Tunable Survivable Spanning Trees

Jose Yallouz, Ori Rottenstreich and Ariel Orda
Department of Electrical Engineering
Technion, Israel Institute of Technology
{jose@tx, or@tx, ariel@ee}.technion.ac.il

ABSTRACT

Coping with network failures has become a major networking challenge. The concept of tunable survivability provides a quantitative measure for specifying any desired level (0%-100%) of survivability, thus offering flexibility in the routing choice. Previous works focused on implementing this concept on unicast transmissions. However, vital network information is often broadcasted via spanning trees. Accordingly, in this study, we investigate the application of tunable survivability for efficient maintenance of spanning trees under the presence of failures. We establish efficient algorithmic schemes for optimizing the level of survivability under various QoS requirements. In addition, we derive theoretical bounds on the number of required trees for maximum survivability. Finally, through extensive simulations, we demonstrate the effectiveness of the tunable survivability concept in the construction of spanning trees. Most notably, we show that, typically, *negligible* reduction in the level of survivability results in *major* improvement in the QoS performance of the resulting spanning trees.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Reliability, availability, and serviceability; Fault tolerance; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms; Network problems; Trees*

Keywords

Survivability; Reliability; Fault-Tolerance; Spanning-Tree

1. INTRODUCTION

Network infrastructures have been progressing very rapidly. While about a few decades ago the $56Kbps$ dial-up modem, was a widely deployed device, nowadays technologies, such as Ethernet and InfiniBand operate at rates of $100Gbps$ and beyond [1]. With this extreme increase in transmission rates, any failure in the network infrastructure may lead to a vast amount of data loss. Hence, failures in the network should be recovered promptly. Accordingly, common standards, e.g. [2] [3], introduce requirements for recovery from a single failure within $50ms$. Several studies have focused on survivability methods for coping with network failures, e.g. [4] [5] [6] [7] [8]. However, most of these studies focus on the protection of *unicast* transmission between a pair of nodes.

A widely employed transmission method is to *broadcast* a message to all recipients simultaneously. Indeed, several protocols at various layers utilize broadcast functionality, e.g. ARP [9], DHCP [10]. While the straightforward broadcast technique of flooding the data over every network link ensures maximum protection from failures, this approach incurs a heavy, at times prohibitive, toll in terms of communication overhead. Furthermore, in order to prevent undesirable effects such as a "broadcast storm" [11], the flooding approach requires the implementation of a complex control mechanism for terminating the transmission process.

A *spanning tree*, i.e. a tree composed of all network nodes, offers an alternative approach for broadcasting messages with minimal communication overhead precluding network loops. Indeed, spanning trees are often employed for broadcasting in various networking environments, such as the Ethernet local area network [12], in which the Spanning Tree Protocol (STP) ensures a single transmission path between any two Ethernet local network nodes. However, the basic implementation of this protocol suffers from lengthy recovery period of about $30-50$ seconds. Accordingly, the Multiple Spanning Tree Protocol [13] has been proposed as an extension for the STP protocol; it enables the establishment of different spanning trees in a local network to accelerate STP recovery period [14].

Spanning trees are not limited to the Ethernet framework, and are also employed in the context of wireless and optical networks. In sensor [15] [16] and ad-hoc [17] networks, spanning trees are utilized to enable efficient energy consumption. In optical networks [18], shared backup trees can be employed to protect a group of working light paths towards the same destination in multi-protocol lambda switching networks. This paper presents *a novel generic approach for network failure protection when employing spanning trees.*

We adopt the widely used single-link failure model that has been the focus of most of the studies on survivability, e.g. [19] [20] [21]. This model is selected for its simplicity and the fact that it addresses the common requirement in various standards of protecting against a single failure, e.g. [2] [3]. According to this model, reliable broadcasting can

SIGMETRICS'14, June 16–20, 2014, Austin, Texas, USA.

http://dx.doi.org/10.1145/2591971.2591997.

be accomplished by transmitting the data through a pair of link-disjoint spanning trees. Algorithmic solutions for finding such trees have been studied in [22] and [23]. Moreover, some interesting properties related to the number of disjoint spanning trees were presented in [24] and [25]. The employment of disjoint spanning trees has been considered also in the context of specific network architectures, e.g. in [26], where it was proposed to improve both aggregate throughput and fault-tolerance in metropolitan area Ethernet networks.

However, the requirement of full link-disjointedness is often too restrictive and, in practice, demands excessive redundancy. Link-disjoint spanning trees of sufficient quality may not exist, occasionally making the requirement unfeasible. Therefore, a milder and more flexible survivability concept is called for, which would relax the rigid requirement of disjointedness by also considering trees that do contain common links.

Previous studies [19] [27] introduced the concept of *tunable survivability*, which provides a quantitative measure for specifying the desired level of survivability. This concept allows any degree of survivability in the range of 0% to 100%, thus transforming survivability into a quantifiable Quality of Service (QoS) metric. Consequently, these studies showed how to combine this new survivability metric with other QoS guarantees, such as bottleneck QoS metrics [19] and additive QoS metrics [27]. However, these studies focused solely on the survivability between a source-destination pair, i.e. a unicast connection. The goal of this work is to employ the novel concept of tunable survivability within the framework of broadcasting through spanning trees while considering additional QoS requirements.

Specifically, tunable survivability within this framework enables the establishment of spanning trees that can survive network failures with any desired probability. Given a connection that consists of several spanning trees under the single failure model, only a failure on a link that is common to *all* trees can disrupt the connection. Accordingly, we characterize the survivability level of a connection as the probability to have all common links operational during the connection's lifetime. Moreover, we aim to guarantee a certain level of bandwidth requirement for this connection. The bandwidth of a connection is a bottleneck metric, given by the worst-case (minimal) bandwidth in any of the links of the spanning trees in the connection.

The following example demonstrates the concept of tunable survivability within the framework of spanning trees, while also considering bandwidth requirements. Consider the network example depicted in Fig. 1, where each link is associated with a failure probability p_e and a bandwidth b_e. Assume that we need to broadcast a reliable message to all nodes in the network through a spanning tree. The network contains several possible spanning trees, namely $T_1 = \{(a,b),(a,c),(a,d),(d,e)\}$, $T_2 = \{(a,e),(b,c),(c,d),(b,e)\}$, $T_3 = \{(a,b),(b,c),(c,d),(b,e)\}$ and $T_4 = \{(a,c),(b,c),(c,d),(d,e)\}$, illustrated by the four sub-figures bellow the network example, respectively. Clearly, the pair of link-disjoint spanning trees T_1 and T_2 guarantees full protection against a single link failure. Indeed, even if a single link fails, at least one of the trees does not contain any faulty links. Yet, this pair sustains just a single unit of bandwidth, since this is the minimal bandwidth over the links of both trees. However, if we are

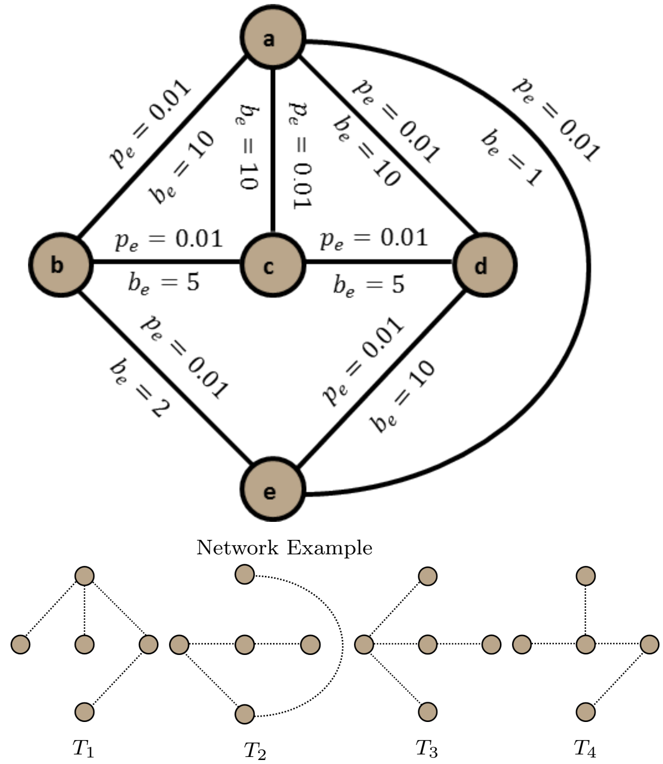


Figure 1: Example of tunable survivable spanning trees

satisfied with a survivability level of 0.99 against single network failures, then the pair of spanning trees T_1 and T_3 is a valid solution, since the only (single) failure that can concurrently damage both trees is in the common link (a,b). As this link fails with a probability of 0.01, the survivability level is 0.99 and the sustained bandwidth grows to 2. Now, suppose that we are satisfied with a survivability level of 0.99^2. Clearly, the spanning trees T_1 and T_4 also constitute a valid solution, for which the sustained bandwidth grows to 5. Finally, assume that we are satisfied with a survivability level of 0.99^4. Now, the single spanning tree T_1 also becomes a valid solution, thus increasing the sustained bandwidth to 10. We can see that there is a clear tradeoff between the survivability level and the bandwidth of the connection. We note that the above cases focused on pairs of spanning trees, while in general we can construct more than a pair. For example, by employing the triplet of spanning trees that consists of the links $T_1 = \{(a,b),(a,c),(a,d),(d,e)\}$, $T_3 = \{(a,b),(b,c),(c,d),(b,e)\}$ and $T_4 = \{(a,c),(b,c),(c,d),(d,e)\}$, we provide a full protection against a single link failure sustaining a bandwidth of 2. Therefore, this scheme improves the survivability level of 0.99 of connection supplied by the pair of spanning trees T_1 and T_3. In general, employing more spanning trees increases the survivability level but it also imposes higher management overhead.

