

# Protection with Multi-Segments (PROMISE) in Networks with Shared Risk Link Groups (SRLG)\*

Dahai Xu    Yizhi Xiong    Chunming Qiao

Department of Computer Science and Engineering, State University of New York at Buffalo  
Buffalo, NY 14260, {dahaixu, yxiong, qiao}@cse.buffalo.edu

## Abstract

Shared Risk Link Group (SRLG) has been widely recognized as an important concept in survivable optical networks. The issues of avoiding “traps” (to be defined later) in path determination and maximizing bandwidth sharing, are more challenging in providing shared SRLG protection than in providing shared path protection without considering SRLGs. In this paper, we proposed a novel survivability approach called PROtection using Multiple SEgments (PROMISE) to provide efficient SRLG protection, which can achieve a higher bandwidth efficiency and lower blocking probability at quick speed compared to previous schemes.

## I. INTRODUCTION

Dense Wavelength Division Multiplexing (DWDM) is a promising technology to accommodate the explosive growth of Internet and telecommunication traffic. Because of the large amount of traffic a fiber carries, a single failure will cause a severe service loss. Hence, survivability is a critical problem in network design.

Recently, the concept of SRLG has been proposed as the fundamental input for failure management in the Generalized Multi-Protocol Label Switching (GMPLS) control plane [1]. An SRLG is a group of network links that share a common physical resource (cable, conduit, node or substructure) whose failure will cause the failure of all links of the group [2, 3]. In general, the information on SRLGs are obtained manually by the network operator with the knowledge of the physical fiber plant of the network, although some SRLG auto-discovery schemes have been proposed [1, 4].

To protect a mission-critical client (e.g. IP, ATM) connection from any single SRLG failure, a straightforward solution is to choose a SRLG-disjoint paths of Active Path (**AP**) and Backup Path (**BP**) from a source (ingress) node to a destination (egress) node. That is, links along AP must not share any common SRLGs with links along the corresponding BP. Since a fiber can traverse several conduits, a network link may belong to several SRLGs, finding a pair of SRLG disjoint paths is more complicated than finding link/node-disjoint path (the latter may be considered as a special case as each link or the set of links incident upon each node could be a what we call a “trivial SRLG”). In fact, the former is a NP-complete problem, while the latter has a polynomial time solution [5–8].

In SRLG failure-independent protection schemes, backup bandwidth can be either shared or not shared. A (backup bandwidth) shared protection scheme is much more bandwidth efficient, and cost-effective. In shared SRLG protection, two or more BPs can share backup bandwidth (called BBW) as long as their corresponding APs do not fail at the same time. The challenge is to find a pair of SRLG-disjoint paths for a given request such that the total bandwidth consumption is minimized. It requires more complex algorithms to determine routing and bandwidth requirement. Generally, providing end to end connections survivable to any single SRLG failure requires much more resources (e.g., bandwidth) usage and a longer recovery time than protection against any single link or node failure.

\*This research is supported by NSF Grant under the contract ANIR 0208331

To date, only a few Integer Linear Programming (**ILP**) models [5] and heuristic algorithms [6] have been proposed to find SRLG disjoint paths. Although it just takes seconds to obtain SRLG-disjoint paths with the ILP [5], this *basic* ILP model does not consider BBW sharing. On the other hand, the computational time of a *comprehensive* ILP solution with additional constraints to facilitate BBW sharing becomes prohibitive, especially for a large network size and/or a large number of SRLG's. A practical solution, which we call the Enhanced Two-Stage ILP hereafter, is to first determine the paths using the basic ILP without considering BBW sharing, and then adjust (reduce) the bandwidth to be allocated along the BP by taking advantage of BBW sharing. However, as to be shown in Sec. V, the BBW sharing potential is far from being fully explored.

Not surprisingly, heuristics such as Active Path First (**APF**), which finds an AP first, followed by an SRLG-disjoint BP, have always been attractive alternatives for its simplicity as well as ability to support policy-based routing and/or optimize the primary working path. The major problem in using the APF heuristic, however, is that once an AP is found, one may not be able to find an SRLG-disjoint BP (even though SRLG-disjoint path pair does exist). This is the so-called trap problem, which is rarely present when finding link/node-disjoint paths using APF [9], but can occur much more frequently (e.g., with probability of 10% to 30% in a typical network) in SRLG networks. To our best knowledge, no effective algorithms have been proposed so far to address the trap problem. In [10], we propose an APF-based heuristic that not only avoids the avoidable traps effectively, but also achieve near-optimize bandwidth efficiency. In this paper, we also adapt an APF heuristic, but attack the trap problem using a different approach.

