IEEE/ACM Transactions on
Networking

Novel Survivable Logical Topology Routing by Logical Protecting Spanning Trees in IP-over-WDM Networks

Zhili Zhou, *Member, IEEE,* Tachun Lin, *Member, IEEE,* Krishnaiyan Thulasiraman, *Fellow, IEEE,* and Guoliang Xue, *Fellow, IEEE*

***Abstract*—The survivable logical topology mapping (routing) problem in IP-over-WDM networks is to map each link in the logical topology (IP layer) onto a lightpath in the physical topology (optical layer) such that failure of a physical link does not cause the logical topology to become disconnected. In this paper, we propose a novel approach based on the concept of protecting spanning tree set of the logical topology. We first present necessary and sufficient conditions based on this concept and study three optimization problems with varying degrees of difficulty. We study a generalized logical routing problem with the objective to protect the logical topology against maximal number of physical link failures. The new problem aims to find a survivable routing if one exists, or achieve maximal protection of physical link failures otherwise. We also show that the problem is equivalent to the minimum dominating set problem in bipartite graphs. We discuss how one can use the column generation technique to speed up the execution of this formulation which obviates the need to find all spanning trees at the beginning of the execution of this formulation. In addition, we also present a heuristic approach which incorporates a method to augment the logical topology with additional links to guarantee a survivable routing, which only requires a shortest path algorithm and an algorithm to generate an appropriate spanning tree. We provide results of extensive simulations conducted to evaluate our formulations and demonstrate the effectiveness of our new approach.**

I. Introduction

Wavelength-Division Multiplexing (WDM) technology is widely applied in long-haul networks because of its high bandwidth and reliability. The communication between two end nodes on a WDM network is carried out through a path, namely a *lightpath*, which utilizes a single wavelength through optical nodes like optical cross-connects and optical add-drop multiplexers without opto-electro-optical conversion on intermediate optical nodes. Most data services nowadays, like HTTP, VoIP, FTP, etc., apply a dominating protocol called Internet Protocol (IP). For an IP-over-WDM network, the traffic on each IP link is carried through a lightpath in the WDM network. For a multi-hop data transmission in IP-over-WDM network as shown in Fig. 1, the traffic on the 1-2-4 path in the IP network is implemented through two lightpaths 1-2 and 2-3-4 in the WDM network.

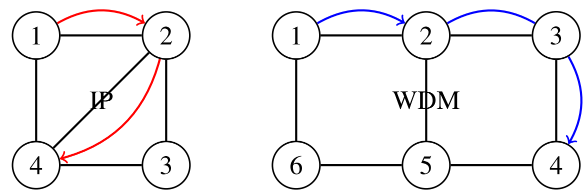


Fig. 1. An example of potential lightpaths for logical link

Given an IP-over-WDM network with physical and logical topologies $G_P = (V_P, E_P)$ and $G_L = (V_L, E_L)$, where $V_P(V_L)$ represents physical (logical) nodes/vertices and $E_P(E_L)$ represents physical (logical) edges/links, a survivable routing in such a network is usually determined by edge-disjoint lightpath routing for logical edges. If any physical link failure does not disconnect the logical topology, this routing is called a survivable routing. For this problem two lines of investigations have been pursued in the literature: the mathematical programming based approach initiated by Modiano and Narula-Tam [1], and the structural approach initiated by Kurant and Thiran [2] and pursued further by Thulasiraman et al. [3][4][5]. The mathematical programming approach is not scalable for large networks, though it gives considerable insight into certain important aspects of the problem. The structural approach requires contraction and expansion of logical graphs and computing link-disjoint lightpaths between pairs of vertices in the physical topology. This approach requires finding a set of mutually disjoint paths between the nodes of a small subset of logical links, and so considerably reduces the complexity.

In this paper, we propose a novel approach using multiple logical spanning trees and their corresponding lightpath routing to guarantee survivability. This idea was motivated by the general concept that an IP network will be survivable if there exists a logical spanning tree after any single link failure in the WDM network. Here we aim to find a set of logical spanning trees such that any physical failure will only cut off chords with respect to one or several spanning trees in the set thereby guaranteeing the existence of a logical spanning tree after a single physical link failure. This approach has several nice features. It only requires a shortest path algorithm and an algorithm to generate an appropriate spanning tree.

Zhili Zhou is with the United Airlines, Chicago, IL, USA (e-mail: zhili.zhou@united.com)

Tachun Lin is with the Department of Computer Science and Information Systems, Bradley University, Peoria, IL 61625, USA

Krishnaiyan Thulasiraman is with the School of Computer Science, University of Oklahoma, Norman, OK 73019, USA

Guoliang Xue is with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, USA

Digital Object Identifier: 10.1109/TNET.2017.2639362

Contractions of graphs and disjoint path generation are not required, which greatly reduces the computation time.

The rest of this paper is organized as follows. In Section II we review early related research works. In Section III we present several basic concepts and notations that form the basis for discussion in the rest of the paper. Given a set of spanning trees of the logical topology in Section IV we present three optimization problems with varying degrees of difficulty relating to our approach and discuss their Integer Linear Program (ILP) formulations. One of these problems is also shown to be equivalent to the minimum dominating set problem in bipartite graphs. The other one presented in Section IV-B gives a 2-stage approach to determine a subset of spanning trees of the smallest cardinality that protects the maximum number of physical links. To handle the general case when the routing and the spanning tree set are to be determined, we present in Section V a new ILP formulation. We discuss how one can use the column generation technique to speed up the execution of this formulation and also obviate the need to store all spanning trees at the beginning of the execution of this ILP. This Mixed-Integer Linear Program (MILP) along with the column generation technique is called RPTS-CGEN. For the general case we also present a heuristic approach. We incorporate in this heuristic a method to augment the logical topology with additional links to guarantee a survivable routing. Our new heuristic has several nice features. In Section VI we provide extensive simulations to evaluate the performance of our MILP formulations and our heuristic. In Section VII we conclude by summarizing our contributions.

This paper is a considerably expanded and revised extension of our earlier work [6].

II. Related Work

Survivability of a logical topology mapping (routing) can be guaranteed if the lightpaths in the physical topology corresponding to this mapping are all link-disjoint. Since finding disjoint paths between pairs of nodes is NP-complete [7], survivable design of the logical topology in an IP-over-WDM network is also an NP-complete problem. Modiano and Narula-Tam [1] proved necessary and sufficient conditions for survivable routing under a single failure in IP-over-WDM networks and formulated the problem as an ILP. Todimala and Ramamurthy [8] adapted the concept of SRLG introduced in [9] and also computed the routing through an ILP formulation. Extensions of the work in [1] are given in [10][11]. Lee et al. [10] introduced certain connectivity metrics for layered networks and provided ILP formulations for the lightpath routing problem satisfying these metrics. In particular, they provided approximation algorithms for lightpath routing maximizing the minimum cross layer cut metric which captures the robustness of the networks after multiple physical link failures. Kan et al. [11] discussed the relationship between survivable lightpath routing and spare capacity requirements on the logical links to satisfy the original traffic demands after failures. Lin et al. [12][13] introduced the concepts of weakly and strongly survivable routings and provided MILP formulations for generating a logical topology routing and rerouting (after a physical link failure) to maximize the total satisfied demand before and after a failure. They also considered the question of spare capacity minimization to maximize the demand satisfaction. Lin et al. [14] considered how lightpaths used for survivable routing can be combined with monitoring trails [15][16] to achieve localization of physical link failures. Zhou et al. [17] provided MILP formulations for cross-layer survivability under multiple metrics, including the cross-layer cut introduced by Lee and Modiano [10]. These formulations have been directly adopted in the study of the survivable network virtualization problem [18][19].

A common drawback of ILP approaches is that they are not scalable as the network size increases. Hence, heuristic approaches that provide approximations to the optimal solutions are presented. To handle the drawback of ILP approaches, Kurant and Thiran [2] proposed the Survivable Mapping by Ring Trimming (SMART) framework which first attempts to find link-disjoint paths for the links of a subgraph of the given logical graph. If such mapping exists, the subgraph is contracted. The procedure is repeated until the logical graph is contracted to a single node, or at some step disjoint mappings cannot be found. In the former case, the resulting mappings define a survivable mapping of the given logical graph. Another approach proposed by Lee et al. [20] utilized the concept of ear-decomposition on bi-connected topologies. One can show that this is, in fact, a special variant of the framework given in [2], though it was developed independently. Javed et al. [21][22] obtained improved heuristics based on SMART. Using duality theory in graphs, a generalized theory of logical topology survivability was given by Thulasiraman et al. [3][4][5]. Thulasiraman et al. [23] considered the problem of augmenting the logical graph with additional links to guarantee the existence of a survivable mapping. It has been shown in [23] that if the physical network is 3-edge connected, survivability-guaranteed augmentation of the logical topology is always possible. An earlier work that discussed augmentation is in [24].

III. Basic Concepts and Notations

We let $G_P = (V_P, E_P)$ and $G_L = (V_L, E_L)$ represent the physical and logical networks, respectively. The relationship between G_P and G_L is that $V_L \subseteq V_P$. We let i, j denote physical nodes and s, t denote the logical nodes. We let e and (i, j) denote physical edges and u, v and (s, t) denote the logical edges. Edges and links as well as nodes and vertices will be used interchangeably.

