

Designing Fast and Bandwidth Efficient Protection Scheme for WDM Optical Networks

Yu Lin, Haitham S. Hamza, and Jitender S. Deogun

Department of Computer Science & Engineering,
University of Nebraska-Lincoln Lincoln, NE 68588-0115, USA
{ylin, hhamza, deogun}@cse.unl.edu

Abstract. In this paper, we introduce the *Pre-cross-connected Segment First* — PXSFirst protection scheme; a new path-based protection schemes for WDM optical networks. The goal of the PXSFirst scheme is to achieve fast recovery, while maximizing bandwidth sharing to improve bandwidth utilization. Similar to pre-cross-connected trails (PXTs) protection scheme, PXSFirst ensures that all backup paths are pre-cross-connected, and hence eliminates the switching configuration delay along backup paths. However, unlike the PXT scheme, where backup paths can only share trails, PXSFirst breaks backup paths into smaller segments by existing end nodes, and hence, increases the possibility of bandwidth sharing. Extensive simulation results for different network topologies and under different traffic patterns show that the proposed scheme has better blocking performance and less bandwidth utilization (an average of 11.0% reduction) compared to existing PXT protection schemes.

1 Introduction

The design of a survivable Wavelength Division Multiplexing WDM network that provides fast recovery with minimum network resources is an important problem. The significance of the problem increases considerably as (WDM) technology matures and more and more wavelengths per fiber become available, making even the failure of a single fiber catastrophic. Under single failure model, a wavelength-routed optical network may fail due to the failure of a single component (e.g. link or node) of the network. Such a component failure will result in a failure of all the connections those utilize that component (e.g. link).

Traditionally, ring protection schemes (e.g. SONET) are used to ensure fast recovery for any single link or node failure, due to the fact that traffic can be rerouted around a *single* node upon any single link or node failure. However, ring protection schemes do not utilize bandwidth efficiently. As a result, techniques that aim at effectively utilizing network bandwidth has been investigated. Most notable is the shared mesh protection scheme that enables more efficient utilization of network resources [1], [6], [11]. In mesh *shared protection* methods, backup resources are *pre-allocated* during the connection setup, where two or more backup lightpaths may share one backup path, given that the corresponding primary lightpaths are link and node disjoint. The sharing of backup paths, however, can lead to slow recovery. This is because switching nodes along the

shared backup paths may need to be reconfigured conditionally based on which particular primary path needs to be recovered.

An effective protection scheme, therefore, should attain the bandwidth efficiency of a mesh protection while providing a ring-like recovery speed [11], as well as to ensure transmission integrity. One of the recent approaches to accomplish this is the *pre-cross-connect trail* (PXT) [11]. The idea of PXT is to provision the network such that all protection paths are pre-cross-connected for prompt use upon failure [11]. However, under dynamic traffic, some nodes that are pre-cross-connected under the PXT scheme may become end nodes of a new request and thus cannot be pre-cross-connected when provisioning future requests. As a result, by arranging protection edges into pre-cross-connected trails, which is equivalent to keep track of the complete protection part of the network, some pre-cross-connected segments may fail to be reused, leading to less efficient utilization of bandwidth in the network.

Therefore, in order to improve bandwidth utilization, we need to reuse any pre-cross-connected segments along the backup paths. We thus believe that to ensure fast restoration while providing efficient bandwidth utilization, neither ring nor trails are sufficient, but rather we need to provision the network based on the *pre-cross-connected segments*, where a *segment* is one or more consecutive links that connects a pair of nodes in the network. In this paper, we therefore propose a new protection scheme that can ensure fast recovery while providing efficient bandwidth utilization. The new scheme is called Pre-cross-connected First (PXSFfirst) scheme. PXSFfirst forces the primary path to take the route which is segmented by existing end nodes.

The remainder of the paper is organized as following. Section 2 summarizes related work in fast protection schemes. The proposed algorithm is presented in Section 3. Simulation results are discussed in Section 4; Conclusions are presented in Section 5.

2 Related Work

Two main techniques have been proposed to achieve a mesh-like bandwidth efficiency, while providing a ring-like recovery speed: the *p-cycles* [12], and the *pre-cross-connect trail* (PXT)[11].