Recently, we have proposed an innovative scheme called PROtection using Multiple SEgments (**PROMISE**) for shared protection against link/node failures (without SRLG) [11]. As illustrated in Fig. 1, the basic idea behind PROMISE is to divide an active path or AP (along which a survivable connection is established) into several possible overlapping<sup>1</sup> active segments or AS's, and then protect each AS with a detour called backup segment or BS (instead of protecting the AP as a whole as in path protection schemes).

In this paper, we will apply the new PROMISE scheme for shared protection design in SRLG networks. It can achieve superior bandwidth efficiency with moderate computation complexity, which is still polynomial time. At the same time, it can deal with traps effectively due to its flexibility in choosing the AS's and BS's. The rest of the paper is organized as follows. We first presents some definitions of trap problems and an overview of the proposed PROMISE scheme in Sec. II. Sec. III describes closely related prior work. Sec. IV presents our proposed PROMISE algorithm for shared protection in SRLG networks. Sec. V presents the performance evaluation model used as well as numerical results of the comparison between the proposed PROMISE algorithm and other existing approaches. Finally, Sec. VI summarizes our contributions.

## II. BACKGROUND

In this section, we first describe the trap problem in more detail, then provide an overview of the proposed PROMISE scheme and discuss its major benefits over existing link or path protection schemes.

### A. Trap Problem

In order to define the trap problem, we first discuss the optical network layer architecture. As shown in Fig. 2. A typical optical network can be classified into two layers: optical layer and physical layer [5, 6, 12]. The physical layer includes fiber spans (e.g. cable, conduit, et

<sup>1</sup>As to be discussed later, when a common link used by the AP belongs to two AS's, it is protected by only one BS.

al) and fiber span nodes. And the optical layer consists of optical links and nodes (a subset of the nodes in the physical layer) where Optical Cross-connects (OXC) or Optical Add/Drop Multiplexer (OADM) locate. An optical link in the optical layer is a connection in the physical layer, which may traverse over several fiber spans and/or fiber span nodes. At the same time, several optical link may pass through the same fiber span. Therefore, a single failure in the physical layer can cause multiple optical link failures. If we treat each fiber span as a SRLG, then a single SRLG failure may result in multiple optical link failures.

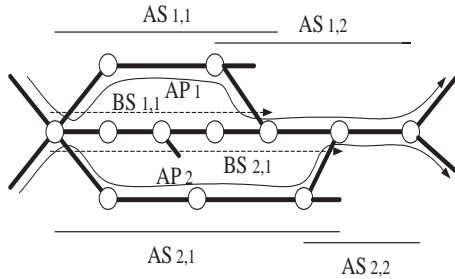


Fig. 1 Illustration of PROMISE

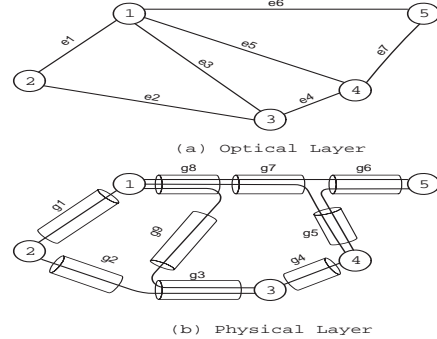


Fig. 2. Layered Architecture of Optical Network

We define the following for SRLG protection. If we replace the word ‘‘SRLG’’ with ‘‘link’’ or ‘‘node’’, the definitions below are also applicable to link or node protection. If an algorithm fails to find a pair of SRLG-disjoint paths for a given source and destination node pair, we say that the algorithm falls into a *trap*. From this definition, traps can be classified into real traps and avoidable traps.

**Real Traps:** It is connectivity-induced (where connectivity refers to topology and link bandwidth or capacity). In other words, there does not exist any SRLG-disjoint paths between a given source and destination node pair. Thus, a real trap cannot be avoided by *any algorithms*.

**Avoidable Traps:** These are traps to a given algorithm but not *real traps*. Then, avoidable traps to a given algorithm are capable to be avoided by other more elaborate algorithms, e.g., Integer Linear Programming.

Clearly, it is desirable to find an appropriate algorithm that falls into no or only few such *avoidable traps*. How to effectively avoid (the avoidable) traps for SRLG protection using heuristics is still an open issue. As to be discussed in Sec. III, only a few heuristics have been proposed [13, 14], and none of them can significantly eliminate the possibility of falling into an avoidable trap.

### B. Overview and Major Benefit of PROMISE

As mentioned earlier, the basic idea of PROMISE is to provide protection for one *active segment* (or AS) at a time using a detour called *backup segment* (or BS)<sup>2</sup>. The BS starts and ends at the same two nodes as the corresponding AS, but is otherwise node-disjoint with the AS, *not just link-disjoint*.

