

# Capacity-Efficient Protection with Fast Recovery in Optically Transparent Mesh Networks\*

Sun-il Kim

University of Illinois at Urbana-Champaign  
Computer Science Department  
Coordinated Science Laboratory  
sunilkim@uiuc.edu

Steven S. Lumetta

University of Illinois at Urbana-Champaign  
ECE Department  
Coordinated Science Laboratory  
lumetta@uiuc.edu

## Abstract

*Survivability becomes increasingly critical in managing high-speed networks as data traffic continues to grow in both size and importance. In addition, the impact of failures is exacerbated by the higher data rates available in optical networks. It is therefore imperative to address network survivability in an efficient manner in order to design and operate reliable networks.*

*Transparent optical networks (TONs) provide several advantages over optically opaque networks for supporting the growing communication demands, but suffer from several drawbacks that reduce the efficacy of most applicable capacity-efficient survivability techniques. In this paper, we introduce a novel protection algorithm (for single link and node failures) called Streams. The Streams algorithm is similar to 1:1 dedicated path protection in terms of implementation and operation overhead, and has identical recovery speeds while requiring less capacity.*

*We compare the Streams algorithm with dedicated and shared path protection in terms of capacity requirements, path lengths, and recovery time. We also extend the flooding based mesh restoration algorithm (FBMR) in order to provide a fair comparison in online routing scenarios, and report the relative tradeoffs between the different algorithms. Our results show that dynamically routed Streams offer attractive tradeoffs in terms of capacity, path length, recovery speed, data loss and implementation complexity.*

## 1. Introduction

With the extremely high volume of traffic carried on wavelength division multiplexed (WDM) networks, failures such as fiber cuts can result in a loss of huge amounts of data and revenue. In order to maintain high quality services for the increasing communications demands, we must be able to provide a desired level of robustness in a cost-efficient manner.

TONs offer several critical advantages over optically opaque networks. Faster switching can be achieved with the absence of electronic and photonic processing delays that can act as a bottleneck on the total transmission time. Unlike opaque networks, TONs can handle signals with different data rates, protocols, and formats, allowing architectures independent of today's standards. However, they suffer from limited functionality in wavelength conversion, signaling capabilities and detailed performance monitoring [21, 26]. These limitations reduce the efficiency of many of the recovery algorithms studied in the literature and make implementation more challenging [3]. These issues, coupled with the fact that the impact of failures in TONs is exacerbated by their high traffic volumes, necessitate a better understanding of the tradeoffs between different survivability algorithms. In considering survivability options, understanding the tradeoffs between recovery speed, data loss, capacity requirements, and the complexity of different protection algorithms is imperative. Our goal then is to provide rapid failure recovery in TONs in an efficient manner, and reduce service disruptions and loss of data. In this paper, we evaluate different protection algorithms and study their tradeoffs.

First, we introduce a novel protection algorithm called *Streams*, which allows for rapid recovery from all single link or single node failures, and is comparable to dedicated path protection while utilizing 10-28% less total capacity. The Streams algorithm adds only a single step to the recovery initiation process (with no added delay) rela-

---

\* We wish to thank Lois Yoo for her editorial efforts, and would also like to thank Canhui Ou for an interesting discussion about FBMR and path protection. The material presented in this paper is based in part upon work supported by National Science Foundation grants ANI 01-21662 ITR and ACI 99-84492 CAREER. Content of the information does not necessarily reflect the position or the policy of that organization.

tive to 1:1 (one-for-one) dedicated path protection (DPP). Streams is powerful because it can also coexist with other algorithms in the same network as it does not require additional equipment support. It can readily be implemented on current networks that utilize DPP or shared path protection (SPP). Second, we implement a dynamic version of the flooding based mesh restoration (FBMR) algorithm [12] and show that a simple online routing algorithm supports FBMR surprisingly well in terms of both capacity and *protection path length expansion* (average ratio of the protection path lengths over shortest paths). We focus on online protection routing, and thus assume that full offline optimization is not an option.

Streams and FBMR provide recovery speed comparable to ring-based protection while requiring less reserved capacity. These algorithms can achieve capacity efficiency comparable to SPP. SPP, FBMR and Streams offer an interesting tradeoff between capacity and protection path length. By adjusting the degree to which protection path lengths can be extended, a network can be operated at different points in the space of capacity versus path length expansion. We quantify this tradeoff in Section 5.

We next briefly touch on some of the previous work done in the area of survivability. Streams is introduced in detail in Section 3, followed by a detailed discussion of the protection algorithms (including DPP, SPP, FBMR) that are used in our evaluation. Section 5 presents the evaluation results, and we conclude the paper in Section 6.

## 2. Background

Survivability schemes are classified into two main categories—Protection and Restoration—in [18]. Restoration locates free  $\lambda$ -channels for backup after a failure occurs requiring minimal backup capacity, but the recovery can take as much as two seconds. Protection preplans backup routes that are used in the event of a failure, and offer faster recovery (tens of milliseconds) due to the absence of the signaling delay necessary for dynamic route discovery [21, 6, 8, 23]. Protection can also be implemented in a capacity efficient manner, and has been given much attention in the literature [12, 24, 7, 22, 2, 17]. With fast recovery in mind, we focus on protection for the rest of this paper.

Protection algorithms can be broadly classified into local (link/node) protection and path protection. Path protection requires selection of disjoint primary and backup path pairs. In DPP, a dedicated backup path is reserved for every primary path, and in the event of a failure, the traffic is simply switched over to the dedicated path. Therefore, DPP offers rapid recovery and is simple to operate and manage. In SPP schemes, backup channels are chosen in advance, but not preconfigured. Instead, the end nodes of a light-

path signal the intermediate nodes to establish the backup route after a failure occurs. Capacity reserved for backup can be shared among different connections that do not share nodes or links, or can be used to carry low priority (unprotected) traffic, which is preempted in the event of a failure. The need to signal and configure intermediate PXC's renders SPP slow compared to DPP, but SPP requires significantly less protection capacity.

