Dynamic Survivable Routing for Shared Segment Protection

János Tapolcai and Pin-Han Ho

Abstract: **This paper provides a thorough study on shared segment protection (SSP) for mesh communication networks in the complete routing information scenario, where the integer linear program (ILP) in [1] is extended such that the following two constraints are well addressed: (a) The restoration time constraint for each connection request, and (b) the switching/merging capacity constraint at each node. A novel approach, called SSP algorithm, is developed to reduce the extremely high computation complexity in solving the ILP formulation. Basically, our approach is to derive a good approximation on the parameters in the ILP by referring to the result of solving the corresponding shared path protection (SPP) problem. Thus, the design space can be significantly reduced by eliminating some edges in the graphs. We will show in the simulation that with our approach, the optimality can be achieved in most of the cases. To verify the proposed formulation and investigate the performance impairment in terms of average cost and success rate by the additional two constraints, extensive simulation work has been conducted on three network topologies, in which SPP and shared link protection (SLP) are implemented for comparison. We will demonstrate that the proposed SSP algorithm can effectively and efficiently solve the survivable routing problem with constraints on restoration time and switching/merging capability of each node. The comparison among the three protection types further verifies that SSP can yield significant advantages over SPP and SLP without taking much computation time.**

Index Terms: **Complete routing information scenario, integer linear program (ILP), shared risk group (SRG), shared segment protection (SSP), working and protection (W-P) paths.**

I. INTRODUCTION

Shared segment protection (SSP) is expected to yield merits in achieving better efficiency, flexibility, and reduction of restoration time in mesh communication networks compared with all the other counterparts [1]–[9]. With SSP, a connection is provisioned by concatenating a series of self-healing unit (or called *protection domains* in the following context), each of which contains a working and protection (W-P) segment-pair for performing local restoration when any working segment is subject to any unexpected interruption. A simple example for the operation of SSP is shown in Fig. 1. When the working path segment of protection domain 2 is impaired unexpectedly (e.g., either link $d-e$, $e-f$, $f-g$, or $g-h$ is cut), the restoration is performed locally within protection domain 2 such that the affected flow switches over to the backup segment at node d (called switching node of the protection domain) and merges back to the original working path at node i (called merging node of the protection domain). Please refer to [1] and [3] for an overview and a comprehensive survey on SSP.

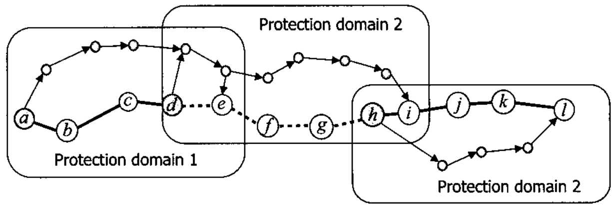


Fig. 1. An illustration of SSP. The working path is $a-b-c-d-e-f-g-h-i-j-k-l$, which is divided into three segments: $a-b-c-d$, $d-e-f-g-h$, and $h-i-j-k-l$.

In our previous work, an integer linear program (ILP) was provided to solve dynamic survivable routing for SSP [1]. Although the formulation can effectively provide optimal solution according to a specific connection request and network-state, the lengthy computation time in solving the ILP nonetheless makes the scheme not suitable for any practical implementation. In addition, the formulation left space to improve in the following two ways: (a) The size of each protection domain is not constrained so that the restoration time for each connection is not considered; and (b) every node can switch and/or merge restoration traffic at the same time, which may not be the case for practical applications. For the former, we claim that an approach for constraining the restoration time must be developed such that the class of service provisioning of survivable bandwidth is possible. The major concern for the latter is that the nodes serving as switching/merging devices need to provide extra signalling efforts and hardware responsiveness, which may not be general to all the network nodes.

In this paper, the ILP in [1] is extended such that the constraint on restoration time in each protection domain of a connection as well as the constraint on the capability of each node to switch/merge restoration traffic can be well addressed.

To improve the computation efficiency, a novel approach, called SSP algorithm, is proposed to reduce the runtime in solving the proposed ILP. Basically, our approach is to derive a good approximation on the parameters in the ILP by referring to the result of solving the corresponding ILP for shared path protection (SPP), with which a significant reduction on the design space can be achieved by eliminating some edges in the graphs. We will verify in the simulation that the optimality of the derived solution can be achieved in most of the cases with our approach.

With the proposed SSP algorithm, experiments are developed to evaluate the performance gain/impairment by

Manuscript received January 2, 2004; approved for publication by Hyeong Ho Lee, Division III Editor, August 22, 2005.

J. Tapolcai is with the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, 2nd Magyar Tudósok Krt., Budapest, H-1117, email: tapolcai@tmit.bme.hu.

P.-H. Ho is with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, N2L3G1, email: pinhan@hph3.uwaterloo.ca.

1229-2370/07/$10.00 © 2007 KICS

adding/removing the switching and merging capability from a specific node in the ERNET and NETNET shown in Figs. 6 and 7, respectively, in which three performance metrics: Average increase of blocking probability, maximum increase of blocking probability, and average routing cost, are focused. The results show the exact location of the nodes how the switching and merging capabilities are relatively crucial in terms of the three performance metrics, by which the deployment of network switching/signaling capacity and the development of heuristic routing algorithms can be greatly benefited. In addition, the simulation compares SSP, shared link protection (SLP), and SPP in terms of average cost and success rate under different restoration time constraints. We expect a suite of comprehensive results along with abundant physical meanings will be generated upon the related topics.

This paper is organized as follows. In Section II, the problem of SSP along with the associated modeling techniques are identified. In Section III, the ILP formulation for solving the SSP problem with the constraints on restoration time and switching/merging capability of nodes is presented, followed by the introduction of the SSP algorithm. In Section IV, the proposed SSP algorithm is verified, and the results are compared with that of two other types of protection, namely, SLP and SPP. Section V concludes this paper.

II. PROBLEM DEFINITION

A. Network Modeling and Problem Definition

Let the network be denoted as $G(N, E)$, where N and E are the set of nodes and directional links in the network, respectively. Let the source and the destination of the upcoming connection request, W, be denoted as s and d with a bandwidth of $b(W)$. Without loss of generality, every link can accommodate a fixed amount of bandwidth. For language precision, in the following context a "link" is always bi-directional and represents a connection between two nodes. In the graph representing the network, a link is modeled as two directed "arcs" between the same ending nodes in both directions. We assume that a link is physically bundled and is comprised of several independent communication channels that provision data flows. Therefore, a failure defined in this paper is limited to a link cut. We take an assumption of single failure scenario, where no more than a single link in the entire network can be possibly interrupted at a moment.

With $G(N, E)$, each node is classified into the following four sets:

(a) $N_f \subset N$ the nodes able to serve as both switching and merging node;
(b) $N_s \subset N$ the nodes able to serve as a switching but not a merging node;
(c) $N_m \subset N$ the nodes able to serve as a merging but not a switching node;
(d) $N_l \subset N$ the nodes neither able to switch nor able to merge.

Please refer to [2] for the modeling of restoration time. A parameter denoted as $\zeta_{a,b}$ is defined for each link, which represents the propagation delay of the signaling on link a, b. A global parameter $\zeta_{\max}$ denotes a limit of the restoration time for the connection, and could be defined in the service level agreement. The restoration time of the connection is defined as the maximum of the restoration time of all segments.

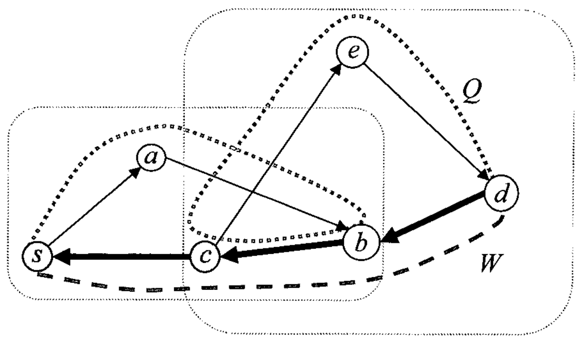


Fig. 2. A simple example showing the design objective and variables in the ILP formulation.

The SSP survivable routing problem is defined as follows. Given a connection request between nodes s and d, a working path W for carrying effective working traffic as well as a *mass protection path* Q composed of all the protection segments link-disjoint from W, are solved in a single step. A simple example is shown in Fig. 2, where Q is $(s - a - b - c - e - d)$. The first protection domain is formed by the W-P segment-pair $(s - c - b)$ and $(s - a - b)$ with the switching and merging node s and b, respectively; while the second is formed by $(c - b - d)$ and $(c - e - d)$ with the switching and merging nodes c and d, respectively. Note that Q is allowed to contain loops and traverse the same links more than once to reflect the fact that spare capacity sharing may happen between two protection segments.