The basic idea of *p-cycles* is to route the primary traffic using an arbitrary mesh routing algorithm, but to constrain the protection routes to lie on certain predetermined rings. Over the last few years, the basic theory of *p-cycles* has evolved and several extensions have been proposed, e.g. [2], [3], [13], [14]. However, *p-cycles* are more suitable for *static* traffic and for link-based protection schemes, also, to achieve high bandwidth utilization more complex management algorithms are needed. In [4], the concept of *p-cycles* was generalized to path segment protection, which includes the pure path-based protection as a special case. Although a technique for provisioning dynamic demands was discussed in [4], the overhead and complexity of the mechanism make it impractical for large networks.

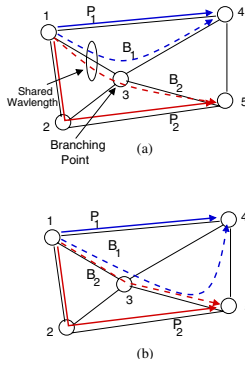


Fig. 1. Basic concept of branch points in PXT approach

In recent work [11], it has been observed that achieving fast recovery is not confined to networks with a ring-like topology. Instead, the key for fast recovery resides in the ability to pre-connect protection paths, and thus, no re-configuration of switching nodes along backup paths is needed. Accordingly, the concept of *pre-cross-connect trail* (PXT) was proposed. The basic concept of PXT is to provision backup paths with no *branch* points. We illustrated this concept by the following example.

Consider the network in Figure 1-a. Two requests need to be provisioned. The first request is from node 1 to node 4; (1, 4), while the second request is from node 1 to node 5; (1, 5). Using a conventional path-based protection scheme with backup sharing, a possible provisioning of primary and backup paths for the two requests is given in Figure 1-a, where P_i and B_i represent, respectively, the primary and backup paths allocated to request i . As shown in figure, the two backup paths share a wavelenght on link (1 – 3). At node 3, each backup path requires connection to a different output. Therefore, node 3 is considered to be a *branch point* in this network as the configurations of node 3 upon the failure of P_1 or P_2 are different. To reduce recovery time, PXT necessitates the elimination of such branch points, so that every node in the network can be pre-connected independent of the failure of a specific primary path. For example, Figure 1-b gives another possible path provisioning for the two requests (1, 4) and (1, 5). As shown in the figure, B_1 does not branch at node 3 as in Figure 1-a, and hence, node 3 is no longer a branch point.

Several recent work has investigated the concept of segment-based protection. In [15], a segment based protection scheme was proposed based on dynamic programming, however in this scheme, a link may be protected by two backup segments and therefore the overlapped protection wastes bandwidth. Moreover, the end nodes of a segment, exclusive of the source and destination node of the request, must be protected by a backup segment, and hence, more bandwidth is needed.

As discussed in [15], the heuristic algorithm proposed in [16] to determine segments cannot efficiently deal with some real trap scenarios and in addition

does not consider backup bandwidth sharing until the paths are found. [15] also suggests the scheme in [17] requires the node immediately upstream from the link/node failure to restore traffic along an alternate outgoing link, which limits its flexibility. [18] divides the primary path into segments of equal length and is therefore not flexible. Furthermore, in order to protect the last link and the end node incident to the last link, the last link of each segment is under the protection of two backup segments, which implies the scheme is not bandwidth efficient either.

3 The Proposed Algorithm

Before we present the formal algorithm of the proposed protection scheme, we first illustrate the underlying concepts of the proposed protection scheme.

3.1 The Basic Concept

We now introduce the main insight of this paper. In all path-based protection scheme, source nodes and destination nodes are assumed to be invulnerable, since the protection scheme needs the end nodes to react to the failure and perform a real time switching to reroute the traffic to the backup path. Therefore a protection segment ends in any two end nodes along the backup paths may be reused. Given the existing pre-cross-connected segments, we want to force the primary path to take the route which is divided by the end nodes in a way such that the resulting segments along the primary path are already under protection and are link and node disjoint with other segments under the protection of the same backup segment. We denote the set of primary segments protected by the same backup segments as a conflict set.

Now, we illustrate the difference between the proposed protection scheme and the conventional shared pre-cross-connected protection scheme with an example. In Figure 2, solid lines represent physical links, dashed lines represent the primary light paths and dotted lines represent the backup light paths of the requests upon link or node failure under the proposed PXSFirst protection scheme. Figure 3 represents the same network as Figure 2 represents except that the conventional shared pre-cross-connected protection scheme is applied. Each link consists of a single wavelength channel and each link has a unit capacity.

