Robustness of Logical Topology Mapping Algorithms for Survivability against Multiple Failures in an IP-over-WDM Optical Network

Krishnaiyan Thulasiraman, Tachun Lin
School of Computer Science
University of Oklahoma
Norman, OK 73019
Email: {thulasi, djlin}@ou.edu

Zhili Zhou
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
Email: zzhou@us.ibm.com

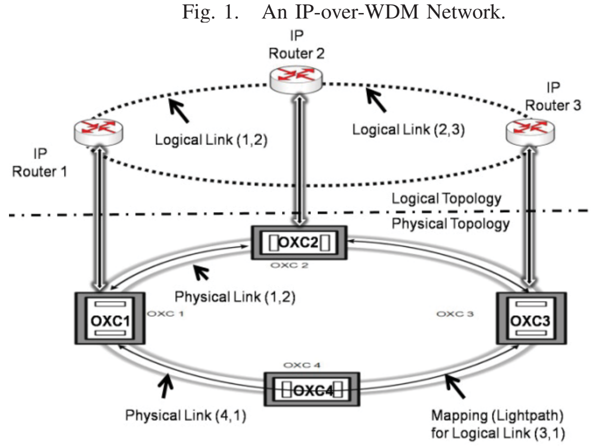
Guoliang Xue
Arizona State University
Tempe, AZ, USA
Email: xue@asu.edu

Abstract*—The survivable logical topology mapping (SLTM) problem in an IP-over-WDM optical network is to map each link (u, v) in the logical topology G_L (at the IP layer) into a lightpath between the nodes u and v in the physical topology G_P (at the optical layer) such that failure of a physical link does not cause the logical topology to become disconnected. It is assumed that both the physical and logical topologies are 2-edge connected. There are two lines of approach for the study of the SLTM problem. One approach uses Integer Linear Programming formulations. The main drawback with this approach is the use of exponential number of constraints, one for each cutset in G_L. Moreover, it does not provide insight into the solution when survivability against multiple physical failures is required. The other approach, called the structural approach, uses graph theory and was pioneered by Kurant and Thiran and further generalized by us. In this paper we first present a generalized algorithmic framework for the SLTM problem. This framework includes several other frameworks considered in earlier works as special cases. We then define the concept of robustness of a mapping algorithm which captures the ability of the algorithm to provide survivability against multiple physical failures. This is similar to the concept of fault coverage used in hardware/software testing. We analyse the different frameworks for their robustness property. Using simulations, we demonstrate that even when an algorithm cannot be guaranteed to provide survivability against multiple failures, its robustness could be very high. The work also provides a basis for the design of survivability mapping algorithms when special classes of failures such as SRLG failures are to be protected against.

I. Introduction

An IP-over-WDM network implements *Internet Protocol* (IP) directly over a *Wavelength Division Multiplexing* (WDM) network by mapping a set of given IP connections as *lightpaths* in the WDM network [1]. A lightpath is an all optical connection established by finding a path between the source and the destination of an IP connection in the WDM network and assigning it a wavelength [2]. Such networks use OXCs to switch network traffic (lightpaths) in the WDM layer and IP routers to route/reroute IP connections at the IP layer. The set of IP connections form the *logical topology* and OXCs along with actual optical fibers form the *physical topology*. In the literature, it is common to refer to IP connections as *IP* or *logical links (edges)*, IP routers as *logical nodes (vertices)*, OXCs as *physical nodes* and set of fibers (a cable) connecting the OXCs as *physical links*. Fig. 1 shows a typical implementation of an IP-over-WDM network.

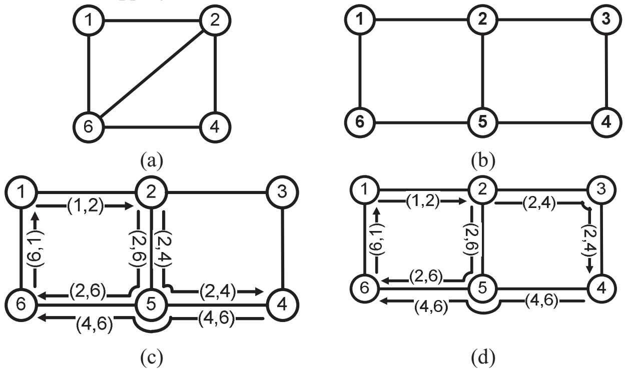
Fig. 1. An IP-over-WDM Network.



An optical fiber simultaneously carries several lightpaths. Therefore, the failure of an optical fiber disconnects all the carried lightpaths, causing multiple failures in the logical topology, which severely impacts the entire network performance. Mechanisms that allow networks to deliver an acceptable level of service in the presence of a failure or failures are referred to as *survivability mechanisms* and IP-over-WDM networks that implement such mechanisms are called *survivable IP-over-WDM networks* (henceforth, simply *survivable networks*).

The two widely discussed survivability mechanisms in the literature are *protection* and *restoration* [1]. Protection is generally provided at the physical layer but can be implemented at the logical layer also. It requires a dedicated *backup lightpath* for each *working lightpath* such that the two lightpaths are disjoint. The backup path is used only when the working lightpath fails. It is always possible to find two disjoint lightpaths, if the physical topology is at least 2-edge connected [3]. Restoration is usually provided at the logical layer by setting up working lightpaths for the IP connections and then provisioning the

Fig. 2. Illustration of mapping and survivability for general networks. (a) A logical topology (b) A physical topology (c) An unsurvivable mapping (d) A survivable mapping.



physical network with some additional (spare) capacity that is used by the IP routers to find backup lightpaths for the failed working lightpaths. However, backup paths can be guaranteed only if the IP topology is initially embedded in such a way that it stays connected after a failure. This leads to the study of the survivable logical topology mapping problem.

The *Survivable Logical Topology Mapping* (SLTM) problem in an IP-over-WDM optical network is to map each link (u, v) in the logical topology (at the IP layer) into a lightpath between the nodes u and v in the physical topology (at the optical layer) such that failure of one or more physical links does not cause the logical topology to become disconnected. It is assumed that both the physical and logical topologies are 2-edge connected (in short, two-connected).