Link protection schemes react more quickly to failures than do path protection schemes by initiating recovery from the nodes at either end of a failed link. In dedicated link protection, the end nodes of a failed link simply send/receive the traffic on the backup path. For shared link protection, the nodes at the end of the failed link signal and configure the intermediate nodes along the backup path after the failure. Link protection is also attractive in the sense that it allows decoupling of the routing and protection allocation problems. All links in the network can be protected, and traffic routed arbitrarily over the protected network without further concern for recovery. The price of this generality is additional capacity, and link protection is generally less efficient than path protection in terms of protection capacity [7, 2, 13]. Node failures can be protected using a link protection scheme by allocating backup routes around possible node failures. Some protection schemes offer a balance between capacity and recovery speed of the link/node and path approaches by dividing up the path protection approach into a number of domains (or segments) [10, 25]. For AONs, these shared protection schemes are more difficult to implement due to the limited ability for signaling. In addition, lack of wavelength conversion reduces their efficacy in sharing backup capacity.

The use of logical ring embeddings in a mesh network allows protection from link failures without reconfiguration delays. Protection schemes that use rings, such as cycle double covers [5], provide rapid recovery through the use of Automatic Protection Switching (APS), which automatically switches traffic over to protection fibers in the event of a failure. Ring-based solutions, however, pose difficult optimization problems of finding ring covers, and do not guarantee 100% connectivity between all pairs of nodes with protection against node failures [11] without complex extensions to enable protection paths to hop between rings. They are also inefficient in terms of protection capacity, requiring between 100% and 300% additional capacity [9]. The p-cycles work [9] solves the capacity problem for rings while providing fast recovery, using preconfigured cycles to protect against failures of links in both existing and newly designed networks. Flow p-cycles [20] extends the concept of p-cycles to path segments and provides protection for both link and node failures. The flow p-cycles algorithm is more capacity-efficient compared to the p-cycles as claimed in [20]. Both p-cycles and flow p-cycles leverage hop-to-

hop O-E-O conversion. An online algorithm for flow p-cycles has not yet been introduced. Therefore, a meaningful comparison requires an online implementation of flow p-cycles with consideration of wavelength continuity constraints, and is out of the scope of this paper.

Generalized loopback uses a flooding-based approach to provide link protection, giving a more flexible and efficient implementation than many ring algorithms while providing rapid, APS-like recovery [15]. As with many link protection schemes, generalized loopback decouples primary routing from protection allocation, and performs well when link loads are fairly uniform. With non-uniform link loading, however, generalized loopback requires substantially more protection capacity when compared with some other schemes. FBMR solves this problem by extending the concept of flooding to path based protection [12], as is discussed in detail in Sections 4 and 5.

### 3. Streams

We now present the Streams algorithm, which can be applied to any two-node connected network (or two-edge connected network, if node failures are not considered). Streams can be thought of as a virtually shared-DPP algorithm. It is like DPP in the sense that all PXC's are preconfigured at the time a lightpath is provisioned, and in the event of a failure, backup traffic is simply sent over the pre-established backup path, termed a *stream*. Preconfiguration enables the PXC's to switch over to backup routes without performing any decision making in the event of a failure, and aids our goal for rapid recovery. The key difference is that it allows sharing of a stream across different connections. Streams fall somewhere between DPP and SPP in terms of the existing classification of survivability techniques. Each connection is protected by a backup path, which must lie entirely on a single stream. All PXC's along a backup path are preconfigured to simply forward the backup traffic along the reserved  $\lambda$ -channels in a specific ingress to egress port setting (identical to 1:1 DPP). The mapping from ingress ports to egress ports at intermediate PXC's is maintained in the PXC configurations themselves, and is updated when lightpaths are provisioned or torn down or in the event of a failure. Recovery with Streams is much faster than with protection algorithms that use soft-reserved backup capacity (such as SPP), as no signaling or configuration of intermediate PXC's is required after a failure.

As with other path-based shared protection algorithms, there are two important constraints that must be obeyed in order to implement Streams. First, to be applicable in TONs, all paths must obey the wavelength continuity constraint. However, the primary and the protection paths for a connection may use different wavelengths. Second, only link-disjoint (or node-disjoint for node protection) primary

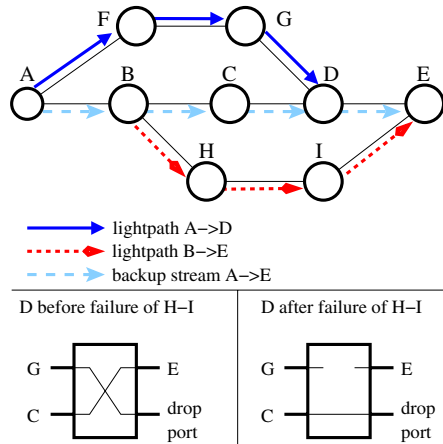
paths can be assigned to common portions of a single stream.

Streams offer similar capacity-recovery speed tradeoffs to FBMR, but requires less amplification and does not rely on the availability of low cost amplifiers with good noise figures, which is required to enable flooding in FBMR. FBMR recovery protocol also adds some complexity, and there are timing issues that require attention. What makes Streams more attractive is that the algorithm and the recovery protocol are extremely simple. It only adds a small computational overhead in routing, and the recovery protocol/implementation remains identical to DPP. The remainder of this section describes the failure detection and recovery process followed by our simple online Streams algorithm.

#### 3.1. Failure Recovery Process

The failure detection-recovery initiation process is simple: for a given channel, the PXC at the end of the lightpath detects a failure by monitoring the signal quality on the channel (many metrics are possible [4]). For simplicity, we assume bidirectional connections, which allows the receiver to initiate recovery as soon as it detects a failure. When a failure is detected, the end-nodes immediately redirect traffic to the assigned stream (preconfigured backup path). At the same time, the two nodes reconfigure themselves to begin listening to the preassigned ingress port for backup traffic. This reconfiguration step does not add to the recovery delay as it is performed simultaneously with redirection of the traffic (or immediately after redirecting the traffic and while waiting to receive from the other end). A node  $N$  may also be both an end node (source or destination) for a connection protected by a stream  $P$  and an intermediate node for the same stream  $P$ . In this case, after the traffic is redirected on a failure (sent over the egress port), the ingress port (if open) on the same stream is closed preventing possible signal collisions in the event of additional failures. The following example clarifies this idea.

