Cross-Layer Network Survivability Under Multiple Cross-Layer Metrics

Zhili Zhou, Tachun Lin, Krishnaiyan Thulasiraman, Guoliang Xue, and Sartaj Sahni

***Abstract*—Given a cross-layer network with logical and physical topologies, the survivable logical topology routing problem is to route each link in the logical layer with a path in the physical topology between the end nodes of a logical link such that the logical topology remains connected after a physical link fails. The mixed-integer linear programming (MILP) formulation to determine such a routing has been considered in a recent paper. Using this formulation as a basic building block, in this paper we present unified MILP formulations to determine a survivable logical topology routing that also satisfies one of four cross-layer metrics: 1) minimizing the number of logical links to be added to guarantee the existence of survivable logical topology routing, 2) maximizing the capacity of the logical topology, 3) maximizing the connectivity of the logical topology after a physical link failure, and 4) maximizing the minimum cross-layer cut. We also provide heuristics for these problems and compare the performance of these heuristics and MILPs using extensive simulations.**

***Index Terms*—Cross-layer networks; Cross-layer reliability metrics; Cross-layer survivability; IP-over-WDM optical networks; Mathematical programming; Network augmentation; Network virtualization.**

I. Introduction

Over the past decade, explosive growth in mobile and Internet traffic has pushed demands for higher capacity in the data transmission of telecommunication networks. The data transmission rate of fiber optics networks has reached 305 Tb/s according to a recent paper [1]. When failure occurs under large-capacity transmission, it causes broader impacts and catastrophic results. Thus, reliability has become an issue of great interest in the design of modern telecommunication networks, and has been intensively studied in single-layer communication networks.

Manuscript received December 22, 2014; accepted April 13, 2015; published May 22, 2015 (Doc. ID 230885).

Z. Zhou (e-mail: zhili@sg.ibm.com) is with the IBM Research Collaboratory, Singapore.

T. Lin is with the Department of Computer Science and Information Systems, Bradley University, Peoria, Illinois 61625, USA.

K. Thulasiraman is with the School of Computer Science, University of Oklahoma, Norman, Oklahoma 73019, USA.

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

S. Sahni is with the Computer and Information Science and Engineering Department, University of Florida, Gainesville, Florida 32611, USA.

http://dx.doi.org/10.1364/JOCN.7.000540

In this paper, we focus on assessing cross-layer network reliability. An example of a cross-layer network architecture is the Internet Protocol over wavelength division multiplexing (IP-over-WDM) network, which is composed of logical (upper-layer, IP) and physical (lower-layer, WDM) networks. The demands of a link in the logical network are transmitted through a path connecting the corresponding node pair in the physical network. This logical link to physical-path mapping is called cross-layer mapping. We will use logical, upper-layer, and IP networks interchangeably, as well as physical, lower-layer, and WDM networks. We also refer to the topologies of the IP and WDM networks as logical and physical topologies, respectively.

In a logical topology, nodes and links represent IP routers and links connecting them, respectively. Similarly, physical nodes represent the optical cross-connect (OXC) and optical add–drop multiplexer (OADM), while the physical edges connecting them represent optical fibers. A lightpath is a cross-layer mapping of a logical link onto a path connecting corresponding physical nodes, through which transmission occurs on a single wavelength thereby bypassing opto-electro-optic (O-E-O) conversions on the intermediate nodes of the path.

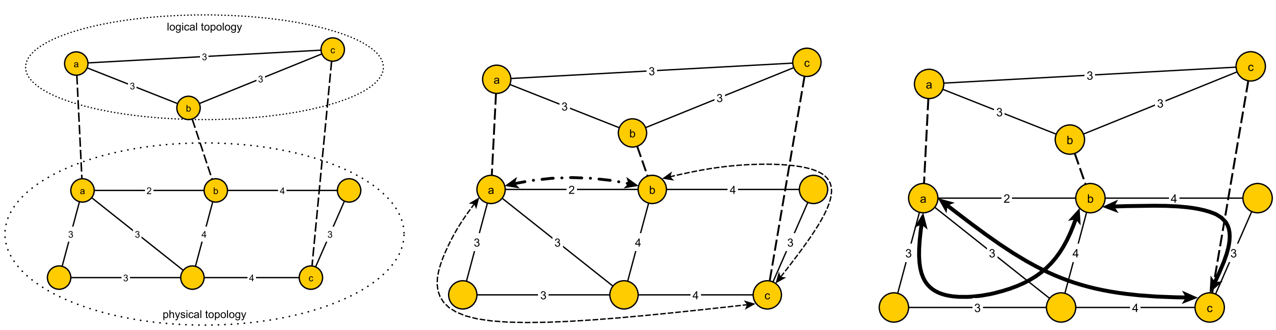
Another example of cross-layer networks is encountered in the study of network virtualization, where the virtual and real (physically existing) networks are, respectively, the logical and physical networks. Under the assumption that each virtual node is mapped to a physical node, the implementation of a virtual link is also realized through the cross-layer mapping.

Due to the fact that each physical link may carry traffic/demands of multiple logical links, a single physical link failure could disconnect multiple logical links in cross-layer networks. This has given rise to extensive interest in the study of reliability issues in cross-layer networks under multiple survivability evaluation metrics. A general definition of cross-layer network survivability is to identify a cross-layer mapping so that the logical network remains connected after any single physical link failure. A mapping that satisfies this evaluation metric is called a survivable cross-layer mapping. This survivability study is applicable to any cross-layer networks. It is obvious that the survivability of a mapping can be guaranteed if the lightpaths corresponding to this mapping are all link-disjoint. However, this is only a sufficient condition.

Furthermore, with the consideration of logical link demands and physical link capacities, the evaluation metric for cross-layer mapping has been extended from a pure survivability metric to an integrated one satisfying survivability and demands. This extension metric was first introduced by Lin *et al.* [2,3]. A cross-layer mapping is called a weakly survivable mapping if the logical network remains connected after a single physical link failure. A weakly survivable routing that also satisfies physical link capacity and logical demand constraints is called strongly survivable. Figure 1 illustrates weakly and strongly survivable cross-layer mappings. Here the vertical dashed lines show the correspondence between logical and physical nodes. Also, the capacities of physical links are shown on the corresponding links. Figure 1(a) provides the topologies and attributes of logical and physical networks. Figures 1(b) and 1(c) illustrate survivable cross-layer mappings with partial and full demand satisfaction, namely weakly and strongly survivable mapping, correspondingly.

Most of the papers in the literature deal with determining a survivable routing of logical links. In contrast, our focus in this paper is to study the integrated design of cross-layer mappings that satisfy certain quality of service requirements besides the survivability condition. For this purpose we identify three cross-layer metrics and present mixed-integer linear programs (MILPs) to determine a routing that maximizes one of these metrics. We also provide an MILP for the logical topology augmentation problem. Our MILPs start with an MILP for the survivable logical topology routing problem. We have chosen the MILP given in [2], which enforces the connectivity requirement through the existence of a spanning tree and does not require the exponential number of variables used by other MILPs. This MILP is reviewed in Section III after a survey of the literature on related works in Section II. All the MILPs developed in this paper use this MILP as a building block. The main contributions of the paper are summarized below.

- Minimum logical topology augmentation problem: Given a logical topology, the augmentation problem is to add a minimum number of additional links to the logical topology that guarantees the existence of a survivable routing of all the logical links of the augmented logical graph. Two earlier works that considered this problem are Thulasiraman *et al.* [4] and Liu and Ruan [5]. Neither of these two approaches solves the minimum augmentation problem. In Section IV we give an MILP that solves the minimum augmentation problem.
- Introduction of a new metric "after-failure connectivity" and determining a routing that maximizes after-failure connectivity: Suppose a logical graph is k-connected. (That is, at least k logical links have to be disconnected to disconnect the logical graph.) A survivable routing only guarantees that the logical graph will remain 1-connected after a physical link failure. Since one physical link failure may disconnect several logical links, a number of links (more than k links) may get disconnected when two or more failures occur, causing disconnection of the logical graph. To achieve a routing that can survive more than one physical link failure, we need to ensure that the connectivity of the logical graph is as high as possible after a single physical failure. With this in view in Subsection V.A we define a new metric called after-failure connectivity of the logical graph that is the connectivity of the logical graph after a single physical failure. The larger the value of after-failure connectivity of a routing, the better its ability to survive multiple physical faults. Thus the after-failure connectivity of a routing is a measure of the ability of the logical graph to remain connected after multiple physical link failures. Our interest is to determine a routing that has as high a value of after-failure connectivity as possible. We give in Subsection V.A an MILP to determine a survivable routing that maximizes the after-failure connectivity of the logical graph. Our MILP has two significant features: 1) normally calculating the connectivity of the logical graph after each physical failure would require $n(n-1)/2$ maximum flow computations. Using a result in network flow theory we reduce the problem to one of n maximum flow computations. 2) We are required to calculate the connectivity of the logical graph after each failure. In other words, connectivity computation has to be repeated on m after-failure logical graphs. Normally this would require solving m separate integer linear programs if we explicitly generate the logical graph after each one of the m physical failures. See [6], where such an approach was taken in a different context. In our MILP we implicitly generate the after-failure graphs so that just one MILP is enough to calculate the after-failure connectivity.



