Survivable Cloud Network Design Against Multiple Failures Through Protecting Spanning Trees

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

***Abstract*—Survivable design of cross-layer networks, such as the cloud computing infrastructure, lies in its resource deployment and allocation and mapping of the logical (virtual datacenter/IP) network into the physical infrastructure [cloud backbone/wavelength division multiplexing (WDM)] such that link or node failure(s) in the physical infrastructure would not result in cascading failures in the logical network. Most of the prior approaches for survivable cross-layer network design aim at single-link failure scenario, which are not applicable to the more challenging multifailure scenarios. Also, as many of these approaches use the cross-layer cut concept, enumeration of all cuts in the network is required and thus introducing exponential number of constraints. To overcome these difficulties, we investigate, in this paper, survivable mapping approaches against multiple physical link failures and its special case, shared risk link group (SRLG) failure. We present the necessary and sufficient conditions based on both cross-layer spanning trees and cutsets to guarantee a survivable mapping when multiple physical link failures occur. Based on the necessary and sufficient conditions, we propose to solve the problem through: mixed-integer linear programs that avoid enumerating all combinations of link failures; and an algorithm that generates/adds logical spanning trees sequentially. Our simulation results show that the proposed approaches can produce survivable mappings effectively against both k- and SRLG-failures.**

***Index Terms*—Cross-layer networks, multiple failures, optical communication, survivability, SRLG failures.**

I. Introduction

CLOUD computing through which demands of geographically distributed individuals and enterprise are realized seamlessly over a common physical cloud infrastructure/backbone has attracted significant attention in recent years both in academia and industry. Cloud service providers offer their users an abstracted layer of computational (processors/memory) and communication resources considered as a virtual datacenter, and cloud services are then carried out through virtual-machine deployment and its mapping to the physical datacenters and their data connections in the cloud infrastructure [1], [2]. Figure 1 illustrates a cloud computing infrastructure, where the virtual datacenter (top-layer) consists of interconnected virtual machines is mapped onto the cloud infrastructure (bottom-layer) with datacenters and communication networks. As each cloud service provider usually supports multiple tenants, each virtual machine is allocated with a fraction of the computational capacity of the correspondingly mapped datacenters, and the links connecting virtual machines would also occupy a portion of the overall capacity of the fiber-optic connections at the cloud backbone. Hence, failure(s) in the physical infrastructure may disconnect both physical datacenters and virtual machines and decrease the computational and communication capacities of the cloud. This has motivated studies on the problems of resilient and survivable network design.

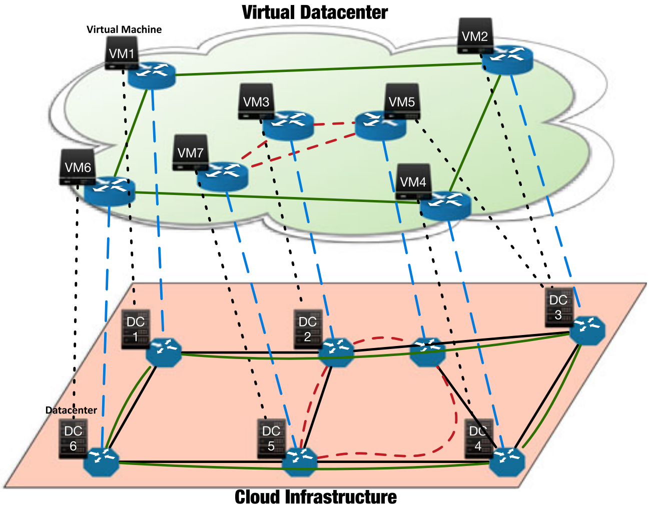

Fig. 1. Cloud computing infrastructure.

A virtual datacenter is claimed to be survivable if its services sustain when failure(s), such as link failures in optical networks [3] or power outage in datacenters [4], occur in the cloud infrastructure. In this paper, we use network survivability as a measure to identify if a network remains connected against instantaneous failures [5]–[8]. The approaches to guarantee virtual datacenter survivability can be achieved in different layers. Since the optical network is the major communication media in a cloud infrastructure connecting datacenters, the first approach is to protect the internet protocol (IP) traffic over optical networks or virtual network traffic over physical substrate network through backup protection, which ensures that the fiber/phyisical link is protected by either a dedicated or shared backup fiber/physical link. $1 + 1$, 1:1, and 1:N protection [9] are

Manuscript received August 13, 2016; revised October 17, 2016 and December 1, 2016; accepted December 1, 2016. Date of publication December 7, 2016; date of current version January 9, 2017.

Z. Zhou is with United Airlines, Chicago, IL 60649 USA (e-mail: zhilizhou@gmail.com).

T. Lin is with the Department of Computer Science and Information Systems, Bradley University, Peoria, IL 61625 USA (e-mail: djlin@bradley.edu).

K. Thulasiraman is with the School of Computer Science, University of Oklahoma, Norman, OK 73019 USA (e-mail: thulasi@ou.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2016.2637352

the traditional but most commonly applied approaches. For the link-failure case, two-stage approaches with reactive re-routing are explored in [10], [11], and network/service backup provisioning as proactive approaches are proposed in [12], [13]. As a node failure results in multiple link failures, the failure-dependent protection is introduced in [6], [14] where backup nodes are utilized; and the failure-independent protection discussed in [15]–[19] targets to provide the same level of protection under a single node/link failure scenario with less substrate resources. A closely related work on reliable cross-layer network design emphasizes controlling the failure probability of the entire network considering link failure probability. Probabilistic analysis [20] and scenario-based approaches [21]–[24] are proposed for multiple link failures and disaster recovery.

Different from backup-provisioning and probability-based approaches mentioned above, our cross-layer survivable routing scheme maximally utilizes existing resources and protects the topology through routing. Due to its NP-completeness [25], early works are mostly concentrated on single physical link failure scenario. Kurant and Thiran [26] proposed a disjoint-path-based protection scheme to guarantee the sufficient condition for cross-layer network survivability, where logical cycles are mapped onto link-disjoint paths iteratively. Zhou *et al.* [27], [28] provided necessary and sufficient conditions for the existence of survivable design using the concept of protecting spanning trees, while Modiano and Narula-Tam [29] proposed a different approach based on cross-layer cutsets. To address the scenario of multiple physical link failures, Todimala and Ramamurthy [30] studied survivable mappings against a single SRLG failure. The approach proposed in [30] is also based on cross-layer cutsets introduced in [29]. Here SRLG means a group of network links routed through the same fiber, thus the fiber cut causes simultaneous failures to this group of links. Jaumard *et al.* [31] and Parandehgheibi *et al.* [32] proposed the use of minimal path sets for survivable routing after single link failure. Xi *et al.* [33] considered rerouting as a restoration scheme to recover SRLG failures in IP-over-WDM networks. Similar concepts have also been applied to the survivable cloud network mapping problem [7]. For instance, an integrated approach with content placement/replica and routing was introduced in [4], [34], anycast routing presented in [35], [36], and multi-path routing schemes discussed in [37], [38].

Multiple physical link failures bring in more challenges for cloud network mapping because (1) none of the survivable mapping designs for single failure would work if a link mapping is routed through multiple failed physical links; (2) compared with a single physical link failure scenario which has in total $|E_P|$ possible single physical link failures, there exist $C_k^{|E_P|}$ number of k-edge combinations under the k-failure scenario, which all need to be considered; (3) intuitively, to capture full information for k-link failures, where $k \in 1, \ldots, |E_P|$, all possible k-link combinations should be enumerated, which has an exponential number of combinations $(2^{|E_P|} - 1)$; and last but not least, even though there are only $|R_E|$ number of failure combinations in the SRLG scenario, each element in R_E may include different number of links. Thus, it is important to explore a solution approach for above problems without enumerating all possible combinations, which is the key question we would like to answer.

In this paper, we study the survivable cloud network mapping problem against the generalized k physical link failures and its special case, the SRLG failure. We propose the necessary and sufficient conditions for survivable cloud network mapping against multiple physical link failures and design the exact solution approach through mixed-integer linear program (MILP) formulations. The contributions of this paper are as follows. We first study the survivable network mapping problem against multiple-link failure scenarios and prove the necessary and sufficient conditions for k- and SRLG-survivability under a cross-layer network setting. Second, different from the approaches in [29], [30], our proposed exact solution approaches do not enumerate all possible combinations of failed physical link sets, which greatly reduces the computation time. In addition to the mathematical formulation approach, we also design a heuristic which can sequentially generate and select logical spanning trees as well as their corresponding tree branch mappings.

The rest of paper is organized as follows. Section II provides the network setting in a cloud infrastructure and definitions of k- and SRLG-survivability. Necessary and sufficient conditions for cross-layer network survivability against k- and SRLG- failures are presented in Section III. From the necessary and sufficient conditions we propose in Section IV the MILP formulations realizing the protecting spanning tree concept for both k- and SRLG- failures. In Section V, we present an SRLG-survivable algorithm which sequentially selects protecting spanning trees and generates their corresponding mappings. We also demonstrate its extension to k-survivability. Finally, simulation results for both the MILP model and proposed heuristic algorithm are reported in Section VI.