Motivated by [19] and [27], we investigate how to implement the tunable survivability concept in the spanning tree framework. To that end, in Section 2, we formulate an optimization problem that considers two requirements, namely the level of survivability and some bottleneck QoS guarantee. Consequently, in Section 3, we establish some fundamental properties of the problem by investigating bounds on the number of trees that is needed to obtain a maximum level of survivability. Then, in Section 4, we design and validate an efficient algorithmic scheme for solving the considered optimization problem. Specifically, we present a novel polynomial-time algorithm for providing an optimal set of any number k of spanning trees that maximizes its survivability level while guaranteeing a required bandwidth. Additionally, in Section 5, we study an alternative optimization problem that considers additive, rather than bottleneck, QoS requirements. Although this is a much more complex class of problems, we present an algorithmic solution that is efficient for a case of practical interest. While the upper bound of Section 3 can be theoretically as large as the number of nodes in the network, in Section 6, through comprehensive simulations, we demonstrate that, in practice, this number is usually much smaller (in the range of $2-4$ trees). Furthermore, we show that, typically, a modest relaxation (of less than a percent) in the survivability level is enough to provide a dramatic improvement in terms of the sustained QoS. In Section 7, we provide some observations in order to extend our model for coping with double failures. Finally, Section 8 summarizes our results and discusses directions for future research.

2. PROBLEM FORMULATION

A *network* is represented by an undirected graph $G(V,E)$, where V is the set of nodes and E is the set of links. Each link $e \in E$ is assigned with a failure probability value $p_e \in (0,1)$[1]. Specifically, p_e represents the probability for a permanent fault in a specific link; we note that these probabilities are often estimated out of the available failure statistics of each network component [28] [29] [30] [31]. We assume that each link $e \in E$ fails independently. Each link $e \in E$ is also associated with a value b_e, which corresponds to some bottleneck QoS metric, e.g. the available bandwidth in the link.

A *tree* is an undirected graph in which any two vertices are connected by exactly one simple path. Given a network $G(V,E)$, a *spanning tree* $T(\tilde{V},\tilde{E})$ of $G(V,E)$ is a tree composed of all network nodes V and some of the links in E, i.e., $\tilde{V}=V$ and $\tilde{E} \subseteq E$. Furthermore, we will also use the abbreviation T for specifying the spanning tree $T(\tilde{V},\tilde{E})$ as well as its link set $\tilde{E}$.

As mentioned, 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 spanning tree T is *operational* if it has no faulty links, i.e. for each $e \in T$, link e is operational; otherwise, the spanning tree is faulty.

We proceed to formulate the concept of survivable spanning trees, through the following definitions.

Definition 2.1. *Given a network $G(V,E)$ and an integer $k>0$, a k-*survivable spanning connection *is a tuple of k spanning trees $(T_1,T_2,...,T_k)$ of $G(V,E)$.*[2]

Survivability is defined as the capability of the network to maintain service continuity in the presence of failures. Accordingly, we say that a k-survivable spanning connection $(T_1,T_2,...,T_k)$ is operational if one of its k spanning trees are operational. Note that the spanning trees of a k-survivable spanning connection are not necessarily disjoint and might contain common links. Under the single link failure model, a k-survivable spanning connection $(T_1,T_2,...,T_k)$ is operational iff the links that are common to *all* k spanning trees are operational. Consequently, as the failure probabilities are independent, we quantify the level of survivability of survivable connections as follows.

Definition 2.2. *Given a k-survivable spanning connection $(T_1,...,T_k)$, its* survivability level *$S((T_1,...,T_k))$ is defined as the probability that at least one of the spanning trees in the connection is operational, i.e. all links that are common to all spanning trees are operational, namely $S((T_1,...,T_k))=\prod_{e\in\bigcap_{i=1}^{k}T_i}(1-p_e)$.*

The above definition formalizes the notion of tunable survivability for the single link failure model. In case that there are no links that appear in all spanning trees $T_1,...,T_k$, i.e. $\bigcap_{i=1}^{k}T_i=\emptyset$, there is no single failure that can make $(T_1,...,T_k)$ fail; for this case, $S((T_1,...,T_k))$ is defined to be 1.

We proceed to quantify the bandwidth of a survivable spanning connection.

Definition 2.3. *Given a network $G(V,E)$ and a* spanning tree *T, its* bandwidth *$B(T)$ is defined as the bandwidth of its bottleneck link, i.e. $B(T)=\min_{e\in T}\{b_e\}$.*

Definition 2.4. *Given a network $G(V,E)$ and a k-survivable spanning connection $(T_1,...,T_k)$, its* bandwidth *$B((T_1,...,T_k))$ is defined as the bandwidth of the bottleneck spanning tree, i.e. $B((T_1,...,T_k)) = \min\{B(T_1),...,B(T_k)\} = \min_{e\in\bigcup_{i=1}^{k}T_i}\{b_e\}$.*

As mentioned earlier, k spanning trees that do not share common links might not necessarily exist. Accordingly, the survivability level that can be obtained by any k spanning trees might be smaller than 1. For a given network $G(V,E)$, there might be several survivable spanning connections, among them we would be interested in those that have the best "quality", giving rise to the following optimization problems.

Definition 2.5. **Constrained Bandwidth Max-Survivability (CBMS) Problem***: Given are a network $G(V,E)$, a specified integer $k>0$ of allowable spanning trees and a bandwidth constraint $B_0>0$. Find a k-survivable spanning connection $(T_1,...,T_k)$ such that:*

$$\max S((T_1,...,T_k))$$

$$s.t.\ B((T_1,...,T_k)) \geq B_0.$$

Definition 2.6. **Constrained Survivability Max-Bandwidth (CSMB) Problem***: Given are a network $G(V,E)$, a specified integer $k>0$ of allowable spanning trees and a lower bound on the survivability level $S_0\in[0,1]$. Find a k-survivable spanning connection $(T_1,...,T_k)$ such that:*

$$\max B((T_1,...,T_k))$$

$$s.t.\ S((T_1,...,T_k)) \geq S_0.$$

[1]For simplicity, we assume $p_e \neq 0,1$, however, the extension for considering these values is trivial.

[2]The spanning trees are not necessarily disjoint.

In the following sections, we will study the above problems and establish efficient algorithmic solutions for them. We begin by investigating the effect of the number k of allowable spanning trees on the quality of the solution.

3. HOW MANY SPANNING TREES?

While, in general, employing more spanning trees increases the survivability level, it also imposes higher management overhead. Moreover, it is clear that, beyond a certain number, the utilization of additional spanning trees would not further improve the obtained survivability level. Thus, an interesting question is how many trees are needed in order to obtain the maximum level of survivability. Accordingly, in this section we establish bounds on this number.

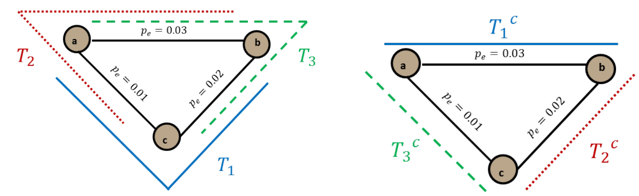
We proceed to demonstrate, through a simple example, the potential effect of the number of employed spanning trees, k, on the survivability level. Fig. 2 depicts a network where the link failure probability p_e is shown next to each link. The pairs of lines next to the network in Fig. 2a represent the three possible spanning trees of this specific network, namely $T_1 = \{(a,c),(b,c)\}$, $T_2 = \{(a,c),(a,b)\}$ and $T_3 = \{(a,b),(b,c)\}$. In case $k = 1$, the highest survivability level is $S((T_1)) = 0.99 \cdot 0.98$ accomplished by the 1-survivable spanning connection (T_1). In case $k = 2$, since this network instance does not contain two fully-disjoint spanning trees, the survivability level of any pair of spanning trees is lower than 1. Specifically, the 2-survivable spanning connection (T_1, T_2) provides the highest survivability level of $S((T_1, T_2)) = 0.99$. However, for $k = 3$, the three illustrated spanning trees, i.e. the 3-survivable spanning connection (T_1, T_2, T_3), provide a survivability level of $S((T_1, T_2, T_3)) = 1$, since upon the failure of any single link, at least one of these trees remains operational. This is because there is not any link that is common to the three trees, i.e. $\bigcap_{i=1}^{k} T_i = \emptyset$.

We note that a given bandwidth constraint B_0 can be translated to a transformed $\tilde{G}(\tilde{V}, \tilde{E})$ containing only the links that accomplish the requirement B_0, i.e $\tilde{E} = \{e | b_e > B_0\}$. Accordingly, the bandwidth consideration is omitted from the following discussion. We continue with several definitions.

Definition 3.1. *Given a network $G(V,E)$ and an integer $k > 0$, the* k-optimum survivability level $OPT^k(G)$ *is the maximum value of the survivability level that can be obtained by any k-survivable spanning connection $(T_1, ..., T_k)$, i.e. $OPT^k(G) = \max_{(T_1,...,T_k)} S((T_1, ..., T_k))$.*

Clearly, the value of the *k-optimum survivability level*, $OPT^k(G)$, depends on the number of spanning trees, k. Specifically, $OPT^k(G)$ is a non-decreasing function of k. The minimum value of this function is accomplished by $k = 1$, namely $OPT^1(G)$, and its maximum value is considered by the following definition.