(a) Given physical/logical topologies (b) Weakly survivable mapping (c) Strongly survivable mapping

Fig. 1. Cross-layer survivability illustration.

- Introduction of a new metric "logical capacity" and determining a routing that maximizes logical capacity: Given a survivable routing, let us associate with each logical link (s, t) a capacity equal to the capacity of the corresponding lightpath. Using these link capacities let us find the maximum $s-t$ flow between the nodes s and t of each logical link. The capacity of the logical graph is the minimum of the maximum $s-t$ flows over all pairs of nodes in the logical graph constructed. When a physical link fails, several logical links could fail. The larger the capacity of the logical graph under a given routing, the greater the capacity available for routing after a physical link failure. Thus the logical capacity is a measure of the available capacity after a failure. In Subsection V.B we provide an MILP that achieves a routing that maximizes the capacity of the logical graph. This is in contrast with our DRCN paper [3] that achieves a routing that maximizes the total capacity (sum of all logical link capacities, not logical capacity as defined above) available before and after a physical link failure.
- Determining a routing that maximizes minimum cross-layer cut (MCLC): In [7] Lee and Modiano defined the concept of MCLC. The MCLC value of a given survivable logical topology routing is the minimum number of physical link failures that would cause the logical layer to become disconnected. For example, if the MCLC value of a routing is 5, this means that the logical graph will remain connected after any set of five simultaneous physical link failures. In Subsection V.C we give an MILP that determines a routing that maximizes MCLC, in contrast to the work in [7] that determines the MCLC value of a given routing.

In Section VI we provide heuristics for all the optimization problems considered in earlier sections. In Section VII we provide extensive simulation results evaluating the performance of the heuristics and the corresponding MILPs.

II. Related Work

A typical network reliability problem is to efficiently calculate the probability that a specified set of nodes can communicate with each other at a given time. The research works in the literature developed an exact solution, Monte Carlo simulation, and the polynomial expression for network reliability estimations in single-layer networks. Agrawal and Barlow [8] provided a survey for early exact factoring algorithms with domination theory. Rosenthal and Frisque [9] and Shooman and Kershenbaum [10,11] utilized network transformation and reduction for network reliability estimation. Dikbiyik *et al.* [12] utilized preprovisioning/reprovisioning and hold-lightpath schemes, which balanced optical network availability, resource efficiency, and protection through excess capacity management. These research works were for single-layer networks and cannot estimate cross-layer network reliability, as they assumed that link failures are independent. In fact, a single physical link failure may cause multiple failures in the logical network. Monte Carlo simulation was also used for estimating single-layer reliability for some fixed link failure probability. In [13], a Monte Carlo method is given to estimate the failure probability with random network link states. With Monte Carlo simulation, network reliability could be approximated to an arbitrary accuracy. The computational performance of simulations was restricted by a large number of iterations and repeating them with different reliability indices. A randomized fully polynomial approximation scheme for the all-pairs reliability problem was given in [14]. A comprehensive study of combinatorial aspects of the network reliability problem may be found in [15]. Recently, there has been interest in the study of the network reliability problem when links and nodes are broken in a probabilistic manner. Liew and Lu [16] and and Neumayer *et al.* [17] discussed this problem and have several references to early related works.

As noted earlier, the survivability of a logical topology mapping can be guaranteed if the lightpaths in the physical topology corresponding to this mapping are all link-disjoint. Since finding mutually disjoint paths between multiple pairs of nodes is NP-complete [18], the cross-layer network reliability problem with survivability as an evaluation metric is also NP-complete. Modiano and Narula-Tam [19] have given a necessary and sufficient condition for a cross-layer mapping to be survivable under a single physical link failure in IP-over-WDM networks, and formulated the problem as an integer linear program (ILP). Todimala and Ramamurthy [20] adapted the concept of a shared risk link group (SRLG) introduced in [21] and also computed the routing through an ILP formulation. Extensions of [19] were given by Kan *et al.* in [6], which discussed the relationship between survivable lightpath routing and spare capacity requirements on the logical links satisfying the original traffic demands after failures. Lin *et al.* [2,3] discussed rerouting disconnected logical links due to a physical link failure to maximize the sum of satisfied demands. Lin *et al.* [3] also studied spare capacity assignment to physical links to guarantee the availability of capacities for rerouting after a failure. Similar to [3], Vadrevu *et al.* [22] also proposed ILPs and heuristics addressing survivability and reroutability for logical links with backup capacity sharing after any single physical link failure.

A common drawback of ILP/MILP approaches is that they are not scalable as the network size increases. Hence, heuristic approaches were designed to overcome such issues. Kurant and Thiran [23] proposed the survivable mapping by ring trimming (SMART) framework, which maps the links of certain subgraphs of the given logical graph into link-disjoint paths. Lee *et al.* [24] solved the same problem utilizing the concept of ear-decomposition in graph theory. Javed *et al.* [25,26] obtained improved heuristics based on SMART. Using duality theory in graphs, a generalized theory of logical topology survivability was given in [27–29]. A new structural approach based on the logical protecting spanning tree set concept was introduced in [30]. Lin *et al.* [31] also presented an integrated approach to design survivable logical topology routing and localize physical link failures. Operations research techniques, such as column generation, have been incorporated into MILP formulations to solve similar problems.

Jaumard and Hoang [32] proposed an MILP formulation and a solution approach based on the column generation technique that can generate exact solutions for different scales of logical networks. Two works related to disaster aware survivability in the context of cloud computing and survivable infrastructure designs are reported in [33,34].

III. Weakly Survivable Routing Maximizing Total Demand Satisfied: Review of an MILP [2,3]

We use the terms network and topology, edge and link, node and vertex, interchangeably throughout the paper. Let $G_L = (V_L, E_L)$ be a logical network and $G_P = (V_P, E_P)$ be a physical network in a cross-layer network. Let (i,j) be a physical link and (s,t) be a logical link. The capacity on physical link (i,j) is c_{ij}, and the demand on logical link (s,t) is d_{st}. We assume that both the logical and physical networks are at least two-edge connected.

Definition 1: A logical topology mapping in a cross-layer network with logical topology $G_L = (V_L, E_L)$ and physical topology $G_P = (V_P, E_P)$ is weakly survivable if G_L remains connected after a link failure in G_P.

In this section we review an MILP given in [3] to determine a weakly survivable routing in a cross-layer network that maximizes the sum of all logical demands satisfied subject to the limits on the capacities of the physical link. This MILP formulation to be called WSR-MD is the basis of all the MILP formulations given in the rest of the paper. See Table I for the definitions of the variables used in the WSR-MD formulation.

WSR-MD: MILP to determine a weakly survivable logical topology routing maximizing the total demand:

$$\max \sum_{(s,t)\in E_L} \rho_{st} \tag{1}$$

$$\text{s. t.} \sum_{(i,j)\in E_P} y_{ij}^{st} - \sum_{(j,i)\in E_P} y_{ji}^{st} = \begin{cases} 1, & \text{if } i = s, \\ -1, & \text{if } i = t, \\ 0, & \text{if } i \neq \{s,t\}, \end{cases} \quad i \in V_P, (s,t) \in E_L, \tag{2}$$

$$y_{ij}^{st} + y_{ji}^{st} \leq 1, \qquad (i,j) \in E_P, (s,t) \in E_L, \tag{3}$$

$$\sum_{(i,j),(j,i)\in E_P} (y_{ij}^{st} + y_{ji}^{st}) \leq 2, \qquad (s,t) \in E_L, \tag{4}$$

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = \begin{cases} \rho_{st}, & \text{if } i = s, \\ -\rho_{st}, & \text{if } i = t, \\ 0, & \text{if } i \neq \{s,t\}, \end{cases} \quad i \in V_P, (s,t) \in E_L, \tag{5}$$

$$\rho_{st} \leq d_{st}, \qquad (s,t) \in E_L, \tag{6}$$

$$f_{ij}^{st} \leq M y_{ij}^{st} \ \text{ and } \ f_{ji}^{st} \leq M y_{ji}^{st}, \qquad (i,j) \in E_P, (s,t) \in E_L, \tag{7}$$

$$\sum_{(s,t)\in E_L} (f_{ij}^{st} + f_{ji}^{st}) \leq c_{ij}, \qquad (i,j) \in E_P, \tag{8}$$

$$\sum_{(s,t)\in E_L} r_{st}^{ij} - \sum_{(t,s)\in E_L} r_{ts}^{ij} = \begin{cases} -1, & \text{if } s = v_1, \\ \frac{1}{|V_L|-1}, & \text{if } s \neq v_1, \end{cases} \quad v_1 \in V_L, (i,j) \in E_P, \tag{9}$$

$$0 \leq r_{st}^{ij} \leq 1 - (y_{ij}^{st} + y_{ji}^{st}), \qquad (i,j) \in E_P, (s,t) \in E_L, \tag{10}$$

$$0 \leq r_{ts}^{ij} \leq 1 - (y_{ij}^{st} + y_{ji}^{st}), \qquad (i,j) \in E_P, (s,t) \in E_L, \tag{11}$$

$$y_{ij}^{st} \in \{0,1\}, r_{ij}^{st}, f_{ij}^{st}, \rho_{st} \geq 0, \qquad (s,t) \in E_L, (i,j) \in E_P.$$

