A New Survivable Mapping Problem in IP-over-WDM Networks

Chang Liu and Lu Ruan
Department of Computer Science
Iowa State University, Ames, IA 50011
email: {liuc, ruan}@cs.iastate.edu

Abstract—We introduce a new version of the widely studied survivable mapping problem in IP-over-WDM networks. Unlike the original survivable mapping problem, the new survivable mapping problem (NSM) allows the given logical topology to be augmented by adding new logical links to it. NSM is stated as follows: given a physical topology and a logical topology, compute a survivable logical topology that contains the given logical topology such that the minimal survivable mapping cost for the resulting logical topology is minimized. The problem is significant for two reasons. First, if the given logical topology does not have a survivable mapping, we can add logical links to it to enable a survivable mapping. Second, if the given logical topology has a survivable mapping, it is possible to reduce the minimal survivable mapping cost by adding logical links to the given logical topology.

We first prove that a solution to NSM always exists and then provide a straightforward Integer Linear Program (ILP) formulation for NSM. Moreover, we present a theoretical result that leads to an improved ILP formulation for NSM and an NP-hardness proof for NSM. The significance of both the new survivable mapping problem and the theoretical result is demonstrated by simulation.

Index Terms—Network survivability, survivable mapping, IP-over-WDM networks

I. INTRODUCTION

There is a growing consensus that the next generation Internet will employ an IP-over-WDM architecture [1]. In this architecture, IP routers are attached to the optical cross-connects (OXCs) in a WDM optical network and IP links are realized by lightpaths in the optical network. We refer to IP network topology, IP routers, and IP links as logical topology, logical nodes, and logical links, respectively. We refer to optical network topology, OXCs, and optical fibers as physical topology, physical nodes, and physical links, respectively. A logical topology is mapped on a physical topology by mapping each logical link to a path in the physical topology. (This reflects the fact that each IP link is realized by a lightpath in the optical network.) There are different ways to map a logical topology on a physical topology. For example, consider the logical and physical topologies shown in Fig. 1. One mapping could map the logical link \((a,c)\) to the physical path \(<A - B - C>\), while another mapping could map \((a,c)\) to the physical path \(<A - E - C>\).

Survivability is an important issue in IP-over-WDM networks since a network failure such as a fiber cut can cause tremendous data loss. Many lightpath protection and restoration schemes have been proposed to achieve survivability in the optical layer [2][3][4][5][6][7]. And many MPLS-based protection and restoration schemes have been proposed to achieve survivability in the IP layer [8][9][10][11][12]. IP layer failure recovery is possible only if a network failure does not disconnect the IP topology. However, a fiber cut in an IP-over-WDM network may leave the IP topology disconnected because all lightpaths using the failed fiber will be disrupted. To maintain IP network connectivity upon a fiber cut, the following survivable mapping problem has been studied in literature: Given a logical topology and a physical topology, map the logical topology on the physical topology such that the logical topology remains connected in case of any single physical link failure. Generally, there are different ways to map a logical topology on a physical topology and not all of them are survivable. For example, consider the logical and physical topologies given in Fig. 1. One possible mapping is to map \((a,b)\) to \(<A - B>\), \((a,c)\) to \(<A - B - C>\), \((b,d)\) to \(<B - D>\), \((b,e)\) to \(<B - A - E>\), \((c,e)\) to \(<C - E>\), and \((d,e)\) to \(<D - E>\). This is not a survivable mapping since the failure of physical link \((A,B)\) will cause logical links \((a,b)\), \((a,c)\), and \((b,e)\) to fail simultaneously,
leaving the logical topology disconnected. On the other hand, a survivable mapping can be obtained by mapping \((a, b)\) to 

\(<A - E - D - B>\) instead.

The survivable mapping problem has been studied in [13][14][15][16][17][18][19][20]. In [13], it is proven that determining whether a survivable mapping exists for a logical topology on a physical topology is NP-Complete. [14] gives a necessary and sufficient condition for a mapping to be survivable. Based on the condition, an integer linear program (ILP) formulation is given to solve the survivable mapping problem with the objective of minimizing the cost of the mapping. In [15], a mixed integer linear program (MILP) formulation for the survivable mapping problem is given. The MILP has faster computation speed than the ILP given in [14] since the number of constraints in the MILP grows as a polynomial with the size of the network while the number of constraints in the ILP grows exponentially with the size of the network. Various heuristic algorithms for the survivable mapping problem are proposed in [16][17][18][19]. In [20], a formal analysis of a previously proposed heuristic algorithm named SMART [19] is given and the analysis shows that SMART can be used to derive practical methods for determining the existence or absence of a survivable mapping for large networks.

While the original survivable mapping problem does not allow the given logical topology to be changed when finding a survivable mapping for it, we note that it is sometimes beneficial to add logical links to the given logical topology for two reasons. First, if the given logical topology does not have a survivable mapping on the given physical topology, adding some logical links to the given logical topology will enable a survivable mapping to be obtained. Second, even if a survivable mapping for the given logical topology exists, adding some logical links to the given logical topology may reduce the minimal survivable mapping cost. As in [14], we define the cost of a mapping as the total number of wavelength channels used to map all the logical links in the logical topology. Since a logical link (i.e., a lightpath) uses one wavelength channel on each link along its physical path, the cost of a logical link equals the number of hops in its physical path and the cost of a mapping equals the total cost of all logical links in the logical topology. To see the benefit of adding logical links to a logical topology in reducing the minimal survivable mapping cost, consider the logical and physical topologies shown in Fig. 1. The left table in Fig. 2 shows the minimal cost survivable mapping for the logical and physical topologies, which has a cost of 10. If we add link \((a, e)\) to the logical topology, a survivable mapping with a cost of 9 can be obtained, as shown in the right table in Fig. 2. Due to the benefits of adding logical links in a logical topology, we propose a new version of the survivable mapping problem and study the problem in this paper. The new survivable mapping problem is the following: Given a logical topology \(G_l\) and a physical topology \(G_p\), compute a logical topology \(G'_l\) by adding 0 or more logical links to \(G_l\) such that \(G'_l\) has a survivable mapping on \(G_p\) and the cost of the survivable mapping is minimized.

The idea of adding logical links to a logical topology to enable a survivable mapping has been explored in [20]. An algorithm is given in [20] to identify a good logical link to add to the given logical topology and simulation results show that adding one logical link can enable a survivable mapping in most cases. A drawback of the algorithm is that it does not guarantee to enable a survivable mapping since only one logical link is added to the logical topology. In [21], we propose a method to add logical links to a given logical topology so that the resulting logical topology has a survivable mapping on the given physical topology and any shortest path mapping of the resulting logical topology on the physical topology is survivable. (A shortest path mapping maps every logical link to the shortest physical path, which is polynomial-time computable.) Clearly, the method can be used to enable a survivable mapping for a logical topology if it does not have one. However, the cost of the survivable mapping for the resulting logical topology may not be minimized because the resulting logical topology is chosen such that a survivable mapping can be computed in polynomial time. To our best knowledge, this work is the first to identify the benefit of augmenting a logical topology in reducing its survivable mapping cost and to study the problem of augmenting a logical topology to minimize its survivable mapping cost.