Definition 3.2. *Given a network $G(V,E)$, the* maximum level of survivability $OPT(G)$ *is the maximal survivability level that can be obtained by a spanning connection with any number k of spanning trees, where*



(a) Three possible spanning trees in a cycle of three nodes (b) The complementary sets of links for the three spanning trees

Figure 2: Three nodes cycle example

$k \in [1, \binom{|E|}{|V|-1}]$,[3] *i.e., $OPT(G) = \max_k (OPT^k(G)) = \max_k \left(\max_{(T_1,...,T_k)} S((T_1, ..., T_k)) \right)$.*

Definition 3.3. *Given an undirected network $G(V,E)$, its* number of sufficient spanning trees $\phi(G)$ *is the minimum number of spanning trees required to obtain the maximum level of survivability $OPT(G)$, i.e. $\phi(G) = \min(\{k | OPT^k(G) = OPT(G)\})$.*

Based on Def. 2.2, in order to maximize the survivability level of a k-survivable spanning connection, we aim to find k spanning trees such that the probability of a failure in at least one of the links that appears in all spanning trees is minimized. Equivalently, we can also consider the set of complementary links, i.e., for a set of links $\tilde{E}$, we have $(\tilde{E}^c) = E \setminus \tilde{E}$. For a spanning connection $(T_1, ..., T_k)$ in a network $G(V,E)$, its survivability level is represented by $S((T_1, ..., T_k)) = \prod_{e \in \bigcap_{i=1}^{k} T_i} (1 - p_e) = \prod_{e \in (\bigcup_{i=1}^{k} (T_i^c))^c} (1 - p_e)$, considering De-Morgan's Law [33]. Informally, given a spanning tree T, its complementary set T^c represents the links that are protected by T. Therefore, we would like to have more links in $(\bigcup_{i=1}^{k} (T_i^c))$, and, specifically, by satisfying the property that each link appears in at least one complement set, i.e. $\bigcup_{i=1}^{k} (T_i^c) = E$, a survivability level of 1 is guaranteed.

Fig. 2b presents the complementary sets of links for the three spanning trees from Fig. 2a. Since $T_1^c = \{(a,b)\}$, $T_2^c = \{(c,b)\}$ and $T_3^c = \{(a,c)\}$ then $\bigcup_{i=1}^{k} (T_i^c) = \{(a,b),(a,c),(c,b)\} = E$ and the survivability level of (T_1, T_2, T_3) satisfies $S((T_1, T_2, T_3)) = 1$.

In some graphs, one or more of the links must be a part of any spanning tree. We call such a link *a bridge*. If a graph contains a bridge, assuming that $p_e > 0$, a survivability level of 1 cannot be achieved.

Definition 3.4. *Given an undirected network $G(V,E)$, a* bridge *is a link e whose deletion increases the number of connected components. Accordingly, we define $Bridge(G)$ as the set of all bridges in the network.*

By the last definition, a bridge is a link that must be a part of any spanning tree while for any other link there exists at least one spanning tree that does not include it. The following theorem shows the dependency of the maximal survivability level of G on the failure probabilities of the set of bridges $Bridge(G)$.

[3] A trivial upper bound for the number of different spanning trees is $\binom{|E|}{|V|-1}$. An exact number is given by Kirchhoff's matrix tree theorem [32].

Theorem 3.1. *The maximum level of survivability of* $G(V, E)$ *satisfies* $OPT(G) = \prod_{e \in Bridge(G)} (1 - p_e)$.

Proof. Let k be the number of all possible spanning trees and let $(T_1, ..., T_k)$ be a spanning connection with these k spanning trees. By Def. 3.4, we have that $Bridge(G) = \bigcap_{i=1}^{k} T_i$. Since $OPT^k(G)$ is a non-decreasing function of k then $OPT(G)$ is obtained by this specific spanning connection, i.e. $OPT(G) = S((T_1, ..., T_k))$. Then, by Def. 2.2 we have $OPT(G) = S((T_1, ..., T_k)) = \prod_{e \in \bigcap_{i=1}^{k} T_i} (1 - p_e) = \prod_{e \in Bridge(G)} (1 - p_e)$. If $Bridge(G) = \emptyset$, a survivability level of 1 is obtained. $\square$

The next theorem provides, for a network G, a simple upper bound on the number of sufficient spanning trees $\phi(G)$ required to obtain $OPT(G)$.

Theorem 3.2. *For a network* $G(V, E)$*, the number of sufficient spanning trees satisfies* $\phi(G) \leq |V|$.

Proof. For a network $G(V, E)$, consider a spanning connection $(T_1, ..., T_k)$ with the minimal possible number of spanning trees $k = \phi(G)$ required to obtain $OPT(G)$. Its survivability level is $S((T_1, ..., T_k)) = \prod_{e \in \bigcap_{i=1}^{k} T_i} (1 - p_e) = \prod_{e \in (\bigcup_{i=1}^{k} (T_i^c))^c} (1 - p_e)$. Consider the first spanning tree T_1. Clearly, $|T_1| = |E \setminus T_1^c| = |V| - 1$ and $|T_1^c| = |E| - |V| + 1$. Each of the $(k-1)$ additional spanning trees $(T_2, ..., T_k)$ must contribute at least one link to the union $(\bigcup_{i=1}^{k} (T_i^c)) \subseteq E$. Otherwise, spanning trees can be eliminated from the spanning connection, contradicting its minimality. We then have $(k - 1) \leq |E \setminus T_1^c| = |V| - 1$ and $\phi(G) = k \leq |V|$. $\square$

Based on the above observations, we proceed to bound the number of sufficient spanning trees, $\phi(G)$, also from below.

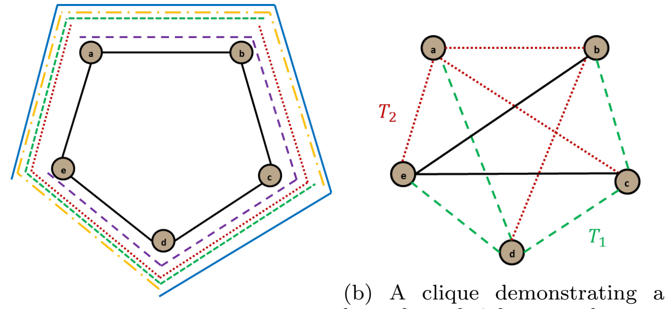
Theorem 3.3. *Given a network* $G(V, E)$*, let* $\tilde{E} \subseteq E$ *be the set of links* $\tilde{E} = E \setminus Bridge(G)$*. Then, the number of sufficient spanning trees of* G *satisfies* $\phi(G) \geq \left\lceil \frac{|\tilde{E}|}{|E| - |V| + 1} \right\rceil$.

Proof. Consider a spanning connection $(T_1, ..., T_k)$ with $k = \phi(G)$ spanning trees that satisfies $S((T_1, ..., T_k)) = OPT(G)$. We showed earlier that $S((T_1, ..., T_k)) = \prod_{e \in (\bigcup_{i=1}^{k} (T_i^c))^c} (1 - p_e)$ and by Theorem 3.1 we have $OPT(G) = \prod_{e \in Bridge(G)} (1 - p_e)$. Then necessarily $\tilde{E} \subseteq \bigcup_{i=1}^{k} (T_i^c)$, i.e. any non-bridge link must be a member of one of the set complements $T_1^c, ..., T_k^c$. The number of links in each spanning tree is fixed and for $i \in [1, k]$ $|T_i| = (|V| - 1)$, $|T_i^c| = |E| - |V| + 1$. Then, $|\tilde{E}| \leq |\bigcup_{i=1}^{k} (T_i^c)| \leq \sum_{i=1}^{k} (|T_i^c|) = k \cdot (|E| - |V| + 1) = \phi(G) \cdot (|E| - |V| + 1)$. In addition, by definition $\phi(G)$ is an integer. The result then follows. $\square$

We proceed to demonstrate the tightness of both previous presented bounds through two network examples. First, we provide a tight upper bound example considering the special case of a *cycle*, i.e. an undirected connected network $G(V, E)$ where every node has exactly two links incident with it.

Theorem 3.4. *Let* $G(V, E)$ *be a cycle of* $|V|$ *nodes. Then:*

1. *Its maximum level of survivability satisfies* $OPT(G) = 1$.



(a) A cycle demonstrating an upper bound tight example

(b) A clique demonstrating a lower bound tight example

Figure 3: Tightness examples

2. *Its number of sufficient spanning trees satisfies* $\phi(G) = |V|$.

Proof. Clearly, for a cycle $G(V, E)$, we have $|E| = |V|$ and $Bridge(G) = \emptyset$. Then, by Theorem 3.1, we have $OPT(G) = \prod_{e \in Bridge(G)} (1 - p_e) = 1$. In addition, by Theorem 3.2. $\phi(G) \leq |V|$. Likewise, by Theorem 3.3, for $\tilde{E} = E \setminus Bridge(G) = E$, we have $\phi(G) \geq \frac{|\tilde{E}|}{|E| - |V| + 1} = |E| = |V|$ and the result follows. $\square$

Next, we demonstrate the tightness of the presented lower bound of Theorem 3.3 on the number of sufficient spanning trees by considering a *clique* with $|V| \geq 4$ nodes, i.e. an undirected connected network $G(V, E)$ where every two nodes are connected by a link.

Theorem 3.5. *Let* $G(V, E)$ *be a clique of* $|V| \geq 4$ *nodes. Then:*

1. *Its maximum level of survivability satisfies* $OPT(G) = 1$.
2. *The value of the lower bound is* $\left\lceil \frac{|\tilde{E}|}{|E| - |V| + 1} \right\rceil = 2$.
3. *Its number of sufficient spanning trees satisfies* $\phi(G) = 2 = \left\lceil \frac{|\tilde{E}|}{|E| - |V| + 1} \right\rceil$.

