# Traffic Grooming for Survivable WDM Networks – Shared Protection

Canhui (Sam) Ou, Keyao Zhu, Hui Zang, Laxman H. Sahasrabuddhe, and Biswanath Mukherjee

*Abstract*— **This paper investigates the survivable traffic-grooming problem for optical mesh networks employing wavelength-division multiplexing (WDM). In the dynamic-provisioning context, a typical connection request may require bandwidth less than that of a wavelength channel, and it may also require protection from network failures, typically fiber cuts. Based on a generic grooming-node architecture, we propose three approaches—protection-at-lightpath (*PAL*) level, mixed protection-at-connection (*MPAC*) level, and separate protection-at-connection (*SPAC*) level—for grooming a connection request with shared protection. In shared-mesh protection, backup paths can share resources as long as their corresponding working paths are unlikely to fail simultaneously. These three schemes explore different ways of backup sharing, and they trade off between wavelengths and grooming ports. Since the existence version of the problem for provisioning one connection request with shared protection is $\mathcal{NP}$-complete, we propose effective heuristics. Our findings are as follows. Under today's typical connection-bandwidth distribution where lower bandwidth connections outnumber higher bandwidth connections, 1) it is beneficial to groom working paths and backup paths separately, as in PAL and SPAC; 2) separately protecting each individual connection—i.e., SPAC—yields the best performance when the number of grooming ports is sufficient; and 3) protecting each specific lightpath—i.e., PAL—achieves the best performance when the number of grooming ports is moderate or small.**

*Index Terms*— **Optical network, WDM, lightpath, provisioning, fault management, shared protection, grooming.**

## I. Introduction

While the transmission rate of a wavelength channel is high (typically STS-192 today and expected to grow to STS-768 in the near future), the bandwidth requirement of a typical connection request can vary from the full wavelength capacity down to STS-1 or lower. To efficiently utilize network resources, sub-wavelength-granularity connections can be groomed onto direct optical transmission channels, or lightpaths[1]. Meanwhile, the failure of a network element can cause the failure of several lightpaths, thereby leading to large data and revenue loss. Fault-management schemes such as protection are essential to survive such failures.

Different low-speed connections may request different bandwidth granularities as well as different protection schemes (dedicated or shared). How to efficiently groom such low-speed connections while satisfying their protection requirements is the main focus of our investigation. Since shared protection is more resource efficient than dedicated protection due to backup sharing, we focus on the problem of dynamic low-speed connection provisioning with shared protection against single-fiber failures. Single-fiber failures are the predominant type of failures in communication networks. Node failures are not considered here because most nodal equipments are 1+1 protected.

The rest of this paper is organized as follows. The remainder of this section provides background information on traffic grooming and protection. Section II presents a generic grooming-node architecture. Section III formally states the problem. Section IV presents three approaches—protection-at-lightpath (PAL) level, mixed protection-at-connection (MPAC) level, and separate protection-at-connection (SPAC) level—and provides a qualitative comparison. Section V presents heuristic algorithms for PAL, MPAC, and SPAC. Section VI compares the three schemes under different network configurations. As some customers may desire dedicated protection for fast protection switching, Section VII discusses traffic grooming with dedicated protection. Section VIII concludes this study.

### A. Traffic Grooming

Traffic grooming refers to the problem of efficiently packing low-speed connections onto high-capacity lightpaths to better utilize network resources [1],[2].

Traffic grooming on SONET/WDM ring networks has been extensively studied; see, for example, [3]-[7]. In WDM mesh networks, the traffic-grooming problem has mainly addressed static traffic where a traffic demand matrix is known a priori [8],[9]. Online approaches for traffic grooming in WDM mesh networks have been recently reported in [10]-[12]. The work in [10] proposes a call-admission-control algorithm to address the capacity-fairness issue, i.e., a connection request with higher bandwidth requirement is more likely to be blocked than a connection request with lower bandwidth requirement. The work in [11] proposes different grooming policies and route-computation algorithms for different network states. The work

This work has been supported in parts by NSF Grant Nos. ANI-9805285 and ANI-0207864, and by a research gift from Sprint. A short, summarized version of this work was presented in OFC 03 conference, March 2003.

C. Ou, K. Zhu, and B. Mukherjee are with the Department of Computer Science, University of California, Davis, CA 95616, USA (e-mail: {ouc, zhuk, mukherje}@cs.ucdavis.edu).

H. Zang is with Sprint Advanced Technology Labs, Burlingame, CA 94010, USA (e-mail: hzang@sprintlabs.com).

L. H. Shasrabuddhe is with SBC Services, Inc., San Ramon, CA 94583, USA (e-mail: ls9526@sbc.com).

[1] We distinguish the terms "lightpath" and "connection" as follows. The bandwidth requirement of a lightpath is the full wavelength capacity (STS-192 in our present study). The bandwidth requirement of a connection can be any quantized value no more than the full wavelength capacity. Later in our examples and results, we use the quantized values STS-1, STS-3c, STS-12c, STS-48c, and STS-192c for illustration purposes since these values have been widely used in current systems (the "c" after the number implies this is a contiguous block of STS-1s that are part of the same connection). We use the term "STS-n" to refer to the payload carried within an OC-n optical interface ($n = 1, 3, 12$, etc.).

in [12] develops an algorithm for dynamically grooming low-speed connections to meet different traffic-engineering objectives based on the generic graph model proposed in [9]. Please see [1] for an extensive review on traffic grooming.

### *B. Protection*

Protection is a proactive procedure in which spare capacity is reserved during connection setup [13],[14]. A path that carries traffic during normal operation is known as a *working* path. When a working path fails, the connection is rerouted over a *backup* path. Below, we review a portion of the closely related work on survivable lightpath provisioning and multiprotocol label switching (MPLS) connection provisioning with shared protection. Please refer to [15] for an extensive review.

*1) Survivable WDM lightpath provisioning:* Online algorithms for survivable lightpath provisioning in WDM networks have been reported in [16]-[18]. The work in [16] presents short leap shared protection (SLSP), which divides a working path into overlapped segments and protects each segment individually. The work in [17] proposes a routing approach, which first computes $K$ working candidate routes, then computes $K$ link-disjoint paths as backup candidate routes, and selects the link-disjoint path-pair of minimum cost. The work in [18] proposes the sharing of primary lightpaths and backup lightpaths under the assumption that connection-holding time is shorter than the mean time between failures.

*2) Restorable MPLS tunnel provisioning:* Online algorithms for dynamic routing of restorable bandwidth-guaranteed connections in a MPLS network have been reported in [19]-[21]. Although these papers are devoted to the MPLS context, their basic ideas—with appropriate variations, e.g., quantized bandwidth granularities and grooming constraints—are applicable to the survivable traffic-grooming problem with shared protection in a WDM mesh network with full wavelength conversion at each node. The work in [19] proposes an algorithm which, for a connection request, selects the minimum-cost path as the working path and computes the minimum-cost link-disjoint path as backup path based on a “bucket”-like [22],[23] link metric. The work in [20] first develops integer linear programs (ILPs) to route a connection request under shared-path protection constraints with no, complete, or partial information of existing connections. The authors then provide a heuristic for routing with partial information. The work in [21] describes distributed partial information management (DPIM) schemes for maintaining aggregated information to provision bandwidth-guaranteed connections with shared-path protection.

### *C. Survivable Traffic Grooming*

The survivable traffic-grooming problem, in which sub-wavelength-granularity connections need to be protected, is a relatively unexplored territory.

Given a static traffic matrix and the protection requirement of each connection request, the work in [24] presents an integer linear program and a heuristic for satisfying the bandwidth and protection requirements of all the connection requests while minimizing the network cost in terms of transmission cost and switching cost.

For dynamically establishing low-speed connection requests with shared protection, the work in [25] presents mixed working-backup grooming policy (MGP) and segregated working-backup grooming policy (SGP). With both schemes employing fixed-alternate routing [26], the work focuses on the effect of different wavelength-assignment algorithms and different topologies.

### *D. Our Proposal*

We propose three approaches—protection-at-lightpath (PAL) level, mixed protection-at-connection (MPAC) level, and separate protection-at-connection (SPAC) level—for dynamically provisioning shared-protected subwavelength-granularity connection requests against single-fiber failures. We investigate their characteristics under a generic grooming-node architecture and design efficient heuristics. Our work differs from previous work in that we focus on route computation, the impact of different backup-sharing approaches, and the tradeoff between wavelength and grooming capacity.

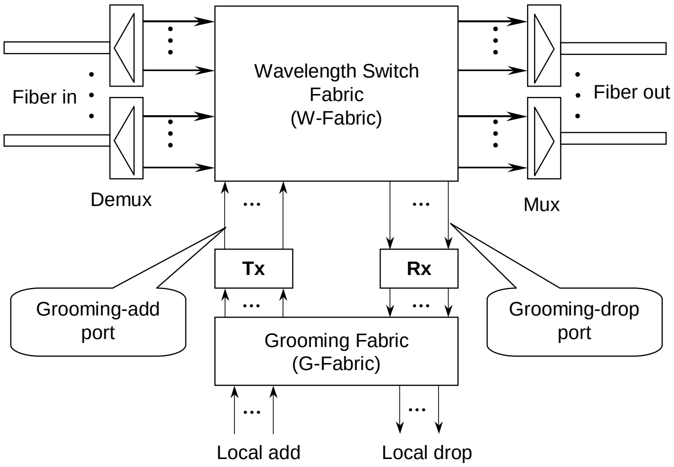


Fig. 1. A simplified grooming-node architecture.

## II. Grooming-Node Architecture

In order to support traffic grooming, a network node should be able to switch traffic at wavelength granularity and finer granularity. Figure 1 shows the logical view of a simplified grooming-node architecture.

This hierarchical grooming node consists of a wavelength-switch fabric (W-Fabric) and a grooming fabric (G-Fabric). The W-Fabric performs wavelength routing; the G-Fabric performs multiplexing, demultiplexing, and switching of low-speed connections. A portion of the incoming wavelengths to the W-Fabric can be dropped to the G-Fabric through the grooming-drop ports for sub-wavelength-granularity switching. The groomed traffic can then be added to the W-Fabric through the grooming-add ports. The number of grooming ports determines the grooming capacity of a node (we assume that there are equal number of grooming-add and grooming-drop ports).

Later, we shall investigate the impact of grooming capacity (number of grooming ports) on the network performance.

Even though crossconnects capable of full grooming–i.e., G-Fabrics–are preferable to network operators today, crossconnects capable of wavelength switching–i.e., W-Fabrics–are expected to be desirable as traffic continues to grow in the future. The G-Fabrics deployed today are unlikely to go away when W-Fabrics are deployed due to economic concerns. One way of effectively utilizing both G-Fabrics and W-Fabrics could be to interconnect a W-Fabric and a G-Fabric through transponders, as shown in Fig. 1.

As a special case, if the number of grooming ports at a node is equal to the number of incoming wavelengths to its W-Fabric, then this grooming node can switch the entire incoming traffic at STS-1 level, as is the case in today's state-of-the-art opaque (i.e., switching with optical-to-electronic-to-optical conversion) intelligent optical switches from many vendors.

While our approaches apply to both wavelength-continuous and wavelength-convertible networks, we hereafter assume without loss of generality that the network has full wavelength-conversion capability.

## III. Problem Statement

We first define the notations and then formally state the dynamic connection-provisioning problem. A network is represented as a weighted, directed graph $G = (V, E, C, \lambda, P)$, where $V$ is the set of nodes, $E$ is the set of unidirectional fibers (referred to as links), $C : E \rightarrow R^+$ is the cost function for each link (where $R^+$ denotes the set of positive real numbers), $\lambda : E \rightarrow Z^+$ specifies the number of wavelengths on each link (where $Z^+$ denotes the set of positive integers), and $P : V \rightarrow Z^+$ specifies the number of grooming ports at each node.

A connection request is represented as a quadruple $\langle s, d, B, t_h \rangle$, which specifies the source node, the destination node, the bandwidth requirement, and the holding time in this order. In this study, every connection needs to be protected, and the backup resources can be shared.

We now formally state the dynamic connection-provisioning problem as follows: Given the current network state (which includes the network topology as a weighted digraph $G$, existing lightpath/connection information [e.g., routes and wavelengths, etc.], wavelength usage, and grooming-port usage), route each new connection request with respect to its bandwidth and protection requirement (shared protection) while minimizing the incremental cost in terms of the total cost of the working and backup paths under the assumptions that existing connections cannot be disturbed and information about future arrivals is not known.

The existence version of the above problem for provisioning one connection request under the current network state is $\mathcal{NP}$-complete. This is because a special case of this problem, in which the number of grooming ports is sufficient and every connection request requires full wavelength capacity, has been proven to be $\mathcal{NP}$-complete in [27]. Thus, practical heuristics are needed.

## IV. Proposed Schemes

To provision a connection request, there are two types of resource constraints—wavelengths and grooming ports. Typically, the more the number of wavelengths the network has, the less the number of grooming ports a node needs, and vice versa.

We propose three schemes—protection-at-lightpath (PAL) level, mixed protection-at-connection (MPAC) level, and separate protection-at-connection level (SPAC)—for provisioning shared-protected connection requests. These three schemes explore different ways of backup routing and the tradeoff between wavelengths and grooming ports.

Below, we shall illustrate the three schemes via an example. For the initial network configuration shown in Fig. 2, every edge corresponds to a bidirectional fiber; each fiber has two wavelengths; the wavelength capacity is STS-192; every node has three grooming ports (T and R represent the number of available grooming-add and grooming-drop ports, respectively).

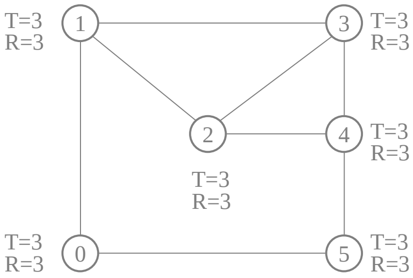


Fig. 2. Example: initial network configuration.