III. INTEGER LINEAR PROGRAMMING FOR SSP

Three graphs are defined to facilitate solving this problem, each carrying one or a few variables for the identifying the W-P segment-pairs. The graph for solving the working segments is denoted as $G_w(N, E_w)$ and is composed of the links which can be taken by the working segment. Obviously the links with $b(W) > f_j$ are not part of E_w, where f_j is the free capacity of link j. The second graph is denoted as $G_p(N, E_p)$ and is to facilitate solving the protection segments. We need this graph to record the *spare link-state* (considered as the link cost defined for solving Q), which is different from the link-state taken for solving the working path due to the resource sharing. Links with insufficient restoration capacity (i.e., $b(W) > f_j + v_j$) are not part of E_p. Let us define the *available restoration capacity* along link j, denoted as m_j, representing the lower bound of the capacity that can be taken by the protection path. Obviously, $m_j \leq g_j + v_j$, where v_j is the total amount of spare capacity along link j.

The third graph $G'_p(N, E'_p)$ is composed of all the links in E_p and the links of E_w in a reversed direction ($\underline{E}_p = E_p \cup reversed(E_w)$). The inclusion of the links of E_w in a reverse direction into E'_p is called *arc-reversal transformation*. With $G'_p(N, E'_p)$, we can identify Q even if it has an overlapped arc with W (e.g., link $b - c$ along Q overlaps with link $c - b$ of W in Fig. 2). Therefore, E'_p contains *forward arcs* denoted as $\overrightarrow{(a, b)}$, which are due to the links of E_p, as well as the *reversed arcs* denoted as $\overleftarrow{(a,b)}$, which are due to the reversal of link (a,b) in E_w. In the latter case, the reversed links direct from node b to node a for $a, b \in E_w$.

All the three graphs (G_w, G_p, and $\underline{G}_p$) are considered and indexed in an array during the implementation. We have to keep track of the indexes of each arc assigned to the same link. Please refer to [1] for more detailed description on the problem modeling and the definition of the link cost for protection segments.

A. An Extended ILP Formulation

This section presents the proposed ILP for SSP. The target function is as follows:

Minimize

$$\sum_{(a,b)\in E_w} b(W)c_{a,b}x_{a,b} + \sum_{(u,v)\in E_p} (b(W)c_{u,v}r_{u,v} + \varepsilon)\, y'_{u,v} \quad (1)$$

where $c_{a,b}$ is the cost per unit of working bandwidth to reserve link (a,b), ε is a small constant that can be set such that

$$\varepsilon \mid E_p \mid < \min_{(a,b)\in E_p} c_{a,b}r_{u,v}$$

is a variable of scaling the cost for the protection path to traverse through link (u,v). Note that $r_{u,v}$ is determined by the feasibility of the spare capacity sharing for the protection path on link (u,v) and plays as a key to handle the dependency of working and spare capacity in the network. Each $x_{a,b}$ and $y'_{u,v}$ is a binary and non-negative integer variable, respectively, which indicates the number of times the W and Q traversing link $(a,b) \in E_w$ and $(u,v) \in \underline{E}_p$.

The reason of setting $y'_{u,v}$ a non-negative integer instead of a binary variable is that path Q, which is composed of the protection segments of all the protection domains and some reverse arcs along W, may traverse through the same link more than once, which leads to the fact that $y'_{u,v}$ may be larger than 1. The concatenation of all links with $x_{a,b}$ = 1 yields W, while the concatenation all the links with $y'_{u,v} \geq 1$ forms Q.

The target function is subject to the following constraints:

$$\sum_{\forall(a,b)\in E_w} x_{a,b} - \sum_{\forall(b,a)\in E_w} x_{b,a} = \begin{cases} 1, & \text{if } a = s, \\ -1, & \text{if } a = d, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

$$\sum_{\forall(a,b)\in E'_p} y'_{a,b} - \sum_{\forall(b,a)\in E'_p} y'_{b,a} = \begin{cases} 1, & \text{if } a = s, \\ -1, & \text{if } a = d, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

for all $a \in N$. (2) and (3) are the flow conservation constraint for the working and mass protection paths, respectively.

It is important to note that $x_{a,b}$ and $y'_{\overrightarrow{a,b}}$ will be exclusive in terms of the physical links they take. However, link (a,b) can be taken by $y'_{\overleftarrow{a,b}}$ in a reversed direction only if $x_{a,b}$ has a value of 1 on it. Besides, each reversed arc can be used only once since the algorithm allows at most two working segments to overlap at a link. The above statements can be formulated into the following two constraints:

Table 1. The indexes for the attributes of node a, where $a \in N$.

	N_f	N_s	N_m	N_l
γ_a	+1	+1	−1	0
δ_a	+1	−1	+1	0

$$k_{\max}x_{a,b} + y'_{\overrightarrow{a,b}} \leq k_{\max} \quad (4)$$

$$x_{a,b} \geq y'_{\overleftarrow{a,b}} \quad (5)$$

for $\forall(a,b) \in E_w, \forall\overrightarrow{(a,b)} \in E'_p$, where $k_{\max}$ is the maximum number of protection domains that can be possibly handled in the problem, and $y'_{\overleftarrow{a,b}}$ is a binary variable indicating the traversal of Q on the reversed arc $\overrightarrow{(a,b)}$ in E'_p. With the above two constraints, Q must be disjoint from W except for those arcs of W being reversed. The constraints in (4) and (5) not only assert the disjointedness of the working and the corresponding protection segment, but also facilitate the indication of the switching/merging nodes for each protection domain along W.

A pair of variables, $lx_{a,b}$ (with a size of $|E_w|$) and $ly'_{a,b}$ (with a size of $|E'_p|$), is assigned to each link along W and Q, respectively, such that the first link from the source has a label of 1; and if a protection domain ends or starts at a node, the labels of the following arcs will be increased by 1. This labeling method is similar to that proposed in [4].

Since each node may not necessarily be able to serve as a switching and/or merging node of a protection domain, the indexes γ_a and δ_a shown in Table 1 are introduced to depict whether or not node a, where $a \in N$, can switch and/or merge the corresponding affected traffic.

The variable $lx_{a,b}$ introduces the following constraints:

$$(2k_{\max} - 1)x_{a,b} \geq lx_{a,b} \geq 0 \quad (6)$$

$$\sum_{\substack{(a,b)\in E_w \\ a\neq d}} lx_{a,b} - \sum_{\substack{(b,a)\in E_w \\ a\neq s}} lx_{a,b} \geq -2k_{\max} + \sum_{\substack{\overrightarrow{(a,b)}\in E'_p \\ a\neq d}} \gamma_a y'_{\overrightarrow{a,b}} + \sum_{\substack{\overrightarrow{(b,a)}\in E'_p \\ a\neq s}} \delta_a y'_{\overrightarrow{a,b}} + \sum_{\substack{(a,b)\in E_w \\ a\neq d}} k_{\max}x_{a,b} + \sum_{\substack{(b,a)\in E_w \\ a\neq s}} k_{\max}x_{a,b} \quad (7)$$

$$\sum_{\substack{(a,b)\in E_w \\ a\neq d}} lx_{a,b} - \sum_{\substack{(b,a)\in E_w \\ a\neq s}} lx_{a,b} \leq \sum_{\substack{\overrightarrow{(a,b)}\in E'_p \\ a\neq d}} \gamma_a y'_{\overrightarrow{a,b}} + \sum_{\substack{\overrightarrow{(b,a)}\in E'_p \\ a\neq s}} \delta_a y'_{\overrightarrow{a,b}} \quad (8)$$

$$\sum_{(s,b)\in E_w} lx_{s,b} = 1 \quad (9)$$

for $\forall(a,b) \in E_w$, $\forall a \in V$, $\forall\, b \in N_f$, and $a \neq s, d$.

The constraint in (6) ensures that $lx_{a,b}$ is upper-bounded by $2k_{\max} - 1$, and is nonzero only if W passes through (a,b). (7) and (8) serve as a special type of flow conservation constraint upon the net change of lx at node a in the network, which is

Table 2. The "flow increase" at different type of vertices.

Description	V_f	V_s	V_m	V_l
W intersects P at a	2	0	0	0
W exits P at a	1	1	forbidden	forbidden
W enters P at a	1	forbidden	1	forbidden
otherwise	0	0	0	0

expressed at the left-hand side of the equations. With different attributes of network nodes, however, the criterion of flow conservation on lx at each node is different. Without loss of generality, the two equations are verified in the following context only in the event that the nodes have a full ability of switching over and merging back the corresponding affected traffic (i.e., node a such that $a \in N_f$).

Four cases are defined for node a when it is taken by W, where $\forall a \in N_f$, $a \neq s, d$, and $\gamma_a = \delta_a = 1$:

(a) Q merges back to W at a;
(b) Q switches out of W at a;
(c) Q merges back and switches out of W at a;
(d) otherwise.

Now, we consider the change of the value of lx from the source node (where $lx_{s,b} = 1$) to the destination (where $lx_{b,d} = 2k_{opt} - 1$). Table 2 summarizes the four possible cases of increasing lx along W (as well as ly that will be seen later) at node a, which are also shown at the right hand side of (7) and (8). In (7) the value of $lx_{a,b}$ of node $a \in N_f$ along W increases by 1 at node a in the cases (a) and (b), and increases by 2 at node a in the case (c), and is unchanged otherwise. The constraint on the net change in terms of $lx_{a,b}$ has a lower bound specified at the right-hand side of the equation, where the term

$$\sum_{\substack{(a,b)\in E_w \\ a\neq d}} k_{\max} x_{a,b} + \sum_{\substack{(a,b)\in E_w \\ a\neq s}} k_{\max} x_{a,b} - 2k_{\max}$$

checks if node a is taken by W. It is clear that the term is 0 if node a is traversed by W, and is $-2k_{\max}$, otherwise. Therefore, even if node a is not taken by W, (7) still holds.