Proof. We start by showing that $\frac{|\tilde{E}|}{|E| - |V| + 1} > 1$ and that $\frac{|\tilde{E}|}{|E| - |V| + 1} \leq 2$. For a clique G with $|V| \geq 4$ nodes, there are no bridges and $Bridge(G) = \emptyset$, $\tilde{E} = E$ and by Theorem 3.1 $OPT(G) = 1$. Likewise, $|E| = \frac{1}{2} \cdot |V|(|V| - 1)$. For $|V| \geq 4$, we have $|V| > 1$ and $|E| > |E| - |V| + 1 > 0$ so the lower bound follows. For $|V| \geq 4$, we also have $2 \cdot (|V| - 1) \leq \frac{1}{2} \cdot |V|(|V| - 1) = |E|$. Thus $|E| \leq 2 \cdot (|E| - |V| + 1)$ and the upper bound follows as well and $\left\lceil \frac{|\tilde{E}|}{|E| - |V| + 1} \right\rceil = 2$.

We would now like to show that in such a clique with $|V| \geq 4$ nodes, the number of sufficient spanning trees is indeed $\phi(G) = 2$. We show that by presenting two link-disjoint spanning trees. Consider four distinct arbitrary nodes a, b, c, d. We define the trees $T_1 = \{(a, x) | x \in V \setminus \{a, d\}\} \bigcup \{(b, d)\}$ and $T_2 = \{(d, x) | x \in V \setminus \{b, d\}\} \bigcup \{(b, c)\}$. Both spanning trees are obtained by replacing one link by another in the simple spanning trees that include only all the links connected to the nodes a and d, respectively. These spanning trees have no links in common. $\square$

Fig. 3 demonstrates a particular instance of the networks presented in Theorems 3.4 and 3.5. Specifically, Fig. 3a depicts a tight upper bound consisting of a cycle of 5 nodes, where its maximum level of survivability of $OPT(G) = 1$ is achieved by the 5 dashed lined spanning trees. Contrarily, the clique of 5 nodes illustrated in Fig. 3b represents a tight lower bound, where its maximum level of survivability of $OPT(G) = 1$ is accomplished by the 2 dashed lined spanning trees.

According to theorem 3.2, for a network $G(V, E)$ we might need to employ up to $|V|$ spanning trees in order to obtain a maximum survivability level. In practice, due to the overhead incurred by managing a large number of trees, one typically seeks solutions with a much smaller number of spanning trees. Fortunately, as shall be demonstrated at Section 6, the number of required trees for achieving the maximum survivability level is, in practice, much smaller than the bound $|V|$, namely, it is typically 2 and rarely above 3.

4. ESTABLISHING SURVIVABLE SPANNING CONNECTIONS

In this section, we show how to construct a k-survivable spanning connection that solves the CBMS and CSMB optimization problems that have been defined in section 2.

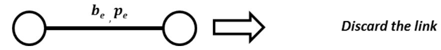
4.1 Establishing k-survivable spanning connections for the CBMS Problem

We proceed to present an efficient polynomial algorithmic scheme, termed the CBMS Algorithm, for solving the CBMS Problem. The algorithm applies, as a building block, the Minimum-Cost Edge Disjoint Spanning Tree Algorithm (EDSTA) [22] [23], which finds k link disjoint spanning trees in a graph with minimum real cost, on a constructed auxiliary graph. Indeed, EDSTA algorithm is based on the well-known matroid theory [34] [35]. Specifically, the links of a network $G(V, E)$ form a matroid by defining a set of edges $F \subset E$ to be independent iff F can be partitioned into k forests. Thus, the matroid greedy algorithm can be employed to solve the problem in polynomial time. The CBMS Algorithm, specified in Fig. 5, consists of four stages.

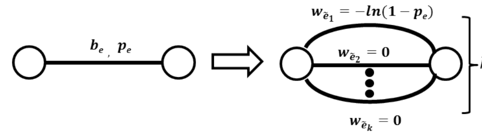
The first stage comprises of the construction of an auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$, such that $\tilde{V} = \{\tilde{u} | u \in V\}, \tilde{E} = \{\tilde{e}_1, \ldots, \tilde{e}_k | e \in E, b_e \geq B_0\}$. That is, each link in the original network that accommodates the bandwidth requirement B_0 is duplicated k times, constituting a set defined as follows.

Definition 4.1. *Given a link $e \in E$ of a network $G(V, E)$, a k-*Duplicate Link set *is the set of k-duplicated links at $\tilde{G}(\tilde{V}, \tilde{E})$ representing a single link $e \in E$. Accordingly, we denote the k-Duplicate Link set of link $e \in E$ as DL_e^k.*

The weight of one link in the k-Duplicate Link set is determined to be $w_{\tilde{e}_1} = -ln(1 - p_e)$, thus transforming the multiplicative (survivability) metric into an additive one. Note that $p_e \neq 0, 1$ by definition of Section 2, therefore $w_{\tilde{e}_1} \in (0, \infty)$. However, this trivial case can be treated at the same manner for $p_e = 0$ and by discarding the links e that $p_e = 1$. Additionally, the weights of the other $(k - 1)$ links are set to be $w_{\tilde{e}_2} = \cdots = w_{\tilde{e}_k} = 0$. Consequently, each k-Duplicate Link set contains exactly one link with a positive weight, i.e. $w_{\tilde{e}_1} > 0$. This transformation is depicted in Fig. 4.



(a) For each link with a bandwidth $b_e < B_0$



(b) For each link with a bandwidth $b_e \geq B_0$

Figure 4: CBMS Algorithm Link Transformation

Input:
$G(V, E)$-network, p_e-link failure probability, B_0-bandwidth constraint, k-the number of allowable spanning trees.
Variables:
$\tilde{G}(\tilde{V}, \tilde{E})$-transformed network, DL_e^k-a k-Duplicate Link set of link e, $w_{\tilde{e}_1}, \ldots, w_{\tilde{e}_k}$-success cost, $(\tilde{T}_1, ..., \tilde{T}_k)$-auxiliary network solution, $(T_1, ..., T_k)$-the survivable spanning connection solution.

Stage *1- Transformed network $\tilde{G}(\tilde{V}, \tilde{E})$ construction.*
- $\tilde{V} \leftarrow \{\tilde{u} | u \in V\}$.
foreach $e : (u, v) \in E$ **do**
  if $b_e \geq B_0$ **then**
    - Construct k links $\tilde{e}_1, ..., \tilde{e}_k$ between $\tilde{u}$ and $\tilde{v}$
    - Assign $w_{\tilde{e}_1}$ to be $-ln(1 - p_e)$
    - Assign $w_{\tilde{e}_2}, ..., w_{\tilde{e}_k}$ to be 0
    - $DL_e^k = (\tilde{e}_1, ..., \tilde{e}_k)$
  end
end

Stage *2- Connectivity Test.*
if *$\tilde{G}(\tilde{V}, \tilde{E})$ is not connected* **then**
  return Fail
end

Stage *3- Minimum Edge Disjoint Spanning Tree Calculation.*
- Execute the Minimum-Cost Edge Disjoint Spanning Tree Algorithm [22] for the instance $< \tilde{G}(\tilde{V}, \tilde{E}), w_{\tilde{e}}, k >$.
- Let $(\tilde{T}_1, ..., \tilde{T}_k)$ represent the solution of the Algorithm.

Stage *4- Survivable Spanning Connection Construction.*
for $i = 1$ *to* k **do**
  $T_i = \{e | (\exists \tilde{l} \in \tilde{T}_i) \wedge \tilde{l} \in DL_e^k\}$
end
return $(T_1, ..., T_k)$

Figure 5: CBMS Algorithm

In the second stage, the algorithm executes a connectivity test on the auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$, using some standard connectivity test procedure, e.g. Depth First Search (DFS) [34]. If $\tilde{G}(\tilde{V}, \tilde{E})$ is found to be disconnected, the algorithm stops with a failure indication.

Otherwise, the third stage finds out k link disjoint spanning trees of minimum cost in the auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$ by applying the EDSTA algorithm from [22] or [23]. We note that the auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$ necessarily contains k link disjoint spanning trees (see Lemma 4.1). Moreover, since the EDSTA algorithm aims to find minimum cost spanning trees, the positive weighted link at each k-Duplicate Link set will be chosen only if all the other alternative (zero-weight) links are selected.

Accordingly, in the fourth stage, we construct the sought k-survivable spanning connection according to the k link-disjoint spanning trees solution identified above. Specifically, each tree T_i in the k-survivable spanning connection in $G(V,E)$ is deduced from a tree $\tilde{T}_i$ out of k link-disjoint spanning trees solution in $\tilde{G}(\tilde{V},\tilde{E})$ by choosing the links $e \in E$ associated with the k-Duplicate Link set DL_e^k containing the links $\{\tilde{e}_1,\ldots,\tilde{e}_k\}$ in $\tilde{T}_i$. Then, the algorithm outputs them as the optimal spanning connection solution $(T_1,...,T_k)$.

We proceed to prove the correctness of the CBMS Algorithm (Fig. 5).

LEMMA 4.1. *If the auxiliary network $\tilde{G}(\tilde{V},\tilde{E})$ is connected then there are k link-disjoint spanning trees.*

PROOF. We will prove the lemma by construction. Since $\tilde{G}(\tilde{V},\tilde{E})$ is connected, there is necessarily a tree spanning the network, denoted as $\tilde{T}$. Let us construct another $k-1$ spanning trees, each containing a different link of the k-Duplicated Links sets containing the links of $\tilde{T}$. We thus have k disjoint spanning trees. □

The following theorem establishes the correctness of the CBMS Algorithm.