The rest of the paper is organized as follows. In section II, we define the necessary terms and notations and give the formal definition of the new survivable mapping problem. In section III, we prove that a solution to the new survivable mapping problem always exists and give a straightforward ILP formulation to solve the problem. In section IV, we first present a theoretical result about the new survivable mapping problem. Based on the theoretical result, we then provide an improved ILP formulation for the problem and give an NP-hardness proof of the problem. Simulation results are discussed in section V. Finally, a conclusion is given in section VI.

II. TERMINOLOGY AND PROBLEM DEFINITION

A. Terminology

Logical and physical topologies are represented by undirected graphs \(G_l = (V_l, E_l)\) and \(G_p = (V_p, E_p)\), respectively. Generally, \(V_l \subseteq V_p\); however, we make the simplifying assumption that \(V_l = V_p = V\) in this paper. Although logical and physical topologies are represented by undirected graphs, sometimes it is useful to treat an undirected edge \(ij \in E_p\) as two directed edges \((ij, ji)\) in opposite directions. We

<table>
<thead>
<tr>
<th>Logical link</th>
<th>Physical path</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a, b)</td>
<td>A-E-D-B</td>
</tr>
<tr>
<td>(a, c)</td>
<td>A-B-C</td>
</tr>
<tr>
<td>(b, d)</td>
<td>B-D</td>
</tr>
<tr>
<td>(b, e)</td>
<td>B-A-E</td>
</tr>
<tr>
<td>(c, e)</td>
<td>C-E</td>
</tr>
<tr>
<td>(d, e)</td>
<td>D-E</td>
</tr>
</tbody>
</table>

Cost of mapping 10

<table>
<thead>
<tr>
<th>Logical link</th>
<th>Physical path</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a, b)</td>
<td>A-B</td>
</tr>
<tr>
<td>(a, c)</td>
<td>A-B-C</td>
</tr>
<tr>
<td>(a, e)</td>
<td>A-E</td>
</tr>
<tr>
<td>(b, d)</td>
<td>B-D</td>
</tr>
<tr>
<td>(b, e)</td>
<td>B-A-E</td>
</tr>
<tr>
<td>(c, e)</td>
<td>C-E</td>
</tr>
<tr>
<td>(d, e)</td>
<td>D-E</td>
</tr>
</tbody>
</table>

Cost of mapping 9

Fig. 2. Minimal cost survivable mapping for the logical topology in Fig. 1(a) before and after adding the logical link \((a, e)\). Note that the logical link \((a, b)\) is mapped differently in the original logical topology and the new logical topology.
denote the set of directed edges obtained by replacing each undirected edge in \( E_p \) by two directed edges of opposite directions as \( E^d_p \), where ‘d’ stands for ‘directed’.

For a graph \( G = (V, E) \) and \( S \subseteq V \) (\( S \neq \emptyset \)), the edge cut of \( G \) defined by \( S \), denoted by \( EC_G(S) \), is the set of edges in \( G \) with one end node in \( S \) and the other end node in \( V - S \). Clearly, removing the edges in \( EC_G(S) \) from \( G \) will disconnect \( G \).

For \( s, t \in V \) (\( s \neq t \)), a path from \( s \) to \( t \) in \( G_p \) is denoted as \( P_{st} = (s - \cdots - t) \). \( P_{st} \) denotes the set of all paths from \( s \) to \( t \) in \( G_p \). A mapping from \( G_t \) to \( G_p \) is a function \( M : E_t \rightarrow \bigcup_{s,t \in V} P_{st} \). That is, \( M \) maps each edge \( st \in E_t \) to a path from \( s \) to \( t \) in \( G_p \). The cost of mapping \( M \), denoted by \( cost(M) \), is the total wavelength channels used to map all the logical links in \( G_t \). It can be computed as \( cost(M) = \sum_{st \in E_t} |M(st)| \), where \( |M(st)| \) is the hop count of \( M(st) \).

A path \( st \in E_t \) is a reflective logical link if there is a physical link between \( s \) and \( t \) in \( G_p \). In other words, \( st \) is a reflective logical link if \( st \in E_t \cap E_p \). A reflective logical link \( st \) is reflectively-routed under mapping \( M \) if \( M(st) = (s - i) \), i.e., \( M \) maps \( st \) to the single-hop path between \( s \) and \( t \) in \( G_p \). \( M \) is a reflectively-routed mapping if all reflective logical links are reflectively-routed.

The load set of a physical link \( ij \in E_p \) under mapping \( M \), denoted as \( l_M(ij) \), is the set of all logical links whose physical path traverses \( ij \), i.e., \( l_M(ij) = \{ st \in E_t \mid M(st) \text{ traverses } ij \} \). The remaining logical topology upon the failure of \( ij \in E_p \) under \( M \), denoted as \( G^M_i(ij) \), is the new logical topology obtained by removing all logical links in \( G_t \) whose physical path traverses \( ij \). Thus, \( G^M_i(ij) = (V, E_t - l_M(ij)) \). A physical link \( ij \in E_p \) is called a critical link under \( M \) if \( G^M_i(ij) \) is not connected.

\( M \) is a survivable mapping from \( G_t \) to \( G_p \) if \( G^M_i(ij) \) is connected for all \( ij \in E_p \). In other words, \( M \) is a survivable mapping if there is no critical link in \( G_p \) under \( M \).

\( G_t \) is a survivable logical topology on \( G_p \) if there exists a survivable mapping from \( G_t \) to \( G_p \). The cost of a survivable logical topology \( G_t \), denoted by \( cost(G_t) \), is the minimal cost of all survivable mappings from \( G_t \) to \( G_p \). A survivable mapping \( M \) from \( G_t \) to \( G_p \) is called a minimal cost survivable mapping if \( cost(M) = cost(G_t) \).

We say \( G_1 = (V_1, E_1) \) contains \( G_2 = (V_2, E_2) \) if \( V_1 \supseteq V_2 \) and \( E_2 \subseteq E_1 \). Given a logical topology \( G_t \) and a physical topology \( G_p \), a minimal cost survivable logical topology that contains \( G_t \) on \( G_p \) is a survivable logical topology \( G^*_t \) on \( G_p \) such that \( G^*_t \) contains \( G_t \) and \( cost(G^*_t) \) is minimized. We denote the minimized cost as \( \text{MIN-COST}_{G_t} \), then \( cost(G^*_t) = \text{MIN-COST}_{G_t} = \min \text{cost}(G_t) \), where the min is taken over all \( G_t \) such that \( G_t \) contains \( G_t \) and \( G_t \) is a survivable logical topology on \( G_p \).