To take a more detailed look at the right-hand side of (7), in the case of (a), the increase of $lx_{a,b}$ is 1 since

$$\sum_{\substack{(\overrightarrow{a,b})\in E'_p \\ a\neq d}} y'_{\overrightarrow{a,b}} = 0 \text{ and } \sum_{\substack{(\overrightarrow{b,a})\in E'_p \\ a\neq s}} y'_{\overrightarrow{b,a}} = 1$$

(since Q merges back to W at node a). In case (b), the increase of $lx_{a,b}$ is still 1 because

$$\sum_{\substack{(\overrightarrow{a,b})\in E'_p \\ a\neq s}} y'_{\overrightarrow{a,b}} = 0 \text{ and } \sum_{\substack{(\overrightarrow{b,a})\in E'_p \\ a\neq d}} y'_{\overrightarrow{b,a}} = 1.$$

In case (c), increase of $lx_{a,b}$ is 2 since

$$\sum_{\substack{(\overrightarrow{a,b})\in E'_p \\ a\neq d}} y'_{\overrightarrow{a,b}} = \sum_{\substack{(\overrightarrow{b,a})\in E'_p \\ a\neq s}} y'_{\overrightarrow{b,a}} = 1.$$

If node a is neither a switching nor a merging node (but it is taken by W), the right-hand-side of (7) becomes 0, in which no change upon $lx_{a,b}$ is required.

Basically, (8) is devised based on the same idea as (7) except that when node a is not taken by W, the right-hand side becomes 0 instead of $-2k_{\max}$. Both (7) and (8) constrain the net change of the value of $lx_{a,b}$ to be either 0, 1, or 2, at any node a taken by W, depending on if path Q switches over and/or merges back to W at node a. (9) sets $lx_{a,b}$ to 1 if node a is the source node.

For $ly'_{a,b}$, we have the following constraints:

$$(2k_{\max} - 1)y'_{a,b} \geq ly'_{a,b} \geq 0, \text{ for } \forall(a,b) \in E'_p, \quad (10)$$

$$\sum_{\substack{(a,b)\in E'_p \\ a\neq d}} ly'_{a,b} - \sum_{\substack{(b,a)\in E'_p \\ a\neq s}} ly'_{b,a} \geq 0, \text{ for } \forall a \in N, \quad (11)$$

$$\sum_{\substack{(a,b)\in E'_p \\ a\neq d}} ly'_{a,b} - \sum_{\substack{(b,a)\in E'_p \\ a\neq s}} ly'_{a,b} \geq -2k_{\max} \sum_{\substack{(a,b)\in E'_p \\ a\neq d}} k_{\max} y'_{a,b} + \sum_{\substack{(b,a)\in E'_p \\ a\neq s}} k_{\max} y'_{a,b} + \sum_{\substack{(a,b)\in E_w \\ a\neq d}} \gamma_a (x_{a,b} - y'_{\overleftarrow{a,b}}) + \sum_{\substack{(b,a)\in E_w \\ a\neq s}} \delta_a (x_{a,b} - y'_{\overleftarrow{a,b}}), \text{ for } \forall a \in V,\ a \neq s,d, \quad (12)$$

$$\sum_{\substack{(a,b)\in E'_p \\ a\neq d}} ly'_{a,b} - \sum_{\substack{(b,a)\in E'_p \\ a\neq s}} ly'_{a,b} \leq \sum_{\substack{(a,b)\in E_w \\ a\neq d}} \gamma_a (x_{a,b} - y'_{\overleftarrow{a,b}}) + \sum_{\substack{(b,a)\in E_w \\ a\neq s}} \delta_a (x_{a,b} - y'_{\overleftarrow{a,b}}), \text{ for } \forall a \in V,\ a \neq s,d, \quad (13)$$

$$\sum_{(s,b)\in E'_p} ly'_{a,b} = 1, \quad \forall b \in N. \quad (14)$$

The constraint in (10) ensures that $ly'_{a,b}$ is non-zero only if Q passes through arc (a,b). The idea behind (12) and (13) is similar to that of (7) and (8). The only difference is that instead of considering the forward arcs of Q (denoted as $y'_{\overrightarrow{a,b}}$), the term $(x_{a,b} - y'_{\overleftarrow{a,b}})$ is used, which is non-zero for link (a,b) along W while not being taken by Q. Therefore, (12) and (13) ensure that the value of $ly_{a,b}$ on Q increases by 1 at node a only if Q merges back to W or Q switches out of W at node a. (8) is to set $ly'_{a,b}$ as 1 if node a is the source node, and that there would be only a single protection link stretching out of the source node. Please refer to Fig. 3 for an example for the variables formulated above. It can be easily observed that the maximum of $lx_{a,b}$ is $2k_{\max} - 1$ and the maximum of $ly'_{a,b}$ is less than $(2k_{\max} - 1)y'_{a,b}$ even if Q have loops.

With lx and ly' link labels, W is divided into segments along which each link is protected by at least one protection segment. This effort introduces $k_{\max}|E_w|$ and $k_{\max}|E_p|$ link-domain incidence binary variables denoted as $x^k_{a,b}$ and $y^k_{a,b}$, which is 1 if link (a,b) is traversed by the working and protection segment of the kth protection domain, respectively. Note that the only variable defined in graph E_p is the variable $y^k_{a,b}$ in the formulation. We can alternatively define the variable $y^k_{a,b}$ upon E'_p instead of having a new graph E_p, in which the formulation turns out to take only two residual graphs. Although it is a way more

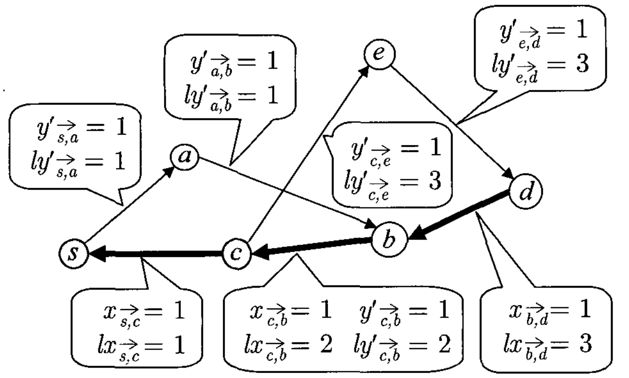


Fig. 3. An example showing variables $x_{a,b}$, $y'_{a,b}$, $lx_{a,b}$, and $ly'_{a,b}$.

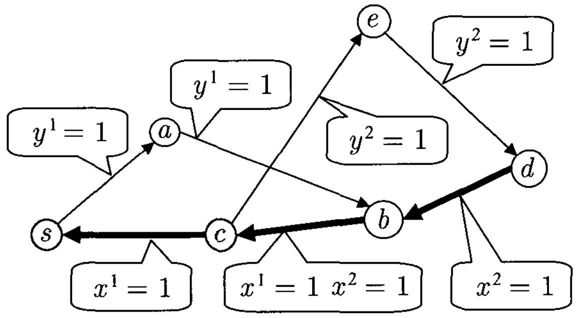


Fig. 4. An example for the values of $x^k_{a,b}$ and $y^k_{a,b}$. For any link without a mark on the figure has a cost zero.

tractable to implement, there would be at most $k_{\max}|E_w|$ variables unnecessarily introduced since $y^k_{a,b}$ on the reversed links of E'_p is redundant. Please refer to Fig. 4 for an example demonstrating the definition of $x^k_{a,b}$ and $y^k_{a,b}$.