Constraint (2) with binary variable y_{ij}^{st} provides a cross-layer routing for each logical link (s,t) with single unit flow. Constraint (3) guarantees that a logical-layer routing is not routed through a physical-layer link in its two directions. Constraint (4) eliminates other loops by avoiding a logical-layer routing to revisit the same node on the physical routing. Without constraints (3) and (4), constraint (2) itself will not be able to avoid self-loops and re-entry of the same node. Constraints (5) and (6) push a flow of value ρ_{st} through the logical-layer routing selected for logical link (s,t). Constraint (7) guarantees that each physical link (i,j) carries flow only if the logical-layer routing is routed through at least one direction of the physical link. Here M is a very large number greater than the maximum link capacity. Constraint (8) requires that the total flow of all the logical-layer routings carried by each physical-layer link is no more than the corresponding physical link capacity. Constraints (9)–(11) ensure that the logical network after the failure of each physical link contains a spanning tree. Constraint (9) is used to guarantee the existence of a logical spanning structure after a physical link failure. Constraints (10) and (11) make sure that the spanning tree structure is composed of the logical edges not disconnected by the failure of a physical link (i,j). We wish to note that

TABLE I
Variables Used in MILP Formulation WSR-MD

Variable	Description
y_{ij}^{st}	binary variable indicating whether the logical link $(s,t) \in E_L$ is routed through the physical link $(i,j) \in E_P$. If yes, $y_{ij}^{st} = 1$; otherwise, $y_{ij}^{st} = 0$.
f_{ij}^{st}	flow on physical link (i,j) due to lightpath (s,t)
r_{st}^{ij}	fractional variable for connectivity constraints. $r_{st}^{ij} = 0$ if logical link (s,t) is routed through physical link (i,j).
ρ_{st}	capacity for the logical link (s,t), where ρ_{st} is the smallest capacity of links in the lightpath.
c_{ij}	capacity on the physical link (i,j). $c_{ij} = c_{ji}$.
d_{st}	demand for the logical link (s,t). $d_{st} = d_{ts}$.
M	a large positive number

the survivability constraints (9)–(11) were first employed in Deng *et al.*'s paper [35].

IV. Minimum Logical Topology Augmentation for Guaranteed Weakly Survivable Routing

Given a cross-layer network, it is possible that the logical topology does not permit a weakly survivable routing. In such a case we need to add additional logical links to guarantee the existence of a weakly survivable routing. This augmentation problem was earlier considered in [4,5]. In this section we now show how to enhance the WSR-MD algorithm to accommodate steps to augment the logical network with a minimum number of additional links so that the augmented network admits a weakly survivable routing. It is shown in [4] that such an augmentation is always possible if the physical network is at least three-edge connected.

Let L_L be the collection of all ordered pairs of vertices in the logical network. Some of these ordered pairs may not represent links in the original logical network. We introduce auxiliary variable h_{st} to indicate whether (s,t) is a link in the logical network. (s,t) could be a link in the original logical network or was added during augmentation. We let g_{st} be the variable that indicates whether (s,t) is a link in the original logical network G_L. Then, we have the following.

Logical link augmentation constraints:

$$g_{st} \le h_{st}, \qquad (s,t) \in L_L, \tag{12}$$

$$y_{ij}^{st} \le h_{st}, \qquad (s,t) \in L_L, (i,j) \in E_P, \tag{13}$$

$$\sum_{(i,j)\in E_P} y_{ij}^{st} - \sum_{(j,i)\in E_P} y_{ji}^{st} = \begin{cases} h_{st}, & \text{if } i = s, \\ -h_{st}, & \text{if } i = t, \\ 0, & \text{if } i \ne \{s,t\}, \end{cases} \quad i \in V_P, (s,t) \in L_L, \tag{14}$$

$$g_{st}, h_{st} \in \{0,1\}, \qquad (s,t) \in L_L. \tag{15}$$

Constraint (12) builds the connection between the indicator g_{st} and variable h_{st}. If (s,t) is a link in the given logical network (that is, $g_{st} = 1$), then (s,t) exists in the augmented network (that is, $h_{st} = 1$). So, in this case, $g_{st} \le h_{st}$. If $g_{st} = 0$, constraint (12) is trivially satisfied. Constraint (13) indicates that a logical routing is generated only if (s,t) is in the augmented network. Constraint (14), similar to constraint (2), provides logical routings for original and augmented links in the logical topology.

The MILP formulation for the minimum logical topology augmentation problem is as follows.

WSR-MLA: Minimal logical topology augmentation for guaranteed weakly survivable routing:

$$\min \sum_{(s,t)\in L_L} h_{st}$$
$$\text{s. t. Constraints (3), (4), (9)–(15).}$$

V. Weakly Survivable Routing Under Cross-Layer Metrics

In this section we develop MILP formulations to determine a survivable logical topology routing that maximizes one of three cross-layer metrics.

A. Maximizing the After-Failure Connectivity of the Logical Topology

The connectivity of a graph is the smallest number of edges whose removal disconnects the graph [36]. Our interest is to find a survivable logical topology routing that maximizes the after-failure connectivity of the logical topology after a single physical link failure. This is equivalent to finding a survivable logical routing that maximizes the number of edges remaining in any cut of G_L after any physical link failure.

In the following G_L^{ij} will refer to the graph that results from deleting those links in G_L that get broken when physical link (i,j) fails. Let $k_{ij}(s,t)$ denote the maximum number of link-disjoint paths between logical nodes s and t in G_L^{ij}, which is the $s-t$ connectivity of G_L^{ij} [36]. Then the after-failure connectivity of G_L under a given routing is $\min_{s,t\in V_L,(i,j)\in E_P}\{k_{ij}(s,t)\}$. Our objective is to determine a survivable logical routing that maximizes the after-failure connectivity of G_L.

Let $k_{vw}^{ij}(s,t)$ denote the flow on link $(v,w) \in E_L$ due to an $s-t$ flow in G_L^{ij}. Picking a node s_0, the following MILP will calculate the maximum number of link-disjoint $s_0 - t$ paths for all $t \in V_L$ in each G_L^{ij}, and then pick the minimum of these numbers as the after-failure connectivity of G_L. Note that we do not have to calculate the maximum number of link-disjoint paths between all pairs of logical nodes s and t. See [37] for the correctness of this approach.

$$\text{Given } s_0 \in V_L, \qquad t \in V_L,$$
$$\max k \tag{16}$$

$$\text{s. t.} \sum_{(v,w)\in E_L} k_{vw}^{ij}(s_0,t) - \sum_{(w,v)\in E_L} k_{wv}^{ij}(s_0,t) = \begin{cases} k_{ij}(s_0,t), & \text{if } v = s_0, & v \in V_L \\ -k_{ij}(s_0,t), & \text{if } v = t, & v \in V_L, \\ 0, & \text{if } v \ne \{s_0,t\}, & v \in V_L \end{cases} \quad (i,j) \in E_P, \tag{17}$$

$$k_{vw}^{ij}(s_0,t) + k_{wv}^{ij}(s_0,t) \le 1,$$
$$(v,w),(w,v) \in E_L, (i,j) \in E_P, t \in V_L, \tag{18}$$

$$\sum_{(v,w),(w,v)\in E_L} (k^{ij}_{vw}(s_0,t) + k^{ij}_{wv}(s_0,t)) \le 2, \quad (i,j) \in E_P, \tag{19}$$

$$k^{ij}_{vw}(s_0,t) \le 1 - y^{vw}_{ij}, \qquad (i,j) \in E_P, (v,w) \in E_L, \tag{20}$$

$$k \le k_{ij}(s_0,t), \qquad (i,j) \in E_P, \tag{21}$$

$$k^{ij}_{vw} \in \{0,1\}, k_{ij}(s_0,t) \in \mathcal{Z}^+, \qquad (i,j) \in E_P, (v,w) \in E_L. \tag{22}$$

Note that constraints (18) and (19) guarantee that logical routings of (s_0,t) generated by constraint (17) are link-disjoint. Constraint (20) guarantees that only edges that are not broken due to $(i,j) \in E_P$ failure are considered when calculating $k_{ij}(s_0,t)$. This interesting feature helps us avoid calculating G^{ij}_L explicitly for each (i,j), thereby allowing us to provide one single formulation instead of several formulations, one for each physical link (i,j).

Now using objective (16) in place of objective (1) and adding the constraints (17)–(22) to the MILP (WSR-MD) we get the following MILP WSR-AFC that determines a survivable logical topology routing that maximizes the after-failure connectivity of G_L.

WSR-AFC: Weakly survivable routing maximizing after-failure connectivity of logical topology:

$$\max k$$
$$\text{s. t. Constraints (2)–(4), (9)–(11), (17)–(22).}$$

We wish to note that after-failure connectivity is a measure of the ability of a logical topology routing to provide survivability against multiple failures. If this connectivity is larger, then one can expect the routing to provide survivability against a larger number of physical link failures.

B. Maximizing the Capacity of the Logical Topology

Given a routing that achieves a demand of ρ_{vw} on logical link (v,w), let $\Psi(s,t)$ be the maximum flow between any pair of nodes $s,t \in V_L$ while treating ρ_{vw} as the capacity of link $(v,w) \in E_L$. Then we define the capacity of G_L under the given routing as $\min_{s,t\in V_L}\{\Psi(s,t)\}$. Our interest is to determine a survivable logical routing that maximizes the logical capacity. We proceed as follows.