II. Network Setting and Survivability

Notations utilized in this paper are presented in Table I. Let $G_S = (V_S, E_S), G_P = (V_P, E_P)$ denote the virtual datacenter and physical cloud infrastructure, respectively, where V_S, V_P represent the node sets and E_S, E_P represent the link sets in their corresponding network. Each $s \in V_S$ is mapped onto a node $u \in V_P$, while requests between nodes s and t, represented as a link $(s,t) \in E_S$, are realized through a route (denoted as p_{st}) in G_P connecting s and t's corresponding nodes in V_P. Here we use *cloud network mapping* to represent both node and link mappings of the virtual datacenter. Let R_E denote an SRLG set containing all SRLG failure scenarios, where each element $r \in R_E$ represents a single SRLG. Let $k(r)$ represent the number of physical links in SRLG r; i.e., $r = \{(i_1, j_1), (i_2, j_2), \cdots, (i_{k(r)}, j_{k(r)})\}$.

Definition 1: A cloud network mapping is *SRLG-survivable* if the virtual datacenter remains connected after any SRLG failure in R_E.

Definition 2: A cloud network mapping is *k-survivable* if the virtual datacenter remains connected after arbitrary k physical link failures.

TABLE I
NOTATIONS

Notation	Representation
G_P	The physical cloud infrastructure, $G_P = (V_P, E_P)$ with V_P and E_P as its node and edge sets
G_S	The virtual datacenter network, $G_S = (V_S, E_S)$ with V_S and E_S as its node and edge sets
(i, j)	The physical link, $i, j \in V_P$, $(i, j) \in E_P$
(s, t)	The virtual link, $s, t \in V_S$, $(s, t) \in E_S$
R_E	Shared risk link groups (SRLGs), whose element $r = \{(e_1^r, e_2^r, \ldots, e_k^r) : e_\ell^r \in E_P\}, r \in R_E$ represents an SRLG
$\mathcal{T}$	Spanning tree set, $\mathcal{T} \subset G_S$, whose element $\tau \in \mathcal{T}$ is a spanning tree
p_{st}	The routing of $(s, t) \in E_S$ in G_P, where $p_{st} \subset E_P$
$\Lambda(i, j)$	A set of links in G_S which are routed through physical link (i, j), i.e., $\{(s, t) : (i, j) \in p_{st}, (s, t) \in E_S\}$
Variables	Representation
y_{ij}^{st}	binary variable which indicates whether p_{st} is routed through (i, j) or not, where $(i, j) \in E_P$; if yes, $y_{ij}^{st} = 1$, otherwise $y_{ij}^{st} = 0$
$y_{i_1 j_1, \cdots, i_k j_k}^{st}$	binary variable which indicates whether p_{st} is routed through any links in $(i_1, j_1), \cdots, (i_k, j_k)$; if yes, $y_{i_1 j_1, \cdots, i_k j_k}^{st} = 1$, otherwise $y_{i_1 j_1, \cdots, i_k j_k}^{st} = 0$
$\mu_{i_1 j_1, i_2 j_2, \ldots, i_k j_k}^{st}$	binary variable which indicates if p_{st} is *not* routed through any of the links in $(i_1, j_1), \cdots, (i_k, j_k)$; if yes, $\mu_{i_1 j_1, i_2 j_2, \ldots, i_k j_k}^{st} = 1$, otherwise $\mu_{i_1 j_1, i_2 j_2, \ldots, i_k j_k}^{st} = 0$

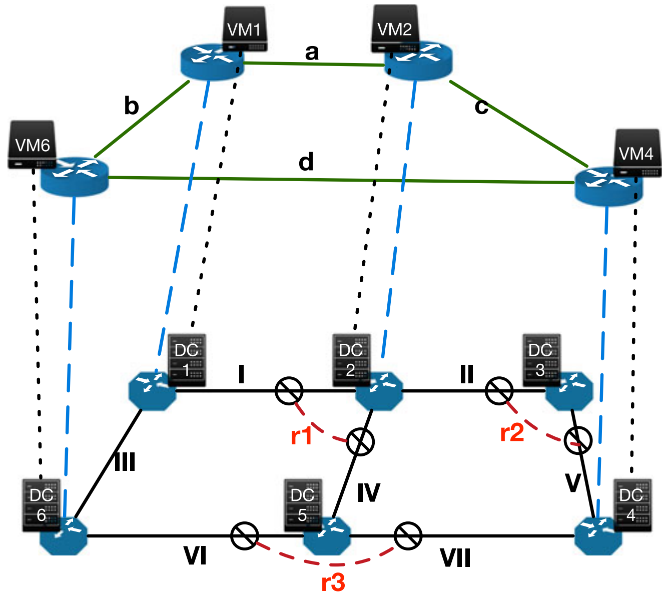

Fig. 2. Survivable cloud network mapping with SRLG failure.

For the virtual datacenter to survive k- and SRLG-failures, at least a spanning tree τ should be available in its residual network after failures. We use Fig. 2 to illustrate an SRLG-survivable cloud network mapping.

Given the SRLG failure set $R_E = \{r_1, r_2, r_3\}$ with $r_1 = \{\text{I}, \text{IV}\}$, $r_2 = \{\text{II}, \text{V}\}$, and $r_3 = \{\text{VI}, \text{VII}\}$. Each virtual machine in the virtual datacenter is mapped onto the physical datacenter with the same index, and the virtual links $\{a, b, c, d\}$ connecting virtual machines mapped onto physical paths $p_a = \{\text{I}\}$, $p_b = \{\text{III}\}$, $p_c = \{\text{II}, \text{V}\}$, and $p_d = \{\text{VI}, \text{VII}\}$. After r_1's failure, a spanning tree $\tau_1 = \{b, c, d\}$ is available in the virtual datacenter, which makes it survivable. With the existence of two other spanning trees $\tau_2 = \{a, b, d\}$ and $\tau_3 = \{a, b, c\}$, the virtual datacenter also remains connected after SRLG r_2 or r_3 failures. Hence, with the given mapping, the virtual datacenter is survivable after any SRLG failure in R_E, and we call spanning trees τ_1, τ_2, τ_3 the *protecting spanning trees*.

III. Necessary and Sufficient Conditions for k- and SRLG-Survivability

We present in this section the necessary and sufficient conditions based on the spanning tree concept for both k- and SRLG-survivability. We also include in Appendix the cutset-based necessary and sufficient conditions.

A. *Survivability Based on Protecting Spanning Trees*

For any $(i, j) \in E_P$, we let $\Lambda(i, j)$ denote the set of links $(s, t) \in E_S$ whose routes pass through (i, j), i.e., $\Lambda(i, j) = \{(s, t) : (i, j) \in p_{st}, (s, t) \in E_S\}$. Let τ be a spanning tree in the virtual datacenter where $\tau \subset G_S$. We now derive the spanning-tree based necessary and sufficient conditions for k- and SRLG-survivability as follows.

Lemma 1: A cloud network mapping is 1-survivable if and only if after the failure of each $(i, j) \in E_P$, there exists at least a spanning tree $\tau \subset G_S$, such that

$$\tau \cap \Lambda(i, j) = \emptyset.$$

This lemma originated from the fact that a mapping is survivable if and only at least a spanning tree is embedded in the virtual datacenter after any single physical link failure (see [39]).

Extending this lemma, we get the following necessary and sufficient conditions for k-survivability.

Theorem 1: A cloud network mapping is k-survivable if and only if there exists at least a spanning tree $\tau \subset G_S$ after arbitrary k physical link failures, such that

$$\tau \bigcap \left(\bigcup_{\beta=1}^{k} \Lambda(i_\beta, j_\beta) \right) = \emptyset,$$

where $(i_1, j_1) \neq (i_2, j_2) \neq \cdots \neq (i_k, j_k)$ and $(i_1, j_1), (i_2, j_2), \cdots, (i_k, j_k) \in E_P$.

Proof: Necessary condition: if after any k-link failures, at least a spanning tree structure is embedded in the residual virtual datacenter, then the virtual datacenter remains connected. Hence, the cloud network mapping is k-survivable.

Sufficient condition: If a cloud network mapping is k-survivable, the virtual datacenter remains connected after any k-link failures. Hence, at least a spanning tree structure exists in the residual virtual datacenter. ■

Corollary 1: A cloud network mapping is SRLG-failure survivable if and only if after any $r \subset R_E$ failure, there exists at least a spanning tree $\tau \subset G_S$, such that

$$\tau \bigcap \left(\bigcup_{(i_\beta^r, j_\beta^r) \in r} \Lambda(i_\beta^r, j_\beta^r) \right) = \emptyset. \qquad (1)$$

The above conclusion is a direct extension of Theorem 1 with the fact that the SRLG failure is a special case of the generalized k-failure scenario.

We may now introduce our solution approaches based on the necessary and sufficient conditions presented in this section.

IV. Solution Approach I: MILP Formulations

In this section, we first present a mathematical programming formulation for the survivable cloud network mapping problem when arbitrary k-link failures occur in the physical cloud infrastructure. We then present a formulation for the case of SRLG failures.

