On Survivable Routing of Mesh Topologies in IP-over-WDM Networks

Maciej Kurant, Patrick Thiran
LCA - School of Communications and Computer Science
EPFL, CH-1015 Lausanne, Switzerland
Email: {maciej.kurant, patrick.thiran}@epfl.ch

***Abstract*— Failure restoration at the IP layer in IP-over-WDM networks requires to map the IP topology on the WDM topology in such a way that a failure at the WDM layer leaves the IP topology connected. Such a mapping is called *survivable*. Finding a survivable mapping is known to be NP-complete [1], making it impossible in practice to assess the existence or absence of such a mapping for large networks. (i) We first introduce a new concept of *piecewise survivability*, which makes the problem much easier, and allows us to formally prove that a given survivable mapping does or does not exist. (ii) Secondly, we show how to trace the vulnerable areas in the topology, and how to strengthen them to enable a survivable mapping. (iii) Thirdly, we give an efficient and scalable algorithm that finds a survivable mapping. In contrast to the heuristics proposed in the literature to date, our algorithm exhibits a number of provable properties that are crucial for (i) and (ii). We consider both link and node failures at the physical layer.**

***Index Terms*— IP-over-WDM, link and node failures, survivability, graph theory**

I. INTRODUCTION

Generally, there are two approaches for providing survivability of IP-over-WDM networks: protection and restoration [2]. Protection uses pre–computed backup paths applied in the case of a failure. Restoration finds dynamically a new path, once a failure has occurred. Protection is less resource efficient (the resources are committed without prior knowledge of the next failure) but fast, whereas restoration is more resource efficient and slower. Protection and restoration mechanisms can be provided at different layers. *IP layer* (or *logical* layer) survivability mechanisms can handle failures that occur at both layers, contrary to *WDM layer* (or *physical* layer) mechanisms that are transparent to the IP topology. It is not obvious which combination (mechanism/layer) is the best; each has pros and cons [3]. IP restoration, however, deployed in some real networks, was shown to be an effective and cost–efficient approach (see e.g., Sprint network [4]). In this paper we will consider exclusively the *IP restoration approach*.

Each logical (IP) link is mapped on the physical (WDM) topology as a *lightpath*. Usually a fiber is used by more than one lightpath (in Sprint the maximum number is 25 [5]). Therefore a single physical link failure usually brings down a number of IP links. With the IP restoration mechanism, these IP link failures are detected by IP routers, and alternative routes in the IP topology are found. In order to enable this, the IP topology should remain *connected* after a failure of a physical link; this in turn may be guaranteed by an appropriate mapping of IP links on the physical topology. We call such a mapping a *link-survivable mapping*.

Physical link failure is a common type of failure, but not the only one. We can also encounter a physical *node* failure (e.g., a failure of an optical switch); it is equivalent to the failure of every fiber beginning in the failing node, making the problem more difficult. If, after any single physical node failure, the logical topology (excluding the failing node) remains connected, then the mapping is declared to be *node-survivable*.

In this paper we consider both link- and node-survivability. Firstly, we are interested in the *existence* of a (link- or node-) survivable mapping for a given pair of logical and physical topologies. There is some work on the topic in the literature, but it assumes *ring topologies* at the physical [6], [7] or the logical [1], [8] layer. We study the existence of a link/node-survivable mapping for general mesh topologies at both layers, which is foreseen to be the main future topology. To date, the only general method verifying the existence of a survivable mapping was an exhaustive search (or equivalent) run for the *entire* topology. Due to NP-completeness of the survivable mapping problem [1], the exhaustive approach is not realizable in practice for the topologies larger than a few nodes. To bypass this difficulty, we introduce a new type of mapping, which preserves the survivability of some subgraphs ('pieces') of the logical topology; we call it a *piecewise survivable mapping*. The formal analysis of the piecewise survivable mapping shows that a survivable mapping of the logical topology on the physical topology exists if and only if there exists a survivable mapping for a *contracted* logical topology, that is, a logical topology where a specified subset of edges is contracted (contraction of an edge amounts to removing it and merging its end-nodes). This new result substantially simplifies the verification of the existence of a survivable mapping, making it, for the first time, often possible for moderate and large topologies.

A second application of a piecewise survivable mapping is tracing the vulnerable areas in the network and pointing where new link(s) should be added to enable a survivable mapping. To the best of our knowledge, this is also a novel functionality.

Thirdly, the formal analysis reveals an easy way to incrementally expand the survivable pieces in a piecewise survivable mapping. This leads us to SMART - an efficient and scalable algorithm that searches for a survivable mapping. SMART is different from the algorithms solving this problem proposed in the literature. These algorithms can be divided into two groups:

(i) greedy search based on Integer Linear Programming (ILP), and (ii) heuristics. The ILP solutions can be found for example in [1], [9]. However, this approach leads to an unacceptably high complexity, even for networks of small size (few tens of nodes). The second approach uses various heuristics, such as Tabu Search [9], [10], [11], Simulated Annealing [3] and others [2]. If a heuristic fails, nothing can be claimed about the existence of a survivable mapping. We introduced the SMART algorithm in [12] as such a heuristic, without any formal analysis. Simulations in [12] showed that SMART is efficient and scales very well. The concept of piecewise survivability introduced in the present paper makes the formal analysis of SMART possible. It revealed that the SMART algorithm actually opens a third group (iii) in the family of algorithms that search for a survivable mapping. One of our key results is that, contrary to the heuristics (ii), SMART never misses a solution if there is one. This is because, even if SMART does not fully converge, the mapping it returns is piecewise survivable. This mapping is defined for a subset of logical links, and leaves the remaining logical links unmapped. We prove that if a survivable solution exists, the remaining unmapped logical links can be still mapped in a way ensuring the survivability of the resulting full mapping.

In contrast to physical link failures, physical *node* failures were rarely addressed before. The solutions proposed in the literature (e.g., [13]) are protection/restoration mechanisms at the WDM layer, but not at the IP layer. To the best of our knowledge, this work is the first one to formally address node failures by an IP restoration approach.

Many of the approaches mentioned above take as a parameter the number of wavelentgths in each fiber, i.e., take fiber capacities into account. Clearly, this better reflects real-life situations. The approach in this paper, like the approaches in [1], [10], releases the capacity constraints, by assuming infinite capacities on each physical link. Hence our approach deals only with topological constraints, not with capacity limitations. This has pros and cons. On one hand, this makes the 'negative results' more general; if we prove that a survivable mapping does not exist for a particular pair of physical and logical topologies with infinite physical capacities, then this proof holds for any combination of finite capacities. On the other hand, the 'positive result' (i.e., a survivable mapping) found for infinite capacities is not necessarily applicable to a scenario with given finite capacities.

The organization of this paper is the following. Section II introduces notations and formalizes the problem. Section III gives two fundamental theorems. For better readability, their proofs are moved to the Appendix. Section IV introduces the SMART algorithm and describes its implementation. Section V discusses a number of possible applications of SMART. Section VI presents the simulation results. Section VII concludes the paper.

II. Notation and problem formulation

A. Generalities

We use the formal notation of graph theory, mainly based on [14]. However, we also introduce the stack of our definitions well suited to the problems we tackle. The following general notation is used:

- ϕ corresponds to the *physical* topology,
- L corresponds to the *logical* topology,
- C corresponds to the *contracted* topology (introduced later in Section II-C),
- $a, b, c, d, e \ldots$ are used to denote edges/links,[1]
- $u, v, w \ldots$ are used to denote vertices/nodes,[2]
- p is used to denote a path, i.e., an alternate sequence of nodes and edges, where two consecutive edges have a common end-node that appears between them in the sequence. A path p from vertex v to vertex u will be denoted by $p_{v,u}$.

Physical and logical topologies are represented by undirected simple graphs: $G^\phi = (V, E^\phi)$ and $G^L = (V, E^L)$, respectively. V is the set of vertices, E^ϕ and E^L are the sets of undirected edges. In reality, not every physical node (i.e., optical switch) has an IP routing capability, which would imply $V^\phi \supseteq V^L$. All the the results in this paper hold for $V^\phi \supseteq V^L$, but for the sake of simplicity we have chosen to keep V^ϕ and V^L identical ($V^\phi \equiv V^L \equiv V$).

B. Lightpath and mapping

Definition 1 (Lightpath): A logical link e^L is mapped on a physical topology as a physical path p^ϕ in such a way that p^ϕ connects the same two vertices in G^ϕ as e^L connects in G^L. In optical networking terminology, such a physical path p^ϕ is called a *lightpath*. The failure of any physical link in p^ϕ breaks the lightpath and consequently brings down the logical link e^L. Note that, since we release the capacity constraints, we do not have to consider the wavelengths assigned to lightpaths and wavelength converters placement.