THEOREM 4.1. *Given are a network $G(V,E)$, a requirement $k>0$ on the number of the spanning trees and a bandwidth requirement $B_0>0$. If there exists a k-survivable spanning connection $(T_1,...,T_k)$ with a bandwidth of at least B_0, then the CBMS Algorithm returns a k-survivable spanning connection that is a solution to the CBMS Problem (Def. 2.5); otherwise, the algorithm fails.*

PROOF. The auxiliary network $\tilde{G}(\tilde{V},\tilde{E})$ constructed at Stage 1 excludes all links with a bandwidth smaller than B_0, thus guaranteeing that the bandwidth of the k-survivable spanning connection $(T_1,...,T_k)$ would be at least B_0. Lemma 4.1 establishes that the algorithm fails if and only if the auxiliary network $\tilde{G}(\tilde{V},\tilde{E})$ is not connected (Stage 2). Therefore, at Stage 3, the EDSTA algorithm [22] outputs a feasible solution $(\tilde{T}_1,...,\tilde{T}_k)$. We note that at each k-Duplicate Link set exactly one link has a positive weight $w_{\tilde{e}_1}>0$ and the rest are of zero weight $w_{\tilde{e}_2}=\cdots=w_{\tilde{e}_k}=0$. Therefore, since the EDSTA algorithm aims to find minimum cost spanning trees, only the k-Duplicate Link sets that are fully selected by all spanning trees $\tilde{T}_1,...,\tilde{T}_k$ affect the weight of the optimal solution $(\tilde{T}_1,...,\tilde{T}_k)$. By applying the indicator function $I(\cdot)$, which takes the value of 1 if the condition that it receives as a parameter is satisfied, and 0 otherwise, the weight of the optimal solution $(\tilde{T}_1,...,\tilde{T}_k)$ minimizes $\sum_{\tilde{T}_i\in(\tilde{T}_1,...,\tilde{T}_k)}\sum_{\tilde{e}_j\in\tilde{T}_i} w_{\tilde{e}_j} =$

$$\sum_{e\in E}\sum_{\tilde{T}_i\in(\tilde{T}_1,...,\tilde{T}_k)}\sum_{j\in[1,k]} I(\tilde{e}_j\in\tilde{T}_i)\cdot w_{\tilde{e}_j} =$$

$$\sum_{e\in E} -\ln(1-p_e)\cdot I((\forall i\in[1,k])(\exists j\in[1,k])\tilde{e}_j\in\tilde{T}_i) =$$

$$\sum_{e\in E} -\ln(1-p_e)\cdot I((\forall i\in[1,k])e\in T_i) = -\ln\prod_{e\in\bigcap_{i=1}^k T_i}(1-p_e).$$

Input: $G(V,E)$-network, p_e-link failure probability, S_0-survivability level constraint, k-the number of allowable spanning trees.
Variables: Q-set of available bandwidth values, $(T_1,...,T_k)$-the survivable spanning connection solution.
- Create a sorted set $Q=\{b_e|e\in E\}$.
- Perform a Binary Search over the set Q in order to find the largest $B_0\in Q$ by executing the CBMS Algorithm with bandwidth constraint of B_0, as follows.
  - Denote $(T_1,...,T_k)$ the solution of CBMS Algorithm.
  if $S((T_1,...,T_k))\geq S_0$ **then**
    increase the value of B_0 according to the binary search.
  end
  else
    reduce the value of B_0 according to the binary search.
  end
return $(T_1,...,T_k)$

Figure 6: CSMB Algorithm

Equivalently, it maximizes $\prod_{e\in\bigcap_{i=1}^k T_i}(1-p_e)$, which is the optimal solution value of the CBMS Problem. Accordingly, the fourth stage outputs the optimal solution. □

We proceed to analyze the running time of the CBMS Algorithm.

THEOREM 4.2. *The time complexity of the CBMS Algorithm is $O(k\cdot|E|\cdot\log(k\cdot|E|)+k^2\cdot|V|^2)$.*

PROOF. We note that, at Stage 1, the construction of the transformed network $\tilde{G}(\tilde{V},\tilde{E})$ contains at most $k\cdot|E|$ links. Moreover, the connectivity test performed at Stage 2 can be performed in $O(k\cdot|E|+|V|)$ [34]. According to [22] and considering the number of links in the new transformed network, the Minimum-Cost Edge Disjoint Spanning Tree Algorithm executed at Stage 3 incurs $O(k\cdot|E|\cdot\log(k\cdot|E|)+k^2\cdot|V|^2)$. Clearly, the time complexity of the CBMS Algorithm is solely affected by the running time of Stage 3. □

4.2 Establishing k-survivable spanning connections for the CSMB Problem

We proceed to establish an efficient algorithmic scheme for the second optimization problem, namely the CSMB Problem (as defined in Def. 2.6). This is achieved by employing the previously presented CBMS Algorithm (Fig. 5), as follows. Given are a network $G(V,E)$, a specified number k of allowable spanning trees and a survivability level constraint S_0. Accordingly, we aim to find the largest value of B_0 such that the survivability level of the most survivable k-survivable spanning connection with a bandwidth of at least B_0, which is a solution to the CBMS Problem, is at least S_0. Clearly, the efficiency of this strategy depends on the number of times that the CBMS Algorithm needs to be executed. We proceed to show that it is sufficient to consider just $O(\log|E|)$ such executions.

First, we observe that the bandwidth of a k-survivable spanning connection $B((T_1,...,T_k))$, per Def. 2.4, is a bottleneck metric. Therefore, for every given network $G(V,E)$

of $|E|$ links, the maximum bandwidth of a k-survivable spanning connection belongs to a set of at most $|E|$ values, i.e. $Q = \{b_e | e \in E\}$. According to Lemma 4.2, it is enough to perform a *binary search* over the set Q executing the CBMS Algorithm. Each time the CBMS Algorithm provides a solution that accomplishes the survivability constraint S_0, the bandwidth constraint B_0 that serves as input to the CBMS Algorithm is increased. Otherwise, the B_0 input is reduced. Accordingly, Fig. 6 depicts a formal specification of this algorithmic scheme, namely the *CSMB Algorithm*.

We proceed to prove the correctness of the CSMB Algorithm. Denote the survivability level value of the solution of the CBMS Problem 2.5 with a constraint of B_0 as $CBMS_{MAX}(B_0)$.

LEMMA 4.2. *The function $CBMS_{MAX}(B_0)$ is a non-increasing monotonic function in the parameter $B_0 \in Q = \{b_e | e \in E\}$, i.e for each $B_{0_1}, B_{0_2} \in Q$ such that $B_{0_1} < B_{0_2}$ then $CBMS_{MAX}(B_{0_1}) \geq CBMS_{MAX}(B_{0_2})$.*

PROOF. Given are $B_{0_1}, B_{0_2} \in Q$ such that $B_{0_1} < B_{0_2}$ and the associated solutions T_{0_1}, T_{0_2} to the CBMS Problem, respectively. Clearly, T_{0_2} is also a feasible solution for the CBMS problem with a bandwidth constraint of B_{0_1}, since $B(T_{0_2}) \geq B_{0_2} > B_{0_1}$. Therefore, $CBMS_{MAX}(B_{0_1}) \geq CBMS_{MAX}(B_{0_2})$, otherwise it contradicts its maximality. $\square$

The following theorem establishes the correctness of the CSMB Algorithm.

THEOREM 4.3. *Given are a network $G(V, E)$, a requirement $k > 0$ on the number of the spanning trees and a survivability level constraint $S_0 \in [0, 1]$. If there exists a k-survivable spanning connection $(T_1, ..., T_k)$ with a survivability level of at least S_0, then the CSMB Algorithm returns a k-survivable spanning connection that is a solution to a CSMB Problem 2.5; otherwise, the algorithm fails.*

PROOF. Lemma 4.2 establishes that the execution of the CBMS Algorithm for incrementing values of B_0 results in solutions with non-increasing survivability levels. Moreover, the set $Q = \{b_e | e \in E\}$ of available bandwidth values contains at most $|E|$ values. Therefore, in order to solve the CSMB Problem, it is enough to perform a binary search over the solution of the CBMS Algorithm for different values of $B_0 \in Q$. $\square$

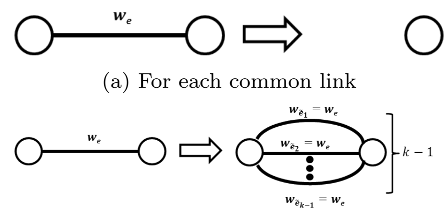
We consider now the complexity incurred by the CSMB Algorithm. To that end, we denote by $T(k, |V|, |E|)$ the running time of the CBMS Algorithm. Since the CSMB Algorithm consists of a binary search over a set of size $|E|$, performing the CBMS Algorithm at each step, its time complexity is $O(T(k, |V|, |E|) \cdot \log |E|)$.

5. ADDITIVE METRICS

We proceed to consider a variant of the model presented in Section 2, which addresses an additive QoS metric instead of the bandwidth (bottleneck) metric. Accordingly, each link $e \in E$ is associated with an additive metric value w_e, which corresponds to an additive QoS target, such as delay or cost. We proceed to quantify the weight of a survivable spanning connection.