Upon the first request (A, C) , path (A, B, C) and (A, G, C) are allocated as primary path and backup path for this request respectively.

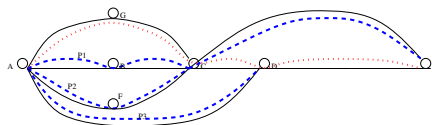


Fig. 2. Basic concept of PXSFirst

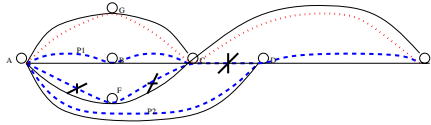


Fig. 3. Basic concept of conventional shared pre-cross-connected protection scheme

Upon the second request (A, E) , the conventional shared pre-cross-connected protection scheme (Fig. 3) applies the shortest path and allocates (A, D, E) as the primary path. It further allocates (A, G, C, E) as the backup path. We will see shortly that this allocation blocks the third request.

Under the proposed PXSFirst (Fig. 2), however, we want to reuse the existing pre-cross-connected backup segments as much as possible and therefore take the path (A, F, C, E) as the primary path, since the segment (A, F, C) is already under protection. Path (A, G, C, D, E) is further allocated as the backup path.

Now suppose the third request (A, D) arrives. Under the conventional shared pre-cross-connected protection scheme, (A, F, C, D) is the only available path from A to D, and there is no way to allocate a backup path for this primary path. The request is thus blocked due to the lack of backup path.

However under the proposed PXSFirst, path (A, D) can be allocated as the primary path and path (A, G, C, D) is reused as backup path. PXSFirst thus successfully escapes the trap by utilizing the segment already under protection. In this example, if segment (A, F, C) is not allocated to the second request, which is the case under the conventional shared pre-cross-connected protection scheme, the segment can be isolated and can thus be wasted, as illustrated by Fig. 3.

3.2 The PXSFirst Protection Algorithm

In the proposed algorithm, we assume the following constraints while provisioning resources for each request:

- C1: the current primary path is link disjoint with the existing primary paths.
- C2: the current primary path is link disjoint with the existing protection paths.
- C3: all links and nodes along the current primary path are protected except the end nodes.
- C4: existing protection edges are reused as much as possible without introducing branch points.

Before we present the details of the algorithm, we first provide a few definitions and notations. Let V_m denote the *conflict set* for a pre-cross-connected segment m . A conflict set V_m contains all primary segments whose backup path traverse the pre-cross-connected segment m . Therefore two primary segments whose backup paths traverse m must be node and link disjoint. It is also obvious that $V_m \neq \emptyset$ implies the segment m belongs to a backup path. Table 3.2 summarizes the notations and variables used in this paper.

Table 1. Summary of notations and variables used in this paper

| Notations | Definition |
|-----------|---|
| m | A segment, which can be an edge; a consecutive set of edges; or an end-to-end path. |
| V_m | The conflict set V_m of segment m . |
| A_r | The set of segments that are link and node disjoint with the current primary path, exclusive of the end nodes of the current request r ; and link disjoint with the existing primary paths. |
| M | The set of pre-cross-connected segments resulting from dividing the backup paths by end nodes. |
| p_m | Given $m \in M$, the shortest path between the end nodes of m , link and node disjoint with any segments in V_m |
| M' | $\bigcup_{m \in M} p_m$ |

Upon a request arrival, PXSFirst factors out, with all combinations of two end nodes, any new segments along the backup paths. The resulting segments, along with the primary paths these segments protect, are appended to the global set M , such that all segments in M are reusable by the future requests.

To provision a primary path for the request, PXSFirst finds a shortest path between the two end nodes of every pre-cross-connected segment m . The shortest path must be link and node disjoint with the primary segments in V_m . The set of shortest paths for all m in M is denoted as M' . PXSFirst then constructs an auxiliary graph P with the same set of nodes as in G , which is the original graph representing the network. The cost of link e in G is denoted as $C_G(e)$. Graph P 's link cost function, denoted as $C_P^E(e)$, associates link e in P with the set of segments E in G that share the same end nodes as e in P . Since we want the primary path to take the route that has already been protected to the maximum extent possible, we multiply $C_P^E(e)$ by ϵ , where $\epsilon < 1.0$, if any segment in E , to which e corresponds to, is in M' . Otherwise, we want the primary path to take the route in G , which is link disjoint with the existing primary paths. The link cost function can be formalized as follows:

$$C_P^E(e) := \begin{cases} \epsilon, & \text{if } \exists m \in E, m \in M'; \\ C_G(e), & \text{Otherwise.} \end{cases} \tag{1}$$

PXSFirst then applies the shortest path algorithm with the new relax step on P . The new relaxation step, described in Fig. 4, is to ensure the resulting path in P is indeed a path in G , that is the segments in G corresponding to the links along the resulting path in P after expansion do not overlap in G .