### A. Protection-at-Lightpath (PAL) Level

*1) Basic idea:* PAL provides end-to-end protection with respect to lightpath. Under PAL, a connection is routed through a sequence of protected lightpaths, or $p$-lightpaths. A $p$-lightpath has a *lightpath* as working path and a link-disjoint *path* as backup path. For example, in Fig. 3(a), $p$-lightpath $l_1$ has lightpath $\langle 0, 1, 2 \rangle$ as working path and path $\langle 0, 5, 4, 2 \rangle$ as backup path. Please note the differences between the working path and the backup path of a $p$-lightpath. The working path of a $p$-lightpath is set up as a lightpath during normal operation. Therefore, as a lightpath does, the working path consumes a grooming-add port at the source node and a grooming-drop port at the destination node of a $p$-lightpath; and the working path of a $p$-lightpath bypasses any intermediate nodes along its path. However, the backup path of a $p$-lightpath is not set up as a lightpath during normal operation. Therefore, the backup path of a $p$-lightpath does not consume any grooming port; and wavelengths along a backup path are only reserved. In case the working path fails, protection switching occurs at lightpath level and the backup path is set up as a lightpath by utilizing the grooming ports previously used by the working path.

Two $p$-lightpaths can share wavelengths along common backup links if their working paths are link-disjoint. Clearly, a connection routed under PAL survives from single-link failures since each $p$-lightpath survives from single-link failures by definition. Since protection occurs at lightpath level, PAL has the advantages of low implementation complexity and low signaling overhead when a failure occurs. This will be further elaborated in Section IV-D. Below, we illustrate PAL in more detail by provisioning three connection requests.

*2) Example:* Upon the arrival of the first connection request $c_1, \langle 0, 2, \text{STS-12c}, t_1 \rangle$, one way of provisioning $c_1$ under PAL is shown in Fig. 3(a). Connection $c_1$ is routed via $p$-lightpath $l_1$, which has lightpath $\langle 0, 1, 2 \rangle$ as working path and path $\langle 0, 5, 4, 2 \rangle$ as backup path. $p$-lightpath $l_1$ consumes a grooming-add port at node $0$ and a grooming-drop node at node $2$. The free capacity of $p$-lightpath $l_1$ is STS-180.

Suppose that $c_1$ remains in the network when the second connection request $c_2, \langle 0, 3, \text{STS-3c}, t_2 \rangle$, arrives. One way of provisioning $c_2$ under the current network state is shown in Fig. 3(b) Connection $c_2$ is routed via $p$-lightpath $l_2$, which has lightpath $\langle 0, 1, 3 \rangle$ as working path and path $\langle 0, 5, 4, 3 \rangle$ as backup path. $p$-lightpath $l_2$ consumes a grooming-add port at node $0$ and a grooming-drop port at node $3$. The free capacity of $p$-lightpath $l_2$ is STS-189. Two wavelengths need to be reserved along links $\langle 0, 5 \rangle$ and $\langle 5, 4 \rangle$ because (1) the working paths of $p$-lightpaths $l_1$ and $l_2$ traverse common link $\langle 0, 1 \rangle$, and (2) protection occurs at lightpath level, i.e., backup sharing only occurs at wavelength level.

Suppose that $c_1$ and $c_2$ remain in the network when the third connection request $c_3, \langle 4, 3, \text{STS-48c}, t_3 \rangle$, arrives. One way of provisioning $c_3$ under the current network state is shown in Fig. 3(c) Connection $c_3$ is routed via $p$-lightpath $l_3$, which has lightpath $\langle 4, 3 \rangle$ as working path and path $\langle 4, 2, 3 \rangle$ as backup path. $p$-lightpath $l_3$ consumes a grooming-add port at node $4$ and a grooming-drop port at node $3$. The free capacity of $p$-lightpath $l_3$ is STS-144. Please note that the backup paths of $p$-lightpaths $l_1$ and $l_3$ share the wavelength reserved along link $\langle 4, 2 \rangle$.

### *B. Mixed Protection-at-Connection (MPAC) Level*

*1) Basic idea:* MPAC and SPAC provide end-to-end protection with respect to connection. Under MPAC, a connection is routed via link-disjoint working and backup paths, each of which traverses a sequence of lightpaths. A lightpath traversed by a working path utilizes a portion of its capacity to carry traffic for that working path during normal operation. A lightpath traversed by a backup path reserves part of its capacity for that backup path. The backup capacity a lightpath reserves can be shared among multiple backup paths provided that their corresponding working paths are link-disjoint. In this context, “mixed” means that the capacity of one wavelength can be utilized by *both* working paths and backup paths; “separate”, on the other hand, means that the capacity of a wavelength can be utilized by *either* working paths or backup paths, but not both. MPAC seems to be the most intuitive approach since it deals with individual connections and therefore can pack connections efficiently. Later in Section IV-D, we shall show that it may not achieve the best performance due to the intricacy of backup sharing. Below, we illustrate MPAC in more detail using the same example as before.

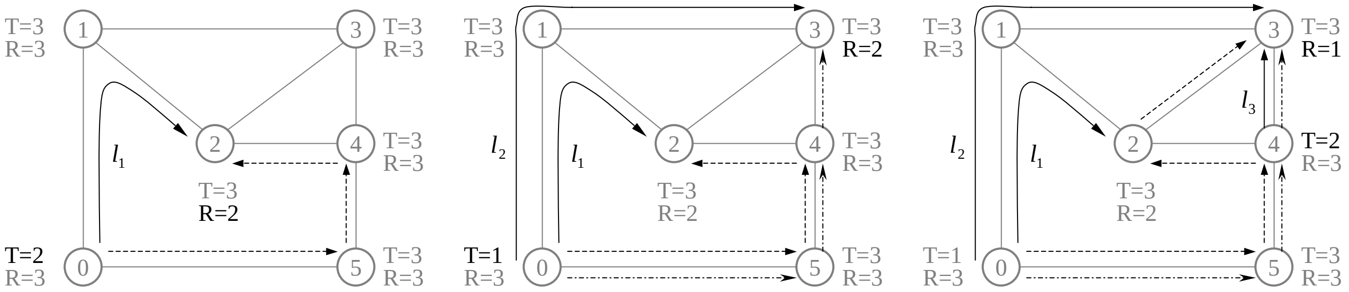
*2) Example:* When the first connection request $c_1, \langle 0, 2, \text{STS-12c}, t_1 \rangle$ arrives, one way of provisioning $c_1$ under MPAC is shown in Fig. 4(a). The working and backup paths of connection $c_1$ traverse lightpaths $l_1$ and $l_2$, respectively. The free capacity of both lightpaths $l_1$ and $l_2$ is STS-180. The backup capacity reserved on lightpath $l_1$ is zero. The backup capacity reserved on lightpath $l_2$ is STS-12c, and it is used to protect connection $c_1$’s working path. Both lightpaths $l_1$ and $l_2$ consume a grooming-add port at node $0$ and a grooming-drop port at node $2$.

Suppose that connection $c_1$ remains in the network when connection request $c_2, \langle 0, 3, \text{STS-3c}, t_2 \rangle$, arrives. One possible solution of provisioning $c_2$ under MPAC is shown in Fig. 4(b). Connection $c_2$ is routed via two link-disjoint paths—lightpath $l_3$ and the two-lightpath sequence $\langle l_2, l_4 \rangle$. The working path can traverse either of the two paths, say lightpath $l_3$. The free capacity of both lightpaths $l_3$ and $l_4$ is STS-189. The free capacity of lightpath $l_2$ is STS-177 since lightpaths $l_1$ and $l_3$ traverse common link $\langle 0, 1 \rangle$. The backup capacity on lightpath $l_2$ is STS-15 (STS-12c capacity is used to protect the working path of connection $c_1$, and STS-3c capacity is used to protect the working path of $c_2$). The backup capacity on lightpath $l_4$ is STS-3c and it is used to protect the working path of connection $c_2$.

Suppose that connections $c_1$ and $c_2$ remain in the network when connection request $c_3, \langle 4, 3, \text{STS-48c}, t_3 \rangle$, arrives. One way of provisioning $c_3$ under MPAC is shown in Fig. 4(c). The working path of $c_3$ traverses lightpath $l_5$ and the backup path traverses the two-lightpath sequence $\langle l_6, l_4 \rangle$. The free capacity of lightpaths $l_5$ and $l_6$ is STS-144. The free capacity of lightpath $l_4$ is STS-144 because the backup paths of connections $c_2$ and $c_3$ can share backup capacity ($c_2$’s working path, $l_3$, and $c_3$’s working path, $l_5$, are link-disjoint). The backup capacity of lightpath $l_4$ is STS-48 (STS-48c capacity is used to protect the working path of connection $c_3$, and STS-3c capacity—shared with the backup path of $c_3$—is used to protect the working path of connection $c_2$). The backup capacity on lightpath $l_6$ is STS-48c and it is used to protect $c_3$’s working path.

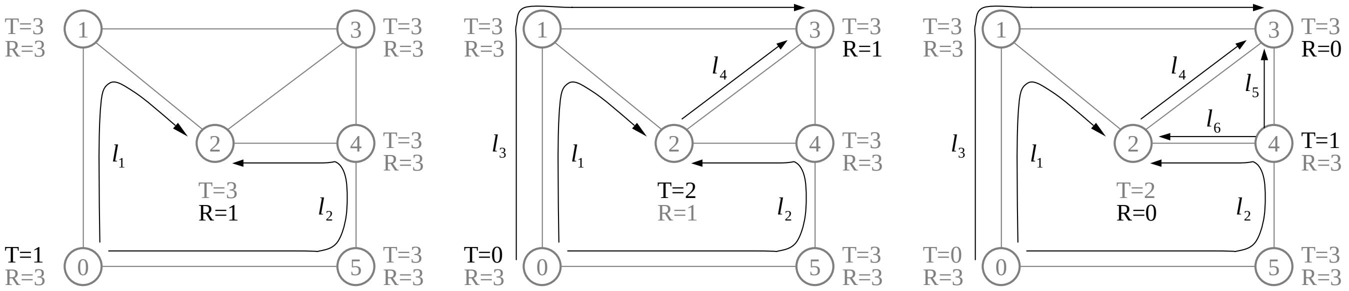
### *C. Separate Protection-at-Connection (SPAC) Level*

*1) Basic idea:* SPAC provides end-to-end protection with respect to connection. Under SPAC, a connection is routed via link-disjoint working and backup paths. A working path traverses a sequence of lightpaths. A backup path traverses a sequence of links, each of which has judiciously reserved a number of wavelengths as backup resources. (This differs from MPAC, in which a backup path traverses a sequence of lightpaths.) In addition, a grooming-add port at the source end of the link and a grooming-drop port at the destination end of the link need to be reserved for each reserved wavelength because multiple backup paths groomed onto the same wavelength on a link may go to different next hops. In this context, “separate” means that the capacity of a wavelength can be utilized by *either* working paths or backup paths, but not both. SPAC is deliberately constructed in a way to trade grooming ports for increased backup sharing, as will be elaborated in Section IV-D. Below, we illustrate SPAC in more detail using the same example as before.



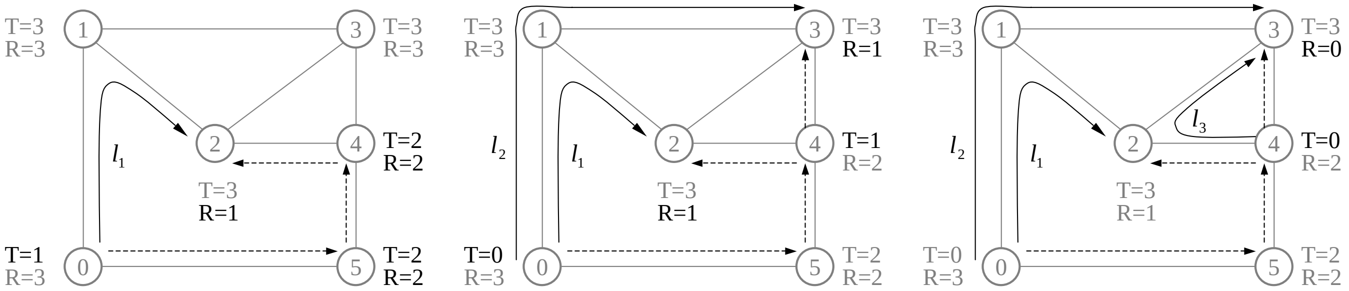
(a) after provisioning $c_1$ (b) after provisioning $c_2$ (c) after provisioning $c_3$

Fig. 3. PAL: provisioning connections $c_1(\langle 0, 2, \text{STS-12c}, t_1\rangle)$, $c_2(\langle 0, 3, \text{STS-3c}, t_2\rangle)$, and $c_3(\langle 4, 3, \text{STS-48c}, t_3\rangle)$.



(a) after provisioning $c_1$ (b) after provisioning $c_2$ (c) after provisioning $c_3$

Fig. 4. MPAC: provisioning connections $c_1(\langle 0, 2, \text{STS-12c}, t_1\rangle)$, $c_2(\langle 0, 3, \text{STS-3c}, t_2\rangle)$, and $c_3(\langle 4, 3, \text{STS-48c}, t_3\rangle)$.



(a) after provisioning $c_1$ (b) after provisioning $c_2$ (c) after provisioning $c_3$

Fig. 5. SPAC: provisioning connections $c_1(\langle 0, 2, \text{STS-12c}, t_1\rangle)$, $c_2(\langle 0, 3, \text{STS-3c}, t_2\rangle)$, and $c_3(\langle 4, 3, \text{STS-48c}, t_3\rangle)$.

*2) Example:* When the first connection request $c_1, \langle 0, 2, \text{STS-12c}, t_1\rangle$ arrives, one way of provisioning $c_1$ under SPAC is shown in Fig. 5(a). The working path of connection $c_1$ traverses lightpath $l_1$, and the backup path traverses path $\langle 0, 5, 4, 2\rangle$. The free capacity of lightpath $l_1$ is STS-180. Every link along the backup path needs to reserve one wavelength as backup capacity, while only STS-12c of the entire wavelength capacity is used to protect $c_1$'s working path. For every link along the backup path, the upstream node needs to reserve one grooming-add port and the downstream node needs to reserve one grooming-drop port since one more wavelength has been reserved.