In Figure 1, two connections  $A - D$  and  $B - E$  are protected by the backup stream  $A - E$ . On the figure, only single directions are shown for presentation purposes. The intermediate nodes  $B$ ,  $C$ , and  $D$  along the backup path are preconfigured (similar to DPP) to simply forward traffic from  $A$  to  $C$ ,  $B$  to  $D$ , and  $C$  to  $E$  respectively. Suppose link  $F - G$  fails. Upon detecting a failure on connection  $A - D$ , source node  $A$  redirects traffic for the path  $A \rightarrow D$  onto the backup stream  $A \rightarrow E$  by simply sending out the traffic towards  $B$ . Nodes  $B$ ,  $C$ , and  $D$  are configured to forward the traffic along the protection path  $A \rightarrow E$ .  $D$  (destination node), however, is also an end node for connection  $A - D$ , and therefore reconfigures itself to receive the traffic coming from  $C$  to the drop port for connection  $A - D$



**Figure 1. Streams setup.**

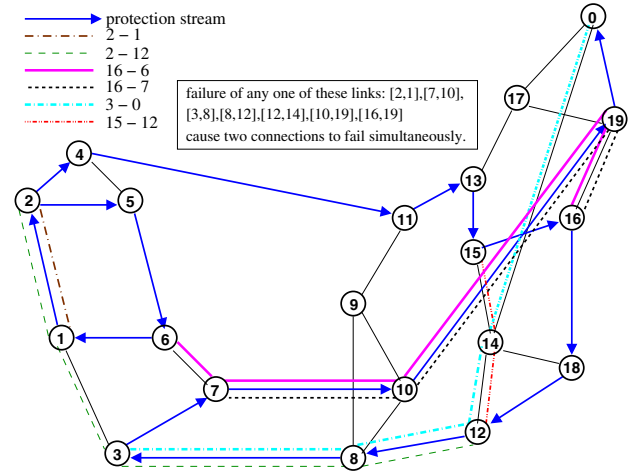
instead of sending it out to  $E$ . The role of  $A$  and  $D$  are simply switched for the traffic originating at  $D$  and terminating at  $A$ . Note that  $D$ , at the same time it reconfigures itself to properly receive traffic from  $C$ , closes the ingress/egress port from/to  $E$ . If there is a non-simultaneous multiple failure, say link  $H - I$ , and given enough time between the failures  $B$  knows that the backup resource is being used by a previous failure and does not attempt to over flood the backup path.  $E$  may or may not (depending on the failure detection and notification mechanism) attempt to send traffic to  $D$ , but the traffic will be dropped at  $D$ .

It is important to note that the Streams algorithm does not limit the detection process to signal monitoring described. Other approaches that allow quick detection and propagation of failure information to network nodes can also be used, allowing existing networks to readily adopt the Streams protection algorithm. For unidirectional connections, common recovery initiation techniques used by other algorithms, such as DPP, can be used.

### 3.2. Lightpath Provisioning

We utilize a simple heuristic for online provisioning of Streams. A greedy approach is used to perform joint search for best cost primary and backup path pairs. Figure 3 outlines our approach for implementation of an online Streams algorithm.

When provisioning a connection for protection against all single link or node failures, we must select a working and protection path pair that share no links or nodes other than the source and destination nodes. Protection is not provided for source and destination node failures; if such failures are handled at all (for any survivability technique), they generally make use of redundant node hardware and redundant connections to the optical network, both of which are orthogonal to the network recovery algorithm. Minimum cost working/protection path pairs can be found utilizing algo-



**Figure 2. Example showing how two connections that share a link can still be assigned to a single stream. The solid lines with arrows represent one direction of the protection stream, which has end points at node 2 and node 0.**

rithms similar to those typically used for different path protection algorithms with small modifications (constraints introduced by Streams).

The primary and protection paths for a connection must obey certain constraints. For example, both paths must obey the wavelength continuity constraint, meaning that the same wavelength must be used along the entire path. A pair of primary/protection paths can, however, use different wavelengths [14].

A stream may be shared across multiple connections that have common failure modes. Two (or more) connections that share a link (or a node) may use the same stream for protection if the connections rely on different parts of a stream for protection. For example, in Figure 2, there are a total of six connections provisioned on the network with a single stream providing protection for all six connections. Solid lines with arrows represent the stream with its preconfigured forwarding directions. Note that the stream starts at node 2 and ends at node 0. Node 2 is configured to forward incoming traffic from node 1 to node 4, i.e., there is no split at node 2. There are seven links where two connections are assigned, and a failure of any one of these links will simultaneously break two connections. It is easy to see that the stream is capable of handling all single link failure scenarios. The process of calculating the sharability of backup  $\lambda$ -channels is similar to most SPP schemes with one additional constraint.

The objective is to find the optimal primary and backup path pairs as connection requests arrive. An outline of three functions are shown in Figure 3. First, *route* establishes a connection between a source (*src*) and destination (*dst*)