DEFINITION 5.1. *Given a network $G(V, E)$ and a spanning tree T, its* weight *$W(T)$ is defined as the sum of the weights of its links, namely $\sum_{e \in T} w_e$. Likewise, for a k-survivable spanning connection $(T_1, ..., T_k)$, its* weight *$W((T_1, ..., T_k))$ is defined as the sum of the weights of its k trees, namely $W((T_1, ..., T_k)) = \sum_{i=1}^{k} W(T_i) = \sum_{i=1}^{k} \sum_{e \in T_i} w_e$.*

We note that the problem of finding a spanning tree in a graph with two additive costs, namely minimizing one cost under a restriction on the other cost, is NP-Hard [36]. Ac-



(a) For each common link

(b) For the rest of the links

Figure 7: MSSC Algorithm Link Transformation

Input:
$G(V, E)$-network, w_e-link weights, l-maximal number of common links.
Variables:
$\tilde{G}(\tilde{V}, \tilde{E})$-auxiliary network, E_{sub}-a subset of E, $DL_e^{(k-1)}$-a $(k-1)$-Duplicate Link set of link e, $\mathcal{P}_{\leq l}(E)$-the power set of E with cardinality of at most l, $(T_1, ..., T_k)$-the survivable spanning connection solution, W_{min}-weight of the survivable spanning connection solution (initialized to ∞), $(\tilde{T}_1, ..., \tilde{T}_k)$-auxiliary network solution, $\tilde{W}_{min}$-weight of the candidate solution.

foreach $E_{sub} \in \mathcal{P}_{\leq l}(E)$ **do**
  - Construct $\tilde{G}(\tilde{V}, \tilde{E})$ as follows:
    - $\tilde{V} \leftarrow \{\tilde{u} | u \in V\}$.
    foreach $e : (u, v) \in E_{sub}$ **do**
      - Concatenate the nodes $\tilde{u}, \tilde{v}$ to a single node in $\tilde{V}$.
    end
    foreach $e : (u, v) \in E \backslash E_{sub}$ **do**
      - Construct $(k-1)$ links $\tilde{e}_1, ..., \tilde{e}_{k-1}$ between $\tilde{u}$ and $\tilde{v}$.
      - Assign $w_{\tilde{e}_1}, ..., w_{\tilde{e}_{k-1}}$ to be w_e.
      - $DL_e^{(k-1)} = (\tilde{e}_1, ..., \tilde{e}_{k-1})$.
    end
  - Execute the Minimum-Cost Edge Disjoint Spanning Tree Algorithm [22] for the instance $< \tilde{G}(\tilde{V}, \tilde{E}), w_{\tilde{e}}, k >$.
  - Let $(\tilde{T}_1, ..., \tilde{T}_k)$ represent the solution of the Algorithm and $\tilde{W}_{min}$ its weight.
  if $W_{min} > \tilde{W}_{min} + k \cdot \Sigma_{e \in E_{sub}} w_e$ **then**
    - $W_{min} = \tilde{W}_{min} + k \cdot \Sigma_{e \in E_{sub}} w_e$
    - $\forall i = 1, ..., k$,
      $T_i = \{e | (\exists \tilde{l} \in \tilde{T}_i) \ \wedge \ \tilde{l} \in DL_e^{(k-1)}\} \cup E_{sub}$
  end
end
- return $(T_1, ..., T_k)$

Figure 8: MSSC Algorithm

cordingly, we consider the following simpler version of the optimization problem. We assume that all links have the same failure probability. In this case, by Def. 2.2 a lower bound on the survivability level is translated into an upper bound on the number of common links. We denote this number by l.

Definition 5.2. **Minimal Survivable Spanning Connection (MSSC) Problem**: *Given are a network $G(V, E)$, a requirement $k > 0$ on the number of the spanning trees and a restriction l on the maximal number of common links. Find a k-survivable spanning connection $(T_1, ..., T_k)$ with at most l common links and of minimum weight.*

The MSSC Algorithm, specified in Fig. 8, solves the MSSC optimization problem 5.2. As will be shown, its time complexity is exponential in the allowable number of common links, l. However, in many practical instances, the required level of survivability is high, and hence, the number of overlapped links needs to be small (thus inducing a small value of l), which in turn makes the algorithm computationally permissible. The algorithm considers all possible combinations of at most l common links out of the set E. Specifically, we construct an auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$ for each set of common links E_{sub} in the power set of E with cardinality of at most l, i.e. the set of all subsets of E whose size is at most l. The auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$ consists of a link transformation, depicted in Fig. 7. Initially, each link $(u, v) \in E_{sub}$ is excluded by concatenating the nodes u and v to a single node. Moreover, the rest of the links are duplicated $k-1$ times and their weights are set to be w_e. Consequently, given the new constructed network $\tilde{G}(\tilde{V}, \tilde{E})$, the algorithm identifies a k link-disjoint spanning tree with minimum cost, by employing the Minimum-Cost Edge Disjoint Spanning Tree Algorithm (EDSTA) [22] [23]. Then, we construct the sought survivable spanning connection out of the links in the link-disjoint spanning tree solution. Finally, the optimal survivable spanning connection solution $(T_1, ..., T_k)$ is the smallest survivable spanning connection discovered out of all constructed auxiliary networks (one for each possible combination of at most l common links).

The following theorem establishes the correctness of the MSSC Algorithm.

Theorem 5.1. *Given are a network $G(V, E)$, a requirement $k > 0$ on the number of the spanning trees and a restriction l on the number of common links. If there exists a k-survivable spanning connection $(T_1, ..., T_k)$ with at most l common links, then the MSSC Algorithm returns a k-survivable spanning connection that is a solution to a MSSC Problem 5.2; otherwise, the algorithm fails.*

Proof. We will prove that at each iteration of the first "foreach" loop, the MSSC Algorithm finds a minimum weight solution which its common links are E_{sub}. Since this loop examines all possible cases (over the $\mathcal{P}_{\leq l}(E)$), the algorithm outputs the optimal solution.

In order to prove the above, we will show that at the auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$ a minimum cost k-edge disjoint spanning tree $(\tilde{T}_1, ..., \tilde{T}_k)$ with a weight $\tilde{W}_{min}$ has an equivalent minimal k-survivable spanning connection solution $(T_1, ..., T_k)$ in $G(V, E)$ whose weight is $\tilde{W}_{min} + k \cdot \Sigma_{e \in E_{sub}} w_e$ and $T_i = \{e | (\exists \tilde{l} \in \tilde{T}_i) \ \wedge \ \tilde{l} \in DL_e^{(k-1)}\} \cup E_{sub}$. By applying the indicator function $I(\cdot)$, which takes the value of 1 if the condition that it receives as a parameter is satisfied, and 0 otherwise, the weight of the optimal solution $(\tilde{T}_1, ..., \tilde{T}_k)$ minimizes $\sum_{\tilde{T}_i \in (\tilde{T}_1, ..., \tilde{T}_k)} \sum_{\tilde{e}_j \in \tilde{T}_i} w_{\tilde{e}_j} + k \cdot \Sigma_{e \in E_{sub}} w_e =$

$$\sum_{e \in E} \sum_{\tilde{T}_i \in (\tilde{T}_1, ..., \tilde{T}_k)} \sum_{j \in [1, k-1]} I(\tilde{e}_j \in \tilde{T}_i) \cdot w_e + k \cdot \Sigma_{e \in E_{sub}} w_e =$$

$$\sum_{e \in E} \sum_{i \in [1, k]} I(e \in T_i) \cdot w_e = \sum_{i=1}^{k} \sum_{e \in T_i} w_e = W((T_1, ..., T_k))$$

Accordingly, the spanning connection obtained from the solution of the Minimum-Cost Edge Disjoint Spanning Tree Algorithm at the auxiliary network $\tilde{G}(\tilde{V}, \tilde{E})$ minimizes the weight of a k-survivable spanning connection solution $(T_1, ..., T_k)$ in $G(V, E)$ where its common links are the set E_{sub}. Therefore, the algorithm outputs the optimal solution. $\square$

The MSSC Algorithm time complexity is $O(\sum_{i=1}^{l} \binom{|E|}{i} \cdot ((k \cdot |E|) \cdot \log(k \cdot |E|) + k^2 \cdot |V|^2))$, i.e., exponential in l but polynomial in all other values. As noted, the value of l would be typically small, in which case the algorithm offers a time-efficient optimal solution.

6. SIMULATION STUDY

In this section, we demonstrate the advantages of employing *tunable* survivable spanning connections over the traditional full protection schemes on common network topologies. First, we examine the effect of the number of different spanning trees k in a survivable spanning connection on the accomplished survivability level. Specifically, we demonstrate that the number of required spanning trees is much smaller than the theoretical upper bound described in Theorem 3.2 and, in practice, two spanning trees are sufficient for getting extremely (namely, in the order of four-nines[4]) close to the maximum survivability level. Motivated by this finding, we focus on $k = 2$ and compare between the performance (in terms of bandwidth and feasibility) of an optimal 2-survivable spanning connection with a survivability level of at least S_0, where $S_0 \in [0.95, 1]$, and the performance of a pair of fully disjoint spanning trees (i.e., $S_0 = 1$). Most notably, we show that, by slightly relaxing the traditional requirement of 100% protection, major improvements in terms of bandwidth and feasibility are accomplished.

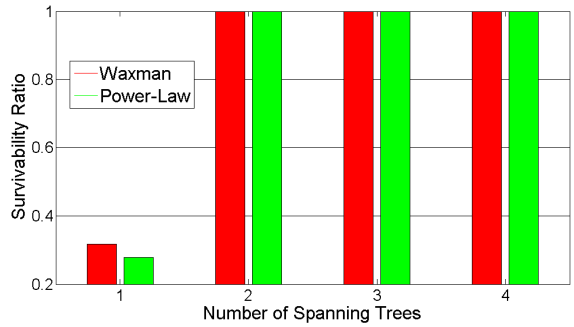
6.1 Setup

We generated two classes of well-known random network topologies, namely Power-Law topology [37] and Waxman topology [38]. First, we consider the Power-Law topology, which is widely employed for modeling typical network interconnections, in particular in the context of the Internet [37]. We then extend our findings to other classes of network topologies by analyzing the Waxman topology.