Suppose that connection $c_1$ remains in the network when connection request $c_2, \langle 0, 3, \text{STS-3c}, t_2\rangle$, arrives. One possible solution of provisioning $c_2$ under SPAC is shown in Fig. 5(b). The working path of connection $c_2$ traverses lightpath $l_2$, and the backup path traverses path $\langle 0, 5, 4, 3\rangle$. The free capacity of lightpath $l_2$ is STS-189. One wavelength along link $\langle 4, 3\rangle$ needs to be reserved as backup capacity, STS-3c capacity of which is used to protect $c_2$'s working path. STS-15 capacity of the entire backup capacity along links $\langle 0, 5\rangle$ and $\langle 5, 4\rangle$ is used to protect the working paths of connections $c_1$ and $c_2$ (STS-12c for $c_1$ and STS-3c for $c_2$).

This step demonstrates why one grooming-drop port and two grooming-add ports need to be reserved at node $4$. If link $\langle 0, 1\rangle$ fails, connection $c_1$ needs to be rerouted along $\langle 0, 5, 4, 2\rangle$, and connection $c_2$ needs to be rerouted along $\langle 0, 5, 4, 3\rangle$. As a result, node $4$ needs to drop one wavelength, say $\lambda_1$, to the G-Fabric via one grooming-drop port. After unpacking wavelength $\lambda_1$, the G-Fabric grooms connection $c_1$ to an appropriate wavelength, say $\lambda_2$, and adds wavelength $\lambda_2$ to the W-Fabric via one grooming-add port; the G-Fabric also grooms connection $c_2$ to an appropriate wavelength, say $\lambda_3$, and inserts wavelength $\lambda_3$ to the W-Fabric via another grooming-add port. Then, the W-Fabric switches wavelength $\lambda_2$ to the outgoing port towards node 2 and wavelength $\lambda_3$ to the outgoing port towards node 3. In general, one grooming-add port and one grooming-drop port are needed for each reserved wavelengths on a link.

Suppose that connections $c_1$ and $c_2$ remain in the network when connection request $c_3, \langle 4, 3, \text{STS-48c}, t_3\rangle$, arrives. One way of provisioning $c_3$ under SPAC is shown in Fig. 5(c). The working path of connection $c_3$ traverses lightpath $l_3$ and the backup path traverses path $\langle 4,3 \rangle$. The free capacity of lightpath $l_3$ is STS-144. STS-48c of the entire backup capacity along link $\langle 4,3 \rangle$ is used to protect the working path of $c_3$; out of this STS-48c capacity, STS-3c is also used to protect the working path of $c_2$.

### *D. A Qualitative Comparison*

The above illustrative examples indicate that the three schemes perform differently in terms of routing and the amount of resources required. Below, we qualitatively compare their characteristics with respect to routing, backup sharing, and operational complexity. For convenience, we will use the term "PAC" to refer to both MPAC and SPAC hereafter whenever appropriate because they have several similar properties.

*1) Routing:* The difference in routing between PAL and PAC is that PAL provides end-to-end protection with respect to lightpath while PAC provides end-to-end protection with respect to connection. Under PAL, when a failure occurs, the end nodes of the affected $p$-lightpaths first configure their backup paths and then switch over; the affected connections are oblivious to the protection-switching process. Under PAC, when a failure occurs, the end-nodes of the affected connections (which could be significantly more than the number of affected $p$-lightpaths) first configure their backup paths and then switch over.

Please note that a connection from node $s$ to node $d$ routed under PAL may not have two link-disjoint paths between node $s$ and node $d$, while the connection still survives from single-link failures. For example, suppose that the connection is routed via two $p$-lightpaths $l_1$ and $l_2$ under PAL. The concatenation of the working paths of $p$-lightpaths $l_1$ and $l_2$ may not be link-disjoint from the concatenation of the backup paths of $p$-lightpaths $l_1$ and $l_2$. This is because the working path of $l_1$ and the backup path of $l_2$ (or the working path of $l_2$ and the backup path of $l_1$) can traverse common links.

Routing-wise, PAL performs at an aggregate level (lightpath) and PAC performs on a per-flow (connection) basis. As a result, PAL trades the bandwidth efficiency in routing each specific sub-wavelength connection request for the savings in grooming-port usage. In PAL, the backup path of a $p$-lightpath does not require any grooming port. When a fiber along the working path of a $p$-lightpath fails, all of the traffic carried by the failed working path can be rerouted to the backup path of that $p$-lightpath, and the grooming ports (at the end nodes of the $p$-lightpath) previously used by the working path can be reused by the backup path. However, in SPAC, the end nodes of a link need to reserve a grooming-add/drop port for each reserved wavelength because multiple backup paths groomed onto the same wavelength on a link may go to different next hops; in MPAC, each lightpath reserves a portion of its bandwidth as backup capacity, thus backup capacity consumes a fraction of the grooming ports.

*2) Backup sharing:* MPAC differs from PAL and SPAC in backup sharing. The backup path of a connection under MPAC is the concatenation of lightpaths. The backup path of a connection under SPAC (or the backup path of a $p$-lightpath under PAL) is the concatenation of links with reserved wavelengths. This difference in backup routing has two implications on backup sharing. First, since a lightpath may span multiple links, the backup capacity reserved on a lightpath (as in MPAC) is less likely to be shared among multiple connections than the backup capacity reserved on a link (as in PAL and SPAC).

The second implication applies to wavelength-convertible networks only. Under MPAC, the backup path of a connection traverses a sequence of lightpaths, thus a backup path has both fixed route and fixed wavelength assignment [28]. Under SPAC, the backup path of a connection (or the backup path of a $p$-lightpath under PAL) traverses a sequence of links with a number of reserved wavelengths, thus a backup path has only fixed route but not fixed wavelength assignment [29]. Basically, under SPAC and PAL, the reserved wavelengths on a link act like a "pool" for all the failure scenarios, and backup-capacity sharing among different wavelengths on a link is facilitated by the existence of wavelength converters. However, under MPAC, the backup-capacity sharing among different wavelengths on a link is not possible because backup capacity resides inside lightpaths, and multiple lightpaths cannot share their reserved backup capacity. We illustrate this difference in the following example.

Consider the changes in backup capacity on an arbitrary link $\langle u,v \rangle$ in a hypothetical network. Suppose that STS-156 capacity will be rerouted on link $\langle u,v \rangle$ when some other link $\langle x,y \rangle$ in this network fails; STS-108 capacity will be rerouted on link $\langle u,v \rangle$ when some other link $\langle i,j \rangle$ in this network fails; and no more than STS-108 capacity will be rerouted on link $\langle u,v \rangle$ when any other link fails. Clearly, STS-156 backup capacity needs to be reserved along link $\langle u,v \rangle$, assuming that any link is not in the same shared-risk-link group (SRLG)[2] as any other link. Under SPAC, link $\langle u,v \rangle$ needs to reserve one wavelength; under MPAC, a lightpath, $l_1$, from node $u$ to node $v$ needs to be set up. When a new connection request $c_1$, $\langle i, j, \text{STS-48c}, t_h \rangle$, arrives, suppose that its working path traverses link $\langle i,j \rangle$ and backup path traverses link $\langle u,v \rangle$ in this hypothetical network under both SPAC and MPAC. Since only STS-156 capacity needs to be rerouted when link $\langle i,j \rangle$ fails, no more backup capacity needs to be reserved under both SPAC and MPAC. Assume that connection $c_1$ remains in the network when another connection request $c_2$, $\langle i, j, \text{STS-48c}, t_h^{'} \rangle$, arrives. Suppose that connection $c_2$'s working path traverses link $\langle i,j \rangle$ and backup path traverses link $\langle u,v \rangle$ in this hypothetical network under both SPAC and MPAC. Then, STS-204 capacity will be rerouted on link $\langle u,v \rangle$ if link $\langle i,j \rangle$ fails. As a result, under SPAC, link $\langle u,v \rangle$ needs to reserve two wavelengths (since wavelength capacity is STS-192), which combine to provide STS-204 backup capacity. Under MPAC, another lightpath, $l_2$, from node $u$ to node $v$ needs to be set up. Lightpath $l_1$ reserves STS-156 capacity and lightpath $l_2$ reserves STS-48 capacity as backup capacity.

The difference appears when connection $c_1$ leaves and connection $c_2$ remains in the network. Only STS-156 capacity will be rerouted on link $\langle u,v \rangle$ when either link $\langle x,y \rangle$ or link $\langle i,j \rangle$ fails after connection $c_1$ leaves. Consequently, under SPAC,

[2] A SRLG is a set of links which share the same risk [30].

only one wavelength needs to be reserved on link $\langle u, v\rangle$, and another previously reserved wavelength can be released. Under MPAC, however, lightpath $l_1$ still needs to reserve STS-156 backup capacity since STS-156 capacity will be rerouted on lightpath $l_1$ when link $\langle x, y\rangle$ fails. Without reconfiguring connection $c_2$'s backup path, lightpath $l_2$ still needs to reserve STS-48 backup capacity. As a result, under MPAC, STS-204 capacity has been reserved while only STS-156 is really needed. (PAL will perform similarly to SPAC in this example.)

In short, PAL and SPAC trade the flexibility in grooming for the freedom in backup sharing. Under PAL and SPAC, working paths are groomed onto lightpaths while backup paths are groomed onto reserved wavelengths. However, MPAC has the flexibility in grooming working paths and backup paths (of different connections) onto the same lightpath.

*3) Operational complexity:* From implementation point of view, PAL is simpler than PAC as PAL demands less information in route computation. While both PAL and PAC need the routing information of all the existing lightpaths to provision a shared-protected connection request, PAL does not require any information about the existing connections. PAC, however, does require the detailed routing information of all the existing connections. Under PAL, the routing information of the working paths of two $p$-lightpaths is sufficient to determine whether the backup paths of these two $p$-lightpaths can share wavelengths along common links. Under PAC, the routing information of the working paths of two connections, which includes lightpath routing information, is needed to decide whether the backup paths of these two connections can share backup capacity along common lightpaths (in the case of MPAC) or common links (in the case of SPAC).

From control point of view, PAL has lower signaling overhead. Assume that a lightpath can carry up to $g$ connections. (In today's networks, $g$ is typically $192$ since wavelength capacity is STS-192 and the lowest bandwidth granularity is typically STS-1.) When a link fails, $W$ lightpaths can be disrupted in the worst case. In PAL, at most $W$ protection-switching processes are needed. However, in PAC, up to $W \times g$ protection-switching processes are required in the worst case. As protection-switching processes for shared protection typically require signaling, PAL demands lower control bandwidth and involves lower signaling complexity compared to PAC.

## V. Heuristic Algorithms

Since it is $\mathcal{NP}$-complete to provision a connection request under the current network state with shared protection, we develop heuristics for MPAC, SPAC, and PAL in this section.

### A. MPAC Heuristic

In response to a new connection request, MPAC computes two link-disjoint paths based on the current network state and appropriate backup-sharing measurement. Below, we elaborate the backup-sharing measurement, network-state representation, a modified k-distinct-loopless-path algorithm, and route computation.

*1) Backup-sharing measurement:* Every lightpath is associated with a conflict set[3] to identify the sharing potential between backup paths. The conflict set $\nu_l$ for lightpath $l$ can be represented as an integer set, $\{\nu_l^e | \ \forall e \in E, 0 \leq \nu_l^e \leq \text{STS-192}\}$, where $\nu_l^e$ represents the amount of traffic that will be rerouted on lightpath $l$ when link $e$ fails. The amount of backup capacity reserved on lightpath $l$ is thus $\nu_l^* = \underset{\forall e}{max}\{\nu_l^e\}$. The difference $\nu_l^* - \nu_l^e$ indicates the potential "free" capacity for backing up a new working path traversing link $e$ (and the corresponding backup path traverses lightpath $l$).

The union of the conflict sets for all the lightpaths aggregates the per-connection-based backup-sharing information, and the size of the conflict set depends only on the number of lightpaths and the number of links, not on the number of connections. In the absence of a mechanism such as the conflict set, per-connection-based information is necessary for identifying shareable backup capacity [32]. Thus, it is advantageous to use conflict set since the number of connections can be significantly more than the number of lightpaths.

*2) Grooming-node modeling and network-state representation:* Under the current network state, a connection request may be carried by existing lightpaths, by newly established lightpaths (based on available wavelengths and free grooming ports), or by both existing lightpaths and newly setup lightpaths. While the graph defined in Section III takes into account wavelength constraints, the graph does not accommodate existing lightpath information. Moreover, grooming-port constraints apply if a connection is to be carried by both existing lightpaths and newly established lightpaths. Therefore, a more powerful mechanism—which can accommodate wavelength constraints, grooming-port constraints, and existing lightpath information—is needed to represent the network state and to facilitate route computation.

We adopt the generic graph model in [9] to represent the network state as an auxiliary graph. For our grooming-node architecture in Fig. 1, W-Fabric is modeled as the $\lambda$ layer consisting of input vertex[4] $\lambda_I$ and output vertex $\lambda_O$; G-Fabric is modeled as the access layer consisting of input vertex $A_I$ and output vertex $A_O$; grooming-add port is modeled by an edge from vertex $A_O$ to vertex $\lambda_O$; and grooming-drop port is modeled by an edge from vertex $\lambda_I$ to vertex $A_I$. A unidirectional fiber is represented as an edge from vertex $\lambda_O$ at the source node to vertex $\lambda_I$ at the destination node of the lightpath. A lightpath layer consisting of input vertex $L_I$ and output vertex $L_O$ is added to model existing lightpaths sourced/sunk at a node. A lightpath is represented as an edge from vertex $L_O$ at the source node to vertex $L_I$ at the destination node. Every edge is associated with two attributes: one indicating the available capacity and the other indicating the cost of the resource which the edge represents.