Let $\bar{f}_{vw}(s,t)$ be the flow on logical link (v,w) due to a flow $\Psi(s,t)$ between nodes s and t in V_L. The following linear program determines $\Psi(s,t)$ for each pair of nodes s and t in V_L and picks the minimum of these maximum flows, which is the capacity of G_L:

$$\max \tilde{f} \tag{23}$$

$$\text{s. t.} \sum_{(v,w)\in E_L} \bar{f}_{vw}(s,t) - \sum_{(w,v)\in E_L} \bar{f}_{wv}(s,t) = \begin{cases} \Psi(s,t), & \text{if } v = s, \quad s,t,v \in V_L \\ -\Psi(s,t), & \text{if } v = t, \quad s,t,v \in V_L, \\ 0, & \text{if } v \ne \{s,t\}, \quad s,t,v \in V_L \end{cases} \tag{24}$$

$$\bar{f}_{vw}(s,t) + \bar{f}_{wv}(s,t) \le \rho_{vw}, \qquad s,t \in V_L, (v,w) \in E_L, \tag{25}$$

$$\tilde{f} \le \Psi(s,t), \qquad s,t \in V_L, \tag{26}$$

$$\bar{f}_{vw}(s,t), \Psi(s,t), \tilde{f} \ge 0, \qquad s,t \in V_L, (v,w) \in E_L. \tag{27}$$

Using the objective (23) in place of objective (1) and adding the constraints (24)–(27) to the MILP (WSR-MD), we get the following MILP that determines a survivable logical topology routing that maximizes the capacity of the logical topology G_L.

WSR-MLC: Weakly survivable routing with maximal logical capacity:

$$\max \tilde{f}$$
$$\text{s. t. Constraints (2)–(11) and (24)–(27).}$$

C. Maximizing the Min-Cross-Layer Cut

Given a logical topology routing $\mathcal{R}$, the MCLC $(\mathcal{R})$ of $\mathcal{R}$ is defined in [7] as the smallest number of physical link failures that will disconnect the logical topology. We wish to find a routing $\mathcal{R}$ that has the maximum value for $(\mathcal{R})$. If this maximum value is greater than or equal to 1, then that routing will be survivable.

In [7] Lee and Modiano showed that finding a routing that has the largest MCLC value is NP-complete. So they presented an ILP to find a routing that minimizes a quantity whose reciprocal gives a lower bound on the MCLC value. Lee and Modiano [7] also presented an ILP M_{MCLC} to find the MCLC value of a given logical topology routing. We now show how M_{MCLC} can be integrated with the constraints (1)–(5) of Section III to yield an ILP for finding a logical topology routing with the largest MCLC value. This ILP is given in objective (28)–(35).

WSR-MCLC: Weakly survivable routing with maximum MCLC value:

$$\min \lambda$$
$$\text{s. t. Constraints (2)–(4)}$$

$$\lambda \ge \sum_{(i,j)\in E_P} \nu_{ij}, \tag{28}$$

$$\xi^{st}_{ij} \le \nu_{ij}, \qquad (i,j) \in E_P, \tag{29}$$

$$\xi^{st}_{ij} \le y^{st}_{ij}, (s,t) \in E_L, \qquad (i,j) \in E_P, \tag{30}$$

$$\xi_{ij}^{st} \geq \nu_{ij} + y_{ij}^{st} - 1, (s,t) \in E_L, \qquad (i,j) \in E_P, \tag{31}$$

$$|\psi_t - \psi_s| \leq \sum_{(i,j)\in E_P} \xi_{ij}^{st}, \qquad (s,t) \in E_L, \tag{32}$$

$$\sum_{s\in V_L} \psi_s \geq 1, \tag{33}$$

$$\psi_0 = 0, \tag{34}$$

$$\psi_s, \nu_{ij}, \xi_{ij}^{st} \in \{0,1\}, \lambda \geq 0, \qquad s,t \in V_L, (s,t) \in E_L, (i,j) \in E_P. \tag{35}$$

First we note that the variables ν_{ij}, y_{ij}^{st}, and ψ_t used in our ILP correspond to the variables y_{ij}, f_{ij}^{st}, and d_t used in M_{MCLC} of [7]. Second, we point out that constraint (3) in [7],

$$\psi_t - \psi_s \leq \sum_{(i,j)\in E_P} \nu_{ij} y_{ij}^{st}, \quad \forall\ (s,t) \in E_L,$$

should be replaced with

$$|\psi_t - \psi_s| \leq \sum_{(i,j)\in E_P} \nu_{ij} y_{ij}^{st}, \quad \forall\ (s,t) \in E_L, \tag{36}$$

due to the fact that the identification of whether s and t are connected is evaluated by whether s is disconnected from t or t is disconnected from s. To obtain this bidirection information, an absolute value is used as shown in constraint (36).

The main difficulty in using M_{MCLC} is that it has a constraint that involves the nonlinear term $\nu_{ij} y_{ij}^{st}$. This constraint needs to be replaced by an equivalent set of constraints. For this purpose we introduced a new variable ξ_{ij}^{st} that indicates whether the routing of (s,t) is impacted by the failure of physical link (i,j). With this new variable constraints (29)–(31) represent a set of constraints equivalent to the following constraint:

$$\sum_{(i,j)\in E_P} \xi_{ij}^{st} = \sum_{(i,j)\in E_P} \nu_{ij} y_{ij}^{st}, \qquad (s,t) \in E_L.$$

This follows because $y_{ij}^{st} = 1 \Rightarrow \nu_{ij} y_{ij}^{st} = \nu_{ij}$ and $\xi_{ij}^{st} = \nu_{ij}$ because of constraints (29) and (31). $y_{ij}^{st} = 0 \Rightarrow \nu_{ij} y_{ij}^{st} = 0$ and $\xi_{ij}^{st} = 0$ because of constraint (30).

Then constraint (32) can be further linearized as

$$\psi_t - \psi_s \leq \sum_{(i,j)\in E_P} \xi_{ij}^{st}, (s,t) \in E_L,$$
$$\psi_s - \psi_t \leq \sum_{(i,j)\in E_P} \xi_{ij}^{st}, (s,t) \in E_L.$$

VI. Heuristics

The MILP formulations presented in Sections IV and V require high execution times, though they produce optimal/feasible solutions. To mitigate the computational complexity of these formulations, we present in this section heuristics for all the problems considered.

A. Heuristic for Logical Topology Augmentation

The heuristic augments the given logical topology with additional links so that a survivable routing is guaranteed for the augmented logical topology. The routing corresponding to each logical link is also generated. This procedure is provided in Algorithm 1.

Algorithm 1 Augment for Survivability

Input: Physical topology $G_P = (V_P, E_P)$ and logical topology $G_L = (V_L, E_L)$
Output: A survivable augmented logical topology and its routing

Sort all nodes in V_L by their degrees
Find datum node $\Delta \in V_L$ with the largest degree
$M_{G'_L} = \emptyset, G'_L = G_L$
while $\exists v \in G'_L$ with degree ≥ 2 **do**
  Sort all v by the cost $C_v = (v\text{'s logical degree}) \times (v\text{'s physical degree})$
  Select node v' with the largest $C_{v'}$, $v' \in V_L$
  Find v''s two adjacent nodes v_2 and v_3 with the largest $C_{v'_2}$ and $C_{v'_3}$
  Map (v', v_2) and (v', v_3) into disjoint paths and update $M_{G'_L}$
  $G'_L = G'_L \backslash \{v'\}$
end while
while $\exists v \in G'_L$ with degree $= 1$ (individual edge $e \in E_L$) **do**
  Add a parallel edge e' to e and find disjoint routings for e and e' in G_P
  Update the routings in $M_{G'_L}$
  $G'_L = G'_L \backslash \{e\}$
end while
while $\exists$ isolated node $v \in G'_L$ **do**
  Add two edges connecting v and Δ and find two disjoint routings for the two edges.
  Update the routings in $M_{G'_L}$
  Remove v from G'_L
end while

The heuristic first sorts all logical nodes by their degrees, and a datum node with the maximum degree is assigned and denoted as Δ. While there exists a logical node v with degree ≥ 2, the heuristic assigns each logical node v with degree ≥ 2 a cost $C_v = (\text{degree of logical node}) \times (\text{degree of correspondingphysical node})$. The heuristic then selects the logical node with the largest C_v and chooses two of v's adjacent nodes v_1 and v_2 with the largest C_{v_1} and C_{v_2}. (v, v_1) and (v, v_2) are then mapped into disjoint paths in the physical topology. After that, v is removed from the logical topology. This procedure is repeated until no logical nodes with degree ≥ 2 are left. Next, the heuristic picks a node v from the remaining logical topology with degree $= 1$, i.e., an edge (u,v). An edge $(u,\nu)'$ parallel to (u,v) is then added, and disjoint mappings for (u,v) and $(u,\nu)'$ in the physical topology are found. This procedure is executed until the elimination of all logical nodes with degree $= 1$. After the previous steps, if there exist nodes v with degree $= 0$, two parallel edges connecting v and Δ are added and they are mapped disjointly in the physical topology. The heuristic is provided in Algorithm 1. Proof of the correctness of Algorithm 1 may be found in [4].