Definition 2 (Mapping): Let P^ϕ be a set of all possible physical paths in the physical topology and $A \subset E^L$ be a set of logical links. A *mapping* M_A is a function $M_A : A \to P^\phi$ associating each logical link from the set A with a corresponding lightpath in the physical topology.

For some particular logical edge $e^L \in A$, M_A returns a physical path $p^\phi = M_A(e^L),\ p^\phi \in P^\phi$. For arguments beyond A, M_A is not defined. We also allow putting a set of logical links $A_{sub} \subset A$ as an argument, which results in a set of lightpaths $M_A(A_{sub}) \subset P^\phi$. Similarly, taking as an argument a logical path p^L whose edges are in A, we obtain a set of lightpaths $M_A(p^L) \subset P^\phi$ associated with the edges of p^L (nodes of p^L are ignored).

Example 1: Fig. 1 illustrates the definitions given above. In Fig. 1a the mapping M_A is defined for the subset A of logical links (marked in bold in the logical topology). For example, we have $M_A(a^L) = \langle c^\phi, d^\phi \rangle$, which means that the lightpath assigned for the logical edge a^L consists of two physical links, c^ϕ and d^ϕ. Fig. 1b presents a mapping defined for the subset B, whereas the mapping M_{E^L} in Fig. 1c is defined for all links of the logical topology $E^L = A \cup B$.

[1]The terms *edge* and *link* will be used interchangeably

[2]The terms *vertex* and *node* will be used interchangeably

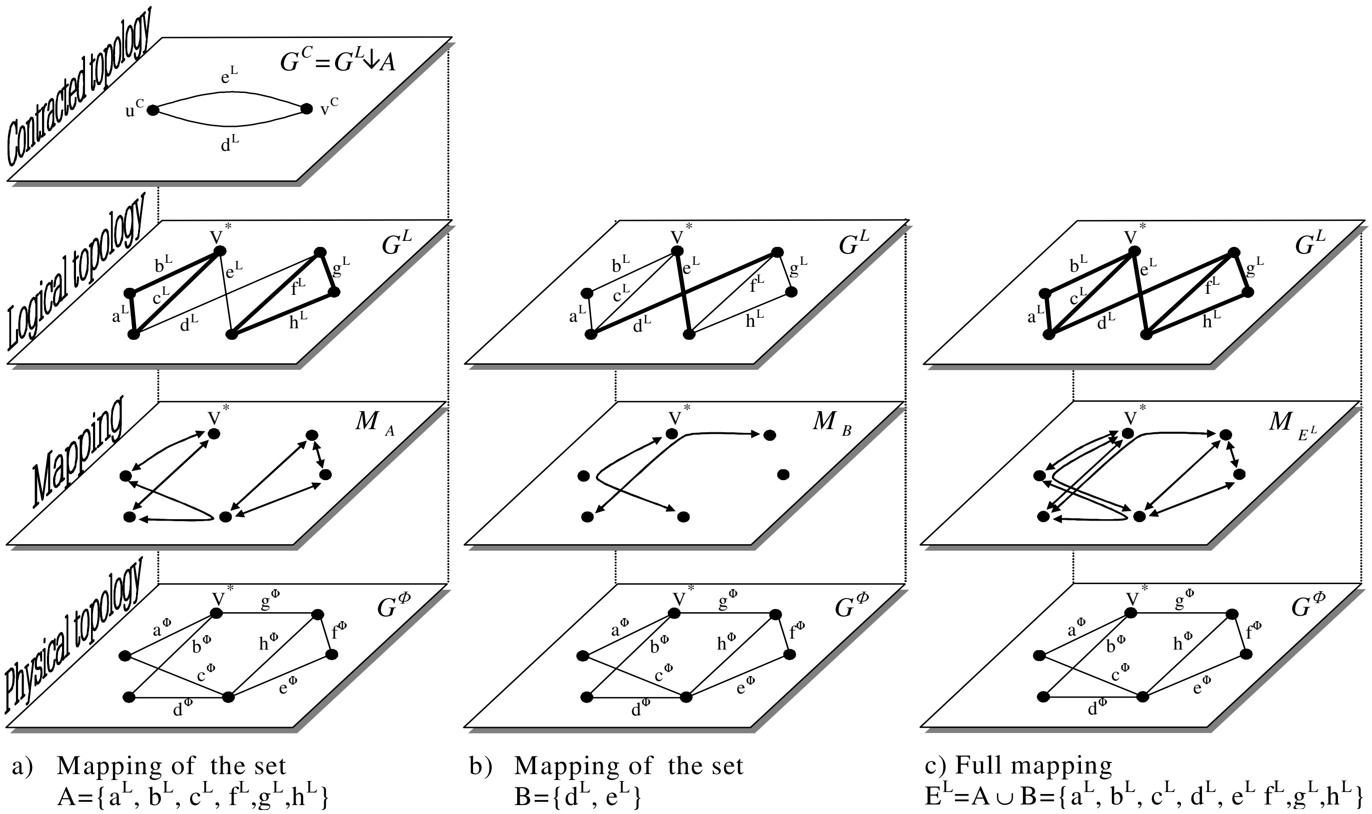


Fig. 1. Three mapping examples. We have four layers, from bottom to top: the physical topology G^ϕ, the mapping M, the logical topology G^L and the contracted logical topology G^C (only in (a)). In (a) the pairs $\left[G^L_{\{a^L,b^L,c^L\}}, M_A\right]$ and $\left[G^L_{\{f^L,g^L,h^L\}}, M_A\right]$ are link- and node-survivable, and therefore the pair $\left[G^L, M_A\right]$ is piecewise link- and node-survivable. In (b) the mapping M_B maps edge-disjointly the set $B = \{d^L, e^L\}$ of two logical links. The contracted topology G^C in (a) is composed of these two links. Taking G^C and B together, we obtain the pair $\left[G^C, M_B\right]$, which is link-survivable, but not node-survivable. In (c) the pair $\left[G^L, M_{E^L}\right]$ is link-survivable, that is M_{E^L} is a link-survivable mapping of the entire logical topology.

We will often deal with mappings of different subsets of logical edges. Let $A_1, A_2 \subset E^L$. For consistency, we always require that:

$$\text{for every } e^L \in A_1 \cap A_2 : \; M_{A_1}(e^L) = M_{A_2}(e^L). \quad (1)$$

The mappings M_{A_1} and M_{A_2} can be merged, resulting in a mapping M_{A_3} defined as follows

$$A_3 = A_1 \cup A_2 \quad (2)$$

$$M_{A_3}(A_3) = M_{A_1}(A_1) \cup M_{A_2}(A_2). \quad (3)$$

For convenience of notation, we will write (2) and (3) as $M_{A_3} = M_{A_1} \cup M_{A_2}$.

C. Contraction and Origin

In the paper we will often use the graph operator of *contraction*, which is illustrated in Fig. 2 and is defined as follows:

Definition 3 (Contraction [14]): Contracting an edge $e \in E$ of a graph $G = (V, E)$ consists in deleting that edge and merging its end–nodes into a single node. The result is called the *contraction of a graph* G *on an edge* e (or simply a *contracted graph*), and is denoted by $G^C = G\downarrow e$.
By extension, we also allow contracting a set of edges $A \subset E$, resulting in a contracted graph $G^C = G\downarrow A$, obtained by successively contracting the graph G on every edge of A. It is easy to show that the order in which the edges of A are taken to contraction, does not affect the final result.

Let $G = (V, E)$, $A \subset E$ and $G^C = (V^C, E^C) = G\downarrow A$.

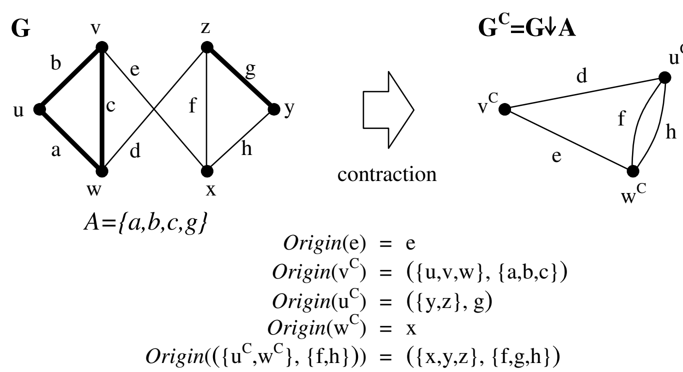


Fig. 2. Contraction of a graph G on a set of edges $A = \{a, b, c, g\}$ ($G^C = G\downarrow A$). The origins of some elements of G^C are also shown (bottom).

Note that by construction $E^C = E \setminus A$. Therefore each edge of G^C can be found in G, as depicted in Fig. 2. This is not always true for vertices. A vertex of V^C may either 'originate' from a single vertex in G (like w^C in Fig. 2), or from a connected subgraph of G (like v^C and u^C). We call this relation an $Origin(\cdot)$.