As an example, the state of node 2 in Fig. 4(c) is modeled in Fig. 6. For the four auxiliary edges—$\langle\lambda_I, \lambda_O\rangle$, $\langle L_I, A_I\rangle$, $\langle A_I, A_O\rangle$, and $\langle A_O, L_O\rangle$—the capacity is infinity and the cost is zero. The available capacity of any other edge $e$ is the avail-

[3] The conflict set defined here is related to the conflict vector in [18], the aggregated square matrix in [31], and the "bucket" link metric in [22],[23].

[4] For clarity, we refer to node and link in the auxiliary graph as vertex and edge.

able capacity of the resource which edge $e$ represents, e.g., the free capacity of edge $\langle \lambda_I, A_I \rangle$ (in gray) is zero since $R = 0$ for node 2. The grayed edge from vertex $\lambda_O$ at node 4 (not shown in this figure) to vertex $\lambda_I$ at node 2 indicates there are no free wavelength on link $\langle 4, 2 \rangle$. The cost of a lightpath edge is the summational cost of the links which the lightpath traverses. The cost of lightpath edge $l_1$ is two if we assume unity link cost (multiple edges between the same vertex pair is distinguished by unique sequence numbers).

By modeling every grooming node as above, the current network state—which includes wavelength usage, grooming-port usage, and available lightpath capacity—can be represented as one auxiliary graph.

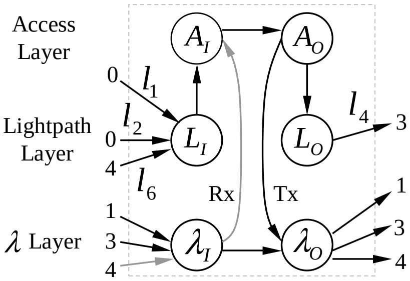


Fig. 6. Graph representation of node 2 in Fig. 4(c).

*3) Route computation:* Based on the auxiliary network-state graph and the backup-sharing measurement, MPAC computes two link-disjoint paths for a connection request. The basic idea of MPAC is to judiciously enumerate paths since joint optimization of the working and backup paths may not be possible due to the NP-completeness of the problem. A formal specification of MPAC is shown in Algorithm 1. Further elaboration follows.

MPAC enumerates $K$ candidate working paths in the auxiliary graph based on the customized algorithm for computing $K$ distinct loopless paths in Section V-A.4 below. For each candidate working path, MPAC computes a SRLG-disjoint minimal-cost path as backup based on cost function $C_{\text{MPAC}}(e)$. (The cost function $C_{\text{MPAC}}(e)$ is similar to the cost functions used in some previous work [18], [19], [22], [33], [34], but it is customized in our context to accommodate grooming constraints.) In $C_{\text{MPAC}}(e)$, $\epsilon$ is an infinitesimal constant such as $10^{-9}$. Then, MPAC selects the path pair of minimal cost. Please note that, in Step 3 of Algorithm 1, the amount of resources—especially grooming ports—used by path $l_w^k$ should be temporarily reflected to the auxiliary graph when computing path $l_b^k$ because the computation of path $l_b^k$ depends on the availability of grooming ports at every node.

In Algorithm 1, the cost of a lightpath $l$ is the sum of the cost of the links lightpath $l$ traverses; the cost of a working path $l_w$ is the sum of the cost of the lightpaths path $l_w$ traverses times the bandwidth granularity of path $l_w$; and the cost of a backup path $l_b$ is the sum of the cost of the edges $l_b$ traverses (the cost of an edge is defined by the cost function $C_{\text{MPAC}}(e)$).

Given an eligible working and backup path pair $\langle l_w, l_b \rangle$, further optimization is possible, as we have shown in [27]. However, the improvement in performance is marginal and the increase in computational complexity is remarkable in the case when connections can have different bandwidth requirements. Thus, we do not consider such optimization further. Instead, we shall investigate how to enumerate $K$ appropriate candidate working paths below.

**Algorithm 1** MPAC

```
Input: $G = (V, E, C, \lambda, P)$, $c = \langle s, d, B, t_h \rangle$, existing lightpath information, wavelength usage, grooming-port usage, and $K$.
Output: SRLG-disjoint working and backup paths, or NULL if no such paths are found.
1) construct the auxiliary graph $G_s$ to represent the current network state (including existing lightpath information, wavelength usage, and grooming-port usage) as shown in Section V-A.2,
2) compute $K$ minimal-cost paths $L_w = \{l_w^k \mid 1 \le k \le K^{'}, 0 \le K^{'} \le K\}$ in $G_s$ from the access-layer output vertex of node $s$ to the access-layer input vertex of node $d$ based on the modified algorithm for computing $k$ distinct loopless paths in Section V-A.4 subject to the constraint that every hop along a path should have at least $B$ units of free capacity (if there are less than $K$ paths between the vertices, then the algorithm will compute all the $K^{'}$, $0 \le K^{'} < K$, eligible paths); return NULL if $L_w$ is empty,
3) for each candidate working path $l_w^k$ in $L_w$, compute a minimal-cost path $l_b^k$ from the access-layer output vertex of node $s$ to the access-layer input vertex of node $d$ based on the following edge-cost function $C_{\text{MPAC}}(e)$:
    a) if edge $e$ does not represent a lightpath or a link, then $C_{\text{MPAC}}(e) = C(e)$,
    b) if edge $e$ represents a link $f$, then
       $C_{\text{MPAC}}(e) = \begin{cases} +\infty & \text{if } e \text{ is not SRLG-disjoint with } l_w^k, \text{ or link } f \text{ does not have any free wavelength} \\ B \cdot C(e) & \text{otherwise} \end{cases}$
    c) if edge $e$ represents a lightpath $l$, then
       $C_{\text{MPAC}}(e) = \begin{cases} +\infty & \text{if lightpath } l \text{ is not SRLG-disjoint with } l_w^k, \text{ or } (\nu_l^* - \nu_l^{e'}) \text{ plus the free capacity of } l \text{ is less than } B \text{ for some link } e' \text{ used by any lightpath that } l_w^k \text{ traverses} \\ \epsilon & \text{if lightpath } l \text{ is SRLG-disjoint with } l_w^k, \text{ and } (\nu_l^* - \nu_l^{e'}) \text{ is no less than } B \text{ for every link } e' \text{ used by any lightpath that } l_w^k \text{ traverses,} \\ B' \cdot C_l & \text{otherwise, where } C_l \text{ is the cost of } l, \text{ and } B^{'} = B - \min\{\nu_l^* - \nu_l^{e'}\} \text{ over all the links } e' \text{ used by any lightpath that } l_w^k \text{ traverses} \end{cases}$
4) select the path pair $\langle l_w^k, l_b^k \rangle$ of minimal cost; return NULL if no such path pair exists,
5) allocate proper resources and update network state according to the paths $l_w^k$ and $l_b^k$ (if necessary): update the free capacity of lightpaths involved in the paths $l_w^k$ and $l_b^k$; set up new lightpaths (consume new wavelengths and free grooming ports) according to the lightpath-setup strategy shown in [35],
6) for every lightpath $l$ that $l_b^k$ traverses, $\nu_l^{e'} \leftarrow \nu_l^{e'} + B$ for every link $e'$ used by any lightpath that $l_w^k$ traverses, and
7) return $l_w^k$ as working path and $l_b^k$ as backup path.
```

*4) Modified algorithm for computing $K$ distinct loopless paths:* The purpose of enumerating paths is to explore the search space of working and backup paths as much as possible so as to compute working and backup paths close to optimum.

To this end, any two of the $K$ candidate working paths should be "*distinct*" in the sense that two equal-cost candidate working paths should lead to different backup paths. Without grooming-port constraints, any algorithm for computing $K$-shortest loopless paths will be sufficient. With grooming-port constraints, some important modifications are needed, as shown below.

The $K$-shortest loopless paths from vertex $s$ to vertex $d$ are the $1^{st}$, $2^{nd}$, $\cdots$, and $k^{th}$ least cost paths, each of which traverses a vertex at most once. One of the algorithms is due to Yen in [36], and it works as follows (if less than $K$ paths exist, the algorithm will find all the $K'$ eligible paths, $0 \leq K' < K$). Initialize the candidate path set with a shortest path (e.g., computed by Dijkstra's algorithm), and set the $K$-shortest loopless path list $L$ as empty. Repeat the following steps until $L$ has $K$ paths or the candidate path set is empty. First, pick the minimal-cost path $p$ from the candidate path set and append $p$ to $L$. Then, for every vertex $u$ ($u \neq d$) along path $p$, compute a shortest path (referred to as *spur*) from vertex $u$ to vertex $d$ subject to the constraints that: (1) the spur path cannot traverse any vertex along the *root*—the segment from vertex $s$ to vertex $u$ along path $p$—except vertex $u$, i.e., a loopless path; (2) the spur path cannot branch from vertex $u$ on any link used by any path in $L$ with the same root, i.e., a new path. If a spur is found, concatenate the root and the spur as a new path $p'$ and put $p'$ into the candidate path set. Yen's algorithm has a computational complexity of $O(K \cdot N^3)$, where $N$ is the number of vertices.

If we directly apply Yen's algorithm to the auxiliary graph, some of the $K$ candidate working paths are very likely to be non-distinct. Consider the following example. Denote the access-layer input (output) vertex of node $u$ as $A_I^u$ ($A_O^u$) and the $\lambda$-layer input (output) vertex of node $u$ as $\lambda_I^u$ ($\lambda_O^u$). Suppose that one of the candidate working paths, $l_w$, traverses edge $\langle \lambda_I^u, \lambda_O^u \rangle$, i.e., $l_w$ is of the form $\langle A_O^s, \cdots, \lambda_I^u, \lambda_O^u, \cdots, A_I^d \rangle$. If we replace edge $\langle \lambda_I^u, \lambda_O^u \rangle$ by segment $\langle \lambda_I^u, A_I^u, A_O^u, \lambda_O^u \rangle$, then the new path $l_w^{'} = \langle A_O^u, \cdots, \lambda_I^u, A_I^u, A_O^u, \lambda_O^u, \cdots, A_I^d \rangle$ has the same cost as path $l_w$ (assuming that node $u$ has free grooming-add and grooming-drop ports). One can easily verify that $l_w$ and $l_w^{'}$ lead to the same backup path. To prevent such a situation, when we consider a vertex $v$ along the current path $p$, we should not compute a spur from vertex $v$ to vertex $d$ if $v$ and the next vertex along $p$ are both induced by the same network node. We then repeat this process for the next vertex along path $p$.

The above situation is due to the construction of the auxiliary graph[5], which is introduced to accommodate the grooming-port constraints. The following scenario is introduced directly by the grooming-port constraints. Denote the lightpath-layer input (output) vertex of node $u$ as $L_I^u$ ($L_O^u$). Suppose that one of the candidate working paths, $l_w$, traverses a lightpath $l_{uv}$ from node $u$ to node $v$, and suppose that there is another lightpath $l_{uv}^{'}$ of sufficient free capacity from node $u$ to node $v$ following exactly the same physical path as $l_{uv}$ does. Clearly, a new path $l_w^{'}$ formed through replacing $l_{uv}$ by $l_{uv}^{'}$ has the same cost as $l_w$, and $l_w^{'}$ is not distinct (under our definition) with $l_w$. We modify the second constraint of computing a spur in Yen's algorithm to prevent such a scenario. Let $L_u$ be the set of edges sourced at node $u$ and used by some path in $L$ with the same root. For any edge $e$ in $L_u$ representing some lightpath $l$, identify as set $L_e$ all the other edges which represent lightpaths having the same physical path as $l$ does. Let $L_u^{'} = \cup L_e$ for any lightpath edge $e$ in $L_u$. The original constraint in Yen's algorithm is that the spur path cannot use any edge in $L_u$. Now, the modified constraint is that the spur path cannot use any edge in $L_u \cup L_u^{'}$.

Our final modification to Yen's algorithm is for achieving loopless paths in the auxiliary graph. Since a network node induces six vertices in the auxiliary graph (please see Fig. 6.), if a spur cannot traverse one of the induced vertices, then the spur should not traverse any of the other five induced vertices as well in the first constraint when computing a spur. Based on the above modifications, the resultant $K$ distinct loopless paths may not be the $K$ least-cost paths, but they explore the search space of the working and backup paths better.

*5) Computational complexity:* The computational complexity of Algorithm 1 is $O(E \cdot W + K \cdot N^3)$, where $N$ is the number of network nodes, $E$ is the number of links, $W$ is the number of wavelengths, and $K$ is the number of distinct paths. Specifically, the complexity of Step 1 is $O(E \cdot W)$ since there can be as many as $E \cdot W$ lightpaths; the complexity of Step 2 is $O(K \cdot N^3)$; the complexity of Step 3 is $O(K \cdot N^2)$ (the complexities of Steps 3a, 3b, and 3c are $O(1)$, $O(N)$, and $O(N^2)$, respectively); and the complexities of Steps 4, 5, 6, and 7 are $O(K)$, $O(N)$, $O(N^2)$, and $O(1)$, respectively.

### B. SPAC Heuristic

Our heuristic for SPAC is similar to the heuristic for MPAC except for the backup-sharing measurement. Grooming node and network state are modeled the same way as shown in Section V-A.2. Below, we elaborate on the backup-sharing measurement and the difference in route computation.

*1) Backup-sharing measurement:* Every link is associated with a conflict set to identify the sharing potential between backup paths. The conflict set $\nu_e$ for link $e$ can be represented as an integer set, $\{\nu_e^{e'} | \; \forall e' \in E, 0 \leq \nu_e^{e'} \leq \text{STS-192} \times \lambda(e)\}$, where $\nu_e^{e'}$ represents the amount of traffic that will be rerouted on link $e$ when link $e'$ fails. The amount of backup capacity reserved on link $e$ is thus $\nu_e^* = \underset{\forall e'}{max}\{\nu_e^{e'}\}$. The difference $\nu_e^* - \nu_e^{e'}$ indicates the potential "free" capacity for backing up a new working path traversing link $e'$ (and the corresponding backup path traverses link $e$). The number of wavelengths which need to be reserved on link $e$ is $\lambda_e^b = \lceil \frac{\nu_e^*}{\text{STS-192}} \rceil$. The number of grooming-add ports at the upstream node (and the number of grooming-drop ports at the downstream node) of link $e$ to be reserved is also $\lambda_e^b$.