In addition to the modification on the constraints upon the variables lx and ly', an extra constraint is required on the routing of path Q for the node set N_s, N_m, and N_l, respectively. At any node in N_s, Q should not merge back to path W, thus, we have:

$$\sum_{\substack{(a,b)\in E_w\\ a\neq d}} y_{\overrightarrow{a,b}} \geq \sum_{\substack{(b,a)\in E_w\\ a\neq s}} y_{\overrightarrow{a,b}}, \text{ for } \forall a \in N_s,\ a \neq s,t. \quad (15)$$

At any node in N_m, Q should not switch out of W. And we have:

$$\sum_{\substack{(a,b)\in E_w\\ a\neq d}} y_{\overrightarrow{a,b}} \leq \sum_{\substack{(b,a)\in E_w\\ a\neq s}} y_{\overrightarrow{a,b}}, \text{ for } \forall a \in N_m,\ a \neq s,t. \quad (16)$$

At any node in N_l, Q should neither merge back nor switch out of path W, thus, we have:

$$\sum_{\substack{(a,b)\in E_w\\ a\neq d}} y_{\overrightarrow{a,b}} = \sum_{\substack{(b,a)\in E_w\\ a\neq s}} y_{\overrightarrow{a,b}}, \text{ for } \forall a \in N_l,\ a \neq s,t. \quad (17)$$

For the variable $y^k_{a,b}$, the following constraints are introduced:

$$\sum_{k=1}^{k_{\max}} (2k-1) y^k_{a,b} = ly'_{\overrightarrow{a,b}} \quad (18)$$

$$\sum_{k=1}^{k_{\max}} y^k_{a,b} = y'_{\overrightarrow{a,b}} \quad (19)$$

$$0 \leq y^k_{a,b} \leq 1 \quad (20)$$

$$\sum_{(a,b)\in E_w} (lx_{a,b} + x_{a,b}) \geq \sum_{(b,a)\in E_p} (2k_{\max}-1) y^k_{b,a} - \sum_{(a,b)\in E_p} (2k_{\max}-1) y^k_{a,b} \quad (21)$$

$$\sum_{(b,a)\in E_w} (lx_{b,a} + x_{b,a}) \geq \sum_{(a,b)\in E_p} (2k_{\max}-1) y^k_{a,b} - \sum_{(b,a)\in E_p} (2k_{\max}-1) y^k_{b,a} \quad (22)$$

for $k = 1, \cdots, k_{\max}$, $\forall (a,b) \in E_p$, and $\forall a \in N$.

(18) can be easily verified by observing Fig. 3, where the value of $ly'_{a,b}$ on Q of the first protection domain is 1; and in the second protection domain $ly'_{a,b}$ is 3; and in the kth protection domain $ly'_{a,b}$ is $2k-1$. (19) ensures that the number of traversals of path Q upon each link is correctly counted. It is clear that (18), (19), and (20) set $y^k_{a,b}$ =1 only when $ly'_{a,b} = 2k-1$. (21) and (22) are flow conservation constraints for $y^k_{a,b}$ since $\sum_{(b,a)\in E_w} lx_{b,a}$ is 0 for all nodes accept for the ones along W. This ensures, that $y^k_{a,b}$ to be a flows starting from a node along W with label $lx = 2k-1-1$ (an incomming arc has the label) and terminate at a node along W with label $lx = 2k-1+1$ (an outgoing arc has the label).

For $x^k_{a,b}$, the following constraints are introduced:

$$\sum_{k=1}^{k_{\max}} (2k-1) x^k_{a,b} = lx_{a,b} + ly'_{\overleftarrow{a,b}} \quad (23)$$

$$\sum_{k=1}^{k_{\max}} x^k_{a,b} = x_{a,b} + y'_{\overleftarrow{a,b}} \quad (24)$$

for $\forall (a,b) \in E_w$.

(23) corresponds to (20) in [1]; however, (23) is much tighter than (20), and the runtime of the CPLEX solver can be significantly reduced since the gap between the relaxed problem and the integer solution is also reduced [10], [11]. (23) can be easily verified by the following argument. On the links of the kth protection domain, we have $lx_{a,b} = 2k-2$ on the non-overlapped links and $lx_{a,b} = 2k-1$ on the overlapped links of the $(k-1)$th and the kth protection domain; we have $lx_{a,b} = 2k$ on the overlapped links of kth and $(k+1)$th protection domain. On the non-overlapped link(s) of the kth protection domain, $ly'_{\overleftarrow{a,b}} = 0$ and (23) holds. For a link serving as the overlapped link of the kth and $(k+1)$th working segments, the left hand side of (23) is equal to $(2k-1)x^k_{a,b} + (2k+1)x^{k+1}_{a,b} = 4k$. Since $ly'_{\overleftarrow{a,b}} = 2k$, the equation naturally holds. It verified that (23) holds in all cases.

Since the overlapped link of two neighbor working segments is counted twice in $x^k_{a,b}$, the corresponding reversed arcs of G'_p are taken by Q. This fact is formulated as (27). Note that variables $x^k_{a,b}$ for $k = 1$ or 2 can be relaxed to real due to the same reason as that of $y^k_{a,b}$.

In order to address the constraint on the restoration time for a connection, the size of each protection domain for the connection must be upper-bounded. Therefore, the following constraint is appended:

$$\sum_{\forall j \in E_w} \zeta_j x_j^k + \sum_{\forall j \in E_p} \zeta_j y_j^k \leq \zeta_{\max}, \quad k = 1, \cdots, k_{\max} \tag{25}$$

where ζ_j is the signaling delay of link j, (25) brings forth $k_{\max}$ constraints upon the sum of the physical length of the working and the corresponding protection segment for each protection domain. The constraint upon the variable $r_{u,v}$ defined in the target function is as follows:

$$x_{a,b}^k + y_{u,v}^k - 1 - \frac{sh_{u,v}^{a,b}}{b(W)} \leq r_{u,v} \tag{26}$$

where $r_{u,v} \geq 0$ for $k = 1, \cdots, k_{\max}$, $\forall (a,b) \in E_w$, $\forall (u,v) \in E_p$, and $sh_{u,v}^{a,b} + f_{u,v} \geq b(W)$.

Here, the shared risk group (SRG) constraint [1]–[5], [12], [13] is considered by using a pre-defined $|E_w| \times |E_p|$ matrix which keeps $sh_{u,v}^{a,b}$, where $(u,v) \in E_p$ and $(a,b) \in E_w$, which is the upper bound of spare capacity along link (u,v) sharable by the protection path segment if the corresponding working segment passes through link (a,b). Please see [3], [4], and [12] for detailed discussion on the preparation of the matrix. (26) ensures that when link (a,b) and (u,v) is taken by the working and protection segments in the kth protection domain, respectively, the resultant amount of scaling (i.e., $r_{u,v}$) is at least $1 - sh_{a,b}^{u,v}/b(W)$ (since $x_{a,b}^k + y_{u,v}^k = 2$). If $sh_{u,v}^{a,b} \geq n(W)$, it means that there is sufficient sharable spare capacity along link (u,v) that can be taken to protect any additional $b(W)$ units of working capacity along link (a,b). In this case, $r_{u,v} = 0$, and the only cost imposed upon the consumption of the sharable spare capacity along link (u,v) is ε, as shown in the target function.

The following constraint imposes a bandwidth limitation upon the consumption of spare capacity

$$x_{a,b}^k + y_{u,v}^k \leq 1 \tag{27}$$

where $k = 1, \cdots, k_{\max}$, $\forall (a,b) \in E_w$, $\forall (u,v) \in \underline{E}_p$, and $sh_{u,v}^{a,b} + f_{u,v} < b(W)$.

(27) ensures that if $sh_{u,v}^{a,b} + f_{u,v} < b(W)$, link (a,b) and (u,v) cannot be used at the same time for a working and protection segment in the same protection domain. Note, $r_{u,v}$ is automatically transformed from E'_p to E_p such that values of $r_{u,v}$ at those reverse arcs in E'_p are set to zero.

It is clear that the adoption of the second graph has successfully defined all the three states for the protection path to take spare capacity, which are the case of $sh_{u,v}^{a,b} \geq b(W)$, the case of $sh_{u,v}^{a,b} + f_{u,v} \geq b(W) > sh_{u,v}^{a,b}$, and the case of $sh_{u,v}^{a,b} + f_{u,v} < b(W)$. The former two cases are jointly defined by (26), where $r_{u,v}$ is constrained no smaller than $1 - sh_{u,v}^{a,b}/b(W)$ and 0 in the two cases, respectively; while the latter case is defined by (27), which prohibits the traversal of any protection segment through (u,v) if there is no sufficient capacity along the link.

With the ILP formulation, we claim that all the three states for the protection path to take spare capacity can be well defined. It can also be observed that the use of the residual graphs E_w and E_p along with the constraints of (26) and (27) has imposed a bandwidth limitation constraint along each link upon the selection of W-P segment-pair of each protection domain, respectively. Without such a design, the extra constraint on the feasibility of spare capacity resource sharing and the link bandwidth limitation for protection paths can never be defined at the same time by using a single graph.

The number of variables in an ILP formulation directly influences the computation time required to solve the formulation. In this formulation, the number of variables is $(K+4)|E_w| + (K+3)|E_p|$, and the number of rows in the constraint matrix (where the linear formulation can be expressed in a general form as $\underline{\underline{A}} \times \underline{x} = \underline{b}$ with a target to minimize $\underline{x} \times \underline{c}$) is: $8|E_w| + 9|E_p| + 11|N|$ plus the SRG constraints shown in (26) and (27). Therefore, the number of rows in the matrix $\underline{\underline{A}}$ has an upper bound $K|E_w||E_p| + 8|E_w| + 9|E_p| + 11|N|$.

B. Heuristic of Improving the Runtime of Solving ILP: The SSP Algorithm

It is clear that the computation complexity in solving the above ILP is huge and the runtime strongly depends on the parameter $k_{\max}$. Thus, selecting a proper value of $k_{\max}$ will significantly improve the computation efficiency. Furthermore, some links in the network are very unlikely to be taken by any working or protection segment, and can be simply excluded for achieving better computation efficiency without losing the optimality. We are motivated by the above observation in the design of the SSP algorithm that is committed to speed up solving the ILP.

