Spare Capacity Allocation in Two-Layer Networks

Yu Liu, *Member, IEEE,* David Tipper, *Senior Member, IEEE,* Korn Vajanapoom

***Abstract*— In this paper we consider the problem of provisioning spare capacity in two-layer backbone networks using shared backup path protection. First, two spare capacity allocation (SCA) optimization problems are formulated as integer linear programming (ILP) models for the cases of protection at the top layer against failures at the bottom layer. The first model captures failure propagation using overlay information between two layers for backup paths to meet diversity requirements. The second model improves bandwidth efficiency by moving spare capacity sharing from the top layer to the bottom layer. This exposes a tradeoff between bandwidth efficiency and extra cross-layer operation. Next, the SCA model for common pool protection is developed to allow spare capacity sharing between two layers. Our previous SCA heuristic technique, successive survivable routing (SSR) is extended for these optimization problems. Numerical results for a variety of networks indicate that the common pool protection is attractive to enhance bandwidth efficiency without loss of survivability and that the SSR heuristic quickly results in near optimal solutions.**

***Index Terms*— multi-layer network design, spare capacity allocation, resilient network design, shared risk link group.**

I. Introduction

SURVIVABILITY in the face of failures has become an essential property of backbone transport networks. Current backbone data networks have multiple layers such as IP/MPLS over SONET over an optical layer and are converging toward a two-layer architecture of IP/MPLS or GMPLS over an optical transport layer.

Survivability techniques in two-layer networks can be classified as: survivability at the bottom layer, survivability at the top layer, and survivability at both layers, depending on, the layer in which the survivability technique is deployed [1]. In the bottom-layer approach, recovery from a failure is performed only at the bottom layer (e.g., recovering failed lightpaths in an optical transport network). This scheme has the benefits that it is simple and provides fast recovery of aggregate traffic. However, the major drawback of this scheme is that it cannot recover from failures that occur at the top layer, such as, the failure of a top-layer router or its interfaces. In the survivability at the top-layer scheme, failure recovery is performed only at the top layer, (e.g., recovering failed label switched paths (LSPs) in a MPLS network using fast reroute). The advantage of this scheme is that it can recover from failures that occur in both layers. It also allows a service differentiation among top-layer flows by recovering each individual flow at the top layer, which is difficult in the bottom-layer survivability scheme where an aggregate of top-layer flows is recovered. Among the drawbacks of this approach are its complexity and slower speed of fault recovery.

One of the major problems in the survivability of such two-layer networks is *failure propagation*, which occurs when the failure of a bottom-layer link or node results in the simultaneous failure of multiple top-layer links [1], [2], [3]. If failure propagation is not considered appropriately in multi-layer networks, the survivability at the top layer technique may fail to recover the communication services after a failure. Several approaches have been proposed to design survivable virtual topologies at the top layer while taking failure propagation into account [4], [5], [6], [7], [8], [9]. In part due to failure propagation, each layer of a network will typically employ self-healing capabilities to address faults occurring in their own layer. In such a multi-layer scheme, coordination between layers is required to provide an efficient recovery process upon a failure. This coordination is called an escalation strategy, which determines which layer will perform a recovery first in response to a particular failure, and when and how a responsibility will be transferred to another layer if the current layer fails to recover from the failure [1], [10].

Recently several papers have appeared on protection in IP/MPLS over WDM networks [11], [12], [13], [14]. In [11], the authors provide two classes of integer linear programming (ILP) models for IP restoration and WDM shared protection. These models consider the equipment constraints on transmitters and receivers and compare restoration requirements at both layers. Heuristic algorithms are used to compare the maximum guaranteed network capacity and recovery time. Zheng and Mohan [12] evaluate protection schemes at the LSP level and the lightpath level for dynamic traffic in IP/MPLS over WDM networks. Their results indicate that inter-level sharing (ILS) could improve the resource utilization. Koo, et. al., [13] study the ILP model to minimize the total cost to route primary and backup LSPs over lightpaths using MPLS protection against the failures of a Label Switched Router (LSR) and Shared Risk Link Groups (SRLG)s. A two-phase heuristic to find primary and backup LSPs sequentially is proposed. Ou, et. al., [14] describe three traffic grooming schemes on IP/WDM networks: Protection-At-Lightpath, Separated Protection-At-Connection, and Mixed Protection-At-Connection. Shared backup protection on lightpaths or links against single fiber failures are assumed. Heuristic routing algorithms are developed to compare the bandwidth blocking ratios and the resource efficiency ratios under various loads.

Manuscript received April 15, 2006; revised Dec. 9, 2006. This work was supported by the National Science Foundation under Grant ANIR 9980516 and by the Defense Advanced Research Projects Agency under Grant F30602-97-1-0257. An earlier version of this paper was presented at the 5th Design of Reliable Communication Networks (DRCN) workshop, Island of Ischia, Italy, 2005.

Y. Liu is with OPNET Technologies, 5201 Great America Parkway, Suite 529, Santa Clara, CA 95054, USA (e-mail: yliu@opnet.com).

D. Tipper and K. Vajanapoom are with the Telecommunications Program, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: tipper@tele.pitt.edu, kov2@pitt.edu).

Digital Object Identifier 10.1109/JSAC.2007.070610.

TABLE I
ACRONYMS

SCA	Spare capacity allocation
SPM	Spare provision matrix
SSR	Successive survivable routing algorithm
BB	Branch and bound algorithm
InP	Integer programming
FID	Failure-independent path restoration
FD	Failure-dependent path restoration
FDStubR	Failure-dependent path restoration with stub release
iff	if and only if

TABLE II
NOTATION

N, L, R, K	Numbers of nodes, links, flows and failures
n, l, r, k	Indices of nodes, links, flows and failures
$\boldsymbol{P} = \{\boldsymbol{p}_r\} = \{p_{rl}\}$	Working path link incidence matrix
$\boldsymbol{Q} = \{\boldsymbol{q}_r\} = \{q_{rl}\}$	Backup path link incidence matrix
$\boldsymbol{M} = \text{Diag}(\{m_r\})$	Diagonal matrix of bandwidth m_r of flow r
$\boldsymbol{G} = \{g_{lk}\}_{L\times K}$	Spare provision matrix, g_{lk} is spare capacity on link l for failure k
$\boldsymbol{G}^r = \{g^r_{lk}\}_{L\times K}$	Contribution of flow r to $\boldsymbol{G}$
$\boldsymbol{s} = \{s_l\}_{L\times 1}$	Spare capacity vector
$\boldsymbol{\phi} = \{\phi_l\}_{L\times 1}$	Spare capacity cost function
W, S	Total working, spare capacity
$\eta = S/W$	Network redundancy
$o(r), d(r)$	Origin/destination nodes of flow r
$\boldsymbol{v}_r = \{v_{rl}\}$	Vector of incremental spare capacity cost for flow r on link l
$\boldsymbol{B} = \{b_{nl}\}_{N\times L}$	Node link incidence matrix
$\boldsymbol{D} = \{d_{rn}\}_{R\times N}$	Flow node incidence matrix
$\boldsymbol{F} = \{f_{kl}\}_{K\times L}$	Failure link incidence matrix, $f_{kl} = 1$ iff link l fails in failure scenario k
$\boldsymbol{U} = \{u_{rk}\}_{R\times K}$	Flow failure incidence matrix, $u_{rk} = 1$ iff failure scenario k will affect flow r's working path
$\boldsymbol{T} = \{t_{rl}\}_{R\times L}$	Flow tabu-link matrix, $t_{rl} = 1$ iff link l should not be used on flow r's backup path

In this paper, we provide several ILP models for the SCA problem in two-layer networks. We consider not only failure propagation between layers but also *cross layer spare capacity sharing*. Initially we focus on the fault independent shared backup path restoration. First, we consider the SCA problem for the case when fault recovery is performed at the bottom layer only. Next we derive two models for the SCA problem when fault recovery occurs only in the top-layer network. The first model captures failure propagation by extending the matrix-based SCA formulation for a single layer network given in [15]. The second model further improves the first one by allowing the top-layer survivability technique to share spare capacity at the bottom layer. Next, the SCA model for protection at both layers with spare capacity sharing across layers is presented. This model will have working and backup routes on both layers. In addition, we show how failure dependent shared backup path restoration with and without stub release can be incorporated into the SCA models. Numerical results for a variety of network scenarios are given for each SCA model. The numerical results are determined using both the commercial optimization solver CPLEX and a heuristic algorithm, called the successive survivable routing (SSR) algorithm. Lastly, these results are discussed and the conclusions are presented.

II. SPARE CAPACITY ALLOCATION MODELING

The survivable two-layer network design problem contains a survivable topology layout problem and the SCA problem. This paper formulates them separately. The main reason is because of their different design cycles. The network topology has a much longer design cycle of multiple years. On the other hand, backbone LSP connections have shorter life spans ranging from weeks to months. The partitioning of these two problems allows one to trade off design optimality with a better solution speed for the survivable network design problem. In the appendix, the survivable topology layout problem is formulated. This problem finds a valid inter-layer mapping so that any single bottom-layer failure will not partition the top-layer topology. This section briefly reviews the SCA model and the SSR algorithm for a single layer network presented in [15], in order to provide background for the two-layer SCA models and SSR extensions in the next section.

A. The Single Layer Spare Capacity Allocation Model

We assume the network under study uses failure independent path restoration (FID) for an arbitrary failure condition. FID is also called path restoration with disjoint routes, where a backup path is always disjoint (either link or node) from its working path. We assume all traffic flows require a 100% restoration level for any failure, which requires that all affected flows be detoured to their backup paths upon any given failure. Provisioning enough spare capacity is the prerequisite condition to such restoration. The acronyms used in this paper are given in Table I. Similarly, the notation adopted is summarized in Table II.