*2) Route computation:* In response to a connection request, SPAC computes link-disjoint working and backup paths in a way similar to Algorithm 1 except for the cost function in Step 3 and the conflict-set update in Step 6. The edge-cost function for computing $l_b^k$ (Step 3 in Algorithm 1) is $C_{\text{SPAC}}(e)$, defined as follows:

1) if edge $e$ does not represent a lightpath or a link, $C_{\text{SPAC}}(e) = C(e)$,
2) if edge $e$ represents a lightpath, then $C_{\text{SPAC}}(e) = +\infty$,

[5] We remark that, if we apply Yen's algorithm to the original graph $G$ instead of the auxiliary graph, then the grooming constraints may not be accommodated without appropriate mechanisms such as the graph model. Please refer to [11], [12] for more details.

3) if edge $e$ represents a link $f$, then: (i) $C_{\text{SPAC}}(e) = +\infty$ if link $f$ is not SRLG-disjoint with $l_w^k$; or (STS-192 $\times \lambda_f^b - \nu_f^{e'}$) is less than $B$ for some link $e'$ used by any lightpath which $l_w^k$ traverses and either link $f$ does not have any free wavelength, or the upstream node of link $f$ does not have any free grooming-add port, or the downstream node of link $f$ does not have any free grooming-drop port. (ii) $C_{\text{SPAC}}(e) = \epsilon$ if link $f$ is SRLG-disjoint with $l_w^k$, and $\nu_f^* - \nu_f^{e'} \geq B$ for every link $e'$ used by any lightpath that $l_w^k$ traverses. (iii) Otherwise $C_{\text{SPAC}}(e) = B' \cdot C(f)$, where $B^{'} = B - \min\{\nu_e^* - \nu_e^{e'}\}$ over all the links $e'$ used by any lightpath that $l_w^k$ traverses.

Step 6 in Algorithm 1 is modified as follows. For any link $e$ that path $l_b^k$ traverses, compute $\nu_e^{e'} \leftarrow \nu_e^{e'} + B$ for every link $e'$ used by any lightpath that $l_w^k$ utilizes, update $\nu_e^*$, and recompute $\lambda_e^b$. Reserve one more grooming-add port at the upstream node of link $e$ and one more grooming-drop port at the downstream node of link $e$ if $\lambda_e^b$ increases by one ($\lambda_e^b$ will increase by at most one).

### C. PAL Heuristic

Upon the arrival of a connection request, PAL computes a survivable route—a sequence of $p$-lightpaths—based on the current network state and appropriate backup-sharing measurement, elaborated below.

*1) Backup-sharing measurement:* We associate a conflict set with a link to identify the sharing potential between backup paths. The conflict set $\nu_e$ for link $e$ can be represented as an integer set, $\{\nu_e^{e'} \,|\, \forall e' \in E, 0 \leq \nu_e^{e'} \leq \lambda(e)\}$, where $\nu_e^{e'}$ specifies the number of working lightpaths that traverse link $e'$ and are protected by link $e$, i.e., their corresponding backup paths traverse link $e$. The number of wavelengths reserved for backup paths on link $e$ is thus $\nu_e^* = \underset{\forall e'}{max}\{\nu_e^{e'}\}$. By definition, there is no need to increase the number of reserved wavelengths on link $e$ for protecting up to $\nu_e^* - \nu_e^{e'}$ more working lightpaths traversing link $e'$.

*2) Network-state representation:* The current network state is collectively represented by the set of existing $p$-lightpaths, the conflict set $\nu = \{\nu_e | e \in E\}$, and a digraph $G = (V, E, C, \lambda^{'}, P_T^{'}, P_R^{'})$ (where $\lambda^{'} : E \rightarrow Z^+$ specifies the number of available wavelengths on each link, and $P_T^{'}(P_R^{'}) : V \rightarrow Z^+$ specifies the number of available grooming-add (grooming-drop) ports at a node). Please note that the detailed routing information about a $p$-lightpath is not needed for route computation since the conflict set has already aggregated that information. The only information about a $p$-lightpath needed for route computation is the source node, the destination node, and the available bandwidth of a $p$-lightpath. We will use NS to denote network state below.

*3) Route computation:* The basic idea of PAL is to extend a standard shortest-path algorithm such that every hop along the resultant shortest path corresponds to a $p$-lightpath, which can be either an existing $p$-lightpath of sufficient free capacity or a new $p$-lightpath consisting of fresh wavelength links and free grooming ports. The challenge here is to accommodate the backup sharing between the existing $p$-lightpaths and the new $p$-lightpaths and the backup sharing among the new $p$-lightpaths while computing the survivable route. PAL is related to the link-protection algorithm in [37] and works as follows.

In order to keep track of backup sharing and to maintain virtual adjacency[6] relationship, we associate to every node $u \in V$ a network state NS($u$). NS($u$) represents the updated network state after the survivable route from node $s$ to node $u$ is set up based on the current network state. We also associate to every node $u \in V$ a cost $C_u$, which represents the cost of the survivable route from node $s$ to node $u$.

For any two nodes $u, v \in V$, to decide whether node $u$ is node $v$'s previous node along the survivable route from the source node $s$ to the destination node $d$, we consider the following two possibilities. First, if an existing $p$-lightpath $l_{uv}^e$ from node $u$ to node $v$ of sufficient free capacity exists and $C_u$ plus the cost of $l_{uv}^e$'s working lightpath is less than $C_v$, then node $u$ is chosen as node $v$'s previous node, $p$-lightpath $l_{uv}^e$ is chosen as node $u$'s previous hop, $C_v$ is updated as the summation of $C_u$ and the cost of $l_{uv}^e$'s working lightpath, and NS($v$) is the same as NS($u$) except that the free capacity of $p$-lightpath $l$ in NS($v$) is reduced by $B$. Second, if an existing eligible $p$-lightpath from node $u$ to node $v$ does not exist, we check if a new one can be set up based on the network state in NS($u$). If a new $p$-lightpath $l_{uv}^n$ can be set up and $C_u$ plus the cost of $l_{uv}^n$ is less than $C_v$, then node $u$ is chosen as node $v$'s previous node, $p$-lightpath $l_{uv}^n$ is chosen as node $u$'s previous hop, $C_v$ is updated as the summation of $C_u$ and the cost of $l_{uv}^n$, and NS($v$) is the network state after setting up $l_{uv}^n$ in NS($u$).

The above procedure is executed similarly to a shortest-path algorithm until $C_d$ reaches its minimum. Once $C_d$ reaches its minimum, the survivable route from node $s$ to node $d$ can be retrieved by backtracking along the previous hop starting from node $d$.

The key aspect of our PAL heuristic is that the tentative changes (in network state) introduced by the survivable route from the source node $s$ to any other node $u$ is captured by the network state at node $u$. As a result, to decide whether a new $p$-lightpath $l_{uv}^n$ from node $u$ to any other node $v$ can be set up, the backup sharing between existing $p$-lightpaths and the new $p$-lightpath $l_{uv}^n$ is accommodated since the conflict set $\nu$ of NS($u$) measures the backup-sharing potential. Furthermore, the backup sharing among new $p$-lightpaths is also accommodated because the survivable route from node $s$ to node $u$ can have new $p$-lightpaths as intermediate hops. In that case, the sharing potential introduced by these new $p$-lightpaths (used by the survivable route from node $s$ to node $u$) is reflected in the conflict set $\nu$ of NS($u$). Since the computation of $p$-lightpath $l_{uv}^n$ is based on NS($u$), the above forms of backup sharing are correctly captured. A formal specification of PAL is shown in Algorithm 2.

*4) Computational complexity:* The complexity of Algorithm 2 is $O(E \cdot W + K \cdot N^5)$, where $E$ is the number of links, $W$ is the number of wavelengths, $K$ is the number of alternate paths, and $N$ is the number of nodes. Specifically, the complexity of Step 1 is $O(E \cdot W)$ since there can be as many as $E \cdot W$ lightpaths; the complexity of Step 2 is $O(K \cdot N^5)$ since Algorithm 3 is executed $O(N^2)$ times and the complex-

[6] Node $v$ is referred to be virtually adjacent to node $u$ if there exists a $p$-lightpath from node $u$ to node $v$.

**Algorithm 2** PAL

```
Input: $G' = (V, E, C, \lambda', P_T', P_R')$, $\nu = \{\nu_e | e \in E\}$,
$c = \langle s, d, B, t_h \rangle$, existing $p$-lightpath information, wavelength usage,
grooming-port usage, and $K$.
Output: A concatenation of $p$-lightpaths from node $s$ to node $d$ of free
capacity no less than $B$, or NULL.
1) Initialization
   $V' \leftarrow V$; $C_u \leftarrow +\infty, \mathrm{NS}(u) \leftarrow$ NULL, $\forall u \in V' - \{s\}$; $C_s \leftarrow$
   $0, \mathrm{NS}(s) \leftarrow$ Current network state
2) Iteration
   while ($d \in V'$) do
       $u \leftarrow \arg\min_{u \in V'} \{C_u\}$, $V' \leftarrow V' - \{u\}$
       if ($C_u = +\infty$) return NULL
       for each node $v \in V'$ do
           PAL_RELAX_NEXT_HOP($u$, $v$, $B$) (Algorithm 3)
3) Post-process: retrieve the path and update network state
   if ($C_d = +\infty$) then return NULL; otherwise
       retrieve the survivable route by following the previous hop
       starting from node $d$
       update the current network state according to NS($d$)
       return the survivable route
```

**Algorithm 3** PAL_RELAX_NEXT_HOP($u$, $v$, $B$)

```
1) let both $p$-lightpaths $l^e_{uv}$ and $l^n_{uv}$ be NULL
2) let $L_{uv}$ be the set of existing $p$-lightpaths from node $u$ to node $v$ of
   free capacity no less than $B$. Let $l^e_{uv}$ be the $p$-lightpath of minimal
   working-lightpath cost among all the $p$-lightpaths in $L_{uv}$
3) if $l^e_{uv}$ is not NULL and $C_u$ plus the cost of $l^e_{uv}$'s working lightpath
   is less than $C_v$, then
       $C_v \leftarrow C_u$ plus the cost of $l^e_{uv}$'s working lightpath
       NS($v$) $\leftarrow$ NS($u$), and reduce the available bandwidth of $l^e_{uv}$ by
       $B$ in NS($v$)
       set $l^e_{uv}$ as node $v$'s previous hop
4) if $l^e_{uv}$ is NULL and node $u$ has free grooming-add ports and node
   $v$ has free grooming-drop ports then execute the following steps
   based on NS(u)
       $L^n_{uv} \leftarrow \emptyset$
       compute $K$ minimal-cost paths $\{l^k_{uv} \mid 1 \le k \le K', 0 \le K' \le$
       $K\}$ from node $u$ to node $v$ using fresh wavelength links based
       on the original Yen's algorithm
       for each path $l^k_{uv} \in \{l^k_{uv} \mid 1 \le k \le K', 0 \le K' \le K\}$ do
           compute a minimal-cost path $\hat{l}^k_{uv}$ based the following link-
           cost function $C_{\mathrm{PAL}}(e)$: (i) $C_{\mathrm{PAL}}(e) = +\infty$ if link $e$ is not
           SRLG-disjoint with $l^k_{uv}$, or link $e$ does not have any free
           wavelength and $\nu^*_e$ is equal to $\nu^{e'}_e$ for some link $e'$ that $l^k_{uv}$
           traverses. (ii) $C_{\mathrm{PAL}}(e) = \epsilon$ if link $e$ is SRLG-disjoint with
           $l^k_{uv}$ and $\nu^{e'}_e < \nu^*_e$ for every link $e'$ that $l^k_{uv}$ traverses. (iii)
           Otherwise, $C_{\mathrm{PAL}}(e) = C(e)$.
           $L^n_{uv} \leftarrow L^n_{uv} \cup \{\langle l^k_{uv}, \hat{l}^k_{uv} \rangle\}$ if $\hat{l}^k_{uv}$ exists
       let $l^n_{uv}$ be the $p$-lightpath of minimal cost in $L^n_{uv}$
5) if $l^n_{uv}$ is not NULL and $C_u$ plus the cost of $l^n_{uv}$ is less than $C_v$,
   then
       $C_v \leftarrow C_u$ plus the cost of $l^n_{uv}$
       NS($v$) $\leftarrow$ NS($u$), and update NS($v$) as follows: reduce the num-
       ber of free grooming-add ports at node $u$ by one; reduce the
       number of free grooming-drop ports at node $v$ by one; consume
       fresh wavelength links along $l^n_{uv}$'s working lightpath; and up-
       date $\nu$: $\nu^{e'}_e \leftarrow \nu^{e'}_e + 1$ for any link $e$ along $l^n_{uv}$'s backup path
       and any link $e'$ along $l^n_{uv}$'s working path
       set $l^n_{uv}$ as node $v$'s previous hop
```

ity of Algorithm 3 is $O(K \cdot N^3)$ (in particular, the complexities of Steps 1-5 are $O(1)$, $O(E \cdot W)$, $O(E \cdot W)$, $O(K \cdot N^3)$, and $O(E \cdot W)$, respectively); and the complexity of Step 3 is $O(E \cdot W)$.

