

Tunable QoS-Aware Network Survivability

Jose Yallouz

Department of Electrical Engineering
Technion, Israel Institute of Technology
Email: jose@tx.technion.ac.il

Ariel Orda

Department of Electrical Engineering
Technion, Israel Institute of Technology
Email: ariel@ee.technion.ac.il

Abstract—Coping with network failures has been recognized as an issue of major importance in terms of social security, stability and prosperity. It has become clear that current networking standards fall short of coping with the complex challenge of surviving failures. The need to address this challenge has become a focal point of networking research. In particular, the concept of *tunable survivability* offers major performance improvements over traditional approaches. Indeed, while the traditional approach is to provide full (100%) protection against network failures through disjoint paths, it was realized that this requirement is too restrictive in practice. Tunable survivability provides a quantitative measure for specifying the desired level (0%-100%) of survivability and offers flexibility in the choice of the routing paths. Previous work focused on the simpler class of “bottleneck” criteria, such as bandwidth. In this study, we focus on the important and much more complex class of *additive* criteria, such as delay and cost. First, we establish some (in part, counter-intuitive) properties of the optimal solution. Then, we establish efficient algorithmic schemes for optimizing the level of survivability under additive end-to-end QoS bounds. Subsequently, through extensive simulations, we show that, at the price of *negligible* reduction in the level of survivability, a major improvement (up to a factor of 2) is obtained in terms of end-to-end QoS performance. Finally, we exploit the above findings in the context of a network design problem, in which we need to best invest a given “budget” for improving the performance of the network links.

I. INTRODUCTION

The internet infrastructure has been progressing rapidly since its deployment. While a decade ago the dial-up modem, which supported rates of 56 kbit/s, was a widely deployed technology, nowadays’ technologies, such as Ethernet and InfiniBand, offer rates of 100 Gbit/s and beyond [1]. Current core routers, such as CRS-3, reach capacities of hundreds of terabits per second. With this extreme increase of transmission rates, any failure in the network infrastructure may lead to a vast amount of data loss. Hence, survivability in the network is becoming increasingly important.

In particular, failures in the network infrastructure should be recovered promptly. For example, some standard recommendations [2] require that recovery from a single failure should be performed within 50ms. The literature distinguishes between two major classes of recovery schemes, namely restoration and protection [3]. In *restoration schemes*, post-failure actions are performed in order to search for a backup path that would avoid the faulty element. In *protection schemes*, on the other hand, pre-failure actions are performed in order to pre-establish a backup solution for any possible failure. Protection schemes have an obvious advantage in terms of recovery time and are

usually achieved by the establishment of pairs of disjoint paths.

We adopt the widely used single link failure model. Although multiple failures can occur in the network, “survivability” under this model aims at handling only single failure events. This model has been the focus of most studies on survivability, e.g. [4] [5] [6], due to its simplicity as well as the fact that protecting against a single failure is a common requirement of various standards, e.g. [2]. Moreover, a common approach for handling multiple failures is to supply protection for the first failure and restoration for any subsequent ones.

Under the single link failure model, the employment of disjoint paths provides full (100%) protection. Hence, this is the common solution approach of path protection schemes. However, the requirement of fully disjoint paths is often too restrictive and demands excessive redundancy in practice. Furthermore, a pair of disjoint paths of sufficient quality may not exist, occasionally making the requirement infeasible. Therefore, a milder and more flexible survivability concept is called for, which would relax the rigid requirement of disjoint paths by also considering paths containing common links. Accordingly, a previous study [7] introduced the novel concept of *tunable survivability*, which provides a quantitative measure to specify the desired level of survivability. This concept allows any degree of survivability in the range 0% to 100%, thus transforming survivability into a quantifiable metric.

Specifically, tunable survivability enables the establishment of connections that can survive network failures with any desired probability. Given a connection that consists of two paths between a source-destination pair under the single failure model, only a failure on a link that is common to both paths can disrupt the connection. Accordingly, we characterize a connection as *p-survivable* if there is a probability of at least *p* to have all common links operational during the connection’s lifetime.

Quality of Service (QoS) refers to a broad collection of networking techniques that provide guarantees to the capability of a network to deliver predictable results. Generally, we distinguish between two classes of QoS metrics, namely: bottleneck metrics, such as bandwidth, which are defined by the weakest component in the path, and additive metrics, such as delay, which are defined by the sum of the corresponding metrics over the path’s links. Algorithmic schemes that combine the concept of tunable survivability with bottleneck metrics were established in [7]. However, the important and much more complex class of additive metrics was not considered.

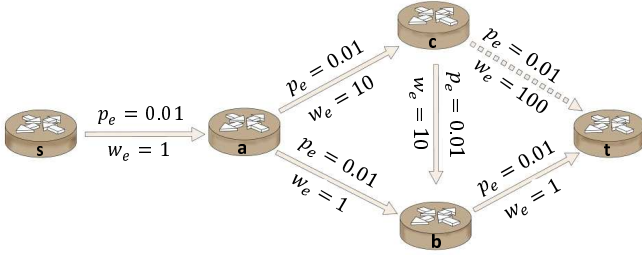


Fig. 1: Example of p -survivable connections under an additive QoS metric

Accordingly, this is the subject of the present work.

The following example demonstrates the concept of p -survivable connections combining an additive QoS metric and its advantages over traditional protection schemes. Consider the network described in Figure 1, where each link is associated with a failure probability p_e and a weight w_e representing an additive metric. Assume that a connection is to be established between s and t . Here, the weight of a p -survivable connection is defined as the sum of all the weights of the connection's links, considering the weight of a link that is common to both paths only once. As no pair of disjoint paths from s to t exists, there is no full protection against single failures, and the traditional survivability requirement is thus infeasible. However, if we are satisfied with 0.99-survivability against single network failures, then a connection that consists of the paths $\pi_1 = \langle s, a, c, t \rangle$ and $\pi_2 = \langle s, a, b, t \rangle$ is a valid solution, since the only (single) failure that can concurrently damage both paths is in the common link (s, a) . Hence, as this link fails with a probability of 0.01, the connection is 0.99-survivable. The weight of the connection, defined by the sum of all link weights w_e (counting the weights of the common links just once), is 113. Now, suppose that we are satisfied with 0.99²-survivability. Clearly, the paths $\pi_1 = \langle s, a, c, b, t \rangle$ and $\pi_2 = \langle s, a, b, t \rangle$ also constitute a valid connection, for which the weight decreases to 23. Finally, assume that we are satisfied with 0.99³-survivability. Now, the single path $\pi = \langle s, a, b, t \rangle$ also becomes feasible, thus decreasing the weight of the connection to 3.