During the second phase of the PXSFirst scheme, we allocate a backup path, that confirms to the following constraints: a) the current primary path and backup path for request r are node disjoint except for any end nodes, b) the

RELAX(u, v)

1. **Expand** all links on the partially available primary path/backup path to the corresponding segments in G , the set of nodes along the segments form S_1 ;
2. **Expand** the link from u to v to the corresponding segment in G , the set of nodes along the segments form S_2 ;
3. **If** S_1 and S_2 intersects on a node other than u , continue;
4. **If** $c^v > c_u^v + c^u$
5. **Set** node u as node v 's previous hop;
6. **Append** S_2 to S_1 .

Fig. 4. RELAX Step in a Standard Shortest Path Algorithm

backup path for request r and the set of existing primary paths are link disjoint, c) backup sharing does not introduce branch points. We construct another auxiliary graph H that captures these constraints and the link cost function of H is denoted as $C_H^E(e)$, where e in H is associated with the set of segments in G , which share the same end nodes as e . The afore-going constraints are captured by the second case in the cost function below. The first case of the link cost function ensures the maximum backup paths sharing. Otherwise the backup path is free to use any idle link in G , as implied by the third case of cost function.

$$C_H^E(e) := \begin{cases} 0, & \text{if } \exists m \in E, m \in M \wedge p(r) \notin V_m; \\ +\infty, & \text{else if } \forall m \in M, m \in \bar{A}_r; \\ C_G(e), & \text{Otherwise.} \end{cases} \quad (2)$$

Overall, the proposed PXSFirst (see Fig. 5) applies shortest path algorithm with the new relaxation step on the auxiliary graphs P and H during the primary path and backup path allocation phase. For any node v in P/H , we associate a cost variable c^v , which denotes the cost of the primary path/backup path from the starting source node to v and a cost variable c_u^v , which denotes the cost from u to v . As the shortest-path algorithm progresses on P/H , we maintain a set S_1 containing the nodes on the segments in G , that correspond to the links on the partially available primary path/backup path, with the new relaxation step.

In PXT algorithm, the resulting path in H expands to a trail in G , as the name of the algorithm alludes. PXT algorithm then stores the trail in L_1 and removes the cycles on the trail to output a backup path. We argue it is not efficient from the storage and algorithm simplicity point view and claim it is more meaningful to embed the cycle control within the standard shortest-path algorithm with the new relaxation step.

4 Numerical Evaluation

In this section, we experimentally evaluate the performance of the proposed algorithm on the Italian network (not shown here), the Pacific Bell network, and

PXSFirst**Input:** Graph G , request r , set M as defined in the table of notations and variables.**Output:** $p(r)$ and $b(r)$ satisfying the constraints C1- C4.

1. **Update** set M by factoring out any new segments that ends in any combination of end nodes.
2. **Construct** graph P with link cost function $C_P^E(e)$;
3. **Apply** shortest path algorithm on P with the new relaxation step and obtain $p(r)$;
4. **Update** G by marking the links used by $p(r)$;
5. **Construct** graph H with link cost function $C_H^E(e)$;
6. **Apply** shortest path algorithm on H with the new relaxation step and obtain $b(r)$;
7. **Update** G by marking the links along $b(r)$, which are not in use before .

Fig. 5. The proposed PXSFirst algorithm

a 12-node random network. Moreover, to validate the findings, we also used a 12-node ring, bipartite, grid, and clique topologies. For each topology, we simulated three different request models [11]: *Uniform Requests*, a *Nearest-Neighbor Requests*, and *Unbalanced Requests*. For experimental purpose we set ϵ in Equation (1) to zero. We obtained the results by running the algorithms once on each distinct sequence of traffic demand arrivals.