Without loss of generality, we keep the indices of logical nodes the same as their corresponding physical nodes. For a logical edge u we find a path p^u in G_P whose start and end nodes are the two corresponding nodes of u. We call p^u the **lightpath** of u. The failure of any physical edge in p^u disconnects p^u and its corresponding logical edge u. We let $i(u)$ and $j(u)$ be the physical nodes of logical edge u, and $s(e)$ and $t(e)$ be the logical nodes of physical edge e. If u connects $s(u)$ and $t(u)$, then, $(s(u), t(u)) = u$.

Definition 1: Given a logical spanning tree denoted as $\tau, \tau \in G_L$, we define τ^C as the co-tree of τ with $\tau^C = G_L \setminus \tau$.

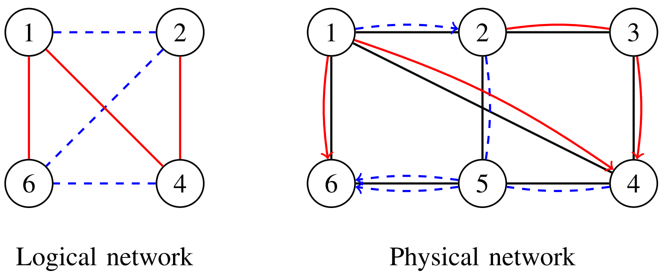


Fig. 2. The lightpath, spanning tree, and mapping example

Definition 2: Given a **logical spanning tree collection**, denoted as $\mathcal{T}$, whose elements are spanning trees in G_L. We define $\mathcal{T}^C$ as the **spanning co-tree collection** of $\mathcal{T}$ if its elements are the co-trees of the elements in $\mathcal{T}$.

Definition 3: **Mapping and co-mapping of substructures in the logical network:**

1. Mapping M maps each logical edge v to a lightpath, that is, $M : v \to p^v$ with $v \in E_L$. (Note: Here $M(v)$ also stands for the edges in the path p^v.)
2. Mapping M maps τ to a subgraph of the physical network; that is, $M : \tau \to \bigcup\limits_{v\in\tau} M(v)$.
3. Mapping M maps $\mathcal{T}$ to a subgraph of the physical network, $M : \mathcal{T} \to \bigcup\limits_{\tau\in\mathcal{T}} M(\tau)$.
4. Co-mapping M^C maps logical edge v to a subgraph of the physical network, $M^C : v \to G_P \setminus M(v)$ with $v \in E_L$.
5. Co-mapping M^C maps τ to a subgraph of the physical network, $M^C : \tau \to G_P \setminus \bigcup\limits_{v\in\tau} M(v)$; that is, $M^C(\tau) = \bigcap_{v\in\tau} M^C(v)$
6. Co-mapping M^C maps $\mathcal{T}$ to a subgraph of the physical network, $M^C : \mathcal{T} \to \bigcup\limits_{\tau\in\mathcal{T}} (G_P \setminus M(\tau))$; that is, $M^C(\mathcal{T}) = \bigcup_{\tau\in\mathcal{T}} M^C(\tau)$.

As in 1. in Definition 3, in all cases $M(\cdot)$ stands for all the edges in the corresponding subgraph.

The relationship between the mapping of logical edge, τ, and $\mathcal{T}$ is as follows:

1. $M(\tau) = \bigcup_{v\in\tau} M(v) = \bigcup_{v\in\tau} p^v$.
2. $M(\mathcal{T}) = \bigcup_{\tau\in\mathcal{T}} M(\tau) = \bigcup_{v\in\tau,\tau\in\mathcal{T}} M(v) = \bigcup_{v\in\tau,\tau\in\mathcal{T}} p^v$.

Definition 4: Given a routing of all the logical links in an IP-over-WDM network and a logical spanning tree collection $\mathcal{T}$. If physical link (i, j) is in $M^C(\tau)$, $\tau \in \mathcal{T}$, then τ is said to **protect** (i, j), and so τ is called a **protecting spanning tree** of (i, j). If for every physical link (i, j), there exists a spanning tree in $\mathcal{T}$ which protects (i, j), then the routing is a **survivable routing**, and $\mathcal{T}$ is called a **protecting spanning tree collection**.

Definition 5: Given a routing of the logical network, the set $[\mathcal{T}, M(\mathcal{T}), M^C(\mathcal{T})]$ where $\mathcal{T}$ is a protecting tree set is **survivable** if $M^C(\mathcal{T}) = E_P$.

Lemma 1: A routing of the logical topology can survive any single link failure if and only if there exists a logical protecting spanning tree set $\mathcal{T}$ such that $M^C(\mathcal{T}) = E_P$.

Proof: We first recall that if a physical link (i, j) is not used in the routing of any branch of a logical protecting spanning tree τ then $(i, j) \in M^C(\tau)$. Also, a graph is connected if and only if it contains a protecting spanning tree. Necessity: Suppose a given logical topology routing is survivable. Then, for any physical link (i, j), the logical topology contains a spanning tree τ that remains connected after the failure of (i, j). In other words, no branch of τ is routed through (i, j). So $(i, j) \in M^C(\tau)$. Let $\mathcal{T}$ be the collection of all spanning trees of the logical topology such that for each (i, j), a spanning tree $\tau \in \mathcal{T}$ remains connected after the failure of (i, j). Then $M^C(\mathcal{T}) = \bigcup_{\tau\in\mathcal{T}} M^C(\tau) = E_P$. So $\mathcal{T}$ is the required spanning tree set for the given survivable routing.
Sufficiency: Given a logical topology routing. Suppose there exists a logical spanning tree set $\mathcal{T}$ such that $M^C(\mathcal{T}) = E_P$. Then for each (i, j) there exists at least one spanning tree $\tau \in \mathcal{T}$ such that $(i, j) \in M^C(\tau)$. This means that none of the branches of τ are routed through (i, j). So when (i, j) fails, the tree τ remains connected and so the logical topology remains connected. Since this is true for each physical link (i, j), it follows that the given logical topology remains connected after any physical link failure and the routing is survivable. ■

■

Example 1: We illustrate the above definitions using Fig. 2. $\tau_1 = \{(1, 2), (2, 6), (6, 4)\}$ and $\tau_2 = \{(1, 6), (1, 4), (4, 2)\}$ are two protecting spanning trees in G_L. The lightpaths are $p^{12} = \{(1, 2)\}$, $p^{26} = \{(2, 5), (5, 6)\}$, $p^{46} = \{(4, 5), (5, 6)\}$, $p^{16} = \{(1, 6)\}$, $p^{14} = \{(1, 4)\}$, and $p^{24} = \{(2, 3), (3, 4)\}$. The mapping and co-mapping of logical edge $(2, 6)$ are $p^{26} = \{(2, 5), (5, 6)\}$ and $\{(1, 2), (1, 4), (1, 6), (2, 3), (3, 4), (4, 5)\}$, respectively. The mapping of τ_1 is $M(\tau_1) = \{p^{12}, p^{26}, p^{64}\} = \{(1, 2), (2, 5), (4, 5), (5, 6)\}$, and its co-mapping is $M^C(\tau_1) = \{(1, 4), (1, 6), (2, 3), (3, 4)\}$. The mapping of τ_2 is $M(\tau_2) = \{p^{16}, p^{14}, p^{24}\}$, and its co-mapping is $M^C(\tau_2) = \{(1, 2), (2, 5), (4, 5), (5, 6)\}$. Based on the definitions above, $\mathcal{T} = \{\tau_1, \tau_2\}$ is a protecting spanning tree collection and the routing of the lightpaths is survivable because $\mathcal{T}$'s co-mapping $M^C(\mathcal{T}) = E_P$.

If a survivable routing cannot be found, we still aim to find a routing which protects a maximal number of physical link failures. Thus, we introduce the following definition.

Definition 6: A given logical topology routing is **partially survivable** if there exists a $\mathcal{T}$, where $M^C(\mathcal{T}) \neq E_P$ and $M^C(\mathcal{T}) \neq \emptyset$.

An example of partial survivability in a given IP-over-WDM network is given as follows.

Example 2: If the spanning tree collection in Example 1 is changed to $\mathcal{T} = \{\tau_1\}$, then any failure in physical edges $(1, 4), (1, 6), (2, 3)$, and $(3, 4)$ does not disconnect τ_1, but a failure of the other physical edges disconnects τ_1. Therefore, $[\mathcal{T}, M(\mathcal{T}), M^C(\mathcal{T})]$ is partially survivable with $M(\mathcal{T}) = \{(1, 2), (2, 5), (4, 5), (5, 6)\}$ and $M^C(\mathcal{T}) = \{(1, 4), (1, 6), (2, 3), (3, 4)\}$.

IV. Protecting Spanning Tree Set Optimization and ILP Formulations

In this section we present three optimization problems with different levels of difficulty relating to protecting tree set selection for survivability. For each problem we present and discuss an (M)ILP formulation. For all these optimization problems we assume that a collection $\mathcal{T}$ of spanning trees of the logical graph is given. We consider the general case without this assumption in Section V.

Basically we want a routing and a spanning tree set (a subset of spanning trees) such that at least one tree in the set remains connected after a physical link failure. If such a tree set and a routing do not exist then the given logical topology does not admit a survivable routing. Ideally, the set of all spanning trees of the logical topology is to be given as input to the ILP. But this set is exponentially large. To handle this complexity, we limit the size of the set of trees to be given as input. The larger the set, the more likely will be the chance of getting a survivable routing for the given set of trees.