Definition 4 (Origin): Let $G^C = G \downarrow A$. Now take a subgraph $G^C_{sub} \subseteq G^C$. We say that $G_{sub} = Origin(G^C_{sub})$, if G_{sub} is the maximal subgraph of G that was transformed into G^C_{sub} by the contraction of A in G.

According to this definition, the result of the $Origin(\cdot)$ function is the *maximal* subgraph transformed in its argument. For example, one could say that in Fig. 2, the vertex $z \in G$ was transformed into the vertex $u^C \in G^C$, however $z \neq Origin(u^C)$ because it is not the only element that was transformed into u^C by contraction. The maximal subgraph in this case is $(\{y, z\}, g) = Origin(u^C)$.

D. Survivability and piecewise survivability

Let M_{E^L} be a mapping of the logical topology G^L on the physical topology G^ϕ. Assume that a physical link e^ϕ fails. Each logical link in G^L using e^ϕ in its mapping (lightpath) will than be cut. This may cause a disconnection of G^L. If, after any single physical link failure, the graph G^L remains connected, then the pair $\left[G^L, M_{E^L}\right]$ is declared *link-survivable*. We extend this property to a family of graphs constructed from the logical topology, and to the other type of failures, in the following definition:

Definition 5 (Link- and node-survivability): Let $G^L = (V, E^L)$, $A \subset E^L$ and $G^C = (V^C, E^C) = G^L \downarrow A$. Take any connected subgraph $G^C_{sub} = (V^C_{sub}, B)$ of the contracted topology G^C, and let M_B be a mapping of the set B of logical links. The pair $\left[G^C_{sub}, M_B\right]$ is *link-survivable* if the failure of any single physical link e^ϕ does not disconnect the graph G^C_{sub}. The pair $\left[G^C_{sub}, M_B\right]$ is *node-survivable* if the failure of any single node $v^* \in V$ does not disconnect the graph

$$G^C_{sub} \backslash \{v^C \in G^C_{sub} : \quad Origin(v^C) = v^*\}. \tag{4}$$

The graph (4) is G^C_{sub} minus one vertex, if its origin is the *single* vertex v^*. If v^* is a part of a larger subgraph of G^L, which was contracted into a single node in G^C_{sub}, then the graph (4) is G^C_{sub}. For example, in Fig 2, let $G^C_{sub} = G^C$. For $v^* = x$, the graph (4) is equal to $G^C_{sub} \backslash \{w^C\} = \{\{v^C, u^C\}, \{e\}\}$. But if $v^* = u$ then (4) is equal to the entire G^C_{sub}. This modification in the definition of node-survivability, compared to that of link-survivability, is necessary, because if node v^* fails it trivially disconnects the logical topology into $G^L \backslash \{v^*\}$ and $\{v^*\}$. The node-survivability definition applies therefore to $G^L \backslash \{v^*\}$.

A direct consequence of Definition 5 is that if $\left[G^C_{sub}, M_B\right]$ is link/node-survivable, then $\left[G^C_{sub}, M_{B'}\right]$ is also link/node-survivable, for any $B \subset B' \subseteq E^L$.

In Definition 5, G^C_{sub} represents a large family of graphs obtained from the logical topology. If $A = \emptyset$, then $G^C = G^L$ and G^C_{sub} is any connected subgraph of G^L (including G^L itself). If $A \neq \emptyset$, then G^C_{sub} is any connected subgraph of $G^L \downarrow A$. The different instances of G^C_{sub} and survivable pairs are given in Fig. 1 and described in the following three examples:

Example 2: One can check that in Fig. 1c the pair $\left[G^L, M_{E^L}\right]$ is link-survivable. It is not, however, node-survivable, because a failure of node v^* splits the remaining logical topology $G^L \backslash \{v^*\}$ into two graphs.

Example 3: In Fig. 1a, let $G^L_{\{a^L, b^L, c^L\}}$ be the subgraph of G^L defined by the edges a^L, b^L, c^L and their end-vertices. The pair $\left[G^L_{\{a^L, b^L, c^L\}}, M_A\right]$ is link-survivable, since the failure of any physical link does not disconnect $G^L_{\{a^L, b^L, c^L\}}$. The pair $\left[G^L_{\{a^L, b^L, c^L\}}, M_A\right]$ is also node-survivable, because the failure of any physical node $v \in V$ does not disconnect $G^L_{\{a^L, b^L, c^L\}} \backslash \{v\}$. Similarly, the pair $\left[G^L_{\{f^L, g^L, h^L\}}, M_A\right]$ is also link- and node-survivable.

Example 4: In Fig. 1a, the contracted topology G^C is the result of the contraction of the logical topology on the set A, i.e., $G^C = G^L \downarrow A$. Take $G^C_{sub} = G^C$. It consists of two logical links, d^L and e^L. An example mapping of the set $B = \{d^L, e^L\}$ is the mapping M_B shown in Fig 1b. Consider the pair $\left[G^C, M_B\right]$; it is link-survivable, because a single physical link failure cannot bring down both d^L and e^L at the same time, hence G^C will remain connected. However, the pair $\left[G^C, M_B\right]$ is *not* node-survivable, because the failure of the node v^* splits G^C. (Note that $v^* \in Origin(u^C)$, but $v^* \neq Origin(u^C)$, so the graph (4) in this case is the entire G^C.) Moreover, it is easy to check that no mapping $M_{\{d^L, e^L\}}$ forms a node-survivable pair with G^C, because the lightpath associated with d^L must go through at least one of the end-nodes of e^L. The failure of this node brings down both d^L and e^L, disconnecting G^C.

Definition 6 (Piecewise survivability): Let M_A be a mapping of a set $A \subset E^L$ on the physical topology. The pair $\left[G^L, M_A\right]$ is *piecewise link/node-survivable* if, for every vertex v^C of the contracted logical topology $G^L \downarrow A$, the pair $\left[Origin(v^C), M_A\right]$ is link/node-survivable.

Unlike survivability, piecewise survivability is defined only for the entire logical topology G^L. We will say that a mapping M_A is (piecewise) link/node-survivable, if the pair $\left[G^L, M_A\right]$ is (piecewise) link/node-survivable (i.e., we take G^L as the default topology).

Example 5: In Fig. 1a, the pair $\left[G^L, M_A\right]$ is piecewise link- and node-survivable. To prove it, we have to show that for vertices u^C and v^C of $G^L \downarrow A$, the pairs $\left[Origin(u^C), M_A\right]$ and $\left[Origin(v^C), M_A\right]$ are link- and node-survivable. Here we have $Origin(u^C) = G^L_{\{a^L, b^L, c^L\}}$ and $Origin(v^C) = G^L_{\{f^L, g^L, h^L\}}$. We have shown in Example 3, that each of these two graphs forms a link- and node-survivable pair with M_A.

III. Two fundamental properties of piecewise survivable mappings

In this section we prove two useful properties of a piecewise link/node-survivable mapping.

A. The expansion of survivability

Given a piecewise link/node-survivable mapping, the logical topology can be viewed as a set of link/node-survivable 'pieces'. This is a general property of a piecewise link/node-survivable mapping. (For instance in Example 5, given the piecewise survivable mapping M_A, there are two link/node-survivable 'pieces' of G^L: $G^L_{\{a^L, b^L, c^L\}} \subset G^L$ and $G^L_{\{f^L, g^L, h^L\}} \subset G^L$.) The following theorem enables us to merge some of these pieces, resulting in a single large link/node-survivable piece.

Theorem 1 (Expansion of survivability): Let M_A be a mapping of a set of logical edges $A \subset E^L$ on the physical topology G^ϕ, such that the pair $\left[G^L, M_A\right]$ is piecewise link/node-survivable. Let $G^C = G^L{\downarrow}A$. Take any subgraph of G^C, call it $G^C_{sub} = (V^C_{sub}, B)$. Let M_B be a mapping of the set B of edges of G^C_{sub} on G^ϕ. If the pair $\left[G^C_{sub}, M_B\right]$ is link/node-survivable then the pair $\left[Origin(G^C_{sub}), M_A \cup M_B\right]$ is also link/node-survivable.

Proof: See Appendix. ■

The following example illustrates this theorem.

Example 6: In Example 5 we have shown that in Fig. 1a, the pair $\left[G^L, M_A\right]$ is piecewise link- and node-survivable (here we will use only its piecewise link-survivability). Take $G^C_{sub} = G^C = G^L{\downarrow}A$ and take M_B as in Fig. 1b. From Example 4, we know that the pair $\left[G^C, M_B\right]$ is link-survivable. Now, by Theorem 1, the pair $\left[Origin(G^C), M_A \cup M_B\right] = \left[G^L, M_A \cup M_B\right]$ is link-survivable. So starting from the piecewise link-survivable mapping M_A and adding the mapping M_B, we merged the two link-survivable pieces $G^L_{\{a^L,b^L,c^L\}}$ and $G^L_{\{f^L,g^L,h^L\}}$ into a single, large, link-survivable piece. In this example the resulting link-survivable piece is the entire logical topology G^L. The full mapping $M_A \cup M_B = M_{E^L}$ is shown in Fig. 1c.