In the SSP algorithm, a pre-calculation mechanism is devised to estimate the value of k_{max} and to exclude those edges with little chance of being taken by the working and protection segments in E_w and E_p. The first step of the pre-calculation is to derive a feasible solution, which can be done by solving the ILP formulation of [13] for SPP with additionally the restoration time constraint or any heuristic algorithm, such as the two-step approach [12]. In this study, we will solve the ILP in [13] to for the pre-calculation with the following equation being added, which addresses the restoration time constraint:

$$\sum_{\forall (a,b) \in E} \zeta_{a,b} x_{a,b} + \sum_{\forall (a,b) \in E} \zeta_{a,b}\, y_{a,b} \leq \zeta_{\max} \tag{28}$$

where $x_{a,b}$ and $y_{a,b}$ are binary flow indicators of working and protection path, respectively.

Fig. 5 shows the flowchart of the SSP algorithm. At the beginning, the algorithm try to find a feasible solution in the pre-calculation; if successful, it can serve as an upper bound on the problem since SPP is a special case of SSP. In 1) of the flowchart, if the working path of SPP is one-hop, the searching process is terminated in 3) because the derived solution must be the optimal solution for the SSP case; and it goes to 4) otherwise. If the working path consists of more hops and it intersects with the protection path, the nodes of intersection can be treated as switching and merging nodes, by which a feasible solution with a better upper bound is identified.

Let us define a detour factor of each edge assigned to the connection and network state, denoted as $\xi_{a,b}^w$ and can be calculated

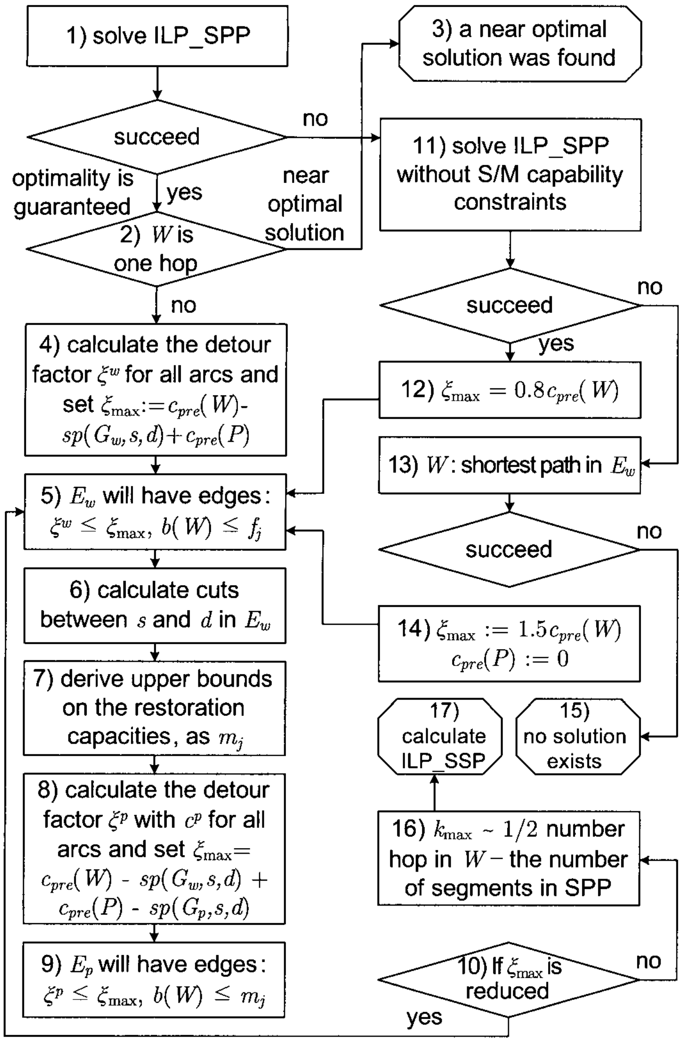


Fig. 5. The flowchart of pre-calculation process, which reduces E_p and E_w while the optimality of the result is still guaranteed.

with the following formula:

$$\xi^w_{a,b} = sp(G_w, s, a) + c_{a,b} + sp(G_w, b, d) \tag{29}$$

where $sp(G_w, s, a)$ represents the cost of the shortest path in G_w between s and a; $c_{a,b}$ represents the cost of $(a - b)$, and $sp(G_w, b, d)$ is the cost of the shortest path between b and d. The detour factor shows the minimal detour compared with that of the shortest path if the working path passes through link (a, b). Let us define $c_{pre}(W)$ and $c_{pre}(P)$ as the cost of the working and protection path derived in the pre-calculation, respectively. Let us define $x_{\max}$, which gives an upper bound on the detour of the optimal solution. In the first step $x_{\max}$ can be set to $c_{pre}(W) - sp(G_w, s, d) + c_{pre}(P)$, since the pre-calculated solution was feasible. As consequence arcs with $\xi^w_{a,b} > \xi_{\max}$ can be removed form E_w. As a result, in 5) of the flowchart, E_w will have all the edges with $\xi^w_{a,b} \leq x_{\max}$ and $b(W) \leq f_j$.

After $|E_w|$ is reduced, a much better upper bound on the available restoration capacity, m_j, can be derived, compared with using $f_j + v_j$, as shown on the 6) and 7) of the flowchart. Obviously, the working path will take at least one edge of any cut between s and d in E_w. We can get an upper bound on the restoration capacity by selecting a cut, analyzing the failure of each arc of the cut, and taking the minimum on the sharable capacity plus the free capacity; in other words,

$$m_j = \max_{C \text{ is a cut of } E_W} \min_{j \in C} sh_j + f_j. \tag{30}$$

With more cuts being analyzed, a better upper bound can be derived, with which E_p can be defined such that all the edges with $b(W) \leq m_j$ are contained. Thus, in 8) of the flowchart we can derive a lower bound on the cost of each edge taken by the protection path as:

$$c'^p_{a,b} = \frac{\max\{f_{a,b} - m_{a,b} + b(W), 0\}}{b(W)} c_{a,b}. \tag{31}$$

The above relationship holds since $f_{a,b} - m_{a,b}$ is the lower bound of the sharable capacity and $\max\{f_{a,b} - m_{a,b} + b(W), 0\}$ $/b(W)$ gives a lower bound on the ratio of the sharable capacity. With this, we can derive the detour factor of each edge in E_p (denoted as $\xi^p_{a,b}$), which is specific to the connection request and the current link-state:

$$\xi^p_{a,b} = sp(G_p, s, a) + c'^p_{a,b} + sp(G_p, b, d)$$

where $sp(G_p, s, a)$ represents the total cost of the shortest path in G_p between the source node and node a, where $c'^p_{a,b}$ is the cost of (a, b) in G_p and $sp(G_p, b, d)$ is the distance between node b and the destination node of the demand. With $sp(G_p, b, d)$ we can derive a better $x_{\max}$, such that $x_{\max} = c_{pre}(W) - sp(G_w, s, d) + c_{pre}(P) sp(G_p, b, d)$. Arcs with $\xi^p_{a,b} > x_{\max}$ are not included in E_p since the optimal protection path will never pass through it due to its large amount of detour. As a result as it is shown in the 9) box of the flowchart E_p will have all the edges where $\xi^p_{a,b} \leq x_{\max}$ and $b(W) \leq m_j$. In (10) of the flowchart, if $sp(G_p, s, d) > 0$, the algorithm switches back to further reduce $x_{\max}$.

If no feasible solution of SPP is derived in 1), a heuristic approach is developed to set $\xi_{\max}$ described as follows. In 11) of the flowchart, we solve the SPP again with the restoration time constraint relaxed. If it succeeds, we go to 12) and remove the edges of the working path according to our rule of thumb such that $\xi_{\max} = 0.7c_{pre}(W)$, and $\xi_{\max} = c_{pre}(W)$ if the restoration constraint is very tight. If the solving of SPP with the restoration constraint being relaxed still fails, we go to 13) and calculate the shortest path, and in 14) we remove the edges for the working path similarly with $\xi_{\max} = 1.2c_{pre}(W)$, $\xi_{\max} = 2c_{pre}(W)$, and $c_{pre}(P) = 0$. Obviously, if the shortest-path-first algorithm fails to find any solution, we go to 15) to halt the algorithm.

Finally, in 17) we set the value of $k_{\max}$ such that the following two quantities are considered: (a) The hop count of the shortest path between the source and destination (denoted as $|W|$), and (b) the number of intersections of the working and protection paths at nodes capable to switching and merging restoration traffic, denoted as |feasible_segments|:

$$k_{\max} = \frac{(|W| - |\text{feasible\_segments}|)}{2}. \tag{32}$$

With the SSP algorithm, the runtime of solving the ILP of SSP can be significantly reduced while the quality of result is

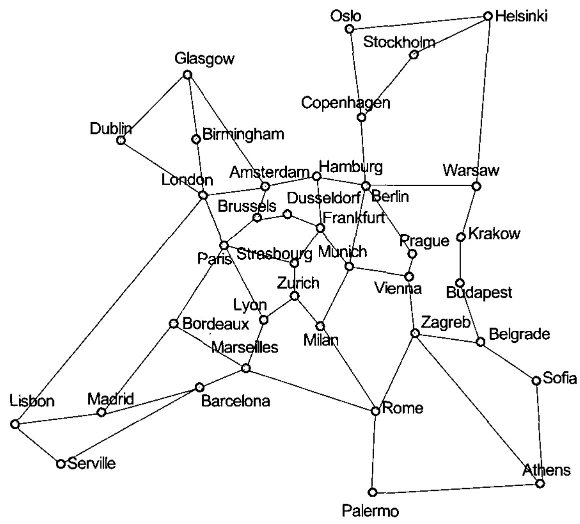


Fig. 6. European Reference Network (ERNet)

almost always guaranteed or very close to the optimal one. We will further verify the heuristic in the following section.

