions. Among them, we may be interested in those that optimize some QoS target, such as end-to-end delay, jitter, cost, etc. Accordingly, we add to the definitions of Problems MSC and LCSC a secondary objective function which, among the corresponding optimal solutions, seeks the one that minimizes the sum of the weights of the paths that constitute the survivable connection. Starting with Problem MSC, its modified version is as follows.

Problem Weighted-MSC Given are an instance $\langle G,(s,t),\{c_e\},\{p_e\},\alpha \rangle$ of Problem MSC and a weight w_e for each $e \in E$. Find a survivable connection (p_1, p_2) that has the minimum weight $W(p_1, p_2) \triangleq W(p_1) + W(p_2)$ such that (p_1, p_2) is an optimal solution to the instance $\langle G,(s,t),\{c_e\},\{p_e\},\alpha \rangle$ of Problem MSC.

Informally, the goal of Problem Weighted-MSC is to route the duplicates of the original demand along the connection that has the maximum survivability to failures while keeping the congestion of the network below some specified level; subject to this goal, the problem seeks a connection that has the minimum weight $W(p_1, p_2)$.

The definition of Problem Weighted-LCSC is similar and therefore omitted.

In the following, we outline the solution methodology for Problem Weighted-MSC, which is a modification of Algorithm MSC. As before, we transform the instance of Problem Weighted-MSC into an instance of the min-cost flow problem using a very similar reduction to the one presented in Fig. 4.1 for Problem MSC. The only difference lies in the weight that each link is assigned in the reduced instance of the min-cost flow problem. More specifically, if *K* denotes a large number, then the weight that each link is assigned in the reduced instance of the min-cost flow problem is incremented in $\frac{w_e}{K}$ with respect to the original reduction. The modified reduction is illustrated in Fig. 4.4.

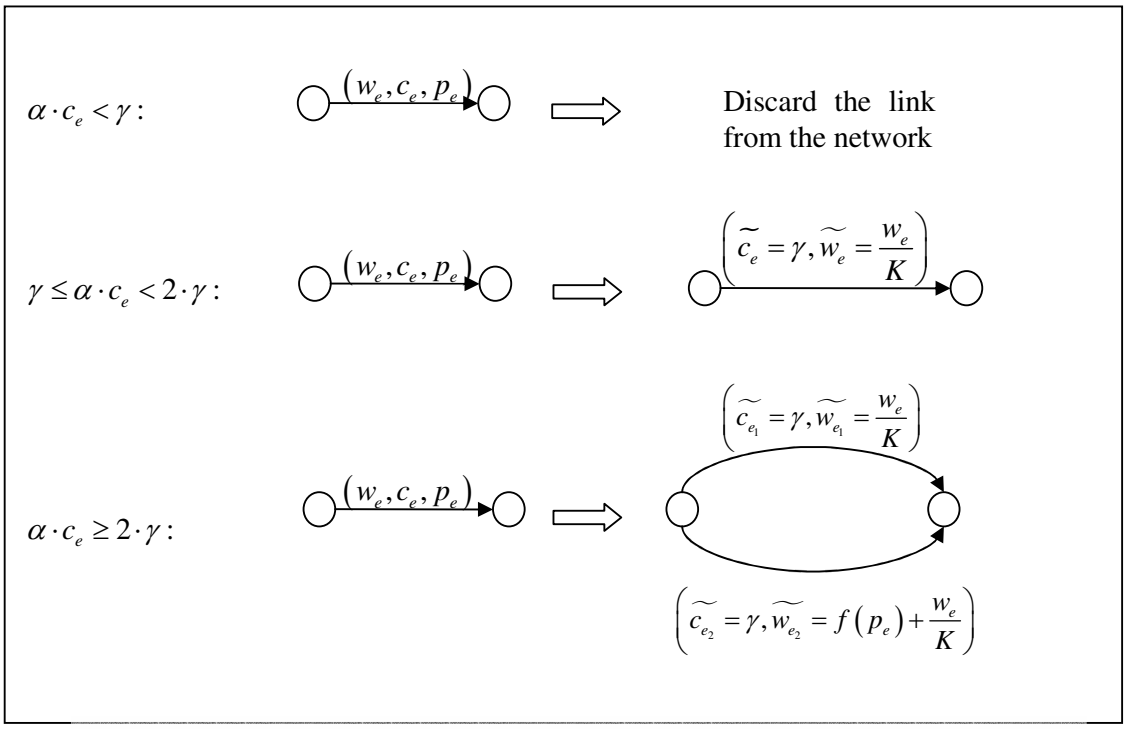

Fig. 4.4: Reducing Problem Weighted -MSC to the Min Cost Flow Problem

The value of *K* must be large enough such that a min-cost flow with respect to the new costs will also be optimal with respect to the original costs. Note that, for $K \to \infty$, the new costs converge into the old costs and a min-cost flow with respect to the new instance is also a min-cost flow with respect to the original instance. Therefore, by choosing arbitrarily large *K* we can arbitrarily get closer to the optimal solution of Problem MSC.

However, in practical settings, the cost of each $e \in E$ is represented as rational number $\frac{m_e}{n_e}$ where m_e and n_e are integers. Therefore, by multiplying all costs by $\prod_{e \in E} n_e$, we obtain integral costs. Hence, for the resulting integral costs if it holds that $\sum_{e \in E} \frac{w_e}{K} < 1$ then it is easy to see that a min-cost flow with respect to the original costs is a min-cost flow with respect to the new costs. Thus, by choosing $K > \sum_{e \in E} w_e \cdot \prod_{e \in E} n_e$, we obtain an optimal solution for Problem MSC i.e., a solution that optimizes the survivability of the connection as the primary objective.

Obviously, in order to maintain polynomial complexity *K* must have a polynomial representation with respect to the input. Indeed, we show now that $\sum_{e \in E} w_e \cdot \prod_{e \in E} n_e$ is polynomial. To that end, define $n_{\max} \triangleq \max_{e \in E} \{n_e\}$ and $w_{\max} \triangleq \max_{e \in E} \{w_e\}$. The number of bits that are required in order to decode $\sum_{e \in E} w_e \cdot \prod_{e \in E} n_e$ is at most

$\log\left(\sum_{e\in E} w_e \cdot \prod_{e\in E} n_e\right) = \log\left(\sum_{e\in E} w_e\right) + \log\left(\prod_{e\in E} n_e\right) \le \log\left(M \cdot w_{\max}\right) + M \cdot \log n_{\max}$. Clearly, this number is polynomial in the input.

We present a sketch of the proof, which shows that this modification provides the optimal solution to Problem Weighted-MSC. As was explained, for each $K > \sum_{e\in E} w_e \cdot \prod_{e\in E} n_e$, a min-cost flow with respect to the new costs is a min-cost flow with respect to the original costs. Thus, for such a *K*, a min cost flow with respect to the new costs is a min cost flow with respect to the original costs such that the increment in the total cost $\sum_{e\in E} f_e \cdot \frac{w_e}{K} = \sum_{e\in p_1\cup p_2} f_e \cdot \frac{w_e}{K}$ is minimal. Therefore, this min-cost flow corresponds to the optimal solution of Problem MSC that has the minimum value for $\sum_{e\in p_1\cup p_2} f_e \cdot w_e$. Finally, since each of the connection's paths transfer γ flow units from *s* to *t*, it follows that $\sum_{e\in p_1\cup p_2} f_e \cdot w_e = \sum_{e\in p_1} \gamma \cdot w_e + \sum_{e\in p_2} \gamma \cdot w_e = \gamma \cdot \left(\sum_{e\in p_1} w_e + \sum_{e\in p_2} w_e\right) = \gamma \cdot W\left(p_1, p_2\right)$. Thus, the min-cost flow with respect to the new costs corresponds to a survivable connection $\left(p_1, p_2\right)$ that has the minimum weight $W\left(p_1, p_2\right)$, such that $\left(p_1, p_2\right)$ is an optimal solution to Problem MSC; hence, the min-cost flow corresponds to the optimal solution of Problem Weighted-MSC.

We conclude this section with the solution to Problem Weighted-LCSC. Given an instance $\left\langle G,(s,t),\{c_e\},\{w_e\},\{p_e\},\gamma,p\right\rangle$ of Problem Weighted-LCSC, we employ a strategy that is very similar to the one presented for Problem LCSC. The strategy searches for the smallest α such that the solution to the instance $\left\langle G,(s,t),\{c_e\},\{w_e\},\{p_e\},\gamma,\alpha\right\rangle$ of Problem Weighted-MSC consists of a *p*-survivable connection. Using the same arguments as in Sec. 4.3, it follows that the optimal network congestion factor α^* of any given instance of Problem Weighted-LCSC must be a member in the set $\left\{\frac{i\cdot\gamma}{c_e} \middle| e\in E, i=0,1,2\right\}$. Since this set consists of a polynomial number of elements, we only need to solve a polynomial number of instances of Problem Weighted-MSC in order to determine the *p*-survivable connection with the smallest α. Finally, as the solution to Problem Weighted-MSC optimizes a secondary objective namely, the weight of the returned connection, it follows that the output of this scheme consists of a *p*-survivable connection that has the minimum network congestion factor α such that the total weight of the returned connection is minimized as a secondary goal. Thus, the output of the latter strategy is the optimal solution to Problem Weighted-LCSC.

4.5 Extensions to Other Protection Architectures

Thus far, we focused on the 1+1 protection architecture. The 1:1 protection architecture is an alternative approach that establishes two paths while sending the data only over one active path, while the backup path is activated only if the active path fails. In this subsection we show that the alternative 1:1 protection architecture can also be addressed using a similar reduction to the one presented in the previous subsections. Then, we show that our relaxation of the standard requirement of disjoint paths gives rise to an efficient third protection architecture, which is a hybrid approach that combines 1:1 protection and 1+1 protection architectures.

4.5.1 The Survivable Connections in 1:1 Protection Architecture

We begin by reformulating Problems MSC and LCSC in order to correspond to a 1:1 protection architecture.

Problem MSC_1:1 Given are a network $G(V,E)$, a pair of nodes *s* and *t*, a congestion constraint $\alpha \geq 0$, a demand γ, and, for each link $e \in E$, a capacity $c_e \geq 0$ and a failure probability $p_e \geq 0$. Find a survivable connection $(p_1, p_2) \in P^{(s,t)} \times P^{(s,t)}$ with the maximum probability to remain operational upon a link failure, such that transferring the demand γ along path p_1 or along path p_2 produces a network congestion factor of at most α.