We first introduce the variables used in the MILP formulations. Let y_{ij}^{st} be a variable which represents whether the mapping of $(s,t) \in E_S$ is routed through $(i,j) \in E_P$; if yes, $y_{ij}^{st} = 1$, otherwise $y_{ij}^{st} = 0$. Let $y_{i_1j_1,\cdots,i_kj_k}^{st}$ be a variable which equals 1 if (s,t)'s corresponding route in G_P passes through one or more of the links $(i_1,j_1),\cdots,(i_k,j_k)$, where $(s,t) \in E_S$, $(i_1,j_1) \neq (i_2,j_2) \neq \cdots \neq (i_k,j_k)$ and $(i_\beta,j_\beta) \in E_P$ with $1 \leq \beta \leq k$; otherwise, this variable equals 0. We let μ_{ij}^{st} be a variable that equals 0 if the mapping of link $(s,t) \in E_S$ is routed through $(i,j) \in E_P$; otherwise, $0 < \mu_{ij}^{st} \leq 1$. Let $\mu_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st}$ be a variable which equals 0 if the link $(s,t) \in E_S$ is disconnected after the failure of one or more of the links $(i_1,j_1),\ldots,(i_k,j_k)$, where $(i_1,j_1) \neq (i_2,j_2) \neq \ldots \neq (i_k,j_k)$ and $(i_\beta,j_\beta) \in E_P$ with $1 \leq \beta \leq k$; otherwise, $0 < \mu_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \leq 1$.

To express the condition in Theorem 1 in terms of variable y, we propose the following lemmas which build the connection between links in the virtual datacenter and those in the cloud infrastructure.

Lemma 2: Given $(i_1,j_1),(i_2,j_2) \in E_P, (i_1,j_1) \neq (i_2,j_2)$ and $(s,t) \in E_S$. (s,t) has the property $(s,t) \in \Lambda(i_1,j_1) \cup \Lambda(i_2,j_2)$ if and only if $y_{i_1j_1,i_2j_2}^{st} = 1$, which can be formulated by

$$y_{i_1j_1,i_2j_2}^{st} \geq y_{i_1j_1}^{st} + y_{j_1i_1}^{st}, \tag{2}$$

$$y_{i_1j_1,i_2j_2}^{st} \geq y_{i_2j_2}^{st} + y_{j_2i_2}^{st}, \tag{3}$$

$$y_{i_1j_1,i_2j_2}^{st} \leq y_{i_1j_1}^{st} + y_{j_1i_1}^{st} + y_{i_2j_2}^{st} + y_{j_2i_2}^{st}. \tag{4}$$

Proof of Lemma 2 is given in Appendix C.

Extending Lemma 2, the following lemma shows the property between a single virtual datacenter link and k links in the cloud infrastructure.

Lemma 3: Given $(i_1,j_1),\ldots,(i_k,j_k) \in E_P, (i_1,j_1) \neq (i_2,j_2) \neq \ldots \neq (i_k,j_k)$ and $(s,t) \in E_S$. (s,t) has the property $(s,t) \in \bigcup_{\beta=1}^{k} \Lambda(i_\beta,j_\beta)$ if and only if $y_{i_1j_1,\cdots,i_kj_k}^{st} = 1$, which can be formulated by

$$y_{i_1j_1,\ldots,i_kj_k}^{st} \geq y_{i_1j_1}^{st} + y_{j_1i_1}^{st}, \tag{5}$$

$$\ldots\ldots$$

$$y_{i_1j_1,\ldots,i_kj_k}^{st} \geq y_{i_kj_k}^{st} + y_{j_ki_k}^{st}, \tag{6}$$

$$y_{i_1j_1,\ldots,i_kj_k}^{st} \leq \sum_{\beta=1}^{k} (y_{i_\beta j_\beta}^{st} + y_{j_\beta i_\beta}^{st}). \tag{7}$$

Proof of this lemma is presented in Appendix C.

With Lemma 3, we may now derive the formulations for k- and SRLG-survivable cloud network mapping.

Proposition 1: The relationship between $\mu_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st}$ and $y_{i_1j_1}^{st},\ldots,y_{i_kj_k}^{st}$ can be captured by the following constraints:

$$\mu_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \leq 1 - (y_{i_1j_1}^{st} + y_{j_1i_1}^{st}), \tag{8}$$

$$\mu_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \leq 1 - (y_{i_2j_2}^{st} + y_{j_2i_2}^{st}), \tag{9}$$

$$\ldots\ldots$$

$$\mu_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \leq 1 - (y_{i_kj_k}^{st} + y_{j_ki_k}^{st}), (s,t) \in E_S, \tag{10}$$

$$(i_1,j_1) \neq (i_2,j_2) \neq \cdots \neq (i_k,j_k), (i_\beta,j_\beta) \in E_P, 1 \leq \beta \leq k$$

$$y_{i_\beta j_\beta}^{st} \in \{0,1\}, (s,t) \in E_S, (i_\beta,j_\beta) \in E_P, 1 \leq \beta \leq k. \tag{11}$$

With variables y and μ, the feasible region of y_{ij}^{st}, $Y = \{y_{ij}^{st}$: Constraints (12) and (13), with $i \in V_P\}$, determines a cloud network mapping, where

$$\sum_{(i,j)\in E_P} y_{ij}^{st} - \sum_{(j,i)\in E_P} y_{ji}^{st} = \begin{cases} 1, & \text{if } i = s, \\ -1, & \text{if } i = t, \\ 0, & \text{if } i \neq \{s,t\}, \end{cases} \tag{12}$$

$$y_{ij}^{st} \in \{0,1\}, (s,t) \in E_S, (i,j) \in E_P. \tag{13}$$

Theorem 2: A cloud network mapping is k-survivable against arbitrary k physical link failures, $k \geq 1$, if and only if the following conditions hold.

$$\mu_{i_1j_1,i_2j_2,\cdots,i_kj_k}^{st} \leq 1 - (y_{i_\beta j_\beta}^{st} + y_{j_\beta i_\beta}^{st}), \quad 1 \leq \beta \leq k \tag{14}$$

$$\sum_{(s,t)\in E_S} \mu_{i_1j_1,\ldots,i_kj_k}^{st} - \sum_{(t,s)\in E_S} \mu_{i_1j_1,\ldots,i_kj_k}^{ts} = \begin{cases} -1, & s = v_0, v_0 \in V_S \\ \dfrac{1}{|V_S|-1}, & s \neq v_0, v_0 \in V_S \end{cases} \tag{15}$$

$$0 \leq \mu_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \leq 1, \qquad (s,t) \in E_S \tag{16}$$

Constraint (15) captures the connectivity requirements (existence of a spanning tree).

Hence, the exact solution approach in MILP for the k-survivable cloud network mapping against multiple physical link failures is as follows:

$$\min \sum_{(s,t)\in E_S} \sum_{(i,j)\in E_P} y_{ij}^{st}$$
$$s.t. \text{ Constraints (12) to (16).}$$

Based on Corollary 1 and Theorem 2, we obtain the following theorem which is a special case of Theorem 2 applicable to the SRLG failure case.

Theorem 3: A cloud network mapping is survivable after SRLG failures, if and only if the following conditions are satisfied. For any $r \in R_E$,

$$\mu^{st}_{i^r_1 j^r_1, i^r_2 j^r_2, \ldots, i^r_{k(r)} j^r_{k(r)}} \leq 1 - (y^{st}_{i^r_1 j^r_1} + y^{st}_{j^r_1 i^r_1}) \tag{17}$$

$$\mu^{st}_{i^r_1 j^r_1, i^r_2 j^r_2, \ldots, i^r_{k(r)} j^r_{k(r)}} \leq 1 - (y^{st}_{i^r_2 j^r_2} + y^{st}_{j^r_2 i^r_2}) \tag{18}$$

$$\ldots\ldots$$

$$\mu^{st}_{i^r_1 j^r_1, i^r_2 j^r_2, \ldots, i^r_{k(r)} j^r_{k(r)}} \leq 1 - (y^{st}_{i^r_{k(r)} j^r_{k(r)}} + y^{st}_{j^r_{k(r)} i^r_{k(r)}}) \tag{19}$$

$$\sum_{(s,t)\in E_S} \mu^{st}_{i^r_1 j^r_1, \ldots, i^r_{k(r)} j^r_{k(r)}} - \sum_{(t,s)\in E_S} \mu^{ts}_{i^r_1 j^r_1, \ldots, i^r_{k(r)} j^r_{k(r)}} = \begin{cases} -1, & s = v_0, v_0 \in V_S \\ \dfrac{1}{|V_S| - 1}, & s \neq v_0, v_0 \in V_S \end{cases} \tag{20}$$

$$0 \leq \mu^{st}_{i^r_1 j^r_1, i^r_2 j^r_2, \ldots, i^r_{k(r)} j^r_{k(r)}} \leq 1, \qquad (s,t) \in E_S \tag{21}$$

where v_0 is a selected root node in the virtual datacenter.

Now we have the following MILP formulation for survivable cloud network mapping under SRLG failures which also minimizes the physical link utilization.

$$\min \sum_{(s,t)\in E_S} \sum_{(i,j)\in E_P} y^{st}_{ij}$$

$$s.t. \text{ Constraints (12), (13), (17) to (21)}$$

We wish to note that the cutset-based formulations are provided in Appendix A. The reasons why we chose the tree-based formulation presented above instead of the cutset-based one are that (1) all variables μ are fractional and (2) the total number of binary variables is $|E_P||E_S|$. It can be observed that the cutset-based formulations are (1) with all binary variables and (2) required to enumerate all cutsets, which significantly increase the computational complexity of the MILP formulation.