A **valid** AS set (or Active Segmentation) for a given AP need follow two constraints:

- Every link along the AP belongs to at least one AS and at most two AS’s.
- An AS can not be a proper subset of any other AS.

In addition, each link belonging to two overlapping AS’s needs to be protected by only one BS corresponding to one of the two AS’s. In an SRLG network, each BS just needs to be SRLG-disjoint with (and have enough backup bandwidth) its protected active links (instead of all the links of its corresponding AS).

<sup>2</sup>hereafter, a segment refers to a channel on one or more consecutive links.

It is worth noting that the proposed PROMISE approach is more than just a trivial combination (or compromise) of link and path protection. More specifically, due to its inherent flexibility in partitioning an AP and protecting it with multiple BS's, PROMISE enjoys the following two major advantages: (1) more bandwidth efficient; more importantly, and (2) can succeed in satisfying a request for a survivable connection, especially requests with certain constraints on the lengths of the detours, and efficiently deal with either real (i.e., network connectivity-induced) or avoidable (algorithm-induced) traps.

In addition, the proposed PROMISE approach can readily be applied to MPLS networks by extending the existing protocols for local repair/recovery in MPLS networks [15].

### III. RELATED WORK

In this section, we review several existed shared SRLG protection schemes. In addition, we will briefly describe some relevant multiple segments protection schemes with a focus on their suitability in SRLG networks.

#### A. Shared SRLG Protection

Existing approaches for shared SRLG protection can be mainly classified into two types: deterministic [14] and stochastic [13]. In [14], an APF-based deterministic approach called survivable routing (SR), (as well as its variation called successive SR or SSR), was proposed. It is similar to ordinary APF except that it can provide shared SRLG protection (by treating each SRLG as a so-called "failure" in SR). In [13], a stochastic approach was proposed to determine the SRLG-disjoint paths for shared SRLG protection. It is similar to Simple APF with the major difference being in how the cost of each link is assigned before BP is determined using a shortest-path algorithm. More specifically, while ordinary APF assigns, say 1 unit (e.g. a channel) to each link that can possibly be used by the BP, this algorithm assigns  $p < 1$  units, where  $p$  is the probability that BBW sharing cannot occur on the link. Its main advantage is that it is simple to implement, but as a price paid, its bandwidth efficiency is not as good as its deterministic counterparts. However, none of these approaches addressed the trap problem.

#### B. Multiple Segments Protection

Note that, several other segmented protection approaches have been proposed [16, 17]. In [16, 17], an APF-based heuristic algorithm was proposed, which cannot avoid the trap problem efficiently and in addition, like the Enhanced Two-Stage ILP, does not consider BBW sharing until the paths are found. The scheme in [18] requires the node immediately upstream from the link/node failure to restore traffic along an alternate outgoing link, which limits its flexibility (and bandwidth efficiency), especially in SRLG networks. Finally, in [19], each segment of an AP is protected using pre-designed BLSR's, and exhaustive search algorithms (with backtracking) and rudimental heuristics were suggested with no performance results.

### IV. PROMISE IN SRLG NETWORKS

In this section, we will show in detail how PROMISE scheme works. More specifically, we will describe efficient algorithms for finding an AP, determining its partition into AS's, and their corresponding BS's for a given request in the on-line case.

As expected, joint optimization of AP and its protecting BS's in PROMISE is much more challenging than similar tasks in path and link protection especially because PROMISE allows AS's (and their corresponding BS's) to overlap (and "criss-cross") one another, thus making it very difficult to formulate the flow constraints. To simplify the task of determining AP and its protecting BS's, we propose to find an AP first, followed by the set of BS's to protect the AP (the set of BS's uniquely determines how the AP is partitioned into a set of AS's).

### A. Find the Backup Segments

We start with the case with complete information on the existing connections and link status, in addition to the information on the set of all SRLG's, denoted by  $\mathcal{G}$ . More specifically, we assume that for every link  $e$ , the total amount of BBW allocated and the residual bandwidth available, denoted by  $B_e$  and  $R_e$ , respectively, are known ( $R_e$  is useful only in networks whose links have limited bandwidth as to described in more detail later). Moreover, for every SRLG  $g \in \mathcal{G}$ ,  $S_g^e$ , which denotes the total amount of bandwidth required by the set of connections whose AS's traverse any link in SRLG  $g$  and whose corresponding *protecting* BS's traverse link  $e$ , is also known.