Motivated by [7], we investigate how to combine the tunable survivability concept with additive QoS guarantees. To that end, in Section II, we formulate an optimization problem that considers two requirements, namely a (minimum) level of survivability and an additive end-to-end QoS guarantee. We establish some fundamental properties of the structure of the optimal solution. In particular, in Section III, we prove that, for an important class of problems, only a (typically small) subset of the network's links may affect the survivability value of the optimal solution. Next, in Section IV, we establish that our class of problems is computationally intractable, hence we design and validate a pseudo-polynomial solution and an efficient fully polynomial-time approximation scheme. In Section V, through comprehensive simulations, we show that, typically, a

modest relaxation (of a few percents) in the survivability level is enough to provide a major improvement in terms of the QoS requirement, e.g., cutting by half the end-to-end delay. Then, in Section VI, we exploit the above findings in the context of a network design problem, in which we need to best invest a given "budget" for improving the performance of the network links. Finally, Section VII summarizes our results and discusses directions for future research.

II. MODEL AND PROBLEM FORMULATION

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. We denote the size of these sets as $N = |V|$ and $M = |E|$, correspondingly. A path is a finite sequence of nodes $\pi = \langle v_0, v_1, \dots, v_h \rangle$ such that, for $0 \leq n \leq h-1$, $(v_n, v_{n+1}) \in E$. A path is *simple* if all its nodes are distinct. Given a source node $s \in V$ and a destination node $t \in V$, the set of all simple paths from s to t is denoted by $P^{(s,t)}$. Each link $e \in E$ is associated with a failure probability value $p_e \in (0, p_{max}]$; we note that these probabilities are often estimated out of the available failure statistics of each network component [8]. We assume that each link $e \in E$ fails independently and its failure probability is upper-bounded by some value $p_{max} < 1$. Accordingly, we define the *minimum network success probability* as $S_{min} = (1 - p_{max})^M$. In addition, each link $e \in E$ is assigned with a positive weight w_e that represents an additive QoS target such as delay, cost, jitter, etc.

We adopt the *single link failure model*, which considers handling at most one link failure in the network. A link is classified as either *faulty* or *operational*: it becomes faulty upon a failure and remains to be such until it is repaired, otherwise it is operational. Likewise, we say that a path π is *operational* if it has no faulty link, i.e., for each $e \in \pi$, link e is operational; otherwise, the path is faulty.

We proceed to formulate the concept of tunable survivability, through the following definitions.

Definition 2.1: Given a source node $s \in V$ and a destination node $t \in V$, a *survivable connection* is a pair of paths $(\pi_1, \pi_2) \in P^{(s,t)} \times P^{(s,t)}$.

Survivability is defined as the capability of the network to maintain service continuity in the presence of failures. Thus, we say that a survivable connection (π_1, π_2) is *operational* if either π_1 or π_2 are operational. Under the single link failure model, a survivable connection (π_1, π_2) is *operational iff* the links that are common to both π_1 and π_2 are operational. As mentioned, under the single link failure model, a link that is not common to both paths can never cause a survivable connection to fail; on the other hand, a failure in a common link causes a failure of the entire connection. Accordingly, as the failure probabilities $\{p_e\}$ are independent, we quantify the level of survivability of survivable connections as follows.

Definition 2.2: Given a survivable connection (π_1, π_2) such that $\pi_1 \cap \pi_2 \neq \emptyset$, we say that (π_1, π_2) is a *p -survivable connection* if $\prod_{e \in \pi_1 \cap \pi_2} (1 - p_e) \geq p$, i.e., the probability that all common links are operational is at least p . The value of p is then termed as the *survivability level* of the connection.

The above definition formalizes the notion of tunable survivability for the single link failure model. In case that there are no common links between π_1 and π_2 , i.e., the paths π_1 and π_2 are disjoint, there is no single failure that can make (π_1, π_2) fail; for this case, (π_1, π_2) is defined to be a *1-survivable connection*.

In [7], it was shown that, for any network, if there exists a p -survivable connection that admits more than two paths, then there exists a p -survivable connection that admits exactly two paths. Therefore, we can indeed focus on survivable connections with just two paths.

We proceed to quantify the weight of a survivable connection.

Definition 2.3: Given a network $G(V, E)$ and a (non-empty) path π , its *weight* $W(\pi)$ is defined as the sum of the weight of its links, i.e., $W(\pi) = \sum_{e \in \pi} w_e$. Accordingly, we define a *weight-shortest path* between two nodes $u, v \in V$ as a path in $G(V, E)$ with minimum weight between u and v .

When a connection is composed of two paths, there are several possibilities to define its weight out of the weights of the connection's paths. A choice of particular interest is to consider the minimum of the lengths of the two paths. However, this approach results in strongly NP-complete optimization problems [9], i.e., even approximate solutions are computationally intractable. Alternatively, we can consider the worst (highest) among the weights of the two paths, yet this also leads to an NP-Hard problem [10] (however here tractable approximations are possible, as per below). Therefore, we adopt the classical approach in the context of connections based on disjoint paths, which attempts to minimize the aggregate weight of the two paths (e.g., [11], [12]). Beyond allowing computationally efficient optimal solutions, this approach also provides a 2-approximation solution to the previous approach, which targets at minimizing the higher length of the two paths (as shown in [13]). However, with tunable survivability, the following refinement is called for.

Since a p -survivable survivable connection (π_1, π_2) potentially contains common links, there are two ways to determine its aggregate weight, namely: counting the weight of a common link either *once* or *twice*. We shall consider both options, formalized as follows.

Definition 2.4: Given a survivable connection (π_1, π_2) , its *CO-weight* $W_{CO}(\pi_1, \pi_2)$ is defined as the sum of its link weights counting the common links *once*, i.e., $W_{CO}(\pi_1, \pi_2) = \sum_{e \in \pi_1 \cup \pi_2} w_e$.

Definition 2.5: Given a survivable connection (π_1, π_2) , its *CT-weight* $W_{CT}(\pi_1, \pi_2)$ is defined as the sum of its link weights counting the common links *twice*, i.e., $W_{CT}(\pi_1, \pi_2) = \sum_{e \in \pi_1} w_e + \sum_{e \in \pi_2} w_e$.

The appropriate choice between the two options depends on the QoS metric that the weights w_e represent. For example, counting the common link once is a good choice for a metric that stands for a monetary cost, which typically would be paid only once if the link is used by both paths. On the other hand, counting the common link twice is a suitable choice if the QoS metric accounts for an average value (over the employed paths),

e.g. average delay.

For a source-destination pair, there might be several p -survivable connections, among them we would be interested in those that have the best “quality”, giving rise to several tunable survivability optimization problems. Each problem, in turn, has its CO-weight (W_{CO}) formulation, namely a “CO-problem”, and its CT-weight (W_{CT}) formulation, namely a “CT-problem”. The following definition formalizes one of the problems.