V. Solution Approach II: Heuristic Algorithm

In this section, we first present the concept of optimal protecting spanning tree collection and how it works for k- and SRLG-survivability. Notations used in the following discussions are as follows. Let $\mathcal{P}$ be the cloud network mapping with $\mathcal{P} = \{p_{st}, (s,t) \in E_S\}$. Let R^k_F indicate the failure set for all k-link failures, and r be an element in R^k_F. For spanning tree $\tau \subset G_S$, we let $E_P(\tau) = E_P \setminus \{\cup_{(s,t)\in\tau} p_{st}\}$ and $R^k_F(\tau) = \{r, r \in R^k_F, r \subseteq E_P \setminus \{\cup_{(s,t)\in\tau} p_{st}\}\}$ denote the physical links and the SRLG failure sets protected by τ, respectively. We let $\mathcal{T} = \{\tau : (s,t) \in \tau \text{ with } p_{st}\}$ be the spanning tree collection in the virtual datacenter.

A. Protecting Spanning Tree Collection and Failed Physical Link Set

For SRLG failures, links in an SRLG fail simultaneously, thus a set of these links is called a failure set. Given an SRLG set R_E, $|R_E|$ represents the total number of elements in R_E and $C^{|E_P|}_k$ is the total number of combinations for arbitrary k-link failures. In this section, we investigate the relation between protecting spanning trees and failure set.

Corollary 2: Given a cloud network and an SRLG set R_E. If a cloud network is SRLG-survivable, there exists at least one spanning tree $\tau \subseteq G_S$ such that $r \in E_P(\tau)$ with $r \in R_E$.

Corollary 2 presents the concept that at least one spanning tree τ which remains connected after the failures of an SRLG $r \in R_E$, thus τ is said to protect all the physical links in r. We may derive the following theorem addressing the maximal number of spanning trees required to protect R_E.

Theorem 4: Given a cloud network and an SRLG set R_E. The sufficient condition to guarantee an SRLG-survivable cloud network design is that there exists up to $|R_E|$ number of spanning trees, where each spanning tree's branch routings only travel through physical links in $E_P \setminus r, r \in R_E$.

Theorem 4 provides an observation that if each SRLG $r \in R_E$ is uniquely protected by one spanning tree, then we need in total $|R_E|$ spanning trees to protect the whole R_E, which is the upper bound. In other words, under this special optimal condition, adding more spanning tree will not help further in terms of survivability. In practice, each selected spanning tree may protect not only one but more SRLGs. Therefore, we will evaluate in Section VI the number of spanning trees required to protect multiple failures.

Following the same concept as in Theorem 4, we may need up to $C^{|E_P|}_k$ spanning trees to protect arbitrary k failures, which is in fact way beyond the number of spanning trees necessary to guarantee survivability. Here we would not further discuss this property in detail but leave it for future study.

B. Heuristic: SRLG-Survivable Protecting Spanning Trees

Based on Theorem 4 and Corollary 2, we present an SRLG-survivable algorithm which sequentially selects protecting spanning trees and generates their corresponding mappings. We also demonstrate its extension to k-survivability.

Let M be a large number. Algorithm 1 starts with generating a minimal spanning tree whose branch mappings are with the minimal costs. The algorithm then evaluates whether existing tree(s) in the spanning tree set can protect at least an SRLG through checking whether mappings of a tree's branches (in $\mathcal{T}$) do not route through links in any SRLG. Then, the algorithm only creates new spanning trees for unprotected SRLGs. We update physical links in a unprotected SRLG set with higher costs (the physical link cost times M). The newly selected spanning tree has the minimal spanning tree cost (based on the newly updated physical link costs). Note here that with the new tree generation, new branch link mappings may be added for the same link in the virtual datacenter. We call it *virtual datacenter augmentation*.

Algorithm 1 can easily be extended to k-survivable cloud network design through constructing the failure sets R_E containing failures r with $r = \{(i_1, j_1), (i_2, j_2), \cdots, (i_k, j_k)\}$, where $(i_1, j_1), (i_2, j_2), \cdots, (i_k, j_k) \in E_P$, and $(i_\ell, j_\ell) \neq (i_\beta, j_\beta)$ and $1 \leq \ell, \beta \leq k$.

VI. Experiment Setting and Simulation Results

We choose the NSF network, illustrated in Fig. 3, as the network of the physical cloud infrastructure. To inspect how the

Algorithm 1: Iterative SRLG-survivable protecting spanning tree algorithm.

```
Input: Given $G_P = (V_P, E_P)$, $G_S = (V_S, E_S)$ and SRLG set $R_E$;
Output: Link mapping for all links in $E_S$, a protecting spanning tree set $\mathcal{T}$, and its protected SRLG sets;
Initialization: set $c_e = 1$ for $e \in E_P$ and $c_u = 1$ for $u \in E_S$; set the protecting spanning tree set $\mathcal{T} = \emptyset$; link route $P_u = \emptyset, u \in E_S$, and vector $m = (0, \cdots, 0)$ indicating protected SRLG set with $m \in \{0,1\}^{1\times|R_E|}$;
for $r \in R_E$ do
    if exists $\tau \in \mathcal{T}$ such that $r \cap (\cup_{u\in\tau}\hat{p_u}) = \emptyset$ then
        for $u \in \tau$ do
            if $P_u$ is $\emptyset$ then
                $P_u = \hat{p}_u$;
    else
        $c_e = c_e \times M$, for all $e \in r$;
        for $u \in E_S$ do
            if $P_u \neq \emptyset$ then
                $c_u = c(P_u)$;
            else
                Generate $\hat{p_u} = \min\{p_u : p_u \in G_P\}$;
                $c_u = c(\hat{p_u})$;
        Generate the minimal spanning tree based on $c_u$: $\hat{\tau} = \min\{\sum_{u\in\tau} c_u, \tau \in G_S\}$;
        $\mathcal{T} = \mathcal{T} \bigcup \{\hat{\tau}\}$;
        Reset link cost: $c_e = 1$ for $e \in E_P$, $c_u = 1$ if $u \in \cup_{\tau\in\mathcal{T}}\tau$; otherwise, $c_u = c_u \times M$.
```

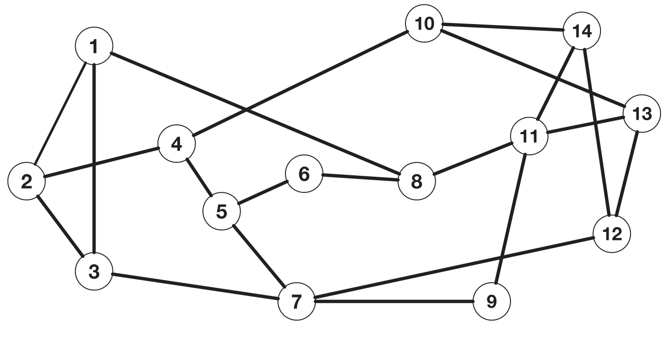

Fig. 3. NSF network.

nodal degree would affect the performance of our algorithms, we also create "NSF$^{(1)}$" which denotes the NSF network augmented with link $(6, 9)$. For the virtual datacenter networks, we randomly generate two 3-edge connected networks denoted as "CLN1" and "CLN3" and illustrated in Fig. 4 in solid lines. For the same rationale, we also augment "CLN1" and "CLN3" into 4-edge connected networks with extra links illustrated in dashed lines, which create "CLN2" and "CLN4" networks, respectively. The mapping of the nodes from virtual datacenter to the physical cloud infrastructure is presented through the same node indices printed in the nodes.

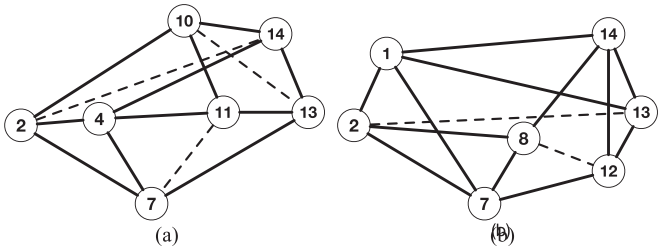

Fig. 4. Virtual datacenter networks.

TABLE II
LOGICAL TOPOLOGIES INFORMATION

	Conn	Nodes	Edges	MinDeg	MaxDeg	AvgDeg
CLN 1	3	7	11	3	4	3.14
CLN 2	4	7	14	4	4	4
CLN 3	3	7	11	3	4	3.29
CLN 4	4	7	14	4	4	4
NSF	2	14	21	2	4	3
NSF$^{(1)}$	3	14	22	3	4	3.14

Table II provides the information of all the network topologies mentioned above, where "Conn", "Nodes", "Edges", "minDeg", "maxDeg", and "AvgDeg" represent the edge-connectivity, number of nodes/edges, minimal, maximal, and average node degrees of the networks.

In practice, the SRLG set is known a priori. We generate a SRLG failure set called the 3-SRLG set, which has the following properties: (1) each SRLG has three or fewer edges, (2) each SRLG is not a subset of another SRLG, (3) each SRLG failure does not disconnect the physical network, and (4) the union of all SRLGs covers the entire physical network [40]. We also consider the k-failure scenario where $k = 2$ due to the connectivity of our physical networks.