### B. Problem Definition

We now give the formal definition of the new survivable mapping problem (NSM).

**NSM:** Given a logical topology \( G_t = (V, E_t) \) and a 2-edge-connected physical topology \( G_p = (V, E_p) \), compute a minimal cost survivable logical topology \( G^*_t \) that contains \( G_t \) on \( G_p \) and a mapping \( M \) from \( G^*_t \) to \( G_p \) such that \( cost(M) = \text{MIN-COST}_{G_t} \).

Note that physical topologies are required to be 2-edge-connected in practice so that any single physical link failure does not disconnect the physical topology. (A graph is 2-edge-connected if the minimal number of edges whose removal disconnects the graph is 2.)

### III. A STRAIGHTFORWARD ILP FORMULATION

First, we prove that a solution to NSM always exists.

**Theorem 1:** Given a logical topology \( G_t = (V, E_t) \) and a 2-edge-connected physical topology \( G_p = (V, E_p) \), there exists a survivable logical topology that contains \( G_t \) on \( G_p \).

**Proof:** Let \( G^*_t = (V, E_t \cup E_p) \). Clearly, \( G^*_t \) contains \( G_t \). We next prove that \( G^*_t \) is a survivable logical topology on \( G_p \) by showing that there exists a survivable mapping from \( G^*_t \) to \( G_p \).

Let \( M \) be a reflectively-routed mapping from \( G^*_t \) to \( G_p \). For any \( ij \in E_p \), \( G^M(ij) \) contains all links in \( E_p - \{ij\} \). Since \( G_p \) is 2-edge-connected, \( G^M(ij) \) must be connected. Thus, \( M \) is a survivable mapping from \( G^*_t \) to \( G_p \).

We now present a straightforward ILP formulation (referred to as ILP1) that solves NSM. Let \( K_n \) denote the undirected complete graph on the vertex set \( V \), where \( n = |V| \). ILP1 considers all edges in \( E(K_n) \) as candidate edges to be included in the resulting logical topology, where \( E(K_n) \) is the edge set of \( K_n \).

Variables to be solved:

- \( f_{ij}^d \): takes value 1 if logical link \( st \) is mapped to a path that contains physical link \( ij \), 0 otherwise.
- \( x_{ij}^d \): takes value 1 if \( st \) is included in the resulting logical topology, 0 otherwise.

Objective function:

\[
\text{Minimize } \sum_{i,j \in E_p^d} x_{ij}^d
\]

Subject to:

(a) Flow conservation constraints:

\[
\sum_{j \text{ s.t. } i,j \in E_p^d} f_{ij}^d - \sum_{j \text{ s.t. } j,i \in E_p^d} f_{ji}^d = \begin{cases} 
  x_{ij}^d & \text{if } s = i \\
  -x_{ij}^d & \text{if } t = i \\
  0 & \text{otherwise}
\end{cases},
\]

\forall i \in V, \forall s \in E(K_n).

(b) Survivability constraints:

\[
\sum_{i \in S} f_{ij}^d + f_{ji}^d < \sum_{i \in S} x_{ij}^d,
\]

\forall i,j \in E_p, \forall S \subseteq V.

(c) Logical links in the given logical topology must be kept:

\[
x_{ij}^d = 1, \forall s \in E_t.
\]

(d) Integer constraints:

\[
f_{ij}^d \in \{0, 1\}, \forall i,j \in E_p^d, \forall s \in E(K_n).
\]
The flow conservation constraints in (a) ensure that a logical link is mapped to a physical path only if it is included in the resulting logical topology, i.e., $x_{st} = 1$. In the survivability constraints in (b), the right hand side is the number of edges in the edge cut $EC_{G'_L}(S)$ and the left hand side is the number of edges in $EC_{G'_L}(S)$ that are mapped to a physical path containing $ij \in E_p$ in either direction, which equals $|L_M(ij)_\cap EC_{G'_L}(S)|$, where $G'_L$ is the resulting logical topology and $M$ is the resulting mapping from $G'_L$ to $G_p$. It is proved in [14] that $M$ is survivable if and only if $|L_M(ij)_\cap EC_{G'_L}(S)| < |EC_{G'_L}(S)|$ for all $ij \in E_p$ and all $S \subseteq V$. Therefore, the constraints in (b) ensure that the resulting mapping is survivable. Constraints in (c) ensure that the logical links in the given logical topology must stay in the resulting logical topology.

IV. A Theorem and Its Applications

ILP1 provides a straightforward method for solving NSM, which considers all links not in $G_L$ as candidate links to be added to $G_L$. In this section, we present a theorem which shows that we can obtain a solution to NSM by adding only reflective logical links to $G_L$, and the resulting logical topology has a reflectively-routed survivable mapping that achieves the minimal cost. We also give two applications of the theorem: an improved ILP for NSM and an NP-hardness proof for NSM.

A. A Theorem

**Theorem 2**: Given a logical topology $G_L = (V, E_L)$ and a 2-edge-connected physical topology $G_P = (V, E_P)$, there exists an edge set $E'' \subseteq E_P - E_L$ such that $G''_L = (V, E_L \cup E'')$ is a minimal cost survivable logical topology that contains $G_L$ on $G_P$. Moreover, there is a reflectively-routed survivable mapping $M''$ from $G''_L$ to $G_P$ such that $cost(M'') = MIN\text{-}COST_{G''_L}$.

**Proof**: See Appendix.

Theorem shows that given a logical topology $G_L$ and a 2-edge-connected physical topology $G_P$, it is always possible to obtain a minimal cost survivable logical topology that contains $G_L$ on $G_P$ by adding only reflective logical links to $G_L$. Furthermore, the resulting logical topology has a reflectively-routed survivable mapping that achieves the minimal cost.

B. An Improved ILP Formulation

Theorem 2 can be used to improve ILP1 in two ways. First, the candidate logical links to be included in the resulting logical topology can be confined to links in $E_L \cup E_P$ instead of links in $E(K_n)$. This helps reduce the number of variables in ILP1. Second, the existence of the minimal cost reflectively-routed survivable mapping for the resulting logical topology makes the mapping job easier since the physical paths for those reflective logical links can be determined right away (they are reflectively-routed). The improved ILP, referred to as ILP2, is given as follows.

\[
\begin{align*}
&\text{Minimize} & & \sum_{ij \in E_P} \sum_{st \in E_L \cup E_P} f_{ij}^{st} \\
&\text{Subject to:} & & f_{ij}^{st} - \sum_{st \in E_L \cup E_P} f_{ji}^{st} = \begin{cases} 
1 & \text{if } s = i \\
-1 & \text{if } t = i \\
0 & \text{otherwise} 
\end{cases} \\
& & & \forall i \in V, \forall st \in E_L - E_P.
\end{align*}
\]

(b). Survivability constraints: Same as those in ILP1.