Fig. 2(a) and Fig. 2(b) show a logical topology and a physical topology, respectively. Fig. 2(c) shows an unsurvivable mapping of this logical topology. In this case, not all the mappings are disjoint and the logical topology is not survivable. For example, the failure of physical link (4, 5) disconnects the logical topology. Fig. 2(d) shows a survivable mapping. In this case also, it can be seen that not all the mappings are disjoint and a physical link failure may disconnect multiple logical links but the logical topology still remains connected. For example, if the physical link (5, 6) fails, logical links (2, 6) and (4, 6) get disconnected but it is possible to reach all the logical nodes through the remaining logical links. It can be observed that finding disjoint mappings for only the subset (1, 2), (2, 4), (4, 6), (6, 1) is sufficient to guarantee survivability in this example. The question then arises as to how to select the groups of logical links to be mapped into disjoint paths. This has been the course of much of the research in this area.

The above example illustrates the important role played by the pair-wise (mutually) disjoint paths problem in finding survivable mappings. The problem of finding pair-wise disjoints paths is well studied and is NP-complete in general [4]. However, it is possible to find pair-wise disjoint paths in some special cases e.g. when the physical topology is undirected and three edge-connected, and the number of pair-wise disjoint paths is two [5].

II. Related Work

There are two lines of approach for the study of the SLTM problem. One approach uses Integer Linear Programming formulations. The main drawback with this approach is the use of exponential number of constraints, one for each cutset in G_L. The other approach uses graph theory and provides a basis for a structural study of the problem. The *Integer Linear Programming* (ILP) approach was initiated by Modiano et al. In [6][7], Modiano and Narula-Tam formally show that the problem of finding survivable mappings is NP-complete for general as well as for ring logical topologies. Therefore, they provide *Integer Linear Programs* (ILPs) to find a solution. The ILP is based on the observation that a logical topology can get disconnected after the failure of a physical link only if the physical link carries an entire cut of the logical topology, or alternatively, every cut of the logical topology must contain at least a pair of edges with pair-wise disjoint mappings in order for the mappings to be survivable. However, the ILP does not scale well as it must examine all the possible cuts, a number that grow exponentially with the size of the topology. In [8] Todimala and Ramamurthy, based on [6][7], provide an improved ILP that applies to *Shared Risk Link Group* (SRLG). The ILP incorporates wavelength assignment constraints and only considers primary cuts, but does not scale well either. However, when applied to planar cycles and hierarchical planar cycles, the ILP can be solved fairly quickly. In [9] Ducatelle and Gambardella also utilize the results from [6][7] and, rather than evaluating all the cutsets, employ a probability function as an estimate of the cutsets. Crochat et al. provide in [10] a comprehensive framework for the logical topology mapping problem in IP-over-WDM networks and define three constraints (that include survivability) that must be respected by a solution. They note that the problem is NP-complete and suggest a heuristic based on Tabu search. Shenai and Sivalingam suggest in [11] a hybrid approach to survivability that uses a combination of restoration and protection.

In [12], certain metrics are defined that capture the quality of a lightpath routing. Specifically, the concept of *Minimum Cross Layer Cut* (MCLC) is defined in this paper. This metric is a measure of the ability of a routing to tolerate multiple physical edge failures. Finding a routing that maximizes MCLC is also intractable. An ILP formulation to find a survivable routing that maximizes a measure that is related to MCLC is given in [12]. In [13], among other things, ILP formulations to find a routing that minimizes spare capacity requirements is presented. In a recent work [14], we have considered the general case of capacitated optical networks with capacities on physical edges and demands on logical edges. We have presented MILP (Mixed Integer Linear Programming) formulations and heuristics to generate a survivable routing that maximizes the logical topology capacity and minimizes spare capacity requirements.

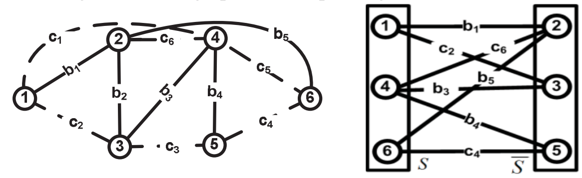
The structural approach to the cross-layer survivability was initiated by Kurant and Thiran (KT) in [15] and was later extended and generalized by us in [16] and [17]. The KT approach can be stated using the concept of a generalized ear decomposition of a graph. An ear decomposition of G_L is a decomposition of the edge set of G_L into a sequence of subgraphs (called circuit ears) $C_0, C_1, C_2, \ldots, C_k$ such that C_0 is a circuit in G_L and each $C_i, i > 0$, is a circuit in the graph $G_L^{'}$ which is obtained by contracting all the edges in $C_0, C_1, C_2, \ldots, C_{i-1}$. The subgraphs $C_0, C_1, C_2, \ldots, C_k$ are called the circuit ears of the ear decomposition. It is shown in [15][16] that a routing is survivable if all the edges in every circuit ear of size at least two are associated with edge-disjoint lightpaths. It is also shown that if no such ear decomposition is available then the there exists no survivable routing for the given G_L. The algorithm for survivable routing based on this result is called CIRCUIT-SMART. In [16], the dual concept of cutest ear decomposition is defined and algorithm CUTSET-SMART is presented. This paper also presents algorithm INCIDENCE-SMART based on the concept of incidence sets that are special cases of cutsets. A drawback of CUTSET-SMART is discussed and resolved in [17]. In this paper it is shown that when circuit/cutest ear decompositions are appropriately selected the distinction between CIRCUIT-SMART and CUTSET-SMART disappears. Note that circuit and cutest ear decompositions specify an order for the selection of subgraphs in these decompositions. The number of ears and sizes of ears in these decompositions play a vital role in the ability of SMART-based algorithms to tolerate multiple physical edge failures. In this context, several interesting issues arise and they are under investigation.