Problem LCSC_1:1 Given are a network $G(V,E)$, a pair of nodes *s* and *t*, a survivability constraint $p \geq 0$, a demand γ, and, for each link $e \in E$, a capacity $c_e \geq 0$ and a failure probability $p_e \geq 0$. Find a *p*-survivable connection $(p_1, p_2) \in P^{(s,t)} \times P^{(s,t)}$ such that transferring the demand γ along path p_1 or along path p_2 produces the minimal network congestion factor.

In the following, we outline the solution to Problem MSC_1:1, which is a modification of Algorithm MSC. As before, we transform the instance of Problem MSC_1:1 into an instance of the min-cost flow problem using a similar reduction to the one presented in Fig. 4.1 for Problem MSC. Since the solution to Problem MSC_1:1 refers only to the cases where at most one path is active, it holds that at any given time at most one path is assigned with γ flow units. Therefore, only two cases should be considered, namely $\alpha \cdot c_e < \gamma$ and $\alpha \cdot c_e \geq \gamma$. Clearly, as before, all the links that satisfy $\alpha \cdot c_e < \gamma$ should be discarded from the network since they cannot be used in order to transfer γ flow units without exceeding the congestion constraint α. On the other hand, in contrast with the solution to Problem MSC all other links can concurrently be employed by the pair of paths that constitute the solution to Problem MSC_1:1. More specifically, the only difference between the solution to Problem MSC (that corresponds to a 1+1 protection architecture) and the solution to Problem MSC_1:1 is the type of links that can be used by both paths; whereas the solution to Problem MSC cannot employ a link $e \in E$ that satisfies $\gamma \leq \alpha \cdot c_e < 2 \cdot \gamma$ for both paths (since each path alone transfers γ flow units), in the solution to Problem MSC_1:1

such a link can be shared by the pair of paths that constitute the survivable connection. The reduction for the 1:1 protection architecture is illustrated in Fig. 4.5.

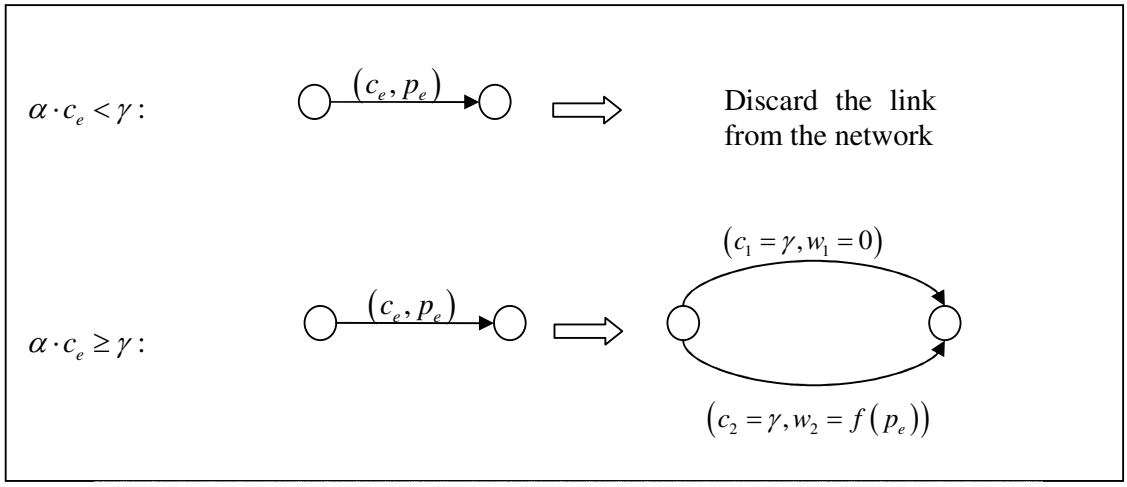

Fig. 4.5: Reducing Problem MSC_1:1 to the Min Cost Flow Problem

We omit the specification of the algorithms that solve Problems MSC_1:1 and LCSC_1:1.

4.5.2 An Hybrid (1:1 and 1+1) Protection Architecture

Thus far, we have focused on 1+1 and 1:1 protection architectures. However, the relaxation of the path disjointedness requirement considered in this work, enables to define a third, hybrid, protection architecture. More specifically, we present a new architecture that, for a connection (p_1, p_2), transfers γ flow units over the links in $p_1 \cap p_2$, as in 1:1 protection, while over the links in $p_1 \cup p_2 \setminus p_1 \cap p_2$ it transfers γ flow units, as in 1+1 protection. This new protection architecture is illustrated through the following example.

Example: Consider the network depicted in Fig. 4.6. Suppose that we are given a survivable connection (p_1, p_2) such that $p_1 = (e_1, e_3, e_4)$ and $p_2 = (e_2, e_3, e_5)$.

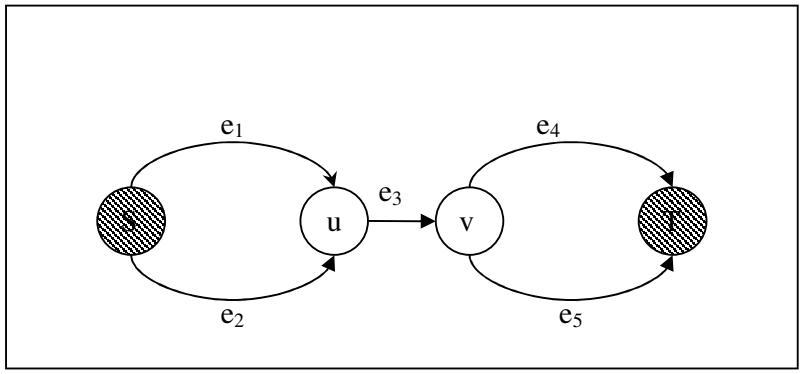

Fig. 4.6: Hybrid Protection

Hybrid Protection transfers one duplicate of the original flow demand through link $e_1 \in p_1$ and another duplicate through link $e_2 \in p_2$. While both duplicates arrive to node *u*, only the first to arrive is assigned to link $u \to v$ and the other one is discarded. When the duplicate that was assigned to $u \to v$ arrives to *v*, Hybrid Protection transfers one duplicate through link $e_4 \in p_1$ and another through link $e_5 \in p_2$. Finally, as before, node *t* considers only the duplicate that is the first to arrive. Note that such an assignment of traffic to links is *not* a link flow.

Hybrid protection architecture has several important advantages. First, it reduces the congestion of all links that are shared by both paths with respect to 1+1 protection architecture. At the same time, upon a link failure, it has a faster restoration time with respect to 1:1 protection. Finally, it provides the fastest propagation of data with respect to the propagation time of all paths that can be constructed from the links in $p_1 \cup p_2$. We demonstrate this property on the above example. Assume that the link propagation delays satisfy $d_{e_1} < d_{e_2}$ and $d_{e_5} < d_{e_4}$. Then, by construction, node *u* assigns over link e_3 the incoming flow of link e_1, and node *t* considers only the duplicate of link e_5. Thus, data is propagated through the path $p = (e_1, e_3, e_5)$, which has the minimum propagation delay among all the paths that employ links out of $p_1 \cup p_2$.

Problem MSC_Hybrid Given are a network $G(V,E)$, a pair of nodes *s* and *t*, a congestion constraint $\alpha \geq 0$, a demand γ, and, for each link $e \in E$, a capacity $c_e \geq 0$ and a failure probability $p_e \geq 0$. Find a survivable connection $(p_1, p_2) \in P^{(s,t)} \times P^{(s,t)}$ with the maximum probability to remain operational upon a link failure, such that $\max_{e \in p_1 \cup p_2} \left\{ \frac{\gamma}{c_e} \right\} \leq \alpha$ i.e., assigning the demand γ to each link $e \in p_1 \cup p_2$ produces a network congestion factor of at most α.

The solution to Problem MSC_Hybrid can be obtained through the solution to Problem MSC_1:1. More specifically, an optimal solution for Problem MSC_Hybrid is an optimal solution for Problem MSC_1:1. The latter follows from the following observation. For a given survivable connection (p_1, p_2), employing the 1:1 protection architecture that transfers γ flow units through either p_1 or p_2, produces a network congestion factor of at most α *iff* $\max_{e \in p_1} \left\{ \frac{\gamma}{c_e} \right\} \leq \alpha$ **and** $\max_{e \in p_2} \left\{ \frac{\gamma}{c_e} \right\} \leq \alpha$. Therefore, it holds that transferring γ flow units through either p_1 or p_2 produces a network congestion factor of at most α *iff* $\max_{e \in p_1 \cup p_2} \left\{ \frac{\gamma}{c_e} \right\} \leq \alpha$. Thus, by the definition of Problems MSC_Hybrid and MSC_1:1, it holds that a survivable connection (p_1, p_2) is an optimal solution to Problem MSC_1:1 *iff* it is an optimal solution to Problem MSC_Hybrid. However, it is important to note that, while 1:1 protection assigns traffic only to the links that belong to either p_1 or p_2, the hybrid protection assigns traffic to *all* the links in $p_1 \cup p_2$.

4.6 Do We Need Survivable Connections that Consist of More Than Two Paths?

In this subsection we show that, under the single failure model, there is no protection architecture that needs more than two paths in order to achieve either the desired degree of protection or the desired level of congestion. More precisely, we cannot improve the degree of protection or the level of congestion, by admitting more than two paths for each survivable connection. Thus, there is no reason to define survivable connections that consist of more than two paths. Moreover, since all the protection schemes that we present in this work produce the optimal solution for survivable connections that consist of at most two paths, our solutions are optimal for any number of paths. The above is established through the following theorem. The theorem focuses on 1:1 protection (hence, also on hybrid protection), and we later extend it to 1+1 protection.

Theorem 4.3 Given are a network $G(V,E)$, a pair of nodes $\{s,t\}$ and, for each $e \in E$, a failure probability $p_e \geq 0$. Let $(p_1, p_2, \cdots, p_k) \in P^{(s,t)} \times P^{(s,t)} \times \cdots \times P^{(s,t)}$ be a *p*-survivable connection that produces a network congestion factor of at most α when a demand of γ flow units is transferred through one of the connection's paths. If at any given time at most one link is not operational (i.e., single link failure), then there exists a *p*-survivable connection $(\overline{p_1}, \overline{p_2}) \in P^{(s,t)} \times P^{(s,t)}$ such that, transferring the demand γ along $\overline{p_1}$ or $\overline{p_2}$ produces a network congestion factor of at most α.