(c). Logical links in the given logical topology must be kept: Same as those in ILP1.

(d). Integer constraints:

\[
\begin{align*}
&f_{ij}^{st} \in \{0, 1\}, \forall ij \in E_P, \forall st \in E_L \cup E_P. \\
&x_{st} \in \{0, 1\}, \forall st \in E_L \cup E_P.
\end{align*}
\]

The flow conservation constraints in (a) are only used for logical links in $E_L - E_P$ because other logical links are reflective and will be reflectively-routed. The constraints in (a’) ensure that reflective logical links are reflectively-routed. Note that the existence of a resulting survivable logical topology and the corresponding reflectively-routed survivable mapping is guaranteed by Theorem 2.

Compared with ILP1, ILP2 has fewer variables and fewer flow conservation constraints. As a result, ILP2 runs faster than ILP1, as will be shown in section V.

C. NP-hardness of NSM

With the help of Theorem 2, we can prove that NSM is NP-hard by showing that M2ECSS is polynomial-time reducible to NSM, where M2ECSS stands for the Minimum 2-Edge-Connected Spanning Subgraph problem that has been proven to be NP-hard [22].

For the purpose of the proof, we define the decision problem of M2ECSS and NSM as follows.

**M2ECSS**: Given a graph $G$ and a positive integer $k$, determine whether $G$ has a 2-edge-connected spanning subgraph containing at most $k$ edges.

**NSM**: Given a logical topology $G_L$, a physical topology $G_P$, and a positive integer $c$, determine whether there is a survivable logical topology $G'_L$ that contains $G_L$ on $G_P$ such that the cost of $G'_L$ is at most $c$.

**Theorem 3**: NSM is NP-hard.

**Proof**: We show that M2ECSS is polynomial-time reducible to NSM.

Given an instance $\langle G, k \rangle$ of M2ECSS, we construct an instance $\langle G_L, G_P, c \rangle$ of NSM as follows:
Let $G_I$ be a graph with the same vertex set as $G$ and no edges, which is denoted as $G_0$.

Let $G_p = G$.

Let $e = k$.

Clearly, the construction is polynomial-time computable.

Next, we show that $G$ has a 2-edge-connected spanning subgraph containing at most $k$ edges if and only if there is a survivable logical topology $G'_I$ that contains $G_0$ on $G$ such that the cost of $G'_I$ is at most $k$.

Suppose $G_{sub}$ is a 2-edge-connected spanning subgraph of $G$ and $|E(G_{sub})| \leq k$. Consider $G_{sub}$ as a logical topology and $G$ as a physical topology, then all links in $G_{sub}$ are reflective. Let $M$ be the reflectively-routed mapping from $G_{sub}$ to $G$, then $cost(M) = |E(G_{sub})| \leq k$. Under $M$, any single link failure in $G$ will affect at most one link in $G_{sub}$. Since $G_{sub}$ is 2-edge-connected, the failure will not disconnect $G_{sub}$. Therefore, $M$ is a survivable mapping. Hence, $G_{sub}$ is a survivable logical topology that contains $G_0$ on $G$ such that $cost(G_{sub}) = cost(M) \leq k$.

Suppose there is a survivable logical topology that contains $G_0$ on $G$ such that its cost is at most $k$, then we have MIN-COST,$G_{sub} \leq k$. By Theorem 2, we can obtain a minimal cost survivable logical topology that contains $G_0$ on $G$ (denoted as $G_{min}$) by adding only reflective logical links to $G_0$. Thus, $G_{min}$ is a spanning subgraph of $G$. Let $M$ be the reflectively-routed mapping from $G_{min}$ to $G$. By Theorem 2, $M$ is a survivable mapping that achieves the minimal cost, i.e., $cost(M) = |E(G_{min})| = MIN-COST,G_{sub} \leq k$. Therefore, $G_{min}$ is a spanning subgraph of $G$ with at most $k$ edges. Also, $G_{min}$ must be 2-edge-connected because the reflectively-routed mapping $M$ from $G_{min}$ to $G$ is survivable. So, $G_{min}$ is a 2-edge-connected spanning subgraph of $G$ with at most $k$ edges.

V. NUMERIC RESULTS

A. Simulation Settings

We use two physical topologies (shown in Fig. 3) in the simulations. The first one is the 14-node 21-link NSFNET and the second one is a 12-node 18-link random graph (referred to as RANDOM). Both physical topologies are 2-edge-connected. For each physical topology, two groups of 2-edge-connected logical topologies, referred to as GROUP1 and GROUP2, are used. For NSFNET, GROUP1 consists of 100 14-node 17-link random topologies and GROUP2 consists of 100 14-node 21-link random topologies. For RANDOM, GROUP1 consists of 100 12-node 15-link random topologies and GROUP2 consists of 100 12-node 18-link random topologies. All simulations are run on a Sun Ultra 10 workstation with a 440MHz CPU, 256MB RAM, and 4GB virtual memory. CPLEX8.1 is used as the ILP solver.

B. Significance of the New Survivable Mapping Problem

As discussed in section I, the new survivable mapping problem provides two benefits by allowing logical link addition to the given logical topology. First, a survivable mapping can be obtained for a non-survivable logical topology. Second, the minimal survivable mapping cost may be reduced for a survivable logical topology. To see these benefits, we run ILP2 and the ILP provided in [14] (denoted as ILP,ORIG) on the physical and logical topologies described in the previous subsection. Note that ILP2 solves the new survivable mapping problem that allows adding new logical links to the given logical topology while ILP,ORIG solves the original survivable mapping problem that does not allow the given logical topology to be changed. Thus, given a pair of logical and physical topologies, ILP2 can always find a minimal cost survivable logical topology that contains the given logical topology and the corresponding minimal cost survivable mapping; on the other hand, ILP,ORIG can obtain a minimal cost survivable mapping for the pair only if the given logical topology is survivable. Although both ILP1 and ILP2 solve the new survivable mapping problem, we use ILP2 in the simulations since it runs faster than ILP1.