We model an uncapacitated network [1] by a directed graph of N nodes, L links, and R flows. Flow $r, 1 \le r \le R$ has its origin/destination node pair $(o(r), d(r))$ and traffic demand m_r. Working and backup paths of flow r are represented by two $1 \times L$ binary row vectors $\boldsymbol{p}_r = \{p_{rl}\}$ and $\boldsymbol{q}_r = \{q_{rl}\}$ respectively. The l-th element in one of the vectors equals to one *if and only if* (iff) the corresponding path uses link l. The path link incidence matrices for working and backup paths are the collections of these vectors, forming two $R \times L$ matrices $\boldsymbol{P} = \{p_{rl}\}$ and $\boldsymbol{Q} = \{q_{rl}\}$ respectively. Let $\boldsymbol{M} = \text{Diag}(\{m_r\}_{R\times 1})$ denote the diagonal matrix representing demand bandwidth. The topology is represented by the node-link incidence matrix $\boldsymbol{B} = (b_{nl})_{N\times L}$ where $b_{nl} = 1$ *or* -1 iff node n is the origin *or* the destination node of link l. $\boldsymbol{D} = (d_{rn})_{R\times N}$ is the flow node incidence matrix where $d_{rn} = 1$ or -1 iff $o(r) = n$ or $d(r) = n$. We characterize K failure scenarios in a binary matrix $\boldsymbol{F} = \{\boldsymbol{f}_k\}_{K\times 1} = \{f_{kl}\}_{K\times L}$. The row vector $\boldsymbol{f}_k$ in $\boldsymbol{F}$ is for failure scenario k and its element f_{kl} equals one iff link l fails in scenario k. In this way, each failure scenario includes a set of one or more links that will fail simultaneously in the scenario. For a failed node, all its adjacent links are marked as failed. We also denote a flow failure incidence matrix $\boldsymbol{U} = \{\boldsymbol{u}_r\}_{R\times 1} = \{u_{rk}\}_{R\times K}$, where $u_{rk} = 1$ iff

[1]The assumption of unlimited link bandwidth allows this study to concentrate on the efficiency of spare capacity sharing with little influence from link capacity. The results could be useful for lightly loaded networks. The realistic constraints such as link capacity constraints or traffic delay constraints can be added for network planning purposes.

flow r will be affected by failure k, and $u_{rk} = 0$ otherwise. The flow tabu-link matrix $\boldsymbol{T} = \{\boldsymbol{t}_r\}_{R\times 1} = \{t_{rl}\}_{R\times L}$ has $t_{rl} = 1$ iff the backup path of flow r should not use link l, and $t_{rl} = 0$ otherwise. We can find $\boldsymbol{U}$ and $\boldsymbol{T}$ given $\boldsymbol{P}$ and $\boldsymbol{F}$ as shown in equations (7) and (8) respectively. A binary matrix multiplication operation "$\odot$" is used in equations (7) and (8). It is a matrix multiply operator that is identical to normal matrix multiplication except that the general numerical addition $1 + 1 = 2$ will be replaced by the boolean addition $1+1 = 1$ as described in [16]. Using this binary operator, the logical relations among links, paths and failure scenarios are simplified into two matrix operations.

Let $\boldsymbol{G} = \{g_{lk}\}_{L\times K}$ denote the *spare provision matrix* (SPM) whose elements g_{lk} are the minimum spare capacity required on link l when failure k occurs. Note that $K = L$ when the SCA protects all single link failures. Given the backup paths $\boldsymbol{Q}$, the demand bandwidth matrix $\boldsymbol{M}$, the working path $\boldsymbol{P}$, and the failure matrix $\boldsymbol{F}$, G can be determined by equation (3) with the help of (7). The minimum spare capacity required on each link is denoted by the column vector $\boldsymbol{s} = \{s_l\}_{L\times 1}$, which is found in equation (2). The function $\max$ in (2) asserts that an element in $\boldsymbol{s}$ is equal to the maximum element in the corresponding row of $\boldsymbol{G}$. It is equivalent to $\boldsymbol{s} \geq \boldsymbol{G}$ in this optimization model. Let ϕ_l denote the cost function of spare capacity on link l. $\boldsymbol{\phi} = \{\phi_l\}_{L\times 1}$ is a column vector of these cost functions and $\boldsymbol{\phi}(\boldsymbol{s})$ gives the cost vector of the spare capacities on all links. The total cost of spare capacity in the network is $\boldsymbol{e}^T\boldsymbol{\phi}(\boldsymbol{s})$, where $\boldsymbol{e}$ is unit column vector of size L. Here for simplicity, we assume all cost functions $\boldsymbol{\phi}(\boldsymbol{s})$ are identity functions, i.e., $\boldsymbol{\phi}(\boldsymbol{s}) = \boldsymbol{s}$.

Given the notation and definitions above the spare capacity allocation (SCA) problem can be formulated as follows.

$$\min_{\boldsymbol{Q},\boldsymbol{s}} \quad S = \boldsymbol{e}^T\boldsymbol{s} \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{s} = \max \boldsymbol{G} \tag{2}$$

$$\boldsymbol{G} = \boldsymbol{Q}^T\boldsymbol{M}\boldsymbol{U} \tag{3}$$

$$\boldsymbol{T} + \boldsymbol{Q} \leq 1 \tag{4}$$

$$\boldsymbol{Q}\boldsymbol{B}^T = \boldsymbol{D} \tag{5}$$

$$\boldsymbol{Q} : binary \tag{6}$$

$$\boldsymbol{U} = \boldsymbol{P} \odot \boldsymbol{F}^T \tag{7}$$

$$\boldsymbol{T} = \boldsymbol{U} \odot \boldsymbol{F} \tag{8}$$

This SCA problem has the objective to minimize the total spare capacity in (1) with the constraints (2)–(8). The decision variables are the backup path matrix $\boldsymbol{Q}$ and the spare capacity vector $\boldsymbol{s}$. Constraints (2) and (3) associates these variables, i.e., the spare capacity allocation $\boldsymbol{s}$ is derived from the backup paths in $\boldsymbol{Q}$. Constraint (4) guarantees that every backup path will not use any link which might fail simultaneously with any link on its working path. Flow conservation constraint (5) guarantees that backup paths given in $\boldsymbol{Q}$ are feasible paths of flows in a directed network. Note, the incidence matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ are precomputed. The matrix $\boldsymbol{U}$ indicates the failure cases that will influence the working paths. The matrix $\boldsymbol{T}$ indicates the links that should be avoided in the backup paths. In this paper, the link load, the traffic flows and their routes are assumed symmetric. In a directed network, each link might have two directions with asymmetric load. In this case, the dimensions of these matrices should be doubled, i.e. $2L$, instead of L. More detailed explanation of the above SCA model is in [15, eq.(7)-(14)].

B. The Successive Survivable Routing Algorithm

The SCA model formulated above is a mixed ILP problem. It is NP-complete. Hence, solving the problem for large networks is infeasible using standard integer programming solution methods. In [15], we proposed the successive survivable routing (SSR) heuristic algorithm. The SSR algorithm finds solutions by routing backup paths iteratively. Each backup path computation uses the shortest path algorithm. The link routing metric is the *incremental spare capacity* $\boldsymbol{v}_r = \{v_{rl}\}$. It is computed from the most recent spare provision matrix that is further based on previously routed backup paths. After all flows find their backup paths, SSR continues to update existing backup paths whenever a new one could use less spare capacity. This process keeps reducing total spare capacity until it converges, (i.e., no more backup path updates). Different random ordering of the flows for routing backup paths are used to provide diversity and avoid local minima. The best solution is used as the final one, which in numerical results given in [15] show as near optimal. The SSR scheme includes five steps for flow r as shown below.

- Step 1 calculates the failure impact vector $\boldsymbol{u}_r$ and the tabu link vector $\boldsymbol{t}_r$ using the failure matrix $\boldsymbol{F}$, the working path row vector $\boldsymbol{p}_r$, and the destination node $d(r)$. $\boldsymbol{u}_r$ and $\boldsymbol{t}_r$ are the r-th row vectors in the matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ found in (7) and (8).
- Step 2 recomputes the spare provision matrix $\boldsymbol{G}$ using (3) periodically, i.e., before each backup route update.
- Step 3 calculates the link metric $\boldsymbol{v}_r$ from $\boldsymbol{G}$ and traffic flow r's contribution $\boldsymbol{G}^r = m_r(\boldsymbol{q}_r^T\boldsymbol{u}_r), 1 \leq r \leq R$ as follows:
  (a) Given $\boldsymbol{G}$, $\boldsymbol{q}_r$ and $\boldsymbol{G}^r$ for current flow r, let $\boldsymbol{G}^{-r} = \boldsymbol{G} - \boldsymbol{G}^r$ and $\boldsymbol{s}^{-r} = \max \boldsymbol{G}^{-r}$ be the spare provision matrix and the link spare capacity vector after current backup path $\boldsymbol{q}_r$ is removed.
  (b) Let $\boldsymbol{q}_r^*$ denote an alternative backup path for flow r, and function $\boldsymbol{G}^{r*}(\boldsymbol{q}_r^*) = m_r\boldsymbol{q}_r^{*T}\boldsymbol{u}_r$. Then, this new path $\boldsymbol{q}_r^*$ produces a new spare capacity reservation vector in a function format of $\boldsymbol{s}^*(\boldsymbol{q}_r^*) = \max(\boldsymbol{G}^{-r} + \boldsymbol{G}^{r*}(\boldsymbol{q}_r^*))$.
  (c) Let $\boldsymbol{q}_r^* = \boldsymbol{e} - \boldsymbol{t}_r$, which assumes the backup path uses all non-tabu links. Then, we can find the vector of *link metrics* for flow r as

$$\begin{aligned} \boldsymbol{v}_r &= \{v_{rl}\}_{L\times 1} \\ &= \boldsymbol{\phi}(\boldsymbol{s}^*(\boldsymbol{e} - \boldsymbol{t}_r)) - \boldsymbol{\phi}(\boldsymbol{s}^{-r}), \end{aligned} \tag{9}$$

  where $\boldsymbol{t}_r$ is the binary flow tabu-link vector of flow r. The element v_{rl} is the cost of the incremental spare capacity on link l if this link is used on the backup path.
- Step 4 uses the shortest path algorithm with the link metric $\boldsymbol{v}_r$ to find a new or updated backup path $\boldsymbol{q}_r^{new}$. This path excludes all the tabu links indicated in $\boldsymbol{t}_r$.
- Step 5 replaces the original backup path $\boldsymbol{q}_r$ with the new backup path $\boldsymbol{q}_r^{new}$ if the new one has the lower

cost based on the link metrics in $\boldsymbol{v}_r$, i.e., only when $\boldsymbol{v}_r^T \boldsymbol{q}_r > \boldsymbol{v}_r^T \boldsymbol{q}_r^{new}$.

- After Step 5, SSR continues to Step 2 to update the backup path for another flow. This iterative process not only finds backup route, but also helps to minimize the required total spare capacity shared among all backup routes.
- After all flows have found backup paths, the iteration continues until the termination condition is met. The termination condition can be: (1) there is no backup update for all flows in the recent iteration, or (2) when the maximum number of updates is reached.

More detailed description of SSR is in [15, §V].