## VI. Illustrative Numerical Results

We simulate a dynamic network environment with the assumptions that the connection-arrival process is Poisson and the connection-holding time follows a negative exponential distribution. For the illustrative results shown here, the capacity of each wavelength is STS-192; the number of the connection requests follows the distribution STS-1 : STS-3c : STS-12c : STS-48c : STS-192c = 300 : 20 : 6 : 4 : 1 (which is close to the bandwidth distribution in a practical backbone network); connection requests are uniformly distributed among all node pairs; average connection-holding time is normalized to unity; the cost of any link is unity; load (in Erlang) is defined as connection-arrival rate times average holding time times a connection's average bandwidth normalized in the unit of STS-192; and our example network topology with 16 wavelengths per fiber is shown in Fig. 7. 100,000 connections were simulated in each experiment. The value of $\epsilon$ in the cost function is set to $10^{-6}$; $\epsilon$ can trade off backup sharing and backup-path length, as shown in [34]. (More results from different topologies leading to the same conclusion are reported in [38].)

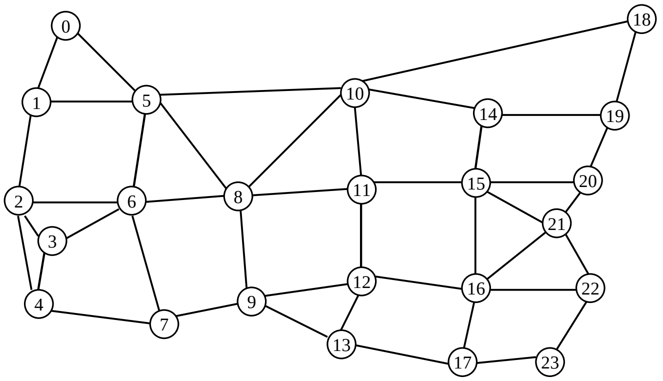


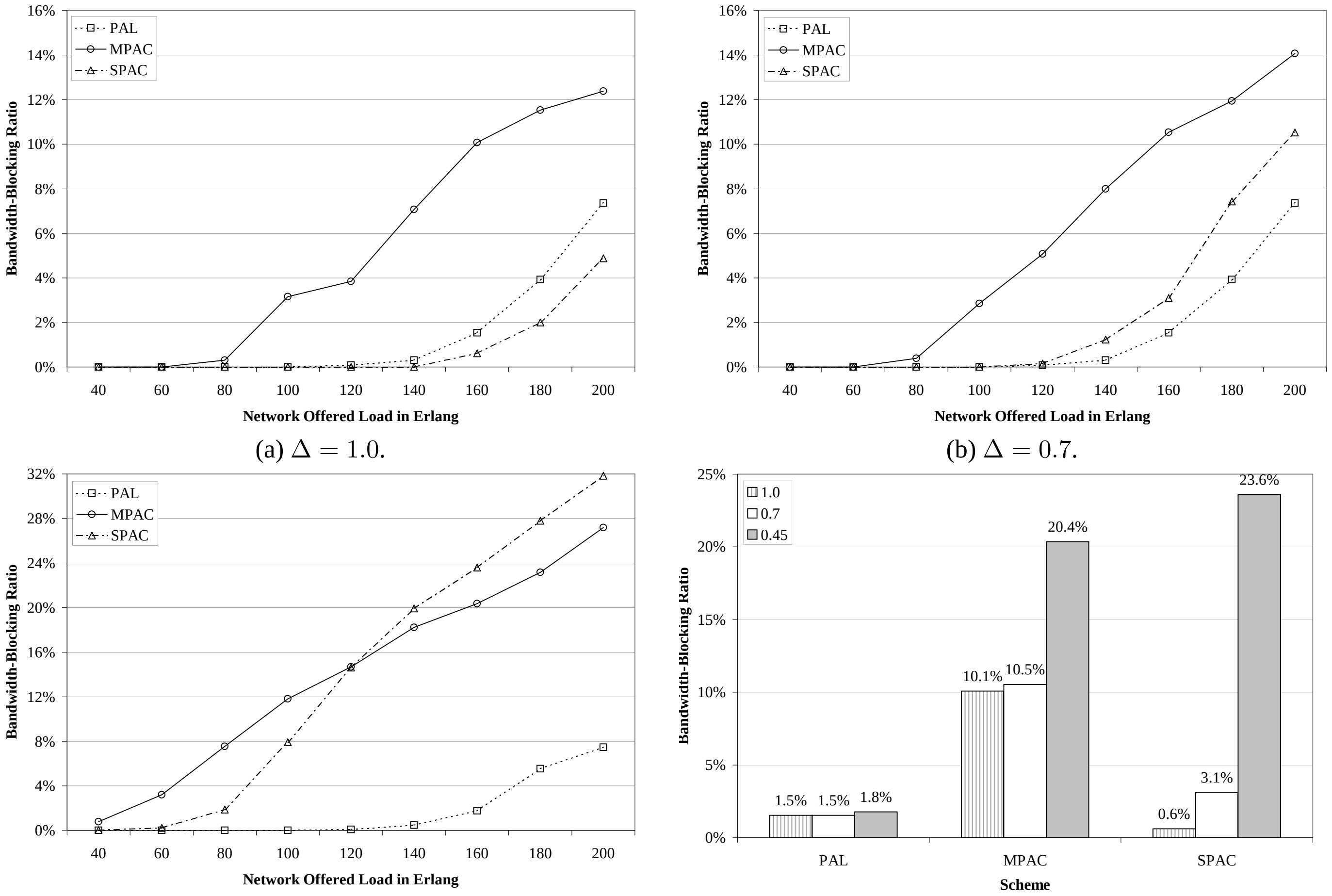
Fig. 7. A 24-node example network topology.

The number of grooming ports at a node is set as the number of wavelengths times its nodal degree times a scalar $\Delta$ ($0 \le \Delta \le 1$, $\Delta = 1$ implies that any incoming wavelength to the W-Fabric can be dropped to the G-Fabric). The number of alternate paths $K$ for the three schemes is two. Later in Section VI-C, we shall examine the effect of different values of these parameters.

We now quantitatively compare PAL to MPAC to SPAC using the following metrics: bandwidth-blocking ratio (BBR) and resource-efficiency ratio (RER).

### A. Bandwidth-Blocking Ratio

BBR is defined as the amount of bandwidth blocked over the amount of bandwidth offered. Please note that pure blocking probability, defined as the percentage of the *number* of connections blocked, cannot reflect the effectiveness of the algorithm as connections have different bandwidth requirements. Figure 8



(a) $\Delta = 1.0$.

(b) $\Delta = 0.7$.

(c) $\Delta = 0.45$.

(d) Erlang = 160.

Fig. 8. Bandwidth-blocking ratio versus network offered load.

plots the BBR of the three schemes with $\Delta = 1.0, 0.7$, and $0.45$. We make the following observations.

We find that PAL always has lower BBR than MPAC, and SPAC has lower BBR than MPAC when the number of grooming ports is large (e.g., $\Delta = 1.0$ and $0.7$) or the number of grooming ports is small and the network offered load is moderate (e.g., $\Delta = 0.45$ and the network offered load is less than $120$ Erlangs). This leads to our first observation: *It is beneficial to groom working paths and backup paths separately, as is the case in PAL and SPAC.*

Our second observation is that *SPAC has the lowest BBR when the number of grooming ports is sufficient (e.g., $\Delta = 1.0$)*, as shown in Fig. 8(a). This is because SPAC has the maximum freedom in backup sharing when the number of grooming ports is sufficient (please see Section IV-D.2).

Our third observation is that *PAL achieves the lowest BBR when the number of grooming ports is moderate or small (e.g., $\Delta = 0.7$ and $0.45$)*, as shown in Figs. 8 (b) and (c). The main reason for this is that backup paths do not consume grooming ports under PAL (Section IV-D.1).

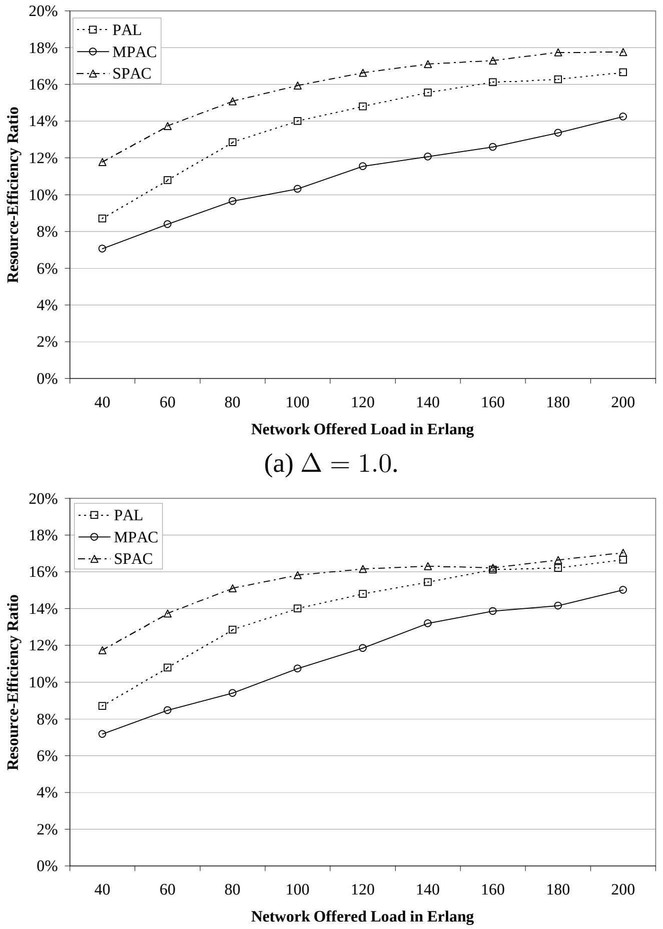
Figure 8(d) shows the BBR of the three schemes with different values of $\Delta$ under the same network offered load, $160$ Erlangs. Clearly, the decrease in grooming capacity impacts PAL the least and SPAC the most. Again, this is because PAL trades bandwidth efficiency in routing for grooming-port savings, and SPAC trades grooming ports for the flexibility in backup sharing (Section IV-D). More reasons for the above observations will be further elaborated below.

### *B. Resource-Efficiency Ratio*

*1) Definition:* To better evaluate the performance of our route-computation heuristics, we introduce a new metric, called *resource-efficiency ratio* (RER) $\mathcal{E}$, which is defined as the carried load (weighted by time and normalized to STS-192 capacity) divided by the amount of allocated resources in terms of wavelength channels and grooming ports (weighted by time). This metric is defined as follows:

$$\mathcal{E}(W_\lambda, W_g) = \frac{\sum_i \rho_i \times t_i}{W_\lambda \times \sum_i \beta_i \times t_i + W_g \times \sum_i \gamma_i \times t_i}$$

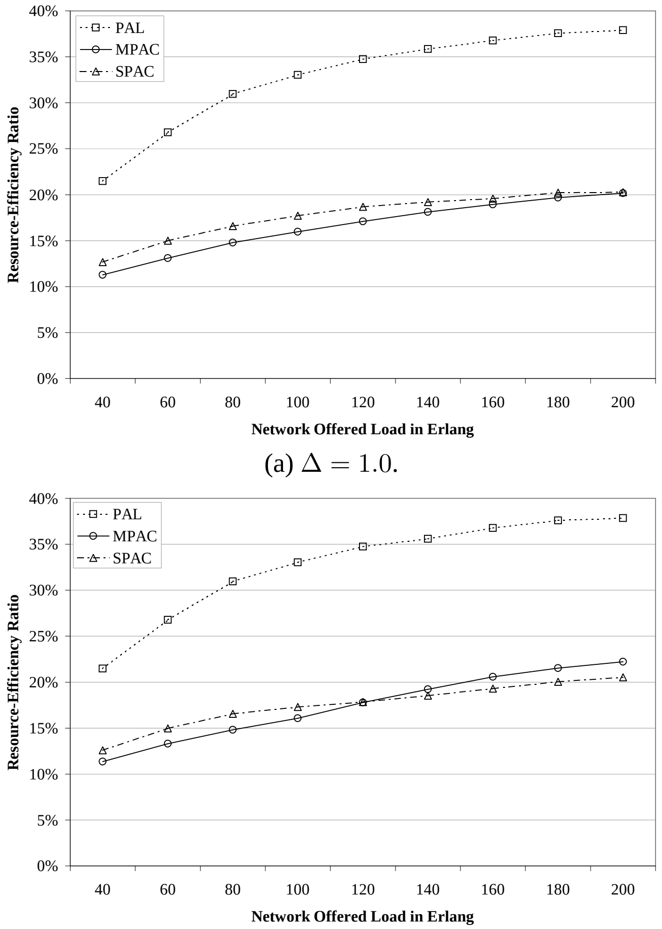
where $t_i$ is the time period between the $i^{th}$ event (connection arrival or departure) and $(i+1)^{th}$ event; $\rho_i$ is the network carried load during the time period $t_i$; $\beta_i$ is the number of wavelength links used during $t_i$; $\gamma_i$ is the number of grooming ports used during $t_i$; $W_\lambda$ and $W_g$ are the relative weight of a wavelength link versus a grooming port. (Please note that $\rho_i$, $\beta_i$, and $\gamma_i$ do not change during time period $t_i$ as there is no other event during the period.) Basically, $\mathcal{E}$ measures how efficiently resources have been used.



(a) $\Delta = 1.0$.

(b) $\Delta = 0.45$.

Fig. 9. Resource-efficiency ratio, $\mathcal{E}(1, 0)$, versus network offered load.



(a) $\Delta = 1.0$.

(b) $\Delta = 0.45$.

Fig. 10. Resource-efficiency ratio, $\mathcal{E}(0, 1)$, versus network offered load.

*2) Wavelength efficiency:* If $W_\lambda = 1$ and $W_g = 0$, RER $\mathcal{E}(1, 0)$ measures how efficiently wavelength channels have been utilized. Figure 9 plots the RER $\mathcal{E}(1, 0)$ for $\Delta = 1.0$ and $0.45$ (the plot for $\Delta = 0.7$ is similar to the one for $\Delta = 1.0$). MPAC has the lowest wavelength efficiency since PAL and SPAC have more flexibility in backup sharing. Furthermore, SPAC has the highest wavelength efficiency because PAL works at lightpath level and lightpaths are not perfectly filled.