Given $G_P = (V_P, E_P)$, $G_L = (V_L, E_L)$, and a logical spanning tree collection $\mathcal{T}$. Our first approach, the MPTS formulation in Section IV-A, aims to find a survivable routing with the minimum number of spanning trees selected from $\mathcal{T}$. The MPTS is with survivability constraints (2) – (5) based on Lemma 1, which requires an additional assumption that a survivable routing does exist with the given $\mathcal{T}$; otherwise, the routing would not be generated and none of the trees in $\mathcal{T}$ would be selected.

We further explore in Section IV-B the generalized solution approach which is capable of providing a survivable routing when it exists, or otherwise protects a maximal number of physical links with a minimal number of spanning trees in $\mathcal{T}$. We solve this generalized problem and demonstrate that it can be solved to optimality through a decomposed two-stage approach.
Stage I: we determine the largest number of physical links that can be protected using the given set of trees. Let the cardinality of this physical link set be Λ; and
Stage II: we determine the smallest subset of the given trees that protects these Λ physical links.
Section IV-C shows the relationship between the MPTS and the minimum dominating set problem, and Section IV-D provides the properties of the optimal protecting spanning tree set. We wish to note that the overall goal of Section IV is to find a survivable routing utilizing the protecting spanning tree set, or achieve partial survivability against a maximal number of physical link failures.

The variables used in the formulations in Section IV are as follows:

- x_{ij}^{ℓ}: binary variable, $x_{ij}^{\ell} = 1$ if ℓ protects G_L after the failure of (i,j)
- y_ℓ: binary variable, $y_\ell = 1$ if spanning tree ℓ is selected
- z_{ij}^{st}: binary variable, $z_{ij}^{st} = 1$ if (s,t) is routed through (i,j)
- f_{st}: binary variable, $f_{st} = 1$ if (s,t) is a branch in a selected spanning tree and has a routing
- β_{ij}^{ℓ}: binary variable, $\beta_{ij}^{\ell} = 1$ indicates that (i,j) is protected by a spanning tree ℓ
- g_{ij}: binary variable, $g_{ij} = 1$ indicates the failure of physical link (i,j) disconnects G_L

A. Minimum Protecting Spanning Tree Set Problem (MPTS)

Given a collection $\mathcal{T}$ of logical spanning trees, and assuming that there exists a survivable routing under which all the physical links are protected by the trees in $\mathcal{T}$, the *Minimum Protecting Spanning Tree Set* (**MPTS**) problem is to determine a routing of all the logical links that minimizes the cardinality of the subset of $\mathcal{T}$ that protects all the physical links. The following is an ILP formulation of the MPTS problem.

$$\textbf{(MPTS)}\ \min_{y} \sum_{\ell \in \mathcal{T}} y_\ell$$

$$s.t. \quad \sum_{(i,j)\in E_P} z_{ij}^{st} - \sum_{(j,i)\in E_P} z_{ji}^{st} = \begin{cases} 1, & \text{if } s = i, (s,t) \in E_L \\ -1, & \text{if } t = i, (s,t) \in E_L \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$z_{ij}^{st} + z_{ji}^{st} \le 1 - x_{ij}^{\ell}, \quad (s,t) \in \ell, \ell \in \mathcal{T}, (i,j) \in E_P \tag{2}$$

$$\beta_{ij}^{\ell} \le x_{ij}^{\ell} + x_{ji}^{\ell}, \quad \ell \in \mathcal{T}, (i,j) \in E_P \tag{3}$$

$$\beta_{ij}^{\ell} \le y_\ell, \quad \ell \in \mathcal{T}, (i,j) \in E_P \tag{4}$$

$$\sum_{\ell \in \mathcal{T}} \beta_{ij}^{\ell} \ge 1, \quad (i,j) \in E_P. \tag{5}$$

$$y_\ell, z_{ij}^{st}, \beta_{ij}^{\ell}, f_{st} \in \{0,1\},\ \ell \in \mathcal{T}, (s,t) \in E_L, (i,j) \in E_P \tag{6}$$

We now discuss each constraint in the ILP for the MPTS problem.

Route (Lightpath) Selection: Constraint (1) selects a lightpath for each logical link (s,t) using the flow conservation principle. $z_{ij}^{st} = 1$ if (s,t) is routed through physical link (i,j); otherwise, $z_{ij}^{st} = 0$. Constraint (2) relates z_{ij}^{st} and x_{ij}^{ℓ} where $x_{ij}^{\ell} = 1$ if tree ℓ protects physical link (i,j). If a logical link $(s,t) \in \ell$ is routed through (i,j) (that is, $z_{ij}^{st} + z_{ji}^{st} = 1$), then constraint (2) forces $x_{ij}^{\ell} = 0$.

Protecting Property of Selected Spanning Trees: We let variable β_{ij}^{ℓ} indicate whether physical link (i,j) is protected by a selected spanning tree ℓ or not. The value of β_{ij}^{ℓ} is bounded by one of the following.

i) If $x_{ij}^{\ell} + x_{ji}^{\ell} = 1$ and $y_\ell = 1$, then, $\beta_{ij}^{\ell} = 0$ or 1.
ii) If $x_{ij}^{\ell} + x_{ji}^{\ell} = 1$ and $y_\ell = 0$, then, $\beta_{ij}^{\ell} = 0$.
iii) If $x_{ij}^{\ell} + x_{ji}^{\ell} = 0$ and $y_\ell = 1$, then, $\beta_{ij}^{\ell} = 0$.
iv) If $x_{ij}^{\ell} + x_{ji}^{\ell} = 0$ and $y_\ell = 0$, then, $\beta_{ij}^{\ell} = 0$.

The above requirements can be achieved by constraints (3) – (5).

To guarantee that each (i,j) is protected by a selected spanning tree ℓ (i.e., $y_\ell = 1$), we need

$$\sum_{\ell \in \mathcal{T}} \left(x_{ij}^{\ell} + x_{ji}^{\ell}\right) y_\ell \ge 1, \quad (i,j) \in E_P,$$

which is a non-linear inequality. To linearize this, we have

$$\begin{aligned}
&\sum_{\ell\in\mathcal{T}} \left(x_{ij}^{\ell}+x_{ji}^{\ell}\right) y_{\ell}\\
&\geq \sum_{\ell\in\mathcal{T}} \beta_{ij}^{\ell}, \quad \text{[because of constraints (3) and (4)]}\\
&= \sum_{\ell\in\mathcal{T},(x_{ij}^{\ell}+x_{ji}^{\ell})=1 \text{ and } y_{\ell}=1} \beta_{ij}^{\ell}, \quad \text{[because } \beta_{ij}^{\ell}=0,\\
&\quad \text{when } (x_{ij}^{\ell}+x_{ji}^{\ell})=0 \text{ or } y_{\ell}=0]\\
&\geq 1, \quad \text{[because of constraint (5)].}
\end{aligned}$$

Thus, constraints (3) – (5) guarantee that $\sum_{\ell\in\mathcal{T}} \left(x_{ij}^{\ell}+x_{ji}^{\ell}\right) y_{\ell} \geq 1$, for $(i,j) \in E_P$. In other words, constraints (3) – (5) provide the survivable condition under any physical link failure as they guarantee that the trees selected protect all the physical links.

We wish to note that if $\mathcal{T}$ includes all the logical spanning trees, the infeasible solution of the MPTS indicates that a survivable routing does not exist in a given IP-over-WDM network.

B. Minimum Protecting Spanning Tree Set and Maximum Link Protection Problem (MPTS-MaxLP)

Given a collection $\mathcal{T}$ of spanning trees of the logical graph, the *Minimum Protecting Tree Set and Maximum Link Protection Problem* (**MPTS-MaxLP**) is to determine a routing of all the logical links that minimizes the cardinality of the subset of $\mathcal{T}$ that protects the largest number of physical links.

Let a pair (g, y) correspond to a routing of the logical links if y is the cardinality of a subset of $\mathcal{T}$ that protects g physical links under the routing. A solution to the MPTS-MaxLP problem gives the pair $(g_{\max}, y_{\min})$ that has the property

$$g_{\max} \geq g \text{ and } y_{\min} \leq y$$

for any (g, y) for the given collection of logical spanning trees.

This problem can be solved using a 2-stage approach. In the following, $g_{ij} = 1$ if after the failure of physical link (i,j) at least one $\ell \in \mathcal{T}$ remains connected; otherwise, $g_{ij} = 0$.

Stage 1. With a given $\mathcal{T}$, determine $\max \sum_{(i,j)\in E_P} g_{ij} = \Lambda$

Stage 2. Determine a minimum subset of $\mathcal{T}$ such that $\sum_{(i,j)\in E_P} g_{ij} \geq \Lambda$.

The following are the formulations for these two stages:

Stage 1:

$$\Lambda = \max_{g} \sum_{(i,j)\in E_L} g_{ij}$$

$$\begin{aligned}
s.t.\ & \text{Constraints (1), (2), and (6)}\\
& g_{ij} \in \{0,1\}, \qquad (i,j) \in E_P \qquad (7)\\
& \sum_{\ell\in\mathcal{T}} x_{ij}^{\ell} \geq g_{ij}, \qquad (i,j) \in E_P \qquad (8)
\end{aligned}$$

Stage 2:

$$\min_{y} \sum_{\ell\in\mathcal{T}} y_{\ell}$$

$$\begin{aligned}
s.t.\ & \text{Constraints (1) – (4) and (6) – (8)}\\
& \sum_{(i,j)\in E_P} g_{ij} \geq \Lambda \qquad (9)\\
& \sum_{\ell\in\mathcal{T}} \beta_{ij}^{\ell} \geq g_{ij}, \qquad (i,j) \in E_P \qquad (10)
\end{aligned}$$

Theorem 1: This two-stage approach provides the Pareto optimal solution for MPTS-MaxLP.