III. Two-Layer SCA Models

In this section, we extend the previous SCA model to a two-layer network. In the top-layer network case, the notation of the previous section is reused, and the same notation with the superscript "b" is used for the bottom layer. A top-layer link is carried by a bottom-layer path. Overlay information is defined by the interlayer link incidence matrix $\boldsymbol{H} = \{h_{ij}\}_{L \times L^b}$, where $1 \leq i \leq L$, $1 \leq j \leq L^b$. Element h_{ij} equals to one *iff* the top-layer link i uses the bottom-layer link j. Given the top-layer spare capacity allocation vector $\boldsymbol{s}$, its equivalent bottom-layer spare capacity vector $\boldsymbol{s}^b$ is given by:

$$\boldsymbol{s}^b = \boldsymbol{H}^T \boldsymbol{s}. \tag{10}$$

Usually, each bottom-layer link carries one or multiple top-layer links; therefore a failure of single bottom-layer link could tear down multiple top-layer links simultaneously. In order to provide restoration at the top layer, the interlayer link incidence matrix $\boldsymbol{H}$ should guarantee that a failure of any single bottom-layer link would not partition the top-layer topology. This property is called *immunity from failure propagation*. A math programming model to find such a mapping $\boldsymbol{H}$ for the single link failure case is provided in [5]. In the Appendix, we provide a matrix based formulation for this survivable topology layout problem. This model considers various failure scenarios in addition to single link failures. The solution of this model provides the inter-layer link mapping $\boldsymbol{H}$ so that the top-layer topology has enough connectivity to be resilient to bottom-layer failures.

Given the interlayer information $\boldsymbol{H}$ several multi-layer SCA problems are formulated. First, the multi-layer SCA problem for the FID path restoration case is discussed. In the FID case, each flow has a single backup path disjoint from any failures that affect its working path. Based on the layer where the FID restoration scheme exists, the SCA model can be classified into three approaches, namely: (1) restoration at the bottom layer, (2) restoration at the top layer, or (3) restoration at both layers. Lastly, the SCA problems for the failure-dependent (FD) path restoration case with and without stub release are discussed.

A. Restoration at the bottom layer

When the traffic flows are at the top layer, but the fault protection is only available at the bottom layer, each top-layer *link* needs to have a bottom-layer backup path preplanned besides its working path. The top-layer traffic flow is not aware of the bottom-layer restoration. This scheme is simple and it can deliver fast restoration upon failure. However, it might not protect all top-layer failures such as the failure of a top-layer router or its interfaces.

The SCA model can be simplified to one at the bottom layer only. Assuming the top-layer links are protected individually at the bottom layer, the modifications to the SCA problem of the previous section are the bottom-layer working paths are derived from the layout information $\boldsymbol{H}$ in (11), and the traffic matrix $\boldsymbol{M}^b$ is changed. Specifically, $\boldsymbol{M}^b$ is a $L \times L$ diagonal matrix with diagonal elements m_r^b equal to the carried traffic w_l on the top-layer links $l, 1 \leq l \leq L$ in (12).

$$\boldsymbol{P}^b = \boldsymbol{H} \tag{11}$$

$$\boldsymbol{M}^b = Diag(m_r^b) = \text{Diag}(w_l, 1 \leq l \leq L). \tag{12}$$

Thus we solve the single layer SCA model of the previous section with modifications of (11) and (12).

B. Restoration at the top layer

When the restoration is at the top layer, equations (1)-(8) in the previous section are extended based on two alternate usages of the overlay information $\boldsymbol{H}$. We denote the two models proposed as [A] and [B] and modifications are denoted by superscripts added to variables in the models.

***Model A**: Use of overlay information for failure propagation:* To protect against failure propagation of any single bottom-layer link failures, the overlay information $\boldsymbol{H}$ is used to derive the failure scenario matrix $\boldsymbol{F}$ for the top-layer SCA model as shown in (13). The flow failure incidence matrix $\boldsymbol{U}$ and the spare provision matrix $\boldsymbol{G}^{[A]}$ are modified in (14) and (15) to replace (7) and $\boldsymbol{G}$ in (3) respectively.

$$\boldsymbol{F} = \boldsymbol{F}^b \odot \boldsymbol{H}^T \tag{13}$$

$$\boldsymbol{U} = \boldsymbol{P} \odot \boldsymbol{F}^T = \boldsymbol{P} \odot (\boldsymbol{F}^b \odot \boldsymbol{H})^T = \boldsymbol{P} \odot \boldsymbol{H} \odot \boldsymbol{F}^{b^T} \tag{14}$$

$$\boldsymbol{G}^{[A]} = \boldsymbol{Q}^T \boldsymbol{M} \boldsymbol{U} = \boldsymbol{Q}^T \boldsymbol{M} (\boldsymbol{P} \odot \boldsymbol{H} \odot \boldsymbol{F}^{b^T}) \tag{15}$$

In addition, the objective function to minimize the total spare capacity (1) is replaced by (16), where $\boldsymbol{e}^T \boldsymbol{H}^T$ is used to compute the actual spare capacity at the bottom layer reserved by the top-layer links.

$$\min_{\boldsymbol{Q}} S^{[A]} \quad = \quad \boldsymbol{e}^T \boldsymbol{s}^b = \boldsymbol{e}^T \boldsymbol{H}^T \boldsymbol{s} \tag{16}$$

***Model B**: Use of overlay information for both failure propagation and cross-layer spare capacity reservation:* In addition to using the overlay information for failure propagation, the second model computes the top-layer spare capacity sharing at the bottom layer instead of at the top layer. Since every top-layer link traverses one or more bottom-layer links, spare capacity sharing on these bottom-layer links could further minimize the total spare capacity required. The disadvantage of this model is the requirement of a method for the top layer to record and reserve spare capacity on the bottom-layer links.

As an example to illustrate this approach consider the two-layer network of Fig. 1. The overlay information $\boldsymbol{H}$ for the network is given in (17). In the figure, two working flows *a–b* and *c–d* at the top-layer traverse the bottom-layer links *1* and *5* respectively. Their backup paths are *a–c–b* and *c–b–a–d*

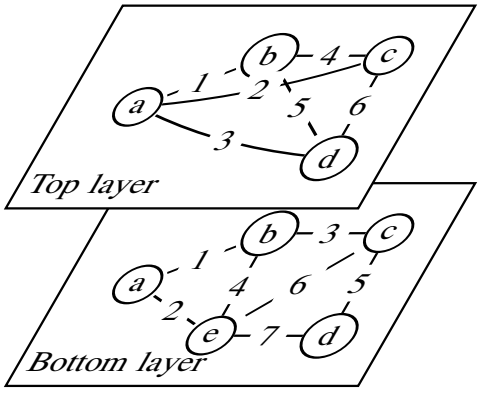

Fig. 1. Network 0: 5-node overlay network

respectively. The backup paths use the top-layer links *2* (*a–c*) and *3* (*a–d*) respectively. The bottom-layer paths of these top-layer links overlap on the bottom-layer link *2* (*a–e*) whose spare capacity could be shared by the backup paths. In this example, the top-layer not only uses the overlay information $\boldsymbol{H}$ to avoid failure propagation, but also shares spare capacity at the bottom layer to achieve lower redundancy.

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{17}$$

This spare capacity sharing scheme is equivalent to: (i) converting all backup paths in $\boldsymbol{Q}$ at the top layer into ones at the bottom layer by multiplexing them with $\boldsymbol{H}$. (ii) finding the spare capacity reserved at the bottom layer

$$\boldsymbol{s}^b = \max((\boldsymbol{QH})^T \boldsymbol{MU}) = \max(\boldsymbol{H}^T \boldsymbol{G}^{[A]}). \tag{18}$$

(iii) then minimizing the objective function, the total spare capacity

$$\min_{\boldsymbol{Q}} S^{[B]} = \boldsymbol{e}^T \boldsymbol{s}^b = \boldsymbol{e}^T \max(\boldsymbol{H}^T \boldsymbol{G}^{[A]}). \tag{19}$$

Since $\max(\boldsymbol{H}^T \boldsymbol{G}^{[A]}) \leq \boldsymbol{H}^T \max(\boldsymbol{G}^{[A]})$, the total spare capacity using Model B will be equal to or smaller than that in Model A.

$$S^{[B]} \leq S^{[A]} \tag{20}$$

C. Modified Successive Survivable Routing Algorithm

Model B requires modifications of the SSR algorithm [15, §V] to approximately solve the optimization problem. In this model, spare capacity sharing has been moved from the top layer down to the bottom layer. Thus the spare provision matrix and the computation of the incremental spare capacity vector are both moved down to the bottom layer as well. Routing backup paths at the top layer needs to use the overlay information matrix $\boldsymbol{H}$ in the modified SSR algorithm as follows.

In Step 1, given the working path vector $\boldsymbol{p}_r$ and the failure matrix $\boldsymbol{F}$, the failure vector $\boldsymbol{u}_r$ and the tabu-link vector $\boldsymbol{t}_r$ are computed.

In Step 2, the top-layer spare provision matrix $\boldsymbol{G}^{[A]}$ is updated periodically. In the end of Step 2, as shown in Model B, the spare provision matrix $\boldsymbol{G}^{[A]}$ is converted to its bottom-layer equivalent $\boldsymbol{H}^T \boldsymbol{G}^{[A]}$ in (18).

In Step 3, we assume symbol $\boldsymbol{G}$ is equivalent to $\boldsymbol{G}^{[A]}$ in Model B. It contains four phases:

(a) First, we need to derive the spare provision matrix after the current backup path $\boldsymbol{q}_r$ is removed. It is denoted by $\boldsymbol{G}^{-r}$, and is found using the top-layer spare provision matrix $\boldsymbol{G}$, the flow r's SPM contribution $\boldsymbol{G}^r$. Specifically, $\boldsymbol{G}^{-r} = \boldsymbol{G} - \boldsymbol{G}^r$. Next, based on (18), the bottom-layer spare capacity $\boldsymbol{s}^{b^{-r}}$ after removing the backup path of flow r is determined as

$$\boldsymbol{s}^{b^{-r}} = \max(\boldsymbol{H}^T \boldsymbol{G}^{-r}). \tag{21}$$

(b) Let $\boldsymbol{q}_r^*$ denote an alternative backup path for flow r. Let $\boldsymbol{G}^{r*}(\boldsymbol{q}_r^*) = m_r \boldsymbol{q}_r^{*T} \boldsymbol{u}_r$ denote the SPM contribution matrix based on this backup path. Then, the new spare capacity reservation vector based on this backup path is denoted as

$$\boldsymbol{s}^{b^*}(\boldsymbol{q}_r^*) = \max(\boldsymbol{H}^T(\boldsymbol{G}^{-r} + \boldsymbol{G}^{r*}(\boldsymbol{q}_r^*)). \tag{22}$$

(c) Let $\boldsymbol{q}_r^* = \boldsymbol{e} - \boldsymbol{t}_r$, which assumes the backup path is using all non-tabu links. Then, we can find the incremental spare capacity vector for flow r at the bottom layer as

$$\begin{aligned} \boldsymbol{v}_r^b &= \{v_{rl}^b\}_{L^b \times 1} = \boldsymbol{\phi}^b(\boldsymbol{s}^{b^*}(\boldsymbol{e} - \boldsymbol{t}_r)) - \boldsymbol{\phi}^b(\boldsymbol{s}^{b^{-r}}) \\ &= \boldsymbol{s}^{b^*}(\boldsymbol{e} - \boldsymbol{t}_r) - \boldsymbol{s}^{b^{-r}}, \end{aligned} \tag{23}$$

where $\boldsymbol{t}_r$ is the binary flow tabu-link vector of flow r. The element v_{rl}^b is the incremental spare capacity cost on link l at the bottom layer if this link is used by a top-layer link on the backup path.