We use two metrics to validate and evaluate the performance of our proposed spanning-tree based approaches: (1) full survivability and maximal partial survivability; and (2) minimal physical resources utilized to guarantee full survivability or maximal partial survivability. Note here that if a survivable mapping does not exist due to the limitation of the given networks (mainly because of their edge-connectivity), our approach would generate a mapping which guarantees the connectivity of the virtual datacenter against the most number of failure scenarios.

We report our simulation results of the proposed MILP approach in Tables III and IV for the SRLG- and 2-failure scenarios, respectively. Out of the seven-element 3-SRLG set, we test our formulation over five, six, and all seven 3-SRLG failures to verify how many of the 3-SRLG failures can be protected by the generated mapping. If not all of them can be survivable, we report the maximal number of the 3-SRLGs which do not disconnect the virtual datacenter. Let "Surv", "MaxS", "PhyS" represent the existence of survivable cloud network mapping for the tested instances, maximal number of survivable SRLGs, and the minimal number of physical links used in the routings. Let $SIdx$ be the survivability index which shows the number

TABLE III
MILP RESULTS FOR 3-SRLG FAILURES

		Five 3-SRLGs			Six 3-SRLGs			Seven 3-SRLGs		
		Surv	MaxS	PhyS	Surv	MaxS	PhyS	Surv	MaxS	PhyS
NSF	CLN 1	Yes	5	22	Yes	6	22	No	6	22
	CLN 2	Yes	5	27	Yes	6	27	No	6	26
	CLN 3	Yes	5	26	Yes	6	26	No	6	24
	CLN 4	Yes	5	31	Yes	6	31	No	6	29
NSF$^{(1)}$	CLN 1	Yes	5	21	Yes	6	21	Yes	7	21
	CLN 2	Yes	5	26	Yes	6	26	Yes	7	26
	CLN 3	Yes	5	23	Yes	6	24	Yes	7	24
	CLN 4	Yes	5	29	Yes	6	29	Yes	7	29

TABLE IV
MILP RESULTS FOR 2-LINK FAILURES

	NSF			NSF$^{(1)}$		
	Surv	SIdx	PhyS	Surv	SIdx	PhyS
CLN 1	No	209	30	No	230	37
CLN 2	Yes	210	40	Yes	231	40
CLN 3	Yes	210	31	Yes	231	29
CLN 4	Yes	210	38	Yes	231	37

TABLE V
ALGORITHM 1 RESULTS FOR 2-LINK FAILURE

	NSF			NSF$^{(1)}$		
	PrctFSet	PrctPct	Time(s)	PrctFSet	PrctPct	Time(s)
CLN1	-/210	-	-	-/231	-	-
CLN2	200/210	95.24%	13.95	221/231	95.67%	13.86
CLN3	182/210	86.67%	12.98	202/231	87.01%	13.51
CLN4	209/210	99.52%	4.45	228/231	98.70%	5.83

TABLE VI
ALGORITHM 1 RESULTS WITH FIVE 3-SRLG SETS

		Surv	Augment#	MaxS	PhyS	Tree#	LogS	Time(s)
NSF	CLN1	Yes	0	5	22	4	10	0.078
	CLN2	Yes	0	5	27	4	9	0.140
	CLN3	Yes	0	5	27	5	11	0.140
	CLN4	Yes	0	5	31	5	11	0.062
NSF$^{(1)}$	CLN1	Yes	0	5	21	5	10	0.124
	CLN2	Yes	0	5	26	4	9	0.078
	CLN3	Yes	0	5	23	5	11	0.102
	CLN4	Yes	0	5	29	5	10	0.156

of arbitrary physical link pairs whose failures do not disconnect the virtual datacenter.

The results in Table III show that with the same physical network, the routings of CLN2 and CLN4 consume more physical links than that of CLN1 and CLN3, which results from the more number of virtual links and higher connectivity in CLN2 and CLN4. On the other hand, with the same virtual network, the routing consumes more physical links over NSF as the physical network than that with NSF$^{(1)}$, which corresponds to the fact that NSF network has fewer physical links and lower connectivity than NSF$^{(1)}$. At least for all testing cases, the connectivity of both virtual and physical networks impact the utilization of physical link consumption in routes of virtual links. In a cross-layer network, higher virtual network connectivity and lower physical network connectivity lead to more physical link consumption. It is not surprising that when the physical network has lower connectivity, the survivable virtual link routings generated may require longer physical paths. Similarly, with the same physical network, the number of routings to be generated for a highly-connected virtual network is larger than those with lower connectivity.

With Table IV, we observe that with the same physical network and CLN1 as virtual network, there is no survivable design for the cross-layer network. Meanwhile, CLN1 is with the smallest average node degree compared with CLN2, CLN3, and CLN4. On the other hand, the node mapping between virtual network and physical network are different between CLN1-2 and CLN3-4. The shortest-path routings between (2, 4) and (2, 7) may share some joint part as physical nodes 2, 4, and 7 are mapped by CLN1-2 concurrently. The mapping may lead to less physical link consumption of CLN3-4 than CLN1-2. At least, for our testing cases, the lower average virtual node degree leads to more unsurvivable cross-layer topology. Meanwhile, the node mapping between virtual network and physical network impacts the physical link consumption for virtual link routing.

As shown in Table IV, with NSF as the physical network, our approach can generate survivable mappings for all virtual datacenter networks against five and six 3-SRLG failures. But none of them could produce survivable cloud mappings with seven 3-SRLGs. After augmenting NSF to NSF$^{(1)}$, all virtual datacenter networks remain connected against all five, six, and seven 3-SRLG failures, which shows that SRLG-survivability is sensitive to the physical node degrees. Meanwhile, it can be observed that the utilization of physical links decreases for some tested instances when increasing the number of 3-SRLGs. Though it seems counterintuitive, it actually reflects that the objective of the MILP formulations helps choose shorter routings effectively while maintaining the (maximal) survivability. All tested cases are solved within 1469 seconds.

We now evaluate the performance of our proposed Algorithm 1 based on (1) whether the algorithm can provide SRLG-survivable mapping, (2) how many SRLGs can be protected if a survivable mapping does not exist, and (3) whether cloud network link augmentation should be applied to provide better protection. The results are presented in Tables V, VI, VII, and VIII, where "Augment#", "Tree#", "LogS" and "Time" denote the number of augmented logical links to produce a survivable cloud mapping, the number of spanning trees generated in the virtual datacenter, the number of links utilized in the generated spanning tree set, and the computational time (in second), respectively. We illustrate the number of physical link utilization by MILP and heuristic approaches for 5, 6, and 7 SRLG sets in Figs. 5–7.

We present the performance of Algorithm 1 for 2-link failure scenario in Table V. With NSF and NSF$^{(1)}$ as the physical

TABLE VII
ALGORITHM 1 RESULTS WITH SIX 3-SRLG SETS

		Surv	Augment#	MaxS	PhyS	Tree#	LogS	Time(s)
NSF	CLN1	Yes	0	6	22	5	10	0.100
	CLN2	Yes	0	6	27	5	10	0.120
	CLN3	Yes	0	6	27	6	11	0.083
	CLN4	Yes	0	6	32	6	11	0.112
NSF$^{(1)}$	CLN1	Yes	0	6	21	5	10	0.124
	CLN2	Yes	0	6	26	5	9	0.078
	CLN3	Yes	0	6	24	5	11	0.096
	CLN4	Yes	0	6	30	5	11	0.113

TABLE VIII
ALGORITHM 1 RESULTS WITH SEVEN 3-SRLG SETS

		Surv	Augment#	MaxS	PhyS	Tree#	LogS	Time(s)
NSF	CLN1	No	1	5	25	5	10	0.085
	CLN2	No	0	6	27	6	10	0.093
	CLN3	No	0	6	28	6	12	0.110
	CLN4	No	0	6	32	6	13	0.123
NSF$^{(1)}$	CLN1	No	1	6	24	6	10	0.081
	CLN2	Yes	0	7	26	6	10	0.097
	CLN3	Yes	0	7	24	5	11	0.116
	CLN4	Yes	0	7	30	5	11	0.146

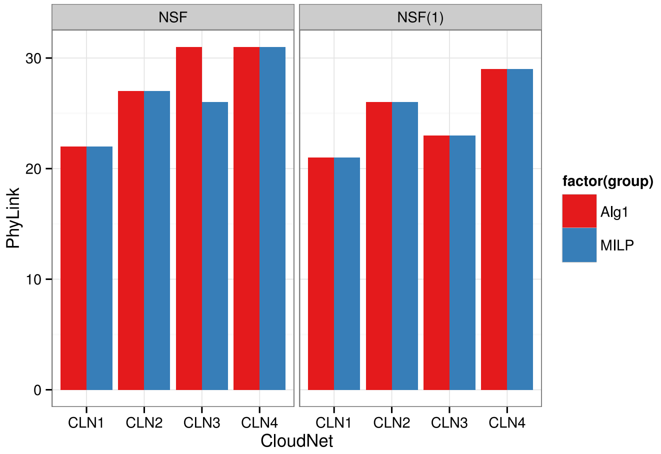

Fig. 5. Performance comparison between Algorithm 1 and MILP with 5 SRLG sets.