Proof: The MPTS-MaxLP problem can be formulated as a bi-criteria optimization problem as follows:

$$\max_{g} \sum_{(i,j)\in E_L} g_{ij} \quad \text{and} \quad \min_{y} \sum_{\ell\in\mathcal{T}} y_{\ell}$$
$$s.t. \text{ Constraints (1) – (4), (6) – (8), and (10)}$$

The Pareto optimal [25] solution of a bi-criteria optimization problem is non-dominated by other feasible solutions. We prove by contradiction that the optimal solution (g^*, y^*) of the stages 1 and 2 formulations is non-dominated. If (g^*, y^*) is dominated, then there exists (g', y'), where $\sum_{(i,j)\in E_L} g'_{ij} \geq \sum_{(i,j)\in E_L} g^*_{ij}$ and $\sum_{\ell\in\mathcal{T}} y'_{\ell} \leq \sum_{\ell\in\mathcal{T}} y^*_{\ell}$, which contradicts the assumption that $g^* = \max_g \sum_{(i,j)\in E_L} g_{ij}$. Thus, our conclusion holds. ■

The objective (number of protecting trees) used in stage 2 is of both theoretical and practical value. It helps to determine the theoretical limit on the smallest size of the protecting tree set that can protect a given number of physical link failures and hence the average number of trees required per each logical link. It is a generalization (to the 2-layer case) of the classical combinatorial optimization problem of determining the number of trees required to cover all the edges of a graph. The generalization lies in finding the smallest subset of logical trees that maximizes the size of the protected physical link set. Its practical value lies in approximating the largest number of failures that can be protected and limiting the computational complexity. If one is interested in only the maximum number of physical links protected then the problem considered in this section can be limited to stage 1 whose objective is to maximize the number of physical links protected. After completion of both the stages we can augment the logical topology to protect the unprotected physical links. A method for augmentation is presented in Section V.

C. Minimum Dominating Set and the MPTS Problem

In the two optimization problems considered we seek a routing of the logical links that achieves certain objectives. A special case of the MPTS problem is to find a minimum protecting tree set given a collection of spanning trees $\mathcal{T}$ as well as a routing. We refer to this problem as **MPTS-S**. Note that in the MPTS-S problem, the values of z_{ij}^{st} are known and so the ILP formulation of this problem is obtained by removing constraint (1) from the ILP formulation given in Section IV-A for the MPTS problem.

$$\textbf{(MPTS-S)}\ \min \sum_{\forall \ell} y_\ell \tag{11}$$

$$s.t.\quad \text{Constraints (2) – (6)} \tag{12}$$

We can also give a graph theoretic formulation using the concept of dominating set. Given an undirected graph $G = (V, E)$, a node v is said to dominate node w if v is adjacent to w. A subset S of V is a dominating set of G if every vertex $u \in V \setminus S$ is adjacent to a vertex in S. A dominating set with minimum cardinality is called a minimum dominating set of G.

Given a logical spanning tree collection $\mathcal{T}$, a routing of the logical links, and a bipartite graph G with bipartition (X, Y). Let each node in X represent a tree in $\mathcal{T}$ and each node in Y represent a physical link. Let edge $(u, v) \in G$ if and only if the tree corresponding to $u \in X$ protects the physical link represented by the node $v \in Y$. Then it can be seen that a minimum dominating set of G is a solution for the MPTS-S problem.

D. Properties of an Optimal Protecting Spanning Tree Collection $\mathcal{T}^*$

Property 1: If a survivable logical topology routing exists, then no logical link is presented in all the spanning trees in $\mathcal{T}^*$.

Proof: Suppose a logical link u is in all spanning trees in $\mathcal{T}^*$. Then failure of any physical edge in the lightpath of u will disconnect u and hence all the spanning tree in $\mathcal{T}^*$. This contradicts the definition of an optimal protecting spanning tree collection. ■

Property 2: If a survivable logical topology routing exists, then the logical topology G_L is a subset of the corresponding optimal co-tree collection $\mathcal{T}^{C*}$.

V. Approaches for the General Case

In this section, we consider the general case when a protecting spanning tree set and a routing are to be determined, which requires to solve a large scale MILP. The column generation technique [26] can be used to solved it efficiently, but not all MILP formulations are amenable for the column generation technique. With this in view, we present in Section V-A a new formulation which is amenable for the incorporation of the column generation technique. This technique allows generating one column (in our case, a spanning tree) at a time as and when needed obviating the need for generating and storing all spanning trees before the execution of the linear program. This technique consists of two parts: a restricted master problem and a subsequent problem that is derived from the original problem. The role of the restricted problem is to find the current optimal solution and compute the dual variables associated with the current solution. Due to the exponential number of columns involved, the restricted master problem keeps a small subset of columns and generates feasible solutions within the feasible region of the original problem. The subproblem is used to test whether the current solution is optimal over all feasible solutions.

A. RPTS-CGEN: A New Formulation Integrated with Column Generation Technique

We introduce extra notations $\mathcal{P}^u$ which is the set of all possible lightpaths for logical link u and $\mathcal{P}$ which is the set of all possible lightpaths for all logical links, i.e., $\mathcal{P} = \cup_{u \in E_L} \mathcal{P}^u$. We let η_p denote the lightpath routing selection, where $\eta_p = 1$ if p is selected. We let δ_e^p represent if path p is routed through physical link e. If not, $\delta_e^p = 1$; otherwise, $\delta_e^p = 0$. We let ξ_u^ℓ represent whether a logical edge u is in a logical spanning tree ℓ or not; if yes, $\xi_u^\ell = 1$; otherwise, $\xi_u^\ell = 0$. We then reformulate the survivable IP-over-WDM routing problem with spanning trees based on the column-based variables as follows:

$$\textbf{(RPTS-1)}\ \max_{x,\eta} \sum_{e \in E_P} \sum_{\ell \in \mathcal{T}} x_e^\ell$$

$$s.t. \sum_{p \in \mathcal{P}^u} \eta_p \le 1, \qquad u \in E_L \tag{13}$$

$$\sum_{\ell \in \mathcal{T}} x_e^\ell \le 1, \qquad e \in E_P \tag{14}$$

$$\sum_{\ell \in \mathcal{T}} \xi_u^\ell x_e^\ell \le \sum_{p \in \mathcal{P}^u} \delta_e^p \eta_p, \qquad u \in E_L, e \in E_P \tag{15}$$

$$x_e^\ell, \eta_p \in \{0, 1\}, \qquad \ell \in \mathcal{T}, p \in \mathcal{P}, e \in E_P \tag{16}$$

Constraint (13) restricts that each logical link only selects a lightpath as its routing. Constraint (14) ensures that a physical link is protected by at most one selected logical spanning tree. Constraint (15) guarantees that if a logical edge u is in a logical tree which protects physical link e, then, the lightpath routing would not go through the protected physical link e. Constraint (16) provides the feasible region for all variables. The above formulation guarantees the survivability of routing in the IP-over-WDM with logical protection spanning trees.

With constraints (13) and (14), the upper bounds for variables η_p and x_e^ℓ are restricted to be one. Hence, the linear relaxation model (RPTS-R) for (RPTS-1) is as follows:

$$\textbf{(RPTS-R)}\ \max_{x,\eta} \sum_{e \in E_P} \sum_{\ell \in \mathcal{T}} x_e^\ell$$

$$s.t.\ \text{Constraints (13) – (15)}$$

$$x_e^\ell, \eta_p \in R^+, \qquad \ell \in \mathcal{T}, e \in E_P, p \in \mathcal{P} \tag{17}$$

1) Pricing: We let α_u, β_e, and γ_e^u represent the dual variables corresponding to constraints (13) – (15) and discuss the pricing for the protecting spanning tree selection variable x_e^ℓ and lightpath selection variable η_p.

a) Pricing for Lightpath Selection Variable: The pricing problems for the lightpath selection variable η_p are disjoint for each logical edge $u \in E_L$ and therefore can be solved separately. Given the dual variable (α_u, γ_e^u) for RPTS-R, the reduced cost for variable η_p, $p \in \mathcal{P}^u$ is

$$-\alpha_u + \sum_{e \in E_P} \gamma_e^u \delta_e^p. \tag{18}$$

With a given $\mathcal{P}^u$, the dual problem of RPTS-R is feasible if the reduced cost $-\alpha_u + \sum_{e \in E_P} \gamma_e^u \delta_e^p \le 0$. Rather than examining them separately, we treat all feasible routes in $\mathcal{P}^u$ implicitly by solving an optimization subproblem

$$\chi = \max_{p \in \mathcal{P}^u} \left\{ -\alpha_u + \sum_{e \in E_P} \gamma_e^u \delta_e^p \right\}. \tag{19}$$

To solve the above subproblem, we let ρ_e^p represent whether lightpath p is routed through e; if yes, $\rho_e^p = 1$, otherwise $\rho_e^p = 0$. Hence, we have $\delta_e^p + \rho_e^p = 1$ for all $p \in \mathcal{P}$ and $e \in E_P$. Since $\alpha \geq 0$,

$$\begin{aligned} \chi = & -\alpha_u + \max_{p \in \mathcal{P}^u} \left\{ \sum_{e \in E_P} \gamma_e^u \delta_e^p : p \in \mathcal{P}^u \right\} \\ = & -\alpha_u + \sum_{e \in E_P} \gamma_e^u - \min_{p \in \mathcal{P}^u} \left\{ \sum_{e \in E_P} \gamma_e^u \rho_e^p : p \in \mathcal{P}^u \right\} \end{aligned}$$