B. The existence of a survivable mapping

In general, for a given pair of physical and logical topologies, it is very difficult to verify the existence of a link/node-survivable mapping. A heuristic approach, if it fails, does not give any answer. The ILP approach or an exhaustive search could provide us with the answer, but due to their high computational complexity their application is limited to the topologies of several nodes. The following theorem shows how this verification problem can be substantially reduced:

Theorem 2 (Existence of a survivable mapping): Let M_A be a mapping of a set of logical edges $A \subset E^L$, such that the pair $\left[G^L, M_A\right]$ is piecewise link/node-survivable. A link/node-survivable mapping $M^{surv}_{E^L}$ of G^L on G^ϕ exists if and only if there exists a mapping $M^{surv}_{E^L \setminus A}$ of the set of logical links $E^L \setminus A$ on G^ϕ, such that the pair $\left[G^L{\downarrow}A, M^{surv}_{E^L \setminus A}\right]$ is link/node-survivable.

Proof: See Appendix. ■

The following example illustrates this theorem.

Example 7: For the pair of logical and physical topologies presented in Fig. 1, the node-survivable mapping does not exist. To prove it, take the mapping M_A, as in Fig. 1a. We know from Example 5 that the pair $\left[G^L, M_A\right]$ is piecewise node-survivable. Note that $E^L \setminus A = \{d^L, e^L\}$. From Example 4 we know that no mapping $M_{\{d^L,e^L\}}$ forms a node-survivable pair with the contracted logical topology $G^L{\downarrow}A$. Therefore, by Theorem 2 we know that no node-survivable mapping of G^L on G^ϕ exists. Note that to prove it, we only considered the two-edge topology $G^L{\downarrow}A$ instead of the entire G^L, which greatly simplified the problem. Clearly, the larger the set A, the more we benefit from Theorem 2.

IV. THE SMART ALGORITHM

In this section we present an algorithm that searches for a link/node-survivable mapping. We call this algorithm SMART (this acronym stands for "Survivable Mapping Algorithm by Ring Trimming", which will be explained at the end of this section). It maps the topology part by part, gradually converging to a final solution. By formal graph theoretic analysis, we prove that if SMART converges completely, a *link/node-survivable* mapping is found. Otherwise, when the algorithm terminates before its complete convergence, a returned mapping is *piecewise link/node-survivable*.

The pseudo-code of SMART is:

- *Step 1* Start from the full logical topology $G^C = G^L$, and an empty mapping $M_A = \emptyset, A = \emptyset$;
- *Step 2* Take some subgraph $G^C_{sub} = (V^C_{sub}, B)$ of G^C and find a mapping M_B, such that the pair $\left[G^C_{sub}, M_B\right]$ is link/node-survivable. IF no such pair is found, THEN RETURN M_A AND $G^C = G^L{\downarrow}A$, END.
- *Step 3* Update the mapping by merging M_A and M_B, i.e., $M_A := M_A \cup M_B$;
- *Step 4* Contract G^C on B, i.e., $G^C := G^C{\downarrow}B$;
- *Step 5* IF G^C is a single node, THEN RETURN M_A, END.
- *Step 6* GOTO Step 2

(The choice of "link" or "node" in Step 2 results in obtaining a (piecewise) link- or node-survivable mapping, respectively.)

The SMART algorithm starts from an empty mapping $M_A = \emptyset$. At each iteration it maps some set B of logical links (Step 2), and, in the case of a success, extends the mapping M_A by M_B (Step 3). Meanwhile, the contracted topology G^C gradually shrinks (Step 4). We will declare that:

- *SMART converges* if the contracted topology G^C converges to a *single* node. We prove later in Corollary 1, that the mapping M_A returned in step 5 is then a link/node-survivable solution;
- *SMART does not converge* if SMART terminates before G^C converges to a single node. This may happen when Step 2 of SMART is impossible or hard to make. The mapping M_A returned in Step 2 is not a link/node-survivable solution. However, we prove below in Theorem 3, that the pair $\left[G^L, M_A\right]$ is piecewise link/node-survivable. The graph $G^C = G^L \downarrow A$ is also returned in Step 2. We call it the *remaining contracted logical topology* since it consists of unmapped logical links $E^L \setminus A$.

Theorem 3 (SMART's piecewise survivability): After each iteration of the SMART algorithm, the pair $\left[G^L, M_A\right]$ is piecewise link/node-survivable.

Proof: See Appendix. ■

Corollary 1 (SMART's correctness): If, in the SMART algorithm, the contracted topology G^C converges to a single node topology, then the pair $\left[G^L, M_A\right], A = E^L$, is link/node-survivable.

Proof: (i) By Theorem 3, the pair $\left[G^L, M_A\right]$ is piecewise link/node-survivable. So for every vertex $v^C \in G^C$ the pair $\left[Origin(v^C), M_A\right]$ is link/node-survivable. (ii) There is only one vertex in G^C (i.e., $G^C = \{v^C\}$), and therefore

$Origin(v^C) = G^L$. Combining (i) and (ii), we have that $[G^L, M_A]$ is link/node-survivable. ■

G^C may converge to a single node topology with *self-loops*, which are the set of remaining unmapped logical links $E^L \backslash A$. However, this does not affect the result, as the links of $E^L \backslash A$ may be mapped in any way (e.g. shortest path) to obtain a link/node-survivable mapping M_{E^L}.

In the *implementation* of the SMART algorithm we take the graph G^C_{sub} in Step 2 in the form of a *cycle*. Therefore we will systematically contract cycles (or 'rings') found in the contracted logical topology, which explains the name of the algorithm ("Survivable Mapping Algorithm by Ring Trimming"). G^C_{sub} in the form of a cycle requires the mapping M_B (Step 2) to be *edge-disjoint*. (Otherwise, if the same physical link e^ϕ is used by two or more logical links in G^C_{sub}, a failure of e^ϕ will bring these links down, disconnecting the cycle G^C_{sub}.) Since finding it is equivalent to the NP-complete edge-disjoint paths problem [15], we applied a simple heuristic, as follows. Let each physical edge have a weight (these weights will be used only by this heuristic), which is initially set to one. At each iteration, map the logical links from G^C_{sub} with the shortest path. If no physical link is used more than once, the disjoint solution is found. Otherwise, the weight of each physical link used more than once is increased, and a new iteration starts. After several unsuccessful iterations the heuristic fails.

The implementation of a *node-survivable* version of SMART is based on a node-disjoint mapping of a cycle, instead of an edge-disjoint mapping.

V. SMART APPLICATIONS

We can apply the SMART algorithm in a number of ways. The general scheme can be found in Fig. 3. The option we choose depends on the nature of the results we want to obtain. Specifically we can distinguish:

- the formal verification of the existence of a link/node-survivable mapping,
- a tool tracing and repairing the vulnerable areas of the network,
- a fast heuristic.

We discuss each of these applications separately, in the following sections.

A. Formal verification of the existence of a survivable mapping (ES-rem and SepPath)

Run SMART to map a logical topology G^L on the physical topology G^ϕ. If SMART converges, the link/node-survivable mapping exists and is returned. If SMART does not converge, it returns a mapping M_A and the remaining contracted logical topology $G^L{\downarrow}A$. Because M_A is piecewise link/node-survivable (see Theorem 3), Theorem 2 reduces the task of verifying the existence of a link/node-survivable mapping for the entire G^L, to the same verification for $G^L{\downarrow}A$. This property is a key feature of SMART: if there is a link/node-survivable mapping of G^L on G^ϕ, then SMART *will never miss it*, because the set of the remaining logical links $E^L \backslash A$ can be still mapped in a way preserving the link/node-survivability of $G^L{\downarrow}A$ (and hence of G^L).