Regarding the computational complexity, the complexity of finding a pair of link-edge disjoint paths dominates the time required, which is $O(m^{3/2})$. So, the overall complexity is $O(nm^{3/2})$.

B. Heuristics for Maximizing Logical Capacity

We present a heuristic for maximizing logical capacity in Algorithm 2. To guarantee survivability, the heuristic would still augment the logical topology with additional links. Steps 1–6 are the same as those in Algorithm 1. After selecting the candidate node C_v with degree ≥ 2 and the largest C_v, instead of mapping its adjacent nodes v_1 and v_2 with the largest C_{v_1} and C_{v_2}, the C_{v_i}'s are included in a priority list. The heuristic selects two nodes at a time with the highest C_{v_i} (in descending order), finds their disjoint mappings, and determines the minimal capacity of the routing. Among all the routings generated, the one that maximizes the minimal capacity is selected. This procedure is repeated until no logical node with degree ≥ 2 is left.

After a survivable routing is generated, to determine the maximal logical capacity, we first push the flow for all logical demands. This step is done by repeatedly pushing the unit flow for each logical demand until the physical capacity cannot carry more logical demands. Here the physical edge capacity of an edge would be updated if the unit flow is routed through this edge. After the above step, the maximum flow is pushed for every logical node pair.

Algorithm 2 Survivability and Maximizing Logical Topology Capacity

Input: Physical topology $G_P = (V_P, L_P)$, logical topology $G_L = (V_L, E_L)$
Output: A logical routing with the maximum capacity
1: Sort all nodes in V_L by their degrees
2: Find datum node $\Delta \in V_L$ with the largest degree
3: $M_{G'_L} = \emptyset$, $G'_L = G_L$
4: **while** $\exists v \in G'_L$ with degree ≥ 2 **do**
5:   Sort all v by the cost $C_v =$ (v's logical degree) × (v's physical degree)
6:   Select node v' with the largest $C_{v'}$, $v' \in V_L$
7:   Sort v' adjacent node v'_i by $C_{v'_i}$
8:   **for** Each v''s adjacent node pair v'_1, v'_2 from sorted $C_{v'_i}$ list (in descending order) **do**
9:     Map (v', v'_1), (v', v'_2) disjointly and decide the minimum capacity physical link on the lightpath
10:     Record the sum of minimum capacities of the two lightpaths
11:   **end for**
12:   Select the node pairs v'_1, v'_2 with the largest sum of minimum capacities and update $M_{G'_L}$ with lightpaths of (v', v'_1), (v', v'_2)
13:   $G'_L = G'_L \backslash \{v'\}$
14: **end while**
15: **while** $\exists v \in G'_L$ with degree $= 1$ (individual edge $e \in E_L$) **do**
16:   Add a parallel edge e' to e and find disjoint routings for e and e' in G_P
17:   Update the routings in $M_{G'_L}$
18:   $G'_L = G'_L \backslash \{e\}$
19: **end while**
20: **while** $\exists$ isolated node $v \in G'_L$ **do**
21:   Add two edges connecting v and Δ and find two disjoint routings for the two edges.
22:   Update the routings in $M_{G'_L}$
23:   Remove v from G'_L
24: **end while**
25: **while** $\exists$ unsatisfied logical demand of $u \in E_L$ and residual physical capacity supporting the demand **do**
26:   Push unit demand between edge nodes of u through u's routing
27:   Update physical link capacity
28: **end while**
29: **for** all v, $w \in V_L$, $v \neq w$ **do**
30:   Push the maximum flow between (v, w)
31: **end for**
32: Return the maximum flow among all v, $w \in V_L$

The complexity is dominated by step 9 in Algorithm 2. At most n^2 pairs of adjacent nodes are picked, and for each pair edge-disjoint paths are to be found. This requires $O(n^2 m^{3/2})$ time. Since this is done for each node, the overall complexity is $O(n^3 m^{3/2})$.

The complexity of Algorithms 1 and 2 is highly conservative as can be seen from the simulation results.

C. Heuristics for Maximizing the After-Failure Connectivity and the MCLC

The heuristics in Algorithms 3 and 4 start with a spanning tree t, and map their links into paths using a shortest path algorithm. At the end of this step all the physical edges that were not used in this mapping of the links in the tree are stored in a set $CB(t)$.

Here a block refers to the set of cotree edges with respect to spanning tree edges. We have to make an effort so that blocks are not disjoint. We want them to overlap as much as possible so that an edge appears in more than one block. If this could be done, then after a physical link failure several trees will remain connected increasing the connectivity of the logical network after a failure. $\mathcal{W}$ is a large number.

Algorithm 3. Maximizing After-Failure Connectivity.

After finding mappings of all links in a tree t_i the set $CB(t_i)$ is created. All the edges in $CB(t_i)$ are assigned a large $\mathcal{W}$ to discourage their selection for the mappings in the subsequent tree. This is to ensure that CB blocks for two subsequent trees overlap as much as possible. This will guarantee that after the failure of an edge that is in two consecutive *CB*'s at least two trees will remain, resulting in increased after-failure connectivity.

Algorithm 4. Maximizing MCLC Value.

In this heuristic all the edges that are used in the mapping of a logical link in a tree t_i are assigned a large weight so that these edges are discouraged from entering the *CB* for t_i. This is to create as large a $CB(t_i)$ as possible. A larger *CB* means that multiple failures of the edges in *CB* will not disconnect the tree t_i, thereby increasing the MCLC value.

Algorithm 3 Heuristic: Maximizing the Connectivity After a Single Physical Link Failure

Input: $G_P = (V_P, E_P)$, $G_L = (V_L, E_L)$, $\mathcal{T} = \emptyset$, $EM[u] = \emptyset$, $\forall\, u \in E_L$, $w(e) = 1$, $\forall\, e \in E_P$, $\tau_i = \emptyset$, $i = 0$, $\mathcal{W}$.

while $\mathcal{T} \neq E_L$ **do**
  Generate τ_i, such that $\exists EM[v] = \emptyset$, $v \in \tau_i$
  $\mathcal{CB}(\tau_i) = E_P$
  for $v \in \tau_i$ **do**
    if $EM[v] \neq \emptyset$ **then**
      for all $k \in EM[v]$ **do**
        $\mathcal{CB}(\tau_i) = \mathcal{CB}(\tau_i) \backslash k$
        $w(k) = 1$
      end for
    end if
  end for
  $\mathcal{T} = \mathcal{T} \bigcup \tau_i$
  for all $v \in \tau_i$ **do**
    if $EM[v] = \emptyset$ **then**
      Generate $p^v, EM[v] = p^v$
      for all $k \in EM[v]$ **do**
        $\mathcal{CB}(\tau_i) = \mathcal{CB}(\tau_i) \backslash k$
      end for
    end if
  end for
  for all $k \in E_P$ **do**
    if $k \in \mathcal{CB}(\tau_i)$ **then**
      $w(k) = w(k) + \mathcal{W}$
    else
      $w(k) = 1$
    end if
  end for
  $i = i + 1$
end while
if $E_P \neq \bigcup EM[u]$, $\forall\, u \in E_L$ **then**
  AUGMENTATION
end if

Algorithm 4 Heuristic: Maximizing the MCLC

Input: $G_P = (V_P, E_P)$, $G_L = (V_L, E_L)$, $\mathcal{T} = \emptyset$, $EM[u] = \emptyset$, $\forall\, u \in E_L$, $w(e) = 1$, $\forall\, e \in E_P$, $\tau_i = \emptyset$, $i = 0$, §.

while $\mathcal{T} \neq E_L$ **do**
  Generate τ_i, where $\exists EM[v] = \emptyset$, $v \in \tau_i$
  $\mathcal{CB}(\tau_i) = E_P$
  for $v \in \tau_i$ **do**
    if $EM[v] \neq \emptyset$ **then**
      for all $k \in EM[v]$ **do**
        $\mathcal{CB}(\tau_i) = \mathcal{CB}(\tau_i) \backslash k$
      end for
    end if
    for all $k \in E_P$ **do**
      if $k \in \mathcal{CB}(\tau_i)$ **then**
        $w(k) = w(k) + \mathcal{W}$
      else
        $w(k) = 1$
      end if
    end for
  end for
  $\mathcal{T} = \mathcal{T} \bigcup \tau_i$
  for all $v \in \tau_i$ **do**
    if $EM[v] = \emptyset$ **then**
      Generate p^v, $EM[v] = p^v$
      for all $k \in EM[v]$ **do**
        $\mathcal{CB}(\tau_i) = \mathcal{CB}(\tau_i) \backslash k$
      end for
      for all $k \in E_P$ **do**
        if $k \in \mathcal{CB}(\tau_i)$ **then**
          $w(k) = w(k) + \mathcal{W}$
        else
          $w(k) = 1$
        end if
      end for
    end if
  end for
  $i = i + 1$
end while
if $E_P \neq \bigcup EM[u]$, $\forall\, u \in E_L$ **then**
  AUGMENTATION
end if

VII. Computational Results

In this section, we report our computational results evaluating the effectiveness and performance of our exact solution approaches (MILP formulations) and heuristics for 1) minimizing logical topology augmentation for weak survivability, 2) survivable routing for maximizing the logical topology capacity, 3) survivable routing for maximizing the connectivity after a single physical link failure, and 4) maximizing the minimal cross-layer cutset.