$\min_{p \in P^u} \left\{ \sum_{e \in E_P} \gamma_e^u \rho_e^p : p \in \mathcal{P}^u \right\}$ leads to a shortest path for the logical node pairs of u with non-negative weights.

b) Pricing for Protecting Spanning Tree Selection Variable: The pricing problems for the tree variable x_e^ℓ are disjoint for each physical link e and can be solved separately. Given the dual variable (β_e, γ_e^u), the reduced cost for variable x_e^ℓ is

$$1 - \beta_e - \sum_{u \in E_L} \gamma_e^u \xi_u^\ell. \tag{20}$$

To evaluate whether a dual solution is feasible or not, we verify the solution of

$$\psi = \max\{1 - \beta_e - \sum_{u \in E_L} \gamma_e^u \xi_u^\ell\}. \tag{21}$$

Since $\beta_e > 0$, maximizing ψ leads to the calculation of $\zeta = \min \left\{ \sum_{u \in E_L} \gamma_e^u \xi_u^\ell : e \in E_P \right\}$, which is the cost of a minimal cost spanning tree in the logical network with nonnegative weights, and can be solved efficiently with Prim's algorithm [27]. If $1 < \zeta + \beta_e$, then, the logical spanning tree selection has a negative reduced cost and is added to the restricted formulation.

2) Generation of New Columns: If in each iteration, either $\chi \geq 0$ or $\psi \geq 0$, then the column corresponding to the optimal primal solution at the iteration has positive reduced cost. The corresponding route and spanning tree based on $-\alpha_u + \sum_{e \in E_P} \gamma_e^u - \min_{p \in \mathcal{P}^u} \left\{ \sum_{e \in E_P} \gamma_e^u \rho_e^p : p \in \mathcal{P}^u \right\}$ and $\min \left\{ \sum_{u \in E_L} \gamma_e^u \xi_u^\ell : e \in E_P \right\}$ can be added to create a new restricted linear programming master problem. Note here that the formulation for ξ and ψ at each iteration corresponds to $u \in E_L$ and $e \in E_P$. Therefore, in each iteration, multiple columns corresponding to selected routes and trees may be added to the master problem.

3) Stop Criteria: If $\chi = 0$ and $\psi = 0$ in an iteration, then the dual variable set $(\alpha_u, \beta_e, \gamma_e^u)$ is dual feasible. By the strong duality theorem of linear programming, the dual and primal optimal of the (RPTS-R) is achieved.

The MILP formulation RPTS-R integrated with the column generation technique will be called RPTS-CGEN.

B. The Protecting Spanning Tree Algorithm ProHst: A Heuristic for the General Case

We now present a heuristic to find a survivable routing for the general case when the logical spanning tree set is not given. We first introduce some extra notations for the algorithm. Let PM be the collection of logical spanning trees and their corresponding lightpath routing; i.e., $PM = \{(\tau, M(\tau))\}$. Let $QM = \{[e, Q(e)]\}, Q(e) = \{u : e \in M(u), e \in E_P, u \in E_L\}$; i.e., QM is a collection of physical edges e and their corresponding logical edges $Q(e)$ whose lightpaths are routed through these physical edges. Let $w(e)$ and $w(u)$ be the weight on physical edge e and logical edge u, respectively, and $w(\tau) = \sum_{u \in \tau} w(u)$ be the weight of τ. α and β are the penalty functions used to adjust the weights of physical and logical links.

Algorithm 1 has two parts: generating a protecting spanning tree set and routing, and logical network augmentation. The weights on logical and physical edges are initialized to be 1 and the first logical spanning tree and the corresponding lightpaths for tree branches are generated. Following that, the weights of tree branches in the selected logical spanning tree τ and the weights of physical edges on the lightpaths of τ are both increased. τ and its lightpath routing, $(\tau, M(\tau))$, are then stored in PM and $M^{C(i)}(\mathcal{T})$ is updated with physical edges not utilized by τ.

Note that the purpose of assigning weights to logical links is to avoid generating new logical spanning trees with edges which are already in the current spanning tree set. We also assign weights to the physical links such that lightpaths for unmapped branches in the newly selected logical spanning tree would also avoid utilizing the same physical links in existing lightpaths.

After increasing the weights, the algorithm picks a minimum weight logical spanning tree which has at least an unmapped logical edge u and generates its lightpath p^u with a shortest path algorithm. The above procedure is repeated till each logical edge has a designated lightpath.

For each tree in the spanning tree collection, there exist physical edges not utilized by the routing of tree branches. In other words, the failure of these physical edges will not disconnect the spanning tree. Hence, the lightpath routing is survivable if the union of unutilized physical edges of all trees in the tree collection is E_P; otherwise, there exists a physical edge whose failure will disconnect all trees in the tree collection. In the latter case, logical augmentation plays an important role to guarantee survivability. We use Fig. 3 to illustrate the logical augmentation method.

Given Fig. 3 as the physical topology, let the initial routing of logical edge u be $M(u) = \{(1,4),(4,5),(5,8)\}$, where $i(u) = 1$ and $j(u) = 8$. Since an edge-disjoint path $p^{\widetilde{u}}$ to p^u does not exist, two edge-disjoint paths for u, $p^u = \{(1,2),(2,3),(3,5),(5,8)\}$ and $p^{\widetilde{u}} = \{(1,4),(4,6),(6,7),(7,8)\}$ are required to protect u from failure. In other words, we need to add two parallel logical edges with $p^u, p^{\widetilde{u}}$ as their routing.

With a given logical link u and its lightpath p^u, we have two types of mapping for logical augmentation: (1) *single augmentation*: we augment logical link u with parallel link $\widetilde{u}$

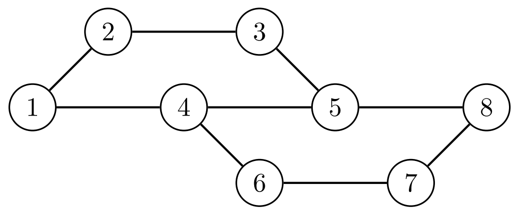


Fig. 3. Example illustrating the need for augmentation

and assign $p^{\widetilde{u}}$ as a lightpath of $\widetilde{u}$ if there exists an edge-disjoint path $p^{\widetilde{u}}$ to p^u; and (2) *double augmentation*: if an edge-disjoint lightpath of p^u does not exist, then we augment two logical links u^1 and u^2 parallel to u and generate two edge-disjoint paths connecting $i(u)$ and $j(u)$ as their lightpaths, which are used to replace the original logical link and its lightpath routing.

Let $m = |E_L|, n = |V_L|, q = |E_P|, r = |V_P|$. Without loss of generality, we assume that $m \geq n, q \geq r$. In our heuristic every tree contains at least one logical link that was not present in previously selected trees. So the heuristic requires at most $m - n + 2$ trees. As regards complexity of the heuristic, it uses three algorithms: minimum spanning tree algorithm (MST), shortest path algorithm (SP) and disjoint paths algorithm (DP). The complexity of MST algorithm (Kruskal's algorithm, running on the logical network) is $\mathcal{O}(m \log n)$ and it is used at most $2(m - n + 1) + 1$ times (due to lines 1 and 12 – 14). The complexity of the SP algorithm (running on the physical network) is $\mathcal{O}(q + r \log r)$ and is used m times. The DP algorithm (running on the physical network) has complexity $\mathcal{O}(q + r \log r)$ and is used at most m times. So the overall complexity of the heuristic is $\mathcal{O}(m(2m-2n+3) \log n + 2m(q + r \log r))$. Note that the DP algorithm can be implemented using a max flow algorithm or Suurballe's algorithm [27][28].

VI. Simulation

In this section, we provide experimental results demonstrating the effectiveness of our approach. Specifically, our results stated below cover comparative evaluation of RPTS-CGEN and our heuristic ProHst.

1) Comparison of RPTS-CGEN with the MILP formulation (to be called DSS) given in [29].
2) Comparison of RPTS-CGEN with ProHst.

Symbols used are defined in Table I.

A. Comparison of RPTS-CGEN with the MILP formulation DSS [29]

DSS was the first MILP formulation for the survivable routing problem that does not require explicit enumeration of all cuts. It uses a novel technique to test the connectivity of the logical topology when a physical link fails. So we used DSS as a benchmark and compared it with RPTS-CGEN. Since DSS takes excessively longer computation time, our tests are limited to the small size physical networks introduced in [30][31] and shown in Figs. 4(a) – 4(f). The logical topologies were chosen to be two-edge connected, whose logical nodes are subsets of

Algorithm 1 The protecting spanning tree algorithm ProHst

Input: $G_P = (V_P, E_P), G_L = (V_L, E_L), \mathcal{T} = \emptyset, QM = \emptyset, PM = \emptyset, w^i(e) = 1, e \in E_P, w^i(u) = 1, u \in E_L, i = 0, \alpha = \sqrt{|E_P|}, \beta = \sqrt{|E_L|}$.