<ul style="list-style-type: none"> <li>- A path is treated as an ordered set of links.</li> <li>- <math>P(src, dst)</math> - Set of shortest primary paths for all node pairs</li> <li>- <math>B(p_i, h)</math> - Set of backup paths corresponding to <math>p_i \in P</math> sorted by length in ascending order from shortest length paths to (shortest length + <math>h</math>) hop paths</li> <li>- <math>S</math> - Set of streams</li> <li>- <math>\lambda(s)</math> - Wavelength used by stream <math>s</math></li> <li>- <math>Free(\lambda)</math> - Set of free <math>\lambda</math>-channels on wavelength <math>\lambda</math></li> <li>- <math>SRG(s, l)</math> - Set of links that form a shared risk group. link <math>l</math> on stream <math>s</math> using <math>\lambda(s)</math> is used for recovery if a link in this set fails.</li> <li>- <math>compatible(s, b)</math> - Checks for compatibility between stream <math>s</math> and backup <math>b</math> on <math>\lambda(s)</math>. Specifically, this method checks for possible splits/merges that may arise as a result of adding <math>b</math> to <math>s</math>.</li> <li>- <math>find\_first\_fit\_wavelength(a)</math> - Finds the lowest wavelength <math>w</math> where path <math>a</math> fits. For some existing <math>w</math> if <math>a \subset Free(w)</math> then <math>\lambda(s)</math> is set to <math>w</math>. A new wavelength is allocated if <math>a</math> does not fit in any of the existing wavelengths.</li> </ul>	<pre> evaluate(<math>p, b</math>) {   mincost <math>\leftarrow</math> #links+1   stream <math>\leftarrow</math> empty   for all <math>s_i \in S</math> {     if (compatible(<math>s_i, b</math>)) {       cost <math>\leftarrow</math> 0       valid <math>\leftarrow</math> true       for all links <math>l_j \in b</math> {         if ((<math>p \cap SRG(s_i, l_j)</math>) = not empty)           valid <math>\leftarrow</math> false         if (<math>l_j \notin s_i</math>)           cost <math>\leftarrow</math> cost + 1       }       if (valid AND cost &lt; mincost) {         mincost <math>\leftarrow</math> cost         stream <math>\leftarrow</math> <math>s_i</math>       }     }   }   if (stream = empty) {     stream <math>\leftarrow</math> <math>s_0</math>     mincost <math>\leftarrow</math> length(<math>s_0</math>)   }   return [stream, mincost] } </pre>
<pre> route(<math>src, dst, ex\_hop</math>) {   mincost <math>\leftarrow</math> #links+1   for all <math>p_i \in P(src, dst)</math> {     for all <math>b_j \in B(p_i, ex\_hop)</math> {       [stream, cost] <math>\leftarrow</math> evaluate(<math>p_i, b_j</math>)       if ((cost + length(<math>p_i</math>)) &lt; mincost) {         mincost <math>\leftarrow</math> cost + length(<math>p_i</math>)         <math>p \leftarrow p_i</math>         <math>b \leftarrow b_j</math>         <math>s \leftarrow</math> stream       }     }   }   update_network(<math>p, b, s</math>) } </pre>	<pre> update_network(<math>p, b, s</math>) {   all_links <math>\leftarrow</math> <math>b \cup s</math>   new_links <math>\leftarrow</math> <math>b \setminus s</math>   if (new_links <math>\not\subset</math> Free(<math>\lambda(s)</math>))     <math>\lambda(s) \leftarrow</math> find_first_fit_wavelength(all_links)   for all links <math>l_i \in b</math>     <math>SRG(s, l_i) \leftarrow SRG(s, l_i) \cup p</math>   Free(<math>\lambda(s)</math>) <math>\leftarrow</math> Free(<math>\lambda(s)</math>) <math>\setminus</math> new_links   <math>s \leftarrow</math> all_links } </pre>

**Figure 3. Outline of the online Streams algorithm.**

pair. Shortest primary paths are used, and backup path lengths depend on the parameter *ex\_hop*, which represents the number of extra hops allowed for backup paths. Second, *evaluate* determines the cost of a candidate primary and backup path pair and checks for its validity. It also checks to see if an existing stream can be used or extended to save capacity. Finally, *update\_network* allocates resources for the new connection and updates the network status. The outline shown allocates connections such that they can be recovered from all single link failures. Performing the *SRG* intersection check for nodes computes protection routes that cover all single node failures.

## 4. Protection Algorithms

Different protection algorithms found in the literature are discussed in this section, and used to evaluate the relative tradeoffs. We provide a working description for each algorithm.

### 4.1. Dedicated and Shared Path Protection

DPP offers fast recovery with little or no data loss because no signaling is required between the source and the destination nodes after the failure. 1+1 DPP actively sends the backup traffic over the backup path, which is kept alive

during the entire lifetime of a connection. With 1:1 DPP, the backup path is only used when there is a failure, and therefore the backup path may be used to carry unprotected traffic. The unprotected traffic can be dropped in the event of a failure. This reuse makes 1:1 more efficient than 1+1 in terms of capacity; the tradeoff is the added delay in the recovery process of the 1:1 DPP. Once the traffic is placed on the 1:1 backup path, it takes about one propagation delay along the path to reach the destination node. 1+1 typically takes about 20ms for recovery with 0 to 20ms data loss, and 1:1 takes about 40ms to recover with 20ms of data loss [1, 12]. DPP is routed using shortest primary path and the shortest corresponding protection path. DPP is easy to implement in TONs and often is used [3].

In SPP, channels are chosen in advance, but not preconfigured. Instead, the end nodes of a lightpath signal the intermediate nodes along its protection path to configure the switches after the failure occurs. Because the switches are not configured to forward traffic from/to specific nodes, capacity reserved for protection can be shared among different connections that do not share links in the primary path. Protection capacity can also be used to carry arbitrary unprotected traffic. The need to signal and configure intermediate switches renders SPP slow compared to DPP, but SPP requires significantly less capacity compared to DPP [17]. SPP typically takes up to 90ms to recover with 90ms of data

loss [1, 12]. In simulating online routing, we use joint selection of link-disjoint primary and backup lightpaths to minimize the capacity cost in a manner similar to [24].

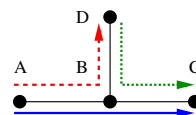
In practice, the signaling necessary to dynamically configure the intermediate switches after a failure can make implementation of SPP more difficult in TONs [3]. However, a simple implementation that leverages a bidirectional signaling wavelength per link may be utilized to solve the problem with the additional capacity of twice the number of links. Discussion of the details for such implementation is outside the scope of this work, and for the results shown in this paper, this cost is ignored. Ignoring this detail results in a tiny reduction in the cost for SPP, and does not affect the relative tradeoffs highlighted in this work.

## 4.2. Flooding-Based Mesh Restoration

Like Streams, FBMR achieves rapid recovery by eliminating signaling and configuration of intermediate nodes along the recovery path. FBMR also allows sharing of protection capacity. FBMR can be thought of as a generalized version of Streams where splitting/merging of streams is allowed. In FBMR, backup traffic is flooded over a protection wavelengths, and negative acknowledgments (NACKs) are used to tear down unnecessary paths (created by splitting) after recovery. By allowing signal splits, FBMR gains freedom in path selection, which results in a decrease in total capacity at the cost of placing signal splitters and amplifiers in the node architecture, and requiring some cleanup work after recovery completes. In addition, a digraph in FBMR, which is equivalent to a stream in Streams, cannot be shared by two connections that share a common link or node in the primary. Unlike Streams, even if the paths for the two connections utilize different portions of the digraph, they cannot share resources due to the possibility of collisions during the flooding process.