Given the above complete (aggregate) information, we can calculate the additional amount of BBW needed to satisfy a new connection requiring  $w$  units of bandwidth as follows. Assume that the new AS (AP) uses at least one link belonging to SRLG  $g$ . Then, the minimum BBW needed on link  $e$  if link  $e$  is used by the new *protecting*, is  $BC_g^e = \max\{S_g^e + w - B_e, 0\}$ . If  $e$  also belongs to  $g$ ,  $BC_g^e$  should be infinity. Accordingly, let  $\mathcal{G}_a$  be the SRLG's to which link  $a$  belong, the minimum BBW needed on link  $e$  to protect link  $a$  is  $BC_a^e = \max_{g \in \mathcal{G}_a} BC_g^e$ . Finally, given the new AP, and  $\mathcal{G}_{AP}$ , which is the union of the SRLG's to which the links along the AP belong, the cost that will be assigned to link  $e$  is  $BC^e = \max_{g \in \mathcal{G}_{AP}} BC_g^e$ .

In the case of partial information (and distributed control) [20], a major difference is that a scalar  $P_{Ag} = \max_{\forall e} S_g^e$  for every SRLG  $g \in \mathcal{G}$ , then  $BC_g^e = \max\{\min(P_{Ag} + w - B_e, w), 0\}$ . Actual required BBW will be adjusted after path determination as in [20].

In the remainder of the section, we will describe how to determine the set of BS's using a simplified ILP or dynamic programming once a AP is found. Both approaches take into consideration the sharing of backup bandwidth among BS's for different APs (inter-sharing) as well as for the same AP (intra-sharing), and are capable of incorporating QoS constraints such as the limitation on the BS's hop count. In addition, both can be applied when either complete aggregated information on existing connections and link status is available, or only partial aggregated information is available [14, 20, 21].

The main difference between the ILP and dynamic programming approaches is that the former can select an "optimal" set of BS's for the given AP, but is feasible only for medium-to-large networks; On the other hand, the latter results in a polynomial time algorithm, but will still achieve *better* performance than shared SRLG path protection schemes using Enhanced Two-Stage ILP, as to be shown later.

### B. An Integer Linear Programming Approach

Even though an AP is given, the ILP formulation to find an optimal set of protecting BS's can be very challenging. In fact, the problem cannot be easily modeled and perhaps is harder than modeling the general multi-commodity flow problem because here, the source and destination for each BS and the number of BS are not known beforehand. We address the challenge by developing a unique **link-labeling** scheme that labels each link along the given AP such that there is a one-to-one mapping between how the links are labeled and how the AP is divided (and thus protected with BS's).

Fig. 3 shows two possible mappings for a 7-link AP. As can be seen from the example, the labels (integer numbers) are assigned to the links along the AP in an ascending order, by labeling the first link along the AP with 0<sup>3</sup>. Compared with the label of the link preceding a node, if the node does not start or end an AS, the label of the link following it will have the same label. Otherwise, if it (only) ends AS (except if it is destination node) or (only) starts at a node (except if it is source node), the label of the link following it will be increased by 1. Finally, if the node ends and starts two AS's respectively, the label of the link following it will be increased by 2.

<sup>3</sup>For the purpose of the discussion, the AP, as well as all its links and AS's, has a start and an end.

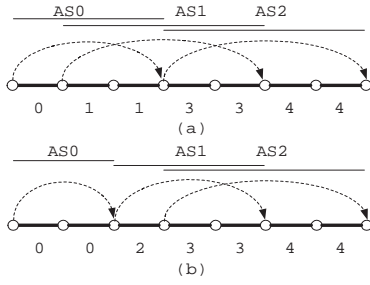


Fig. 3. Two ways to label links or divide an AP

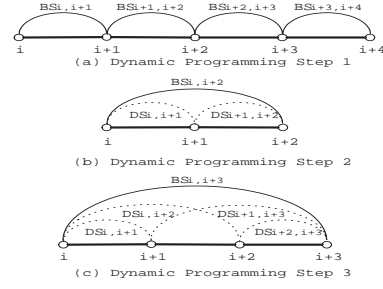


Fig. 4. Dynamic Programming Illustration

The basic idea of the ILP formulation is that if we can compare all *feasible* BS sets corresponding to given *partition* of the given AP, we can select the best BS set as a candidate solution. Then, for all possible *partitions* of the AP, the best candidate will be the optimal solution.

The following symbols are used in the ILP formulation:

- $H_a$ : number of hops along the given AP. This is also an upper bound on the number of BS's.
- $N_{AP}$ : set of nodes  $\{n\}$  along the AP, whose source and destination nodes are denoted by  $s$  and  $d$ , respectively.
- $\bar{K}$ : Specified maximal number of BS's to be used. We can specify an appropriate value less than  $H_a$  to obtain a sub-optimal result to save time. If  $\bar{K}$  is set to 1, the solution is reduced to providing shared path protection (with a node-disjoint path pair).
- $K^*$ : Current number of BS's to be used. We will run  $\bar{K}$  times with different  $K^*$  value, from 1 to  $\bar{K}$ . The best result of these  $\bar{K}$  sub-ILP problems will be used as the final solution.
- $F(n)$ : Set of links *from* (i.e. beginning with) node  $n$  that are not on the AP (i.e.  $\notin AP$ ).
- $T(n)$ : Set of links *to* (i.e. ending with) node  $n$  that are  $\notin AP$ .