1: Pick τ, $\mathcal{T} = \mathcal{T} \bigcup \tau$
2: **for all** $u \in \tau$ **do**
3: Generate lightpath p^u {p^u forms $M(\tau)$}
4: $w^i(e) = w^i(e) + \alpha, \forall e \in p^u$
5: $w^i(u) = w^i(u) + \beta$
6: **end for**
7: **if** $w^i(e) = 1, \forall e \in E_P$ **then**
8: $M^{C(i)}(\mathcal{T}) = M^{C(i)}(\mathcal{T}) \bigcup \{e\}$
9: **end if**
10: $PM = PM \bigcup \{(\tau, M(\tau))\}$
11: **while** $\mathcal{T} \neq E_L$ **do**
12: Generate a minimum weight protecting spanning tree, τ^*, i.e., $w(\tau^*) = \min_{\tau \in G_L} \{w(\tau)\}$
13: **if** $\mathcal{T} = \mathcal{T} \bigcup \tau^*$ **then**
14: Find u with the largest weight. Set all edge weight to $(w^i(u) + 1)$ for all edges in $\mathcal{T}$. Go to 11
15: **else**
16: $\mathcal{T} = \mathcal{T} \bigcup \tau^*$
17: **for all** $u \in \tau^*, M(u) = \emptyset$ **do**
18: Generate the minimum weight lightpath p^u
19: $w^{i+1}(e) = w^i(e) + \alpha, \forall e \in p^u$
20: **end for**
21: **end if**
22: **for all** $u \in \tau^*$ **do**
23: $w^{i+1}(u) = w^i(u) + \beta$
24: **end for**
25: **for all** $e \in E_P$ **do**
26: **if** $w^{i+1}(e) = w^i(e)$ **then**
27: $M^{C(i+1)}(\mathcal{T}) = M^{C(i)}(\mathcal{T}) \cup \{e\}$ {Update the unutilized physical edge set}
28: **end if**
29: **end for**
30: $\mathcal{T} = \mathcal{T} \cup \tau^*, PM = PM \cup \{(\tau^*, M(\tau^*))\}$
31: **if** $M^{C(i+1)}(\mathcal{T}) = E_P$ **then**
32: **TERMINATE** {Found survivable routing}
33: **else**
34: i = i + 1
35: **end if**
36: **end while**
37: Generate all-utilized physical edge collection $\Omega^i, \Omega^i = E_P \setminus M^{C(i)}(\mathcal{T})$
38: Generate LM, the mapping of the physical edge e to logical edges whose lightpaths are routed through e, $LM = \{(e, LM(e))\}, LM(e) = \{u : e \in M(u), e \in \Omega^i\}$
39: **for all** $e \in \Omega^i$ **do**
40: **for all** $u \in LM(e)$ **do**
41: **if** $\exists$ path $p^{\widetilde{u}}$ edge-disjoint to p^u **then**
42: Add a logical edge $\widetilde{u}$ parallel to u, $M(\widetilde{u}) = p^{\widetilde{u}}$
43: **else**
44: Add two logical edges $\widetilde{u^1}, \widetilde{u^2}$ parallel to u in G_L
45: Map $\widetilde{u^1}, \widetilde{u^2}$ into edge-disjoint lightpaths
46: Remove u from G_L and $M(u)$ from $M(\mathcal{T})$
47: **end if**
48: **end for**
49: **end for**

TABLE I
NOTATIONS USED IN SIMULATION RESULTS

Notation	Explanation
$r_{L/P}$	logical to physical node ratio
MinDeg/MaxDeg /AvgDeg	minimum/maximum/average node degree
lCon/pCon	logical/physical topology connectivity
RPTS-CGEN	MILP with column generation technique
tNum	cardinality of protecting spanning tree set
Time	execution time of simulation
DSS	protecting spanning tree formulation
oGap	optimality gap
pnN	number of nodes in physical topology
2Con/3Con	2-/3-edge connectivity of physical topologies
ProHst	heuristic for the general case
RunNum	number of testing instances where survivable routing is found

physical nodes with cardinality $|V_L| = r_{L/P} * |V_P|$, where $r_{L/P} = 0.5$. The logical nodes were randomly mapped to physical nodes and the logical links were selected to satisfy the connectivity requirements. Tables II and III summarize the physical and logical network information corresponding to Figs. 4(a) – 4(f). In total, 40 testing cases were generated to compare the performance of RPTS-CGEN with DSS. For each case, we report the average value of 20 instances with corresponding randomly generated logical networks.

We implemented and tested all (M)ILP approaches, with CPLEX 12.3 C++ concert callable library. We assigned a single thread to solve each (M)ILP program to avoid the complexity of the default cut generation brought in by the multi-thread functionality in CPLEX. ProHst is implemented in C++ with COIN-OR LEMON library [32], which also utilizes a single thread during execution. All codes were run on a machine with quad-core (with hyperthreading) AMD Opteron processor and 32GB memory.

Table IV gives the simulation results comparing RPTS-CGEN with DSS. Note that both are exact solution approaches. Since DSS takes excessive computation time, we set a time limit of 30 minutes for each testing instance. We report the time taken (“Time”, in seconds) and the number of trees generated (“tNum”) in Table IV for RPTS-CGEN which converges within the time limit and produces an optimal solution. If DSS terminates within 30 minutes, we report the optimal solution with the computation time and set its optimality gap (“oGap”) to 0%; otherwise, we report the optimality gap obtained from CPLEX and the time value is the time limit (30 minutes). Table IV shows that RPTS-CGEN efficiently solves all testing cases within a second, while DSS does not converge within 30 minutes for four cases. For physical networks G6, NOBEL-Germany, Norway, DFN, and NSF, the optimality gap is at least 33.33%. For the other two cases, DSS takes at least five times the computational time taken by RPTS-CGEN. It is obvious that RPTS-CGEN outperforms DSS in terms of the time taken.

B. Comparative Performance of RPTS-CGEN with ProHst

We next evaluate both RPTS-CGEN and ProHst on median size networks which were randomly generated. The generated

TABLE II
PHYSICAL TOPOLOGIES INFORMATION [13]

	Nodes	Edges	MinDeg	MaxDeg	AvgDeg
G6	17	31	2	5	3.64
NOBEL-Germany	17	26	2	6	3.06
Norway	27	51	2	6	3.78
DFN	11	47	2	10	8.55
PDH	11	34	4	8	6.18
NSF	14	21	2	4	3

TABLE III
LOGICAL TOPOLOGIES INFORMATION

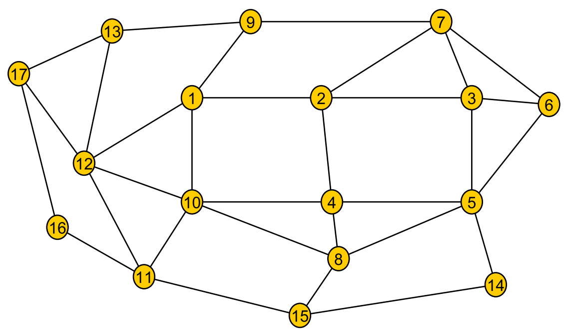
	lCon	$r_{L/P}$	Nodes	Edges	MinDeg	MaxDeg	AvgDeg
G6	3	0.5	8	12	3	3	3.00
NOBEL-Germany	3	0.5	8	12	3	3	3.00
Norway	3	0.5	13	20	3	4	3.77
DFN	3	0.5	5	8	3	4	3.20
PDH	3	0.5	5	8	3	4	3.20
NSF	3	0.5	5	8	3	4	3.20

TABLE IV
RESULTS FOR ILP COMPARISON WITH 2-EDGE CONNECTIVITY PHYSICAL NETWORK

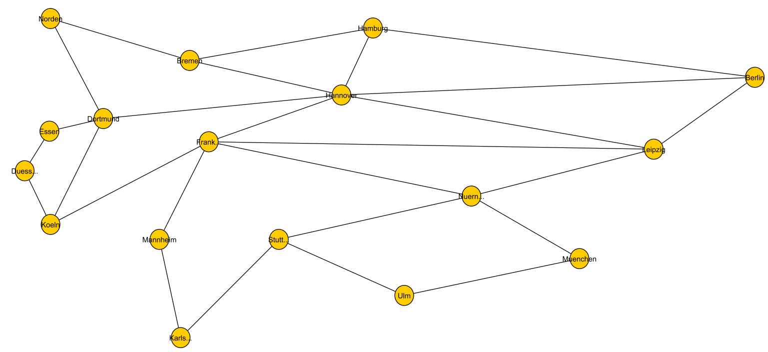
	$r_{L/P}$	lCon	RPTS-CGEN		DSS	
			tNum	Time	Time	oGap
G6	0.5	3	6.56	0.230	30m	66.67%
NOBEL-Germany	0.5	3	6.22	0.241	30m	33.33%
Norway	0.5	3	9.78	0.615	30m	85.71%
DFN	0.5	3	3.50	0.162	1.326s	0.00%
PDH	0.5	3	3.10	0.128	0.714s	0.00%
NSF	0.5	3	5.20	0.179	30m	33.33%

physical networks are with nodes 20, 30, and 40 with two-edge and three-edge connectivity. The logical networks are with node ratio $r_{L/P}$ = 0.4, 0.6, 0.8, and 1.0, and three-edge and four-edge connectivity. Therefore, in total, 48 testing cases were generated and tested. For each testing case, we run 20 random instances and report the average value of the computation time and the number of logical spanning trees generated in Table V. In some instances a survivable routing does not exist, so we report results only for the cases when a survivable routing is found. “-” in Table V denotes those cases when ProHst did not converge within 30 minutes.