FBMR also utilizes preconfigured backup paths for recovery. It is similar to using APS at the wavelength level by the two end nodes of each connection. In SPP, backup resources are simply reserved for protection use, and after detecting a failure, signaling is used to setup a precalculated (or prequalified) path using those resources. In contrast, FBMR assigns specific switching configurations to the reserved backup channels, and preconfigures the protection paths. Upon detection of a failure, the end nodes of an affected connection redirect the traffic to the existing backup path. We provide a brief summary of the routing and recovery protocols for FBMR, and detailed descriptions can be found in [12].

In [12], a pair of precalculated conjugate digraphs (called flooding digraphs) are used to find recovery paths. Routing is restricted to a pair of precalculated flooding digraphs, which is used to route the backup paths. We extend FBMR



**Figure 4. Cross directional protection path sharing constraint for FBMR.**

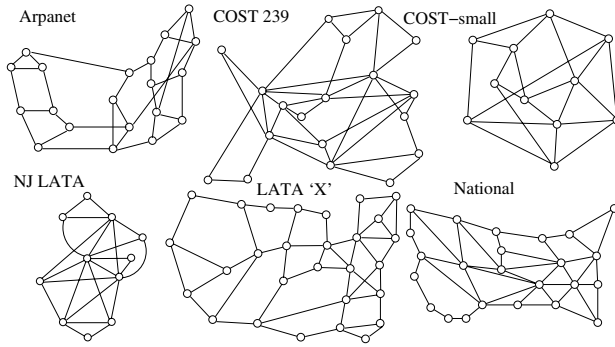
by allowing the use of the same path selection algorithm used for SPP. This extension improves the capacity requirements for FBMR and also makes the comparison between different algorithms more insightful. First, protection paths are computed as in SPP. These paths are then merged to preconfigure the protection paths and dynamically create a flooding digraph in the network. However, in order to be able to redirect the traffic (on the the flooding digraph) immediately after the detection of failure, as is done in DPP, one additional constraint must be met.

Merging protection paths while allowing backup traffic flooding adds an additional constraint that makes it slightly less efficient in terms of capacity compared to SPP. In FBMR, only a single direction can be utilized on a wavelength for each link. Consider the following example. Consider three protection paths  $A \rightarrow B \rightarrow C$ ,  $A \rightarrow B \rightarrow D$ ,  $D \rightarrow B \rightarrow C$  shown in Figure 4, and assume that there are no conflicts for the primaries associated with these paths—the primaries do not share a common failure mode (link or node). With SPP, paths can be merged on links AB and BC, which results in a total of four assigned wavelength-channels. With FBMR, paths cannot be merged on both links AB and BC (only one of the two can be merged) because node B must be preconfigured to either accept or forward traffic from/to node D on a given wavelength. On a single wavelength, it cannot be configured to both forward and receive traffic from the same node because this leads to a collision when bidirectional connections are routed. Therefore an additional wavelength must be used for either path  $A \rightarrow B \rightarrow D$  or path  $D \rightarrow B \rightarrow C$ , yielding a total of five wavelength-channels.

Our new routing algorithm performs a greedy optimization for capacity cost by searching for the best cost primary and protection path solutions. The recovery times for FBMR are identical to those of 1:1 protection. Flooding is effected immediately after the detection of a failure, incurring about 20 ms of data loss and takes about 40ms to complete recovery [12].

## 5. Evaluation of Algorithms

Our experiments are based on simulations of online provisioning with uniform full-mesh traffic demands. We assume that we have no knowledge of future demands. Each connection is assumed to be bidirectional. This approach is consistent with some of the evaluation models found in the



**Figure 5. Networks used for evaluation.**

Algorithms	Failure detection	Total recovery	Data loss
SPP	10 ms	50 to 90 ms	50 to 90 ms
FBMR	10 ms	40 ms	20 ms
Streams	10 ms	40 ms	20 ms
DPP (1:1)	10 ms	40 ms	20 ms
DPP (1+1)	10 ms	20 ms	0 to 20 ms

**Figure 6. Approximate data loss and recovery times.**

literature [17]. While, in practice, traffic demands may not be uniformly distributed, the study of uniformly distributed demands suffices to illustrate the characteristics of the different protection algorithms for comparison purposes. Approaches such as design-based routing (DBR) [19] that use offline ILP optimization to provision for online allocation decisions may reduce overall capacity requirements, but we do not expect that they will lead to substantial changes in the relative costs of algorithms. Also, DBR can be applied to any algorithm that shares protection resources. We used 200 randomly selected orderings of the demands and report the mean value measured over these orderings. For consistency, the same set of orderings is used in all experiments. We assign each  $\lambda$ -channel on each link a cost of one when calculating capacity. The total capacity cost is therefore the sum of the number of  $\lambda$ -channels reserved for both primary and protection paths. Six different networks are used to evaluate the different protection algorithms; COST-small (11 nodes, 24 links), NJ LATA (11,23), COST 239 (19,37) [16], National (24,44), LATA 'X' (28,47), Arpanet (20,32). These networks are ordered by average node degrees from highest to lowest.

### 5.1. Recovery time and data loss

We first discuss recovery times for the different algorithms. Although the primary metrics for evaluation of recovery algorithms have been protection capacity requirements and speed of recovery, most studies in the literature do not distinguish between overall recovery time and the ac-

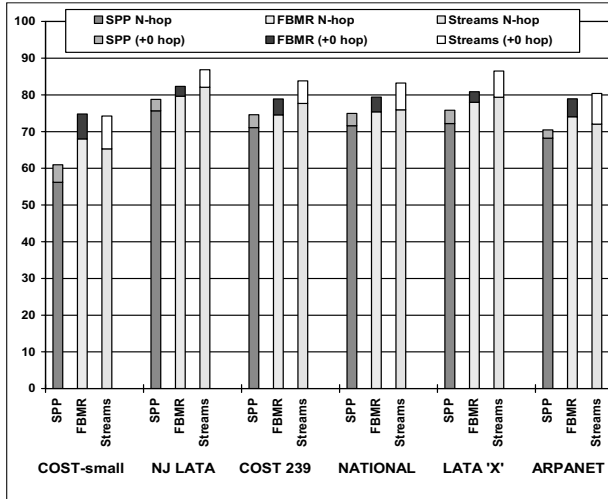
tual period of data loss [12, 1]. The time between a failure and initiation of recovery, for example, differs among the schemes discussed above. For link-based protection, failure detection requires approximately one link propagation delay, whereas path-based protection only reacts to a failure after a propagation delay on the whole path. Data is lost until backup traffic is sent out over the backup path, and may continue to be lost while switches along the backup path are reconfigured. The total period of data loss, therefore, includes both the time for detecting a failure and the time to set up the restoration path. As a result, more data is lost with path protection relative to link protection. With protection schemes such as FBMR and generalized loopback, flooding is initiated immediately upon detection of a failure, preventing further loss of data.