Proof Let $\widehat{E} \triangleq \left\{ e \middle| e \in \bigcup_{i=1}^{k} p_i \right\}$ i.e., the collection of all links that are employed by the paths of the given survivable connection $(p_1, p_2, \cdots, p_k)$. We shall construct a survivable connection $(\overline{p_1}, \overline{p_2}) \in P^{(s,t)} \times P^{(s,t)}$ such that $\overline{p_1} \cup \overline{p_2} \subseteq \widehat{E}$. Since transferring γ flow units through any single path that belongs to the survivable connection $(p_1, p_2, \cdots, p_k) \in P^{(s,t)} \times P^{(s,t)} \times \cdots \times P^{(s,t)}$ produces a network congestion factor of at most α, it holds that transferring γ flow units through each $e \in \widehat{E}$ produces a network congestion factor of at most α. Thus, since we shall construct the survivable connection $(\overline{p_1}, \overline{p_2})$ only from links in $\widehat{E}$, it holds that transferring the demand γ along path $\overline{p_1}$ or along path $\overline{p_2}$ produces a network congestion factor of at most α.

We now construct a pair of paths $\overline{p_1}, \overline{p_2} \in P^{(s,t)}$ from links in $\widehat{E}$ such that the probability that at least one path remains operational upon a link failure is not less than the probability that some path in $(p_1, p_2, \cdots, p_k)$ is operational upon that failure. To that end, we first determine the probability of $(p_1, p_2, \cdots, p_k)$ to remain operational upon a failure.

In subsection 4.1, we showed that for each $e \in E$, the failure probability under the condition of a failure $\widetilde{p_e}$ satisfies $\widetilde{p_e} = 1 - \frac{p_e}{1 - \prod_{e' \in E} (1 - p_{e'})}$. Given a survivable connection $(p_1, p_2, \cdots, p_k)$, denote by $\overline{E}$ the set of all links that are common to the paths $p_1, p_2, \cdots, p_k$ i.e., $\overline{E} \triangleq \left\{ e \middle| e \in \bigcap_{i=1}^{k} p_i \right\}$. Since we assume the single link failure model, it follows that *only* a link $e \in \overline{E}$ i.e., a that is common to all the paths of the given survivable connection $(p_1, p_2, \cdots, p_k)$, can break the connection upon a link failure. Thus, the probability that at least one path in $(p_1, p_2, \cdots, p_k)$ remains operational under the condition of a failure, equals to the probability that all of the common links are operational under that condition. Thus, since we assume independent failure probabilities, it holds that the probability that at least one of the connection's path remains operational under an event of a failure equals to $\prod_{e \in \overline{E}} \left(1 - \widetilde{p_e}\right)$. Thus, the connection $(p_1, p_2, \cdots, p_k)$ is a $\prod_{e \in \overline{E}} \left(1 - \widetilde{p_e}\right)$-survivable connection.

Since $(p_1, p_2, \cdots, p_k)$ is a $\prod_{e \in \overline{E}} \left(1 - \widetilde{p_e}\right)$-survivable connection it follows that in order to establish the theorem we only need to show that there exists a pair of paths $\overline{p_1}, \overline{p_2} \in P^{(s,t)}$, $\overline{p_1} \cup \overline{p_2} \subseteq \widehat{E}$ such that the probability that at least one of them remains operational under an event of a failure is at least $\prod_{e \in \overline{E}} \left(1 - \widetilde{p_e}\right)$. According to corollary 4.2 the probability that either $\overline{p_1}$ or $\overline{p_2}$ remains operational upon a link failure is equal to $\prod_{e \in \overline{p_1 \cap p_2}} \left(1 - \widetilde{p_e}\right)$. Therefore, we have to show that $\prod_{e \in \overline{p_1 \cap p_2}} \left(1 - \widetilde{p_e}\right) \geq \prod_{e \in \overline{E}} \left(1 - \widetilde{p_e}\right)$. Thus, it is enough to show the existence of a pair of paths $\overline{p_1}, \overline{p_2} \in P^{(s,t)}$ that satisfies $\overline{p_1} \cup \overline{p_2} \subseteq \widehat{E}$ and $\overline{p_1} \cap \overline{p_2} \subseteq \overline{E}$.

Remove from the network all the links that are not used by the paths of $(p_1, p_2, \cdots, p_k)$ i.e., all the links that are not in $\widehat{E}$. We have to show that there exists a pair of paths $\overline{p_1}, \overline{p_2} \in P^{(s,t)}$ over $G\left(V, \widehat{E}\right)$ such that $\overline{p_1} \cap \overline{p_2} \subseteq \overline{E}$. To that end, we employ the following construction that transforms $G\left(V, \widehat{E}\right)$ into a flow network. Assign to each $e \in \overline{E}$, two units of capacity, and assign to each $e \in \widehat{E} - \overline{E}$ one unit of capacity. We prove now that there exists a pair of paths $\overline{p_1}, \overline{p_2} \in P^{(s,t)}$ over $G\left(V, \widehat{E}\right)$ such that $\overline{p_1} \cap \overline{p_2} \subseteq \overline{E}$ *iff* it is possible to define an integral link flow that transfers two flow units from *s* to *t* over $G\left(V, \widehat{E}\right)$.

$\Rightarrow$: Assume that there exists a pair of paths $\overline{p_1}, \overline{p_2} \in P^{(s,t)}$ over $G\left(V, \widehat{E}\right)$ such that $\overline{p_1} \cap \overline{p_2} \subseteq \overline{E}$. Assign one unit of flow to each path. Obviously, if the capacity constraints are satisfied, the corresponding link flow is an integral link flow that transfers two flow units from *s* to *t* over $G\left(V, \widehat{E}\right)$. It is left to be shown that such an assignment satisfies the capacity constraints. To that end, observe that assigning one unit of flow to each path produces two units of flow over the links in $\overline{p_1} \cap \overline{p_2}$ and one unit of flow over the links in $\overline{p_1} \cup \overline{p_2} \,/\, \overline{p_1} \cap \overline{p_2}$. Since $\overline{p_1} \cap \overline{p_2} \subseteq \overline{E}$, it follows by construction that all the links in $\overline{p_1} \cap \overline{p_2}$ are assigned with two units of capacity; hence, the capacity constraints are satisfied for these links. Similarly, since $\overline{p_1} \cup \overline{p_2} \,/\, \overline{p_1} \cap \overline{p_2} \subseteq \widehat{E}$, it follows that all the links in $\overline{p_1} \cup \overline{p_2} \,/\, \overline{p_1} \cap \overline{p_2}$ are assigned with at least one unit of capacity; hence the capacity constraints are also satisfied for the links in $\overline{p_1} \cup \overline{p_2} \,/\, \overline{p_1} \cap \overline{p_2}$.

$\Leftarrow$: Assume that it is possible to define an integral link flow that transfers two flow units from *s* to *t* over $G\left(V, \widehat{E}\right)$. Hence, by the flow decomposition theorem [1], it is possible to define a pair of paths such that each path transfers one flow unit from *s* to *t* over $G\left(V, \widehat{E}\right)$. Moreover, the corresponding paths can intersect only on the links that have two units of capacity; hence, by construction, these paths intersect only on links that belong to $\overline{E}$. Thus, there exists a pair of paths $\overline{p_1}, \overline{p_2} \in P^{(s,t)}$ over $G\left(V, \widehat{E}\right)$ such that $\overline{p_1} \cap \overline{p_2} \subseteq \overline{E}$.

Hence, in order to prove the theorem, it remains to be shown that it is possible to define an integral link flow that transfers two flow units from *s* to *t* over $G\left(V, \widehat{E}\right)$. However, since all the links have an integral capacity, the maximum flow that can be transferred from *s* to *t* under the integrality restriction is equal to the maximum flow that can be transferred from *s* to *t* when the integrality restriction is omitted [1]; hence, it is sufficient to show that it is possible to transfer two flow units from *s* to *t* over $G\left(V, \widehat{E}\right)$.

Suppose by way of contradiction, that it is impossible to transfer two flow units from *s* to *t* over $G\left(V, \widehat{E}\right)$. Thus, according to the max-flow min-cut theorem [7], there exists a cut (S,T) with $s \in S$ and $t \in T$ such that $C(S,T) \triangleq \sum_{x \in S,\, y \in T} c_{x \to y} < 2$. Therefore, since the capacity of all links is integral, it follows that $C(S,T) \leq 1$. Thus, since each link has *at least* one unit of capacity, it follows that at most one link $x \to y \in \widehat{E}$, such that $x \in S$ and $y \in T$, crosses (S,T). However, since each path in $\left(p_1, p_2, \cdots, p_k\right)$ is a path from *s* to *t*, it follows that there exists at least one link that connects some node in *S* to some node in *T*. Thus, it follows that *exactly* one link

$x \to y \in \hat{E}$, $x \in S$ and $y \in T$, crosses the cut (S,T). Denote this link by e. Since $C(S,T) \le 1$, it follows that $c_e \le 1$. Obviously, each path from *s* to *t* must traverse through the link *e*. In particular, all the paths of $(p_1, p_2, \cdots, p_k)$ must traverse the link *e*. Hence, by definition, it follows that $e \in \overline{E}$. Since $c_e \le 1$, it follows that there is a link in $G(V, \hat{E})$ that belongs to $\overline{E}$ whose capacity is at most 1. However, this contradicts the fact that $c_e = 2$ for each $e \in \overline{E}$. Thus, it is possible to transfer two flow units from *s* to *t* over $G(V, \hat{E})$. ■

Theorem 4.3 allows that, for 1:1 and hybrid protections, there is no benefit in maintaining survivable connections with more than two paths. We now show that the same can be concluded for 1+1 protection. To that end, we observe that, for 1+1 protection, it holds that a connection $(p_1, p_2, \cdots, p_k)$ transfers $k \cdot \gamma$ flow units over each link $e \in \bigcap_{i=1}^{k} p_i$; moreover, the flow over the rest of the links in $\bigcup_{i=1}^{k} p_i$ equals to at least γ. In the proof of Theorem 4.3, we showed that there exists a pair of paths $(\overline{p_1}, \overline{p_2})$ that intersect only on links that belong to $\bigcap_{i=1}^{k} p_i$ and employ only links that belong to $\bigcup_{i=1}^{k} p_i$. Note that, under the 1+1 protection, the connection $(\overline{p_1}, \overline{p_2})$ transfers $2 \cdot \gamma$ flow units over the links in $\bigcap_{i=1}^{k} p_i$ and γ flow units over the rest of the links in $\bigcup_{i=1}^{k} p_i$. Thus, the traffic that each link $e \in E$ carries under the connection $(\overline{p_1}, \overline{p_2})$ is never larger than the traffic that is carried by link e under the connection $(p_1, p_2, \cdots, p_k)$. Thus, employing 1+1 protection over $(\overline{p_1}, \overline{p_2})$ produces a network congestion factor that is not larger than over $(p_1, p_2, \cdots, p_k)$. Moreover, since $p_1 \cap p_2 \subseteq \bigcap_{i=1}^{k} p_i$, it follows that the degree of survivability of $(\overline{p_1}, \overline{p_2})$ is at least the degree of survivability of $(p_1, p_2, \cdots, p_k)$.Hence, Theorem 4.3 is valid also for 1+1 protection.