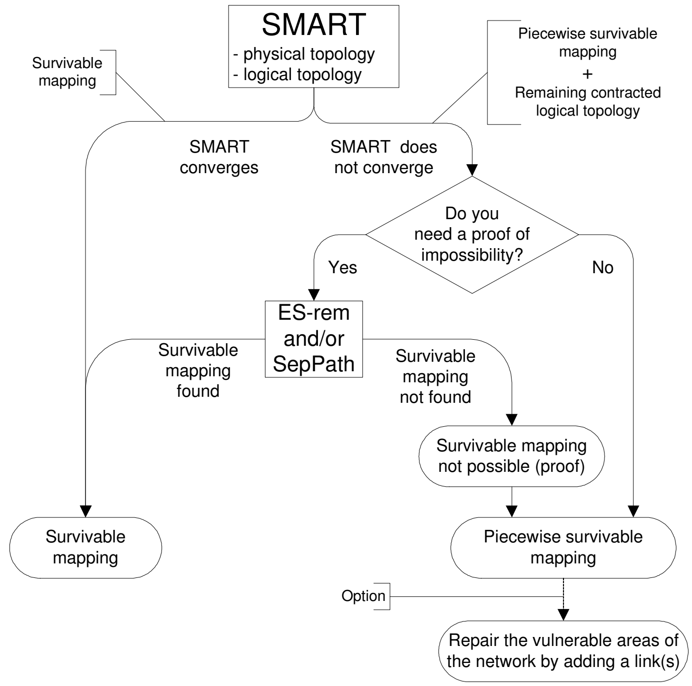


Fig. 3. Applications of the SMART algorithm. This scheme is valid for both link- and node-survivability.

We use two methods to verify the existence of a link/node-survivable mapping for $G^L{\downarrow}A$:

1) Exhaustive Search (ES-rem) uses exhaustive search to find a link/node-survivable mapping of the contracted logical topology $G^L{\downarrow}A$.

2) Separated Path check (SepPath) is defined as follows. If the contracted logical topology $G^L{\downarrow}A$ contains a path p^C such that all nodes on p^C, but the first and the last ones, are of degree two, then clearly all the logical links in p^C must be mapped edge-disjointly to enable link-survivability. Therefore the failure of an exhaustive search for an edge-disjoint mapping of p^C will prove impossibility. A simple modification adapts SepPath to the node-survivability case.

Compared to an unrestricted exhaustive search, the exhaustive search respecting the edge-disjointness constraint is relatively easy (though still NP-complete). For this reason SepPath is better suited to larger topologies than ES-rem.

B. A tool tracing and repairing the vulnerable areas of the network

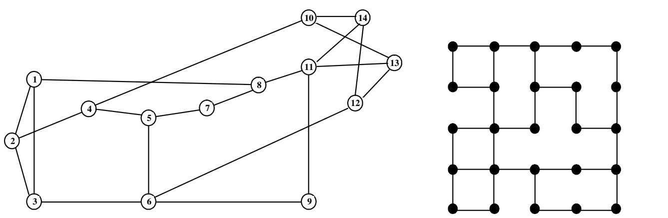
We have developed two methods to verify the existence of a link/node-survivable mapping: ES-rem and SepPath. Once we know that a particular pair of physical and logical topologies cannot be mapped in a link/node-survivable way, a natural question is to modify the topologies to enable such a mapping. Where should a new link be added? The SMART algorithm helps us in answering this question. Run SMART and wait until it terminates. The remaining contracted logical topology $G^L{\downarrow}A$ and the piecewise-survivable mapping M_A are returned. Choose at random two nodes u^C, v^C in $G^L{\downarrow}A$ and pick any two nodes u, v in G^L, such that $u \in Origin(u^C)$ and $v \in Origin(v^C)$. Now connect u and v with an additional logical/physical link (remember that we assume identical vertices at both layers). If this link already exists, repeat the procedure. The simulation results in Section VI-D discuss the efficiency of this approach.

C. A fast heuristic

The application of SMART as a heuristic was studied in [12]. With SMART a link/node-survivable mapping is found orders of magnitude more rapidly and usually more often than with other heuristics proposed in the literature to date.

VI. Simulation results

A. Physical and logical topologies



a) NSFNET b) f-lattice (2-node-connected)

Fig. 4. Physical topologies used in simulations. (a) NSFNET; (b) f–lattice constructed from full square lattice by deleting fraction f of links, while preserving 2–node–connectivity (here $f \simeq 0.25$).

In the simulations we use various topologies. A relatively small physical topology is NSFNET (14 vertices, 21 edges) presented in Fig. 4a. To imitate larger, real-life physical topologies we also generate square lattices in which a fraction f of edges is deleted, as shown in Fig. 4b; we call them f-lattices. The parameter f is often fixed to $f = 0.3$, which resulted in an f-lattice with an average vertex degree slightly smaller than that of NSFNET. Since the IP graph is less regular (for instance, there is no reason why it should be planar), the logical topologies are 2-node-connected random graphs of various average vertex degree. (Clearly, 2-node-connectivity of both physical and logical topologies is a necessary condition for the existence of a node-survivable mapping.)

B. ES-rem and SepPath efficiency, and 'unknown area'

In Section V-A we defined two methods of verification of the existence of a survivable mapping, ES-rem and SepPath. In this section we examine the benefits of these approaches.
The physical topology is an f-lattice with the parameter $f = 0.3$. The logical topology is a random graph with average vertex degree $\langle k^L \rangle = 4$. For each number of nodes N, we generate a number of physical/logical topology pairs, and keep the first 1000 for which SMART does not converge. In Fig. 5a, we present the cumulative distribution function (CDF) of the number of logical links in the remaining contracted logical topology $G^L\!\downarrow\! A$ returned by the algorithm. We can see that, if SMART does not converge, the size of $G^L\!\downarrow\! A$ is usually relatively small. For instance, for $N = 36$, SMART leaves six or fewer logical links out of the total number of 72, in about 80% of cases. Moreover, this property seems to depend only slightly on the topology size.
The distribution of run-times of ES-rem is plotted in Fig. 5b. For $N = 16$, about 90% of topologies need less than 0.001 sec to run ES-rem.[3] Only very few need more than 0.1 sec. For comparison purposes we also ran a full exhaustive search without prior contraction by SMART for $N = 16$. We observe the difference in run-times of at least 7 orders of magnitude. Most of the runs of the full exhaustive search last more than 10000 seconds ($\sim$ 3 hours), the maximal time allowed in the simulations. This limits the application of the full exhaustive search to the topologies of size of several nodes.
Fig. 5b also exemplifies the tradeoff we faced in simulations. On one hand, the ES-rem runs quickly for the majority of the topologies, but on the other hand, the remaining few topologies will take orders of magnitude more time. We observed the same phenomenon when applying the SepPath verification method. Therefore we have decided to use a strict, *one minute stopping time*. If neither ES-rem nor SepPath finishes the computation within 60 seconds, the question of the existence of a survivable mapping is left unanswered. As the result, the figures in the following sections display two curves: the lower one is the percentage of survivable mappings found within 1 minute, the upper one is the percentage of logical topologies proved to be unmappable in a survivable way within 1 minute. The curves are separated by an 'unknown area' set in gray.
The results in Fig. 5 were generated for link-survivability, however, in the case of the node-survivability we obtained very similar results.

C. Survivability of random graphs on various physical topologies

It is interesting to see what fraction of randomly chosen topologies can/cannot be mapped in a link- and node-survivable way. To the best of our knowledge, it is the first time these results can be obtained in a reasonable time for moderate and large topologies.
For a particular pair of physical and logical topologies, we first apply the SMART algorithm. If SMART does not converge, we try ES-rem and SepPath to verify the existence of a survivable solution. Their run-times are restricted to the 'one minute bound', as explained in Section VI-B.
In Fig. 6a we present the results of the mapping of random graph logical topologies on NSFNET. We vary the average vertex degree $\langle k^L \rangle$ of the logical graph; for each $\langle k^L \rangle$ we generate 1000 topologies. As expected, link-survivability is far easier to obtain than node-survivability. Note also that the results strongly depend on $\langle k^L \rangle$.
In order to examine a larger spectrum of physical topologies and topology sizes, in Fig. 6b,c we map a random graph logical topology on the f-lattice physical topology. This time we fix the average vertex degree of the logical topology $\langle k^L \rangle = 4$ and we vary the parameter f of the physical topology (Fig. 6b)

[3]We implemented the SMART algorithm in C++ and ran it on a Pentium 4 machine.