The SLTM problem requires mapping logical links into mutually disjoint paths in the physical topology. It is likely that such paths may not exist unless the physical topology satisfies certain connectivity requirements. In such cases, we may have to augment the logical topology with additional logical links to guarantee that the augmented topology admits a survivable mapping. This augmentation problem has been studied in [18] [19].

Most of the research on the SLTM problem has focused on survivability against a single physical link failure. Usually, survivability against multiple failures is achieved by providing multiple pair-wise link disjoint paths between the nodes of each logical link. Such an approach results in provisioning of excessive spare capacity and is also not theoretically satisfying.

In this paper we study the SLTM problem for the multiple failure scenario. Work in this paper is based on the concepts and results presented in our earlier works [16][17]. So, we give a brief discussion of these concepts in Section III. In Section IV we present an algorithmic framework called GEN-SMART which includes as special cases earlier frameworks for the SLTM problem. In Section V we define the concept of robustness of a logical mapping algorithm that captures the ability of the algorithm to provide survivability against multiple failures. This is similar to the concept of fault coverage used in hardware/software testing. We analyse GEN-SMART and its special cases for their robustness property. We conclude in Section VI with simulations that demonstrate that even when an algorithm cannot be guaranteed to provide survivability against multiple failures, its robustness could be very high. The work also provides a basis for the design of survivability mapping algorithms when special classes of failures such as SRLG failures are to be protected against.

Fig. 3. (a) A graph with a spanning tree (bold lines) (b) A cut.



(a) (b)

III. Basic Concepts

Consider a connected undirected graph $G(V, E)$ with vertex set V and edge set E. Without loss of generality, we assume that there are no parallel edges or self loops in G. Let G have $|V| = n$ vertices (or nodes) and $|E| = m$ edges (or links).

A connected acyclic subgraph of G containing all the n nodes is called a *spanning tree* T of G. The edges of a spanning tree T are called *branches* of T. The remaining edges of G are called *chords* with respect to T. We may also refer to chords as *non-tree edges*.

Consider a partition $(S, \overline{S})$ of vertex set V. Here $\overline{S}$ denotes the complement of S in V, i.e. $\overline{S} = V - S$. Then the set of edges with one node in S and the other in $\overline{S}$ is called a *cut* of G.

For example, consider the graph G in Fig. 3(a). Here the vertices are numbered $1, 2, \ldots, 6$. The bold edges in this figure denote the branches of a spanning tree T of G and the dotted edges are the chords of this tree. The partition $(S, \overline{S})$ with $S = \{1, 4, 6\}$ and $\overline{S} = \{2, 3, 5\}$ defines the cut shown in Fig. 3(b).

Adding a chord c to a spanning tree T produces exactly one circuit. This is called the *fundamental circuit* (in short, f-circuit) of T with respect to the chord c. We denote this circuit as $B(c)$. For example, if we add chord c_1 to the tree in Fig. 3(a) we get the fundamental circuit $B(c_1)$ consisting of the edges $\{c_1, b_1, b_2, b_3\}$.

Suppose we remove a branch b from a spanning tree T, then the tree T gets disconnected resulting in two trees (not spanning) T_1 and T_2. The sets of nodes in T_1 and T_2 define a partition of V. The corresponding cut $Q(b)$ is called the *fundamental cutset* (in short, f-cutset) of T with respect to branch b. For example, if we remove the branch b_3 from the tree T of Fig. 3(a) then we get trees T_1 and T_2 given by branches $\{b_1, b_2, b_5\}$ and $\{b_4\}$ respectively. The corresponding fundamental cutset $Q(b_3)$ consists of the edges $\{b_3, c_1, c_3, c_4, c_5, c_6\}$. Note that the subgraphs induced by the vertex sets of T_1 and T_2 are both connected. Cuts with this property are also called primary cuts [8].

Circuits and cutsets are dual concepts. Whereas the CIRCUIT-SMART algorithm of Kurant and Thiran [16] is based on circuits, the CUTSET-SMART based algorithms in [16] are based on cutsets. The distinction between these two classes of these algorithms for the SLTM problem are lost when the circuits or cutsets are selected in a special way. So, in the following we presents only those concepts and results related to cutset.

The *fundamental cutset matrix* with respect to the tree T can be defined as $Q_f = [q_{ij}]_{(n-1)*(m)}$. Q_f has $(n-1)$ rows, one for each fundamental cutset, and m columns, one for each edge. The entry q_{ij} is defined as

$$\begin{aligned} q_{ij} \quad &= 1, \text{if } Q(b_i) \text{ contains edge } j \\ &= 0, \text{otherwise.} \end{aligned}$$

Arranging the rows of Q_f such that the j^{th} row corresponds to f-cutset $Q(b_j)$ and arranging the columns to correspond to edges in the order $\{b_1, b_2, \ldots, b_{n-1}, c_1, c_2, \ldots, c_{m-n+1}\}$ the Q_f matrix can be written as $Q_f = [U|Q_{f_c}]$. For example, the Q_f matrix with respect to the tree T of Fig. 3(a) is given in (1).

$$\begin{array}{c} \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{array} \begin{array}{c} \begin{array}{ccccc|cccccc} b_1 & b_2 & b_3 & b_4 & b_5 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\ \left[\begin{array}{ccccc|cccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{array} \right] \end{array} \tag{1}$$

An ordered sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$ is a *cutset cover sequence* or simply a $Q-sequence$ of length k if

$a)$ $\left[Q(b_j) - b_j - \bigcup_{p=1}^{j-1} Q(b_p)\right] \neq \emptyset, 2 \leq j \leq k$
$b)$ $\bigcup_{p=1}^{k} Q(b_p) = E - \{\text{branches not in the } Q\text{-sequence}\}$.

The set of branches not in the cutset cover sequence be called *unmapped branches*.