A. Experimental Setup

The testing cases for both exact solution approaches (MILP formulations) and heuristic algorithms are as follows. We adopted the networks introduced in [38,39] as physical topologies. They are DFN, EURO1 (E1), EURO2 (E2), G3, G6, and NSF, as shown in Figs. 2–7. Two groups of logical topologies are constructed: a) 2-connected planar graphs and b) the more sparse networks, Delaunay triangulation, and spanning tree, denoted as "2CON," "DELA," and "TREE," respectively.

DELA and TREE are used to test the topology augmentation MILP and the heuristic. Two-connected planar graphs are used to test all metrics. Note here that links are randomly removed from 2-connected and Delaunay

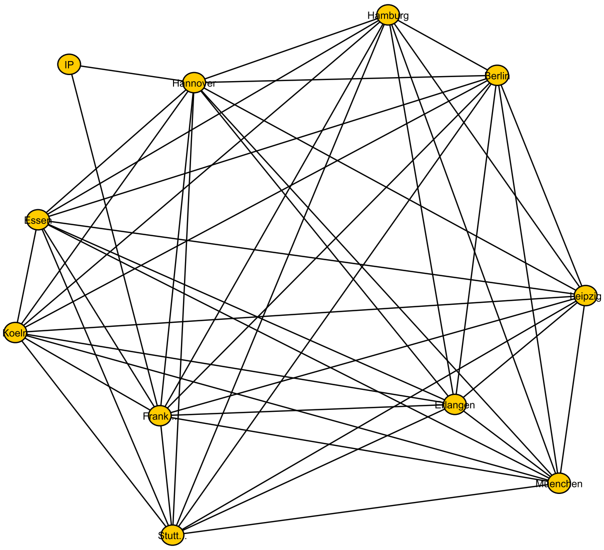


Fig. 2. DFN.

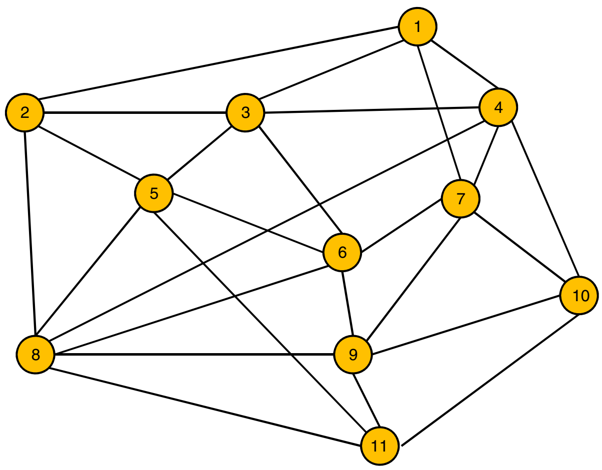


Fig. 3. EURO 1.

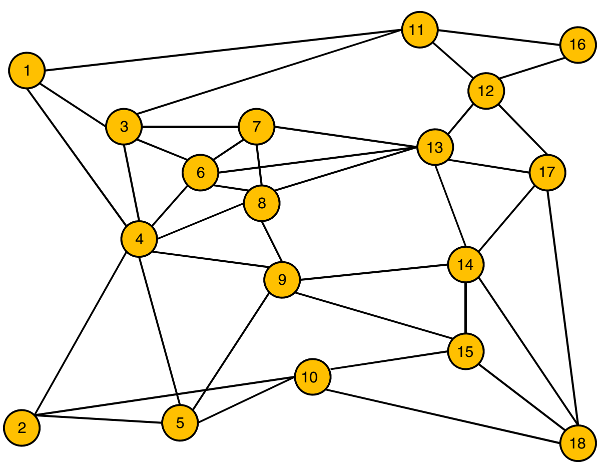


Fig. 4. EURO 2.

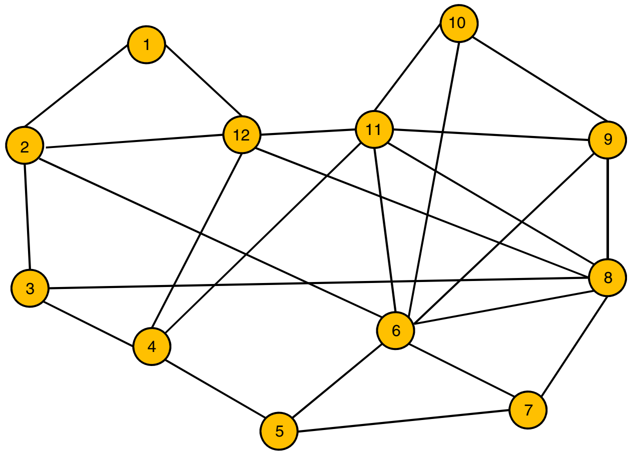


Fig. 5. G3.

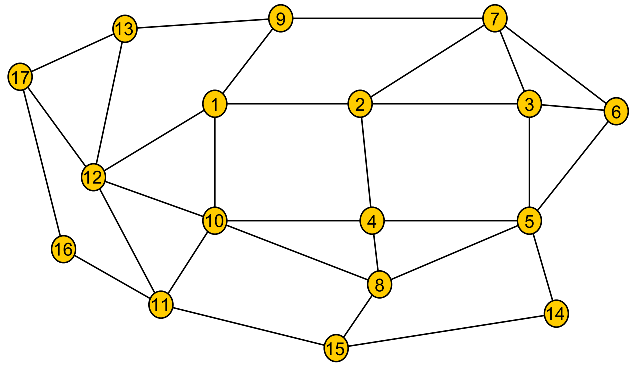


Fig. 6. G6.

triangulation logical topologies to generate the tested logical networks. For all constructed logical topologies, the logical topologies' nodes are subsets of physical nodes; that is, $|V_L| = \lfloor 0.5 * |V_P| \rfloor$ and $\lfloor 0.7 * |V_P| \rfloor$. We let $r_{L/P}$ denote the "logical to physical node ratio." We used $r_{L/P} = 0.5$ or 0.7. Detailed information for physical and logical topologies is

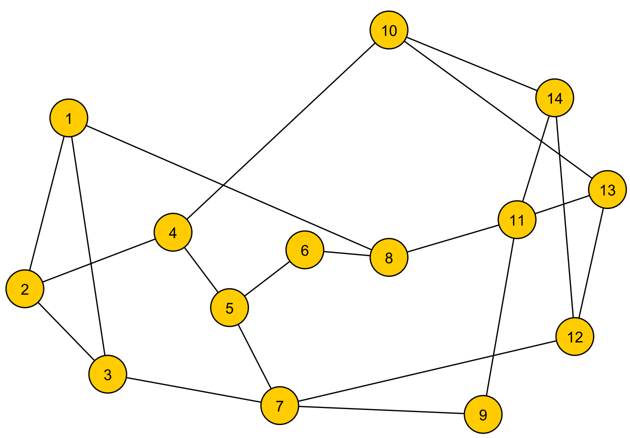


Fig. 7. NSF.

TABLE II
Physical Topology Information

PhyG	Nodes	Edges	MinDeg	MaxDeg	AvgDeg
DFN	11	47	2	10	8.55
E1	11	26	4	6	4.73
E2	18	39	2	7	4.33
G3	12	25	2	7	4.17
G6	17	31	2	5	3.65
NSF	14	21	2	4	3

listed in Tables II and III. In all tested cases, capacities and demands assigned to physical and logical links are randomly generated following uniform distributions in [1,10] and [1,100], respectively. We run 10 cases for each tested scenario and report the average performance of these 10 cases.

We used CPLEX 12.4 to run the MILP formulations on a machine with a quad-core (with hyper-threading) Intel Core i7-3770K processor and 16 GB memory. We assigned four threads to solve each MILP problem and limited the maximum execution time to 30 min. The heuristics were implemented using the LEMON library [40], which ran in a single thread during execution.

TABLE III
Logical Topology Information

PhyG	LogG	$r_{L/P}$	V_L	E_L	MinDeg	MaxDeg	AvgDeg
DFN	2CON	0.5	6	7	2	4	2.33
		0.7	8	10	2	4	2.5
	DELA	0.5	6	47	5	31	15.67
		0.7	8	47	10	13	11.75
	TREE	0.5	6	5	1	2	1.67
		0.7	8	7	1	3	1.75
E1	2CON	0.5	6	7	2	4	2.33
		0.7	8	10	2	6	2.5
	DELA	0.5	6	26	5	12	8.67
		0.7	8	26	5	9	6.5
	TREE	0.5	6	5	1	3	1.67
		0.7	8	7	1	3	1.75
E2	2CON	0.5	9	11	2	4	2.44
		0.7	13	15	2	4	2.31
	DELA	0.5	9	39	7	11	8.67
		0.7	13	39	3	9	6
	TREE	0.5	9	8	1	3	1.78
		0.7	13	12	1	3	1.85
G3	2CON	0.5	6	7	2	4	2.33
		0.7	8	8	2	2	2
	DELA	0.5	6	25	5	11	8.33
		0.7	8	25	5	8	6.25
	TREE	0.5	6	5	1	2	1.67
		0.7	8	7	1	3	1.75
G6	2CON	0.5	9	9	2	2	2
		0.7	12	14	2	4	2.33
	DELA	0.5	9	31	3	10	6.89
		0.7	12	31	3	7	5.17
	TREE	0.5	9	8	1	2	1.78
		0.7	12	11	1	2	1.83
NSF	2CON	0.5	7	8	2	4	2.29
		0.7	10	10	2	2	2
	DELA	0.5	7	21	5	7	6
		0.7	10	21	3	8	4.2
	TREE	0.5	7	6	1	2	1.71
		0.7	10	9	1	3	1.8