IV. VERIFICATION

Experiments are conducted to verify the ILP formulation using CPLEX 8.0 on Sun Ultra 80 workstation with 2 GB memory and several Linux workstations. The following three aspects are investigated:

(a) The performance gain by equipping each node with the capability of switching and merging restoration traffic;
(b) the performance impact by addressing different restoration time constraint (i.e., different upper bounds on the size of protection domains);
(c) the comparison among SLP, SPP, and SSP.

A. Performance Impact by the Switching/Merging Ability

We assume a node is either both switching- and merging-capable, or both switching- and merging-incapable. In other words, N_s and N_m do not exist in the networks at all (see Table 1). To verify the impact of having different assignments of switching/merging capability of each node, the simulation is conducted on two realistic network topologies. The first one is the pan-European fiber-optic network resulted by IST project LION and COST action 266 as [14]. It has 28 nodes and 57 bi-directional links as shown on Fig. 6. The second one is based on the US NSF Network [15] with 26 nodes and 43 bi-directional links as shown in Fig. 7. For both networks a traffic matrix in year 2005 is estimated according to [16], which is a slightly improved model than that provided in [17]. A dynamic traffic pattern is generated according to the traffic matrix such that an interrupted Poisson process [18] and Pareto inter-arrival times [19] are integrated together with exponential holding time.

Three performance metrics for evaluating the importance of the switching/merging capability at a specific node are adopted and described as follows:

- The average increase in blocking probability caused by removing the switching and merging capability at the node.

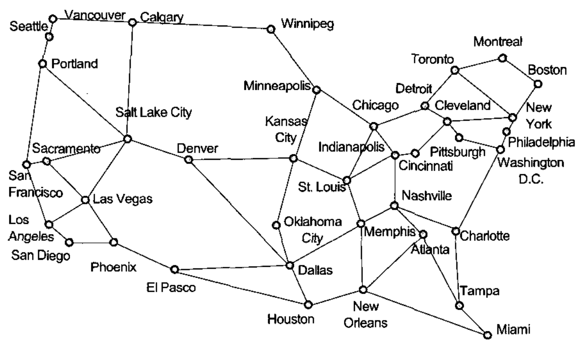


Fig. 7. North-American Reference Network (NARNet)

- The maximum increase in blocking probability caused by removing the switching and merging capability at the node.
- The average increase of total cost in allocating a connection request if the switching and merging capability at the node is removed.

To evaluate the above mentioned performance metrics at node i, the pre-defined traffic pattern is launched, each being routed by using the SSP algorithm such that node i is set to switching- and merging-incapable, while all the other nodes are switching- and merging-capable. As a result, totally $|N|$ times of invocation on the SSP algorithm is required for a connection request and each time the case that one node becomes not capable for switching and merging restoration traffic is simulated.

Figs. 8 and 9 show the simulation results of the European Reference Network with 726 connection requests and of the North-American Reference Network with 1,432 connection requests. The networks are lightly loaded with 5–10% link utilization. In NARNet, the recursion function invokes the SSP algorithm for 61,139 times and 3,265 for ERNet. In order to speed up the solving process, each connection arrival is first analyzed with the above mentioned recursive function and then is routed with the proposed SSP algorithm assuming that all nodes are switching- and merging-capable without any delay constraint.

The same simulation is conducted under heavy traffic load with 50–70% average link utilization. Very similar results to that in the light load case are observed except that the blocking probability is much higher. Thus, we will not show the results due to the lack of space.

For the European network, the switching and merging capability for restoration traffic in Paris, Lyon, Amsterdam, Berlin, Rome, and London are crucial for reducing the average blocking probability. Limiting their switching and merging capability would increase the blocking probability of the network in average of 1–1.5‰ each (altogether 6.8‰). These nodes are part of the main cuts of the network. For example, London and Amsterdam switch all traffic between the European continent and Great Britain, and removing their switching and merging capability for restoration traffic increases the blocking probability by 2.5‰ over the case before removing. In terms of cost increase, London, Berlin, Amsterdam, Paris, Munich, Frankfurt, and Rome are the crucial nodes. Removing their switching and merging capability would increase the average (weighted) capacity by 0.6–1‰ each and 5‰ altogether. It is notable that

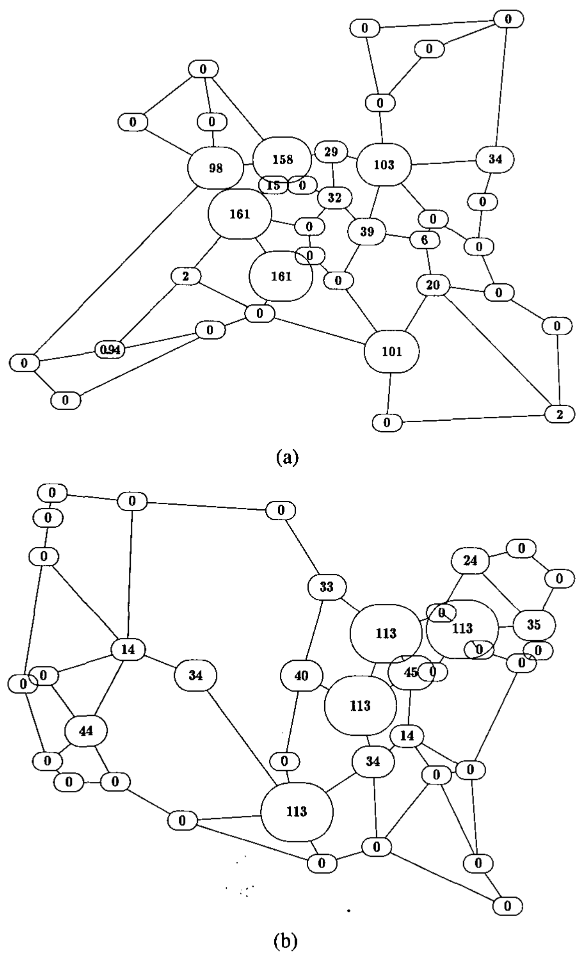


Fig. 8. The average increase of blocking probability; (a) limiting the switching, (b) merging capability of the corresponding node (each value is multiplied with 10^5).

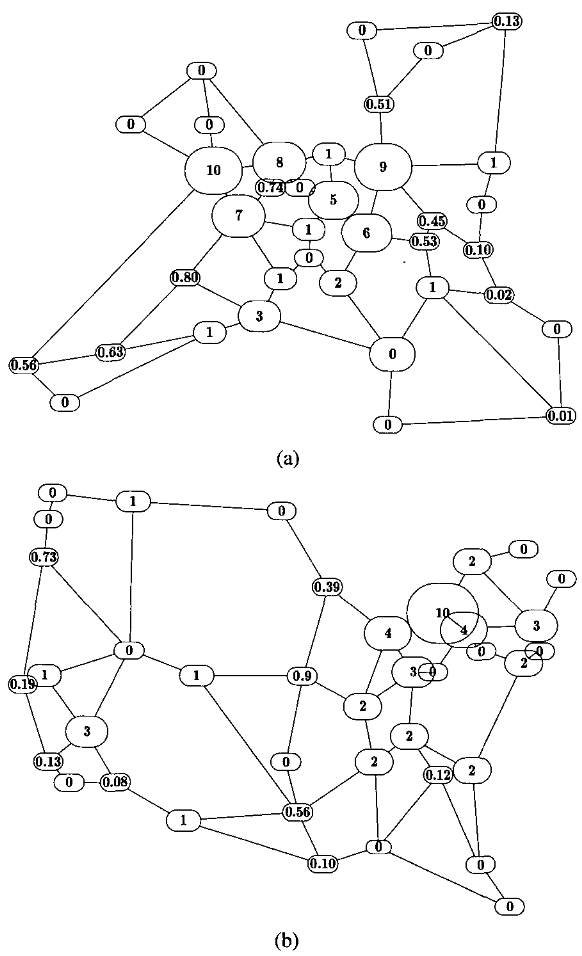


Fig. 9. The average increase of cost; (a) limiting the switching, (b) merging capability of the corresponding node (the values are in ‰).

these nodes are in the center of the network with a high nodal degree. As it was expected, these nodes are almost the same as the crucial nodes in the case for blocking probability except Munich and Frankfurt, which are in the center of Europe but not a significant part of any cut separating the network.