(d) At the end of Step 3, the incremental spare capacity vector $\boldsymbol{v}_r$ at the top layer is derived from its equivalent vector $\boldsymbol{v}_r^b$ at the bottom layer by

$$\boldsymbol{v}_r = \boldsymbol{H} \boldsymbol{v}_r^b. \tag{24}$$

Step 4 and 5 remain the same as the original SSR algorithm. Step 4 first excludes all the tabu links marked in $\boldsymbol{t}_r$, then uses a shortest path algorithm with link metrics $\boldsymbol{v}_r$ to find an updated backup path $\boldsymbol{q}_r^{new}$ at the top layer. In Step 5, the original backup path $\boldsymbol{q}_r$ is replaced by the new path $\boldsymbol{q}_r^{new}$ when the new path has a lower cost based on the link metrics $\boldsymbol{v}_r$:

$$\boldsymbol{q}_r = \boldsymbol{q}_r^{new}, \text{if } \boldsymbol{v}_r^T \boldsymbol{q}_r > \boldsymbol{v}_r^T \boldsymbol{q}_r^{new}. \tag{25}$$

Then the spare provision matrix $\boldsymbol{G}^{[A]}$ and the spare capacity vector $\boldsymbol{s}$ are updated to reflect this change accordingly.

After Step 5, SSR continues to Step 2 to compute or update the backup path of another flow. This iteration process keeps improving the total cost of the spare capacity. It stops when no backup path update in all flows improves the cost. Additional termination conditions after Step 5 use a maximum number of total backup updates or a maximum execution time to terminate the algorithm.

Notice that the above steps are modified from the original ones to use the overlay information matrix $\boldsymbol{H}$ to translate SPM information between layers.

TABLE III
RESULTS OF TWO TOP-LAYER SPARE CAPACITY ALLOCATION (SCA)

net	Top Layer Only								Time (second)			
	N	L	R	W	$S^{[A]}$		$S^{[B]}$		$S^{[A]}$		$S^{[B]}$	
					BB	SSR	BB	SSR	BB	SSR	BB	SSR
Full mesh top layer												
0†	4	6	6	9	11	11-13	11	11	0.01	0.73	0.01	0.75
1	6	15	15	27	15	16-22	15	16-22	0.07	0.81	0.08	1.00
2	7	21	21	39	28	30-37	28	29-35	4.09	0.88	0.54	1.34
3	8	28	28	63	39	40-53	37	40-49	29.37	0.98	1.09	1.94
4	10	45	45	106	48	56-68	48	54-68	88.04	1.67	11.98	5.36
5	10	45	45	134	121	129-141	118	124-131	56.53	1.53	64.49	4.26
6	10	45	45	157	121	132-147	117	125-135	1,145	1.69	290	7.23
7	8	28	28	116	103	103-130	102	105-111	1.07	1.11	3.54	2.66
8	12	66	66	309	217	230-251	-	221-253	2,353	4.48	5days	40.19
Partial mesh top layer												
1	6	9	15	34	32	32-35	31	32-33	0.03	0.81	0.04	0.95
2	7	12	21	54	45	46-50	45	46-50	0.06	0.86	0.07	1.17
3	8	14	28	72	49	49-57	48	49-57	0.1	0.89	0.17	1.36
4	10	16	45	136	97	98-103	97	98-104	0.34	1.16	0.29	2.34
5	10	18	45	157	114	115-125	114	115-123	0.55	1.09	4.29	2.44
6	10	22	45	184	162	164-172	160	161-170	1.12	1.19	0.71	3.19
7	8	13	28	126	100	100-108	100	100-111	0.34	0.97	1.37	1.89
8	12	24	66	389	320	323-339	320	323-338	1.9	2.08	2.61	13.25

†: Network 0 is shown in Fig. 1.

D. Restoration at both layers

When both layers use shared backup path protection, sharing the spare capacity across layers might further reduce the redundancy. This concept is also called *common pool survivability* in [10]. When both layers are resilient to the same set of bottom-layer failure scenarios, the spare capacity on both layers can be shared if their spare provision matrices are exchanged. The top-layer spare provision matrix $\boldsymbol{G}^{[A]}$, used in both Model A and B, is transformed and merged with the spare provision matrix $\boldsymbol{G}^b$ at the bottom layer

$$\boldsymbol{G}^{[C]} = \boldsymbol{G}^b + \boldsymbol{H}^T \boldsymbol{G}^{[A]}. \tag{26}$$

The objective function for the SCA problem is modified as shown in (27). Notice that this objective value $S^{[C]}$ is less than the total spare capacity used when both layers have the shared path protection individually, i.e. $S^b + S^{[B]}$.

$$\begin{aligned} \min_{\boldsymbol{Q},\boldsymbol{Q}^b} S^{[C]} &= \boldsymbol{e}^T \max \boldsymbol{G}^{[C]} = \boldsymbol{e}^T \max(\boldsymbol{G}^b + \boldsymbol{H}^T \boldsymbol{G}^{[A]}) \\ &\le \boldsymbol{e}^T \max \boldsymbol{G}^b + \boldsymbol{e}^T \max(\boldsymbol{H}^T \boldsymbol{G}^{[A]}) \\ &= S^b + S^{[B]} \end{aligned} \tag{27}$$

In the SSR algorithm, both layers perform their single layer SSR algorithms but using different link routing metrics. The top layer uses $\boldsymbol{H}$ in the modified SSR algorithm in §III-C. It needs an additional change in Step 3: The $\max(x)$ operation to compute the spare capacity vector $\boldsymbol{s}^b$ in (21) and (22) should include the bottom-layer SPM matrix $\boldsymbol{G}^b$, i.e. $\max(\boldsymbol{G}^b + x)$. In this way, both layers share a common spare capacity provision matrix $\boldsymbol{G}^{[C]}$. In effect the layers cooperate to further improve spare capacity sharing.

E. Failure-dependent path restoration at the top layer

All of the models above assume FID path restoration. The SCA problem for the failure-dependent (FD) path restoration has been discussed in [17], [18]. It allows multiple backup routes to protect the same working route. Each of these backup routes will protect one or more failure scenarios. Previously in [19] and [20, Ch. 6], the FD path restoration has been modeled in a matrix format. This section extends this matrix model for FD path restoration to two-layer networks.

The arbitrary bottom-layer failures are captured in the failure matrix $\boldsymbol{F}$ in (13). In order to compute backup paths for any individual failures, equation (15) that finds the spare provision matrix $\boldsymbol{G}^{[A]}$ is replaced by $\boldsymbol{G}^{[D]}$, where

$$\boldsymbol{G}_k^{[D]} = \boldsymbol{Q}^{k^T} \boldsymbol{M} \boldsymbol{U}_k, \quad 1 \le k \le K. \tag{28}$$

The k-th column vector $\boldsymbol{G}_k^{[D]} = \{g_{lk}\}_{L\times 1}$ in $\boldsymbol{G}^{[D]}$ is determined by the k-th column vector $\boldsymbol{U}_k = \{u_{rk}\}_{R\times 1}$ of the failure matrix $\boldsymbol{U}$ in (14), the demand matrix $\boldsymbol{M}$, and the backup path matrix $\boldsymbol{Q}^k$. The backup path matrix $\boldsymbol{Q}^k$ indicates backup paths for all flows upon failure k. In this way, each flow could have more than one backup routes, one for each failure cases its working path might encounter. Note, the FD path restoration approach has many more backup path design variables than FID.

F. Stub Release

When a working path is disconnected due to a failure, some of its links might still reserve bandwidth which do not carry traffic anymore. Releasing this bandwidth to be reused by other backup paths is termed *stub release* and could further improve bandwidth efficiency. The drawback of this process is the extra overhead involved in determining where the stubs are and signaling to release the capacity. Here we consider stub release for the FD path restoration approach.

A new backup path matrix $\bar{\boldsymbol{Q}}^k$ is defined by its elements as

$$\bar{q}_{rl}^k = \begin{cases} q_{rl}^k - 1, & p_{rl} = 1, u_{rk} = 1, f_{kl} \ne 1 \\ q_{rl}^k, & otherwise. \end{cases} \tag{29}$$

TABLE IV
RESULTS OF BOTTOM-LAYER SPARE CAPACITY ALLOCATION (SCA)

net	N^b	L^b	R^b	W^b	S^b		Time (Second)	
					BB[1]	SSR[2]	BB	SSR
0	5	7	10	13	11	11	0.01	0.78
1	10	22	45	71	23	26-31	36	1.2
2	12	25	66	112	51	53-58	14	1.5
3	13	23	78	162	67	71-77	43	1.7
4	17	31	136	320	124	126-137	2.8hr	3.4
5	18	27	153	413	242	243-252	10.46	3.0
6	23	33	253	835	563	569-582	52.07	5.5
7	26	30	325	1366	901	920-931	5.78	6.5
8	50	82	1225	5552	-	2863-2896	-	156

Note: 1) Branch and Bound (BB) results are from AMPL/CPLEX v9.10 on a Sun Fire V240 Server with 1GHz CPU and 2GByte memory.
2) SSR results are obtained on a Intel Pentium 4 1.5GHz CPU with 1GByte memory. These results are shown as the range of 64 random cases in the format of "min-max".