Definition 2.6: CT-Constrained QoS Max-Survivability (CT-CQMS) Problem: Given are a network $G(V, E)$, a source node $s \in V$, a destination node $t \in V$ and a QoS bound B . Find a survivable connection $(\pi_1, \pi_2) \in P^{(s,t)} \times P^{(s,t)}$ from s to t such that:

$$\begin{aligned} \max \quad & \prod_{e \in \pi_1 \cap \pi_2} (1 - p_e) \\ \text{s.t.} \quad & W_{CT}(\pi_1, \pi_2) \leq B. \end{aligned}$$

The CO version of the above problem, namely the *CO-Constrained QoS Max-Survivability* (CO-CQMS) Problem, is defined in the same way but replacing the term $W_{CT}(\pi_1, \pi_2)$ with the term $W_{CO}(\pi_1, \pi_2)$. In [13], we also consider dual problems, where we try to minimize the connection's weight while observing a constraint on the level of survivability. Their solution is obtained quite simply out of the solutions of the problems defined above.

In the following sections, we shall establish algorithmic solutions for the above problems, namely CT-CQMS and CO-CQMS. We begin by establishing some interesting structural properties of the CT-problem.

III. THE STRUCTURE OF CT SOLUTIONS

As explained, CT-problems are an important class in which the QoS metric represents either an average or aggregate measure over the employed survivable connection, e.g., the average delay over the two paths of the connection. We proceed to show that, when addressing the optimization problems of the CT class, the links that may affect the survivability level of the optimal solution are restricted to a (typically small) subset of the network's links. We start with the following definitions.

Definition 3.1: Given a survivable connection (π_1, π_2) , a *critical link* is a link $e \in E$ that is common to both paths π_1 and π_2 . Accordingly, the set of critical links of a survivable connection is defined as $\mathbb{C}(\pi_1, \pi_2) = \{e | e \in \pi_1 \cap \pi_2\}$.

Definition 3.2: Given a source s and a destination t , $L^{(s,t)}$ is the set of all the weight-shortest paths between s and t . Note that $L^{(s,t)} \subseteq P^{(s,t)}$.

Definition 3.3: Given a source node $s \in V$ and a destination node $t \in V$, an *in-all-weight-shortest-paths link* is a link $e \in E$ that is common to all paths in $L^{(s,t)}$. Accordingly, the set of in-all-weight-shortest-paths links is defined as $\mathbb{L} = \{e | e \in \bigcap_{\pi \in L^{(s,t)}} \pi\}$.

Note that if there is a unique weight-shortest path between s and t , i.e. $|L^{(s,t)}| = 1$, then \mathbb{L} consists of precisely its links. Moreover, \mathbb{L} is a subset of the set of links of any weight-shortest path. We are ready to present the main result of this section.

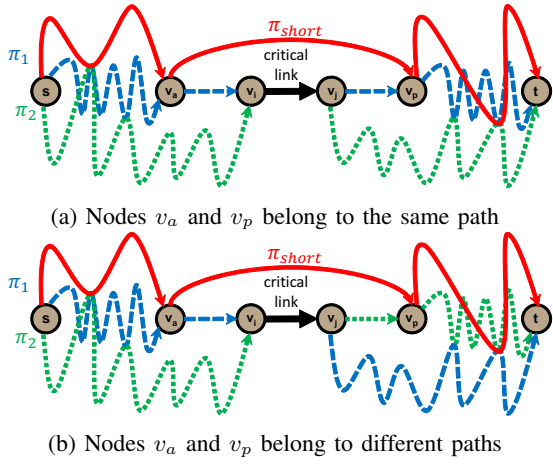


Fig. 2: Intersection between π_{short} and the optimal survivable connection (π_1, π_2)

Theorem 3.1: For any bound B on the additive end-to-end QoS, an (any) survivable connection (π_1, π_2) that is an optimal solution of the respective CT-Constrained QoS Max-Survivability Problem (per Definition 2.6) is such that **all its critical links are in-all-weight-shortest-paths links**. That is, $\mathbb{C}(\pi_1, \pi_2) \subseteq \mathbb{L}$.

Proof: Let (π_1, π_2) be an optimal survivable connection that solves the CT-CQMS Problem (2.6). Assume by contradiction that there is a critical link $e \in \pi_1 \cap \pi_2$ that is not an *in-all-weight-shortest-paths link*, i.e., $\exists e \mid e \in \pi_1 \cap \pi_2 \wedge e \notin \bigcap_{\pi \in \mathcal{L}(s,t)} \pi$. Let v_i and v_j be the nodes of this critical link e , henceforth denoted as $v_i \rightarrow v_j$. From the assumption, there is a weight-shortest path π_{short} that does not contain the critical link $v_i \rightarrow v_j$, i.e., $v_i \rightarrow v_j \notin \pi_{short}$. Moreover, π_{short} is not identical to π_1 nor π_2 , since π_{short} does not contain $v_i \rightarrow v_j$, a common link of both π_1 and π_2 . Consider all nodes that are common to π_{short} and to at least one of the optimal survivable connection paths, π_1 or π_2 . Denote by v_a the last such common node on the corresponding sub-path from the source s to v_i . Similarly, v_p denotes the first such common node on the corresponding sub-path from v_j to the destination t . Note that v_a, v_p can include v_i, v_j as well as the source s and the destination t , respectively. From the assumption, a pair of disjoint paths between v_a to v_p necessarily exists. Moreover, π_{short} intersects with either π_2 or π_1 , only in the sub-paths between s to v_a or v_p to t . Through Figure 2, we proceed to consider two possible cases of an intersection between π_{short} (the full-lines path) and the optimal survivable connection (π_1, π_2) (the dashed-lines paths).

In the first case, illustrated in Figure 2a, v_a and v_p belong to the same path in the optimal survivable connection (π_1, π_2) . Without loss of generality, we assume that $v_a, v_p \in \pi_1$. Consider the pair of sub-paths from v_a to v_p , where one path contains links from π_1 and the other path contains links from π_{short} denoted as $(\pi_1^{sub}, \pi_{short}^{sub})$. It is obvious from the definition that $(\pi_1^{sub}, \pi_{short}^{sub})$ are disjoint. Denote by π_1^{pre} the sub-path of π_1 from the source s to v_a , and by π_1^{post} the sub-

path of π_1 from v_p to the destination t . Now, define a new path π_{new} described as $\pi_1^{pre} \rightarrow v_a \rightarrow \pi_{short}^{sub} \rightarrow v_p \rightarrow \pi_1^{post}$, composed by the sub-paths π_1^{pre} , π_{short}^{sub} and π_1^{post} . Consider the survivable connection (π_{new}, π_2) . Since $\pi_{new} \cap \pi_2$ does not include the critical link $v_i \rightarrow v_j$, the survivability level of (π_{new}, π_2) is higher than that of (π_1, π_2) , i.e.

$$\prod_{e \in \pi_{new} \cap \pi_2} (1 - p_e) > \prod_{e \in \pi_1 \cap \pi_2} (1 - p_e).$$

Also, since for additive metrics a sub-path of a weight-shortest path is also a weight-shortest path between its endpoints, we have that $W(\pi_{short}^{sub}) \leq W(\pi_1^{sub})$. Therefore, the CT-weight of (π_{new}, π_2) is not larger than that of (π_1, π_2) , i.e. $W_{CT}(\pi_{new}, \pi_2) \leq W_{CT}(\pi_1, \pi_2)$. Thus, the survivable connection (π_{new}, π_2) strictly outperforms (π_1, π_2) in terms of survivability while not incurring a higher weight, which contradicts the assumption that (π_1, π_2) is optimal.