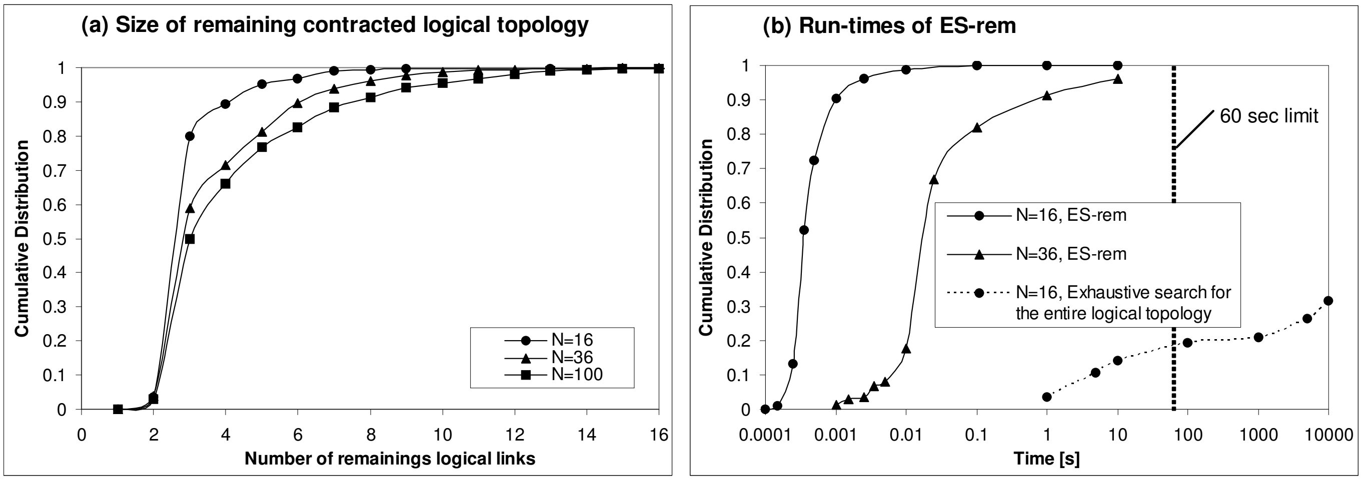


Fig. 5. (a) CDF of the number of logical links in the remaining contracted logical topology. $f = 0.3$, $N = 16 \ldots 100$; (b) CDF of ES-rem and full exhaustive search times. $f = 0.3$, $N = 16 \ldots 100$.

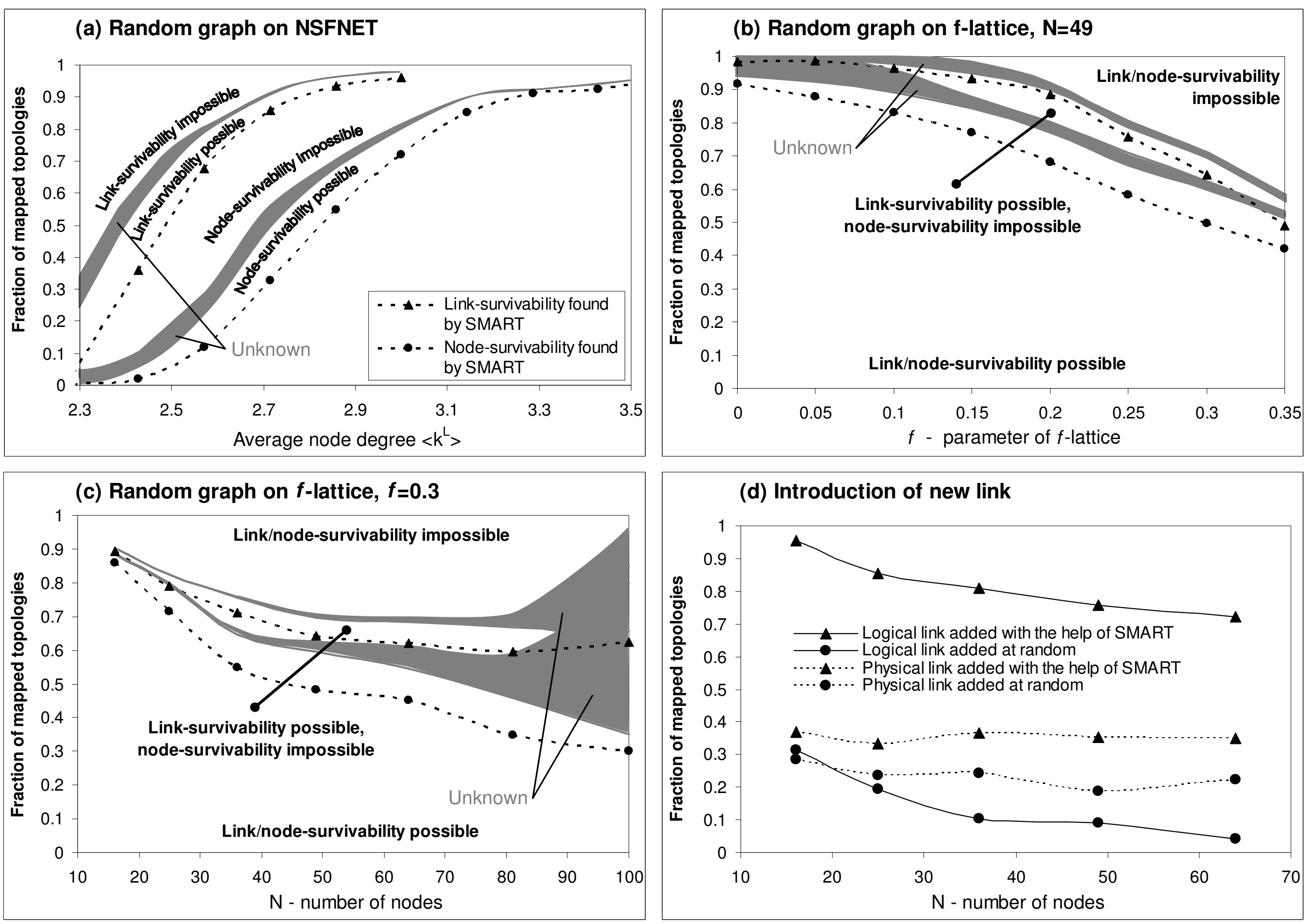


Fig. 6. Link- and node-survivability under various scenarios. Parameters: N–number of nodes ($N = |V|$); f–parameter of the f-lattice physical topology; $\langle k^L \rangle$–average node degree of the logical topology. (a) Random graph logical topologies mapped on NSFNET. $N = 14$, $\langle k^L \rangle = 2.3 \ldots 3.5$; (b) Random graph logical topologies mapped on f-lattices. $N = 49$, $f = 0 \ldots 0.35$, $\langle k^L \rangle = 4$; (c) Random graph logical topologies mapped on f-lattices. $N = 16 \ldots 100$, $f = 0.3$, $\langle k^L \rangle = 4$; (d) Enabling link-survivability by introducing an additional link. Random graph logical topologies mapped on f-lattices. $N = 16 \ldots 100$, $f = 0.3$, $\langle k^L \rangle = 4$.

or the number of nodes N (Fig. 6c). We generated 1000 topologies for each parameter. Fig. 6b shows that the fraction of topologies mappable in a survivable way decreases with growing f. This was expected, since it is more difficult to map the logical topology on a sparser physical graph. In Fig. 6c, the 'unknown area' quickly widens for $N > 80$ because of the 'one minute bound'.

The dashed curves in Fig. 6a,b,c show the fraction of topologies mapped in a link-survivable (triangles) or node-survivable (circles) way by SMART alone, without being followed by any exhaustive search approach. The distances between these curves and the mapping-impossible areas are relatively small, which confirms the high efficiency of SMART as a heuristic.

D. Introduction of an additional link

Another property of the SMART algorithm is the ability to trace and repair the vulnerable areas of the network. In particular, in Section V-B we described a way to introduce an additional logical or physical link to enable a survivable mapping. In this section we verify the efficiency of that approach.

We map random graph logical topologies on f-lattices and vary N. For each N, we generate 1000 pairs of physical and logical topologies, such that for each pair separately, a link-survivable mapping does not exist. For each topology pair, we add one logical or physical link with the help of SMART, as described in Section V-B. Next, the existence of a link-survivable mapping is verified again, for this extended pair of topologies. For comparison purposes we also simulate a completely random placement of an additional link.

The results are shown in Fig. 6d. For better readability, we do not include the 'unknown area', which lie above each curve. The application of SMART enables a very efficient placement of an additional *logical* link, which helps in 70% to 95% of cases (depending on N). In contrast, the completely random placement helps far less, and only for small topologies - for larger N its efficiency becomes insignificant. This is because only new logical links connecting *different* nodes in $G^L{\downarrow}A$ (i.e., different link-survivable pieces in G^L) may help; the larger the topology, the lower the probability of achieving it with a completely random placement. The efficiency of the placement of a new *physical* link has a more random nature. Again, the SMART approach helps, however, its impact is not as significant nor dependent on N, as in the case of logical links. This is because the introduction of a new physical link *within the same* link-survivable piece may also help.

The results concerning the *node-survivability* are very similar to those presented in Fig. 6d.

VII. Conclusion and future work

In this paper we defined a *piecewise survivable mapping* which preserves the survivability of some subgraphs of the logical topology. The formal analysis of the piecewise survivable mapping enabled us to specify the necessary and sufficient conditions for the existence of a link/node-survivable mapping. This substantially simplifies the verification of the existence of a survivable mapping. A second application of a piecewise survivable mapping is tracing vulnerable areas in the network and pointing where new link(s) should be added to enable a survivable mapping. Finally, we showed that the SMART algorithm is not only an efficient and scalable algorithm that searches for a survivable mapping, it also exhibits a number of provable properties that are crucial for the applications we consider in the paper. Therefore the combination of the SMART algorithm and the formal analysis of the survivability problem gives us a powerful tool to designing, diagnosing and upgrading the topologies in IP over WDM networks. We have tested these applications in simulations, for a large spectrum of physical and logical topologies.