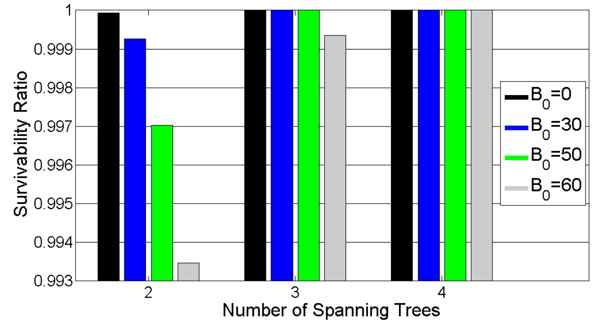
For both classes, we generated 10,000 random networks, each containing 200 nodes. In all simulation instances, we assumed a uniform distribution of the link bandwidth, in the range of $[5, 150]$ MB/s. Additionally, the failure probability of each link was distributed normally with a mean of 0.01 and a standard deviation of 0.003 assuming a non-negative value.

For each of the above generated random networks, we employed the CBMS Algorithm (Fig. 5) with several band-

[4] i.e. 0.9999.

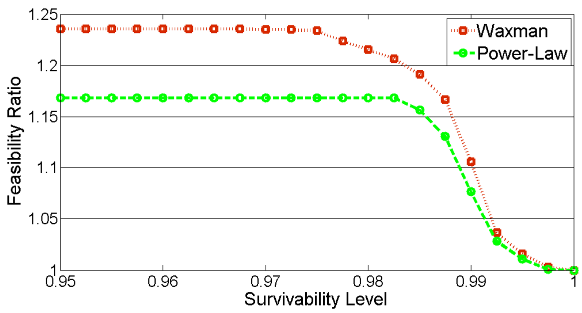


(a) Maximum survivability level ratio $OPT^k(G)/OPT(G)$ versus the number of spanning trees k

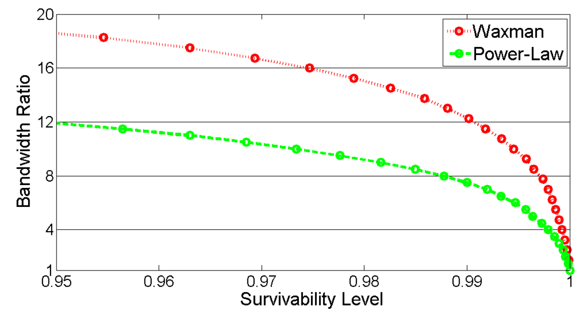


(b) Maximum survivability level ratio $OPT^k_{B_0}(G)/OPT_{B_0}(G)$ versus the number of spanning trees k for different bandwidth requirements B_0 in Waxman Topology

Figure 9: Maximum survivability as a function of the number of spanning trees



(a) Feasibility ratio $N(S_0)/N(1)$ versus the survivability level requirement S_0



(b) Bandwidth ratio $B(S_0)/B(1)$ versus the survivability level requirement S_0

Figure 10: Performance as a function of the survivability level requirement

width constraint values, namely $B_0 = 0, 30, 50, 60$, and with various values of the number of allowable spanning trees, namely $k = 1, 2, 3, 4$. For each bandwidth constraint B_0, we measured the network's maximum level of survivability denoted as $OPT_{B_0}(G)$ and the maximum survivability of the k-survivable spanning connection $OPT^k_{B_0}(G)$. Then, we derived the *survivability ratio*, defined as $\sigma_{B_0}(k) \triangleq \frac{OPT^k_{B_0}(G)}{OPT_{B_0}(G)}$ and, accordingly, calculated the average value over the simulated networks, denoted as $\overline{\sigma_{B_0}(k)}$.

Next, we consider the case of 2-survivable spanning connection through the following simulations. For each generated network and survivability level constraint S_0 in the range of $[0.95, 1]$, we employed the CSMB Algorithm (Fig. 6) and conducted the following measurements. First, we measured the number of networks that admit a 2-survivable spanning connection with a survivability level of at least S_0 (without any bandwidth requirement) among the 10,000 random networks, denoted as $N(S_0)$. Note that $N(1)$ represents the number of networks that admit fully disjoint spanning trees since $p_e > 0$. Accordingly, we derived the *feasibility ratio* defined as $\rho(S_0) \triangleq \frac{N(S_0)}{N(1)}$. Then, for each of the $N(1)$, we measured the maximum bandwidth of a 2-survivable spanning connection with a survivability level of at least S_0, denoted as $B(S_0)$. Note that $B(1)$ is the maximum bandwidth of a fully disjoint spanning connection. Thus, we derived the *bandwidth ratio*, defined as $\beta(S_0) \triangleq \frac{B(S_0)}{B(1)}$ and, accordingly, calculated the average value over the corresponding $N(1)$ networks, denoted as $\overline{\beta(S_0)}$.

We proceed to further specify the generation of the random topologies. For Power-Law topologies, following [37], we 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, $\alpha = 0.61$ 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 link (u, 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 [38]. Initially, we located the source and the destination at the diagonally opposite corners of a square of unit dimension. Then, we randomly spread 198 additional 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: $prob(u, v) = \alpha \cdot \exp \frac{-\delta(u,v)}{\beta \cdot \sqrt{2}}$ considering $\alpha = 1$ and $\beta = 0.058$. Each simulated Waxman network consists of 200 nodes, and, in average, 1200 links.

6.2 Results

The simulation results are illustrated in Fig. 9 and Fig. 10. First, the graph depicted in Fig. 9a presents the average survivability ratio $\overline{\sigma_{B_0}(k)}$ without a bandwidth restriction, i.e. $B_0 = 0$, as a function of the allowable different spanning trees $k \in [1, 4]$, for each of the two network topology classes, i.e. Power-Law and Waxman. For both classes, we observe a substantial improvement (resulting in a survivability level of almost $OPT(G)$) by utilizing two spanning trees rather than one. Furthermore, the employment of $k > 2$ spanning trees has a very marginal effect on the survivability level. Accordingly, Fig. 9b exhibits a zoomed view of the above Waxman-class results for a range of values close to the optimal solution, i.e. $\overline{\sigma_{B_0}(k)} \in [0.993, 1]$, and, additionally, considers other bandwidth restrictions, namely $B_0 = 0, 30, 50, 60$. The graph shows that, for all bandwidth restrictions, 4 spanning trees are sufficient for the maximum level of survivability and 3 spanning trees achieves a survivability ratio of 1 for the three softer bandwidth restrictions $B_0 = 0, 30, 50$. However, a 2-survivable spanning connection provides a solution that is just one percentile away from optimum. Furthermore, by alleviating the restriction B_0 on the bandwidth, there is an improvement in the efficiency of a 2-survivable spanning connection. Specifically, while for $B_0 = 60$ its survivability ratio is $\overline{\sigma_{60}(2)} = 0.9935$, for a non-restricted network $B_0 = 0$, a 2-survivable spanning connection's survivability level is in the order of four-nines from the optimum. Thus, we can conclude that, in practice, two spanning trees are enough for providing a close-to-optimal survivability level.

The graphs depicted in Fig. 10 demonstrate the advantages of employing a 2-survivable spanning connection, with not-necessarily-disjoint trees, over the traditional approach of employing fully disjoint spanning trees. Specifically, the graph depicted in Fig. 10a presents the feasibility ratio $\rho(S_0)$ as a function of the survivability level requirement $S_0 \in [0.95, 1]$, for Power-Law and Waxman classes. We observe that, with a relaxation of just 1% in the survivability level, the feasibility ratio increases by a factor of 24% for Waxman networks and 17% for Power-Law networks. Next, the graph illustrated in Fig. 10b presents the average bandwidth ratio $\overline{\beta(S_0)}$ as a function of the survivability level requirement $S_0 \in [0.95, 1]$ out of the networks that admit the establishment of a pair of fully disjoint spanning trees, i.e. accomplish a survivability level of $S_0 = 1$. We note that the number of such networks was in the range of 8,000 to 9,000 (out of 10,000), hence the samples were always significant. Overall, we observe that a minor relaxation, of a few percentiles, in the survivability level, is enough to provide dramatic improvement in terms of the sustained bandwidth. Specifically, a relaxation of about 0.5% in the survivability level provides an improvement of a factor of 12 for Waxman networks and a factor of 8 for Power-Law networks.

7. TWO FAILURES PROTECTION

We proceed to briefly discuss a generalization of the model in order to handle failures in up to two links. As in the single link failure model, we again define the survivability level of a spanning connection as the probability that at least one of the spanning trees in the connection is operational, i.e. at least one of the spanning trees does not contain any faulty links.

In order to have a survivability level of 1, we need to guarantee that we have a spanning tree that does not include any of the (up to two) faulty links. To do so, we require that, for any pair of links, there is at least one spanning tree that does not include both of the links in the faulty pair. Consider a network $G(V, E)$ with a spanning connection $(T_1, ..., T_k)$. As in the single link failure model, the spanning tree $T_i(V, E_i)$ guarantees the existence of an operational spanning tree in any single failure in one of the links $T_i^c = E \setminus E_i$. Furthermore, this spanning tree remains operational if both links in a pair of links in $T_i^c \times T_i^c$ fail. Accordingly, if the spanning connection satisfies the property $\bigcup_{i=1}^{k}(T_i^c \times T_i^c) = E \times E$, a survivability level of 1 is guaranteed.

We saw that, in the single link failure model, a spanning connection with two link-disjoint spanning trees results in a survivability level of 1. The next theorem shows that this is not the case in the generalized model.