In the second case, illustrated in Figure 2b, v_a and v_p belong to different paths in the optimal survivable connection (π_1, π_2) . Without loss of generality, we assume that $v_a \in \pi_1$ and $v_p \in \pi_2$. Denote π_1 's sub-paths from the source to v_i and from v_j to the destination as π_1^{pre} and π_1^{post} , respectively. Similarly, we define π_2^{pre} and π_2^{post} . Now, consider the survivable connection (γ_1, γ_2) described as $(\pi_1^{pre} \rightarrow v_i \rightarrow v_j \rightarrow \pi_2^{post}, \pi_2^{pre} \rightarrow v_i \rightarrow v_j \rightarrow \pi_1^{post})$, composed by the sub-paths π_1^{pre} , π_1^{post} , π_2^{pre} and π_2^{post} and the critical link $v_i \rightarrow v_j$. It has precisely the same links as (π_1, π_2) , hence it has the *same survivability level*, i.e. $\prod_{e \in \pi_1 \cap \pi_2} (1 - p_e) = \prod_{e \in \gamma_1 \cap \gamma_2} (1 - p_e)$, and the same CT-weight, i.e. $W_{CT}(\pi_1, \pi_2) = W_{CT}(\gamma_1, \gamma_2)$. Furthermore, v_a and v_p belong to the same path in the survivable connection (γ_1, γ_2) i.e., we are back in the realm of the first case. ■

We shall employ the above property of the CT-problem in order to reduce the computational complexity of the solution algorithms. Furthermore, we shall exploit this property in order to establish a design scheme for efficiently upgrading the performance of the network in terms of survivability.

IV. ESTABLISHING QoS AWARE p -SURVIVABLE CONNECTIONS

In [13], we prove that our previously formulated optimization problems, namely CT-CQMS and CO-CQMS, are *NP-Hard*. However, efficient solution schemes are still possible. Indeed, in this section, we shall establish exact solutions of pseudo-polynomial complexity, and near (i.e., ϵ -optimal) solutions of polynomial complexity, for the two considered problems.

The solution approach is based on a graph transformation that reduces our problem to a standard *Restricted Shortest Path* (RSP) problem. We recall that RSP is the problem of finding a shortest (in terms of an additive metric) path while obeying an additional (additive) constraint. Although the RSP problem is known to be NP-Hard [14], the literature provides several pseudo-polynomial solutions [15] as well as ϵ -optimal fully polynomial approximation schemes (FPTAS) [16], which we employ in order to solve our problems. Moreover, we use the findings of Section III in order to further reduce the complexity of the solutions for the CT problem.

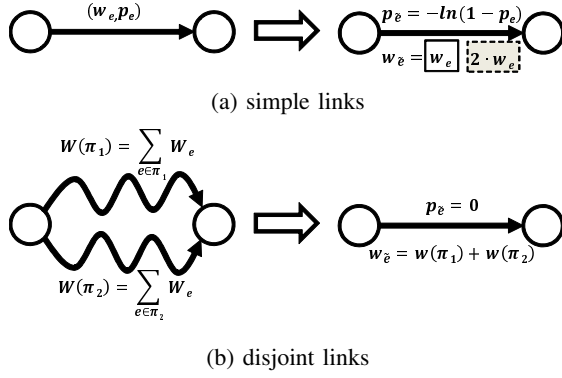


Fig. 3: CO/CT links transformation

A. Pseudo-Polynomial Schemes for Establishing CO-QoS Aware p -Survivable Connections

We begin by establishing a pseudo-polynomial algorithmic scheme for solving the CO-CQMS Problem, denoted as the *CO-QoS Aware Max Survivable Connection* (CO-QAMSC) Algorithm and specified in Figure 4. Note that the algorithm does not include the dashed-boxed text (with gray background), which shall be later used for handling the CT-CQMS Problem. The CO-QAMSC Algorithm employs two well-known algorithms: the first, *Edge-Disjoint Shortest Pair* (EDSP) Algorithm [11], finds two edge-disjoint paths with minimum sum of edge weight between two nodes in a weighted directed graph; the second is a pseudo-polynomial algorithmic scheme, such as [15], for solving the NP-Hard RSP problem. The CO-QAMSC Algorithm consists of three stages.

The first stage comprises the construction of a transformed network $\tilde{G}(\tilde{V}, \tilde{E})$, where each link is associated with two metrics: a weight $w_{\tilde{e}}$ and a success probability $p_{\tilde{e}}$. Specifically, the transformed network consists of two types of links, as follows. The first one, denoted as *simple link*, consists of the original network links. The weight of a simple link is set to be the weight of the original link. The success probability of a simple link is set to $p_{\tilde{e}} = -\ln(1 - p_e)$, thus transforming our multiplicative (survivability) metric into an additive one. The second type of links, denoted as *disjoint link*, consists of additional links representing possible *Edge-Disjoint Shortest Pair of Paths* (EDSPoP) between pairs of nodes in the network. The weight of a disjoint link is set to be the weight of the EDSPoP between these two nodes, which we compute by employing the EDSP Algorithm [11]. The success probability of a disjoint link is set to be 0, due to the fact that a disjoint path provides full protection against a single link failure. Figure 3 illustrates these transformations, where the dashed-boxed text (with gray background) should be disregarded at this stage. Given the above transformed network $\tilde{G}(\tilde{V}, \tilde{E})$, the second stage calculates a restricted shortest path that minimizes the sum of its $p_{\tilde{e}}$ from the source s to the destination t with a constraint of B . Specifically, the algorithm finds a pair of paths

Input: $G(V, E)$ -network, s -source, t -destination, p_e -link failure probability, w_e -link weight, B -weight constraint

Variables: $\tilde{G}(\tilde{V}, \tilde{E})$ -transformed network, \tilde{s} -source, \tilde{t} -destination, $p_{\tilde{e}}$ -link success probability, $w_{\tilde{e}}$ -link weights, π_{min} -a weight-shortest path, $\tilde{\pi}$ -RSP solution

Stage 0 – Shortest Path Search

0.1) Employ a Shortest-Path Algorithm[17] between s and t in order to find a weight shortest path π_{min} .

Stage 1 - Transformed network $\tilde{G}(\tilde{V}, \tilde{E})$ construction

1.1) $\tilde{V} \leftarrow V$.