The numbers shown in Figure 6 are based on a few assumptions about the time required for basic operations [1], and were first presented in [12]. First, failure detection by the end nodes of a path takes about 10 ms; as most of this time is made up of propagation delay over half of the path, we assume for simplicity that total propagation delay on a path is about 20 ms. Next, switching a single PXC takes about 10 ms. Finally, signaling and configuration (usually uploading of maps) of intermediate PXC's take about 40 to 80 ms.

For the 1+1 case of DPP, recovery completes when the receiving node detects a failure and switches to the backup stream. With some effort, the backup stream can be delayed relative to the primary stream, allowing the receiver to avoid any data loss. With 1:1 DPP, there is some data loss until a sender switches to the protection path, and recovery is complete only after the data reach the receiver. As we have assumed bidirectional connections, we do not add a propagation delay to wait for the receiver to signal the sender; instead, both sides detect the failure and switch to the protection path. Total data loss is thus roughly 20 ms with this approach, and recovery time is one propagation delay longer, or about 40 ms. Streams has the same data loss and recovery time as the 1:1 DPP. The times for FBMR are also identical to those of 1:1 DPP [12]. With SPP, the PXC reconfiguration costs dominate both data loss and recovery time, bringing both to between 50 and 90 ms.

### 5.2. Capacity vs. Average protection path length

In this section, we report measurements based on shortest primary routing with varying protection path lengths (starting from shortest protection path to protection paths that are arbitrarily longer than the shortest protection path in single hop increments). Path lengths are computed in terms of number of hops. We also present the results using the *average path length expansion ratio* [11] (pl-expansion for short), which provides more insight in terms of the penalty



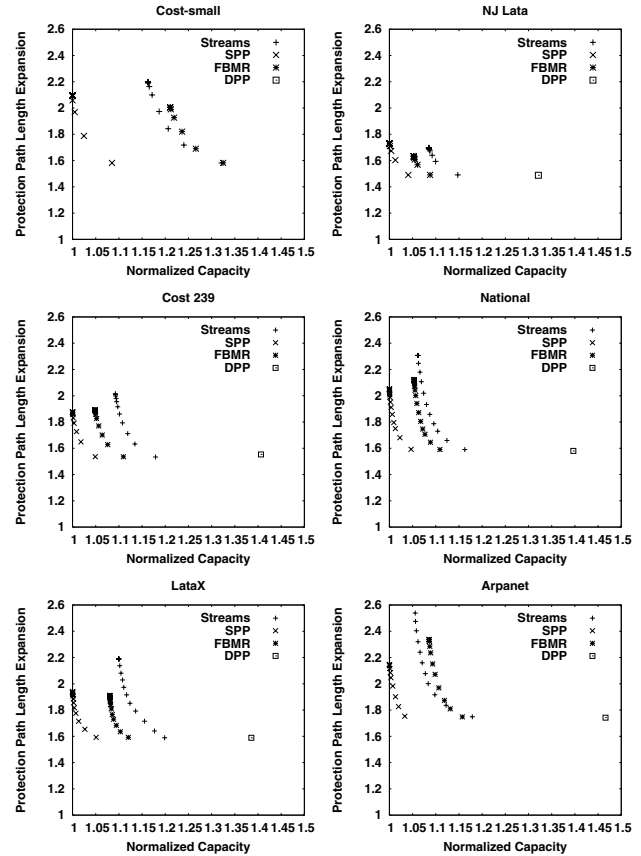
**Figure 7. Total capacity normalized to DPP with shortest primary paths for link failure protection.**

Network [node deg.]		Primary	DPP	SPP	FBMR	Streams
COST-small [4.4]	+0 hop	1.60	2.27	2.27	2.27	2.27
	N-hop	—	—	2.99	2.83	3.12
NJ LATA [4.2]	+0 hop	1.75	2.31	2.32	2.32	2.32
	N-hop	—	—	2.68	2.52	2.65
COST 239 [3.9]	+0 hop	2.24	3.16	3.17	3.17	3.17
	N-hop	—	—	3.81	3.78	4.04
National [3.7]	+0 hop	2.90	4.17	4.20	4.20	4.19
	N-hop	—	—	5.20	5.30	5.70
LATA 'X' [3.4]	+0 hop	3.28	4.67	4.69	4.70	4.68
	N-hop	—	—	5.61	5.34	6.10
Arpanet [3.2]	+0 hop	2.75	4.14	4.16	4.14	4.15
	N-hop	—	—	4.90	5.10	5.58

**Figure 8. Average protection path lengths in hops with shortest primaries.**

an algorithm pays due to an increase in path lengths. Path length expansion ratio for a connection is the ratio between the allocated protection path over the shortest path length for a given source/destination pair. Average path length expansion ratio (pl-expansion), then, is the average over all connections. With shortest primary paths, pl-expansion is also the average ratio between backup and primary paths. Considering tradeoffs in terms of path lengths is interesting because it provides insights as to how much delay is introduced. In addition, in the case of TONs, signal qualities as well as possible amplification requirements can be seen. One important issue, often overlooked, with extending paths is that an increase in path lengths (of some number of hops or distance) more greatly affects shorter connections. By capturing this effect, pl-expansion provides information about the actual penalty an algorithm may incur in optimizing for capacity.