Table I IMPROVEMENT OF ILP2 OVER ILP,ORIG

<table>
<thead>
<tr>
<th>Physical topology</th>
<th>NSFNET</th>
<th>RANDOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical topology</td>
<td>GROUP1</td>
<td>GROUP2</td>
</tr>
<tr>
<td># non-survivable logical topologies fixed by ILP2</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td># survivable logical topologies improved by ILP2</td>
<td>50</td>
<td>42</td>
</tr>
<tr>
<td>Maximum(Average) cost saving ratio* among improved survivable logical topologies</td>
<td>20.8% (7.2%)</td>
<td>12.0% (3.5%)</td>
</tr>
</tbody>
</table>

* cost saving ratio is defined as cost computed by ILP,ORIG-cost computed by ILP2 cost computed by ILP,ORIG

Table I shows the improvement made by ILP2 over ILP,ORIG. For GROUP1 over NSFNET, 43 out of 100 logical topologies are not survivable (i.e., ILP,ORIG can’t obtain a survivable mapping). However, ILP2 can transform these non-survivable logical topologies into survivable ones by adding new logical links. Among the 57 survivable logical topologies, ILP2 obtains lower cost than ILP,ORIG for 50 (about 88%) of them. That is, 50 survivable logical topologies can achieve lower survivable mapping cost by adding new logical links. Moreover, among the 50 improved logical topologies, the maximum/average cost saving ratio is 20.8%/7.2%. For GROUP2 over NSFNET, 1 logical topology is not survivable, for which ILP,ORIG can’t find a solution while ILP2 can.
Among the 99 survivable logical topologies, 42 (about 42%) can achieve lower survivable mapping cost by adding new logical links and the maximum/average cost saving ratio is 12.0%/3.5%. These results show that 1) GROUP1 has a larger number of non-survivable logical topologies than GROUP2, 2) among the survivable logical topologies, the percentage of improved ones is larger in GROUP1 than in GROUP2, and 3) the maximum/average cost saving ratio among the improved survivable logical topologies is larger in GROUP1 than in GROUP2. Thus, the overall improvement on GROUP1 is more than on GROUP2. This suggests that the new survivable mapping problem exhibits more significance on sparser logical topologies than on denser ones. This is intuitive because denser logical topologies are generally closer to survivable, and the room to reduce the survivable mapping cost is generally smaller in denser logical topologies. For RANDOM, the results in Table I also show the benefits of adding logical links to the given logical topology in enabling a survivable mapping and reducing minimal survivable mapping cost. Again, the overall improvement on GROUP1 is more than on GROUP2 since the logical topologies in GROUP1 are sparser than the logical topologies in GROUP2.

C. Running Time Comparison Between ILP1 and ILP2

To evaluate the running time improvement made by ILP2 over ILP1, we run ILP1 and ILP2 on GROUP1 over NSFNET and GROUP1 over RANDOM. For NSFNET, the average running time of ILP2 over all the 100 logical topologies in GROUP1 is 3505 seconds (about 1 hour), and the running time of ILP1 for a randomly selected logical topology in GROUP1 is 128593 seconds (about 35 hours and 43 minutes). (We didn’t run ILP1 for all the 100 logical topologies in GROUP1 due to its long running time.) For RANDOM, the average running time over the 100 logical topologies in GROUP1 taken by ILP1 and ILP2 are 544 seconds and 28 seconds respectively. The average speedup of ILP2 over ILP1 is 544 sec/28 sec ≈ 20. For both physical topologies, ILP2 runs much faster than ILP1. As explained in section IV-B, this is because ILP2 has fewer variables and fewer flow conservation constraints than ILP1.

VI. CONCLUSION

In this paper we proposed the following new survivable mapping problem: given a physical topology and a logical topology, compute a minimal cost survivable logical topology that contains the given logical topology and the corresponding minimal cost survivable mapping. The problem is significant for two reasons: 1) If the given logical topology is not survivable, we can add logical links to it to make it survivable; 2) If the given logical topology is survivable, we may reduce the minimal survivable mapping cost by adding logical links to it. We proved that a solution to the new survivable mapping problem always exists and provided a straightforward ILP formulation (ILP1) to solve the problem. Furthermore, we proved that we can find a solution to the new survivable mapping problem by only adding reflective logical links to the given logical topology, and the resulting logical topology has a reflectively-routed survivable mapping that achieves the minimal cost. Based on this result, we developed an improved ILP formulation (ILP2) that solves the new survivable mapping problem more efficiently and proved that the new survivable mapping problem is NP-hard. The benefits of adding logical links to a logical topology in enabling a survivable mapping and reducing minimal survivable mapping cost are demonstrated through simulations. Simulation results also show that ILP2 achieves significant speedup over ILP1.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant No. 0237592. The authors would like to thank the anonymous reviewers for their valuable comments which helped us improve the paper.

REFERENCES

We prove Theorem 2 in this section. First, we prove the following lemma.

**Lemma 1:** Given a physical topology $G_p = (V, E_p)$, for any survivable logical topology $G_l = (V, E_l)$ on $G_p$, there exists a set $E' \subseteq E_p - E_l$ such that $G_l' = (V, E_l \cup E')$ has a survivable reflectively-routed mapping $M'$ and $cost(M') \leq cost(G_l')$.

**Proof:** Let $M$ be a minimal cost survivable mapping from $G_l$ to $G_p$. If $M$ is reflectively-routed, just let $E' = \emptyset$. Then $G_l' = G_l$ has a survivable reflectively-routed mapping $M' = M$ and $cost(M') = cost(G_l)$.

If $M$ is not reflectively-routed, we call the procedure \textsc{Transform}($G_l, G_p, M$) to transform $M$ and $G_l$ such that the resulting $G_l'$ is obtained by adding links in $E_p - E(G_l^{old})$ to $G_l^{old}$ and the resulting $M$ is a survivable reflectively-routed mapping from $G_l$ to $G_p$ with $cost(M) \leq cost(G_l^{old})$, where $G_l^{old}$ and $M^{old}$ denote the old logical topology and the old mapping inputted to the procedure. The pseudocode of the procedure \textsc{Transform} is given below. The correctness proof of the procedure follows the pseudocode.

\textsc{Transform}($G_l, G_p, M$)

\begin{algorithm}
\begin{algorithmic}
\State \textbf{Procedure} \textsc{Transform}($G_l, G_p, M$)
\State \textbf{Input:} $G_l$, $G_p$, $M$
\State \textbf{Output:} $G_l^{old}$, $M^{old}$
\State \textbf{Procedure} \textsc{Transform}($G_l, G_p, M$)
\State \textbf{Input:} $G_l$, $G_p$, $M$
\State \textbf{Output:} $G_l^{old}$, $M^{old}$
\State \textbf{Procedure} \textsc{Remove}($G_l, G_p, M$, $st$)
\State \textbf{Input:} $G_l$, $G_p$, $M$, $st$
\State \textbf{Output:} $G_l^{old}$, $M^{old}$
\State \textbf{Procedure} \textsc{Remove}($G_l, G_p, M$, $st$)
\State \textbf{Input:} $G_l$, $G_p$, $M$, $st$
\State \textbf{Output:} $G_l^{old}$, $M^{old}$
\State \textbf{Procedure} \textsc{Critical}($G_l, G_p, M$, $st$)
\State \textbf{Input:} $G_l$, $G_p$, $M$, $st$
\State \textbf{Output:} $G_l^{old}$, $M^{old}$
\State \textbf{Procedure} \textsc{Critical}($G_l, G_p, M$, $st$)
\State \textbf{Input:} $G_l$, $G_p$, $M$, $st$
\State \textbf{Output:} $G_l^{old}$, $M^{old}$
\end{algorithm}
\end{algorithm}

**Observation 1:** In the resulting $G_l$, all links in $G_l^{old}$ are kept and the newly added links are all from $E_p - E(G_l^{old})$.