Basically, the fundamental cutsets in the cutset cover sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$ contains all the edges in the graph except the unmapped branches, and every cutset $Q(b_j)$ in this sequence has at least one edge that is not in any cutset that precedes $Q(b_j)$ in this sequence.

Note that for a given spanning tree and its f-cutsets, there may be more than one Q-sequence. For example for the fundamental cutsets given in (2), following are three Q-sequences.

1) $Q(b_4), Q(b_5), Q(b_2)$
2) $Q(b_4), Q(b_5), Q(b_1), Q(b_2)$
3) $Q(b_1), Q(b_2), Q(b_4)$.

Without the loss of generality we assume that $Q(b_1), Q(b_2), \ldots, Q(b_k)$ is a Q-sequence of length k. Let us define $\hat{S}(b_j)$ as follows:

$$\begin{aligned} &\hat{S}(b_1) = Q(b_1) - b_1 \\ &\hat{S}(b_j) = Q(b_j) - b_j - \textstyle\bigcup_{p=1}^{j-1} Q(b_p), 2 \leq j \leq k. \end{aligned}$$

Basically, $\hat{S}(b_1)$ is the set of all chords in the cutset $Q(b_1)$ and $\hat{S}(b_j)$, $j \neq 1$, is the set of all chords in the

$$\begin{array}{cccccccccccccc} Q(b_1) & Q(b_2) & Q(b_3) & \cdots & Q(b_{k-1}) & Q(b_k) & \hat{S}(b_1) & \hat{S}(b_2) & \hat{S}(b_3) & \cdots & \cdots & \cdots & \hat{S}(b_{k-1}) & \hat{S}(b_k) \\ \hline U & O & O & \cdots & O & O & I & O & O & \cdots & O & \cdots & O & O \\ O & U & O & \cdots & O & O & \times & I & O & \cdots & O & \cdots & O & O \\ O & O & U & \cdots & O & O & \times & \times & I & \cdots & O & \cdots & O & O \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ O & O & O & \cdots & U & O & \times & \times & \times & \times & \times & \times & I & O \\ O & O & O & \cdots & O & U & \times & \times & \times & \times & \times & \times & O & I \end{array} \tag{2}$$

cutset $Q(b_j)$ that are not in any of the cutsets that precede $Q(b_j)$ in the cutset cover sequence. One can show that the subgraphs $b_j \bigcup \hat{S}(b_j)$ are the cutset ears of size at least two in a generalized cutset ear decomposition of G.

Deletion of an edge and *contraction* of an edge are dual operations. Here by contraction of an edge we refer to the operation of identifying the end vertices of the edge, short-circuiting the end vertices and removing self loops that result from this short-circuiting. It can be shown that deletion of a row from the Q_f matrix corresponds to contraction of the corresponding branch from the graph. So, we can see that the fundamental cutsets in a cutset cover sequence are the fundamental cutsets of the graph obtained by contracting the unmapped branches.

An ordered sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$ is a *generalized cutset cover sequence* if

$a)$ this sequence is a cutset cover sequence, and
$b)$ for every unmapped branch $b_i, Q(b_i) \bigcap \hat{S}(b_j) = \hat{S}(b_j)$, where j is the largest index such that $Q(b_i) \bigcap \hat{S}(b_j) \neq \emptyset$. In this we say that the unmapped branch b_i is *covered* by the branch b_j.We also say that chord b_j covers itself.

Given a generalized cutset cover sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$, we define the set $Q-Cover(b_i)$ for each $i = 1, 2, \ldots, k$ as the set of all branches (including itself) covered by the branch b_i. The $Q-Cover$ sets define a partition of the branches of the given spanning tree. If we arrange the rows of the f-cutset matrix to correspond to the branches in the sets $Q-Cover(b_1), Q-Cover(b_2), \ldots, Q-Cover(b_k)$ in that order and arrange the columns to correspond to the sets, then the f-cutset matrix will have the form shown in (2). In this figure, I stands for a matrix of all 1's, O is a matrix of all 0's and U refers to the unit matrix of appropriate size. Also $Q(b_i)$ stands for $Q-Cover(b_i)$.

The f-cutset matrix of a hypothetical graph arranged as in (2) will look as in (3).

For the example in (3), if we select the cutset cover sequence $Q(b_1), Q(b_4), Q(b_6), Q(b_8)$, the corresponding $Q-Cover(b_i)$'s are :

$Q-Cover(b_1) = \{b_1, b_2, b_3\}$; $Q-Cover(b_4) = \{b_4, b_5\}$; $Q-Cover(b_6) = \{b_6, b_7\}$; and $Q-Cover(b_8) = \{b_8, b_9\}$.
The $\hat{S}(b_j)$'s are:
$\hat{S}(b_1) = \{c_1, c_2, c_3\}$; $\hat{S}(b_4) = \{c_4, c_5, c_6\}$;
$\hat{S}(b_6) = \{c_7, c_8\}$; and $\hat{S}(b_8) = \{c_9, c_{10}, c_{11}\}$.

IV. GEN-SMART: A GENERALIZED ALGORITHMIC FRAMEWORK FOR THE SLTM PROBLEM

In this section we first present GEN-SMART, an algorithmic framework for the SLTM problem. This framework shown in Fig. 4 includes as special cases the other SMART-based

$$
\begin{array}{c|cccccccccccccccccccc|}
 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
 & & & & & & & & & & \multicolumn{3}{c}{\hat{S}(b_1)} & & & & & & & & \\
b_1 & 1&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0 \\
b_2 & 0&1&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0 \\
b_3 & 0&0&1&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0 \\
\hline
 & & & & & & & & & & & & & \multicolumn{3}{c}{\hat{S}(b_4)} & & & & & \\
b_4 & 0&0&0&1&0&0&0&0&0&1&0&1&1&1&1&0&0&0&0&0 \\
b_5 & 0&0&0&0&1&0&0&0&0&0&1&1&1&1&1&0&0&0&0&0 \\
\hline
 & & & & & & & & & & & & & & & & \multicolumn{2}{c}{\hat{S}(b_6)} & & & \\
b_6 & 0&0&0&0&0&1&0&0&0&1&1&0&1&1&1&1&1&0&0&0 \\
b_7 & 0&0&0&0&0&0&1&0&0&1&1&0&1&1&1&1&1&0&0&0 \\
\hline
 & & & & & & & & & & & & & & & & & & \multicolumn{3}{c}{\hat{S}(b_8)} \\
b_8 & 0&0&0&0&0&0&0&1&0&0&1&1&0&0&1&0&0&1&1&1 \\
b_9 & 0&0&0&0&0&0&0&0&1&0&1&1&0&0&1&0&1&1&1&1
\end{array} \quad (3)
$$