**5.2.1. Link failure protection** Figures 7 and 8 show capacity requirements and average path lengths for different networks and algorithms configured to handle all single link



**Figure 9. Normalized capacity and pl-expansion for SPP, FBMR and Streams. The lowest points represent shortest (+0 hop) backup with data points corresponding to additional hops from the shortest.**

failures (node failure coverage is discussed in the following section). Results corresponding to shortest primary paths, and shortest backup (+0 hop) and arbitrary length backup (N-hop) solutions are shown. Figure 7 shows total capacity normalized to that of DPP. Evaluation results show the relative efficiency between different algorithms. SPP, FBMR and Streams incur a small overhead in path length expansion, with a significant reduction in capacity compared to DPP. Our results confirm that SPP is the most capacity-efficient, as is well understood in the literature. FBMR and Streams both provide faster recovery and minimal data loss with a small capacity overhead compared to SPP.

Figure 9 shows total capacity (normalized to best SPP solution) and the pl-expansion. SPP, FBMR, and Streams can be operated with flexibility in terms of these two measures by varying the maximum allowed backup path lengths. We use the *number of additional hops* compared to the shortest possible backup path to control the tradeoff between capacity and pl-expansion. The full range of solutions from shortest backup (+0 Hop) to arbitrary



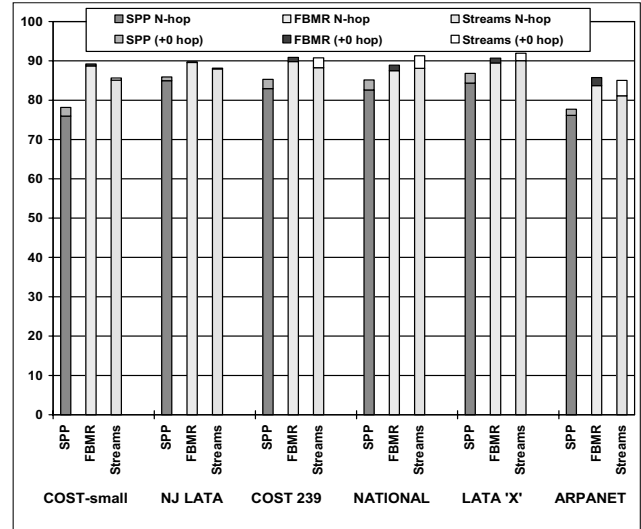
trary length backup (N-hop) is shown. The lowest (and right most) data points for each algorithm represent +0 hop solutions with single hop increments to the maximum allowed backup path length shown to the left in series. Naturally, different networks and algorithms have different number of data points as the effect of allowing extra hops for backup paths plateaus at different points. The top most data point, then, corresponds to N-hop solutions (where N is different for different networks and algorithms). For DPP, only shortest backup solutions are presented, as increasing backup paths only makes it less efficient. The pl-expansion value for DPP reflects an absence of disjoint shortest paths for many pairs.

For most networks, one or two extra hops in the backup allows for a substantial reduction in capacity while maintaining pl-expansion overhead relatively low. This phenomenon is somewhat intuitive given that the possible choices for paths between two nodes quickly increase with increased maximum allowed length. It is interesting to see that Streams outperforms FBMR in networks that are three-connected (Cost-small and Arpanet). Given that Streams does not allow backup path splitting/merging, this may be counterintuitive, but due to the increase in cross-SRG sharing flexibility (explained in the example in Section 4.2), Streams is able to find better solutions (compared to FBMR) for these networks. For the Cost-small network, Streams outperforms FBMR throughout the entire range of solutions, and for the Arpanet network, allowing one or more extra hops on the backup paths enables Streams to find solutions that are more capacity-efficient compared to FBMR.

Our experiments showed that allowing longer primary paths does not affect the relative results between different algorithms. Detailed results do not provide additional information other than the fact that each algorithm may improve slightly in terms of capacity due to the increase in the number of candidate primary and backup path pairs, and, therefore, have been omitted.

**5.2.2. Node failure protection** Figure 10 shows the total required capacity (normalized to DPP) and Figure 11 shows the path lengths (and pl-expansion) for the different protection algorithms with node protection. SPP, FBMR and Streams all incur some capacity overhead when node failures are considered. The overhead shown by our results is expected as there are fewer number of node-disjoint paths compared to link-disjoint paths, which limits the number of choices for primary and backup path pairs.

With the added overhead, relative efficiencies between algorithms are closer compared to link failure protection. Streams outperforms FBMR in the majority of the networks tested. With a more limited number of choices for path pairs, protection path lengths tend to be shorter (for other than +0 hop solutions) as longer paths may be chosen in



**Figure 10. Total capacity normalized to DPP with shortest primary paths for link and node failure protection.**

Network [node deg.]	hops	DPP	SPP	FBMR	Streams
COST-small [4.4]	+0	2.27 (1.58)	2.27 (1.58)	2.27 (1.58)	2.27 (1.58)
	N	–	2.66 (1.89)	2.30 (1.61)	2.45 (1.70)
NJ LATA [4.2]	+0	2.31 (1.49)	2.31 (1.49)	2.31 (1.49)	2.31 (1.49)
	N	–	2.41 (1.57)	3.32 (1.49)	2.36 (1.52)
COST 239 [3.9]	+0	3.24 (1.56)	3.25 (1.56)	3.24 (1.56)	3.25 (1.56)
	N	–	3.61 (1.78)	3.45 (1.61)	3.65 (1.76)
National [3.7]	+0	4.15 (1.58)	4.13 (1.58)	4.12 (1.57)	4.17 (1.59)
	N	–	4.63 (1.83)	4.36 (1.70)	4.72 (1.82)
LATA 'X' [3.4]	+0	4.74 (1.61)	4.76 (1.61)	4.75 (1.61)	4.75 (1.61)
	N	–	5.23 (1.83)	4.96 (1.71)	5.31 (1.86)
Arpanet [3.2]	+0	4.14 (1.75)	4.15 (1.75)	4.14 (1.75)	4.15 (1.75)
	N	–	4.64 (2.03)	4.41 (1.89)	4.92 (2.13)

**Figure 11. Average protection path lengths in hops with shortest primaries. PL-expansion numbers are shown inside the parentheses.**

link failure scenarios to optimize for capacity. The relative trends between different length solutions for the different algorithms for the capacity versus pl-expansion plots were similar to the link failure scenario (graphs are not included as they do not provide additional information).

## 6. Conclusion

Most survivability algorithms developed for optically opaque networks are not readily applicable to all-optical networks due to the effect of imposing the  $\lambda$ -continuity constraint to different algorithms, which lower their efficiency (or make them inapplicable).