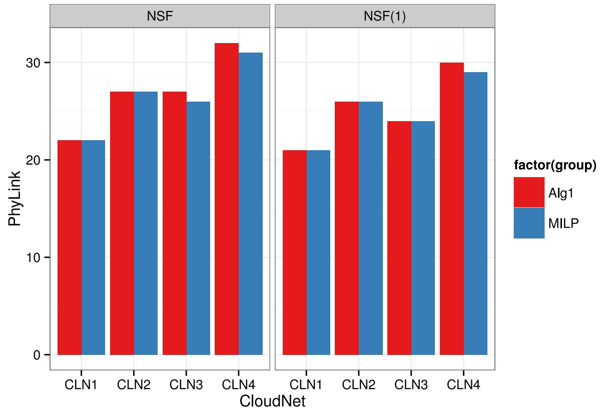

Fig. 6. Performance comparison between Algorithm 1 and MILP with 6 SRLG sets.

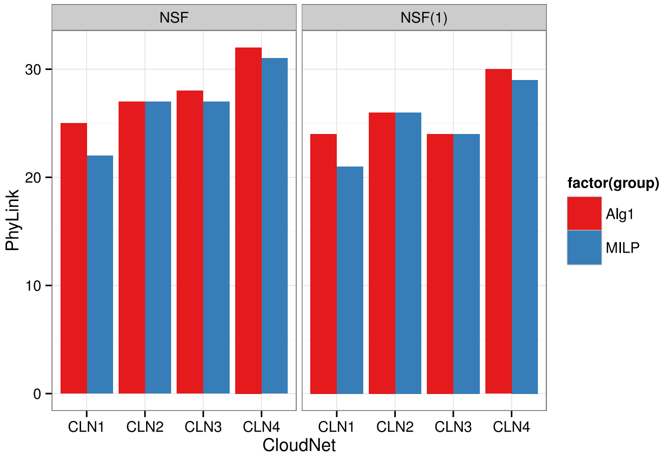

Fig. 7. Performance comparison between Algorithm 1 and MILP with 7 SRLG sets.

networks, the total number of 2-link failures is 210 and 231, respectively. Let "PrctFSet" represent the number of protected failure set and "PrctPct" as the percentage of the number of protected sets to the total number of failure sets. The two numbers in column "PrctFSet" represent the number of protected set followed by the total number of failure sets. The best performance achieved by our algorithm is with NSF and CLN4 as the physical and virtual networks. The average protected percentage for all these testing cases are 93.81%.

There is no surprise that the MILP approach's computation time is much longer than that of the heuristic algorithm. The average computation time with our MILP approach for survivable routing with 3-SRLG failure and 2-failure are 11.93 and 194.88 seconds, respectively. Among all testing cases, the longest computation time is 852.01 seconds (2-failure, with NSF and CLN2 as the physical and logical networks, respectively). As shown in Tables VI–VIII, the computation time for our heuristic algorithm over all test cases is less than 0.16 seconds.

It can be observed in Tables VI to VIII that: (1) if SRLG-survivable cloud network mapping exists, the results produced by Algorithm 1 are as good as the ones generated by the MILP formulation, (2) the number physical links utilized in the routings produced by Algorithm 1 is the same or close to the results of MILP formulation, and (3) for a physical network with higher node degree, the generated cloud network mappings seem to utilize less number of links both in the physical cloud infrastructure and the virtual datacenter to construct all protecting spanning trees and their routings.

As expected, Algorithm 1 cannot generate the survivable cloud network mappings for all instances in Table VIII, which triggers the augmentation of link(s) in the virtual datacenter network. We wish to note here that we only augment CLN1 with both NSF and NSF$^{(1)}$ as the physical networks because (1) we want to do both vertical (between NSF and NSF$^{(1)}$) and horizontal (between Algorithm and MILP) comparisons, and (2) we want to show how many 3-SRLGs can be protected without augmentation. As seen in Table VIII, though not all mappings generated by the heuristic are survivable, they still can protect majority number of 3-SRLG failures.

VII. Conclusion

In this paper, we studied the SRLG- and k-survivable cloud network mapping problems and discussed how the concept of the spanning tree can be used to guarantee survivability. Based on the spanning tree concept, we proposed the necessary and sufficient conditions and the corresponding MILP formulations, which are applied in this paper as the general framework to handle SRLG and k failures in a cloud network. In addition to the mathematical formulation, we designed a heuristic algorithm also based on the spanning tree concept. We compared the performance of the simulation results generated by both the MILP formulation and heuristic algorithm, which showed that our proposed Algorithm 1 can effectively generate SRLG-survivable cloud network mappings with quality similar to the optimal solution generated by the MILP formulation. Also, the difference of physical resource utilization between Algorithm 1 and the optimal solution is within 5% in average and within 17% for all tested cases.

Appendix

This appendix includes three parts. We first present the cutset-based cloud network mapping formulation for k- and SRLG-survivability, which is the counterpart of our spanning-tree based approach. We also provide their corresponding MILP formulations. The second part provides the comparisons of complexity for the cutset- and spanning-tree-based approaches in terms of the number of formulations generated. The last one provides the proofs for Lemmas 2 and 3.

A. Cutset-Based Necessary and Sufficient Conditions for Survivable Cloud Network Mapping

Let χ be a cutset in the virtual datacenter network, $\chi \subset G_S$.

Lemma 4: A cloud network mapping is 1-survivable if and only if for each $(i,j) \in E_P$,

$$|\chi \cap \Lambda(i,j)| \leq |\chi| - 1, \quad \text{for all cutsets } \chi \subset G_S,$$

where $|\cdot|$ represents the cardinality of the set. This lemma follows the fact that a mapping is survivable if and only if any (i,j) failure would not disconnect all links in every single cutset in the virtual datacenter.

Extending this lemma, we get the following necessary and sufficient conditions for k-survivability.

Theorem 5: A cloud network mapping is k-survivable if and only if after arbitrary k physical link failures,

$$\left|\chi \bigcap \left(\bigcup_{\beta=1}^{k} \Lambda(i_\beta, j_\beta) \right) \right| \leq |\chi| - 1, \ \ \forall \chi \subset G_S, \tag{22}$$

where $(i_1, j_1) \neq (i_2, j_2) \neq \cdots \neq (i_k, j_k)$ and $(i_1, j_1), (i_2, j_2), \cdots, (i_k, j_k) \in E_P$.

Proof: Necessary condition: after any k-link failures, if all cuts are not fully disconnected (at least an edge in the cut remains connected), the virtual datacenter remains connected. Hence, the cloud network mapping is $k-$survivable.

Sufficient condition: If a cloud network is $k-$survivable, the virtual datacenter is connected after any k-link failures, hence, no cut is fully disconnected. ■

Corollary 3: A cloud network mapping is SRLG-survivable if and only if after any $r \in R_E$ failure,

$$\left|\chi \bigcap \left(\bigcup_{(i_\beta^r, j_\beta^r) \in r} \Lambda(i_\beta^r, j_\beta^r) \right) \right| \leq |\chi| - 1, \\ \text{for all cutsets } \chi \subset G_S. \tag{23}$$

The corollary above is a direct extension of Theorem 5.

Based on Theorem 5 and Lemma 3, we provide another MILP formulation which guarantees that the cloud network mapping is k-survivable. Following Lemma 3, the formulation determines the relationship between $y_{i_1 j_1, \ldots, i_k j_k}^{st}$ and $y_{i_\beta j_\beta}^{st}$, $1 \leq \beta \leq k$, which indicates whether (s,t) is routed through links in a subset of $\{(i_1, j_1), \cdots, (i_k, j_k)\}$.

Proposition 2: A cloud network mapping is k-survivable, $k \geq 1$, if and only if the following condition holds:

$$\sum_{(s,t) \in \chi} y_{i_1 j_1, \cdots, i_k j_k}^{st} \leq |\chi| - 1,$$

$$\chi \subset G_S, \ (i_1, j_1), \cdots, (i_k, j_k) \in E_P,$$

$$\text{and } (i_q, j_q) \neq (i_\ell, j_\ell), \ 1 \leq q, \ell \leq k, \tag{24}$$

$$y_{i_1 j_1, \cdots, i_k, j_k}^{st} \in \{0, 1\}. \tag{25}$$

It is easy to verify that $\sum_{(s,t) \in \chi} y_{i_1 j_1, \cdots, i_k j_k}^{st} = |\chi \bigcap \bigcup_{\beta-1}^{k} \Lambda(i_\beta, j_\beta)|$ based on the definition of $y_{i_1 j_1, \cdots, i_k j_k}^{st}$, when $\{(i_1, j_1), \cdots, (i_k, j_k)\}$ is a k-link failure.

We provide the ILP formulation for $k-$survivability based on the cutset concept as follows:

$$\min \sum_{(s,t) \in E_S} \sum_{(i,j) \in E_P} y_{st}^{ij}$$

s.t. Constraint (2) to (4), (12), (13), (24) and (25).