Variables used in Integer Linear Programming

- $S_{k_e}$ : Should be 1 if link  $e$  is used by or protected by the  $BS_k$ , where the subscript  $k$  is from 0 to  $K^* - 1$ ; 0 otherwise.
- $S_{k_f}(n) (= \sum_{e \in F(n)} S_{k_e})$ : Should be 1 if  $BS_k$  is either from (i.e. begins at) node  $n$  for  $n \in N_{AP}$  or uses node  $n$  as an intermediate node for  $n \notin N_{AP}$ ; 0 otherwise.
- $S_{k_t}(n) (= \sum_{e \in T(n)} S_{k_e})$ : Should be 1 if  $BS_k$  goes to node  $n$  for  $n \in N_{AP}$  or uses node  $n$  as an intermediate node for  $n \notin N_{AP}$ ; 0 otherwise.
- $S_f(n) (= \sum_k S_{k_f}(n), n \in N_{AP})$  Should be 1 if node  $n$  is the beginning of any BS; 0 otherwise.
- $S_t(n) (= \sum_k S_{k_t}(n), n \in N_{AP})$  Should be 1 if node  $n$  is the end of any BS; 0 otherwise.
- $L_{a_i}$ : The labels assigned to links  $a_i \in AP$  (as discussed above). For the  $i$ -th and  $(i + 1)$ -th link on AP, we have  $L_{a_i} \leq L_{a_{i+1}} \leq L_{a_i} + 2$ , where  $i \geq 0$ . In addition, we specify  $L_{a_0} = 0$ . In this way, the  $BS_k$  protects links labeled  $2k - 1$  or  $2k$ .
- $F_k (= \sum_{n \in N_{AP}} n S_{k_f}(n))$ : The *order* of the node starting  $BS_k$ .
- $T_k (= \sum_{n \in N_{AP}} n S_{k_t}(n))$ : The *order* of the node ending  $BS_k$ .
- $L_{in}(n)$ : *In Label* of node  $n$ , i.e. the label of the link preceding  $n$  along the AP.
- $L_{out}(n)$ : *Out Label* of node  $n$ , i.e. the label of the link following  $n$  along the AP.

The objective function in the ILP formulation can be written as  $\min \sum_{e \in E} BC^e$  since the AP is already given and hence we only need to minimize the total backup bandwidth requirement.

The following are the constraints:

$$S_{kt}(s) = 0; S_{kf}(s) = 1 \text{ if } k = 0, 0 \text{ otherwise} \quad (1)$$

$$S_{kf}(d) = 0; S_{kt}(d) = 1 \text{ if } k = K^* - 1, 0 \text{ otherwise} \quad (2)$$

The above equations state that the source node  $s$  should be (and only be) the beginning of the 0-th BS, while the destination node  $d$  should be (and only be) the ending of the last BS.

$$S_f(n) + S_t(n) = L_{out}(n) - L_{in}(n) \Rightarrow L_{a_i} \leq L_{a_{i+1}} \leq L_{a_i} + 2 \quad (H_a - 1 > i \geq 0), n \neq s, d; n \in N_{AP} \quad (3)$$

Eq. (3) is a result of the link labeling scheme.

$$F_k < T_k; F_k < F_{k+1}, T_k < T_{k+1}, F_{k+1} \leq T_k; T_k \leq F_{k+2} \quad (4)$$

Eq. (4) implements the two criterium of valid Active Segmentation as discussed in Sec. II-B.

$$S_{kf}(n) = S_{kt}(n) \quad n \notin N_{AP} \quad (5)$$

Eq. (5) ensures flow balance at nodes that are not along the AP.

$$0 \leq \sum_k 2k \cdot S_{k_a} - L_a \leq 1, \sum_k S_{k_a} = 1 \quad a \in AP \quad (6)$$

From Eq. (6),  $S_{k_a}$  will be set to 1, only when  $2k - 1 \leq L_a \leq 2k$ . i.e., active link  $a$  is protected by the  $BS_k$ .

$$BC^e \geq BC_g^e(S_{k_a} + S_{k_e} - 1) \quad \forall a \in AP, \forall e \notin AP \quad (7)$$

Eq. (7) states the minimum additional backup bandwidth required on link  $e$  if active link  $a$  is protected by a BS traversing link  $e$ , where  $BC_g^e$  is determined as discussed in Sec. IV-A.

$$S_{k_e}, S_{kf}(n), S_{kt}(n), S_f(n), S_t(n) \in \{0,1\}, w \geq BC^e \geq 0 \quad (8)$$

Moreover, the range of the value of some of the variables are constrained as above.