The Streams algorithm is attractive because it can be readily implemented in existing networks and can also coexist with other algorithms such as SPP and DPP in the same network. In contrast, FBMR requires some extra equipment to support digraph flooding, and its application

to existing networks is likely to be limited by signal amplification constraints.

Our results show that Streams is efficient in terms of capacity requirements, allowing us to meet the goal of designing survivability techniques that are capacity-efficient and incur little data loss. The Streams algorithm's recovery speed is identical to 1:1 DPP, but requires no more than 87% of the total capacity in protection all single link failures for the six networks evaluated in this paper. SPP is understood to be the most capacity-efficient, and requires 79% or less total capacity compared to DPP.

The online heuristic for FBMR is surprisingly effective in reducing capacity requirements and also path lengths. FBMR can provide full link failure protection with less than 82% of protection the capacity compared to DPP while utilizing short protection paths that are comparable to DPP. In most cases FBMR requires less than 10% additional capacity compared to SPP, which is the lower bound for FBMR. Both Streams and FBMR provides DPP like recovery time while utilizing protection capacity comparable to SPP, and offer an interesting tradeoff between recovery speed, data loss, capacity, and path lengths.

Although this paper focuses on TONs, Streams can be applied to optically transparent networks using the same failure detection/recovery protocol. We expect the relative tradeoffs compared to other algorithms to be similar to the results presented in this work.

## References

- [1] E. Bouillet, K. Kumaran, G. Liu, and I. Saniee. Wavelength usage efficiency versus recovery time in path-protected dwdm mesh networks. In *Proc. of IEEE/OSA OFC*, 1998.
- [2] B. Caenegem, B. Wauters, and P. Demeester. Spare capacity assignment for different restoration strategies in mesh survivable networks. In *Proc. of IEEE ICC*, volume 1, pages 288–92, 1997.
- [3] S. Chaudhuri, E. Bouillet, and G. Ellinas. Addressing transparency in dwdm mesh survivable networks. In *Proc. of IEEE/OSA OFC*, 2002. Tu05.
- [4] Y. C. Chung. Optical monitoring techniques for wdm networks. In *2000 Digest of the LEOS Summer Topical Meeting on Broadband Optical Networks*, pages 43–4, July 2000.
- [5] G. Ellinas, A. G. Hailemariam, and T. E. Stern. Protection cycles in mesh wdm networks. *IEEE JSAC*, 18(10), Oct. 2000.
- [6] H. Fujii and N. Yoshikai. Double search self-healing algorithm and its characteristics. *Electronics and Communications in Japan-Part 1*, 77(3):75–8, 1994.
- [7] A. Fumagalli and L. Valcarenghi. The preplanned weighted restoration scheme. In *IEEE Workshop on High Performance Switching and Routing*, pages 36–41, 2001.
- [8] W. D. Grover. The selfhealing network. In *Proc. of IEEE GLOBECOM*, volume 2, pages 1090–5, 1987.
- [9] W. D. Grover and D. Stamatelakis. Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network reconfiguration. In *Proc. of IEEE ICC*, volume 1, pages 537–43, 1998.
- [10] P. Ho and H. Mouftah. Slsp: A new path protection scheme for the optical internet. In *Proc. of IEEE/OSA OFC*, 2001.
- [11] S. Kim and S. S. Lumetta. Addressing node failures in all-optical networks. *OSA Journal of Optical Networking*, 1(4):154–63, April 2002.
- [12] S. Kim and S. S. Lumetta. Restoration of all-optical mesh networks with path-based flooding. *IEEE/OSA JLT*, 21(11), November 2003.
- [13] M. Kodialam and T. Lakshman. Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information. In *Proc. of IEEE INFOCOM*, volume 1, pages 376–85, 2001.
- [14] K. Kumaran and I. Saniee. Shared recovery in transparent optical networks. In *Proc. of the 39<sup>th</sup> Annual Allerton Conf. on Comm., Control, and Computing*, Oct. 2001.
- [15] M. Médard, S. G. Finn, R. A. Barry, W. He, and S. S. Lumetta. Generalized loop-back recovery in mesh networks. *IEEE Trans. on Networking*, 10(1):153–64, February 2002.
- [16] M. J. O'Mahony, D. Simeonidou, A. Yu, and J. Zhou. The design of a european optical network. *IEEE/OSA JLT*, 13(5):817–29, 1995.
- [17] R. Ramamurthy, Z. Bogdanowicz, S. Samieian, D. Saha, B. Rajagopalan, S. Sengupta, S. Chaudhuri, and K. Bala. Capacity performance of dynamic provisioning in optical networks. *IEEE/OSA JLT*, 19(1):40–8, 2001.
- [18] S. Ramamurthy, L. Sahasrabudde, and B. Mukherjee. Survivable wdm mesh networks. *IEEE/OSA JLT*, 21(4):870–83, April 2003.
- [19] I. Saniee, I. Widjaja, A. Elwalid, , and D. Mitra. Design-based routing for online traffic engineering in connection-oriented networks. In *Proc. of the 40<sup>th</sup> Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2002.
- [20] G. Shen and W. D. Grover. Extending the p-cycle concept to path segment protection for span and node failure recovery. *IEEE JSAC*, 21(8):1306–19, Oct. 2003.
- [21] T. E. Stern and K. Bala. *Multiwavelength Optical Networks; A Layered Approach*. Prentice-Hall, Upper Saddle River, NJ, 2000.
- [22] X. Su and C. Su. An online distributed protection algorithm in wdm networks. In *Proc. of IEEE ICC*, volume 5, pages 1571–5, 2001.
- [23] T. H. Wu. A passive protected self-healing mesh network architecture and applications. *IEEE/ACM Trans. on Networking*, 2(1):40–52, February 1994.
- [24] C. Xin, Y. Ye, S. Dixit, and C. Qiao. A joint working and protection path selection approach in wdm optical networks. In *Proc. of IEEE GLOBECOM*, volume 4, pages 2165–8, 2001.
- [25] D. Xu, Y. Xiong, and C. Qiao. Novel algorithms for shared segment protection. *IEEE JSAC*, 21(8):1320–31, Oct. 2003.
- [26] Z. Zhang, J. Fu, D. Guo, and L. Zhang. Lightpath routing for intelligent optical networks. *IEEE Network*, pages 28–35, July/August 2001.