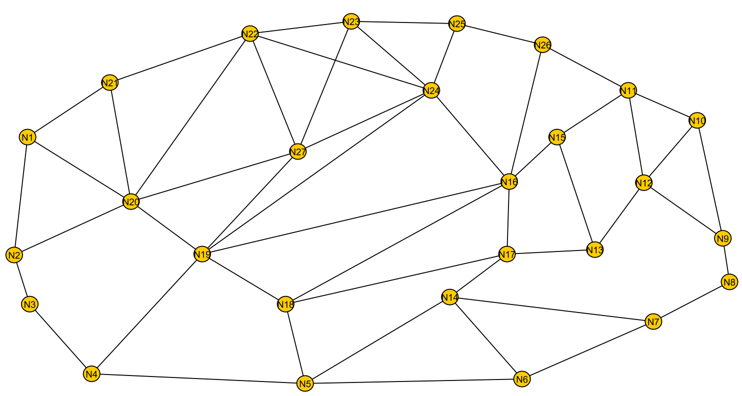
As shown in Table V, RPTS-CGEN converges in more instances within the 30-minute time limit when the physical connectivity is higher. In general, ProHst, which has the computation time of less than 0.02 seconds for all testing cases, is much faster than RPTS-CGEN. We observe that RPTS-CGEN does not converge within the 30-minute time limit when the logical network is sparse (as measured by the ratio of connectivity to number of nodes). This is because when the logical network is sparse, RPTS-CGEN searches more trees for achieving protection.



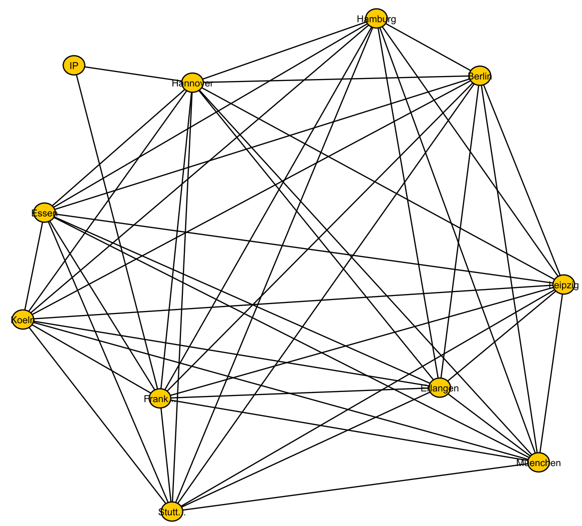
(a) G6 [30]



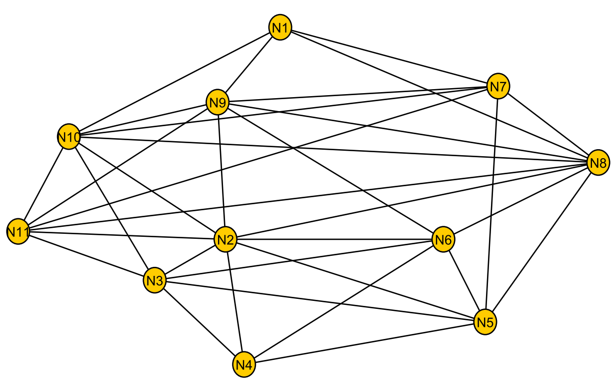
(b) NOBEL-GERMANY [31]



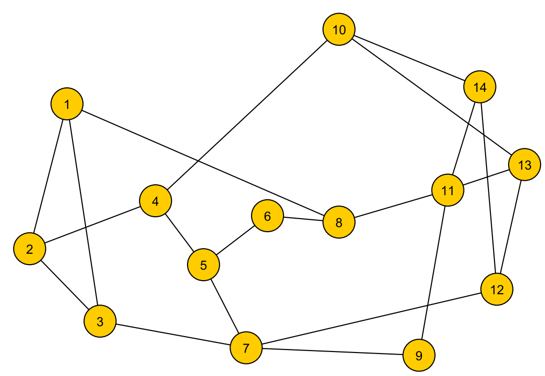
(c) Norway [31]



(d) DFN [31]



(e) PDH [31]



(f) NSF [31]

Fig. 4. Testing physical networks from literature

TABLE V
RESULTS FOR COLUMN GENERATION FOR MEDIAN-SIZE NETWORKS

pnN	$r_{L/P}$	lCon	2Con		3Con	
			ProHst	RTPS-CGEN	ProHst	RPTS-CGEN
20	0.4	3	0.004	0.276 / 8	0.004	0.325 / 7
		4	0.004	0.418 / 8	0.005	0.490 / 6
	0.6	3	0.005	0.829 / 12	0.005	0.785 / 9
		4	0.005	3.982 / 12	0.006	1.208 / 9
	0.8	3	0.006	–	0.007	2.602 / 14
		4	0.006	–	0.007	2.633 / 13
	1.0	3	0.006	–	0.008	5.806 / 15
		4	0.007	–	0.008	4.243 / 15
30	0.4	3	0.005	1.142 / 11	0.007	1.555 / 10
		4	0.006	2.730 / 12	0.008	2.656 / 10
	0.6	3	0.006	1.104/8	0.010	2.825 / 15
		4	0.009	–	0.010	8.419 / 13
40	0.4	3	0.007	1.910 / 18	0.008	10.11 / 14
		4	0.007	1.455 / 7	0.010	14.86 / 13
	0.6	3	0.010	1.951 / 4	0.013	–
		4	0.010	3.764 / 12	0.012	–

TABLE VI
RESULTS OF HEURISTIC ALGORITHM FOR MEDIAN SIZE NETWORKS

pnN	$r_{L/P}$	lCon	RunNum	tNum	Time
20	0.4	3	17	4.71	0.003
		4	20	3.70	0.004
	0.6	3	15	5.27	0.004
		4	19	4.00	0.005
	0.8	3	9	7.11	0.006
		4	19	4.05	0.006
	1	3	3	6.33	0.006
		4	14	4.57	0.007
30	0.4	3	12	5.08	0.006
		4	17	4.06	0.006
	0.6	3	7	7.57	0.008
		4	12	4.35	0.009
	0.8	3	3	8.33	0.011
		4	14	4.43	0.010
	1	3	1	9.00	0.011
		4	15	4.60	0.012
40	0.4	3	5	6.20	0.009
		4	15	4.53	0.010
	0.6	3	5	8.25	0.014
		4	18	4.72	0.012
	0.8	3	1	12.00	0.015
		4	13	5.08	0.016
	1	4	14	5.21	0.017

In Table VI, we provide results to demonstrate the effectiveness of ProHst. We report the number of instances (out of a total of twenty) when ProHst produces survivable routings. This number is referred to as RunNum. In general, it is observed that the success rate is higher when the logical network is dense. This is because in such cases, more spanning trees will be available to provide protection.

We have done simulations to compare SMART with ProHst with 40 randomly generated instances for each physical network, in which $r_{L/P} = 0.5$ and the logical networks are three-connected. To compare these two heuristics fairly, the augmentation in ProHst is disabled. The results are presented in Table VII. We observe that the success rate for ProHst is close to the rate for SMART. We wish to note that SMART involves identification of circuits, contractions and expansion of circuits, and finding link disjoint paths in the physical network between pairs of nodes of the logical links in selected circuits. Contractions and expansion operations require sophisticated data structures for implementation. Finding link-disjoint paths between multiple pairs of nodes is NP-complete, and approximation algorithms available for this are far from satisfactory. On the other hand, ProHst requires Dijkstra's shortest path algorithm, Kruskal/Prim's minimum spanning tree algorithm, and Suurballe's disjoint path algorithm, which are quite easy to implement.

TABLE VII
SIMULATIONS COMPARING PROHST WITH SMART

	G6	NOBEL-Germany	Norway	DFN	PDH	NSF
SMART	40	38	40	40	40	40
ProHst	40	32	33	40	38	40

To verify how our proposed heuristic performs over large-scale optical networks, we took the CORONET Continental United States (CONUS) topology [33], illustrated in Fig. 5, as the physical network which has 75 nodes, 99 edges, and an average of 2.6 nodal degrees. Different from prior testing cases, the CONUS network has only 36 nodes (48%) with nodal degrees three and above. We constructed logical networks with 36 nodes and connectivities three and four, where logical nodes are randomly mapped onto physical nodes with degrees three and above. For these two testing scenarios, we ran 10 cases each and report the computational results as follows. With logical connectivity three, our heuristic cannot find a survivable routing. Our heuristic can obtain one survivable routing when the logical connectivity is four. Even if the survivable routing is not found for the tested cases, ProHst still generates a collection of logical spanning trees and routing. On average, 34 and 33 spanning trees were generated for the tested cases with logical connectivity three and four, respectively. Though these generated logical spanning trees and the routing cannot guarantee survivability, on average, our algorithm protects about 70.9% and 92.4% of the total number of physical edges for logical connectivities three and four, respectively. We wish to note that neither RPTS-CGEN nor DSS can produce feasible solutions after hours of computation for the same testing cases. We observe from these computational results that (1) when the physical network is with low nodal degree, it is hard to find a survivable routing, this is because in sparse networks choices available for mappings are limited; and (2) even if the proposed heuristic cannot find a survivable routing, it can still generate a routing which can protect the logical topology against majority of the physical link failures in significantly shorter time than RPTS-CGEN and DSS.