THEOREM 7.1. *Let $G = (V, E)$ be a network for which the link failure probabilities satisfy $\forall e \in E, p_e > 0$. In the two links failure model,* any *spanning connection with $k = 2$ spanning trees $(T_1(V, E_1), T_2(V, E_2))$ has a survivability level smaller than* 1.

PROOF. Consider a pair of links $(e_1, e_2) \in E_1 \times E_2$. Since $p_{e_1}, p_{e_2} > 0$, there is a positive probability that both links would fail. In this case, both spanning trees are faulty and there is not any operational spanning tree. □

On the other hand, it is easy to see that a spanning connection with three link-disjoint spanning trees results in a survivability level of 1.

THEOREM 7.2. *In the two links failure model, a spanning connection with $k = 3$ link-disjoint spanning trees $(T_1(V, E_1), T_2(V, E_2), T_3(V, E_3))$ has a survivability level of* 1.

PROOF. Consider the worst-case in which there are two faulty links e_1, e_2. Since $E_1 \bigcap E_2 = E_1 \bigcap E_3 = E_2 \bigcap E_3 = \emptyset$, we must have each of the faulty links e_1, e_2 appear in at most one of the three spanning trees. Thus, one of the three spanning trees does not include any faulty links and is operational. □

8. CONCLUSIONS

Tunable survivability is a novel quantitative approach for coping with network failures, which can be tuned to accommodate any desired level (0%-100%) of survivability, yet till now it has been studied only in the context of unicast connections [19] [27]. This study extended the tunable survivability concept for handling broadcast through spanning trees. Specifically, we established a novel polynomial-time algorithm for providing an optimal set of (any) k spanning trees that maximizes its survivability level while ensuring a guaranteed bandwidth. Additionally, we provided tight bounds on the number of spanning trees that may be needed in order to achieve a maximum level of survivability. Finally, through comprehensive simulations, we showed that the maximum level of survivability can be well-approximated (namely, within order of four-nines) by establishing just two spanning trees. Moreover, the simulations clearly demonstrated the advantages of tunable survivability over traditional survivability schemes.

Motivated by [39], we are currently investigating the practical aspects of our findings by implementing tunable survivability schemes in Ethernet architectures, through an extension to the Multiple Spanning Tree Protocol [13] (as has been proposed in [26] for the link-disjoint case). Moreover, similarly to [40] and [41], we consider the extension of our model beyond the traditional single failure in order to cope with multiple failures extending the preliminary results presented in Section 7.

While there is still much to be done towards the actual deployment of the tunable survivability approach, we believe that this study provides evidence for the benefits of employing this concept in the scope of spanning trees and broadcast connections. Furthermore, it establishes a substantial milestone towards the construction of a comprehensive methodology.

9. ACKNOWLEDGMENTS

This research was supported by the Israel Science Foundation (grant no. 1129/10), by the Israeli Ministry of Science and Technology, and by the United States - Israel Binational Science Foundation (grant no. 2010414).

10. REFERENCES

[1] Christopher R. Cole. 100-Gb/s and beyond transceiver technologies. *Optical Fiber Technology*, 17(5):472 – 479, 2011.

[2] ITU-T. G.8032: Ethernet ring protection switching, 2008.

[3] Nurit Sprecher and Adrian Farrel. MPLS Transport Profile (MPLS-TP) Survivability Framework. RFC 6372, 2011.

[4] Greg Kuperman and Eytan Modiano. Network protection with guaranteed recovery times using recovery domains. In *IEEE Infocom*, 2013.

[5] Abdelhamid E. Eshoul and Hussein T. Mouftah. Survivability approaches using p-cycles in wdm mesh networks under static traffic. *IEEE/ACM Trans. Networking*, 17(2):671–683, 2009.

[6] Jianping Wang, Chunming Qiao, and Hongfang Yu. On progressive network recovery after a major disruption. In *IEEE Infocom*, 2011.

[7] Daniel Sadoc Menasché, Alberto Avritzer, Sindhu Suresh, Rosa M Leão, Edmundo Souza e Silva, Morganna Diniz, Kishor Trivedi, Lucia Happe, and Anne Koziolek. Assessing survivability of smart grid distribution network designs accounting for multiple failures. *Concurrency and Computation: Practice and Experience*, 2014.

[8] Martin Suchara, Dahai Xu, Robert Doverspike, David Johnson, and Jennifer Rexford. Network architecture for joint failure recovery and traffic engineering. In *ACM SIGMETRICS*, 2011.

[9] David Plummer. Ethernet Address Resolution Protocol. RFC 826, 1982.

[10] Ralph Droms. Dynamic Host Configuration Protocol. RFC 2131, 1997.

[11] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless networks*, 8(2-3):153–167, 2002.

[12] IEEE. 802.1d: Standard for local and metropolitan area networks: Media access control (mac) bridges, 1990.

[13] IEEE. 802.1s: Standard for local and metropolitan area networks: Multiple spanning trees, 2001.

[14] Jian Qiu, Mohan Gurusamy, Kee Chaing Chua, and Yong Liu. Local restoration with multiple spanning trees in metro ethernet networks. *IEEE/ACM Trans. on Networking*, 19(2):602–614, 2011.

[15] Bart Braem, Benoit Latre, Ingrid Moerman, Chris Blondia, and Piet Demeester. The wireless autonomous spanning tree protocol for multihop wireless body area networks. In *Mobile and Ubiquitous Systems*. IEEE, 2006.

[16] Junbin Liang, Jianxin Wang, Jiannong Cao, Jianer Chen, and Mingming Lu. An efficient algorithm for constructing maximum lifetime tree for data gathering without aggregation in wireless sensor networks. In *IEEE Infocom*, 2010.

[17] Hichem Baala, Olivier Flauzac, Jaafar Gaber, Marc Bui, and Tarek El-Ghazawi. A self-stabilizing distributed algorithm for spanning tree construction in wireless ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(1):97–104, 2003.

[18] Adelbert Groebbens, Didier Colle, Sophie De Maesschalck, Ilse Lievens, Mario Pickavet, Piet Demeester, Lan Tran, Kris Steenhaut, and Ann Nowé. Efficient protection in mpλs networks using backup trees. *Photonic Network Communications*, 6(3):191–206, 2003.

[19] Ron Banner and Ariel Orda. The power of tuning: A novel approach for the efficient design of survivable networks. *IEEE/ACM Trans. Networking*, 15(4):737 –749, 2007.

[20] Poul E. Heegaard and Kishor S. Trivedi. Network survivability modeling. *Computer Networks*, 53(8):1215–1234, 2009.

[21] János Tapolcai, Bin Wu, and Pin-Han Ho. On monitoring and failure localization in mesh all-optical networks. In *IEEE Infocom*, 2009.

[22] James Roskind and Robert E Tarjan. A note on finding minimum-cost edge-disjoint spanning trees. *Mathematics of Operations Research*, 10(4):701–708, 1985.

[23] Jens Clausen and Lone Aalekjær Hansen. Finding k edge-disjoint spanning trees of minimum total weight in a network: an application of matroid theory. *Combinatorial Optimization II*, pages 88–101, 1980.

[24] Crispin St. John Alvah Nash-Williams. Edge-disjoint spanning trees of finite graphs. *J. of the London Mathematical Society*, 1(1):445–450, 1961.

[25] William T. Tutte. On the problem of decomposing a graph into n connected factors. *J. of the London Mathematical Society*, 1(1):221–230, 1961.

[26] Srikant Sharma, Kartik Gopalan, Susanta Nanda, and Tzi-cker Chiueh. Viking: A multi-spanning-tree ethernet architecture for metropolitan area and cluster networks. In *IEEE Infocom*, 2004.

[27] Jose Yallouz and Ariel Orda. Tunable QoS-aware network survivability. In *IEEE Infocom*, 2013.

[28] Andrea Fumagalli and Marco Tacca. Optimal design of differentiated reliability (dir) optical ring networks. In *Int. Workshop on QoS in Multiservice IP Net.*, 2001.

[29] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *ACM Embedded networked sensor systems*, 2003.

[30] Weidong Cui, Ion Stoica, and Randy H. Katz. Backup path allocation based on a correlated link failure probability model in overlay networks. In *IEEE ICNMP*, 2002.

[31] Abhishek Chandra, Rohini Prinja, Sourabh Jain, and ZhiLi Zhang. Co-designing the failure analysis and monitoring of large-scale systems. *ACM SIGMETRICS*, 2008.

[32] Seth Chaiken and Daniel J. Kleitman. Matrix tree theorems. *Journal of combinatorial theory, Series A*, 24(3):377–381, 1978.

[33] Augustus De Morgan. *On the syllogism: and other logical writings*. Yale University Press, 1966.

[34] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. MIT Press, 3rd edition, 2009.

[35] Eugene L. Lawler. *Combinatorial optimization: networks and matroids*. DoverPublications. com, 2001.

[36] Ram Ravi and Michel X. Goemans. The constrained minimum spanning tree problem. In *Algorithm Theory*, pages 66–75. Springer, 1996.

[37] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM*, 1999.

[38] Bernard M. Waxman. Routing of multipoint connections. *IEEE JSAC*, 6(9):1617–1622, 1988.

[39] David Applegate, Lee Breslau, and Edith Cohen. Coping with network failures: Routing strategies for optimal demand oblivious restoration. In *ACM SIGMETRICS*, 2004.

[40] Matthew Johnston, Hyang-Won Lee, and Eytan Modiano. A robust optimization approach to backup network design with random failures. In *IEEE Infocom*, 2011.

[41] Kayi Lee, Hyang-Won Lee, and Eytan Modiano. Reliability in layered networks with random link