Fig. 4. Algorithm GEN-SMART

1: Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Let $A \subseteq \hat{S}(b_i)$ and $B \subseteq Q - Cover(b_i)$
4: Map the edges in the set $b_i \bigcup A \bigcup B$ into disjoint lightpaths in G_P.
5: **end for**

TABLE I
SPECIAL CASES OF GEN-SMART ALGORITHMS

Choice of A and B	Special case of GEN-SMART
$\lvert A\rvert = 1, \lvert B\rvert = 1$	CUTSET-SMART-SIMPLIFIED
$A = \hat{S}(b_i), \lvert B\rvert = 1$	CUTSET-SMART
$\lvert A\rvert = 1, B = Q - Cover(b_i)$	CIRCUIT-SMART
$A = \hat{S}(b_i), B = Q - Cover(b_i)$	GEN-CUTSET-SMART

algorithms discussed in [17]. We also discuss the extent to which GEN-SMART can provide survivability against multiple failures.

For the sake of simplicity in presentation we have assumed in the description of GEN-SMART that all the edges in the set $A \subseteq \hat{S}(b_i)$ and $B \subseteq Q - Cover(b_i)$ can be mapped into disjoint paths in G_P. But this may not always be possible. In such cases, we map a maximum subset of these edges into disjoint paths. To the other edges in this set we add protection edges and map each edge and its protection edge into disjoint paths in G_P. (see [19]). Also, if we choose $A = \hat{S}(b_i)$ and $B = Q - Cover(b_i)$ then GEN-SMART becomes the same as GEN-CUTSET-SMART presented in [17]. Also, different choices of A and B in GEN-SMART lead to different versions of SMART-based algorithms discussed in earlier works. These choices and the corresponding versions are given next.

For the sake of completeness, we repeat these special versions in Fig. 5 - 8. See also Table I.

Fig. 5. Algorithm CUTSET-SMART-SIMPLIFIED

1: Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Pick a chord c in $\hat{S}(b_i)$.
4: Map the edges b_i and c into disjoint lightpaths in G_P.
5: **end for**

Fig. 6. Algorithm CUTSET-SMART

1: Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Map the edges in the set $b_i \bigcup \hat{S}(b_i)$ into disjoint lightpaths in G_P.
4: **end for**

Fig. 7. Algorithm CIRCUIT-SMART

1: Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Pick a chord c in $\hat{S}(b_i)$.
4: Map the edges in the set $c \bigcup Q - Cover(b_i)$ into disjoint lightpaths in G_P.
5: **end for**

Fig. 8. Algorithm GEN-CUTSET-SMART

1: Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Map the edges in the set $\hat{S}(b_i) \bigcup Q - Cover(b_i)$ into disjoint lightpaths in G_P.
4: **end for**

Some important observations on the different versions of GEN-SMART are now in order:

- CUTSET-SMART-SIMPLIFIED, the simplest of all these algorithms, does not guarantee survivability even against a single physical link failure, unless protection edges are added to the unmapped branches [16][17].
- CUTSET-SMART doest not guarantee survivability even against a single physical link failure, unless protection edges are added to the unmapped branches [17]. But it has potential to provide some degree of survivability against multiple failures.
- CIRCUIT-SMART guarantees survivability against a single failure [15][16], but its potential to provide survivability against multiple failures is limited.
- GEN-CUTSET-SMART guarantees survivability against

a single failure, and its potential to guarantee survivability against multiple failures is very high.

Both CUTSET-SMART and GEN-CUTSET-SMART have higher potential to provide survivability against multiple failures because in both these algorithms all the edges in $\hat{S}(b_i)$ are mapped. In the next section we provide an analytical evaluation of the extent to which these algorithms provide survivability against multiple failures.

V. Robustness of Survivable Logical Topology Mapping Algorithms

In this section we first define the concept of robustness of an algorithm that is a measure of the ability of the algorithms to provide survivability against multiple physical failures.

Given a logical topology G_L and a physical topology G_P, the robustness $\beta(A, r)$ of a logical topology mapping algorithm A with respect to G_P and G_L is defined as the ratio of the number of cuts of G_L that are protected by algorithm A against r physical link failures to the total number of cuts in G_L.

For these algorithms we now proceed to evaluate $\beta(A, r)$. In the following A_1, A_2, A_3 and A_4 denote algorithms CUTSET-SMART-SIMPLIFIED, CUTSET-SMART, CIRCUIT-SMART, and GEN-CUTSET-SMART, respectively.

Given a generalized cutset cover sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$. Let us first partition all cuts in G_L into the sets $Q_1, Q_2, \ldots, Q_k$ where Q_i is the set of all cuts that contain at least one branch from the set $Q - Cover(b_i)$ and no branch from any set $Q - Cover(b_j)$, $j > i$. Note that this partition is well defined since every cut must have at least one branch.

Consider now a cut $S \in Q_i$. Assume that S contains p branches from Q_i. Now we recall the following results from [16][17].