1.2) **foreach** $e : u \rightarrow v \in E$ $\tilde{\pi}_{min}$ **do**

– Construct a “simple link” \tilde{e} between \tilde{u} and \tilde{v} .

– Assign $w_{\tilde{e}}$ to be $\lfloor w_e \rfloor \cdot \lceil \frac{1}{2} \cdot \frac{1}{w_e} \rceil$.

– Assign $p_{\tilde{e}}$ to be $-\ln(1 - p_e)$.

end

1.3) **foreach** $u \in V$ $\tilde{\pi}_{min}$ **do**

foreach $v \in V$ $\tilde{\pi}_{min}$ **do**

– Employ the EDSP Algorithm [11] between u and v .

if (there is a solution to the EDSP Algorithm) **then**

– Construct a “disjoint link” \tilde{e} between \tilde{u} and \tilde{v} .

– Assign $w_{\tilde{e}}$ to be the sum of w_e ’s of the

Edge-Disjoint Shortest Pair of Paths (EDSPoP).

– Assign $p_{\tilde{e}}$ to be 0.

end

end

end

Stage 2 - RSP Calculation

2.1) Solve the instance $\langle \tilde{G}(\tilde{V}, \tilde{E}), \tilde{s}, \tilde{t}, p_{\tilde{e}}, w_{\tilde{e}}, B \rangle$ of the RSP problem [15] with a constraint of B while minimizing the survivability level of a path.

2.2) **if** there is no feasible solution for the RSP **then**

– **return Fail**

end

else

– Let $\tilde{\pi}$ represent the solution of the RSP problem.

end

Stage 3- Survivable Connection (π_1, π_2) Construction

3.1) **foreach** $\tilde{e} \in \tilde{\pi}$ **do**

if \tilde{e} is a “simple link” **then**

– π_1 and π_2 contain the corresponding e link.

end

else if \tilde{e} is a “disjoint link” **then**

– π_1 contains the links of one the paths of the EDSPoP represented by \tilde{e} .

– π_2 contains the links of the other path of the EDSPoP represented by \tilde{e} .

end

end

return (π_1, π_2)

Fig. 4: CO/CT-QoS Aware Max Survivable Connection (CO/CT-QAMSC) Algorithm

that minimizes

$$- \sum_{e \in \pi_1 \cap \pi_2} \ln(1 - p_e) = -\ln \prod_{e \in \pi_1 \cap \pi_2} (1 - p_e)$$

and, therefore, maximizes $\prod_{e \in \pi_1 \cap \pi_2} (1 - p_e)$, the connection's survivability level. Here, we may employ any pseudo-polynomial time algorithm, e.g. [15], for solving the RSP problem. In [13], we show that there is no solution to our problem if there is no feasible solution to the defined RSP problem. Note that each disjoint link is associated with a pair of disjoint paths in the original network, while each simple link is associated with a regular link. Accordingly, in the third stage, we construct the sought pair of paths of a survivable connection (π_1, π_2) out of the links of the RSP solution. Then, the algorithm outputs the optimal survivable solution (π_1, π_2) .

The following theorem establishes the correctness of the CO-QAMSC Algorithm.

Theorem 4.1: Given are a network $G(V, E)$, a pair of nodes s and t and a co-weight constraint $B \geq 0$. If there exists a survivable connection with a CO-weight of at most B , then the CO-QAMSC Algorithm returns a survivable connection that is a solution to a CO-CQMS Problem; otherwise, the algorithm fails.

Proof: See [13]. ■

We proceed to analyze the running time of the CO-QAMSC Algorithm. As mentioned, the input size is represented by N and M , which are the numbers of nodes and links in the network, respectively. We denote by $R(N, M)$ and $D(N, M)$ the running time expressions of the employed (standard) RSP algorithm and EDSP algorithm respectively.

Theorem 4.2: The time complexity of the CO-QAMSC Algorithm is $O(N^2 \cdot D(N, M) + R(N, N^2))$, i.e. $O(M \cdot N^2 + N^3 \cdot (\log(N) + B))$.

Proof: See [13]. ■

B. Pseudo-Polynomial Schemes for Establishing CT-QoS Aware p -Survivable Connections

We will now exploit the rather salient property of the optimal solutions to the CT-problem, as established in Section III. This will allow us to improve the computational complexity of the algorithmic solution of the CT-CQMS Problem. We proceed to present an algorithmic scheme for solving the CT-CQMS Problem (2.6), denoted as the *CT-QoS Aware Max Survivable Connection* (CT-QAMSC) Algorithm. The algorithm is specified (again) in Figure 4, however now the full-line-boxed text should be disregarded while the dashed-boxed text (with gray background) should be considered.

The CT-QAMSC Algorithm is similar to the previously presented CO-QAMSC Algorithm, except for two important changes. The first is in the transformation of simple links in the new constructed network in step 1.2. Recall that simple links represent critical links of the solution, i.e., the survivable connection (π_1, π_2) . Since in the CT problems the weight of each such link is counted twice, the weight of simple links is set to be twice the weight of the links in the original network, as illustrated (again) in Figure 3a, where the full-line-boxed

text should be disregarded now while the dashed-boxed text (with gray background) should be considered.

The second change is the addition of a preliminary stage, namely Stage 0, to the CT-QAMSC Algorithm. At this initial stage, the algorithm first finds a weight-shortest path in the network $G(V, E)$ by employing a well-known shortest path algorithm, such as [17]. According to Theorem 3.1, in an optimal solution of a CT problem, each of the critical links is included in any weight-shortest path. Therefore, we can have the CT-QAMSC Algorithm focus on just nodes and links that belong to some (any) weight-shortest path. Accordingly, at Stage 1 of the CT-QAMSC Algorithm, the transformed network $\tilde{G}(\tilde{V}, \tilde{E})$ is limited to simple links that correspond to the weight-shortest path found at Stage 0, and to disjoint links that correspond to EDSPoPs between pairs of nodes of the identified weight-shortest path. As shall be shown, this change improves the computational complexity of the solution.

We denote the number of links in the identified weight-shortest path as k . The running time expression of the weight-shortest path algorithm is denoted as $SP(N, M)$.

Theorem 4.3: The time complexity of the CT-QAMSC Algorithm is $O(SP(N, M) + k^2 \cdot D(N, M) + R(k, k^2))$, i.e. $O(k^2 \cdot (M + N \cdot \log(N)) + k^3 \cdot B)$.

Proof: See [13]. ■

According to [18], in power-law networks, which are known to be a good model for some portions of the Internet [19], the number of links of the shortest path grows proportionally to the logarithm of the number of the network nodes, i.e. $k \sim \log(N)$. Therefore, the running time of our algorithm can be significantly reduced in such networks.