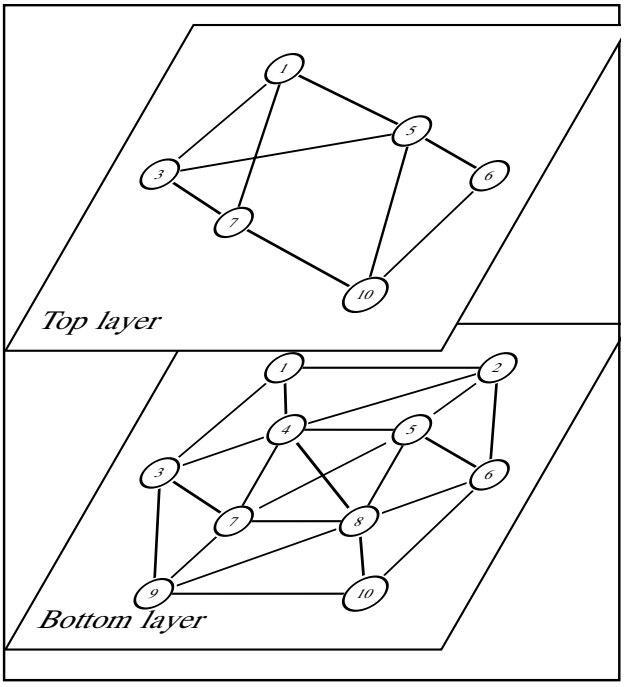

Fig. 2. Net 1 ($N = 6, L = 9, N^b = 10, L^b = 22$)

In this equation, the stub release function is represented by adding "-1" to the appropriate positions in the backup paths matrix $\boldsymbol{Q}^k$. The working capacity of flow r on link l will be released only when its working path uses this link ($p_{rl} = 1$); failure scenario k disconnects its working path ($u_{rk} = 1$); and failure k does not affect this link ($f_{kl} \neq 1$). The matrix $\bar{\boldsymbol{Q}}^k$ replaces $\boldsymbol{Q}^k$ to find $\boldsymbol{G}$ in (28). Since the working capacity released by the stub release function depends only on failures, the SSR algorithm is unchanged. This allows stub release to be implemented as a pluggable option in the SSR algorithm.

IV. NUMERICAL RESULTS

Numerical studies of the proposed SCA models and their associated SSR algorithms were conducted for a variety of scenarios on a set of nine different networks. The top and bottom layer topologies of the networks studied are shown in Figs. 1–9. The number of links, nodes and traffic flows in the two layers are summarized in Table III and IV respectively.

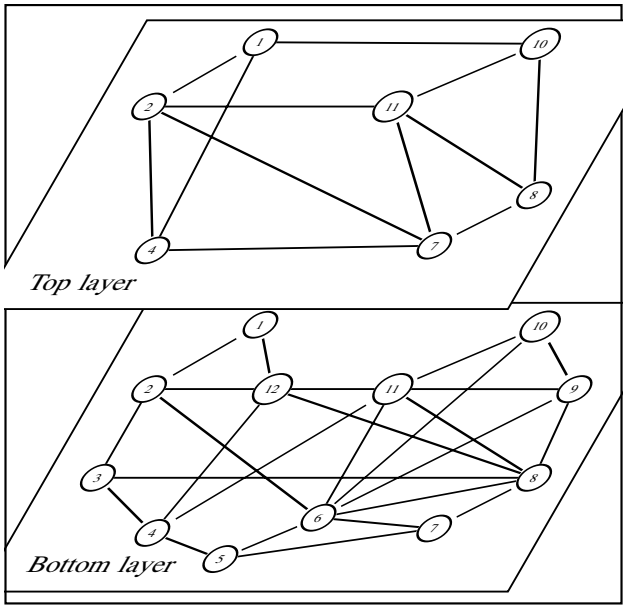

Fig. 3. Net 2 ($N = 7, L = 12, N^b = 12, L^b = 25$)

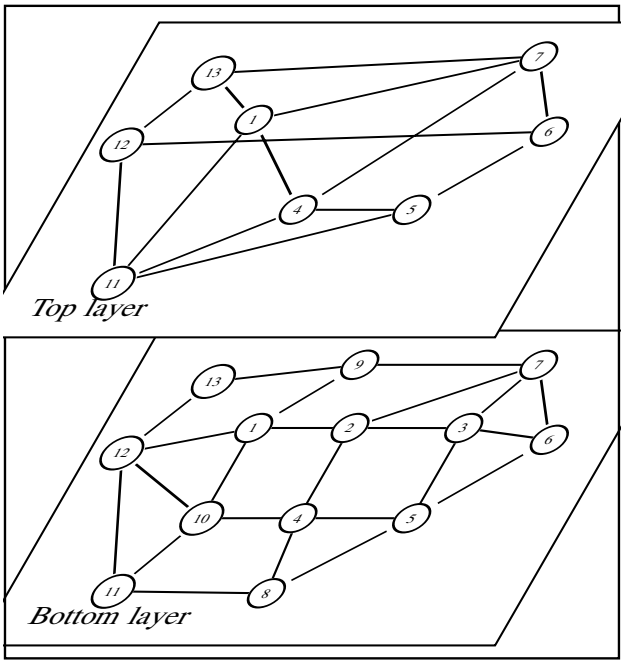

Fig. 4. Net 3 ($N = 8, L = 14, N^b = 13, L^b = 23$)

Two top layer topology cases were studied: full mesh and partial mesh. In the full mesh case, all top layer nodes are directly connected to all other top layer nodes. In the partial mesh case, the top layer has a sparser interconnection with the topology given in Figs. 1–9.

The traffic flows on both layers were a full mesh, each with unit traffic demand. The assumption of full-meshed flows exposes any possible spare capacity sharing opportunities among all node pairs. The unit flow bandwidth is based on the previous studies in the literature. Furthermore, our earlier work on the single layer SCA [20] did not show a significant impact on the network redundancy by varying the traffic distribution. The demand matrix in the top layer is given as

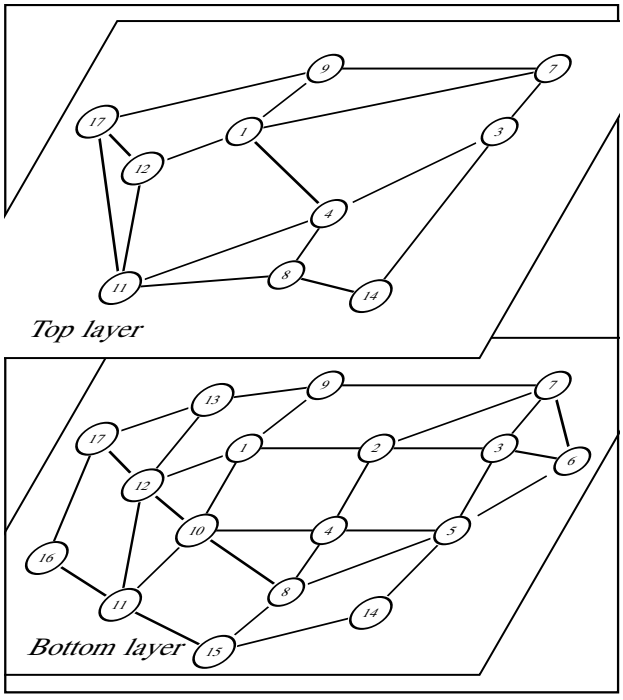

Fig. 5. Net 4 ($N = 10, L = 16, N^b = 17, L^b = 31$)

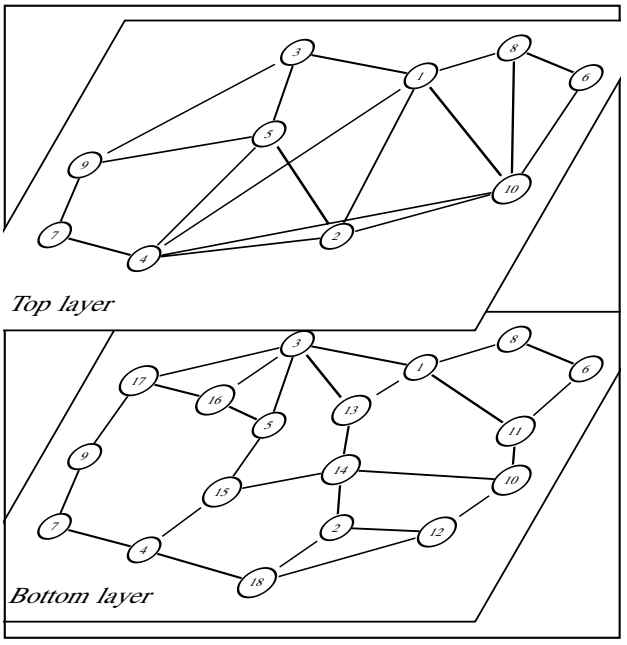

Fig. 6. Net 5 ($N = 10, L = 18, N^b = 18, L^b = 27$)

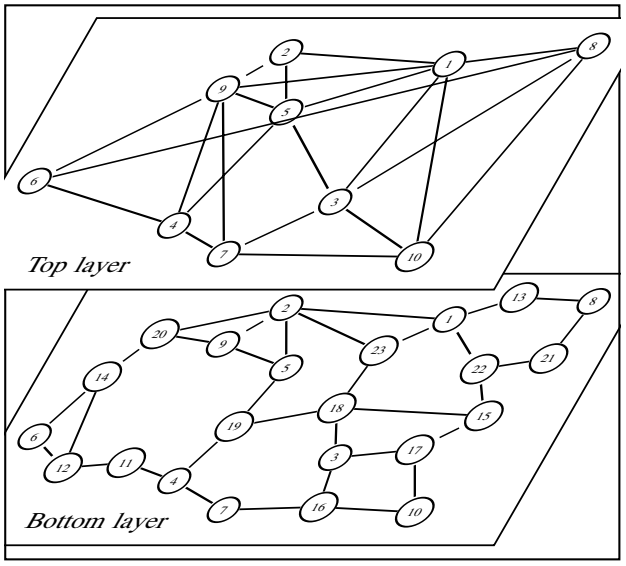

Fig. 7. Net 6 ($N = 10, L = 22, N^b = 23, L^b = 33$)

$\boldsymbol{M} = \boldsymbol{I}_{R\times R}$, where R is the number of flows. In this case, $R = N(N-1)/2$ for a full mesh of flows, with $|N|$ denoting the number of nodes. The demands in the bottom layer have a similar formulation with superscript ^{b} on their variables. The working paths are the shortest hop paths in both layers. These working paths have at least one failure-disjoint backup path.