4.1 Assumptions

In this study, we consider an incremental request model, where requests are fed one at a time to the algorithm; Full-range Wavelength converters, where an input wavelength can be converted to any output wavelength; a fixed set of routes between every pair in the network are pre-computed and are not affected by the dynamics of network; and a single failure model [7].

Three different request models are considered [11]:

1. *Uniform Requests*. Every pair of nodes appears five times in the request list.
2. *Nearest-Neighbor Requests*. Every pair of neighboring nodes appears 5 times.
3. *Unbalanced Requests*. We choose three arbitrary nodes as *Critical* (C) nodes and the rest of the nodes as *Normal* (N) nodes. Critical nodes may represent nodes that groom high-priority traffic, for example. The combination of a C node with an C node, a C node with an N node, and an N node with an N node appear, respectively, 12, 8, and 5 times in the request list.

4.2 Evaluation Metrics

We use five metrics to evaluate the performance of the proposed algorithm:

1. *Blocking Probability*. The blocking probability, b_p , for a set of requests, R :

$$b_p = \frac{\# \text{of blocked requests}}{|R|} \quad (3)$$

A request is blocked if there is no available primary path or available backup path that can be allocated to this request.

2. *Bandwidth Utilization.* The bandwidth utilization, b_u , for a given protection scheme is defined as:

$$b_u = \frac{\sum_{\forall e \in E} \# \text{of used wavelengths on } e}{|E|.W} \tag{4}$$

b_u measures the number of W used to provision all requests.

3. *Mean Primary Path length.* The mean of primary paths length, $|\overline{p(r)}|$, for a given protection scheme is defined as:

$$|\overline{p(r)}| = \frac{\sum_{\forall r \in R} |p(r)|}{|R|}, \tag{5}$$

where R is set of all requests

4. *Mean Backup Path length.* The mean of backup paths length, $|\overline{b(r)}|$, for a given protection scheme is defined as:

$$|\overline{b(r)}| = \frac{\sum_{\forall r \in R} |b(r)|}{|R|}. \tag{6}$$

Table 2. Blocking probability (b_p) and Bandwidth utilization (b_u) of different protection schemes ($W = 20$). Only instances in which the PXT and PXSFirst have $b_p = 0$ are considered.

| Simulation Parameters | | | Blocking Probability | | | | Bandwidth Utilization | | |
|-----------------------|------------|-------|----------------------|------|-------------|----------|-----------------------|-------------|----------|
| Network | Traffic | # Req | 1+1 | PXT | Shared Path | PXSFirst | PXT | Shared Path | PXSFirst |
| bipartite | uniform | 660 | 51.8 | 27.4 | 9.5 | 6.0 | — | — | — |
| | nearestNB | 360 | 18.1 | 0.0 | 0.0 | 0.0 | 30.0 | 26.3 | 17.5 |
| | unblanced | 864 | 63.9 | 41.2 | 20.7 | 17.0 | — | — | — |
| clique | uniform | 660 | 5.9 | 0.0 | 0.0 | 0.0 | 30.0 | 25.9 | 17.3 |
| | nearestNB | 660 | 7.1 | 0.0 | 0.0 | 0.0 | 30.0 | 25.8 | 17.2 |
| | unblanced | 864 | 15.9 | 0.0 | 0.0 | 0.0 | 37.7 | 33.5 | 22.3 |
| ring | uniform | 660 | 96.2 | 91.2 | 75.6 | 82.0 | — | — | — |
| | nearestNB | 120 | 69.2 | 0.0 | 0.0 | 0.0 | 30.0 | 30.0 | 19.6 |
| | unbalanced | 864 | 97.5 | 93.8 | 88.8 | 83.7 | — | — | — |
| grid | uniform | 660 | 88.5 | 76.1 | 55.3 | 61.8 | — | — | — |
| | nearestNB | 165 | 21.2 | 0.0 | 0.0 | 0.0 | 30.0 | 30.0 | 18.8 |
| | unbalanced | 864 | 90.4 | 80.0 | 65.2 | 68.1 | — | — | — |
| random | uniform | 660 | 81.5 | 70.5 | 45.9 | 50.2 | — | — | — |
| | nearestNB | 200 | 5.0 | 0.0 | 0.0 | 0.0 | 30.0 | 30.0 | 27.5 |
| | unbalanced | 864 | 86.5 | 78.9 | 58.9 | 60.1 | — | — | — |
| Pacific | uniform | 1050 | 92.8 | 84.7 | 67.3 | 74.0 | — | — | — |
| | nearestNB | 210 | 21.0 | 0.0 | 0.0 | 0.0 | 30.0 | 30.1 | 19.0 |
| | unbalanced | 1308 | 92.4 | 87.2 | 71.8 | 76.5 | — | — | — |

4.3 Results

We conducted experiments on networks with 10 and 20 wavelengths per fiber. The results follow the same trends, thus we only show results for the 20 wavelengths case. For comparison, we also implemented the traditional 1+1, the original PXT and the Shared Path protection scheme. In the following, we summarize our main results.