It’s easy to verify that there is no logical link deletion anywhere in \textsc{Transform} and in \textsc{RemoveCritical}.

The only place where logical link addition occurs is at line 3 of \textsc{RemoveCritical} and the added logical link belongs to $E_p - E(G_l^{old})$.

**Observation 2:** The cost of $M$ never increases.

Every time before \textsc{RemoveCritical} is called (at line 2 of \textsc{Transform} or at line 6 of \textsc{RemoveCritical}), a reflective logical link is rerouted from a multi-hop physical path to a single-hop physical path, which decreases the cost of $M$ by at least 1. On the other hand, within \textsc{RemoveCritical}, at most one logical link is added (at line 3). The added logical link is a reflective logical link and is reflectively-routed, which increases the cost of $M$ by 1. Thus, the cost of $M$ never increases.

In \textsc{Transform}, if $M$ is a survivable mapping from $G_l$ to $G_p$ at the beginning of an iteration of the for loop, then we have the following two claims.

**Claim 1:** After line 2 is executed and before \textsc{RemoveCritical} is called, $M$ is a non-survivable mapping from $G_l$ to $G_p$ with $st \in E_p$ being the critical link. The failure of $st$ will disconnect $G_l$ into two connected components with one component containing $s$ and the other component containing $t$.

**Claim 2:** \textsc{RemoveCritical} always returns. Moreover, when \textsc{RemoveCritical} returns, $M$ is a survivable mapping from $G_l$ to $G_p$.

If Claim 1 and Claim 2 hold, then each iteration of the for loop in \textsc{Transform} eliminates at least one non-reflectively-routed reflective logical link and end up with a survivable mapping $M$ from $G_l$ to $G_p$ with no new non-reflectively-routed reflective logical link being introduced. (\textsc{RemoveCritical} only adds reflectively-routed reflective logical links to $G_l$ at line 3.) Since there is a finite number of non-reflectively-routed reflective logical links, \textsc{Transform} always terminates with a survivable reflectively-routed mapping $M$ from $G_l$ to $G_p$. Together with Observation 1 and Observation 2, we know that \textsc{Transform} returns $G_l$ and $M$ such that $G_l$ is obtained by adding links in $E_p - E(G_l^{old})$ to $G_l^{old}$, and $M$ is a reflectively-routed survivable mapping from $G_l$ to $G_p$ with $cost(M) \leq cost(G_l^{old})$. Thus, we can prove Claim 1 and Claim 2, the proof of Lemma 1 is done. In the following, we give the proofs for Claim 1 and Claim 2.

**Proof of Claim 1:** Let $M^{before}/M^{after}$ denote the mapping before/after line 2 of \textsc{Transform} is executed. Assume $M^{after}$ is survivable. We have $cost(M^{after}) \leq cost(M^{before})$ since $st$ is rerouted from a multi-hop physical path to a single-hop physical path at line 2 of \textsc{Transform}. This contradicts the fact that $M^{before}$ is a minimal cost survivable mapping from $G_l$ to $G_p$. Therefore, $M^{after}$ must be non-survivable. Furthermore, $st \in E_p$ must be the only critical link under $M^{after}$ since $st \in E_p$ is the only physical link whose load set expands because of the reroute. (The reroute causes $st \in E'_l$ to be included in the load set of $st \in E_p$.) Since $st \in E_p$ becomes a critical link under $M^{after}$ and $st \in E'_l$.
is the only new logical link added to the load set of $st \in E_p$ due to the reroute, $st \in E_1$ must be a bridge in $G_i^{M^{\text{after}}}(st)$. Thus, under $M^{\text{after}}$, the failure of $st \in E_p$ will disconnect $G_i$ into two connected components, one containing $s$ and the other containing $t$.

(End of Proof of Claim 1)

**Proof of Claim 2:** The correctness of Claim 2 is based on the following four facts about the procedure REMOVE_CRITICAL_LINK. Each fact is followed by a proof.

**Fact 1:** At line 5, it is always possible to find $x$ and $y$ that meet the condition. And $xy$ found in line 5 is a non-reflectively-routed reflective logical link.

To enter line 5, we must have

$$\forall s' \in V(C_1), t' \in V(C_2), s't' \in E_p \Rightarrow s't' \in E_t \tag{\*}$$

Because $G_p$ is 2-edge-connected, the edge cut $EC_{C_1}(V(C_1))$ must contain at least one more physical link $xy \neq st$ besides $st$. By (\*), $xy$ must also be in $E_t$. Therefore, it’s always possible to find $x \in V(C_1)$ and $y \in V(C_2)$ such that $xy \in E_t \cap E_p$ and $xy \neq st$.

Since $st \in E_p$ is a critical link when line 5 is executed, we have $EC_{G_i}(V(C_1)) \subseteq I_M(st)$. Since $xy$ is in $EC_{G_i}(V(C_1))$, $xy$ is also in $I_M(st)$. This means that $xy \in E_t \cap E_p$ is routed on $st(\neq xy) \in E_p$. So $xy$ is a non-reflectively-routed reflective logical link.

**Fact 2:** After line 6 is executed, $st$ is not a critical link. Moreover, $xy$ is the only critical link whose failure will disconnect $G_i$ into two connected components, one containing $x$ and the other containing $y$.

After line 6 is executed, $xy \in E_t$ is no longer in the load set of $st \in E_p$. And because $x \in V(C_1)$ and $y \in V(C_2)$, the failure of $st \in E_p$ will not disconnect $G_i$ now. So $st$ is not a critical link. However, the mapping becomes non-survivable after line 6 is executed. Let $M^{\text{before}}/M^{\text{after}}$ denote the mapping before/after line 6 is executed. Assume $M^{\text{after}}$ is survivable. We have $\text{cost}(M^{\text{before}}) < \text{cost}(M^{\text{after}})$ since $xy$ is rerouted from a multi-hop physical path to a single-hop physical path at line 6. This contradicts the fact that $M^{\text{before}}$ is a minimal cost survivable mapping from $G_i$ to $G_p$. Therefore, $M^{\text{after}}$ must be non-survivable. Also, $xy \in E_p$ must be the only critical link under $M^{\text{after}}$ because $xy \in E_p$ is the only physical link whose load set expands because of the reroute. (The reroute causes $xy \in E_t$ to be included in the load set of $xy \in E_p$.) Since $xy \in E_p$ becomes a critical link under $M^{\text{after}}$ and $xy \in E_t$ is the only new logical link added to the load set of $xy \in E_p$ due to the reroute, $xy \in E_t$ must be a bridge in $G_{i}^{M^{\text{after}}}(xy)$. Thus, under $M^{\text{after}}$, the failure of $xy \in E_p$ will disconnect $G_i$ into two connected components, one containing $x$ and the other containing $y$.