Proposition 3: A cloud network mapping is survivable after SRLG failures if and only if the following conditions hold: for any $r \in R_E$,

constraint (24),

$$y_{i_1^r j_1^r, i_2^r j_2^r, \ldots, i_{k(r)}^r j_{k(r)}^r}^{st} \geq (y_{i_1^r j_1^r}^{st} + y_{j_1^r i_1^r}^{st}), \tag{26}$$

$$y_{i_1^r j_1^r, i_2^r j_2^r, \ldots, i_{k(r)}^r j_{k(r)}^r}^{st} \geq (y_{i_2^r j_2^r}^{st} + y_{j_2^r i_2^r}^{st}), \tag{27}$$

$$\ldots\ldots$$

$$y_{i_1^r j_1^r, i_2^r j_2^r, \ldots, i_{k(r)}^r j_{k(r)}^r}^{st} \geq (y_{i_{k(r)}^r j_{k(r)}^r}^{st} + y_{j_{k(r)}^r i_{k(r)}^r}^{st}), \tag{28}$$

$$y_{i_1^r j_1^r, i_2^r j_2^r, \ldots, i_{k(r)}^r j_{k(r)}^r}^{st} \in \{0, 1\}, \qquad (s,t) \in E_S. \tag{29}$$

TABLE IX
COMPLEXITY OF MILP FORMULATIONS WITH NUMBER OF VARIABLES AND CONSTRAINTS

	Model	SRLG	k-survivable
Vrbs	Tree	$\mathcal{O}[\lvert E_S\rvert(\lvert R_E\rvert + \lvert E_P\rvert)]$	$\mathcal{O}[\lvert E_S\rvert(C_k^{\lvert E_P\rvert} + \lvert E_P\rvert)]$
	Cutset	$\mathcal{O}[\lvert E_S\rvert(\lvert R_E\rvert + \lvert E_P\rvert)]$	$\mathcal{O}[\lvert E_S\rvert(C_k^{\lvert E_P\rvert} + \lvert E_P\rvert)]$
Cnts	Tree	$\mathcal{O}[\lvert E_S\rvert\lvert V_P\rvert + \lvert R_E\rvert$ $(\lvert E_S\rvert + \lvert V_S\rvert)]$	$\mathcal{O}[\lvert E_S\rvert\lvert V_P\rvert + C_k^{\lvert E_P\rvert}$ $(\lvert E_S\rvert + \lvert V_S\rvert)]$
	Cutset	$\mathcal{O}[\lvert E_S\rvert\lvert V_P\rvert + \lvert R_E\rvert$ $(\lvert E_S\rvert + \lvert V_S\rvert + 2^{\lvert V_S\rvert})]$	$\mathcal{O}[\lvert E_S\rvert\lvert V_P\rvert + C_k^{\lvert E_P\rvert}$ $(\lvert E_S\rvert + \lvert V_S\rvert + 2^{\lvert V_S\rvert})]$

The ILP formulation for SRLG-survivability based on the concept of cutset is as follows:

$$\min \sum_{(s,t)\in E_S} \sum_{(i,j)\in E_P} y_{ij}^{st}$$

s.t. Constraint (2) to (4), (12), (13), and (24) to (29).

B. Complexity Comparision of the Spanning-Tree and Cutset Based Approaches

The complexity of MILP models for both spanning-tree and cutset-based formulations in terms of the numbers of variables and constraints for SRLG- and k-survivability is shown in Table IX. Let "Tree" and "Cutset" represent the spanning-tree and cutset based MILP formulations, respectively. Here $|V_P|$, $|E_P|$, $|V_S|$, $|E_S|$ represent the node and edge numbers in the physical cloud infrastructure and virtual datacenter, respectively. We let $r_{\max}$ be an SRLG set with the maximum number of physical edges in R_E and N represent the number of cutsets in G_S.

We observe that the complexity of the formulation depends mainly on the number of failure scenarios. Given an SRLG set, the complexity of the formulation is bounded by the cardinality of the SRLG set, the number of physical links in an SRLG set, and the size of the physical infrastructure and virtual datacenter. But for the generalized k-failure problem, the complexity of the formulation increases exponentially with the number of failed links. Comparing the complexity of tree-based and cutset-based MILP formulations, the number of variables are the same for both formulations, but the tree-based formulation uses much less binary variables. The number of constraints in the cutset-based model is significantly more than that based on spanning trees because the cutset-based formulation needs to enumerate all cutsets in the physical network, which significantly increases the number of constraints. In general, it takes longer time to compute the optimal result for an MILP formulation with more binary variables and constraints. Hence, in this paper our focus is mainly on the MILP formulation and heuristic based on the spanning tree concept.

C. Proofs

Proof of Lemma 2:

Proof: Necessary condition: given a $(s,t) \in E_S$ and two physical links $(i_1, j_1), (i_2, j_2) \in E_P$, one of the following four cases occurs: (i) $(s,t) \in \Lambda(i_1, j_1) \cap \overline{\Lambda}(i_2, j_2)$; (ii) $(s,t) \in \overline{\Lambda}(i_1, j_1) \cap \Lambda(i_2, j_2)$; (iii)$(s,t) \in \Lambda(i_1, j_1) \cap \Lambda(i_2, j_2)$; or (iv) $(s,t) \in \overline{\Lambda}(i_1, j_1) \cap \overline{\Lambda}(i_2, j_2)$. For case (i), $y_{i_1j_1}^{st} + y_{j_1i_1}^{st} = 1$, so $y_{i_1j_1,i_2j_2}^{st} = 1$; for case (ii), $y_{i_2j_2}^{st} + y_{j_2i_2}^{st} = 1$, then, $y_{i_1j_1,i_2j_2}^{st} = 1$; for case (iii), $y_{i_1j_1}^{st} + y_{j_1i_1}^{st} = 1$ and $y_{i_2j_2}^{st} + y_{j_2i_2}^{st} = 1$, then, $y_{i_1j_1,i_2j_2}^{st} = 1$; for case (iv), $y_{i_1j_1}^{st} = y_{j_1i_1}^{st} = 0$ and $y_{i_2j_2}^{st} = y_{j_2i_2}^{st} = 0$, then, $y_{i_1j_1,i_2j_2}^{st} = 0$. In all cases, constraints (2) to (4) hold.

Sufficient condition: with constraints (2)–(4), if either $y_{i_1j_1}^{st} + y_{j_1i_1}^{st} = 1$ or $y_{i_2j_2}^{st} + y_{j_2i_2}^{st} = 1$, then, $y_{i_1j_1,i_2j_2}^{st} = 1$, which implies that $(s,t) \in \Lambda(i_1, j_1) \cup \Lambda(i_2, j_2)$. Meanwhile, if $y_{i_1j_1}^{st} = y_{j_1i_1}^{st} = y_{i_2j_2}^{st} = y_{j_2i_2}^{st} = 0$, then, $y_{i_1j_1,i_2j_2}^{st} = 0$, which implies that $(s,t) \in \overline{\Lambda}(i_1, j_1) \cap \overline{\Lambda}(i_2, j_2)$. ■

Proof of Lemma 3:

Proof: We prove this conclusion by induction for both necessary and sufficient conditions.

Necessary condition: with Lemma 2, if $\beta = 2$, the conclusion holds. We assume that if $\beta = k - 1$, the conclusion holds. With $\beta = k$, four cases occur with $[(i_1, j_1), \cdots, (i_{k-1}, j_{k-1})]$ and (i_k, j_k) failures: (i) (s,t)'s mapping routes through (i_k, j_k) only; (ii) (s,t)'s mapping routes through all or part of $[(i_1, j_1), \cdots, (i_{k-1}, j_{k-1})]$; (iii) (s,t)'s mapping routes through all or part of $[(i_1, j_1), \cdots, (i_{k-1}, j_{k-1})]$ and (i_k, j_k); and (iv) (s,t)'s mapping routes though none of $\{(i_q, j_q)$ with $1 \leq q \leq k\}$. For case (i), $y_{i_kj_k}^{st} = 1$, so $y_{i_1j_1,\cdots,i_kj_k}^{st} = 1$. For case (ii), with $q = k - 1$, there exists some z $(1 \leq z \leq k - 1)$, such that $y_{i_zj_z}^{st} + y_{j_zi_z}^{st} = 1$, so $y_{i_1j_1,\cdots,i_{k-1}j_{k-1}}^{st} = 1$. Hence, $y_{i_1j_1,\cdots,i_kj_k}^{st} = 1$. For case (iii), if both $y_{i_kj_k}^{st} = 1$ and $y_{i_1j_1,\cdots,i_{k-1}j_{k-1}}^{st} = 1$, $y_{i_1j_1,\cdots,i_kj_k}^{st} = 1$. These three cases imply that constraints (5)–(6) hold. For case (iv), if $y_{i_1j_1}^{st} = y_{j_1i_1}^{st} = \cdots = y_{i_kj_k}^{st} = y_{j_ki_k}^{st} = 0$, then $y_{i_1j_1,\cdots,y_kj_k}^{st} = 0$, which leads to constraint (7).