A general observation is made for the above simulation results as follow: Equipping nodes with switching and merging capability in the edge of the network is not effective, while nodes in the center of the graph with a higher nodal degree are more crucial than the others for both cases of blocking probability and cost.

B. Performance Impact by the Size of Protection Domains

Simulation is arranged as follows to investigate the impact by addressing the restoration constraint on the connection request. Without loss of generality, unit delay is assigned to each link in the simulation, which yields a fact that $\varsigma_j = 1$ for all link j. Therefore, the restoration constraint parameter $\varsigma_{\max}$ is nothing but the hop count of the working and protection segment-pair of each protection domain. For each connection request, the ILP is solved with different values of $\varsigma_{\max}$ (i.e., the restoration time constraint).

Fig. 10 gives a flowchart for the simulation process. For each connection request, a series of SSP problems is solved with different values of $\varsigma_{\max}$, where two performance metrics are evaluated: One is the number of successful set-up and the other is the average cost derived using the target function of the ILP. Two suites of statistic data are generated: Cost and failure statistics, which will be further detailed in the following paragraphs.

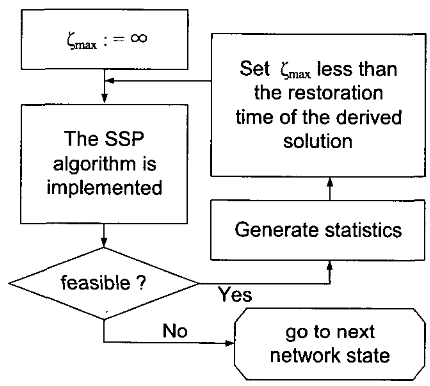


Fig. 10. The simulation process for the performance impact by addressing a restoration time constraint.

Fig. 11 shows some illustrative examples, where simulation is conducted on the network N16 (with 16 node and 27 bi-directional links) with heavy load (at average of 73% total net-

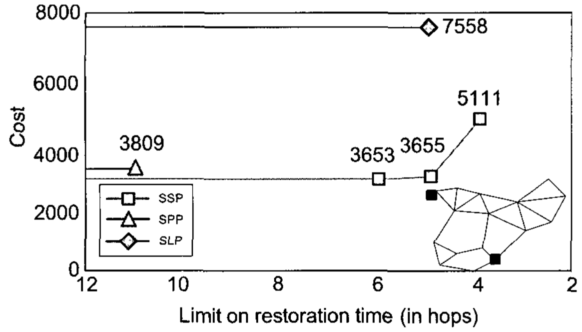


Fig. 11. An example of the restoration time versus cost.

work utilization). The graphs with the selected $s - d$ pairs are drawn on the right-bottom corner of the chart. The y axis represents the cost of the connection yielded by the target function of the ILP. It is clear that as the restoration time constraint is getting more relaxed, the performance impairment (in terms of the cost) for each connection request is reduced.

As a comparison, the optimal SPP is evaluated using the ILP formulation in [13], where the derived result for each connection is marked with triangular on the charts. It is notable that the cost of solving the SPP is at no less than that of for the SSP case since SPP is a special case of SSP (with $k_{\max} = 1$). SLP is also a special case of SSP where each working segment consists of only one link. The ILP for SLP is derived by adding an additional constraint (33) to the ILP of SSP:

$$\sum_{\forall j \in E_w} x_j^k \leq 1 \tag{33}$$

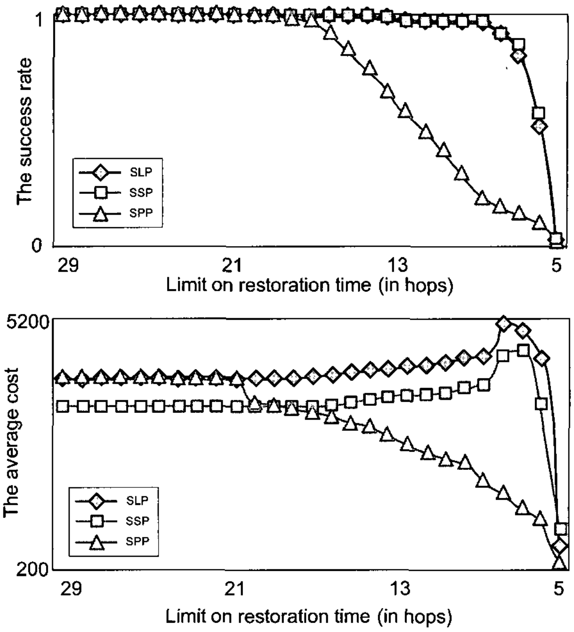
for $k = 1, \cdots, k_{\max}$.

The simulation results of SLP are marked with boxes on the charts, in which the length limitation on the protection path can be addressed using (29). The selected node-pairs are selected far from each other such that it illustrates the increase of the cost of SSP if we gradually sharpen the restoration time constraint.

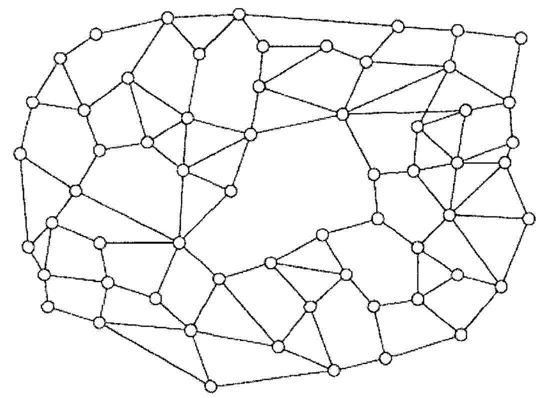
Fig. 12 shows the results using a 61-node network (shown in Fig. 12(c) with a light traffic load for comparing the three types of protection in terms of average cost and the success rate under different restoration time constraints. Results of 100 random connection requests are averaged for each data.

It is shown that an average of approximately 10% reduction in the cost can be achieved with SSP over the case of SPP if the restoration time constraint is relaxed to 13 hops. The average cost in the SSP and SLP cases increases when the restoration time constraint is sharpened, as it was expected. However, the average cost drops dramatically when the restoration time constraint is very tight since at this moment most of the long connections (with large cost) are blocked, and only those short connections (with small cost) can be allocated. The results may serve as important basis in the effort of setting up the pricing policy for connections with different lengths and restoration time requirements.

It can also be observed that SSP can yield much higher success rate for those connection requests under a tight restoration time constraint than that with SPP at the expense of taking extra cost, as shown in Fig. 11. SLP yields the highest average cost



(a)



(b)

Fig. 12. Performance impairment by addressing the restoration time constraint using the 61-node network at light load (19%); (a) the success rate and average cost, (b) 61-node network.

in all cases with a close success rate with SSP, and is seen less competitive with the other two types of protection.

We further extend the simulation study on ERNet and NARNet with the same traffic pattern and simulation environment. In order to solve a large number of connection requests, we set $x_{\max} := 0.1c_{pre}(W)$ at function block 12) and 14) in the flowchart of Fig. 5, and the maximum number of segments are set to $k_{\max} = |\text{feasible\_segments}| + 2$ at function block 16) in the flowchart. The above two steps can significantly reduce the runtime of solving the ILP for each connection at the expense of taking a little bit higher cost as well as having a larger blocking probability when the restoration time constraint is tight.

The simulation results of NARNet with the high traffic load are shown in Fig. 13, where The advantage demonstrated in using SSP is that the success rate outperforms that of the SPP case by 2 times or more at the expense of taking a little bit higher cost (as shown in Fig. 13(b)). For SLP, the overall performance is not comparable to the other two cases, although it can guarantee the

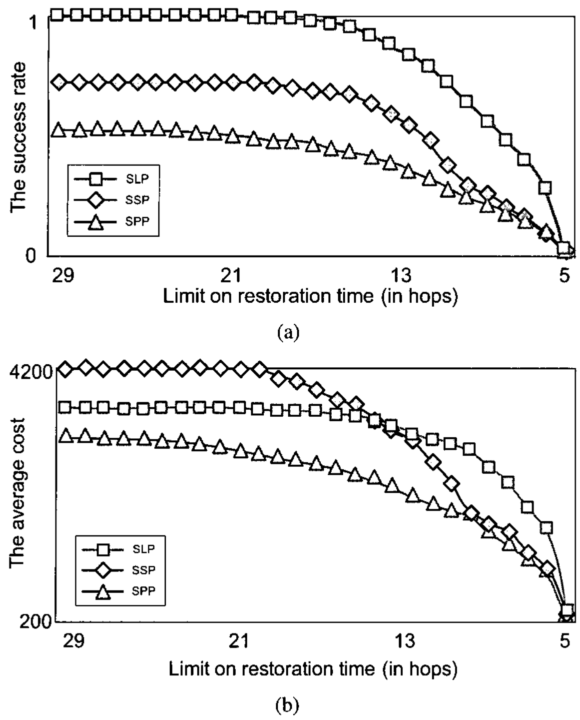
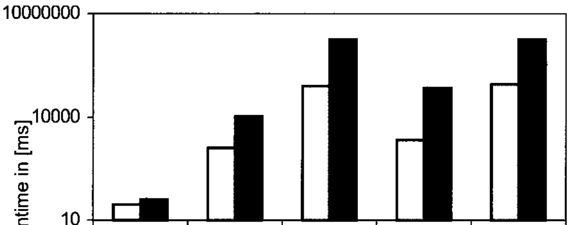


Fig. 13. Performance impairment by addressing the restoration time constraint with NARNet at high load; (a) the success rate, (b) the average cost.