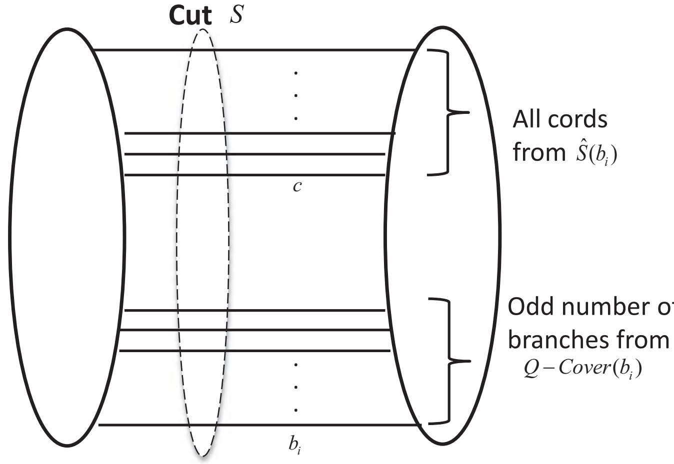
Theorem 1:
a) If a cut contains the branches $\{b_1, b_2, \ldots, b_j\}$ then the corresponding cut vector can be represented as *modulo 2* addition of the vectors $Q(b_1), Q(b_2), \ldots, Q(b_j)$. That is, the cut vector is equal to $Q(b_1) \bigoplus Q(b_2) \bigoplus \ldots \bigoplus Q(b_j)$.
b) Given a cutset cover sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$, let $Q(b_{i_1}), Q(b_{i_2}), \ldots, Q(b_{i_l})$ be a subsequence of this sequence then $\hat{S}(b_{i_l}) \subseteq Q(b_{i_1}) \bigoplus Q(b_{i_2}) \bigoplus \ldots \bigoplus Q(b_{i_l})$.

In view of Theorem 1 (b), the cut S will have the form in Fig. 9 if S has an odd number p of branches from the set $Q - Cover(b_i)$. Note that if p is even then none of the chords in $\hat{S}(b_i)$ will be in S. The numbers of edges mapped disjointly by the different Algorithms A_1, A_2, A_3, and A_4 are:

- Algorithm A_1 maps b_i and a chord c in $\hat{S}(b_i)$ disjointly, if S contains b_i.
- Algorithm A_2 maps b_i and all edges in $\hat{S}(b_i)$ disjointly, if S contains b_i.
- Algorithm A_3 maps all the p branches and a chord c in $\hat{S}(b_i)$.
- Algorithm A_4 maps all the p branches and all the chords in $\hat{S}(b_i)$.

Thus we have the following:

Fig. 9. Cut S



- Algorithm A_1 protects S against at least one physical link failure, if S contains b_i.
- Algorithm A_2 protects S against at least $|\hat{S}(b_i)|$ physical link failures, if S contains b_i.
- Algorithm A_3 protects S against at least p physical link failures.
- Algorithm A_4 protects S against at least $p + |\hat{S}(b_i)| - 1$ physical link failures.

Since $p \geq 1$, we can restate the last statement as:

- Algorithm A_4 protects S against at least $|\hat{S}(b_i)|$ physical link failures.

Let us now calculate the total number of cuts in Q_i that has an odd number of branches from the set $Q - Cover(b_i)$. Let this number be denoted as $ODD(Q_i)$.

Let $h_i = |Q - Cover(b_i)|, g_i = |\hat{S}(b_i)|$.
$h = \min h_i$ and $g = \min g_i$.
Also, let $N_i = h_1 + h_2 + \ldots + h_i$.

Robustness of Algorithm A_1

Algorithm A_1 will protect against a single physical failure all cuts from each Q_i that have an odd number of branches from the set $Q - Cover(b_i)$ and contain branch b_i. This number is equal to

= (Number of combinations of branches from the sets $Q - Cover(b_k)$, $k = 1, 2, \ldots, i-1$) $\times$ (Number of combinations of odd number of branches from the set $Q - Cover(b_i)$ that contain b_i).
$= 2^{N_{i-1}} \times 2^{h_i - 2}$
$= 2^{N_i}/4$.

Since the number of cuts in G_L is $2^{n-1} - 1$, where n is the number of nodes in G_L, and $n - 1 = h_1 + h_2 + \ldots + h_k$, we get

$$\beta(A_1, 1) \geq 1/4(\sum_{i=1}^{k} 2^{N_i})/(2^{n-1} - 1) \quad (4)$$

Note that if $p \geq 2, \beta(A, p) \geq 0$, since there is no guarantee that algorithm A_1 will protect any cut if 2 or more physical failures occur.

Robustness of Algorithm A_2:

Algorithm A_2 will protect against g_i physical failures all cuts from each Q_i that have an odd number of branches from the set $Q - Cover(b_i)$ and contain branch b_i. This follows from the fact that each such cut will have b_i and all edges in $\hat{S}(b_i)$ that are mapped disjointly.

So,

$$\beta(A_2, g) \geq 1/4(\sum_{i=1}^{k} 2^{N_i})/(2^{n-1} - 1) \quad (5)$$

Robustness of Algorithm A_3:

Algorithm A_3 will protect against at least p physical failures all cuts from each Q_i that have an odd number p of branches from the set $Q - Cover(b_i)$. This follows from the fact that each such cut will have p branches and at least one chord c in $\hat{S}(b_i)$ that are mapped disjointly.

This number is equal to

= (Number of combinations of branches from the sets $Q - Cover(b_k), k = 1, 2, \ldots, i-1) \times$ (Number of combinations of p branches from the set $Q - Cover(b_i)$)

$= 2^{N_{i-1}} C(h_i, p)$.

So

$$\beta(A_3, p) \geq \left(\sum_{i=1}^{k} 2^{N_{i-1}} \sum_{\text{odd } q \geq p}^{h_i} C(h_i, q) \right) /(2^{n-1} - 1),$$
$$\text{for odd } p \geq 1 \quad (6)$$

where $C(h_i, q)$ is the number of q-combinations of h_i elements.

If $p = 1$, then it can be verified that $\beta(A_3, 1) = 1$, confirming that CIRCUIT-SMART protects G_L against any single physical link failure [15][16].