C. Approximation Schemes for Establishing QoS Aware Survivable Connections

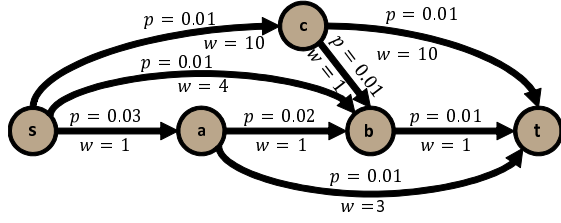
We proceed to establish Fully Polynomial Time Approximation Schemes (FPTAS) for the considered problems.

First, we establish that an ε -approximation scheme for the CO-CQMS Problem can be accomplished by employing the previously defined CO-QAMSC Algorithm (Fig. 4) with the following change. Consider a desired approximation ratio ε , and recall the minimum survivability level $S_{min} = (1 - p_{min})^M$ specified in Section II. In Stage 2 of the CO-QAMSC Algorithm, instead of employing a pseudo-polynomial (exact) solution scheme, apply an (any of those proposed in the literature, e.g. [16]) FPTAS for solving the RSP problem with an approximation ratio of $\frac{-\ln(1+\varepsilon)}{\ln S_{min}}$. This modified algorithm shall be referred to as the F-CO-QAMSC Algorithm.

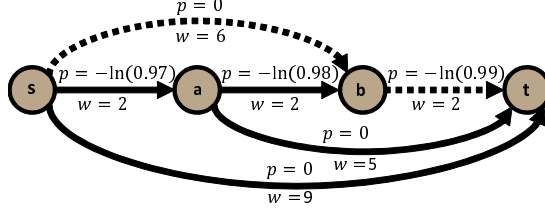
Theorem 4.4: The F-CO-QAMSC Algorithm is a Fully Polynomial Time Approximation Scheme (FPTAS) for the CO-CQMS Problem. Specifically, the weight of the provided connection is bounded by B (as required) and its survivability level is at most $(1 + \varepsilon)$ smaller than the optimal survivability level. The time complexity of the algorithm is bounded by $O(N^2 \cdot (M + N \cdot \log(N)) + N^3 \cdot (\log \log(N) + \frac{M}{\varepsilon}))$.

Proof: See [13]. ■

A Fully Polynomial Time Approximation Scheme (FPTAS) for the CT-CQMS Problem can be established by employing



(a) Network illustrating the operation of the algorithm



(b) CT-algorithm transformed network

Fig. 5: Example of the algorithm execution

the same approach to the CT-QAMSC Algorithm.

D. A Numerical Example

We further demonstrate the operation of the CT-QAMSC algorithm through an example, depicted in Figure 5.

Consider the network illustrated in Figure 5a, where the links weights w_e and failure probabilities p_e are depicted next to each link. Assume that we aim at finding a survivable connection (π_1, π_2) with an upper-bound of 8 on the total weight. As shown in Figure 4, the CT-QAMSC Algorithm starts by finding a weight-shortest path between the source and destination at Stage 0, which in this case is the path $\pi_{min} = \langle s, a, b, t \rangle$. Consequently, the next stage only focuses on the nodes and links of the shortest path π_{min} . At Step 1.2, the algorithm creates the simple links of the transformed network by duplicating the links of the shortest path π_{min} and setting its weight to be $2 \cdot w_e$ and its survivability level to be $-\ln(1 - p_e)$. At Step 1.3, the algorithm finds, between each couple of nodes along the shortest path $\pi_{min} = \langle s, a, b, t \rangle$, an Edge-Disjoint Shortest Pair of Paths (EDSPoP). In this case, three EDSPoPs are found: the first is found between node s and node b (π_1^{sb}, π_2^{sb}) = ($\langle s, b \rangle, \langle s, a, b \rangle$) with a total weight of 6, the second is found between node a and node t (π_1^{at}, π_2^{at}) = ($\langle a, t \rangle, \langle a, b, t \rangle$) with a total weight of 5, and the third is found between node s and node t (π_1^{st}, π_2^{st}) = ($\langle s, b, t \rangle, \langle s, a, t \rangle$) with a total weight of 9. We create a disjoint link for each of the above three EDSPoPs, setting its survivability level to 0 and its weight to the EDSPoP's total weight. At the end of Stage 1, we obtain the transformed network illustrated in Figure 5b. At stage 2, in the transformed network, the algorithm solves the RSP problem, considering a bound of 8. Accordingly, we obtain the dashed path $\tilde{\pi} = \langle s, b, t \rangle$ in Figure 5b. Finally, at stage 3, the algorithm constructs and outputs the survivable connection $(\pi_1, \pi_2) = (\langle s, b, t \rangle, \langle s, a, b, t \rangle)$.

V. SIMULATION STUDY

In this section, we demonstrate the advantages of employing tunable survivability over the traditional protection (full survivability) schemes. For concreteness, we consider delay as the additive QoS metric. Through comprehensive simulations, we compare between the minimum delay of the optimal p -survivable connections, where $p \in [0.9, 1)$, and the minimum delay of the optimal 1-survivable connections, the latter being obtained through pairs of disjoint paths. In particular, we show that, by slightly relaxing the traditional requirement of 100% protection, major improvement in terms of delay is accomplished.

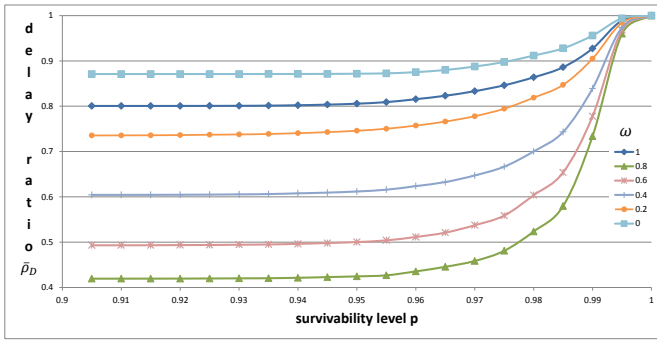
A. Setup

We generated two classes of random networks, namely Power-Law [19] topologies and Waxman [20] topologies. The Power-Law topology has been shown to quite adequately model typical network interconnections, in particular in the context of the Internet [19]. We demonstrate that our findings extend to other classes of network topologies by experimenting with another well known class, namely Waxman topologies.