**Fact 3:** After line 3 is executed, $s't' \neq st$ and neither $st \in E_p$ nor $s't' \in E_p$ is a critical link.

Before line 3 is executed, $st \in E_1$ and $s't' \notin E_1$, so $s't' \neq st$. After line 3 is executed, the newly added logical link $s't'$ is not routed on $st$, so the failure of $st \in E_p$ will not affect $s't' \in E_1$. As a result, the remaining logical topology upon the failure of $st \in E_p$ will be connected with $s't' \in E_i$ being a bridge between $C_1$ and $C_2$. Thus, $st \in E_p$ is no longer a critical link. As of $s't' \in E_p$, a new logical link $s't' \in E_1$ is added to the load set of $s't' \in E_p$ after line 3 is executed. Assume that $s't' \in E_p$ is critical now, it must be critical also before $s't' \in E_1$ is added to $G_1$, which contradicts the fact that $st \in E_p$ is the only critical link at that time (this fact is shown by Claim 1 if REMOVE_CRITICAL_LINK is called from line 3 of TRANSFORM, and by Fact 2 if REMOVE_CRITICAL_LINK is called from line 7 of itself). Thus, $s't' \in E_p$ is not a critical link after $s't' \in E_i$ is added to $G_i$.

**Fact 4:** After $st \in E_p$ becomes non-critical in REMOVE_CRITICAL_LINK, it will never become critical again till the end of TRANSFORM. Also, for each logical link $s't'$ added to $G_i$ at line 3 of REMOVE_CRITICAL_LINK, the corresponding physical link $s't'$ will never become critical again either.

After $st \in E_p$ becomes non-critical in REMOVE_CRITICAL_LINK, $st \in E_t$ is reflectively-routed. The load set of $st \in E_p$ will never include other logical links till the end of TRANSFORM because

1. All new logical links added at line 3 of REMOVE_CRITICAL_LINK will be reflectively-routed.
2. We only reroute non-reflectively-routed reflective logical links to make them reflectively-routed (at line 2 of TRANSFORM and at line 6 of REMOVE_CRITICAL_LINK).

So $st \in E_p$ will never become critical again.

Because of the same reasons, $s't' \in E_p$ will never become critical again either.

During the execution of REMOVE_CRITICAL_LINK, if the “then” branch is entered, Fact 3 tells us that $st \in E_p$ will become non-critical, and for the newly added logical link $s't' \in E_1$, the corresponding $s't' \in E_p$ is not critical either. On the other hand, if the “else” branch is entered, Fact 1 and Fact 2 tell us that $st$ will become non-critical and another physical link $xy$ will become critical. Thus, REMOVE_CRITICAL_LINK always eliminates one critical link (st) and may introduce another critical link (xy). Fact 4 guarantees that once $st$ becomes non-critical, it will never become critical again. Also, for each logical link $s't'$ added to the logical topology, the corresponding $s't' \in E_p$ will never become critical again either. Since we have a finite number of physical links that are potential critical links, REMOVE_CRITICAL_LINK will always return with no critical link existing in $G_p$. Therefore, when REMOVE_CRITICAL_LINK returns, $M$ is a survivable mapping from $G_i$ to $G_p$.

(End of Proof of Claim 2)

As argued earlier, Observations 1 and 2 together with Claims 1 and 2 prove Lemma 1.

We now give the proof of Theorem 2.

**Theorem 2:** Given a logical topology $G_i = (V,E_i)$ and a 2-edge-connected physical topology $G_p = (V,E_p)$, there exists an edge set $E'' \subseteq E_p - E_i$ such that $G''_i = (V,E_i \cup E'')$ is a minimal cost survivable logical topology that contains $G_i$ on $G_p$. Moreover, there is a reflectively-routed survivable mapping $M''$ from $G''_i$ to $G_p$ such that $\text{cost}(M'') = \text{MIN-COST}_{G_i}$.

**Proof:** Let $G = (V,E)$ be a minimal cost survivable logical topology that contains $G_i$ on $G_p$. Let $M$ be a minimal
cost survivable mapping from $G$ to $G_p$. Then $\text{cost}(M) = \text{cost}(G) = \text{MIN-COST}_{G_l}$.

CASE I: All logical links in $E - E_l$ are reflective, i.e., $E - E_l \subseteq E_p - E_l$.

If $M$ is a reflectively-routed mapping, then $E'' = E - E_l$, $G''_l = G$, and $M'' = M$ are the edge set, the logical topology, and the mapping we are looking for.

If $M$ is not a reflectively-routed mapping from $G$ to $G_p$, then by Lemma 1, there exists $E' \subseteq E_p - E$ such that $G' = (V, E \cup E')$ has a survivable reflectively-routed mapping $M'$ from $G'$ to $G_p$ and $\text{cost}(M') = \text{cost}(M)$. (It is impossible to get $\text{cost}(M') < \text{cost}(M)$ since $\text{cost}(M) = \text{MIN-COST}_{G_l}$.) Then $E'' = (E \cup E') - E_l$, $G''_l = G'$, and $M'' = M'$ are the edge set, the logical topology, and the mapping we are looking for.

CASE II: At least one logical link in $E - E_l$ is non-reflective, i.e., $\exists s \notin E_l$ such that $s \notin E_p$.

In this case, we call the procedure PURIFY($G_l, G_p, G, M$) to transform $G$ and $M$ so that the resulting $G$ is a minimal cost survivable logical topology that contains $G_l$ on $G_p$ and $E - E_l \subseteq E_p - E_l$. And the resulting $M$ is a reflectively-routed survivable mapping from $G$ to $G_p$ with $\text{cost}(M) = \text{MIN-COST}_{G_l}$. The pseudocode of PURIFY is given below. The correctness proof of PURIFY follows the pseudocode.

PURIFY($G_l, G_p, G, M$)

$G_l, G_p$: in parameter
$G, M$: inout parameter

1. if $M$ is not reflectively-routed then
2. Find $G' = (V, E \cup E')$ and $M'$ such that $E' \subseteq E_p - E$ and $M'$ is a reflectively-routed survivable mapping from $G'$ to $G_p$ and $\text{cost}(M') = \text{cost}(M)$;
3. let $G = G'$; $M = M'$;
4. for each $st \in (E - E_l) - E_p$ do
5. let $E = E - \{st\}$;
6. for each $ij \in M(st)$ do
7. let $E = E \cup \{ij\}$;
8. let $M(ij) = (i - j)$;

We have the following observations about PURIFY.