Robustness of Algorithm A_4:

Algorithm A_4 will protect against at least $\left|\hat{S}(b_i)\right|$ physical failures all cuts from each Q_i that have an odd number of branches from the set $Q - Cover(b_i)$. This follows from the fact that each such cut will have at least one branch and all the chords in $\hat{S}(b_i)$ that are mapped disjointly.

This number is equal to

= (Number of combinations of branches from the sets $Q - Cover(b_k), k = 1, 2, \ldots, i-1) \times$ (Number of combinations of p branches from the set $Q - Cover(b_i)$).

$= 2^{N_i - 1} \times 2^{h_i - 1}$

$= 2^{N_i}/2$.

So

$$\beta(A, p) \geq 1/2(\sum_{i=1}^{k} 2^{N_i})/(2^{n-1} - 1) \quad (7)$$

Let $SUM = \left(\sum_{i=1}^{k} 2^{N_i}\right) /(2^{n-1} - 1)$.

Then we can rewrite (4), (5), (7) as

$\beta(A_1, 1) \geq 1/4\, SUM$

$\beta(A_2, g) \geq 1/4\, SUM$

$\beta(A_4, g) \geq 1/2\, SUM$

The value of *SUM* depends on the choice of generalized cutset cover sequence selected.

The lower bounds in the above are the numbers of cuts that are guaranteed to be protected by the respective algorithms. Depending on the length of the generalized cutset cover sequence, the sizes of h_i's and g_i's, the location of physical link failures and the mappings used, the number of protected cuts could be much larger. The higher the value of $\beta(A, r)$ the higher will be the probability that algorithm A will protect G_L from any set of r physical link failures.

VI. Simulation Results and Analysis

To compare the performance of CUTSET-SMART-SIMPLIFIED, CIRCUIT-SMART, CUTSET-SMART, and GEN-CUTSET-SMART with respect to their ability to provide multiple failure survivability simulation studies were conducted using LEMON (Library for Efficient Modeling and Optimization in Networks) [20] and G++ under Linux system. The physical and logical topologies were regular topologies with connectivity equal to 3, 4, and 5 constructed using a procedure originally given by Harary and described in [21]. The number of nodes in the physical topologies was set to 50, 60, 70, 80, 90, and 100 nodes. The nodes in logical topologies were a subset of the physical nodes and the number of nodes in a logical topology was set to 50% of the nodes in the corresponding physical topology.

For each combination of (topology connectivity, number of nodes in physical topology, number of physical link failures), 100 physical and corresponding logical topology pairs were generated and tested against 4 algorithms described in the previous section. Given k-connected physical and logical topologies, the survivability of the G_L under multiple (2 to $k-1$) physical link failures is determined by the number of G_L's which remain connected against physical link failures. Our simulation enumerated all possible combinations of physical link failures and evaluated how many G_L's could remain connected. The success rate in each case is calculated.

First a spanning tree on a logical topology was generated and the fundamental circuits and cutsets with respect to the spanning tree were found. The generalized cutset cover sequence was generated using the algorithms in [17]. With the information of the fundamental cutsets, the $Q - Cover(b_i)$ and $\hat{S}(b_i)$ sets were generated as shown in (2) and (3). Then we applied the four algorithms (CUTSET-SMART-SIMPLIFIED, CIRCUIT-SMART, CUTSET-SMART, and GEN-CUTSET-SMART) and mapped maximal number of edges disjointly in $b_i \bigcup A \bigcup B$. If the disjoint mappings for some of the edges in $b_i \bigcup A \bigcup B$ do not exist, a parallel edge is added to the logical topology and the newly added edge is mapped disjointly with the original edge. At the end of the procedure, the unmapped logical edges were randomly mapped, which could increase the chance of survivability for the logical mapping.

The simulation results giving the success rate are shown in Table II, III, and IV. Notice that in Table II, extra tests for the single failure case in 3-connected physical and logical topologies are presented, which show that CUTSET-SMART and GEN-CUTSET-SMART can guarantee 100% survivability for the logical topology under a single physical link failure,

TABLE II

SUCCESS RATE FOR 3-CONNECTED PHYSICAL AND LOGICAL TOPOLOGIES

3-conn	50 nodes		60 nodes		70 nodes	
failures \ Algorithms	1	2	1	2	1	2
A_1	92.173	71.857	89.711	65.294	89.429	64.338
A_2	92.987	73.701	90.533	67.080	90.371	66.024
A_3	100	85.367	100	83.775	100	82.263
A_4	100	86.426	100	84.406	100	83.375
3-conn	80 nodes		90 nodes		100 nodes	
failures \ Algorithms	1	2	1	2	1	2
A_1	87.617	57.744	86.570	55.356	84.427	52.313
A_2	88.700	59.710	87.963	57.2	85.853	54.405
A_3	100	78.811	100	78.377	100	76.149
A_4	100	79.913	100	79.367	100	77.073

TABLE III

SUCCESS RATE FOR 4-CONNECTED PHYSICAL AND LOGICAL TOPOLOGIES

4-conn	50 nodes		60 nodes		70 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	94.709	85.841	93.907	84.533	93.655	81.356
A_2	95.975	88.679	95.272	86.979	94.841	84.513
A_3	96.646	88.262	95.950	87.549	95.383	85.219
A_4	97.367	90.263	96.665	89.159	96.235	86.984
4-conn	80 nodes		90 nodes		100 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	92.381	80.498	91.575	78.445	91.000	76.815
A_2	94.018	83.343	93.373	81.564	93.043	79.780
A_3	94.801	83.473	93.983	81.802	93.466	79.700
A_4	95.639	85.396	95.018	83.819	94.41	81.582

TABLE IV

SUCCESS RATE FOR 5-CONNECTED PHYSICAL AND LOGICAL TOPOLOGIES

5-conn	50 nodes		60 nodes		70 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	99.764	99.450	99.785	99.366	99.809	99.246
A_2	99.912	99.653	99.880	99.634	99.888	99.583
A_3	99.877	99.617	99.869	99.541	99.867	99.473
A_4	99.956	99.810	99.935	99.771	99.937	99.746
5-conn	80 nodes		90 nodes		100 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	99.772	99.231	99.668	99.184	99.674	99.089
A_2	99.858	99.557	99.785	99.510	99.787	99.507
A_3	99.848	99.827	99.827	99.437	99.804	99.363
A_4	99.916	99.915	99.915	99.725	99.899	99.654

while CUTSET-SMART-SIMPLIFIED and CIRCUIT-SMART can not.