The total working capacity in the bottom layer reserved by the top layer working paths is $W = \boldsymbol{e}^T\boldsymbol{P}\boldsymbol{H}\boldsymbol{e}$, where $\boldsymbol{P}$ is the working path matrix in the top layer and $\boldsymbol{H}$ is the interlayer link incidence matrix. The failure scenarios considered were all single bottom layer link failures. Hence, the bottom layer failure matrix $\boldsymbol{F}^b = \boldsymbol{I}_{L^b\times L^b}$, where L^b is the number of the bottom-layer links.

The following SCA models were studied.

- The SCA model in the bottom layer protection as discussed in §III-A.
- The FID path restoration case for both Models A and B for top layer protection in §III-B.
- The SCA model for common pool protection at both layers in Model C in §III-D.
- The FD path restoration at the top layer scheme in section III-E.
- The FD path restoration at the top layer with the stub release feature in §III-F.
- Finally, the above SCA models protecting any single bottom layer node failure were studied.

The numerical results for the SCA models are given in Tables III–VI . Two algorithms, branch and bound (BB) and SSR, were used to find the solutions to the SCA models. Since the SSR solution depends in part on the ordering of the flows, a range of solutions from 64 random cases are listed with the format of the minimum and the maximum results between a hyphen "-". The execution time results reported for the SSR algorithm are the total time to find the entire 64 solutions. The BB results are obtained from the commercial software CPLEX which could find the optimal solution. Note, that in the Tables, some cases for Network 8 are marked with a '-' in the BB column indicating that no results were obtained for during the execution time limit of 5 days.

The results for the bottom layer SCA are provided in Table IV. Note, that the bottom layer results are the same regardless of the partial/full mesh layout of the top layer. Notice, that the SSR results are very close to the BB results while typically having a faster execution time.

The spare capacity required for the top layer restoration approach ($S^{[A]}$ or $S^{[B]}$), and the SCA solution time are provided in Table III. Again one can see that the SSR algorithm provides near optimal results with lower execution speeds.

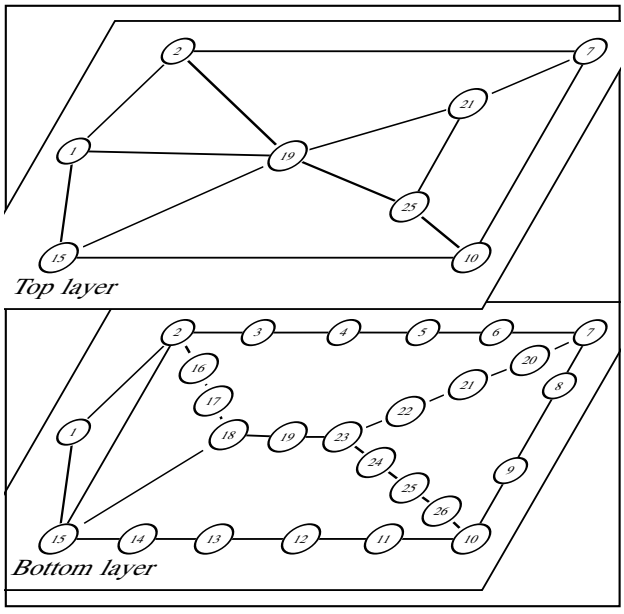

Fig. 8. Net 7 ($N = 8, L = 13, N^b = 26, L^b = 30$)

TABLE V
COMPARISON OF $S^{[A]}$ FOR PATH RESTORATION SCHEMES

Net	FID	FD	FDStubR
Link Failure			
1	32– 35	24– 28	23– 28
2	46– 50	32– 39	28– 35
3	49– 57	40– 49	36– 40
4	98–103	75– 85	71– 78
5	115–125	104–118	103–115
7	100–108	78–108	75–101
Node Failure			
1	23– 25	22– 26	20– 25
2	60– 64	39– 43	35– 42
3	37– 44	30– 38	28– 37
4	124–125	90– 98	76– 86
5	106–115	97–109	94–107
7	102–115	82–103	76– 96

In Table VI, the results for the SCA problem for common pool protection at both layers in Model C is provided. The second column, $W^b + W$, provides the total working capacity used by working paths on both layers in the network. The third column, $S^b + S^{[B]}$, list the best total spare capacity when both layers find their SCA solution using Model B individually. In the next two columns, the results of the total spare capacity, $S^{[C]}$, for common pool protection using Model C are provided. The solution time values are listed in the next two columns.

Table V presents the results for the FD and FD with stub release cases for top layer path restoration using SCA Model A. Note, only SSR results are given here.

Considering all the numerical results several interesting observations can be made as follows.

A. SSR versus branch and bound solutions

1) SSR finds a *near* optimal solution for all cases. Among all numerical results, the minimum spare capacity found by SSR is typically near or the same as the optimal solution found by BB. Most results have gaps smaller than 5% .
2) SSR scales well in comparison to BB. Among all numerical cases, SSR is typically much faster than BB and is always able to find a solution well within the maximum execution time limit (5 days). In contrast for large networks (e.g., Network 8), the BB algorithm takes a great amount of solution time and in some cases is unable to find a solution within the maximum time limit.

Hence, it might be a good compromise to use SSR as a fast algorithm to find a solution for large networks.

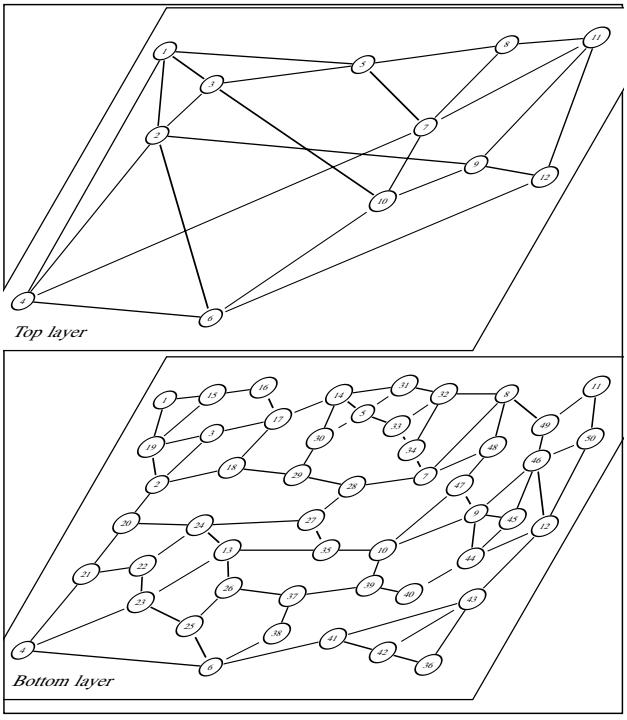

Fig. 9. Net 8 ($N = 12, L = 24, N^b = 50, L^b = 82$)

B. Top layer cross-layer spare capacity reservation

From the results of Table III comparing Model B with Model A one can see the cross layer spare capacity reservation in Model B has very small gain in the bandwidth. Comparing the best results in the total spare capacity between Model A and B for top-layer protection, the results in the $S^{[B]}$ column are about 1%, on average, less than those in the $S^{[A]}$ column. These results indicate that Model B is suitable for an off-line traffic engineering to further improve bandwidth efficiency as the small benefits might not justify an online implementation in the GMPLS control plane. Since Model B requires the top-layer backup route to be able to reserve the bottom-layer link spare capacity. This cross layer reservation is expensive in terms of the information exchanged between two neighboring layers.

C. Common pool protection vs. single layer protection on both layers

Comparing the results of the common pool approach versus having separate independent protection schemes on each layer

TABLE VI
RESULTS OF COMBINING TWO SINGLE-LAYER SCA AND THE COMMON POOL APPROACHES

net	$W^b + W$	Separate	Common pool			
		$S^b + S^{[B]}$	$S^{[C]}$		Time $S^{[C]}$	
		Best[1]	BB	SSR	BB	SSR
Full mesh top layer						
0	22	22	19	19-19	0.02	1.49
1	98	38	34	37-43	118	2.09
2	151	79	73	77-84	9.95	2.70
3	225	104	97	101-110	18.91	3.48
4	426	172	157	167-182	4.8day	8.52
5	547	360	336	349-362	212.22	6.74
6	992	680	653	672-689	295.4	12.36
7	1482	1004	978	982-992	15.32	8.70
8	5861	3084	-	3033-3077	-	182.28
Partial mesh top layer						
1	105	54	45	47-52	49.33	3.19
2	166	96	82	85-90	3.17	3.55
3	234	115	108	111-119	88.89	4.14
4	456	221	192	201-213	1,115	6.55
5	570	356	338	352-360	18.16	6.78
6	1019	723	657	678-695	256.9	9.64
7	1492	1001	975	975-990	5.9	9.7
8	5941	3183	-	3045-3078	-	156.61

Note: 1) Best solutions from BB and SSR results in previous Table III are listed here. In most cases, BB provides the best solution. However, when BB cannot find feasible solution, SSR results are used as the best solution here.