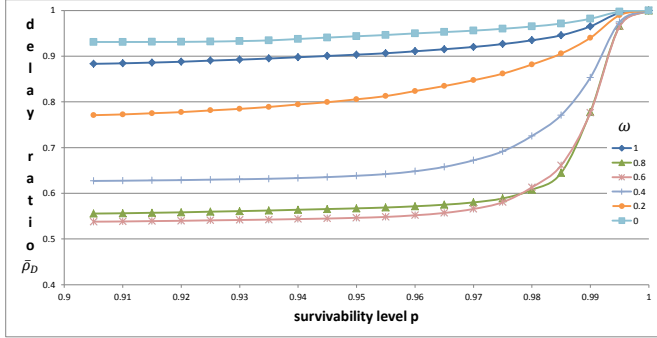
We generated 10000 random networks, each containing 200 nodes, in which we identified a source-destination pair, in a manner that shall be explained later. For ease of presentation, in this section we consider a “reverse” version of the CT-CQMS Problem (2.6), in which we minimize delay under a survivability constraint. As mentioned, a (fully polynomial approximation) solution to this problem, termed the CT-TSMQ Algorithm, is presented in [13], where it is obtained through a simple change of the above presented CT-QAMSC Algorithm. For each generated network and survivability level constraint S in the range of $[0.9, 1]$ with intervals of 0.005, we employed the CT-TSMQ Algorithm for the Power-Law class and for the Waxman class. We then considered only those networks that admit 1-survivable connections (i.e., sustain a pair of disjoint paths between source and destination). For each such network, we measured the minimum delay of a p -survivable connection, denoted as $D(p)$, and computed the *delay ratio*, defined as $\rho_D(p) = \frac{D(p)}{D(1)}$. Finally, we derived the corresponding average delay ratio $\bar{\rho}_D(p)$, computed over all considered network instances (of either the Power-Law or Waxman class).

In terms of delay, we considered two types of links: “slow” links, whose delay is set to 100 time units, and “fast” links, whose delay is randomly (uniformly) distributed in $[1, 5]$ time units. This choice represents typical mixes of links, e.g., satellite links with large propagation delays vs. terrestrial links, or low-bandwidth (hence, large transmission delay) links vs. high-bandwidth links. Specifically, a link was classified as “fast” with probability of $\omega \in [0, 1]$ and as “slow” otherwise, i.e. with probability of $1 - \omega$. We ran simulations for each $\omega \in [0, 1]$ value in steps of 0.1. The failure probability of each link was distributed normally with a mean of 1% and a standard deviation of 0.3%.

We proceed to further specify the generation of the random topologies. For Power-law topologies, following [19], we



(a) Power Law simulations for different values of ω



(b) Waxman simulations for different values of ω

Fig. 6: Average Delay Ratio versus Survivability Level

randomly assigned a certain number of out-degree credits to each node, using the power-law distribution $\beta \cdot x^{-\alpha}$, where x is a random number out of the number of network nodes, $\alpha = 0.756$ and $\beta = 100$. We connected the nodes so that every node obtained the assigned out-degree. Specifically, we randomly picked pairs of nodes u and v , such that u still had some remaining out-degree credits and then assigned a directed link $u \rightarrow v$ between them in case that such a link had not been assigned yet. Upon assigning such a new link, we decreased the out-degree credit of node u . Each simulated Power-law networks consists of 200 nodes, and, in average, 900 links.

We turn to specify the generation of the Waxman topologies, following the lines of [20]. Initially, we located the source and the destination at the diagonally opposite corners of a square of unit dimension. Then, we randomly spread 198 nodes over the square. Finally, for each pair of nodes u, v we introduced a link (u, v) with the following probability, where $\delta(u, v)$ is the distance between the nodes:

$$p(u, v) = \alpha \cdot \exp \frac{-\delta(u, v)}{\beta \cdot \sqrt{2}}$$

considering $\alpha = 1.8$ and $\beta = 0.05$. Each simulated Waxman network consists of 200 nodes, and, in average, 1800 links.

B. Results

The simulation results are illustrated in figure 6. We recall that the average delay ratio $\bar{\rho}_D(p)$ is a normalized metric for comparing the improvement of p -survivable connections

over the traditional fully disjoint path approach (i.e., 1-survivable connections). The number of networks that admitted 1-survivable connections was in the range of 7000 to 8000 (out of 10,000), hence the samples were always significant.

The chart depicted in Figure 6 presents the average delay ratio improvement as a function of the required level of survivability, for different mixes of “fast” and “slow” links (i.e., values of ω), for each of the two classes of network topologies, namely Power Law and Waxman. Overall, we observe that a modest relaxation, of a few percents in the survivability level, is enough to provide significant improvement in terms of delay. Specifically, for Power Law networks (Fig. 6a), alleviating the survivability level by about 5% provides an improvement of about 20% for the homogeneous case of all-fast links (i.e., $\omega = 1$), and it grows to about 40%–60% for the heterogeneous cases in the range $\omega = 0.4 - 0.8$. Quite similar results are observed for Waxman networks (Fig. 6b).

Moreover, in all cases, most of the delay improvement is already achieved by alleviating the survivability level by about 1.5%. We thus conclude that, in a typical setting, where there is some presence of relatively slower links (e.g., due to large propagation delays or low bandwidth), a *modest alleviation* in the survivability level about *doubles the performance* in terms of delay. As an aside, we note that the results for ω and $1 - \omega$ are not symmetric, since the delays of “fast” links are taken out of an interval of values whereas “slow” links assume a single delay value.

VI. A NETWORK DESIGN PERSPECTIVE

Suppose that we are provided with a “budget” in order to improve the total survivability level between a couple of nodes in the network, by way of upgrading the links in terms of their robustness to failures. Within this problem setting and the class of CT problems, we proceed to indicate how to exploit the particular structure of the CT solutions that has been established in section III. Theorem 3.1 significantly reduces the amount of links that affect the optimal solution of the CT problems. Specifically, the set of candidate critical links \mathbb{C} is limited to a (typically small) subset of E , namely the in-all-weight-shortest-paths links \mathbb{L} . This means that only these links should be considered as candidates for an upgrade.

A. Discovering the in-all-weight-shortest-paths links

We begin by sketching an algorithmic scheme for finding the in-all-weight-shortest-paths links set \mathbb{L} , denoted as the *In-All-Weight-Shortest-Paths Links* (IAWSPL) Algorithm; the details can be found in [13].

Given are a network $G(V, E)$ and a pair of nodes s and t . First, our scheme finds a weight-shortest path π_{short} between s and t and its weight l in the original network $G(V, E)$ by employing a well-known shortest path algorithm, such as [17]. For each link e in that weight-shortest path π_{short} , consider $\tilde{G}(\tilde{V}, \tilde{E})$, which is a replica of the original network $G(V, E)$ excluding the specified link e . Next, find in the network $\tilde{G}(\tilde{V}, \tilde{E})$ a weight-shortest path between s and t and its weight \tilde{l} , which is, clearly, greater than or equal to l . If its weight \tilde{l} is

greater than l , then the excluded link e belongs to the in-all-weight-shortest-paths links set \mathbb{L} . Otherwise, i.e., if its weight \tilde{l} is equal to l , then the excluded link e does not belong to the set \mathbb{L} . This process is repeated for all links of the weight-shortest path between s and t of the original graph $G(V, E)$.

B. Optimal Links Upgrade Problem

We proceed to formulate a network design problem that seeks to allocate a given “upgrade budget” among the various links of the network, in a way that optimizes the total survivability level between a given pair of nodes. According to Theorem 3.1, we should limit our attention only to the links that belong to the in-all-weight-shortest-paths links set \mathbb{L} . Consequently, we should execute the IAWSPL Algorithm in order to find \mathbb{L} .