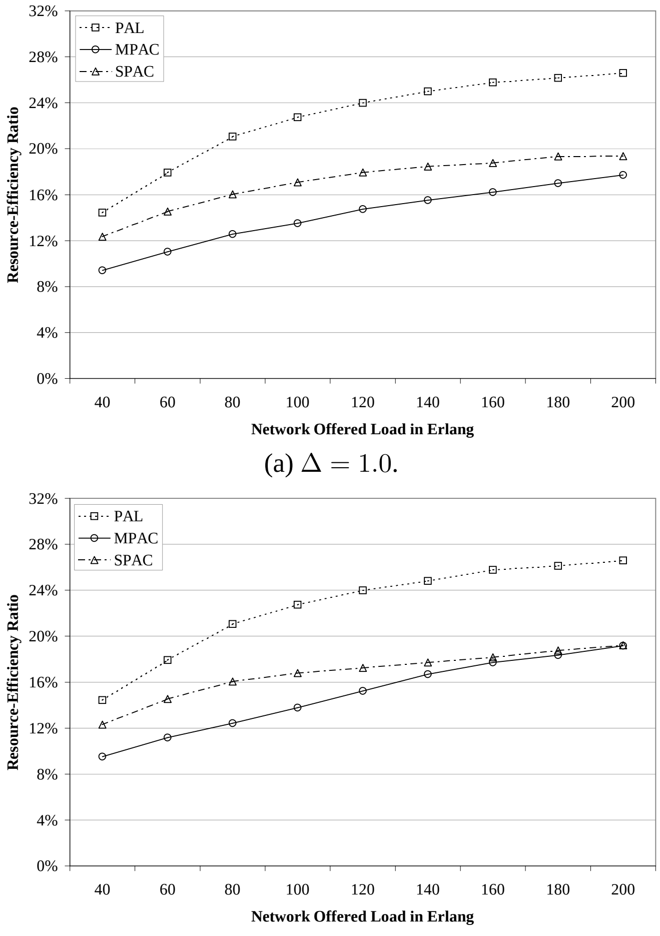
*3) Grooming-port efficiency:* If $W_\lambda = 0$ and $W_g = 1$, RER $\mathcal{E}(0, 1)$ measures how efficiently grooming ports have been utilized. Figure 10 plots the RER $\mathcal{E}(0, 1)$ for $\Delta = 1.0$ and $0.45$ (the plot for $\Delta = 0.7$ is similar to the one for $\Delta = 1.0$).

Our first observation is that PAL has the highest grooming-port efficiency under different values of $\Delta$. The reason is that backup paths consume grooming ports under both SPAC and MPAC, but not under PAL. This confirms our result in Fig. 8(d) and the analysis in Section IV-D that PAL trades bandwidth efficiency for the savings in grooming ports. As a result, under PAL, connections are more likely blocked due to insufficient wavelengths.

Our second observation is that SPAC has higher grooming-port efficiency than MPAC when the network offered load is moderate or low and SPAC has lower grooming-port efficiency than MPAC when the network offered load is high. When the network offered load is moderate or low, the lightpaths under MPAC are only moderately loaded. That is, a substantial number of grooming ports is used by these lightpaths to carry relatively moderate network load. Therefore, MPAC has lower grooming-port efficiency when the network offered load is moderate or low. When the network offered load is high, the lightpaths under MPAC are heavily loaded. Under SPAC, a significant number of grooming ports is used by backup paths since every single reserved wavelength needs a grooming-add port and a grooming-drop port. Thus, SPAC has lower grooming-port efficiency when the network offered load is high.

*4) Tradeoff between wavelengths and grooming ports:* The three schemes trade off the utilization between wavelengths and grooming ports. Below, we show that either PAL or SPAC can have the highest RER, depending on the relative weight of a wavelength channel ($W_\lambda$) and a grooming port ($W_g$). However, the intuitive scheme, MPAC, will not have the highest RER with any possible $W_\lambda$ and $W_g$ combination because both RER $\mathcal{E}(1, 0)$ and RER $\mathcal{E}(0, 1)$ for MPAC are lower than those for PAL, as shown in Figs. 9 and 10. The fundamental reason for this is that MPAC has disadvantages in backup sharing compared to either PAL or SPAC (please refer to Section IV-D.2).

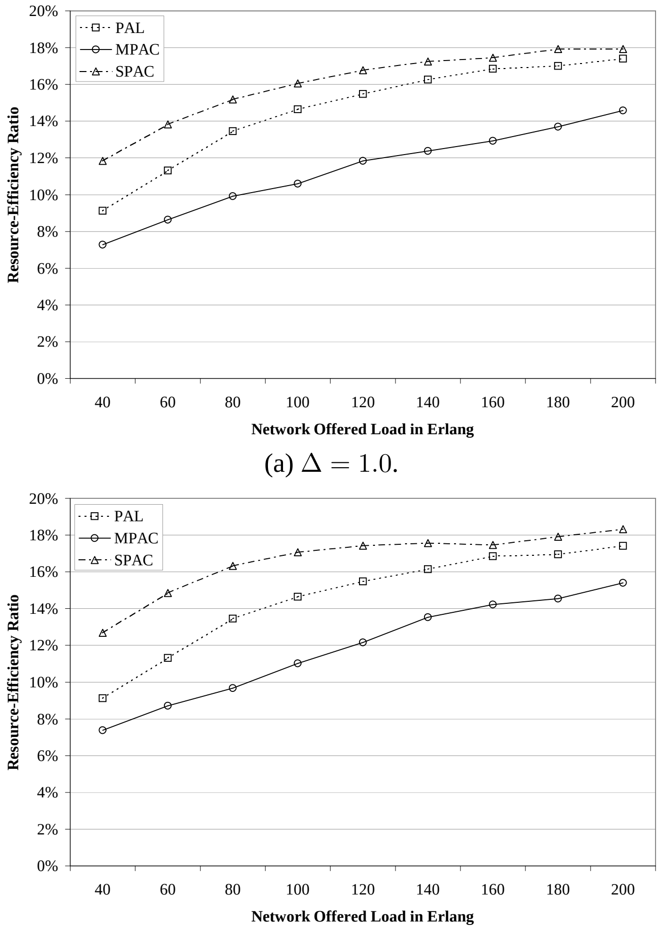
Figure 11 plots the RER $\mathcal{E}(\frac{1}{3}, \frac{2}{3})$ for $\Delta = 1.0$ and $0.45$. We observe that PAL has the highest RER $\mathcal{E}(\frac{1}{3}, \frac{2}{3})$. In general, based on the results in Figs. 9-11, PAL has the highest RER



(a) $\Delta = 1.0$.

(b) $\Delta = 0.45$.

Fig. 11. Resource-efficiency ratio, $\mathcal{E}(\frac{1}{3}, \frac{2}{3})$, versus network offered load.



(a) $\Delta = 1.0$.

(b) $\Delta = 0.45$.

Fig. 12. Resource-efficiency ratio, $\mathcal{E}(\frac{12}{13}, \frac{1}{13})$, versus network offered load.

$\mathcal{E}(W_\lambda, W_g)$ when a grooming port weights more than a wavelength link, e.g., $W_\lambda : W_g = 1 : 2$.

Figure 12 plots the RER $\mathcal{E}(\frac{12}{13}, \frac{1}{13})$ for $\Delta = 1.0$ and $0.45$. We observe that SPAC has the highest RER $\mathcal{E}(\frac{12}{13}, \frac{1}{13})$. In general, based on the results in Figs. 9, 10, and 12 SPAC has the highest RER $\mathcal{E}(W_\lambda, W_g)$ when a wavelength link weights significantly more than a grooming port, e.g., $W_\lambda : W_g = 12 : 1$.

### C. Effects of Different Parameters

Figure 13 plots the impact of $K$, the number of distinct alternate paths, on BBR for the three schemes. For MPAC and SPAC, when $K$ increase from one to two, we observe a modest reduction in BBR; when $K$ further increases from two to three, the decrease in BBR is marginal or none. This is expected since larger $K$ implies larger search space for MPAC and SPAC. However, BBR for PAL increases when $K$ increases. This is because we can only apply the $K$-shortest-path algorithm to compute new $p$-lightpaths, but not to compute the final survivable route. Since the cost of an existing $p$-lightpath is defined as the cost of its working path to encourage the use of existing $p$-lightpaths, increasing $K$ does more harm than good because larger $K$ basically prefers the use of new $p$-lightpaths in PAL.

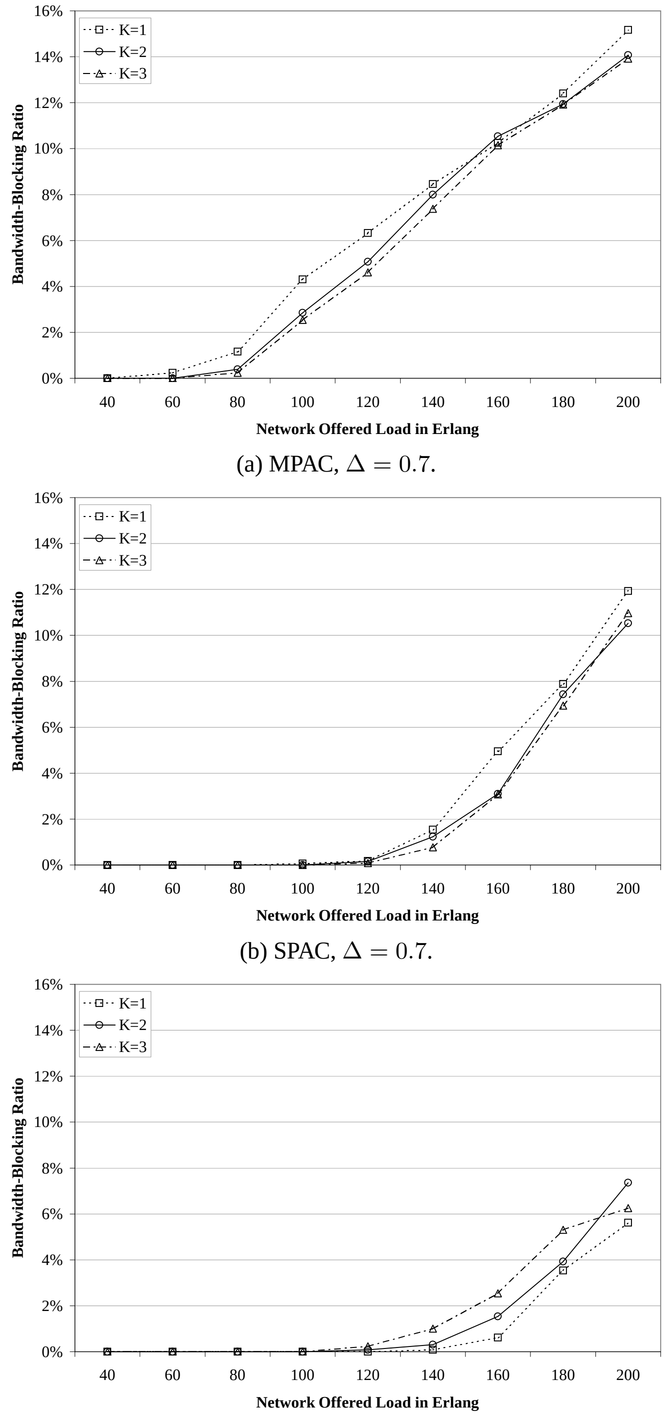
## VII. Traffic Grooming with Dedicated Protection

Some mission-critical connections may require dedicated protection instead of shared protection to achieve fast recovery in case of failures. This section discusses the problem of dynamic sub-wavelength-granularity connection provisioning with *dedicated* protection against single-fiber failures under the same grooming-node architecture as shown in Fig. 1. We highlight the differences from connection provisioning with *shared* protection discussed earlier. For a more detailed treatment, please see [35].

### A. Proposed Approaches

We propose two schemes—protection-at-lightpath (PAL) level and protection-at-connection (PAC) level[7]. Under PAL, a connection is routed through a sequence of $p$-lightpaths. A *$p$-lightpath* in this context is defined as a pair of link-disjoint *lightpaths* between two nodes. Under PAC, a connection is routed via link-disjoint working and backup paths, each of which traverses a sequence of lightpaths. Dedicated PAC works similar to MPAC except that there is no backup sharing in dedicated PAC.

[7] Please note that the scheme PAL (or PAC) in the context of this section works differently from the scheme PAL (or PAC) in the context of connection provisioning with shared protection discussed earlier.



(a) MPAC, $\Delta = 0.7$.

(b) SPAC, $\Delta = 0.7$.

(c) PAL, $\Delta = 0.7$.

Fig. 13. BBR versus network offered load with $K = 1, 2$, and 3.

### B. Illustrative Numerical Results

Figure 14 plots the BBR of PAL and PAC with $\Delta = 1.0, 0.7$, and $0.45$ under the same simulation configuration as in Section VI. We make the following observations.

*1) PAL vs. PAC:* When the number of grooming ports is high, e.g., $\Delta = 1.0$ or $0.7$, PAC has much lower BBR than PAL under moderate or high network offered load. However, when the number of grooming ports is small, e.g., $\Delta = 0.45$, PAL has much lower BBR than PAC under moderate or high network offered load. This is because PAC trades grooming ports for bandwidth efficiency.

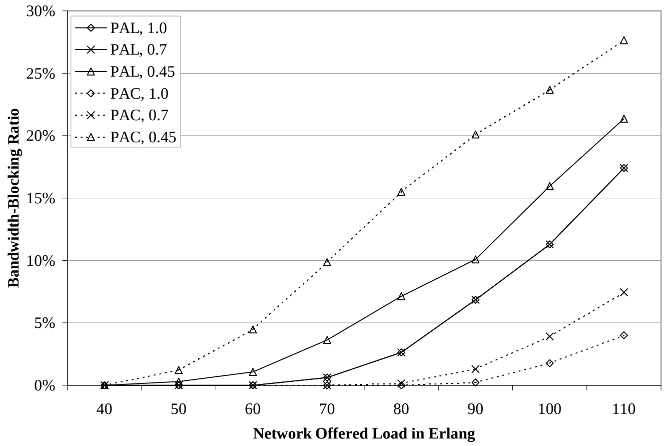


Fig. 14. BBR for $\Delta = 1.0, 0.7$, and $0.45$ (the two curves for "PAL, 1.0" and "PAL, 0.7" overlap each other).