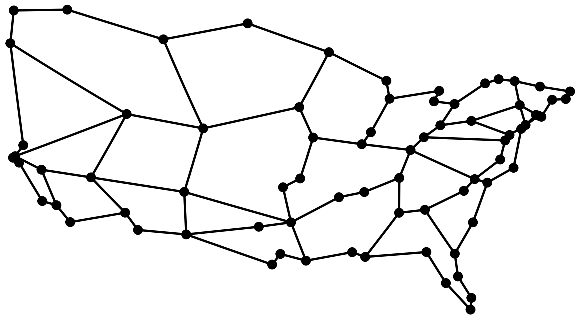


Fig. 5. CONUS

VII. Conclusion

In this paper, we proposed a novel approach based on the concept of protecting spanning tree set of the logical topology. The basic idea is to identify a set of spanning trees of the logical topology and a routing of the logical links such that at least one of these trees remains connected after a physical link failure. Given a set of logical spanning trees we first presented three optimization problems with varying degrees of difficulty relating to this approach and discussed their (M)ILP formulations. To handle the general case when the routing and the tree set are to be determined we presented a new MILP formulation. We discussed how one can use the column generation technique to speed up the execution of this formulation and also obviate the need to store all spanning trees at the beginning of the execution of this MILP. This MILP along with the column generation technique is called RPTS-CGEN. For the general case we also presented a heuristic approach. We incorporate in this heuristic a method to augment the logical topology with additional links to guarantee a survivable routing.

The new heuristic has several nice features. It only requires a shortest path algorithm and an algorithm to generate appropriate spanning trees. An algorithm such as the one in [34] that generates spanning trees one at a time in a simple and elegant manner is an appropriate candidate for use in this approach. Though this spanning tree generation algorithm was not used in Algorithm 1, incorporating it in Algorithm 1 will be an interesting future direction of research.

Moreover, the approach identifies a group of spanning trees of the logical graph and a lightpath routing of the logical links. Each tree is identified with a group of physical edges such that failure of one or more of these edges will leave at least one of the trees remains connected, guaranteeing the survivability of the logical topology against these group failures. Thus the algorithm provides a framework for generating a survivable routing for the SRLG failure case.

Extensive simulations have been performed to evaluate our MILP formulation RPTS-CGEN and our heuristic ProHst. We have provided comparison of RPTS-CGEN with another MILP formulation DSS [29]. RPTS-CGEN provides survivable routing in all cases considered in less than 0.2 seconds, whereas DSS did not converge within the specific time limit of 30 minutes.

We have also provided comparison of RPTS-CGEN with ProHst. In all cases, ProHst took less than 0.02 seconds and is much faster than RPTS-CGEN. It is also found that the success rate of ProHst in finding survivable routings is much better when the logical graph is dense. In most cases, we observed that only a small number of augmented edges are required for our heuristic when the routing is not survivable.

We believe that the approach along with mathematical programming [1] and structural approaches [2][3][4] provide several insights into the survivable logical topology routing problem in an IP-over-WDM optical network.

References

[1] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," in *Proc. IEEE INFOCOM*, vol. 1, 2001, pp. 348–357.

[2] M. Kurant and P. Thiran, "On survivable routing of mesh topologies in IP-over-WDM network," in *Proc. IEEE INFOCOM*, vol. 2, 2005, pp. 1106–1116.

[3] K. Thulasiraman, M. S. Javed, and G. Xue, "Circuits/Cutsets duality and a unified algorithmic framework for survivable logical topology design in IP-over-WDM optical networks," in *Proc. IEEE INFOCOM*, 2009, pp. 1026–1034.

[4] K. Thulasiraman, M. Javed, and G. Xue, "Primal meets dual: A generalized theory of logical topology survivability in IP-over-WDM optical networks," in *Proc. IEEE COMSNETS*, 2010, pp. 1–10.

[5] K. Thulasiraman, T. Lin, Z. Zhou, and G. Xue, "Robustness of logical topology mapping algorithms for survivability against multiple failures in an IP-over-WDM optical network," in *Proc. IEEE COMSNETS*, 2012, pp. 1–9.

[6] Z. Zhou, T. Lin, K. Thulasiraman, G. Xue, and S. Sahni, "Novel survivable logical topology routing in IP-over-WDM networks by logical protecting spanning tree set," in *Proc. IEEE ICUMT*, 2012, pp. 650–656.

[7] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, ser. The IBM Research Symposia Series, R. Miller, J. Thatcher, and J. Bohlinger, Eds. Springer US, 1972, pp. 85–103.

[8] A. Todimala and B. Ramamurthy, "A scalable approach for survivable virtual topology routing in optical WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 25, pp. 63–69, 2007.

[9] J. Strand, A. L. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Commun. Mag.*, vol. 39, pp. 81–87, 2001.

[10] K. Lee, E. Modiano, and H.-W. Lee, "Cross-layer survivability in WDM-based networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1000–1013, Aug. 2011.

[11] D. D. Kan, A. Narula-Tam, and E. Modiano, "Lightpath routing and capacity assignment for survivable IP-over-WDM networks," in *Proc. IEEE DRCN*, 2009, pp. 37–44.

[12] T. Lin, Z. Zhou, and K. Thulasiraman, "Logical topology survivability in IP-over-WDM networks: Survivable lightpath routing for maximum logical topology capacity and minimum spare capacity requirements," in *Proc. IEEE DRCN*, Oct. 2011, pp. 1–8.

[13] T. Lin, Z. Zhou, K. Thulasiraman, G. Xue, and S. Sahni, "Unified mathematical programming frameworks for survivable logical topology routing in IP-over-WDM optical networks," *IEEE J. Opt. Commun. Netw.*, vol. 6, no. 2, pp. 190–203, 2014.

[14] T. Lin, Z. Zhou, K. Thulasiraman, and G. Xue, "Optimum logical topology routing in an IP-over-WDM optical network and physical link failure localization: An integrated approach," in *Proc. IEEE ICUMT*, 2013, pp. 82–88.

[15] B. Wu, P. H. Ho, and K. L. Yeung, "Monitoring trail: on fast link failure localization in all-optical WDM mesh network," *J. Lightw. Technol.*, vol. 27, pp. 4175–4185, 2009.

[16] S. Ahuja, S. Ramasubramanian, and M. Krunz, "Single link failure detection in all-optical networks using monitoring cycles and paths," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 1080–1093, 2009.

[17] Z. Zhou, T. Lin, K. Thulasiraman, G. Xue, and S. Sahni, "Cross-layer network survivability under multiple cross-layer metrics," *IEEE J. Opt. Commun. Netw.*, vol. 7, no. 6, pp. 540–553, June 2015.

[18] M. R. Rahman and R. Boutaba, "SVNE: Survivable virtual network embedding algorithms for network virtualization," *IEEE Trans. Netw. Service Manag.*, vol. 10, no. 2, pp. 105–118, 2013.

[19] Q. Hu, Y. Wang, and X. Cao, "Survivable network virtualization for single facility node failure: A network flow perspective," *Opt. Switching and Netw.*, vol. 10, no. 4, pp. 406–415, 2013.

[20] S. Lee, J. Liu, and Y. Chen, "An ear-decomposition based approach for survivable routing in WDM networks," in *Proc. IEEE AINA*, vol. 1, 2005, pp. 459–464.

[21] M. Javed, K. Thulasiraman, M. Gaines, and G. Xue, "Survivability aware routing of logical topologies: On Thiran-Kurant approach, enhancements and evaluation," in *Proc. IEEE GLOBECOM*, 2006, pp. 1–6.

[22] M. Javed, K. Thulasiraman, and G. Xue, "Lightpaths routing for single link failure survivability in IP-over-WDM networks," *J. Commun. Netw.*, pp. 394–401, 2007.

[23] K. Thulasiraman, T. Lin, M. Javed, and G. Xue, "Logical topology augmentation for guaranteed survivability under multiple failures in IP-over-WDM optical networks," *Opt. Switching Netw.*, pp. 206–214, 2010.

[24] C. Liu and L. Ruan, "A new survivable mapping problem in IP-over-WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 25, pp. 25–34, 2007.

[25] M. Ehrgott, *Multicriteria optimization*. Springer Science & Business Media, 2006.

[26] J. Desrosiers and M. Lübbecke, "A primer in column generation," in *Column Generation*, G. Desaulniers, J. Desrosiers, and M. Solomon, Eds. Springer US, 2005, pp. 1–32.

[27] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Feb. 1993.

[28] 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.

[29] Q. Deng, G. Sasaki, and C.-F. Su, "Survivable IP over WDM: An efficient mathematical programming problem formulation," in *Proc. Allerton Conference on Commun., Control and Comput.*, Monticello, IL, Oct. 2002.

[30] K. S. Ho and K. W. Cheung, "Generalized survivable network," *IEEE/ACM Trans. Netw.*, vol. 15, pp. 750–760, Aug. 2007.

[31] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proc. INOC*, vol. 55, 2007, pp. 278–286.

[32] "Library for efficient modeling and optimization in networks (LEMON)." [Online]. Available: http://lemon.cs.elte.hu/trac/lemon

[33] G. Clapp, R. Skoog, A. Von Lehmen, and B. Wilson, "Management of switched systems at 100 Tbps: The DARPA CORONET program," in *Proc. IEEE Photon. in Switching*, 2009, pp. 1–4.

[34] R. Jayakumar, K. Thulasiraman, and M. N. S. Swamy, "Complexity of computation of a spanning tree enumeration algorithm," *IEEE Trans. Circuits Syst.*, pp. 853–860, Oct. 1984.