Finally, we show that, with more than a single link failure, the property that was just established does not hold. To that end, consider the following example.

Example: Consider the network depicted in Fig. 4.7. Suppose that each of the links has a failure probability of 0.1 under the condition of a failure. Moreover, assume that we seek a survivable connection that, upon a link failure, has a probability of at least 0.999 to survive. In such a case, if we employ only two links $i, j \in \{1,2,3\}$, $i \ne j$, we have a probability of $1 - p_i \cdot p_j = 0.99$ to survive the failure. The requirement for a survivability of at least 0.999 is achieved only if we employ all three links.

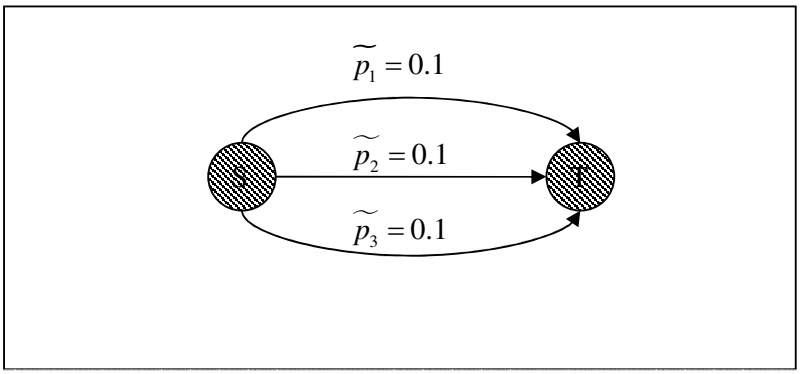

Fig. 4.7: Survivability outside the Single Link Failure Model

5. Solution of problem ReMP

In this section we aim at solving problem ReMP, i.e., the problem of minimizing congestion under end-to-end reliability requirements. First, we establish that the problem is intractable.

5.1 Intractability of Problem ReMP.

Theorem 5.1 Problem ReMP is NP-hard.

Proof First, let us define the single-commodity case of problem ReMP as a decision problem.

Given are a network $G(V,E)$, for each link $e \in E$, a failure probability $p_e > 0$ and a capacity $c_e > 0$, and, for some commodity $(s,t) \in V \times V$, a demand $\gamma > 0$ and a (minimal) success probability Π. Is there a path flow with network congestion factor of at most α such that, for path p that transfers a positive amount of flow, it holds that $\prod_{e \in p}(1-p_e) \geq \Pi$?

In [3] we considered the following problem, termed as Problem RMP (Restricted Multipath): given are a network $G(V,E)$, for each link $e \in E$, a weight $w_e \geq 0$ and a capacity $c_e > 0$, and, for some commodity $(s,t) \in V \times V$, a demand $\gamma > 0$ and a weight restriction W. Is there a path flow $f : P \to \mathbb{R}^+ \cup \{0\}$ with a network congestion factor of at most α such that, for path p that transfers a positive amount of flow, it holds that $W(p) \leq W$?

In [3], we have shown that Problem RMP is NP hard. Given an instance $\langle G(V,E), \{w_e\}, \{c_e\}, \gamma, W \rangle$ of Problem RMP, we transform it into an instance $\langle G(V,E), \{p_e\}, \{c_e\}, \gamma, \Pi \rangle$ of Problem ReMP as follows:

- Define a failure probability $p_e \triangleq 1-2^{-w_e}$ for each link $e \in E$ with a weight $w_e \geq 0$.
- Define a success probability restriction $\Pi \triangleq 2^{-W}$.

We now prove that it is possible to transfer γ flow units over paths whose success probabilities are not smaller than Π without exceeding the network congestion factor α *iff* it is possible to transfer the same flow demand over paths whose weights are not larger than W while satisfying the same restriction α on the network congestion factor. To that end, we prove that every path flow that is a solution to the given instance of Problem RMP is also a solution to the transformed instance of Problem ReMP and vice versa. Thus, we have to show that each feasible path with respect to the instance $\langle G(V,E),\{w_e\},\{c_e\},\gamma,W\rangle$ of Problem RMP is also feasible with respect to the transformed instance $\langle G(V,E),\{p_e\},\{c_e\},\gamma,\Pi\rangle$ of Problem ReMP and vice versa. i.e., for each path $p \in P^{(s,t)}$ it holds that $\Pi(p) \geq \Pi \Leftrightarrow W(p) \leq W$. Indeed:

$$\Pi(p) \geq \Pi \Leftrightarrow \prod_{e\in p}(1-p_e) \geq \Pi \Leftrightarrow \prod_{e\in p}\left(1-\left(1-2^{-w_e}\right)\right) \geq 2^{-W} \Leftrightarrow \prod_{e\in p} 2^{-w_e} \geq 2^{-W} \Leftrightarrow$$

$$\Leftrightarrow 2^{\sum_{e\in p}-w_e} \geq 2^{-W} \Leftrightarrow \sum_{e\in p} -w_e \geq -W \Leftrightarrow \sum_{e\in p} w_e \leq W \Leftrightarrow W(p) \leq W,$$ thus Problem ReMP is NP hard. ■

5.2 Approximation Scheme for Problem ReMP

In this section we establish an ε-optimal approximation scheme for Problem ReMP. We are given a network $G(V,E)$, for each link $e \in E$ a failure probability $p_e > 0$ and a capacity $c_e > 0$, and, for each commodity $(i,j) \in V \times V$ a demand $\gamma^{(i,j)}$ and a success probability restriction $\Pi^{(i,j)}$. We construct a *quantized* instance of Problem ReMP where each link has discrete failure probability and each commodity has a discrete restriction on the success probability. To that end, we first define a set of classes $\left\{1,\left(1+\frac{\varepsilon}{N}\right)^{-1},\left(1+\frac{\varepsilon}{N}\right)^{-2},\cdots\right\}$ where $\varepsilon > 0$ is a given approximation parameter. Then, we quantize the link success probability $1-p_e$, $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)} \leq 1-p_e \leq \left(1+\frac{\varepsilon}{N}\right)^{-i}$ into a discrete success probability $1-\widehat{p_e}$ where $1-\widehat{p_e} = \left(1+\frac{\varepsilon}{N}\right)^{-i}$. In addition, we quantize the restriction on the success probability Π, $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)} \leq \Pi \leq \left(1+\frac{\varepsilon}{N}\right)^{-i}$ into a discrete restriction $\widehat{\Pi}$ where $\widehat{\Pi} = \left(1+\frac{\varepsilon}{N}\right)^{-(i+1)}$. Note that the new (quantized) instance relaxes the original instance and therefore the optimal solution to the quantized instance produces a network

congestion factor that is not larger than the optimal network congestion factor of the original instance. We will later establish that the success probability of each routing path is violated by at most a factor of $1+\varepsilon$. Unless stated otherwise, in the following we refer to the quantized instance.

In order to solve the quantized version of Problem ReMP, we employ linear programming. To that end, we need some additional terminology.

Given is a quantized instance of Problem ReMP, and let α be the network congestion factor of its solution. Define $f_e^{\pi,(i,j)}$ as the flow of commodity (i,j) over link $e=(u,v)\in E$ that has traversed through paths $p\in P^{(i,u)}$ with a success probability $\Pi(p)=\pi$. Finally, for each $v\in V$, denote by $O(v)$ the set of links that emanate from v, and by $I(v)$ the set of links that enter that node, namely $O(v)=\{(v,l)|(v,l)\in E\}$ and $I(v)=\{(w,v)|(w,v)\in E\}$. Then, the quantized version of Problem ReMP can be formulated as the following linear program:

Program Quantized ReMP $\left(G(V,E),\ \left\{\widehat{p_e}\right\},\ \{c_e\},\ \left\{\gamma^{(i,j)}\right\},\ \left\{\widehat{\Pi^{(i,j)}}\right\} \right)$

Minimize α

s.t.

$$\sum_{e\in O(v)} f_e^{\pi\cdot\left(1-\widehat{p_e}\right),(i,j)} - \sum_{e\in I(v)} f_e^{\pi,(i,j)} = 0 \quad ,\forall (i,j)\in\beta \quad ,\forall v\in V-\{i,j\} \quad ,\forall \pi\in\left[\widehat{\Pi^{(i,j)}},1\right] \tag{1}$$

$$\sum_{e\in O(i)} f_e^{\pi\cdot\left(1-\widehat{p_e}\right),(i,j)} - \sum_{e\in I(i)} f_e^{\pi,(i,j)} = 0 \quad ,\forall (i,j)\in\beta \quad ,\forall \pi\in\left[\widehat{\Pi^{(i,j)}},1\right) \tag{2}$$

$$\sum_{e\in O(i)} f_e^{1,(i,j)} = \gamma^{(i,j)} \qquad ,\forall (i,j)\in\beta \tag{3}$$

$$\sum_{(i,j)\in\beta} \sum_{\pi=\widehat{\Pi^{(i,j)}}}^{1} f_e^{\pi,(i,j)} \le c_e\cdot\alpha \qquad ,\forall e\in E \tag{4}$$

$$f_e^{\pi,(i,j)} = 0 \qquad ,\forall \pi\notin\left[\widehat{\Pi^{(i,j)}},1\right] \ ,\forall (i,j)\in\beta \ ,\forall e\in E \tag{5}$$

$$f_e^{\pi,(i,j)} \ge 0 \qquad ,\forall (i,j)\in\beta \ ,\forall \pi\in\left[\widehat{\Pi^{(i,j)}},1\right] \ ,\forall e\in E \tag{6}$$

$$\alpha \ge 0 \tag{7}$$

Fig 5.1 Program Quantized ReMP

Note that the variables of the linear program are the link flows $\left\{f_e^{\pi,(i,j)}\right\}$ and the network congestion factor α.