Finally, if we want to limit the length of each BS (in hops) to  $m$ , we can add the following constraint:  $\sum_{e \notin AP} S_{k_e} \leq m$

### C. A Dynamic Programming Based Solution

Although solving the ILP formulation will result in an ‘‘optimal’’ set of BS’s for a given AP, its exponential time complexity makes it impractical for very large networks. In this section, we describe an alternative using dynamic programming heuristic that has only a polynomial time complexity and thus is feasible for very large networks.

Let us number the nodes along the AP from 0 to  $H_a$ . In addition, assume an AS covers node  $i$  to node  $j$ , and let  $B_{i,j}$ , where  $0 \leq i < j \leq H_a$ , be the minimum-cost BS for the AS. After removing all edges on AP, the cost for link  $e$  is estimated as  $\max_{g \in \mathcal{G}_{AS}} BC_g^e$ , where  $\mathcal{G}_{AS}$  is the union of the SRLG’s to which the links along the AS belong, then the BS can be found using a shortest-path algorithm.

Now let  $D_{i,j}$  be the ‘‘best’’ way (known to dynamic programming) to protect the part of AP from node  $i$  to node  $j$  by possibly dividing it into multiple (overlapping) AS’s, and protecting them with corresponding BS’s, without considering how other parts of the AP are protected.

The algorithm works by determining  $D_{i,j}$  as follows: In Step 1, pick  $B_{i,i+1}$  to be  $D_{i,i+1}$  (see Fig. 4 (a)); In Step 2, pick either the combination of  $D_{i,i+1}$  with  $D_{i+1,i+2}$ , or  $B_{i,i+2}$ , whichever is better, to be  $D_{i,i+2}$  (see Fig. 4 (b)); In Step 3, pick the best among the following four choices to be  $D_{i,i+3}$  (see Fig. 4 (c)): the combination of  $D_{i,i+1}$  with  $D_{i+1,i+3}$ , the combination of  $D_{i,i+2}$  with  $D_{i+1,i+3}$ , the combination of  $D_{i,i+2}$  with  $D_{i+2,i+3}$ , and  $B_{i,i+3}$ . The process ends when  $D_{0,H_a}$  is decided (after the  $H_a$ -th step).

To facilitate a more detailed description of the algorithm, let  $\text{Comb}(X,Y)$  denote a heuristic function which accepts two BS sets  $X$  and  $Y$  as parameters, and outputs a new BS set, which

is essentially a union of the two BS sets after removing any redundant BS (in order to reduce the backup bandwidth required). More specifically, each BS will be assigned a cost, which is the total bandwidth allocated to it divided by the number of links protected by the BS (i.e., along the corresponding AS). All the BS's in sets  $X$  and  $Y$  are then sorted in the decreasing order according to their costs. Then, each BS is examined, starting from the highest-cost BS, to see if it can be removed. If all the links on its corresponding AS are also covered/protected by another BS (or a combination of other BS's) in sets  $X$  and  $Y$  with a (combined) lower cost, the BS is considered redundant and will be removed. This part of the algorithm has a time complexity of  $O((|X| + |Y|)^2)$ , where  $|X|$  and  $|Y|$  are the number of BS's in sets  $X$  and  $Y$ , respectively.

The pseudo-code of the dynamic programming based algorithm is shown below. There,  $JT$  is a variable used to specify whether we want to protect against all single-link failure or all single-node failure. Its value is set to 0 if we want the former and 1 otherwise.

---

```

for  $m = 1 + JT$  to  $H_a$  do
  for  $i = 0$  to  $H_a - m$  do
     $D_{i,i+m} \leftarrow B_{i,i+m}$ 
    for  $j = 1 + JT$  (up) to  $m - 1$  do
      for  $k = 1$  to  $j - JT$  do
         $D_{i,i+m} \leftarrow \min(\text{Comb}(D_{i,i+j}, D_{i+k,i+m}), D_{i,i+m})$ 
      end for
    end for
  end for
end for

```

---

Once the BS set is chosen, the minimum amount of additional backup bandwidth will be allocated on each link used by one or more BS's, taking into consideration inter-sharing as well as intra-sharing. As to be shown later, this dynamic programming heuristic can obtain near-optimal results (with respect to PROMISE with ILP) that are still better than those achieved by shared path protection using Enhanced Two-Stage ILP.