Based on the simulations, we summarize our observations as follows.

- The value of *SUM* is at most 2. This can be reached when each $h_i = 1$. In such cases, (4) and (5) simplify to $\beta(A_1, 1) \geq 1/2, \beta(A_2, g) \geq 1/2$. In spite of this low value on the corresponding robustness, algorithms A_1 and A_2 have higher ability to provide survivability against multiple physical link failures.
- As expected, A_2 has higher potential to provide survivability against multiple failures compared to A_1.
- As expected, algorithms A_3 and A_4 have higher success rate compared to A_1 and A_2.
- The success rate of all algorithms is higher for higher values of connectivity of physical topologies. This could be due to the survivability of a large number of disjoint paths. This calls for future research.

VII. CONCLUSION

The survivable logical topology mapping (SLTM) problem in an IP-over-WDM optical network is to map each link (u, v) in the logical topology G_L (at the IP layer) into a lightpath between the nodes u and v in the physical topology G_P (at the optical layer) such that failure of a physical link does not cause the logical topology to become disconnected. It is assumed that both the physical and logical topologies are 2-edge connected. Most research in this area has focused on logical topology survivability against a single physical link failure. Also, existing approaches do not provide insight into the problem when multiple physical link failures, such as SRLG failures, occur. In this paper we pursued the structural approach developed in [15][16][17] to study the logical topology mapping problem for the case of multiple failures. We first presented a generalized algorithmic framework for the SLTM problem. This framework includes several other frameworks considered in our earlier works [16][17] as special cases. We then defined the concept of robustness of a mapping algorithm which captures the ability of the algorithm to provide survivability against multiple physical link failures. This is similar to the concept of fault coverage used in hardware/software testing. The higher the value of the robustness of an algorithm the higher the probability that the algorithm will be able to provide survivability. We analyzed the different frameworks for their robustness property. Specifically, we provided lower bounds for the robustness for the different algorithms. These lower bounds give the number of cuts which an algorithm is guaranteed to protect against multiple failures. The quantity *SUM* used in these formulas depends on several structural features such as the choice of the generalized cutset cover sequence to be used to provide higher degree of robustness. Using simulations, we demonstrate that even when an algorithm cannot be guaranteed to provide survivability against all multiple failures, its robustness could be very high. The work also provides a basis for the design of survivable mapping algorithms when special classes of failures such as SRLG failures are to be protected against. Further work along these lines is in progress.

ACKNOWLEDGMENT

The work in this paper has been supported by NSF grant CNS-1115130 awarded to Krishnaiyan Thulasiraman.

REFERENCES

[1] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee, "Fault management in IP-over-WDM networks: WDM protection versus IP restoration," *IEEE Journal on Selected Areas in Communications*, pp. 21–33, 2002.

[2] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: an approach to high bandwidth optical WAN's," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171–1182, 1992.
[3] D. West, *Introduction to Graph Theory*. Prentice Hall, 1996.
[4] J. Kleinberg, "Approximation algorithms for disjoint paths problems," Ph.D. dissertation, MIT, 1996.
[5] Y. Shiloach and Y. Perl, "Finding two disjoint paths between two pairs of vertices in a graph," *Journal of the ACM*, vol. 25, no. 1, pp. 1–9, Jan. 1978.
[6] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," *INFOCOM*, 2001.
[7] ——, "Survivable lightpath routing: A new approach to the design of WDM-based networks," *IEEE Journal of Selected Areas in Communication*, vol. 20, no. 4, pp. 800–809, 2002.
[8] A. Todimala and B. Ramamurthy, "A scalable approach for survivable virtual topology routing in optical WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 63–69, August 2007.
[9] F. Ducatelle and L. Gambardella, "FastSurv: a new efficient local search algorithm for survivable routing in WDM networks," *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, pp. 1925–1929, 2004.
[10] O. Crochat, J.-Y. L. Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, no. 3, pp. 384–395, 2000.
[11] R. Shenai and K. Sivalingam, "Hybrid survivability approaches for optical WDM mesh networks," *Journal of Lightwave Technology*, vol. 23, no. 10, pp. 3046–3055, Oct. 2005.
[12] K. Lee and E. Modiano, "Cross-layer survivability in WDM networks with multiple failures," in *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*, feb. 2008.
[13] D. D.-J. Kan, A. Narula-Tam, and E. Modiano, "Lightpath routing and capacity assignment for survivable IP-over-WDM networks," in *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on*, October 2009, pp. 37–44.
[14] 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," *Design of Reliable Communication Networks*, October 2011.
[15] M. Kurant and P. Thiran, "Survivable mapping algorithm by ring trimming (SMART) for large IP-over-WDM networks," *BroadNets*, 2004.
[16] 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 *INFOCOM*, 2009, pp. 1026–1034.
[17] K. Thulasiraman, M. Javed, and G. Xue, "Primal meets dual: A generalized theory of logical topology survivability in IP-over-WDM optical networks," in *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, 2010.
[18] C. Liu and L. Ruan, "Logical topology augmentation for survivable mapping in IP-over-WDM networks," *GLOBECOM*, 2005.
[19] K. Thulasiraman, T. Lin, M. Javed, and G. Xue, "Logical topology augmentation for guaranteed survivability under multiple failures in IP-over-WDM optical networks," *Optical Switching and Networking*, pp. 206–214, 2010.
[20] "Library for efficient modeling and optimization in networks (LEMON)." [Online]. Available: http://lemon.cs.elte.hu/trac/lemon
[21] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*. Wiley-Interscience, 1992.