In the linear program, the objective function is to minimize the network congestion factor. Constraints (1), (2) and (3) are nodal flow conservation constraints. Equation (1) states that the traffic flowing into node v, through paths $p\in P^{(i,u)}$ and links $e=(u,v)\in E$ such that $\Pi(p)=\pi,$ has to be equal to the traffic flowing out of node v through paths $p'\in P^{(i,v)}$ that have a success probability $\Pi(p')=\pi\cdot\left(1-\widehat{p_e}\right)$; since $\pi\in\left[\widehat{\Pi^{(i,j)}},1\right]$, the restriction on the probability to success is preserved for each commodity $(i,j)\in\beta$; finally, equation (1) must be satisfied for each node other than the source node and the destination node for each commodity with a positive demand. Equation (2) extends the validity of equation (1) to hold for traffic that encounters the source node i after it has already traversed over paths with a non-zero probability of failure. Informally, equation (2) states that "old" traffic that was already traversing over at least one link must satisfy equation (1) (that focuses only on nodes in $V-\{i,j\}$) when it emanates from source i not for the first time. Equation (3) states that, for each commodity $(i,j)\in\beta$, the traffic flowing out of source i that has not traversed through any path yet (thus $\pi=1$), must be equal to the demand $\gamma^{(i,j)}$. Equation (4) is the link capacity constraint. Expression (5) rules out non-feasible flows, and Expressions (6) and (7) restrict all variables to be non-negative.

We can solve Program ReMP using any polynomial time algorithm for linear programming [14]. The solution to problem ReMP is then achieved by decomposing the link flow $\left\{f_e^{\pi,(i,j)}\right\}$ into a path flow that satisfies the success probability restrictions. This is done by Algorithm PFC, specified in Fig. 5.2, which is an (elaborated) generalization of the Flow Decomposition Algorithm [1].

For each commodity $(i,j)\in V\times V,$ Algorithm PFC uses the link flow $\left\{f_e^{\pi,(i,j)}\right\}$ in order to define paths with a success probability of at least $\widehat{\Pi^{(i,j)}}$ that transfer a total flow of $\gamma^{(i,j)}$ flow units. To that end, at each iteration the algorithm selects a commodity $(i,j)\in V\times V$ with a demand $\gamma^{(i,j)}>0$. Then, it employs Procedure Path Construction, specified in Fig. 5.3, that returns a collection of pairs $S\triangleq\left\{\left(e_k,\pi_k\right)\right\},$ such that the corresponding set $\left\{f_{e_k}^{\pi_k,(i,j)}\right\}$ is a set of *positive variables* that identifies a path $p\triangleq\left(e_1,e_2,\cdots,e_{|S|}\right)$ from source *i* to destination *j* with a success probability of at least $\widehat{\Pi^{(i,j)}}$. The flow over the path *p* is defined to be equal to the smallest variable $f_{e_k}^{\pi_k,(i,j)}$ that belongs to *p* i.e., $\min_{(e_k,\pi_k)\in S}\left\{f_{e_k}^{\pi_k,(s,t)}\right\}$. Then, the algorithm subtracts the flow that traverses through *p* from the demand $\gamma^{(i,j)}$ and from each variable in the path i.e., from each variable $f_{e_k}^{\pi_k,(i,j)}$ such that $\left(e_k,\pi_k\right)\in S$. The algorithm stops when the demand $\gamma^{(i,j)}$ is zeroed for each $(i,j)\in V\times V$. Thus, for each $(i,j)\in V\times V$, the resulting path flow transfers $\gamma^{(i,j)}$ flow units from source *i* to destination *j* over paths with a success probability of at least $\widehat{\Pi^{(i,j)}}$.

Algorithm PFC $\left(G(V,E),\left\{f_e^{\pi,(i,j)}\right\},\left\{\gamma^{(i,j)}\right\}\right)$

Initialization:

For each commodity $(i,j)\in V\times V$ and each path $p\in P^{(s,t)}$: $f(p)\leftarrow 0$.

For each commodity $(s,t)\in V\times V$:

While $\gamma^{(s,t)}>0$ do:

1. $S\leftarrow$ Path_Construction$\left(G(V,E),\{s,t\},\left\{f_e^{\pi,(i,j)}\right\}\right)$.
2. $f_{e_k}^{\pi_k,(s,t)}\leftarrow f_{e_k}^{\pi_k,(s,t)}-\min_{(e_k,\pi_k)\in S}\left\{f_{e_k}^{\pi_k,(s,t)}\right\}$, for each $(e_k,\pi_k)\in S$.
3. $\gamma^{(s,t)}\leftarrow\gamma^{(s,t)}-\min_{(e_k,\pi_k)\in S}\left\{f_{e_k}^{\pi_k,(s,t)}\right\}$.
4. Denote $p\triangleq s\xrightarrow{e_1}v_1\xrightarrow{e_2}v_2\cdots v_{|s|}\xrightarrow{e_{|s|}}t$, where e_k corresponds to the pair $(e_k,\pi_k)\in S$. $f(p)\leftarrow\min_{(e_k,\pi_k)\in S}\left\{f_{e_k}^{\pi_k,(s,t)}\right\}$.

Return the path flow *f*.

Fig 5.2: Algorithm Path Flow Construction (PFC)

We turn to explain the main idea behind Procedure Path Construction, specified in Fig. 5.3. Given a commodity $(s,t)\in V\times V$, the procedure identifies a collection of pairs $S\triangleq\left\{(e_k,\pi_k)\right\}_{k=1}^{h}$ whose corresponding variables $\left\{f_{e_1}^{\pi_1,(s,t)},f_{e_2}^{\pi_2,(s,t)},\cdots,f_{e_k}^{\pi_k,(s,t)}\right\}$ are all positive such that $u_0\xrightarrow{e_1}u_1\xrightarrow{e_2}u_2\cdots u_{h-1}\xrightarrow{e_h}u_h$ is a path from *s* to *t*, and for each $1\le k\le h$ it holds that $\pi_k=\left(1-\widehat{p_{e_1}}\right)\cdot\left(1-\widehat{p_{e_2}}\right)\cdots\left(1-\widehat{p_{e_{k-1}}}\right)$. This is done by employing the following property that characterizes the solution to Program Quantized ReMP. If a *positive* flow $f_e^{\pi,(s,t)}$ enters through link $e=(u,v)$ into a node $v\in V-\{s,t\}$, then there exists a *positive* flow $f_{e'}^{\pi\left(1-\widehat{p_e}\right),(s,t)}$ that emanates out of *v* through a link $e'=(v,w)$. Since each positive flow $f_e^{\pi,(s,t)}$ must satisfy $\pi\ge\widehat{\Pi^{(s,t)}}$ it follows that, if we follow these positive flows from the source *s*, we establish a path that satisfies the success probability restriction $\widehat{\Pi^{(s,t)}}$. Unless we follow these positive flows on some cycle *C* such that $\left(1-\widehat{p_e}\right)=1$ for each $e\in C$, we establish in Lemma 5.1 that we must discover a positive flow that enters into the destination i.e., we must identify a directed path from *s* to *t*. Otherwise, if we follow the positive flows on a 1-probabilty cycle *C* (i.e., a cycle that consists of only links with a success probability of $\left(1-\widehat{p_e}\right)=1$), then we eliminate the cycle as follows. First, we determine the flow over the cycle to be equal to the smallest variable $f_{e_k}^{\pi_k,(i,j)}$ that belongs to *C*. Then, we subtract the flow that traverses through *C* from each variable in the cycle and repeat the above process of establishing a path of positive variables from source *s* to destination *t*. Note that since each time we identify a 1- probability cycle we zero at

least one variable, the number of times that such 1- probability cycles can be detected is at most the number of variables in $\left\{f_e^{\pi,(i,j)}\right\}$. Therefore, after at most $\left|\left\{f_e^{\pi,(i,j)}\right\}\right|$ times that we detect and eliminate flow cycles, we must obtain a directed path from *s* to *t* by following positive flows from the source *s*. Finally, note that since "regular" cycles are not detected, the constructed path is not necessarily simple.

Procedure Path Construction $\left(G(V,E),\{s,t\},\left\{f_e^{\pi,(i,j)}\right\}\right)$

Initialization

$S \leftarrow \phi$, $u_0 \leftarrow s$, $\pi_0 \leftarrow 0$, $k \leftarrow 0$

While $u_k \neq t$ do

1. Select a positive variable $f_{e_k}^{\pi_k,(s,t)}$ such that $e_k \triangleq (u_k,u_{k+1}) \in E$.
2. If $(e_k,\pi_k) \in S$ (i.e., a 1-probability cycle was detected) then
   a. Let $\overline{C} \triangleq \{(e_i,\pi_i),(e_{i+1},\pi_{i+1}),\cdots,(e_{k-1},\pi_{k-1})\}$ be a subset of *S* such that $(e_i,\pi_i)=(e_k,\pi_k)$.
   b. $f_{e_k}^{\pi_k,(s,t)} \leftarrow f_{e_k}^{\pi_k,(s,t)} - \min_{(e_k,\pi_k)\in C}\left\{f_{e_k}^{\pi_k,(s,t)}\right\}$, for each $(e_k,\pi_k) \in C$. * the cycle flow was eliminated *\
   c. $S \leftarrow \text{Path\_Construction}\left(G(V,E),\{s,t\},\left\{f_e^{\pi,(i,j)}\right\}\right)$
   d. Return *S*
3. $S \leftarrow S \cup (e_k,\pi_k)$, $\pi_{k+1} \leftarrow \pi_k \cdot \left(1-\widehat{p_{e_k}}\right)$, $k \leftarrow k+1$.

Return *S*

Fig 5.3: Procedure Path Construction

Lemma 5.1: Consider the returned set *S* (Fig. 5.3) and define $e_k \triangleq u_k \rightarrow u_{k+1}$ for each $(e_k,\pi_k) \in S$. If $\gamma^{(s,t)}>0$, then there exists an *h*, $h \le -(M+1)\cdot \log_{1+\frac{\varepsilon}{N}} \widehat{\Pi^{(s,t)}}$ such that the sequence $(u_0,u_1,\ldots,u_h)$ is a path from *s* to *t* with a succsess probability of at least $\widehat{\Pi^{(s,t)}}$.