In our future work we will address the capacity-constrained version of the problem. We also plan to consider the case of multiple failures.

VIII. Acknowledgements

The work presented in this paper was financially supported by grant DICS 1830 of the Hasler Foundation, Bern, Switzerland.

References

[1] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: a new approach to the design of WDM-based networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 800–809, May 2002.

[2] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee, "Fault management in IP-Over-WDM Networks: WDM Protection vs. IP Restoration," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, January 2002.

[3] A. Fumagalli and L. Valcarenghi, "IP Restoration vs. WDM Protection: Is There an Optimal Choice?," *IEEE Network*, Nov/Dec 2000.

[4] G. Iannaccone, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "Feasibility of IP restoration in a tier-1 backbone," *Sprint ATL Research Report Nr. RR03-ATL-030666*.

[5] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C-N. Chuah, and C. Diot, "Characterization of Failures in an IP Backbone," *Proc. of IEEE INFOCOM'04*, 2004.

[6] L-W. Chen and E. Modiano, "Efficient Routing and Wavelength Assignment for Reconfigurable WDM Networks with Wavelength Converters," *Proc. of IEEE INFOCOM 2003*, 2003.

[7] H. Lee, H. Choi, S. Subramaniam, and H.-A. Choi, "Survival Embedding of Logical Topology in WDM Ring Networks," *Information Sciences : An International Journal, Special Issue on Photonics, Networking and Computing*, 2002.

[8] A. Sen, B. Hao, B.H. Shen, and G.H. Lin, "Survivable routing in WDM networks logical ring in arbitrary physical topology," *Proceedings of the IEEE International Communication Conference ICC02*, 2002.

[9] F. Giroire, A. Nucci, N. Taft, and C. Diot, "Increasing the Robustness of IP Backbones in the Absence of Optical Level Protection," *Proc. of IEEE INFOCOM 2003*, 2003.

[10] J. Armitage, O. Crochat, and J. Y. Le Boudec, "Design of a Survivable WDM Photonic Network," *Proceedings of IEEE INFOCOM 97*, April 1997.

[11] A. Nucci et al., "Design of Fault-Tolerant Logical Topologies in Wavelength-Routed Optical IP Networks," *Proc. of IEEE Globecom 2001*, 2001.

[12] M. Kurant and P. Thiran, "Survivable Mapping Algorithm by Ring Trimming (SMART) for large IP-over-WDM networks," *Proc. of BroadNets 2004*, 2004.

[13] S. il Kim and S. Lumetta, "Addressing node failures in all-optical networks," *Journal of Optical Networking*, vol. 1, no. 4, pp. 154–163, April 2002.

[14] J. Gross and J. Yellen, *Graph Theory and its Applications*, CRC Press, 1999.

[15] A. Frank, *Packing paths, circuits and cuts - a survey (in Paths, Flows and VLSI-Layout)*, Springer, Berlin, 1990.

IX. Appendix

In this section we prove Theorems 1, 2 and 3. For this purpose we use the following definition of link- and node-survivability, equivalent to Definition 5.

Definition 7 (Link- and node-survivability): Let $G^L = (V, E^L)$, $A \subset E^L$ and $G^C = (V^C, E^C) = G^L \downarrow A$. Take any connected subgraph $G^C_{sub} = (V^C_{sub}, B)$, $B \subseteq E^C$, of the contracted topology G^C, and let M_B be a mapping of the set B of logical links. The pair $\left[G^C_{sub}, M_B\right]$ is *link-survivable* if for any physical link e^ϕ and for any two vertices $u, v \in V^C_{sub}$, there exists a path $p^L_{u,v} \subset G^C_{sub}$ between vertices u and v such that $e^\phi \notin M_B(p^L_{u,v})$. The pair $\left[G^C_{sub}, M_B\right]$ is *node-survivable* if for any node $v^* \in V$ and for any two vertices $u^C, v^C \in V^C_{sub}$ such that $Origin(u^C) \neq v^*$ and $Origin(v^C) \neq v^*$, there exists a path $p^L_{u^C,v^C} \subset G^C_{sub}$ between vertices u^C and v^C such that $v^* \notin M_B(p^L_{u^C,v^C})$.

(We keep L in $p^L_{u,v}$ in order to stress that $p^L_{u,v}$ consists of logical links.)

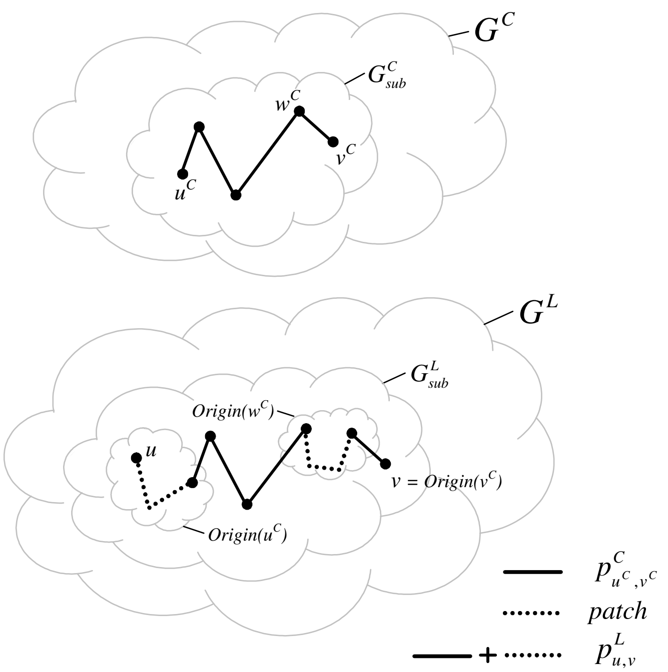


Fig. 7. Illustration of proof of Theorem 1. A first portion of the path $p^L_{u,v}$ is the path $p^C_{u^C,v^C}$ found in G^C_{sub}. Next it is completed, where necessary, with the patches found in origins of the nodes of $p^C_{u^C,v^C}$.

Proof of Theorem 1: (Please refer to Fig. 7.)

A. Link-survivability case:

First note that since $G^C = G^L \downarrow A$, no logical edge from the set A can be found in G^C, which implies that $A \cap B = \emptyset$. Therefore the operation $M_A \cup M_B$ is always well defined, as in (2) and (3).

Let $M_{A\cup B} = M_A \cup M_B$ and $G^L_{sub} = Origin(G^C_{sub})$. We have to prove that the pair $\left[G^L_{sub}, M_{A\cup B}\right]$ is link-survivable. Take any single physical link e^ϕ and two vertices $u, v \in G^L_{sub}$. According to Definition 7 we have to show that there exists a path $p^L_{u,v}$ in G^L_{sub} such that $e^\phi \notin M_{A\cup B}(p^L_{u,v})$. The path $p^L_{u,v}$ is constructed in two steps, (i) and (ii).

(i) A first portion of $p^L_{u,v}$ is found in the contracted graph G^C (recall that G^C consists of logical edges), as follows. Call $u^C, v^C \in V^C_{sub}$ the vertices in $G^C_{sub} = (V^C_{sub}, B)$ whose origins contain u and v, respectively, i.e., such that $u \in Origin(u^C)$ and $v \in Origin(v^C)$. Find a path $p^C_{u^C,v^C}$ in G^C_{sub}, such that $e^\phi \notin M_B(p^C_{u^C,v^C})$. This is always possible since the pair $\left[G^C_{sub}, M_B\right]$ is link-survivable. We take $p^C_{u^C,v^C}$ as the first portion of $p^L_{u,v}$.

(ii) We now turn our attention to the origins of vertices of $p^C_{u^C,v^C}$. Take any two consecutive edges a^L and b^L of $p^C_{u^C,v^C}$, and let w^C be their common end–node in G^C_{sub}. If $Origin(w^C)$ is not a single node in G^L_{sub}, then a^L and b^L might not have a common end–node in G^L_{sub}. However, by piecewise link-survivability of $\left[G^L, M_A\right]$, the pair $\left[Origin(w^C), M_A\right]$ is link-survivable. Therefore, if we denote respectively by $v_a, v_b \in Origin(w^C)$ the end–nodes of a^L and b^L, that belong to $Origin(w^C)$, we can find a logical path $p^L_{v_a,v_b}$ in $Origin(w^C)$ connecting v_a and v_b, such that $e^\phi \notin M_A(p^L_{v_a,v_b})$. We call this path a patch of w^C and denote it by $patch(w^C)$. If for a given w^C, the edges a^L and b^L have a common end–node v^L in G^L_{sub} then $patch(w^C) = v^L$.