B. Experimental Results

First, we present the results for the minimizing logical network augmentation. Note that logical network augmentation is triggered only when a cross-layer survivable route cannot be generated. We consider DELA and TREE as logical networks, which are less dense than 2CON. The performance results for the MILP formulation and the heuristic for minimizing logical network augmentation for weak survivability are presented in Table IV. We let "PhyG," "LogG," "$r_{L/P}$," "AugEdge," and "Time" represent the tested physical network, the logical network, the ratio of the number of logical nodes over that of physical nodes, the number of augmented logical edges, and the computational time (in seconds). In all cases, the WSR-MLA formulation does not augment the logical network. The computational time of the heuristic is much less than that in the MILP approach. We note that for the same physical network, the denser the logical graph (as measured by the ratio m/n^2), the lower the number of edges required for augmentation. This is because a denser graph can afford to tolerate more link failures before it becomes

TABLE IV
Results for Minimizing Augmented Logical Edges for Weak Survivability

			MILP		Heuristics	
PhyG	LogG	$r_{L/P}$	AugEdge	Time (s)	AugEdge	Time (s)
DFN	DELA	0.5	0	0.0696	1	4E-04
		0.7	0	0.0965	0.5	6E-04
	TREE	0.5	2	0.2474	6	3E-04
		0.7	2.8	2.9485	6	4E-04
E1	DELA	0.5	0	0.0535	0.9	3E-04
		0.7	0	0.0741	0.8	4E-04
	TREE	0.5	2	0.1341	4.2	2E-04
		0.7	2	0.4021	3.8	3E-04
E2	DELA	0.5	0	0.3593	0.7	6E-04
		0.7	0	0.5092	1.3	8E-04
	TREE	0.5	2	1.5197	5.3	4E-04
		0.7	3	5.8057	8.5	6E-04
G3	DELA	0.5	0	0.0738	0.9	3E-04
		0.7	0	0.0934	0.6	4E-04
	TREE	0.5	1.4	0.1687	5	2E-04
		0.7	2	0.4438	6.3	3E-04
G6	DELA	0.5	0	0.2942	0.6	6E-04
		0.7	0	0.3771	1.4	7E-04
	TREE	0.5	2	1.4556	7.1	4E-04
		0.7	2.2	27.7924	7.1	5E-04
NSF	DELA	0.5	0	0.1293	0.4	3E-04
		0.7	0	0.1750	0.8	5E-04
	TREE	0.5	2	0.3120	5.6	3E-04
		0.7	2.5	13.0718	7.2	4E-04

TABLE V
Computation Results With 2-Connected Planar Logical Topologies and Logical to Physical Node Ratio = 0.5

	MinAug				MaxCapa				MaxConn				MaxMCLC			
	MILP		Heuristic		MILP		Heuristic		MILP		Heuristic		MILP		Heuristic	
PhyG	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time
DFN	0.0	0.078	3.0	3E-04	3.5	2.727	1.3	6E-04	2	0.127	1	0.003	2	3.043	1	0.003
E1	0.0	0.077	4.2	2E-04	2.7	0.984	1.2	5E-04	2	0.083	1	0.002	2	0.831	1	0.002
E2	0.0	1.182	4.1	4E-04	3.3	6.184	1.3	0.001	2	0.299	1	0.003	2	9.268	1	0.003
G3	0.0	0.097	2.2	2E-04	5.3	0.967	1.3	5E-04	2	0.104	1	0.002	2	1.681	1	0.002
G6	0.2	1.244	3.8	4E-04	3.6	5.274	1.1	9E-04	1	0.348	1	0.003	2	71.62	1	0.003
NSF	0.3	0.314	3.6	2E-04	3.0	1.149	0.8	7E-04	2	0.127	1	0.002	2	1.177	1	0.002

TABLE VI
Computation Results With 2-Connected Planar Logical Topologies and Logical to Physical Node Ratio = 0.7

	MinAug				MaxCapa				MaxConn				MaxMCLC			
	MILP		Heuristic		MILP		Heuristic		MILP		Heuristic		MILP		Heuristic	
PhyG	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time
DFN	0.0	0.14	3.6	3E-04	3.5	4.325	1.2	8E-04	2	0.213	1	0.003	2	6.802	1	0.003
E1	0.0	0.282	2.2	3E-04	4.0	2.184	1.3	6E-04	2	0.15	1	0.002	2	3.206	1	0.003
E2	0.1	185	4.7	6E-04	2.3	33.56	1.0	0.001	2	4.024	1	0.004	2	613.6	1	0.004
G3	0.0	0.143	6.4	3E-04	3.7	1.247	1.4	7E-04	2	0.19	1	0.002	2	16.76	1	0.002
G6	0.7	8.48	5.9	5E-04	3.3	15.3	0.8	0.001	1	0.398	1	0.004	1	11.66	1	0.004
NSF	0.9	6.423	6.4	4E-04	4.0	3.449	0.9	8E-04	2	0.257	1	0.002	xx	xx	1	0.003

disconnected. This can be seen from the simulation results. In fact, DELA, which is denser than TREE, requires no augmentation at all.

Next, we present the performance of the MILPs and heuristics for all four cross-layer evaluation metrics with 2CON (with logical physical node ratio, 0.5 and 0.7) as logical topologies. Computational results are reported in Tables V and VI. For all testing cases, heuristic algorithms ran significantly faster than exact solution approaches (MILP formulations). For all evaluation metrics except augmentation, we report the average computational time of MILPs that do not require logical augmentation. As we can see, the heuristics are much faster than the MILPs. In general the computational times increase as the difficulty increases in the order "MinAug," "MaxCon," and "MaxMCLC." Also, to achieve feasible solutions for these four evaluation metrics, the realization of feasible solutions becomes harder and harder.

VIII. Summary and Conclusions

In this paper we have presented a comprehensive treatment of mathematical programming frameworks for the survivable logical topology routing problem in capacitated cross-layer optical networks under multiple cross-layer evaluation metrics. We have enhanced the survivability routing formulation WSR-MD given in Section III by developing MILP formulations for 1) minimum logical topology augmentation for guaranteed weakly survivable routing, 2) maximizing the after-failure connectivity of the logical topology, 3) maximizing the capacity of the logical topology, and 4) maximizing the MCLC. An interesting feature of our MILP formulations is that the optimization is carried out in a single stage, in contrast to previous approaches that consider logical subgraphs obtained after each physical link failure. For example, see [7]. The contributions reported in this paper assume considerable significance in view of the increasing interest in network virtualization and topology abstraction incorporating survivability requirements [41,42].

References

[1] J. Sakaguchi, B. Puttnam, W. Klaus, Y. Awaji, N. Wada, A. Kanno, T. Kawanishi, K. Imamura, H. Inaba, K. Mukasa, R. Sugizaki, T. Kobayashi, and M. Watanabe, "305 Tb/s space division multiplexed transmission using homogeneous 19-core fiber," *J. Lightwave Technol.*, vol. 31, no. 4, pp. 554–562, Feb. 2013.

[2] 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.

[3] 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. Int. Workshop on the Design of Reliable Communication Networks (DRCN)*, Krakow, Poland, Oct. 2011, pp. 1–8.

[4] 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.*, vol. 7, no. 4, pp. 206–214, 2010.

[5] C. Liu and L. Ruan, "A new survivable mapping problem in IP-over-WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 25–34, 2007.

[6] D. D.-J. Kan, A. Narula-Tam, and E. Modiano, "Lightpath routing and capacity assignment for survivable IP-over-WDM networks," in *Proc. Int. Workshop on Design of Reliable Communication Networks (DRCN)*, Washington, DC, Oct. 2009, pp. 37–44.

[7] K. Lee and E. Modiano, "Cross-layer survivability in WDM-based networks," in *IEEE Int. Conf. on Computer Communications (IEEE INFOCOM)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1017–1025.

[8] A. Agrawal and R. E. Barlow, "A survey of network reliability and domination theory," *Oper. Res.*, vol. 32, pp. 478–492, 1984.

[9] A. Rosenthal and D. Frisque, "Transformations for simplifying network reliability calculations," *Networks*, vol. 7, pp. 97–111, 1977.

[10] A. Shooman and A. Kershenbaum, "Exact graph-reduction algorithms for network reliability analysis," in *IEEE Global Communications Conf. (IEEE GLOBECOM)*, Phoenix, AZ, Dec. 1991, pp. 1412–1420.

[11] A. Shooman and A. Kershenbaum, "Methods for communication network reliability analysis: Probabilistic graph reduction," in *Annu. Reliability and Maintainability Symp.*, Las Vegas, NV, Jan. 1992, pp. 441–448.

[12] F. Dikbiyik, M. Tornatore, and B. Mukherjee, "Exploiting excess capacity for survivable traffic grooming in optical backbone networks," *J. Opt. Commun. Netw.*, vol. 6, no. 2, pp. 127–137, Feb. 2014.