Proof: Since $\gamma^{(s,t)}>0$, it follows from constraint (3) (of Program Quantized ReMP) that there exists at least one link $e_0=(u_0,u_1)$ such that the variable $f_{e_0}^{1,(s,t)}$ is positive; by construction the algorithm selects one such variable and assigns its corresponding pair $(e_0,\pi_0), \pi_0 = 1$ to the set *S*. Then, from constraints (1) and (2), it follows that unless $u_1=t$, there exists at least one link $e_1=(u_1,u_2)$ such that the variable $f_{e_1}^{1\cdot(1-\widehat{p_{e_0}}),(s,t)}$ is positive; like before the pair (e_1,π_1) that corresponds to $f_{e_1}^{1\cdot(1-\widehat{p_{e_0}}),(s,t)}$ is assigned to *S*. Thus, applying constraints (1) and (2) for any index *k*, it follows that, if the set *S* contains a pair (e_k,π_k) that corresponds to some positive variable $f_{e_k}^{\pi_k,(s,t)}$ where

$\pi_k \triangleq \prod_{l\in[0,k-1]} \left(1-\widehat{p_{e_l}}\right)$ and $e_k=(u_k,u_{k+1})$, then, unless u_{k+1}=t, it holds that S must contain some pair $\left(e_{k+1},\pi_{k+1}\right)$ where $\pi_{k+1}=\pi_k\cdot\left(1-\widehat{p_{e_k}}\right)$ and e_{k+1}=(u_{k+1},u_{k+2}) such that $f_{e_{k+1}}^{\pi_{k+1},(s,t)}$ is positive. Therefore, if S consists of a pair $\left(e_k,\pi_k\right)$ where $\pi_k \triangleq \prod_{l\in[0,k-1]} \left(1-\widehat{p_{e_l}}\right)$ and $e_k=(u_k,u_{k+1})$, and lacks a pair $\left(e_{k+1},\pi_{k+1}\right)$ where $\pi_{k+1}=\pi_k\cdot\left(1-\widehat{p_{e_k}}\right)$ and e_{k+1}=(u_{k+1},u_{k+2}), it follows that u_{k+1}=t; hence, the sequence $(u_0,u_1,\ldots,u_{k+1})$ that corresponds to $\left\langle f_{e_0}^{\pi_0,(s,t)}, f_{e_1}^{\pi_1,(s,t)},\cdots, f_{e_k}^{\pi_k,(s,t)}\right\rangle$ is a path from s to t. Thus, in order to prove the Lemma, it is sufficient to show that $k+1\le -(M+1)\cdot\log_{1+\frac{\varepsilon}{N}}\widehat{\Pi^{(s,t)}}$. More specifically, we have to show that there is no pair $\left(e_k,\pi_k\right)\in S$ such that $k>-(M+1)\cdot\log_{1+\frac{\varepsilon}{N}}\widehat{\Pi^{(s,t)}}-1$.

Suppose, by the way of contradiction, that there exists a pair $\left(e_k,\pi_k\right)\in S$ such that $k\ge -(M+1)\cdot\log_{1+\frac{\varepsilon}{N}}\widehat{\Pi^{(s,t)}}$. Since it follows by construction that the path p=$(u_0,u_1,\ldots,u_{h+1})$ that corresponds to the pairs $\left(e_0,\pi_0\right),\left(e_1,\pi_1\right),\cdots,\left(e_h,\pi_h\right)\in S$ has no 1-probability cycle, it follows that $\pi_k>\pi_m$ for each $\left(e_k,\pi_k\right),\left(e_m,\pi_m\right)\in S$ that satisfy $e_k=e_m$ and $k<m$. Thus, since any subpath $P'\subseteq P$, $\left|p'\right|>M$ has at least two links $e_1,e_2\in p'$ such that $e_1=e_2$, it follows that $\pi_k>\pi_m$ for each $\left(e_k,\pi_k\right),\left(e_m,\pi_m\right)\in S$ that satisfy $m-k>M$; hence, since $\pi_k,\pi_m\in\left\{\left(1+\frac{\varepsilon}{N}\right)^i \middle| i\in\mathbb{N}\right\}$, it follows that $\pi_k\ge\left(1+\frac{\varepsilon}{N}\right)\cdot\pi_m$. Therefore, since by construction $\pi_0=1$, it follows that $\pi_{M+1}<\left(1+\frac{\varepsilon}{N}\right)^{-1}$. In general, it is easy to see that $\pi_{i\cdot(M+1)}<\left(1+\frac{\varepsilon}{N}\right)^{-i}$. Thus, since we assume that there exists a pair $\left(e_k,\pi_k\right)\in S$ such that $k\ge -(M+1)\cdot\log_{\left(1+\frac{\varepsilon}{N}\right)}\widehat{\Pi^{(s,t)}}$, it follows that there exists a pair $\left(e_k,\pi_k\right)\in S$ such that $\pi_k<\left(1+\frac{\varepsilon}{N}\right)^{\log_{\left(1+\frac{\varepsilon}{N}\right)}\widehat{\Pi^{(s,t)}}}=\widehat{\Pi^{(s,t)}}$. Thus, since the pair $\left(e_k,\pi_k\right)\in S$ identifies a positive variable $f_{e_k}^{\pi_k,(s,t)}$, it holds that there exists a positive variable $f_e^{\pi,(s,t)}$ such that $\pi<\widehat{\Pi^{(s,t)}}$. However, this contradicts constraint (5) of Program Quantized ReMP that zero each variable $f_e^{\pi,(s,t)}$ with a success probability $\pi<\widehat{\Pi^{(s,t)}}$. Thus, we have established that there exists an h, $h\le -(M+1)\cdot\log_{1+\frac{\varepsilon}{N}}\widehat{\Pi^{(s,t)}}$ such that the sequence $(u_0,u_1,\ldots,u_h)$ is a path from s to t.

It is only left to prove that path $(u_0,u_1,\ldots,u_h)$ has a succsess probability of at least $\widehat{\Pi^{(s,t)}}$. However, as S corresponds only to positive variables $f_e^{\pi,(s,t)}$, it follows from (5) and (6) that $\pi \in \left[\widehat{\Pi^{(s,t)}},1\right]$ i.e., it follows that the path $(u_0,u_1,\ldots,u_h)$ has a success probability of at least $\widehat{\Pi^{(s,t)}}$. ■

We are now ready to specify the approximation scheme for Problem ReMP. Given an instance of Problem ReMP and an approximation parameter ε, the scheme solves the corresponding quantized instance using Program Quantized ReMP. The output of the program (i.e., the link flow $\left\{f_e^{\pi,(i,j)}\right\}$) is decomposed by Algorithm PFC into a path flow that satisfies the end-to-end reliability requirements. Finally, we convert each non-simple path in the output of Algorithm PFC into a simple path by eliminating loops; this is essential, since the total error in the evaluation of the success probability of each path depends on the hop count. In theorem 5.2, we establish that the resulting path flow violates the reliability requirement of each path by a factor of at most $(1+\varepsilon)$ and has a network congestion factor that is not larger than the optimal network congestion factor. Fig. 5.4 specifies this approximation scheme for Problem ReMP.

ReMP Approximation Scheme $\left\langle G(V,E),\ \{p_e\},\ \{c_e\},\ \left\{\gamma^{(\mathrm{i,j})}\right\},\ \left\{\Pi^{(\mathrm{i,j})}\right\},\varepsilon\right\rangle$

Parameters:

$G(V,E)-$ network

$\{p_e\}-$ failure probabilities

$\{c_e\}-$ capacities

$\left\{\gamma^{(\mathrm{i,j})}\right\}-$ demands

$\left\{\Pi^{(\mathrm{i,j})}\right\}-$probability restrictions

$\varepsilon-$approximation parameter

1. Construct a quantized instance $\left\langle G\left(V,\widehat{E}\right),\ \left\{\widehat{p_e}\right\},\ \left\{\widehat{c_e}\right\},\ \left\{\gamma^{(\mathrm{i,j})}\right\},\ \left\{\widehat{\Pi^{(\mathrm{i,j})}}\right\}\right\rangle$ of Problem ReMP as follows:
   - Let $\widehat{E}\subseteq E$ be the collection of all links in E such that for each $e\in E$ it holds that $(1-p_e)\geq \min_{(i,j)\in V\times V}\left\{\Pi^{(i,j)}\right\}$.
   - For each link $e\in E$ with $i\in\mathbb{Z}$ such that $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)}\leq 1-p_e\leq\left(1+\frac{\varepsilon}{N}\right)^{-i}$, define a corresponding link $e'\in\widehat{E}$ such that $1-\widehat{p_{e'}}=\left(1+\frac{\varepsilon}{N}\right)^{-i}$ and $\widehat{c_{e'}}=c_e$.
   - For each commodity $(i,j)\in V\times V$ with $i\in\mathbb{Z}$ such that $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)}\leq\Pi^{(i,j)}\leq\left(1+\frac{\varepsilon}{N}\right)^{-i}$, define a corresponding restricion $\widehat{\Pi^{(i,j)}}$ such that $\widehat{\Pi^{(i,j)}}=\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)}$.
2. $\left\{f_e^{\pi,(i,j)}\right\}\leftarrow$**Quantized_ReMP** $\left(G\left(V,\widehat{E}\right),\ \left\{\widehat{p_e}\right\},\ \left\{\widehat{c_e}\right\},\ \left\{\gamma^{(\mathrm{i,j})}\right\},\ \left\{\widehat{\Pi^{(\mathrm{i,j})}}\right\}\right)$.
3. $f\leftarrow$**PFC** $\left(G(V,\widehat{E}),\left\{f_e^{\pi,(i,j)}\right\},\left\{\gamma^{(i,j)}\right\}\right)$
4. Construct a path flow f as follows:
   a. Let $h:P^{(i,j)}\rightarrow P^{(i,j)}_{\text{simple}}$ map each path $(v_0,v_1,\cdots v_{h-1},v_h,\cdots v_h,v_{h+1},\cdots v_n)\in P^{(i,j)}$ into a simple path $(v_0,v_1,\cdots v_{h-1},v_h,v_{h+1},\cdots v_n)\in P^{(i,j)}_{\text{simple}}$.
   b. For each $p\in P^{(i,j)}_{\text{simple}}$, $f(p)\triangleq\sum_{p'|h(p')=p}g(p')$.
5. Return path flow f.