	SPP	pre-calc + SLP	SLP	pre-calc + SSP	SSP
□ N16	29.8	1308	78804	2235	90257
■ ERNet	43.75	10879.75	1806225	70915.5	1805463.8

Fig. 14. Runtime (in millisecond) for solving the SPP, SLP with pre-calculation, SLP, SSP with pre-calculation, and SSP on N16 and ERNet.

shortest restoration time. In short, SSP gives the best compromise in terms of cost versus restoration time, while SLP requires an average of 10–20% additional capacity allocation compared to SSP or SPP.

As you can see with the pre-calculation does not decrease significantly the performance of SSP. It decreases the performance of SLP, but it was not scope of this study.

V. CONCLUSIONS

This paper studies dynamic survivable routing for SSP, in which a novel ILP formulation is proposed such that the switching/merging node-pairs and the corresponding least-cost W-P segment-pair for a connection request can be jointly determined in a single step. A novel approach called arc reduction pre-calculation is devised to initiate a graceful compromise between

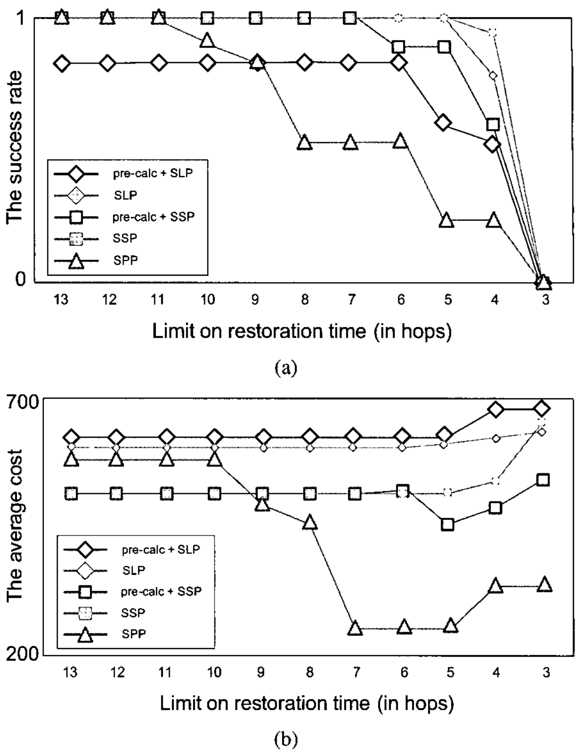


Fig. 15. Average cost and success rate of SSP-optimal and SSP with pre-calculation step. For SLP, there is an average of 3.3 gap; (a) the success rate, (b) the average cost.

the optimality and the computation time for solving the ILP. Different from the ILP model introduced in [1], this formulation addresses the capability of each node to serve as a switching and merging node for each working-protection segment during the routing process, which meets the practical requirement of network control and management with any possible limitation of hardware and signaling efforts. Extensive studies are conducted on North-American and European Reference Networks based on the estimated traffic pattern in year 2005 to investigate the benefits of equipping each node to be capable of switching and merging restoration traffic. The study also investigates the performance impairment by addressing different restoration time constraints (or having different sizes of protection domain). A comparison is made among SLP, SPP, and SSP in terms of average cost and success rate of setting up connections. The simulation results show that equipping only a small number of selective nodes with switching and merging capability can be solidly beneficial to the network performance. We also observe that SSP can initiate a graceful compromise between average cost and network throughput under a wide range of restoration time constraint. While the overall performance of SLP is not distinguished compared with the other two types of protection, it yields an ultra-fast restoration process. The modeling of the whole survivable routing process can further facilitate the deployment, and dimensioning of the network switching capacity, and can serve as a reference for setting up the pricing policy.

REFERENCES

[1] P.-H. Ho, J. Tapolcai, and T. Cinkler, "Segment shared protection in mesh communication networks with bandwidth guaranteed tunnels," *IEEE/ACM Trans. Networking*, vol. 12, pp. 1105–1118, Dec. 2004.

[2] P.-H. Ho and H. T. Mouftah, "A framework of service guaranteed shared protection for optical networks," *IEEE Commun. Mag.*, pp. 97–103, Feb. 2002.

[3] P.-H. Ho and H. T. Mouftah, "A novel survivable routing algorithm for segment shared protection in mesh wdm networks with partial wavelength conversion," *IEEE J. Sel. Areas Commun.*, to be published.

[4] D. Xu, Y. Xiong, and C. Qiao, "Protection with multi-segments PROMISE in networks with shared risk link groups SRG," in *Proc. the 40th Annual Allerton Conf. Commun., Control, and Computing*, 2002.

[5] C. V. Saradhi and C. S. R. Murthy, "Dynamic establishment of segmented protection paths in single and multi-fiber WDM mesh networks," in *Proc. SPIE OPTICOMM*, 2002, pp. 211–222.

[6] M. Kodialam and T. V. Lakshman, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," in *Proc. IEEE INFOCOM*, 2001, pp. 376–385.

[7] C.-F. Su and X. Su, "An on-line distributed protection algorithm in WDM networks," in *Proc. IEEE ICC*, 2001, pp. 1571–1575.

[8] Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi, and A. Sprintson, "Algorithms for computing QoS paths with restoration," in *Proc. IEEE INFOCOM*, 2003, pp. 1435–1445.

[9] L. Li, M. M. Buddhikot, C. Chekuri, and K. Guo, "Routing bandwidth guaranteed paths with local restoration in label switched networks," in *Proc. IEEE ICNP*, 2002, pp. 110–120.

[10] R. W. Ashford and R. C. Daniel, "Some lessons in solving practical integer programs," *J. Operational Research Society*, vol. 43, no. 5, pp. 425–433, 1992.

[11] A. M. Geoffrion and R. E. Marsten, "Integer programming algorithms: A framework and state-of-the-art survey," *Management Science*, vol. 18, pp. 465–491, 1972.

[12] P. H. Ho, (2004), "State-of-the-art progresses in developing survivable routing strategies in the optical internet." *IEEE Commun. Surveys and Tutorials*. [Online]. *6(4)*, Available:http://bbcr.uwaterloo.ca/~pinhan /CST2003.pdf

[13] P.-H. Ho, J. Tapolcai, H. T. Mouftah, and C.-H. Yeh, "Linear formulation for path shared protection," in *Proc. IEEE ICC*, 2004, pp. 20–24.

[14] LION and COST 266, "Reference networks," 2003. Part of the European Information Society Technologies (IST) Fifth Framework program.

[15] M. De, V. Mariappan, V. Chandramouli, and S. K. Kuppusamy, "US national network design," 2002. Presetation held at CReWMaN, University of Texas at Arlington.

[16] R. W. M. Vaughn, "Metropolitan network traffic demand study," in *Proc. 13th Annual Meeting LEOS*, vol. 1, 2000, pp. 102–103.

[17] R. W. A Dwivedi, "Traffic model for USA long-distance optical network," in *Proc. OFC*, 2000, pp. 156–158.

[18] K. J. Christense. (2004). Tools page. [Online]. Available:http://www. csee.usf.edu/~christen/toolpage.html

[19] M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar vbr video traffic," *ACM Computer Commun. Review*, vol. 24, pp. 269–280, Sept. 1994.

János Tapolcai received his M.Sc. (2000 in Technical Informatics) and Ph.D. (2005 in Computer Science) degrees in Technical Informatics from Budapest University of Technology and Economics (BUTE), Budapest, Hungary. Currently, he is an associate professor at the High-Speed Networks Laboratory at the Department of Telecommunications and Media Informatics at BUTE. His research interests include applied mathematics, combinatorial optimization, linear programming, linear algebra, and routing in circuit switched survivable networks. He has been involved in a few related European and Canadian projects (IP NOBEL; NoE e-Photon/ONe; BUL). He is an author of over 25 scientific publications.

Pin-Han Ho received his B.Sc. and M.Sc. Degrees from the Electrical and Computer Engineering department in the National Taiwan University in 1993 and 1995, respectively. He started his Ph.D. study in the year 2000 at Queen's University, Kingston, Canada, focusing on optical communications systems, survivable networking, and QoS routing problems. He finished his Ph.D. in 2002, and joined the Electrical and Computer Engineering Department in the University of Waterloo, Waterloo, Canada, as an assistant professor at the same year. Professor Pin-Han Ho is the author and coauthor of more than 100 refereed technical papers and book chapters, and the co-author of a book on optical networking and survivability. He is the recipient of Distinguished Research Excellent Award in the ECE department of University of Waterloo and Early Researcher Award (Premier Research Excellence Award) in 2005, the Best Paper Award in SPECTS'02, ICC'05 Optical Networking Symposium, and ICC'07 Wireless Security Symposium, and the Outstanding Paper Award in HPSR'02.