1. *Blocking Probability (p_b)*. Table 2 gives the (p_b) of the four protection schemes for the different topologies and under the three request models. As shown in the tables, the new algorithm has the best performance compared to 1+1 and PXT in all cases and Shared Path protection schemes for bipartite and ring topology under certain traffic patterns. Comparing to the PXT protection scheme, PXSFirst reduces the blocking probability by an average of 8.75%.
2. *Bandwidth Utilization (b_u)*. We compared b_u for cases in which both the PXT, Shared Path and PXSFirst protection scheme have zero blocking (i.e. all requests were successfully provisioned). Table 2 shows the values of b_u for different topologies. As shown in the table, PXSFirst provides better bandwidth utilization compared to PXT for all the topologies. Under the selective request models, the improvement reaches an average of 11.0%

Table 3. Mean Primary Path Length ($\overline{|p(r)|}$) and Mean Backup Path Length ($\overline{|b(r)|}$) of different protection schemes for ($W = 20$). Only PXT and PXSFirst protection schemes are considered.

| Simulation Parameters | | | $\overline{ p(r) }$ | | $\overline{ b(r) }$ | |
|-----------------------|------------|-------|---------------------|----------|---------------------|----------|
| Network | Traffic | # Req | PXT | PXSFirst | PXT | PXSFirst |
| bipartite | uniform | 660 | 1.62 | 1.51 | 1.53 | 3.38 |
| | nearestNB | 360 | 1.0 | 1.0 | 1.0 | 3.36 |
| | unblanced | 864 | 1.58 | 1.55 | 1.52 | 3.72 |
| clique | uniform | 660 | 1.0 | 1.0 | 1.0 | 3.46 |
| | nearestNB | 660 | 1.0 | 1.0 | 1.0 | 4.20 |
| | unblanced | 864 | 1.0 | 1.0 | 1.0 | 3.44 |
| ring | uniform | 660 | 2.76 | 2.80 | 6.34 | 4.78 |
| | nearestNB | 120 | 1.0 | 1.0 | 1.0 | 2.25 |
| | unbalanced | 864 | 2.66 | 2.44 | 6.21 | 4.70 |
| grid | uniform | 660 | 1.93 | 2.06 | 2.93 | 3.93 |
| | nearestNB | 165 | 1.0 | 1.0 | 1.0 | 2.82 |
| | unbalanced | 864 | 1.98 | 2.03 | 2.17 | 4.85 |
| random | uniform | 660 | 1.94 | 2.05 | 3.63 | 2.58 |
| | nearestNB | 200 | 1.0 | 1.0 | 1.0 | 3.0 |
| | unbalanced | 864 | 1.96 | 2.05 | 2.32 | 3.73 |
| Pacific | uniform | 1050 | 1.97 | 2.12 | 2.70 | 3.27 |
| | nearestNB | 210 | 1.0 | 2.0 | 1.0 | 1.9 |
| | unbalanced | 1308 | 1.90 | 3.02 | 2.53 | 3.48 |

3. *Mean Primary Path Length* ($|\overline{p(r)}|$). In order to assess the tradeoff for lower blocking probability and better bandwidth utilization, we also compared the mean primary path length for the PXT and PXSFirst protection scheme. As Table 3 shows, the mean primary path length using PXSFirst increases by an average of 0.54%, compared to that of the PXT scheme.
4. *Mean Backup Path Length* ($|\overline{b(r)}|$). While evaluating the mean backup path length, we first realize that in some topologies such as in ring and random networks, the mean backup path length under certain request models using PXSFirst is shorter than the one in PXT, whereas in other topologies or in the preceding topologies but under some request models the mean backup path length is shorter, as shown in Table 3. However the overall mean backup path length using PXSFirst increases by an average of 54.8%.