Fig 5.4 ReMP Approximation Scheme

We now establish the performance of ReMP Approximation Scheme. First, we show that the complexity of the scheme is polynomial with respect to the input and ε.

Theorem 5.2 Given an instance $\left\langle G,\{c_e\},\{p_e\},\left\{\gamma^{(\mathrm{i,j})}\right\},\left\{\Pi^{(\mathrm{i,j})}\right\}\right\rangle$ of problem ReMP and an approximation parameter ε, the ReMP Approximation Scheme has a polynomial complexity with respect to the input and the approximation parameter ε.

Proof Clearly, the complexity of the ReMP Approximation Scheme is determined by steps (2) and (3). Thus, in order to prove the theorem, we have to prove that both steps have a polynomial complexity with respect to the input and the approximation parameter ε.

In step (2), (the linear) Program Quantized ReMP, which was defined in Fig. 5.1, solves the instance $\left\langle G\left(V,\widehat{E}\right),\left\{\widehat{p_e}\right\},\left\{\widehat{c_e}\right\},\left\{\gamma^{(\mathrm{i,j})}\right\},\left\{\widehat{\Pi^{(\mathrm{i,j})}}\right\}\right\rangle$. The complexity incurred by solving that program is polynomial in the number of variables $\left\{f_e^{\pi,(i,j)}\right\}$ [14]. Thus, it is sufficient to show that the number of variables of Program Quantized ReMP is polynomial with respect to the input and ε.

Obviously, the number of variables in $\left\{f_e^{\pi,(i,j)}\right\}$ is the product of the number of links in $\widehat{E}$, the number of commodities $V\times V$ and the number of different values that π can take. Therefore, we only have to show that π can take a polynomial number of different values with respect to the input and the approximation parameter ε. To that end, consider the set of classes $\overline{A}\triangleq\left\{1,\left(1+\frac{\varepsilon}{N}\right)^{-1},\left(1+\frac{\varepsilon}{N}\right)^{-2},\cdots,\left(1+\frac{\varepsilon}{N}\right)^{k},\cdots\right\}$. It follows by construction that all the success probabilities $\left\{\left(1-\widehat{p_e}\right)\right\}$ and all the restrictions $\left\{\widehat{\Pi^{(\mathrm{i,j})}}\right\}$ of the quantized instance are in $\overline{A}$. Therefore, by the construction of Program Quantized ReMP, it holds that, for each variable $f_e^{\pi,(i,j)}$, there exists an $i\in\mathbb{Z}$ such that $\pi=\left(1+\frac{\varepsilon}{N}\right)^{-i}$. On the other hand, since Program Quantized ReMP refers only to variables $\left\{f_e^{\pi,(i,j)}\right\}$ such that $\pi\in\left[\widehat{\Pi^{(i,j)}},1\right]$ it holds that, for a commodity $(i,j)\in V\times V$ and a link $e\in E$, the number of different variables $\left\{f_e^{\pi,(i,j)}\right\}$ is at most the number of different values in the interval $\left[\widehat{\Pi^{(i,j)}},1\right]$ that can be expressed as $\left(1+\frac{\varepsilon}{N}\right)^{-i}$, where $i\in\mathbb{Z}$. In other words, if $\Pi^{\min}\triangleq\min_{(i,j)\in V\times V}\left\{\widehat{\Pi^{(i,j)}}\right\}$ then, for each variable $f_e^{\pi,(i,j)}$, it holds that $\pi\in\left\{1,\left(1+\frac{\varepsilon}{N}\right)^{-1},\left(1+\frac{\varepsilon}{N}\right)^{-2},\cdots,\left(1+\frac{\varepsilon}{N}\right)^{k}=\Pi^{\min}\right\}$. Thus, it is easy to see that $k=\log_{1+\frac{\varepsilon}{N}}\Pi^{\min}$ and therefore π can take at most $\left|\log_{1+\frac{\varepsilon}{N}}\Pi^{\min}+1\right|=O\left(\frac{N}{\varepsilon}\cdot\log\,\Pi^{\min}\right)$ different values. Therefore, we established that π can take a polynomial number of different values with respect to the input and the approximation parameter ε.

We turn to consider Step (3). There, we employ Algorithm PFC (Fig. 5.2) in order to decompose the link flow $\left\{f_e^{\pi,(i,j)}\right\}$ into a corresponding path flow. Note that each iteration of Algorithm PFC zeroes at least one variable $f_e^{\pi,(i,j)}$. Therefore, Algorithm PFC iterates for no more than the number of variables $\left\{f_e^{\pi,(i,j)}\right\}$. However, as was just shown, this number is polynomial with respect to the input and the approximation parameter ε. Thus, we only have to prove that each iteration consists of a polynomial number of operations.

It is easy to see that the complexity of each iteration is determined by Procedure Path Construction, specified in Fig. 5.3. The procedure seeks a collection of pairs $\left\{\left(e_k,\pi_k\right)\right\}_{k=1}^{h}$ such that the corresponding set $\left\{f_{e_k}^{\pi_k,(s,t)}\right\}_{k=1}^{h}$ is a set of positive variables that identify a path (u_0,u_1,…,u_h) from *s* to *t*. As the procedure is recursive, we first show that the number of recursive calls is polynomial. By construction, we perform a new recursive call whenever we pick a *positive* variable that was already assigned to *S*. Thus, since the recursive call is executed only after we zero at least one positive variable in $\left\{f_e^{\pi,(i,j)}\right\}$, the number of recursive calls is at most $\left|\left\{f_e^{\pi,(i,j)}\right\}\right|$, which was shown to be polynomial with respect to the input and ε; hence, the number of calls is polynomial as well. Next, we show that each recursive call has a polynomial complexity. To that end, observe that we terminate each recursive call when a positive variable that was already selected and assigned to *S*, is selected again. Thus, each variable is identified at most twice; hence, since identifying each positive variable $f_e^{\pi,(s,t)}$ consumes $O(1)$ operations, it follows that each recursive call consumes $O\left(\left|\left\{f_e^{\pi,(i,j)}\right\}\right|\right)$ operations. Therefore, the complexity of each recursive call is polynomial with respect to the input and the approximation parameter ε.

Thus, steps (2) and (3) have a polynomial complexity with respect to the input and ε and the Theorem is established. ■

Theorem 5.3 Given an instance $\left\langle G(V,E),\{c_e\},\{p_e\},\left\{\gamma^{(i,j)}\right\},\left\{\Pi^{(i,j)}\right\}\right\rangle$ of Problem ReMP and an approximation parameter ε, the *ReMP Approximation Scheme*, specified in Fig. 5.4, outputs a path flow *f* that satisfies the following:

a. For each commodity $(i,j)\in V\times V$, $\sum_{p\in P^{(i,j)}} f(p)=\gamma^{(i,j)}$ i.e., the flow demand requirements are satisfied for all commodities.
b. If α^* is the network congestion factor of the optimal solution, then, for each e$\in E$, it holds that $\sum_{(i,j)\in V\times V}\sum_{p\in P^{(i,j)}}\Delta_e(p)\cdot f(p)\le\alpha^*\cdot c_e$, i.e., the network congestion factor is at most α^*.

c. For each commodity $(i,j) \in V \times V$, if $p \in P^{(i,j)}$ and $f(p) > 0$, then $\Pi(p) \geq \dfrac{\Pi^{(i,j)}}{(1+\varepsilon)}$ i.e., the restriction on the probability of success is violated by a factor of at most $(1+\varepsilon)$.

Proof

a. Follows from constraint (3) of Program Quantized ReMP.

b. Since the algorithm rounds up the success probability of each link and rounds down the restrictions on the minimum success probability, it follows that the resulting quantized instance $\left\langle G\left(V,\hat{E}\right), \left\{\widehat{p_e}\right\}, \left\{\widehat{c_e}\right\}, \left\{\gamma^{(i,j)}\right\}, \left\{\widehat{\Pi^{(i,j)}}\right\}\right\rangle$ of problem ReMP *relaxes* the original constraints. Hence, the resulting network congestion factor is at most α^*.

c. We now prove that, if a path $p \in P^{(i,j)}$ is assigned with a positive amount of flow by the ReMP Approximation Scheme, then $\prod_{e \in p}(1-p_e) \geq \dfrac{\Pi^{(i,j)}}{(1+\varepsilon)}$. Consider the original instance $\left\langle G(V,E), \{p_e\}, \{c_e\}, \left\{\gamma^{(i,j)}\right\}, \left\{\Pi^{(i,j)}\right\}\right\rangle$ of Problem ReMP and the corresponding quantized instance $\left\langle G\left(V,\hat{E}\right), \left\{\widehat{p_e}\right\}, \left\{\widehat{c_e}\right\}, \left\{\gamma^{(i,j)}\right\}, \left\{\widehat{\Pi^{(i,j)}}\right\}\right\rangle$.