#### D. Find a candidate AP

Due to the flexibility of PROMISE, the proposed PROMISE rarely fails to find a BS set to protect an AP found using a shortest-path algorithm even in SRLG networks infested with avoidable traps. However, as a safety measure, we can make use of the results from the ILP in [5]. More specifically, for each node pair, we compute an SRLG-disjoint path pair ( $\mathcal{P}_1$  and  $\mathcal{P}_2$ ) between them with the basic ILP off-line, (which can obtain results quickly), and keep the union of the directed edges in  $\mathcal{P}_1$  and  $\mathcal{P}_2$  as the "Preferred Set",  $\mathcal{P}$ . In case we cannot find a BS set to protect an AP candidate, we exclude one edge in AP that does not belong to  $\mathcal{P}$  from the selection of all future AP candidates, and then finds a new AP candidate. The algorithm will give up if no new AP candidate can be found (e.g., due to real traps). In a network with unlimited bandwidth on each link (i.e., an uncapacitated network), we can prove that, as long as  $\mathcal{P}$  exists between a pair of end nodes, for any AP consisting of the edges of  $\mathcal{P}$  (instead of just  $\mathcal{P}_1$ ), a SRLG-disjoint BS set is guaranteed (the proof is omitted from this paper due to space limitation.)

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of four PROMISE schemes, all of which use the same APF function to determine AP candidates for a given request. The differences among

them are that two of them use complete information (called PROMISE with CI or simply PCI), while the other two used partial information (called PROMISE with PI or simply PPI). The two PCI schemes are further classified into **PCI-I** and **PCI-D** where the last letter indicates whether ILP or dynamic programming is used to determine the set of BS's for a given AP. Similarly, the two PPI schemes will be referred to as **PPI-I** and **PPI-D**. As the comparison, we also implement Enhanced Two-Stage ILP (ETI). Note that in ETI, once the AP and BP are chosen, minimal BBW ( $\leq w$ ) will be allocated on each link along the BP, whether complete or partial information is available. In fact, since it determines the same path pair with whether complete or partial information is available, ETI performs the same in either case.

The two PCI schemes are compared to ETI. More specifically, we have simulated these five approaches by injecting a large number (e.g., few hundreds) of randomly generated requests (one after another in an on-line fashion) into networks with various topologies. The following are the performance metrics used and typical results that have been obtained for a typical carrier-like transport backbone network, which consists of 119 nodes, 190 bi-directed edges and 388 SRLG's.

#### A. Performance Metrics

1) *Bandwidth Saving Ratio (BSR)*: is defined as the saving of total bandwidth consumed by each of the five approaches compared to that by basic ILP scheme (without BBW sharing). Note that, even if an ideal scheme that achieves maximum BBW sharing is used, the bandwidth saving ratio will be upper-bounded by 50% (achievable only if no BBW is needed at all) [20]. To make fair comparison between different schemes, the results are obtained assuming networks where links have a sufficiently large capacity, and that each connection has a sufficiently long duration.

2) *Earning Fulfillment Ratio (EFR)*: In a different set of experiments (simulations), we assume that each link has a finite capacity and fluctuating traffic (i.e. an established connection may terminate after a certain duration). More specifically, each link is assumed to have a capacity of 48 units in each direction (to model an OC-48 link). As a result, some requests may be rejected under a heavy traffic load. In addition to blocking probability, we use the total earnings (or revenues) as a metric as in [20], which based on a scheme-independent *Earning Rate* matrix whose entry at  $(i, j)$  represents earnings per bandwidth unit and time unit by satisfying a connection from ingress node  $i$  to egress node  $j$ . The earnings from a connection from  $i$  to  $j$  is thus the product of the earning rate, requested units of bandwidth, and the connection duration. We compare the Earning Fulfillment Ratio (EFR), which is the percentage of the total possible earnings obtained by each scheme (if a scheme satisfies all the requests, its EFR is 100%). Note that the blocking probability may not be a fair metric since it does not differentiate the blocking of two requests with completely different resource requirements (e.g. hop lengths) [20].

#### B. Performance Results

Table I shows the results on BSR and EFR on capacitated and un-capacitated network respectively. It shows that the PROMISE schemes consistently outperform their share path protection counterparts with ETI in simulation either in capacitated and un-capacitated network, and with either complete or partial information.

Fig. 5 shows blocking probability for above simulation on capacitated network as complement. From this figure, we can find the similar results.

Scheme	BSR	EFR
ETI	13.6%	76.50%
PCI-I	28.0%	93.80%
PCI-D	26.0%	89.70%
PPI-I	19.2%	84.50%
PPI-D	17.9%	82.30%

TABLE I  
BSR & EFR

## VI. CONCLUSION

In this paper, we have developed a scalable (with polynomial time-complexity) and efficient

solution for shared SRLG protection. Our solution is to use dynamic programming algorithm based on PROMISE scheme for finding a set of backup segments (BS's) to protect a given active path (AP) in case of a single SRLG failure.