Given a network $G(V, E)$, each link $e \in \mathbb{L}$ is associated with a cost u_e , referred to as its *upgrade level*. Accordingly, the *upgrade vector* is the vector of the upgrade levels of all links in the set \mathbb{L} , i.e. $U = (e \in \mathbb{L} | u_e)$. The upgrade level constitutes an additive improvement to the link’s survivability level, i.e., its success probability. Such an upgrade incurs some (monetary) cost, which is considered to be equal to the upgrade level. Since the survivability level of any link can never exceed 1(100%), the upgrade level of a link cannot exceed p_e . We thus define the following optimization problem.

Definition 6.1: Optimal Additive Upgrade Problem: Given are a network $G(V, E)$, a source node $s \in V$, a destination node $t \in V$, the in-all-weight-shortest-paths links set \mathbb{L} between s and t and an upgrade budget \mathbf{B} . Each link $e \in E$ in the network is associated with a failure probability value $p_e \in (0, 1)$. Find an upgrade vector $U = (e \in \mathbb{L} | u_e)$ such that:

$$\begin{aligned} \max \quad & \prod_{e \in \mathbb{L}} (1 - p_e + u_e) \\ \text{s.t.} \quad & \sum_{e \in \mathbb{L}} u_e \leq \mathbf{B} \\ & \forall e \in \mathbb{L} \quad u_e \geq 0 \\ & \forall e \in \mathbb{L} \quad u_e \leq p_e. \end{aligned}$$

In [13], we show that the above optimization problem can be transformed into an instance of the well-known Water-filling problem [21]. Consequently, the optimal solution is to repeatedly split the upgrade budget among the links of the in-all-weight-shortest-paths links set \mathbb{L} with the (currently) highest failure probability, until either the budget is exhausted or all the links assume zero failure probability.

Furthermore, we consider other optimization variants in [13], e.g. a variant where the costs of upgrades are related to multiplicative (rather than additive) improvement factors.

VII. CONCLUSIONS

Tunable survivability is a novel quantitative approach, which can be tuned to accommodate any desired level (0%-100%) of survivability, while alleviating the full (100%) protection requirement of the traditional survivability schemes. In this work, we established efficient algorithmic schemes for optimizing the level of survivability while obeying an additive end-to-end QoS constraint. Additionally, for an important class of problems, we characterized a fundamental property, by which the links that affect the total survivability level of the optimal routing paths

belong to a typically small subset. This finding gave rise to an efficient design scheme for improving the network end-to-end survivability. Finally, through comprehensive simulations, we demonstrated the advantage of tunable survivability over traditional survivability schemes.

We are currently investigating the practical aspects of our findings in order to implement tunable survivability schemes in MPLS network architectures, similarly to [12].

While there is still much to be done towards the actual deployment of the tunable survivability approach, we believe that this study provides evidence to the profitability of implementing this novel concept, as well as useful insight and building blocks towards the construction of a comprehensive solution.

Acknowledgement: This research was supported in part by the European Union through the ETICS project (<https://www.ict-etics.eu/>) in the 7th Framework Programme and in part by the Israel Science Foundation (grant no. 1129/10).

REFERENCES

- [1] C. R. and Cole, “100-gb/s and beyond transceiver technologies,” *Optical Fiber Technology*, vol. 17, no. 5, pp. 472 – 479, 2011.
- [2] N. Sprecher and A. Farrel, “MPLS Transport Profile (MPLS-TP) Survivability Framework,” RFC 6372, Sep. 2011.
- [3] A. Askarian *et al.*, “Protection and restoration from link failures in DWDM networks: A cross-layer study,” in *IEEE ICC*, 2008.
- [4] R. Banner and A. Orda, “Designing low-capacity backup networks for fast restoration,” in *IEEE Infocom*, 2010.
- [5] P. Heegaard and K. Trivedi, “Network survivability modeling,” *Computer Networks*, vol. 53, no. 8, pp. 1215–1234, 2009.
- [6] J. Tapolcai *et al.*, “On monitoring and failure localization in mesh all-optical networks,” in *IEEE Infocom*, 2009.
- [7] R. Banner and A. Orda, “The power of tuning: A novel approach for the efficient design of survivable networks,” *IEEE/ACM Trans. Networking*, vol. 15, no. 4, pp. 737 –749, aug. 2007.
- [8] A. Fumagalli and M. Tacca, “Optimal design of optical ring networks with differentiated reliability (dir),” in *QoS-IP*, 2001.
- [9] D. Xu *et al.*, “On the complexity of and algorithms for finding the shortest path with a disjoint counterpart,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 147–158, 2006.
- [10] C. Li *et al.*, “The complexity of finding two disjoint paths with min-max objective function,” *Discrete Applied Mathematics*, vol. 26, no. 1, pp. 105–115, 1990.
- [11] J. W. Suurballe and R. E. Tarjan, “A quick method for finding shortest pairs of disjoint paths,” *Networks*, vol. 14, no. 2, pp. 325–336, 1984.
- [12] A. Sprintson *et al.*, “Reliable routing with qos guarantees for multi-domain ip/mpls networks,” in *IEEE Infocom*, 2007.
- [13] J. Yallouz and A. Orda, “Tunable QoS-Aware network survivability,” Comnet, Technion, Israel, Tech. Rep., 2012. [Online]. Available: <http://www.ee.technion.ac.il/Sites/People/ArielOrda/Info/Other/YO12.pdf>
- [14] M. R. Garey and D. S. Johnson, “*Computers and Intractability: A Guide to the Theory of NP-Completeness*”. W. H. Freeman & Co., 1979.
- [15] H. Joksche, “The shortest route problem with constraints,” *Journal of Mathematical analysis and applications*, vol. 14, pp. 191–197, 1966.
- [16] D. H. Lorenz and D. Raz, “A simple efficient approximation scheme for the restricted shortest path problem,” *Operations Research Letters*, vol. 28, no. 5, pp. 213 – 219, 2001.
- [17] M. Fredman and R. Tarjan, “Fibonacci heaps and their uses in improved network optimization algorithms,” *J. of the ACM*, vol. 34, no. 3, pp. 596–615, 1987.
- [18] L. Braunstein *et al.*, “Optimal paths in disordered complex networks,” *Physical review letters*, vol. 91, no. 16, p. 168701, 2003.
- [19] M. Faloutsos *et al.*, “On power-law relationships of the internet topology,” in *ACM SIGCOMM*, 1999.
- [20] B. Waxman, “Routing of multipoint connections,” *IEEE JSAC*, vol. 6, no. 9, pp. 1617–1622, 1988.
- [21] S. Boyd and L. Vandenberghe, “*Convex Optimization*”. Cambridge University Press, 2004.