Suppose that a path $p \in P^{(i,j)}$ transfers a positive amount of flow in the output of the ReMP Approximation Scheme. Therefore, by construction, it follows that p is a simple path that satisfies the quantized restriction on $\widehat{\Pi^{(i,j)}}$ i.e., (1) $\prod_{e \in p}\left(1-\widehat{p_e}\right) \geq \widehat{\Pi^{(i,j)}}$. By the construction of the quantized instance (Step 1), it holds that $\left(1-\widehat{p_e}\right) \leq \left(1+\dfrac{\varepsilon}{N}\right)\cdot(1-p_e)$. Thus, it holds that $\prod_{e \in p}\left(1-\widehat{p_e}\right) \leq \prod_{e \in p}\left(1+\dfrac{\varepsilon}{N}\right)\cdot(1-p_e) = \left(1+\dfrac{\varepsilon}{N}\right)^{|p|}\cdot\prod_{e \in p}(1-p_e)$. Finally, since p is simple, it holds that $|p| \leq N-1$ and, therefore, $\prod_{e \in p}\left(1-\widehat{p_e}\right) \leq \left(1+\dfrac{\varepsilon}{N}\right)^{|p|}\cdot\prod_{e \in p}(1-p_e) \leq \left(1+\dfrac{\varepsilon}{N}\right)^{N-1}\cdot\prod_{e \in p}(1-p_e)$. Therefore, applying (1), we conclude that $\widehat{\Pi^{(i,j)}} \leq \prod_{e \in p}\left(1-\widehat{p_e}\right) \leq \left(1+\dfrac{\varepsilon}{N}\right)^{N-1}\cdot\prod_{e \in p}(1-p_e)$; thus, (2) $\widehat{\Pi^{(i,j)}}\cdot\left(1+\dfrac{\varepsilon}{N}\right) \leq \left(1+\dfrac{\varepsilon}{N}\right)^{N}\cdot\prod_{e \in p}(1-p_e)$. Finally, since by the

construction of the quantized instance it follows that $\Pi^{(i,j)} \le \widehat{\Pi^{(i,j)}} \cdot \left(1+\frac{\varepsilon}{N}\right)$ for each $(i,j) \in V \times V$, we employ inequality (2) in order to obtain that $\Pi^{(i,j)} \le \widehat{\Pi^{(i,j)}} \cdot \left(1+\frac{\varepsilon}{N}\right) \le \left(1+\frac{\varepsilon}{N}\right)^N \cdot \prod_{e \in p}(1-p_e)$. Thus, it holds that $\Pi^{(i,j)} \le \left(1+\frac{\varepsilon}{N}\right)^N \cdot \prod_{e \in p}(1-p_e) \approx \left(1+N \cdot \frac{\varepsilon}{N}\right) \cdot \prod_{e \in p}(1-p_e) = (1+\varepsilon) \cdot \prod_{e \in p}(1-p_e)$, and $\prod_{e \in p}(1-p_e) \ge \frac{\Pi^{(i,j)}}{(1+\varepsilon)}$. ■

6. Conclusions and Future Research

Multipath routing is a potentially powerful tool in two major aspects of data networks, namely load balancing and resilience to network failures. However, its practical deployment requires to efficiently handle *both* aspects. Accordingly, we investigated two classes of problems, both aimed to optimize the congestion state of the network. However, whereas the first class considers ways to *avoid* network failures, the second class considers ways to *recover* from failures.

Standard survivability schemes enhance the ability to recover from network failures by establishing pairs of disjoint paths. However, as this strategy is restrictive, it often leads to the selection of poor routing paths (if any). Accordingly, we relax the standard requirement of path disjointedness into a "continuous" survivability requirement. Somewhat surprisingly, yet as the standard requirement of disjoint paths, the new requirement can also be accommodated by efficient polynomial schemes. However, as opposed to the standard requirement, the new requirement allows a flexible choice of the desired degree of survivability, hence enabling to consider important tradeoffs. Note that, although this study has focused on the resilience-congestion tradeoff, the new more flexible, requirement can be employed also when considering other interesting tradeoffs (e.g., resilience-delay, resilience-jitter, etc).

We have employed the continuous survivability concept both for the 1:1 and the 1+1 protection architectures. Moreover, the new concept enabled to define a third hybrid protection architecture, which was also accommodated through an efficient polynomial scheme. Finally, we established that, under the single link failure model, multipath routing schemes that enhance the ability to recover from failures should not employ more than two paths for each connection. Since the single link failure assumption is practically valid in many cases of interest, this finding suggests an important network design rule in terms of survivability.

Another important contribution of this study is the establishment of a multipath routing scheme that avoids network failures by routing over reliable paths exclusively. To that end, we investigated the congestion minimization problem under end-to-end reliability requirements. After showing that this problem is intractable, we established an efficient ε-optimal approximation scheme.

While this study has established efficient solutions to some fundamental multipath routing problems that consider the resilience-congestion tradeoff, there are still several challenges that remain for future work. One major challenge is to restrict the number of paths that each commodity is allowed to employ when the focus is on congestion minimization under end-to-end reliability requirements. The latter restriction is important due to several reasons. First, the complexity of the schemes that distribute the traffic among multiple paths considerably increases with the number of paths. Second, since each path is prone to network failures and each such failure results in a failure of the entire transmission, the vulnerability of the connections may increase when the traffic is splitted among too many paths. Finally, there may be a limit on the number of paths that can be set up between a pair of nodes, as is the case with label-switched paths in MPLS. Since it was observed that congestion is largely reduced when either one or two paths are identified in addition to the traditional single path [6], future research in this context should mainly focus on multipath routing schemes that balance the network's load while usually employing no more than a small number (e.g., three) reliable paths per connection.

Other challenges and ideas for future work that were identified during the evaluation of this study and are behind the scope of this paper are described as follows.

Multipath Routing and Diversity Coding

Consider the two objectives of multipath routing that are the subject of this study. Although focusing on each objective alone severely deteriorates the quality of the other, it is possible to combine between these objectives by employing the idea of *diversity coding* [21]. The latter concept increases fault tolerance by adding redundant information, like error detection and correction codes, into the data stream. Then, the redundant information is routed along paths, which are disjoint to the paths that transfer the original data flow. Employing diversity coding in order to combine between the above objectives can be done by developing new schemes that minimize congestion and satisfy the fundamental property that restricts each path that transfers some positive flow to have an adequate set of disjoint paths with enough bandwidth to protect this flow.

Fast Recovery Schemes for Multipath Routing

Multipath routing can provide additional benefits in networks where resource reservation must be made before data can be sent along a route (e.g. ATM). Consider for example a routing scheme that uses multipath routing in order to reduce congestion. Then, in case of a failure in one of the paths, we may split the data stream that was traveling over the failed path among the remaining paths, *without* any additional path computation or resource reservation. This fast recovery property can be obtained in several ways. For example, it may be employed by imposing a restriction that, upon a path failure, the sum of the *spare* capacities of the remaining paths is not smaller than the flow that was traveling over the failed path.

Survivability in Wireless Networks

Standard protection schemes that employ multipath routing in order to enhance the survivability to failures may be inappropriate for wireless networks. This is due to the

fact that electrical interferences (noises) usually cause highly correlated network failures. Thus, the single link failure model is no longer valid. For example, for survivable connections that establish a pair of paths, a single random noise can cause concurrent failures on the two paths. Since survivability is a major consideration in such networks, protection schemes that address the specific properties of wireless networks are called for.

References

[1] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network Flows: Theory, Algorithm, and Applications", Prentice Hall, 1993.

[2] D. Awduche, J. Malcolm, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS" , Internet Draft <draft-awduche-mpls-traffic-eng-00.txt>, April, 1998.

[3] R. Banner and A. Orda, "Efficient Multipath Routing Schemes for Congestion Minimization", CCIT Report No. 429, Department of Electrical Engineering, Technion, Haifa, Isreal, June 2003. Availble from: ftp://ftp.technion.ac.il/pub/supported/ee/Network/bo03.ps

[4] D. Bertsekas and R. Gallager, “Data networks”, Prentice-Hall, 1992.

[5] R. Bhandari, "Survivable Networks: Algorithms for Diverse", Kluwer Academic Publishers, Boston 1999.

[6] I. Cidon, R. Rom, Y. Shavitt, “Analysis of Multi-Path Routing”, IEEE Transactions on Networking, 7:885–896, 1999.

[7] T. Cormen, C. Leiserson, R. Rivest, "Introduction to Algorithms", MIT press, Cambridge, 1990.

[8] S. Datta, S. Sengupta and S. Biswa, "Efficient Channel Reservation for Backup Paths in Optical Mesh Networks," Proceedings IEEE Globecom'01, San Antonio, Texas , Nov. 2001.

[9] A. Fumagalli and M. Tacca, "Optimal Design of Optical Ring Networks with Differentiated Reliability (DiR)", Proceedings International Workshop on QoS in multiservice IP networks, 2001.

[10] O. Gerstel and G. Sasaki, Quality of Protection (QoP): A Quantitative Unifying Paradigm to Protection Service Grades", Proceedings SPIE OptiComm 2001, vol. 4599, pages 12-23, 2001.

[11] O. Hauser, M. Kodialam and T.V. Lakshman “Capacity Design of Fast Path Restorable Optical Networks“, IEEE/ACM Transactions on Networking, October 2000.

[12] P. Ho, J. Tapolcai, and H. T. Mouftah, "On Achieving Optimal Survivable Routing for Shared Protection in Survivable Next-Generation Internet", IEEE Transaction on Reliability (to appear in March 2004)

[13] S. Iyer, S. Bhattacharyya, N. Taft, N. McKeoen, C. Diot, "A measurement Based Study of Load Balancing in an IP Backbone", Sprint ATL Technical Report, TR02-ATL-051027, May 2002.

[14] N. Karmarkar, "A New Polynomial-Time Algorithm For Linear Programming", Combinatorica, vol. 4, pages 373-395, 1984.

[15] M. Kodialam and T. V. Lakshman, "Restorable Dynamic Quality of Service Routing", IEEE Communication Magazine, vol. 40, no. 6, June 2002.

[16] W. Lai, Ed. and D. McDysan, Ed.," Network Hierarchy and Multilayer Survivability", IETF RFC 3386, November 2002.

[17] G. Maier, A. Pattavina, S. De Patre, and M. Martinelli, "Optical Network Survivability: Protection Techniques in the WDM Layer", Photonic Networks Communication, vol. 4, no. 3-4, pages 251–69, July-Dec. 2002.

[18] G. Mohan and A. K. Somani, "Routing Dependable Connections with Specified Failure Restoration Guarantees in WDM Networks," in Proceedings of Infocom 2000: The Conference on Computer Communications, Institute of Electrical and Electronics Engineers, New York, 2000, pages 1761– 70.

[19] R. Ogier, B. Bellur, and N. Taft-Plotkin, "An Efficient Algorithm for Computing Shortest and Widest Maximally Disjoint Paths", SRI International Technical Report ITAD-1616-TR-170, November 1998.

[20] N. Taft-Plotkin, B. Bellur, and R. Ogier, "Quality-of –Service Routing Using Maximally Disjoint Paths", In Proceedings of IEEE IWQoS99, pages 119-128, London, UK, January 1999.

[21] A. Tsirigos and Z. J. Haas, "Multipath Routing in Mobile Ad Hoc Networks or How to Route in the Presence of Topological Changes", In Proceedings of IEEE MILCOM 2001, pages 878-883, October 2001.

[22] Y. Wang and Z. Wang, "Explicit Routing Algorithms For Internet Traffic Engineering", In Proceedings of ICCN'99, Boston, October 1999.

[23] Dahai Xu and Chunming Qiao, "Distributed Partial Information Management (DPIM) Schemes for Survivable Networks- Part 2", In Proceedings of IEEE INFOCOM 2002.