5 Conclusions

In this paper, we address the problem of developing a fast and bandwidth efficient path protection scheme for WDM optical networks. We introduce the concept of PXSFirst — Pre-cross-connected Segment First that further improves the existing PXT scheme by forcing the primary path to take the route which has been protected by pre-cross-connected segments to the maximum extent possible. Extensive simulation results of PXSFirst on several network topologies and under three different traffic models confirm an improvement in both blocking performance (an average of 8.75%) and bandwidth utilization (an average of 11.0%) compared to PXT protection scheme by a slight increase in primary path length (an average of 0.54%) and a large increase in backup path length (an average of 54.8%).

In the future, we plan to evaluate the algorithm's primary path and backup path length further. We plan to investigate the primary path length and backup path length under unlimited resources.

References

1. C. Ou, J. Zhang, H. Zang, L.H. Sahasrabudde, and B. Mukherjee, "New and improved approaches for shared-path protection in WDM mesh networks," *J. Light-wave Technology*, vol. 22, no. 5, pp. 1223 - 1232, 2004.
2. D. A. Schupke, C. G. Gruber, and A. Autenrieth "Optimal configuration of p -cycles in WDM networks," *Proc. of IEEE ICC '02*, pp. 2761-2765, 2002.
3. D. Stamatelakis, "Theory and algorithms for preconfiguration of sparse capacity in mesh restorable networks," *M.Sc. Thesis*, University of Alberta, Canada, 1997.
4. G. Shen and W.D. Grover, "Extending the p - cycle concept to path segment protection for span and node failure recovery," *J. of Selected Areas in Communications*, vol. 21, no. 8, 2003.
5. G. Mohan, C. Siva Ram Murthy, and A. K. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," *IEEE/ACM Transactions on Networking*, vol. 9, no.5, pp. 553- 566, October 2001.

6. S.-I. Kim and S.S. Lumetta, "Capacity-efficient protection with fast recovery in optically transparent mesh networks," *Proc. 1st International Conference on Broadband Networks (BroadNets 2004)*, pp. 290 - 299, 2004.
7. S. Ramamurthy, L. Sahasrabudhe, and B. Mukherjee, "Survivable WDM Mesh Networks," *Journal of Lightwave Technology*, vol. 21, no. 4, 2003, pp. 870-882.
8. S. Ramamurthy and B. Mughherjee, "Survivable WDM mesh networks- Part I: Protection," *Proc. IEEE INFOCOM*, Mar. 1999, pp. 744-751.
9. S. Ramamurthy and B. Mughherjee, "Survivable WDM mesh networks- Part II: Restoration," *Proc. IEEE Integrated Circuits Conf.*, June 1999, pp. 2023-2030.
10. R. Sabella, E. Iannone, M. Listanti, M. Berdusco, and S. Binetti, "Impact of transmission performance on path routing in all-optical transport networks", *IEEE J. Select. Areas Commun.*, vol.6, pp. 1617-1622, Dec. 1988.
11. T.Y. Chow, F. Chudak, and A.M. Ffrench, "Fast optical layer mesh protection using pre-cross-connected trails," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 539-548, 2004.
12. W.D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," *Proc. IEEE Int. Conf. Communications*, pp. 537-543, 1998.
13. W. D. Grover and J. E. Doucette, "Advances inoptical network dsign with p -cycles: joint opitimization and pre-selection of candidate p - cycles," *Proc. IEEE-LEOS Topical Meeting*, WA2-49-WA2-50, 2002.
14. W. Grover and D. Stamatelakis, "Bridging the ring-mesh dichotomy with p - cycles," *Proc. IEEE/VDE DRCN 2000*, pp. 92-104, 2000.
15. D. Xu, Y. Xiong, C. Qiao, "A New PROMISE Algorithm in Networks with Shared Risk Link Group," IEEE Globecom, San Francisco, CA, Dec. 2003
16. C.V. Saradhi and C.S.R. Murthy, "Dynamic establishment of segmented protection paths in single and multi-fiber WDM mesh networks," *OPTICOMM'02*, pp. 211-22.
17. M. Kodialam and T. V. Lakshman, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," *INFOCOM'01*, 2001, pp. 376-385
18. P-H Ho and H. Mouftah, "A framework for service-guaranteed shared protection in WDM mesh networks," *IEEE Comm. Mag.*, vol. 40, no. 2, 2002, pp. 97-103