[13] R. Karp and M. Luby, "Monte-Carlo algorithms for the planar multiterminal network reliability problem," *J. Complexity*, vol. 1, pp. 45–64, 1985.

[14] D. Karger, "A randomized fully polynomial time approximation scheme for the all terminal reliability problem," *SIAM Rev.*, vol. 43, pp. 499–522, 2001.

[15] C. Colbourn, *The Combinatorics of Network Reliability.* Oxford University, 1987.

[16] S. C. Liew and K. W. Lu, "A framework for characterizing disaster-based network survivability," *IEEE J. Sel. Areas Commun.*, vol. 12, pp. 52–58, 1994.

[17] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," *IEEE/ACM Trans. Netw.*, vol. 19, pp. 1610–1623, 2011.

[18] R. M. Karp, *Reducibility Among Combinatorial Problems*. New York: Plenum, 1972, ch. 8, pp. 85–103.

[19] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," in *IEEE Int. Conf. on Computer (IEEE INFOCOM)*, Anchorage, AK, Apr. 22, 2001, pp. 348–357.

[20] 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.

[21] J. Strand, A. L. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 81–87, Feb. 2001.

[22] C. Vadrevu, M. Tornatore, R. Wang, and B. Mukherjee, "Integrated design for backup capacity sharing between IP and wavelength services in IP-over-WDM networks," *J. Opt. Commun. Netw.*, vol. 4, no. 1, pp. 53–65, Jan. 2012.

[23] M. Kurant and P. Thiran, "Survivable mapping algorithm by ring trimming (SMART) for large IP-over-WDM networks," in *Proc. Int. Conf. on Broadband Networks (BROADNETS)*, San Jose, CA, 2004, pp. 44–53.

[24] S. Lee, J.-C. Liu, and Y. Chen, "An ear-decomposition based approach for survivable routing in WDM networks," in *Proc. Int. Conf. on Advanced Information Networking and Applications (AINA)*, New Taipei City, Taiwan, Mar. 2005, vol. 1, pp. 459–464.

[25] M. S. Javed, K. Thulasiraman, M. A. Gaines, and G. Xue, "Survivability aware routing of logical topologies: On Thiran-Kurant approach, enhancements and evaluation," in *IEEE Global Communications Conf. (IEEE GLOBECOM)*, San Francisco, CA, 2006.

[26] M. Javed, K. Thulasiraman, and G. Xue, "Lightpaths routing for single link failure survivability in IP-over-WDM networks," *J. Commun. Netw.*, vol. 9, no. 4, pp. 394–401, 2007.

[27] 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 *IEEE Int. Conf. on Computer Communications (IEEE INFOCOM)*, Rio de Janeiro, Brazil, 2009, pp. 1026–1034.

[28] 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. Int. Conf. on Communication Systems and Networks (COMSNETS)*, Bangalore, India, 2010, pp. 128–137.

[29] T. Lin, K. Thulasiraman, and G. Xue, "Robustness of logical topology mapping problems for survivability under multiple physical links failures in IP over WDM optical networks," in *Proc. Int. Conf. on Communication Systems and Networks (COMSNETS)*, Bangalore, India, Jan. 2012.

[30] 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 *4th Int. Congr. on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, St. Petersburg, Russia, 2012, pp. 650–656.

[31] 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 *5th Int. Workshop on Reliable Networks Design and Modeling (RNDM)*, Almaty, Kazakhstan, Sept. 2013, pp. 82–88.

[32] B. Jaumard and H. A. Hoang, "Design and dimensioning of logical survivable topologies against multiple failures," *J. Opt. Commun. Netw.*, vol. 5, no. 1, pp. 23–36, Jan. 2013.

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

[34] H. Yu, V. Anand, and C. Qiao, "Virtual infrastructure design for surviving physical link failures," *Comput. J.*, vol. 55, no. 8, pp. 965–978, 2012.

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

[36] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms.* Wiley-Interscience, 1992.

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

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

[39] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, “SNDlib 1.0–survivable network design library,” in *Proc. 3rd Int. Network Optimization Conf. (INOC)*, Spa, Belgium, Apr. 2007, vol. 55, pp. 278–286.

[40] Library for Efficient Modeling and Optimization in Networks (LEMON) [Online]. Available: http://lemon.cs.elte.hu/trac/lemon.

[41] M. 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, June 2013.

[42] R. Ravindran, C. Huang, and K. Thulasiraman, “Topology abstraction service for IP-VPNs,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, pp. 184–197, Jan. 2013.

Zhili Zhou (M’11) received her Ph.D. degree in industrial and system engineering from the University of Florida, Gainesville, Florida, USA, in 2010, and B.S. and M.S. degrees in mathematics from Nanjing University, Nanjing, China, in 2003 and 2006, respectively. She is a research staff member at the IBM Research Collaboratory, Singapore. Dr. Zhou took second place for the 2011 Pritsker Doctoral Dissertation Award from the Institute of Industrial Engineers for her dissertation. She was the Best Paper Award Runner-up of DRCN’2011 and RNDM’2012. Her research interests are in mixed integer programming, combinatorial optimization, stochastic programming, and quality of service in communication networks, transportation networks, and power grids. Dr. Zhou is a member of the IEEE and INFORMS.

Tachun Lin (M’08) received the Ph.D. degree in computer science from the University of Oklahoma, Norman, Oklahoma, USA, in 2011, and the B.S. and M.S. degrees in computer science from the National Chiao Tung University, Hsinchu, Taiwan, in 1999 and 2001, respectively. He is an Assistant Professor in the Department of Computer Science and Information Systems, Bradley University. He received the Best Paper Runner-up Award for his papers at DRCN 2011 and RNDM 2012. His research interests include interdependent network design, network function virtualization, network optimization, and graph theory. Dr. Lin is a member of the ACM, IEEE, and INFORMS.

Krishnaiyan Thulasiraman (M’72-SM’84-F’90) received his Ph.D. degree from the Indian Institute of Technology (IITM), Madras, India, in 1968. Since 1994 he has held the Hitachi Chair and is Professor with the School of Computer Science, University of Oklahoma, Norman. Prior to joining the University of Oklahoma, he was a Professor (1981–1994) and Chair (1993–1994) of the Electrical and Computer Engineering Department, Concordia University, Montreal, Canada. He has published more than 100 papers in archival journals. He has coauthored with M. N. S. Swamy *Graphs, Networks, and Algorithms* (Wiley, 1981) and *Graphs: Theory and Algorithms* (Wiley, 1992) and several book chapters. He has a U.S. patent on distributed QoS routing. His research interests have been in graph theory, combinatorial optimization, algorithms, and applications in a variety of areas in computer science and electrical engineering, particularly fault diagnosis, tolerance, and testing. His most recent interest is in the emerging area of network science and engineering. Dr. Thulasiraman has received several awards and honors: Distinguished Alumnus Award of IIT Madras, IEEE Circuits and Systems Society Technical Achievement Award, IEEE CAS Society Golden Jubilee Medal, Senior Fellow of the Japan Society for Promotion of Science, Gopalakrishnan Endowed Chair Professor of IIT Madras, and Fellow of the IEEE, AAAS, and the European Academy of Sciences.

Guoliang Xue (M’98-SM’99–F’11) received the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, Minnesota, USA, in 1991. He is a Professor of computer science and engineering with Arizona State University, Tempe, Arizona, USA. His research interests include quality of service provisioning, resource allocation, survivability, security, and privacy issues in networks. He has received Best Paper Awards at IEEE GLOBECOM 2007, IEEE ICC 2011, IEEE MASS 2011, IEEE GLOBECOM 2011, and IEEE ICC 2012, and a Best Paper Runner-up at IEEE ICNP 2010. He is an Associate Editor of *IEEE/ACM Transactions on Networking* and *IEEE Network*, and a member of the Standing Committee of IEEE INFOCOM. He was General Co-Chair of the IEEE Conference on Communications and Network Security (IEEE CNS 2014). He was a Keynote Speaker at IEEE LCN 2011, a TPC Co-Chair of IEEE INFOCOM 2010, and a Co-Organizer of the ARO Workshop on Human Centric Computing with Collective Intelligence—Challenges and Research Directions (HC3I 2012). He served on the editorial boards of *Computer Networks*, *IEEE Transactions on Wireless Communications*, and *IEEE Transactions on Circuits and Systems*. He was a Distinguished Lecturer of the IEEE Communications Society. He is an IEEE Fellow.

Sartaj Sahni (M’78-SM’86–F’88) is a Distinguished Professor of Computer and Information Sciences and Engineering at the University of Florida. He is also a member of the European Academy of Sciences, a Fellow of the IEEE, ACM, AAAS, and Minnesota Supercomputer Institute, and a Distinguished Alumnus of the Indian Institute of Technology, Kanpur. In 1997, he was awarded the IEEE Computer Society Taylor L. Booth Education Award “for contributions to Computer Science and Engineering education in the areas of data structures, algorithms, and parallel algorithms,” and in 2003, he was awarded the IEEE Computer Society W. Wallace McDowell Award “for contributions to the theory of NP-hard and NP-complete problems.” Dr. Sahni was awarded the 2003 ACM Karl Karlstrom Outstanding Educator Award. Dr. Sahni has published more than 300 research papers and written 15 texts. His research publications are on the design and analysis of efficient algorithms, parallel computing, interconnection networks, design automation, and medical algorithms.