*2) Impact of grooming capacity on PAL:* If we examine the three PAL curves in Fig. 14, we observe that PAL is not very sensitive to the changes in the number of grooming ports. For example, the BBRs for PAL under $\Delta = 1.0$ and $\Delta = 0.7$ are the same. When $\Delta$ further decreases to $0.45$, the BBR for PAL increases moderately. The reason for this is that PAL trades bandwidth efficiency for grooming ports, therefore PAL exploits wavelengths more quickly than grooming ports..

*3) Impact of grooming capacity on PAC:* If we examine the three PAC curves in Fig. 14, we observe that PAC is very sensitive to the changes in the number of grooming ports. For example, the BBR for PAC increases moderately when $\Delta$ decreases from $1.0$ to $0.7$; and the BBR for PAC increases a lot when $\Delta$ further decreases to $0.45$. Again, this is because PAC utilizes grooming ports more aggressively than PAL does.

## VIII. Conclusion and Future Work

We investigated the survivable traffic-grooming problem for optical WDM mesh networks in a dynamic context. Based on a generic grooming-node architecture, we explored three approaches—protection-at-lightpath (*PAL*) level, mixed protection-at-connection (*MPAC*) level, and separate protection-at-connection (*SPAC*) level—for grooming a connection request with *shared* protection against single-fiber failures. Our findings are as follows. Under today's typical connection-bandwidth distribution, 1) it is beneficial to groom working paths and backup paths separately, as in PAL and SPAC; 2) separately protecting each individual connection—i.e., SPAC—yields the best performance when the number of grooming ports is sufficient; 3) protecting each specific lightpath—i.e., PAL—achieves the best performance when the number of grooming ports is moderate or small. For traffic grooming with *dedicated* protection, findings 2) and 3) hold, while finding 1) does not apply because we typically do not need to distinguish between working and backup paths in dedicated protection.

Another dimension of the problem is *residual connection-holding time*. One can define residual connection-holding time for an established connection as the period between now and the time the connection will be released. We expect residual connection-holding time to have a significant impact on both backup sharing and grooming. As an example, suppose that two established connections $c_1$ and $c_2$ have residual connection-holding times of 1 unit and 1000 units, respectively; the current connection request $c$ can share the same amount of backup resources with either $c_1$ or $c_2$; and $c$ has a connection-holding time of 1000 units. Clearly, it is beneficial for connection $c$ to share backup resources with $c_2$. For grooming, similar situations, e.g., whether the current connection should be groomed onto lightpath $l_1$ or lightpath $l_2$ based on the residual connection-holding time of the connections traversing lightpaths $l_1$ and $l_2$, can arise. Further study is needed to quantify the benefits of accommodating residual connection-holding time into route computation.

## REFERENCES

[1] K. Zhu and B. Mukherjee, "A review of traffic grooming in WDM optical networks: architectures and challenges," *SPIE Optical Networks Mag.*, vol. 4, pp. 55–64, Mar./Apr. 2002.

[2] E. Modiano and P. J. Lin, "Traffic grooming in WDM networks," *IEEE Communication Mag.*, vol. 39, pp. 124–129, July 2001.

[3] J. M. Simmons, E. L. Goldstein, and A. A. M. Saleh, "Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic," *IEEE/OSA J. Lightwave Tech.*, vol. 17, pp. 48–57, Jan. 1999.

[4] O. Gerstel, R. Ramaswami, and G. H. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Trans. Networking*, vol. 8, pp. 618–630, Oct. 2000.

[5] X. Zhang and C. Qiao, "An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings," *IEEE/ACM Trans. Networking*, vol. 8, pp. 608–617, Oct. 2000.

[6] J. Wang, W. Cho, V. R. Vemuri, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM ring networks: ILP formulations and single-hop and multiphop connections," *IEEE/OSA J. Lightwave Tech.*, vol. 20, pp. 122–133, Nov. 2001.

[7] J. Wang and B. Mukherjee, "Interconnected WDM ring networks: strategies for interconnection and traffic grooming," *SPIE Optical Networks Mag.*, vol. 3, pp. 10–20, Sep./Oct. 2002.

[8] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE J. Selected Areas in Communications*, vol. 20, pp. 122–133, Jan. 2002.

[9] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel, generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Trans. Networking*, vol. 11, pp. 285–299, Apr. 2003.

[10] S. Thiagarajan and A. K. Somani, "Capacity fairness of WDM networks with grooming capabilities," *SPIE Optical Networks Mag.*, vol. 2, pp. 24–32, May/June 2001.

[11] K. Zhu and B. Mukherjee, "On-line approaches for provisioning connections of different bandwidth granularities in WDM mesh networks," in *Proc. OFC*, p. ThW5, Mar. 2002.

[12] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "Dynamic traffic grooming in WDM mesh networks using a novel graph model," in *Proc. IEEE Globecom*, pp. 2681–2685, Nov. 2002.

[13] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *IEEE/OSA J. Lightwave Tech.*, vol. 21, pp. 870–883, Apr. 2003.

[14] M. Medard, R. A. Barry, S. Finn, W. He, and S. Lumetta, "Generalized loop-back recovery in optical mesh networks," *IEEE/ACM Trans. Networking*, vol. 10, pp. 153–164, Feb. 2002.

[15] G. Ellinas, E. Bouillet, R. Ramamurthy, J. Labourdette, S. Chaudhuri, and K. Bala, "Routing and restoration architectures in mesh optical networks," *SPIE Optical Networks Mag.*, vol. 4, pp. 91–106, Jan./Feb. 2003.

[16] P.-H. Ho and H. Mouftah, "A framework for service-guaranteed shared protection in WDM mesh networks," *IEEE Communication Mag.*, vol. 40, pp. 97–103, Feb. 2002.

[17] C. Xin, Y. Ye, S. Dixit, and C. Qiao, "A joint working and protection path selection approach in WDM optical networks," in *Proc. IEEE Globecom*, vol. 4, pp. 2165–2168, 2001.

[18] G. Mohan, C. S. R. Murthy, and A. K. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 9, pp. 553–566, Oct. 2001.

[19] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, "Efficient distributed path selection for shared restoration connections," in *Proc. IEEE INFOCOM*, pp. 140–149, Jun. 2002.

[20] M. Kodialam and T. V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *Proc. IEEE INFOCOM*, vol. 2, pp. 902–911, Mar. 2000.

[21] C. Qiao and D. Xu, "Distributed partial information management (DPIM) schemes for survivable networks – Part I," in *Proc. IEEE INFOCOM*, pp. 302–311, June 2002.

[22] X. Su and C. Su, "An online distributed protection algorithm in WDM networks," in *Proc. IEEE ICC*, vol. 5, pp. 1571–1575, June 2001.

[23] C. Su and X. Su, "Protection path routing on WDM networks," in *Proc. OFC*, vol. 2, pp. TuO2–T1–3, Mar. 2001.

[24] A. Lardies, R. Gupta, and R. A. Patterson, "Traffic grooming in a multi-layer network," *SPIE Optical Networks Mag.*, vol. 2, pp. 91–99, May/Jun. 2001.

[25] S. Thiagarajan and A. K. Somani, "Traffic grooming for survivable WDM mesh networks," in *OptiCom*, pp. 54–65, 2001.

[26] R. Ramamurthy and B. Mukherjee, "Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks," *IEEE/ACM Trans. Networking*, vol. 10, pp. 351–367, June 2002.

[27] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, "Near-optimal approaches for shared-path protection in wdm mesh networks," in *Proc. IEEE ICC*, pp. 1320–1324, May 2003.

[28] R. Doverspike and J. Yates, "Challenges for MPLS in optical network restoration," *IEEE Communication Mag.*, vol. 39, pp. 89–96, Feb. 2001.

[29] S. Datta, S. Sengupta, S. Biswas, and S. Datta, "Efficient channel reservation for backup paths in optical mesh networks," in *Proc. IEEE Globecom*, pp. 2104–2108, 2001.

[30] J. Strand, A. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Communication Mag.*, vol. 39, pp. 81–87, Feb. 2001.

[31] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," in *Proc. IEEE INFOCOM*, vol. 2, pp. 699–708, Apr. 2001.

[32] E. Bouillet, J.-F. Labourdette, G. Ellinas, R. Ramamurthy, and S. Chaudhuri, "Stochastic approaches to compute shared mesh restored lightpaths in optical network architectures," in *Proc. IEEE INFOCOM*, pp. 801–807, Jun. 2002.

[33] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, "Optical network design and restoration," *Bell Labs Technical Journal*, vol. 4, pp. 58–84, Jan.-Mar. 1999.

[34] E. Bouillet, J.-F. Labourdette, R. Ramamurthy, and S. Chaudhuri, "Enhanced algorithm cost model to control tradeoffs in provisioning shared mesh restored lightpaths," in *Proc. OFC*, p. ThW2, Mar. 2002.

[35] C. Ou, K. Zhu, H. Zang, J. Zhang, H. Zhu, L. Sahasrabuddhe, and B. Mukherjee, "Traffic grooming for survivable WDM networks – dedicated protection," Tech. Report CSE-2003-12, May 2003, Dept. of Computer Science, University of California, Davis. Available at http://networks.cs.ucdavis.edu/∼ouc/tr/ou_tr03_1.pdf.

[36] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Management Science*, pp. 712–716, July 1971.

[37] M. Kodialam and T. V. Lakshman, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," in *Proc. IEEE INFOCOM*, vol. 1, pp. 376–385, Apr. 2001.

[38] C. Ou, K. Zhu, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, "Traffic grooming for survivable WDM networks – shared protection," Tech. Report CSE-2003-13, May 2003, Dept. of Computer Science, University of California, Davis. Available at http://networks.cs.ucdavis.edu/∼ouc/tr/ou_tr03_2.pdf.

PLACE
PHOTO
HERE

**Canhui (Sam) Ou** (S’02) received the B.S. degree from Peking University, Beijing, China, in 2000, and the M.S. degree from the University of California, Davis, in 2001. He is a Ph.D. candidate in the Computer Science Department, University of California, Davis. He worked as a summer intern at Sprint Advanced Technology Labs, Burlingame, CA, during Summer 01 and 02. His research interests include design and analysis of wavelength-routed WDM networks with focus on survivability and traffic grooming. His email address is: ouc@cs.ucdavis.edu.

PLACE
PHOTO
HERE

**Keyao Zhu** (S’98) received the B.S. degree from Peking University, Beijing (China) in 1998 and the M.S. degree from University of California, Davis (USA), in July 2000. Currently, he is a Ph.D. student in Computer Science Department, University of California, Davis, where he works as a research assistant in networks lab. under the supervision of professor Biswanath Mukherjee. His research interests include wavelength-routed WDM network design and analysis, WDM network protection, restoration, and traffic grooming.

PLACE
PHOTO
HERE

**Hui Zang** (S’97–M’02) received the B.S. degree in computer science from Tsinghua University, Beijing, China in 1997, and the M.S. and Ph.D. degrees in computer science from the University of California, Davis in 1998 and 2001, respectively. In 2000, she joined Sprint Advanced Technology Laboratories, Burlingame, CA, where she is a principal applied R&D engineer. She was one of the guest editors of IEEE Network special issue on “Traffic Engineering in Optical Networks.” She is the author of the book “WDM Mesh Networks–Management and Survivability” (Kluwer Academic, 2002). She has published over 30 conference papers and journal articles and currently has three US patents pending in the field of optical networking. Her research interests include WDM optical networks, IP over WDM networks, multi-service platforms and quality of service. Dr. Zang serves or has served as a Technical Committee member of IPoW’00–02, APOC’02, DRCN’03, OptiComm’03 and Globecom’03 Optical Networking and Systems Symposium. She also helped organizing OptiComm’02 as a panel co-chair. Her email address is: hzang@sprintlabs.com.

PLACE
PHOTO
HERE

**Laxman H. Sahasrabuddhe** Laxman Sahasrabuddhe received the B.Tech. degree from the Indian Institute of Technology, Kanpur (India) in 1992, the M.Tech. degree from the Indian Institute of Technology, Madras (India) in 1994, and the Ph.D. degree from the University of California, Davis (USA) in 1999. He is the receipient of the “Best Doctoral Dissertation Award”, from the College of Engineering, University of California, Davis, for his research on WDM Optical Networks. Currently he is a Principal Member of Technical Staff at SBC Communications, Inc.

PLACE
PHOTO
HERE

**Biswanath Mukherjee** (S’82–M’87) received the B.Tech. (Hons) degree from Indian Institute of Technology, Kharagpur (India) in 1980 and the Ph.D. degree from University of Washington, Seattle, in June 1987. At Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship. In July 1987, he joined the University of California, Davis, where he has been Professor of Computer Science since July 1995, and served as Chairman of Computer Science during September 1997 to June 2000. He is co-winner of paper awards presented at the 1991 and the 1994 National Computer Security Conferences. He serves or has served on the editorial boards of the IEEE/ACM Transactions on Networking, IEEE Network, ACM/Baltzer Wireless Information Networks (WINET), Journal of High-Speed Networks, Photonic Network Communications, and Optical Network Magazine. He also served as Editor-at-Large for optical networking and communications for the IEEE Communications Society. He served as the Technical Program Chair of the IEEE INFOCOM ’96 conference. He is author of the textbook “Optical Communication Networks” published by McGraw-Hill in 1997, a book which received the Association of American Publishers, Inc.’s 1997 Honorable Mention in Computer Science. He is a Member of the Board of Directors of IPLocks, Inc., a Silicon Valley startup company. He has consulted for and served on the Technical Advisory Board of a number of startup companies in optical networking. His research interests include lightwave networks, network security, and wireless networks. His e-mail address is: mukherje@cs.ucdavis.edu.