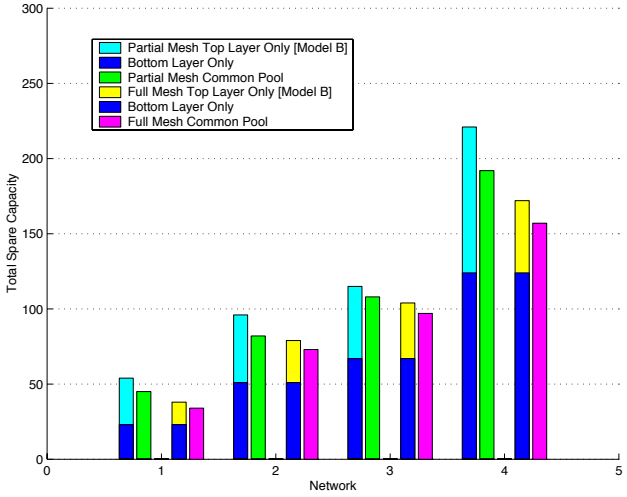

Fig. 10. Total Spare Capacity on two-layer networks on Network 1-4

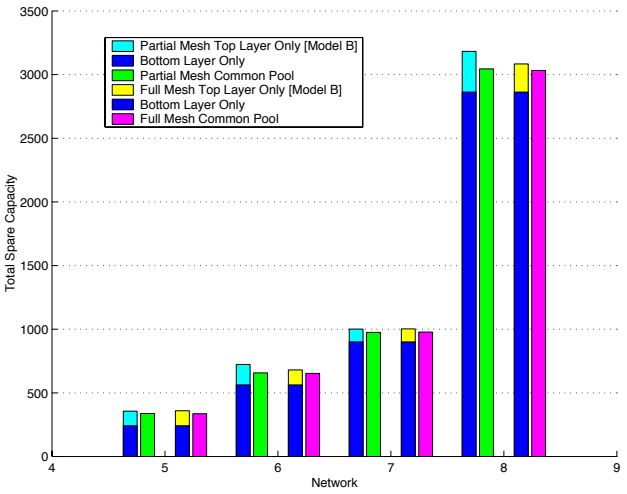

Fig. 11. Total Spare Capacity on two-layer networks on Network 5-8

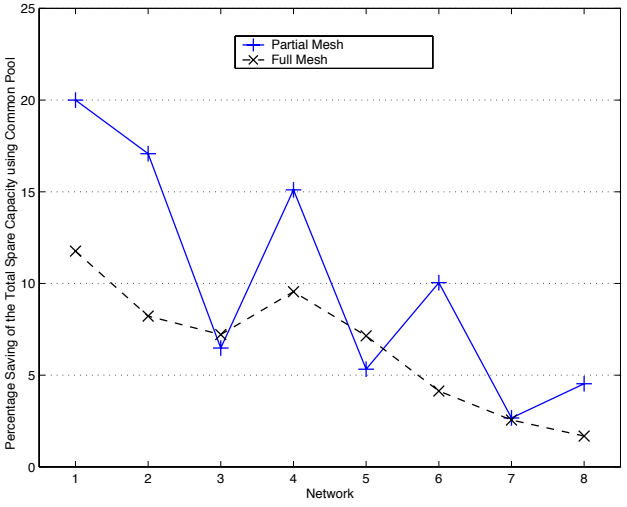

Fig. 12. Percentage Saving of the Total Spare Capacity using SCA on both layers

one can see some clear differences in the results. Drawing on the data from Table III and Table VI one can construct Fig. 10 and 11 to compare the two approaches. In the Figures the total spare capacity results on eight networks are plotted as bar charts. The network numbers are used as the X axis. Each network has two groups of bars.

- The first group contains two bars to the left of the network number on the X axis. They are results from the partial mesh top-layer topology.
- The second group of two bars are to the right of the network number. They are results from the full mesh top-layer topology.
- Within each group of bars, the leftmost bar is split into two stacked components showing the amount of spare capacity reserved by the two independent layers. The bottom bar indicates the value of the total spare capacity in the bottom-layer S^b. The top bar indicates the total spare capacity reserved in the bottom layer for the top-layer protection using Model B, i.e. $S^{[B]}$.
- Also, within a group of bars, the rightmost bar indicates the total spare capacity in both layers using the common pool protection, i.e. $S^{[C]}$.

One can clearly see that common pool protection uses less spare capacity than the total of the two single layer protection schemes. The percentage savings in the total spare capacity using common pool protection can be computed as

$$\delta = \frac{S^b + S^{[B]} - S^{[C]}}{S^{[C]}}. \tag{30}$$

Fig. 12 plots the percentage savings for the eight networks for both the full mesh and partial mesh cases. The figure shows that the common pool approach can yield a total spare capacity reduction in the range of 2–20%.

D. Network redundancy and scalability

Fig. 13 plots the network redundancy values (S/W) achieved using the various SCA models. The group of bars to

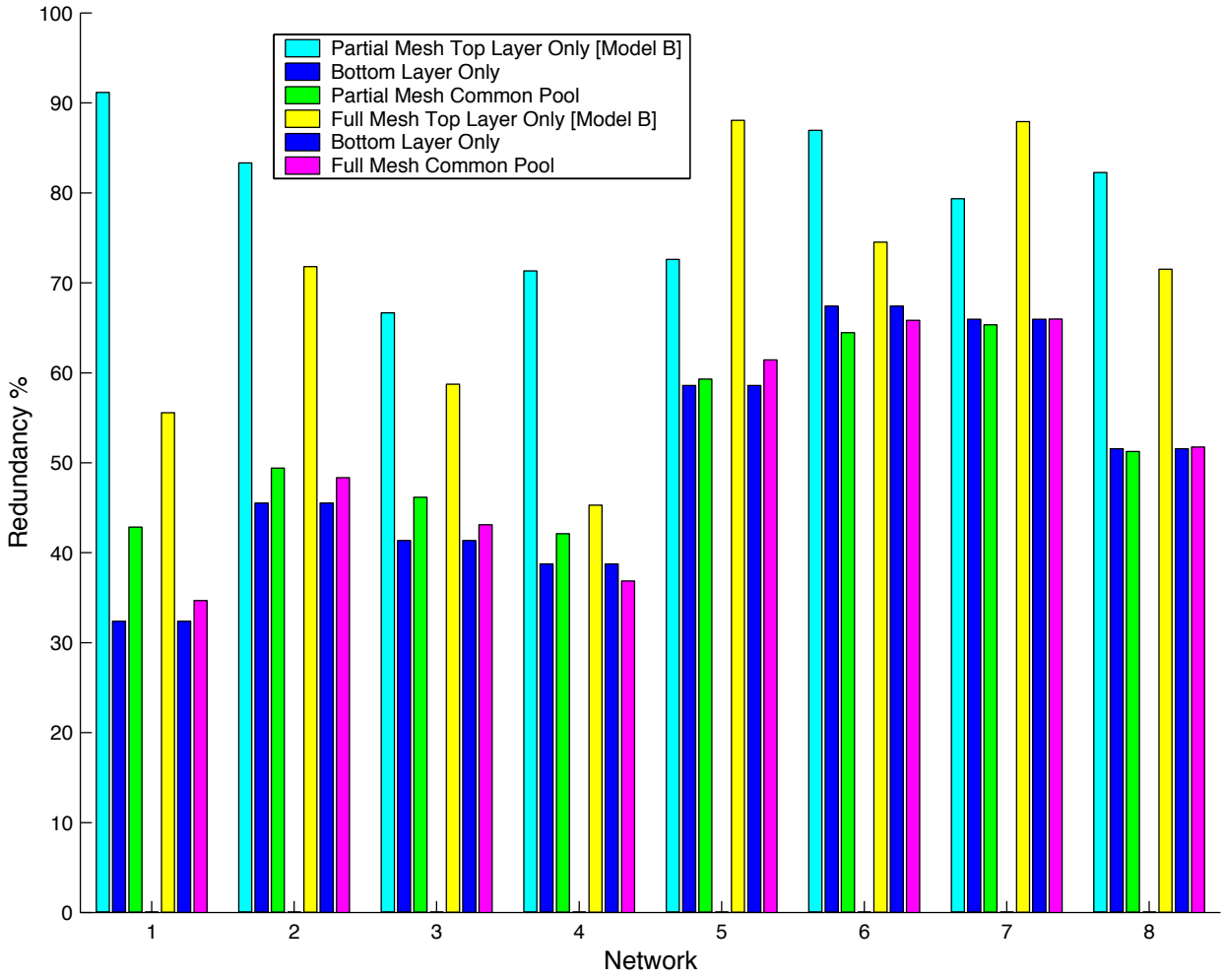

Fig. 13. Redundancy on two-layer networks

the left of a network number are for the partial mesh top-layer topology while the bars to the right are for the full mesh case. Notice that within a group, the redundancy bars for the top layer and the bottom layer are plotted stand-alone instead of stacked together. This is because the two layers have different working capacity values, thus, their redundancy values cannot be added together.

The first observation is that the network redundancy values at the top layer are always larger than those at the bottom layer. One explanation for this comes from the fact that less traffic flow in the top-layer protection produces less chance to share spare capacity on their backup routes. The other explanation is that the top-layer traffic have to be routed over top-layer links, then mapped down to the bottom layer. This mapping reduces the chance of spare capacity sharing, and consequently increases the network redundancy. The second observation is that the common pool protection scheme has redundancy values similar to bottom layer only protection. This is important for the scalability of common pool protection on multi-layer networks.

E. FD path restoration schemes and stub release

In Table V, we use Model A and the SSR algorithm to compare various path restoration schemes, i.e. FID, FD, and FD with Stub Release (FDStubR), to protect either single link or single node failures. Model B is not used in this comparison because of its requirement of a cross-layer capacity reservation protocol and the small gain in the total spare capacity observed in the previous results.

The numerical results using partial mesh top layer on six networks are shown in Table V. The best results in all cases are normalized by their FID results on the same network and plotted in Fig. 14 for link protection and Fig. 15 for node protection. In these two figures, the network number is shown in the X axis. The normalized total spare capacity percentage values, for FID, FD, and FDStubR, are drawn as three bars above the network number. The values of FID are always equal to 100% since FID values are used as the base in the normalization. One can see that FD path restoration has about 10–30% lower total spare capacity values than those of FID using link protection. This indicates that multiple backup paths in FD path protection could increase the chance of spare capacity sharing, and consequently, reduce network redundancy. Using FD path restoration scheme with stub release (FDStubR), the total spare capacity can be further reduced by 1–10% from those in FD. In node protection, similar to the link protection above, FDStubR could provide the lowest redundancy while FD is slightly higher than FDStubR and about 4–35% lower

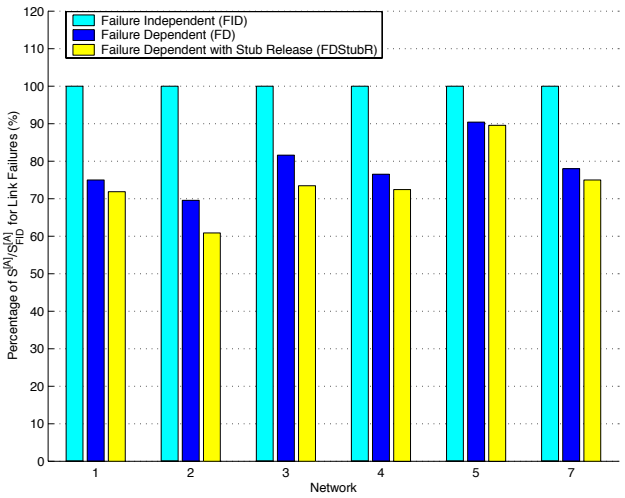

Fig. 14. Comparison of various path protection schemes for link failures in two-layer networks

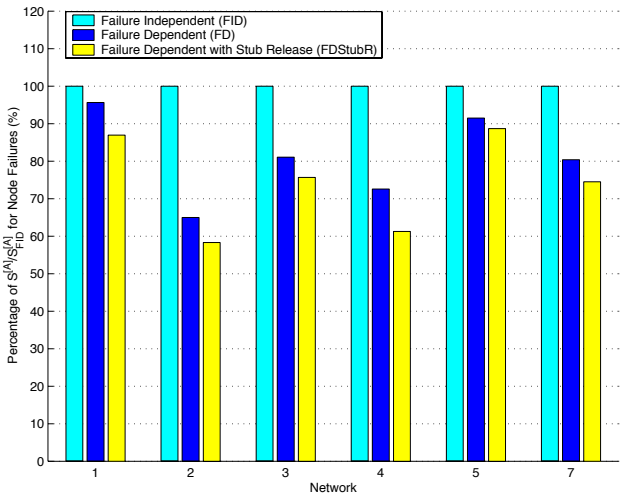

Fig. 15. Comparison of various path protection schemes for node failures in two-layer networks

than the FID results. It is important to be aware that the smaller spare capacity values in node protection does not necessarily indicate a better bandwidth efficiency than link protection. The smaller value could come from the dropped demands that cannot be recovered at the failure of their end nodes.