We have found that PROMISE can achieve a superior bandwidth efficiency than shared SRLG-disjoint path protection (with enhanced two-stage ILP), and at the same time, deal with both real and avoidable traps effectively in SRLG networks using only a fast polynomial time algorithm, due to its inherent flexibility in partitioning an AP and protecting it with multiple BS's.

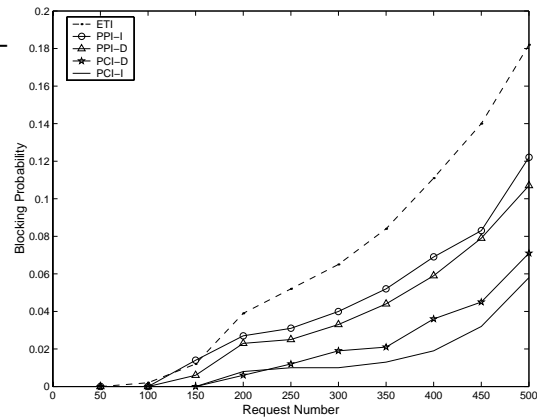


Fig. 5. Blocking Probability

## REFERENCES

- [1] P. Sebos, J. Yates, and et al., "Effectiveness of shared risk link group auto-discovery in optical networks," in *OFC'02*, 2002, p. Th05.
- [2] J. Luciani et al., "IP over optical networks a framework," in *Internet draft, work in progress*, Mar. 2001.
- [3] D. Papadimitriou et al., "Inference of shared risk link groups," in *Internet draft, work in progress*, Nov. 2001.
- [4] P. Sebos, J. Yates, and et al., "Auto-discovery of shared risk link groups," in *OFC'01*, 2001, p. WDD3.
- [5] J.Q. Hu, "Diverse routing in mesh optical networks," in *Submission*, <http://people.bu.edu/hqiang/papers/srlg.pdf>, 2002.
- [6] Guangzhi Li, Bob Doverspike, and Chuck Kalmanek, "Fiber span failure protection in mesh optical networks," in *Optical Networks Magazine Vol. 3, No. 3*, May/June. 2002.
- [7] R. Bhandari, "Survivable networks: algorithms for diverse routing," in *Kluwer*, 1999.
- [8] K. Lee and K. Siu, "An algorithmic framework for protection switching WDM networks," in *NFOEC'01*, Jul. 2001, pp. 402–410.
- [9] D. Anthony Dunn, Wayne D. Grover, and Mike H. MacGregor, "Comparison of k-shortest paths and maximum flow routing for network facility restoration," in *IEEE Journal on Selected Areas of Communications, Vol. 2, No. 1*, Jan. 1994, pp. 88–99.
- [10] Dahai Xu, Yizhi Xiong, and Guangzhi Li, "Trap avoidance and protection schemes in networks with shared risk link groups," in *Submitted to INFOCOM'03*.
- [11] Dahai Xu, Yizhi Xiong, and Chunming Qiao, "Promise- a novel survivability approach for high-speed networks," in *Submitted to INFOCOM'03*.
- [12] R. Doverspike and J. Yates, "Challenges for MPLS in optical network restoration," in *IEEE Communications Magazine, Vol. 39, No. 2*, Feb. 2001, pp. 89–96.
- [13] E. Bouillet, J. Labourdette, and et al., "Stochastic approaches to compute shared mesh restored lightpaths in optical network architectures," in *INFOCOM'02*, 2002, pp. 801–807.
- [14] Yu Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," in *INFOCOM'01*, 2001, pp. 699–708.
- [15] Vishal Sharma et al., "Framework for MPLS-based recovery," in *Internet Draft, draft-ietf-mpls-recovery-fmwk-03.txt (work in progress)*, Jul. 2001.
- [16] C. V. Saradhi and C. Siva Ram Murthy, "Dynamic establishment of segmented protection paths in single and multi-fiber WDM mesh networks," in *OPTICOMM'02*, 2002, pp. 211–222.
- [17] Krishna P. Gummadi, M. J. Pradeep, and C. Siva Ram Murthy, "An efficient primary-segmented backup scheme for dependable real-time communication in multihop networks," in *To appear in IEEE/ACM Trans. on Network*, 2002.
- [18] Murali Kodialam and T V. Lakshman, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," in *INFOCOM'01*, 2001, pp. 376–385.
- [19] Pin-Han Ho and H.T. Mouftah, "A framework of a survivable optical internet using short leap shared protection (SLSP)," in *2001 IEEE Workshop on High Performance Switching and Routing*, 2001, pp. 21–25.
- [20] Chunming Qiao and Dahai Xu, "Distributed partial information management (DPIM) schemes for survivable networks - part I," in *INFOCOM'02*, Jun. 2002, pp. 302–311.
- [21] Murali Kodialam and T V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *INFOCOM'00*, 2000, pp. 902–911.