Observation 3: When PURIFY returns, $G$ contains all logical links in $G_l$ and all logical links in $E - E_l$ are reflective.

In PURIFY, logical link removal only occurs at line 5, where $st \in (E - E_l) - E_p$ is removed. Thus, all logical links in $G_l$ are kept in $G$. In the for loop from line 4 to line 8, each non-reflective logical link $st \in E - E_l$ is removed and replaced by a set of reflective logical links corresponding to the physical links in $M(st)$. Therefore, all logical links in $E - E_l$ are reflective.

Observation 4: The cost of $M$ never increases or decreases.

Clearly, the cost of $M$ cannot decrease because $\text{cost}(M) = \text{MIN-COST}_{G_l}$, when $M$ is inputted to PURIFY.

We now show that the cost of $M$ never increases. If the mapping $M$ inputted to PURIFY is not a reflectively-routed mapping, then line 2 is executed. By Lemma 1, we can successfully find $G'$ and $M'$ at line 2 and the cost of the mapping does not increase. In each iteration of the for loop at line 4, on the one hand, $st \in (E - E_l) - E_p$ is removed from $G$, which decreases the cost of $M$ by $|M(st)|$; on the other hand, at most $|M(st)|$ reflectively-routed logical links are added to $G$, which increases the cost of $M$ by at most $|M(st)|$. Thus, the cost of $M$ does not increase in the for loop. Overall, the cost of $M$ never increases in PURIFY.

Within one iteration of the for loop at line 4 in PURIFY, we use $G^{before}$, $M^{before}$ and $G^{after}$, $M^{after}$ to denote the logical topology/mapping before removing $st$ and after adding $ij$'s in $M(st)$ and mapping them reflectively. For each iteration of the for loop, if $M^{before}$ is a survivable mapping, then we have the following two claims.

Claim 3: $\forall ij \in M^{before}(st)$, $ij$ is not a critical link under $M^{after}$.

Claim 4: $\forall ij \in E_p - M^{before}(st)$, $ij$ is not a critical link under $M^{after}$.

If Claim 3 and Claim 4 hold, each iteration of the for loop at line 4 eliminates exactly one non-reflective logical link in $E - E_l$ without breaking the survivability of the mapping or introducing any new non-reflective logical links. Since we have a finite number of non-reflective logical links in $E - E_l$, PURIFY always terminates with $M$ being a survivable mapping from $G$ to $G_p$. In addition, $M$ is a reflectively-routed mapping when PURIFY terminates. This is because $M$ is a reflectively-routed mapping before the for loop is executed and all the logical links added in the for loop are reflective and are reflectively-routed. By Observation 3 and Observation 4, the logical topology $G = (V, E)$ returned by PURIFY is a minimal cost survivable logical topology that contains $G_l$ on $G_p$ and $E - E_l \subseteq E_p - E_l$, and the mapping $M$ returned by PURIFY is a reflectively-routed survivable mapping from $G$ to $G_p$ with $\text{cost}(M) = \text{MIN-COST}_{G_l}$.

In the following, we give the proofs for Claim 3 and Claim 4.

Proof of Claim 3: For all $ij \in M^{before}(st)$, we have

\[ G^{after} \]

Fig. 4. Illustration for the Proof of Claim 3. Solid (thin and thick) lines are logical links in $l_{G^{after}}(ij)$. Thick solid lines are logical links in $E_{G^{after}}(V \cup \{C_i\})$. The dashed line is $st \in E(G^{after})$. The dotted lines denote the logical path from $s$ to $i$ and the logical path from $j$ to $t$ in $G^{after}$. This diagram shows that if the removal of $l_{G^{after}}(ij)$ disconnects $G^{after}$, then the removal of $l_{G^{after}}(ij) = \{l_{G^{after}}(ij)\} \cup \{st\}$ would also disconnect $G^{before}$. 

Proof of Claim 4: For all $ij \in E_p - M^{before}(st)$, we have
\[ l_{M_{\text{after}}}(ij) = (l_{M_{\text{before}}}(ij) - \{st\}) \cup \{ij\}. \]

Assume that \( ij \in E_p \) is critical under \( M_{\text{after}} \), then the failure of \( ij \in E_p \) will disconnect the \( G_{\text{after}} \) into two connected components \( C_1 \) and \( C_2 \), one containing \( i \) and the other containing \( j \). This is because \( ij \in E_p \) is not critical under \( M_{\text{before}} \) and \( ij \in E(G_{\text{after}}) \) is the only logical link that is contained in \( l_{M_{\text{after}}}(ij) \) but not in \( l_{M_{\text{before}}}(ij) \).

On the other hand, all logical links in \( E(G_{\text{after}}) \) except \( ij \) along the logical path corresponding to \( M_{\text{before}}(st) \) are reflectively-routed, so these logical links will not be affected by the failure of \( ij \in E_p \) under \( M_{\text{after}} \). In other words, only \( ij \in E(G_{\text{after}}) \) is broken on the logical path corresponding to \( M_{\text{before}}(st) \) when \( ij \in E_p \) fails. Without loss of generality, suppose \( s, i \in V(C_1) \) and \( t, j \in V(C_2) \). As can be seen from Fig. 4, if \( ij \in E_p \) is critical under \( M_{\text{after}} \), \( ij \in E_p \) must be critical under \( M_{\text{before}} \) because removing \( l_{M_{\text{before}}}(ij) = (l_{M_{\text{after}}}(ij) - \{ij\}) \cup \{st\} \) from \( G_{\text{before}} \) would disconnect \( G_{\text{before}} \). This contradicts the fact that \( M_{\text{before}} \) is survivable. Thus, \( ij \in E_p \) is not critical under \( M_{\text{after}} \).

(End of Proof of Claim 3)

**Proof of Claim 4:** Consider \( ij \in E_p - M_{\text{before}}(st) \). Assume \( ij \) is critical under \( M_{\text{after}} \). Since the load set of \( ij \) under \( M_{\text{after}} \) is the same as that under \( M_{\text{before}} \), the only possible reason that makes \( ij \) critical under \( M_{\text{after}} \) is the loss of \( st \in E(G_{\text{before}}) \) in \( G_{\text{after}} \). So, the failure of \( ij \) will disconnect \( G_{\text{after}} \) into two connected components, one containing \( s \) and the other containing \( t \). However, this is impossible because there exists a path in \( G_{\text{after}} \) from \( s \) to \( t \) when \( ij \) fails since all the logical links along the logical path corresponding to \( M_{\text{before}}(st) \) are reflectively-routed under \( M_{\text{after}} \) and therefore not affected by the failure of \( ij \). Thus, \( ij \) is not a critical link under \( M_{\text{after}} \).

(End of Proof of Claim 4)

As argued earlier, Observations 3 and 4 together with Claims 3 and 4 prove Theorem 2.