V. Conclusions

In this paper, several variations of the SCA problem for two-layer networks are formulated as ILP models. Specifically, we present SCA models for protection at the bottom layer only, protection at the top layer only, protection at the top layer only with cross layer sharing of spare capacity and common pool cross layer protection. The extension of these models to failure dependent path restoration with and without stub release are also presented. A fast routing based heuristic termed successive survivable routing (SSR) was proposed for solution of the SCA models. Numerical results show that the SSR algorithm can be used to efficiently find near optimal solutions.

Comparison numerical studies show that it might not be cost effective to reserve spare capacity across two layers using only a top layer restoration scheme as the bandwidth gains are small. In contrast when common pool protection is compared against using single layer protection schemes at both the top layer and bottom layer, major spare capacity savings are possible. The common pool protection scheme also maintains its network redundancy across increasing network size, indicating good scalability.

Appendix

The survivable topology layout problem is modeled in the following. As described in §III, the two-layer link incident matrix $\boldsymbol{H}$ should guarantee that any single bottom-layer link failure will not partition the top layer. Each pair of nodes at the top layer must have at least two paths which have disjoint bottom-layer links. This property is a special type of two-connectivity. It is called two-bottom-layer-link-connectivity and has been discussed by Modiano and Narula-Tam in [5]. In the following, a matrix model is given to extend their model to find the survivable top-layer topology that can be resilient to an arbitrary set of failure scenarios at the bottom layer. The basic formulation is given below.

$$\min_{\boldsymbol{H}} \quad \boldsymbol{e}^T \boldsymbol{H} \boldsymbol{e} \tag{31}$$

$$s.t. \quad \boldsymbol{H}{\boldsymbol{B}^b}^T = [\boldsymbol{B}^T | \boldsymbol{0}] \tag{32}$$

$$\boldsymbol{C}\boldsymbol{H}_f < \boldsymbol{C}\boldsymbol{e} \tag{33}$$

$$\boldsymbol{H}_f = \boldsymbol{H} \odot {\boldsymbol{F}^b}^T \tag{34}$$

The objective (31) is to minimize the total number of bottom-layer links used by all top-layer links. This objective can be easily extended to consider cost, bandwidth, etc.. The flow conservation constraint in (32) requires the top-layer link to map to a bottom-layer path. The zero matrix $\boldsymbol{0}$ in the right hand side has the dimensions of $L \times (N^b - N)$. It is used to fill up the matrix equation for the nodes that appear only at the bottom layer but not at the top layer. These nodes have indexes n, where $N < n \leq N^b$.

A topology is survivable if and only if its cut-set matrix of the top-layer topology $\boldsymbol{C}$ follows constraint (33), where the matrix $\boldsymbol{H}_f$ is calculated from the layout information matrix $\boldsymbol{H}$ and the bottom-layer failure matrix $\boldsymbol{F}^b$ in (34). The operator "$<$" in (33) means that each element in the vector on the right hand side is greater than any elements in the corresponding row of the matrix on the left hand side.

When the lower layer failure matrix $\boldsymbol{F}^b$ considers all single link failures, this model is equivalent to the model in [5]. The *cut-set matrix* $\boldsymbol{C} = (c_{ij})$ of a graph G has elements $c_{ij} = 1$ if the i-th cut-set of G contains the edge e_j, and $c_{ij} = 0$ otherwise. This matrix is defined in [21]. The number of rows in $\boldsymbol{C}$ is the total number of cut-sets in graph G, and equals to 2^{N-1}. Due to this exponential term, this model has exponential number of constraint (33) and it cannot scale with the number of top-layer nodes. A heuristic algorithm for determining C has been discussed in [5]. In this paper, we used the solution $\boldsymbol{H}$ of the above extended formulation solved by the branch and bound solver CPLEX.

For the case of node failure at the bottom layer, the constraint (34) has to be modified. A bottom node failure can disconnect the source or destination nodes of a top-layer link in the topology. To overcome this difficulty, we remove any cut-sets that contains one node in one of its partitioned subgraph from $\boldsymbol{C}$ in constraint (34). Meantime, the bottom-layer topology has to be 2-node-connected. For general failures at the bottom layer, this cut-set model may have difficulty in finding a unique interlayer matrix $\boldsymbol{H}$. This is a topic of the on-going research, such as the generalized failure-cut in [22].

REFERENCES

[1] J. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP and MPLS*, Morgan Kaufmann MKP, 2004.

[2] Wayne Grover, *Mesh-based Survivable Networks: Options and Strategies for Optical , MPLS, and ATM Networking*, Prentice Hall PTR, 2003.

[3] M. Pioro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan Kaufmann MKP, 2004.

[4] O. Crochat, J.-Y. Le Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 8, no. 3, pp. 384–395, June 2000.

[5] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," in *Proc. IEEE INFOCOM*, Apr. 2001.

[6] M. Kurant and P. Thiran, "On survivable routing of mesh topologies in IP-over-WDM networks," in *Proc. IEEE INFOCOM*, 2005.

[7] F. Ducatelle and L. M. Gambardella, "FastSurv: A new efficient local search algorithm for survivable routing in WDM networks," in *Proc. IEEE Global Communications Conference*, Dallas, TX, USA, Nov. 2004.

[8] F. Giroire, A. Nucci, N. Taft, and C. Diot, "Increasing the robustness of IP backbones in the absence of optical level protection," in *Proc. IEEE INFOCOM*, 2003.

[9] D. Xu, Y. Xiong, C. Qiao, and G. Li, "Failure protection in layered networks with shared risk link groups," *IEEE Network*, pp. 36–41, May 2004.

[10] P. Demeester and M. Gryseels, "Resilience in multilayer networks," *IEEE Commun. Mag.*, vol. 37, no. 8, pp. 70–76, 8 1999.

[11] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee, "Fault management in IP-Over-WDM networks: WDM protection versus IP protection," *IEEE J. Select. Areas Commun.*, vol. 20, no. 1, pp. 21–33, Jan. 2002.

[12] Q. Zheng and G. Mohan, "Multi-layer protection in IP-over-WDM networks with and with no backup lightpath sharing," *Computer Networks*, vol. 50, pp. 301–316, Jan. 2006.

[13] S. Koo, G. Sahin, and S. Subramaniam, "Cost efficient LSP protection in IP/MPLS-over-WDM overlay networks," in *Proc. IEEE International Conference on Communications*, June 2003, pp. 1278–1282.

[14] C. Ou, K. Zhu, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, "Traffic grooming for survivable WDM networks – shared protection," *IEEE J. Select. Areas Commun.*, vol. 21, no. 9, pp. 1367–1383, Nov. 2003.

[15] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," *IEEE/ACM Trans. Networking*, vol. 13, no. 1, pp. 198–211, Feb. 2005.

[16] Bernard Kolman, Robert C Busby, and Sharon Ross, *Discrete Mathematical Structures*, Prentice Hall, 1996.

[17] R. Iraschko, M. MacGregor, and W. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh survivable networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 3, pp. 325–336, June 1998.

[18] Y. Xiong and L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Trans. Networking*, vol. 7, no. 1, pp. 98–110, Feb. 1999.

[19] Y. Liu and D. Tipper, "Spare capacity allocation for non-linear cost and failure-dependent path restoration," Third International Workshop on Design of Reliable Communication Networks (DRCN), Budapest, Hungary, October 7–10 2001.

[20] Yu Liu, *Spare capacity allocation method, analysis and algorithm*, Ph.D. dissertation, School of Information Sciences, University of Pittsburgh, 2001, `http://www.sis.pitt.edu/~yliu/dissertation/`.

[21] L. R. Foulds, *Graph Theory Applications*, Universitext. Springer-Verlag, 1992.

[22] Andras Farago, "A graph theoretic model for complex network failure scenarios," in *The Eighth INFORMS Telecommunications Conference*, Mar. 2006.

Yu Liu (S'96-M'02) received the B.E. degree in information science and technology from Xi'An Jiaotong University, China, in 1993, the M.E. degree in communications and electronic systems from Tsinghua University, China, in 1996, and the Ph.D. in telecommunications from the University of Pittsburgh, Pittsburgh, PA, in 2001. He joined OPNET Technologies in August 2001 and is currently a senior software engineer in network design and optimization.

His research interests include network design, performance and analysis, mathematical programming, queueing theory, embedded systems, and distributed systems. At OPNET, he is developing automated design solutions for IP/MPLS networks, such as traffic engineering, fast reroute deployment, link dimensioning, topology design, multi-layer network design, and capital expenditure optimization in the IT Guru/SP Guru Network Planner products.

David Tipper (S'78-M'88-SM'95) received the B.S.E.E. degree from Virginia Tech, M.S.S.E and Ph.D.E.E. degrees from the University of Arizona. He is an Associate Professor in the Telecommunications Program with a secondary appointment in the Electrical Engineering Department at the University of Pittsburgh. Prior to joining Pitt in 1994, he was an Associate Professor in the Electrical and Computer Engineering Department at Clemson University. His current research interests are network design and traffic restoration procedures for survivable networks, infrastructure protection, network control techniques and performance analysis. He is the co-author of the textbook *The Physical Layer of Communication Systems* published by Artech House in 2006. Also, he is a co-editor of the book *Information Assurance: Survivability and Security in Networked Information Systems*, to be published by Elsevier/Morgan Kaufmann in 2007.

Korn Vajanapoom received the B.E. degree in Electrical Engineering from Chulalongkorn University, Thailand, in 1998, and the M.S. degree in Telecommunications from the University of Maryland, College Park, in 2002. He is currently working toward the Ph.D. degree in Telecommunications at the University of Pittsburgh, Pittsburgh, PA. From 1998 to 1999, he was an engineer at the Department of International Telephone, Communications Authority of Thailand (CAT). His research interests include incremental survivable network design, availability analysis in survivable networks, and multi-layer network survivability.