For every vertex $w^C \in p^C_{u^C,v^C}$, find $patch(w^C)$. If $w^C = u^C$ then $patch(u^C)$ will connect the logical vertex u with an end–node of the first logical edge in $p^C_{u^C,v^C}$, instead of connecting two end–nodes. The same holds for $w^C = v^C$.

To summarize, in step (i) we have found the path $p^C_{u^C,v^C}$ in the contracted subgraph G^C_{sub}. Next, in step (ii), we have constructed a set of *patches* for each vertex of this path. Now we combine steps (i) and (ii) to obtain the full path $p^L_{u,v}$:

$$p^L_{u,v} = \text{edges}(p^C_{u^C,v^C}) \cup \Big\{ \bigcup_{w^C \in p^C_{u^C,v^C}} patch(w^C) \Big\}. \qquad (5)$$

(Note that the vertices of $p^C_{u^C,v^C}$ belong to G^C, but not necessarily to G^L. Therefore, to avoid confusion we took "edges($p^C_{u^C,v^C}$)"; appropriate vertices from G^L will be provided by the *patch*es.)

The logical path $p^L_{u,v}$ connects the vertices u and v and has been constructed in such a way, that for every $w^C \in p^C_{u^C,v^C}$:

$$e^\phi \notin M_B(p^C_{u^C,v^C}) \qquad (6)$$

$$e^\phi \notin M_A(patch(w^C)). \qquad (7)$$

Since $M_A \cup M_B = M_{A\cup B}$ and $A \cap B = \emptyset$, we can rewrite (6) and (7) as follows

$$e^\phi \notin M_{A\cup B}(p^C_{u^C,v^C}) \qquad (8)$$

$$e^\phi \notin M_{A\cup B}(patch(w^C)). \qquad (9)$$

Combining (5), (8) and (9) yields finally that

$$e^\phi \notin M_{A\cup B}(p^C_{u,v}). \qquad (10)$$

B. Node-survivability case:

To consider node failures, we replace in the above proof (i) "link-" with "node-" and (ii) "(edge) e^ϕ" with "(vertex) v^*". Additionally require, when taking u and v in G^L_{sub}, that $u, v \neq v^*$. (The latter is the consequence of the condition "$Origin(u^C) \neq v^*$ and $Origin(v^C) \neq v^*$" of Definition 7.)

With these modifications we obtain the proof of Theorem 2 for the node-survivability case.

■

Proof of Theorem 2:

A. Link-survivability case:

$\Leftarrow$ We know that the pair $\left[G^L, M_A\right]$ is piecewise link-survivable. Suppose that there exists a mapping $M^{surv}_{E^L\setminus A}$, such that the pair $\left[G^L\!\downarrow A, M^{surv}_{E^L\setminus A}\right]$ is link-survivable. Then, by Theorem 1, the pair $\left[Origin(G^L\!\downarrow A),\ M_A\cup M^{surv}_{E^L\setminus A}\right]$ = $\left[G^L,\ M_A\cup M^{surv}_{E^L\setminus A}\right]$ is also link-survivable. So the mapping $M^{surv}_{E^L} = M_A \cup M^{surv}_{E^L\setminus A}$ is a link-survivable mapping of G^L on G^ϕ.

$\Rightarrow$ Assume that a link-survivable mapping of G^L on G^ϕ exists, call it $M^{surv}_{E^L}$. Let $M^{surv}_{E^L\setminus A}$ coincide with $M^{surv}_{E^L}$ for all links in $E^L\setminus A$:

$$M^{surv}_{E^L\setminus A}(E^L\setminus A) = M^{surv}_{E^L}(E^L\setminus A). \tag{11}$$

We now show that the pair $\left[G^L\!\downarrow A, M^{surv}_{E^L\setminus A}\right]$ is link-survivable. Take any physical link e^ϕ and any two vertices u^C and v^C in $G^L\!\downarrow A$. According to Definition 7, we have to show that there exists a path $p^L_{u^C,v^C} \subset G^L\!\downarrow A$ between vertices u^C and v^C such that $e^\phi \notin M^{surv}_{E^L\setminus A}(p^L_{u^C,v^C})$.

Take any two vertices u, v in the logical topology, such that $u \in Origin(u^C)$ and $v \in Origin(v^C)$. Since the pair $\left[G^L, M^{surv}_{E^L}\right]$ is link-survivable, there exists a path $p^L_{u,v} \subset G^L$ between vertices u and v, such that

$$e^\phi \notin M^{surv}_{E^L}(p^L_{u,v}). \tag{12}$$

Construct $p^L_{u^C,v^C}$ by contracting in $p^L_{u,v}$ the logical edges that belong to the set A:

$$p^L_{u^C,v^C} = p^L_{u,v} \downarrow A. \tag{13}$$

(Call nodes in $p^L_{u^C,v^C}$ after the nodes of $G^L\!\downarrow A$).

Since $p^L_{u,v}$ is a path in G^L, and since the contraction an edge merges its two end-nodes and thus preserves its continuity, $p^L_{u^C,v^C}$ is a path in $G^L\!\downarrow A$. Moreover, since $u \in Origin(u^C)$ and $v \in Origin(v^C)$, the path $p^L_{u^C,v^C}$ connects u^C and v^C. Relations (11), (12) and (13) yield that $e^\phi \notin M^{surv}_{E^L\setminus A}(p^L_{u^C,v^C})$, which proves the claim.

B. Node-survivability case:

Apply the same two changes (i)-(ii) as in the proof of Theorem 1-B, and additionally require, when taking u^C and v^C in $G^L\!\downarrow A$, that $Origin(u^C) \neq v^*$ and $Origin(v^C) \neq v^*$ (as in Definition 7).

■

Proof of Theorem 3: [By induction]

A. Link-survivability case:

INITIALIZATION:

Initially $G^C = G^L$. Therefore the origin of any vertex $v^C \in V^C$ is a single node in G^L, and it cannot be disconnected. Hence for every $v^C \in V^C$, the pair $\left[Origin(v^C), M_A\right]$ is link-survivable and consequently the pair $\left[G^L, M_A\right]$ is piecewise link-survivable.

INDUCTION:

Assume that after some iteration the pair $\left[G^L, M_A\right]$ is piecewise link-survivable. We have to prove that after the next iteration of the algorithm, the updated mapping $\widehat{M}_A$ will still form a piecewise link-survivable pair $\left[G^L, \widehat{M}_A\right]$.

One iteration of the SMART algorithm consists of Steps 2, 3 and 4, which we recall here:

2. Find $G^C_{sub} = (V^C_{sub}, B)$ and M_B, such that the pair $\left[G^C_{sub}, M_B\right]$ is link-survivable.
3. $\widehat{M}_A := M_A \cup M_B$
4. $\widehat{G}^C := G^C \downarrow B$

(For clarity we indicated the updated M_A and G^C by a hat: '$\widehat{\ }$')

The updated contracted topology $\widehat{G}^C = (\widehat{V}^C, \widehat{E}^C)$ was created from G^C by replacing $G^C_{sub} = (V^C_{sub}, B)$ by a single node, which we call $\widehat{v}^C_{sub}$; the remaining nodes stayed unchanged. So $\widehat{V}^C = \{\widehat{v}^C_{sub}\} \cup V^C\setminus V^C_{sub}$. Take any $\widehat{v}^C \in \widehat{V}^C$; we have two possibilities:

(i) $\widehat{v}^C = \widehat{v}^C_{sub}$: Since $G^C_{sub} = (V^C_{sub}, B)$ was contracted into $\widehat{v}^C_{sub}$, their origins coincide: $Origin(G^C_{sub}) = Origin(\widehat{v}^C_{sub})$. Since $\widehat{M}_A = M_A \cup M_B$, the pair $\left[Origin(\widehat{v}^C_{sub}), \widehat{M}_A\right] = \left[Origin(G^C_{sub}), M_A \cup M_B\right]$ is link-survivable because of Theorem 1.

(ii) $\widehat{v}^C \neq \widehat{v}^C_{sub}$: In this case $\widehat{v}^C \in V^C \setminus V^C_{sub}$, so $\widehat{v}^C = v^C$. By piecewise link-survivability of the pair $\left[G^L, M_A\right]$, the pair $\left[Origin(v^C = \widehat{v}^C), M_A\right]$ is link-survivable. Since $\widehat{M}_A = M_A \cup M_B$, the pair $\left[Origin(\widehat{v}^C), \widehat{M}_A\right]$ is link-survivable as well.

Combining (i) and (ii), we have proven that for every $\widehat{v}^C \in \widehat{V}^C$, the pair $\left[Origin(\widehat{v}^C), \widehat{M}_A\right]$ is link-survivable. So, by Definition 6, the pair $\left[G^L, \widehat{M}_A\right]$ is piecewise link-survivable.

B. Node-survivability case:

In the proof above, replace "link-" with "node-".

■