Sufficient condition: if $\beta = 2$, the conclusion holds with Lemma 3. We assume that the conclusion holds when $\beta = k - 1$. Now we prove that the conclusion holds when $\beta = k$. Here, only two cases may occur: (i) $(s,t) \in \Lambda(i_k, j_k)$, and (ii) $(s,t) \in \bar{\Lambda}(i_k, j_k)$. For case (i), with $y_{i_1j_1,\cdots,i_kj_k}^{st} \geq y_{i_kj_k}^{st} + y_{j_ki_k}^{st}$, it implies $(s,t) \in \cup_{\kappa=1}^{k} \Lambda(i_\kappa, j_\kappa)$. For case (ii), with $\beta = k - 1$, if constraints (5)–(6) lead to $(s,t) \in \cup_{\kappa=1}^{k-1} \Lambda(i_\kappa, j_\kappa)$, then $(s,t) \in \cup_{q=1}^{k} \Lambda(i_q, j_q)$; otherwise, $y_{i_1j_1,\cdots,i_{k-1}j_{k-1}}^{st} \leq \sum_{q=1}^{k-1} y_{i_qj_q}^{st} + y_{j_qi_q}^{st}$ which implies $(s,t) \in \cap_{q=1}^{k-1} \overline{\Lambda}(i_q, j_q)$. With $y_{i_1j_1,\cdots,i_{k-1}j_{k-1}}^{st} \leq \sum_{q=1}^{k-1} y_{i_qj_q}^{st} + y_{j_qi_q}^{st} + y_{i_kj_k}^{st} + y_{j_ki_k}^{st}$, it implies $(s,t) \in \cap_{q=1}^{k} \overline{\Lambda}(i_q, j_q)$. Hence, the conclusion holds when $\beta = k$.

REFERENCES

[1] C. Develder *et al.*, "Optical networks for grid and cloud computing applications," *Proc. IEEE*, vol. 100, no. 5, pp. 1149–1167, May 2012.

[2] N. Chowdhury, M. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM*, 2009, pp. 783–791.

[3] C. Colman-Meixner, F. Dikbiyik, M. F. Habib, M. Tornatore, C.-N. Chuah, and B. Mukherjee, "Disaster-survivable cloud-network mapping," *Photonic Netw. Commun.*, vol. 27, no. 3, pp. 141–153, 2014.

[4] M. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks," *J. Lightw. Technol.*, vol. 30, no. 16, pp. 2563–2573, Aug. 2012.

[5] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightw. Technol.*, vol. 21, no. 4, pp. 870–883, Apr. 2003.

[6] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, “Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures,” in *Proc. IEEE Global Telecommun. Conf.*, 2010, pp. 1–6.

[7] M. R. Rahman and R. Boutaba, “SVNE: Survivable virtual network embedding algorithms for network virtualization,” *IEEE Trans. Netw. Serv. Manage.*, vol. 10, no. 2, pp. 105–118, Jun. 2013.

[8] O. Soualah, I. Fajjari, N. Aitsaadi, and A. Mellouk, “PR-VNE: Preventive reliable virtual network embedding algorithm in cloud’s network,” in *Proc. 2013 IEEE Global Commun. Conf.*, 2013, pp. 1303–1309.

[9] S. Seetharaman, A. Durresi, R. Jagannathan, R. Jain, N. Chandhok, and K. Vinodkrishnan, “Ip over optical networks: A summary of issues,” 2001. [Online]. Available: https://tools.ietf.org/html/draft-osu-ipo-mpls-issues-02

[10] M. R. Rahman, I. Aib, and R. Boutaba, “Survivable virtual network embedding,” in *Proc. Int. Conf. Res. Netw.*, 2010, pp. 40–52.

[11] 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,” *J. Opt. Commun. Netw.*, vol. 6, no. 2, pp. 190–203, 2014.

[12] T. Guo, N. Wang, K. Moessner, and R. Tafazolli, “Shared backup network provision for virtual network embedding,” in *Proc. IEEE Int. Conf. Commun.*, 2011, pp. 1–5.

[13] Y. Chen, J. Li, T. Wo, C. Hu, and W. Liu, “Resilient virtual network service provision in network virtualization environments,” in *Proc. 2010 IEEE 16th Int. Conf. Parallel Distrib. Syst.*, 2010, pp. 51–58.

[14] C. Qiao, B. Guo, S. Huang, J. Wang, T. Wang, and W. Gu, “A novel two-step approach to surviving facility failures,” in *Proc. Opt. Fiber Commun. Conf*, 2011, Paper OWAA3.

[15] X. Liu, C. Qiao, and T. Wang, “Robust application specific and agile private (ASAP) networks withstanding multi-layer failures,” in *Proc. Opt. Fiber Commun. Conf.*, 2009, Paper OWY1.

[16] W.-L. Yeow, C. Westphal, and U. C. Kozat, “Designing and embedding reliable virtual infrastructures,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 57–64, 2011.

[17] B. Guo *et al.*, “Survivable virtual network design and embedding to survive a facility node failure,” *J. Lightw. Technol.*, vol. 32, no. 3, pp. 483–493, Feb. 2014.

[18] G. Sun, V. Anand, D. Liao, C. Lu, X. Zhang, and N.-H. Bao, “Power-efficient provisioning for online virtual network requests in cloud-based data centers,” *IEEE Syst. J.*, vol. 9, no. 2, pp. 427–441, Jun. 2015.

[19] S. R. Chowdhury *et al.*, “Protecting virtual networks with drone,” in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, 2016, pp. 78–86.

[20] K. Lee, H.-W. Lee, and E. Modiano, “Reliability in layered networks with random link failures,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1835–1848, Dec. 2011.

[21] M. P. Anastasopoulos, A. Tzanakaki, and D. Simeonidou, “Stochastic planning of dependable virtual infrastructures over optical datacenter networks,” *J. Opt. Commun. Netw.*, vol. 5, no. 9, pp. 968–979, 2013.

[22] G. Luo *et al.*, “Survivable virtual optical network embedding with probabilistic network-element failures in elastic optical networks,” in *Proc. 2014 13th Int. Conf. Opt. Commun. Netw.*, 2014, pp. 1–4.

[23] C. C. Meixner, F. Dikbiyik, M. Tornatore, C.-N. Chuah, and B. Mukherjee, “Disaster-resilient virtual-network mapping and adaptation in optical networks,” in *Proc. 2013 17th Int. Conf. Opt. Netw. Des. Model.*, 2013, pp. 112–107.

[24] F. Dikbiyik, M. Tornatore, and B. Mukherjee, “Minimizing the risk from disaster failures in optical backbone networks,” *J. Lightw. Technol.*, vol. 32, no. 18, pp. 3175–3183, Sep. 2014.

[25] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.

[26] M. Kurant and P. Thiran, “On survivable routing of mesh topologies in IP-over-WDM network,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2005, pp. 1106–1116.

[27] 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. 4th Int. Congr. Ultra Modern Telecommun. Control Syst.*, 2012, pp. 650–656.

[28] Z. Zhou, T. Lin, and K. Thulasiraman, “Survivable cloud network mapping with multiple failures,” in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 7113–7118.

[29] E. Modiano and A. Narula-Tam, “Survivable routing of logical topologies in WDM networks,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2001, pp. 348–357.

[30] A. Todimala and B. Ramamurthy, “A scalable approach for survivable virtual topology routing in optical WDM networks,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 63–69, Aug. 2007.

[31] B. Jaumard, A. H. Hoang, and M. N. Bui, “Path vs. cutset approaches for the design of logical survivable topologies,” in *Proc. 2012 IEEE Int. Conf. Commun.*, 2012, pp. 3061–3066.

[32] M. Parandehgheibi, H.-W. Lee, and E. Modiano, “Survivable path sets: A new approach to survivability in multilayer networks,” *J. Lightw. Technol.*, vol. 32, no. 24, pp. 4139–4150, Dec. 2014.

[33] K. Xi, H. Chao, and C. Guo, “Recovery from shared risk link group failures using IP fast reroute,” in *Proc. IEEE 19th Int. Conf. Comput. Commun. Netw.*, 2010, pp. 1–7.

[34] M. Pourvali *et al.*, “Progressive recovery for network virtualization after large-scale disasters,” in *Proc. 2016 Int. Conf. Comput., Netw. Commun.*, 2016, pp. 1–5.

[35] M. Bui, B. Jaumard, and C. Develder, “Anycast end-to-end resilience for cloud services over virtual optical networks,” in *Proc. 15th Int. Conf. Transparent Opt. Netw.*, 2013, pp. 1–7.

[36] J. Wu, Y. Liu, C. Yu, X. Wang, and J. Zhao, “Multicast protection based on enhanced intelligent p-cycle method in optical mesh networks,” *Opt. Eng.*, vol. 52, no. 1, p. 015008, 2013.

[37] M. M. A. Khan, N. Shahriar, R. Ahmed, and R. Boutaba, “Multi-path link embedding for survivability in virtual networks,” *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 253–266, 2016.

[38] M. M. A. Khan, N. Shahriar, R. Ahmed, and R. Boutaba, “Simple: Survivability in multi-path link embedding,” in *Proc. 2015 11th Int. Conf. Netw. Service Manage.*, 2015, pp. 210–218.

[39] T. Lin, K. Thulasiraman, and G. Xue, “Robustness of logical topology mapping problems for survivability under multiple physica links failures in IP over WDM optical networks,” in *Proc. 2012 4th Int. Conf. Commun. Syst. Netw.*, 2012, pp. 1–9.

[40] C. Liu and L. Ruan, “p-cycle design in survivable WDM networks with shared risk link groups (SRLGs),” *Photonic Netw. Commun.*, vol. 11, pp. 301–311, 2006.

Authors’ biographies not available at the time of publication.