Extension of Segment Protection for Bandwidth Efficiency and Differentiated Quality of Protection in Optical/MPLS Networks

Canhui (Sam) Ou, Smita Rai, and Biswanath Mukherjee
Dept. of Computer Science, University of California, Davis, CA 95616, USA
Tel: +1 530-752-5129, Fax: +1 530-752-4767, Email: {ouc, rai, mukherje}@cs.ucdavis.edu

***Abstract*— This paper investigates the problem of dynamic survivable lightpath provisioning against single node/link failures in optical mesh networks employing wavelength-division multiplexing (WDM).**

We unify various forms of segment protection into generalized segment protection (GSP). In GSP, the working path of a lightpath is divided into multiple overlapping working segments, each of which is protected by a node/link disjoint backup segment. We design an efficient heuristic which, upon the arrival of a lightpath request, dynamically divides a judiciously-selected working path into multiple overlapping working segments and computes a backup segment for each working segment while accommodating backup sharing. Compared to the widely-considered share-path protection scheme, GSP achieves much lower blocking probability and shorter protection-switching time for a small sacrifice in control and management overhead.

Based on generalized segment protection, we present a new approach to provisioning lightpath requests according to their differentiated quality-of-protection (QoP) requirements. We focus on one of the most important QoP parameters—namely, protection-switching time—since lightpath requests may have differentiated protection-switching-time requirements. For example, lightpaths carrying voice traffic may require 50-ms protection-switching time while lightpaths carrying data traffic may have a wide range of protection-switching-time requirements. Numerical results show that our approach achieves significant performance gain which leads to a remarkable reduction in blocking probability.

While our focus is on optical WDM network, the basic ideas of our approaches can be applied to Multi-Protocol Label Switching (MPLS) networks with appropriate adjustments, e.g., differentiated bandwidth granularities.

***Index Terms*—Optical network, WDM, lightpath, survivability, shared segment protection, quality of protection.**

This work has been supported in part by NSF Grant No. ANI-98-05285. Part of this work was presented at the IFIP Networking 2004 conference, Athens, May 2004.

I. INTRODUCTION

In a wavelength-routed optical network, the failure of a network element (e.g., fiber, crossconnect, etc.) can cause the failure of several lightpaths, thereby leading to large data and revenue loss. Protection—a proactive procedure in which spare capacity is reserved during lightpath setup [4], [6], [20], [21], [22], [28]—is essential for recovering from such failures in a short time period, e.g. 50 ms. Protection schemes can be classified by the type of routing used (link-based versus path-based) and by the type of resource sharing (dedicated versus shared). A path carrying traffic during normal operation is known as a *working* path[1]. When a working path fails, the lightpath is rerouted over a *backup* path. High bandwidth efficiency and short protection-switching time are two of the most important and desirable features of a protection scheme [14], where protection-switching time for a lightpath is the time duration the network takes to properly signal/configure the nodes along the backup path before switching traffic to the backup path after a failure occurs on the working path [28].

We consider the problem of dynamic survivable lightpath provisioning against single node (crossconnect) and single link (fiber) failures. Specifically, we focus on shared protection (because of its desirable resource efficiency) with the assumptions that existing lightpaths cannot be disturbed and no knowledge of future arrivals is available at the time of provisioning the current lightpath request. While we consider full wavelength-convertible networks here, the extension to the wavelength-continuous case is straightforward.

Much work has been conducted on dynamic shared protection [8], [23], [25], [34] in optical WDM networks and on dynamic routing of restorable bandwidth-guaranteed connections in MPLS networks [11], [12], [13], [26]. A widely-considered approach—shared-path

[1]Working path is also referred to as primary path, active path, and service path in the literature.

protection [28]—is bandwidth efficient due to backup sharing. Consequently, how to increase backup sharing based on different cost models and route-computation techniques is of particular interest and has been reported in [3], [12], [16], [19], [30], [31], [33]. The complexity of shared-path protection is high as shown in [5], [25] that it is NP-complete to find a working path and a backup path for a new lightpath request when backup sharing with existing backup paths is allowed. As a result, practical heuristics are usually employed.

One possible limitation of shared-path protection is that backup paths may sometimes become longer due to backup sharing [3]. Consequently, protection-switching time may increase because of longer backup paths. The relation between backup sharing and backup-path hop distance for path protection have been shown to be that one trades off another in [3], [34].

Furthermore, lightpath requests may have differentiated protection-switching-time requirements. For example, lightpaths carrying voice traffic may require 50-ms protection-switching time while lightpaths carrying data traffic may have a wide range of protection-switching-time requirements. Due to the path-wise node-/link-disjoint nature of path protection, shared-path protection may not provision lightpath requests according to their protection-switching-time requirements effectively in practical-sized networks [17], [24]. Clearly, proper mechanisms are needed to provision such lightpath requests in a resource-efficient manner.

Motivated by the above considerations, we first unify various forms of segment protection into generalized segment protection (GSP) and propose an effective heuristic in Section II. Then, based on GSP, we present a new and effective approach to provisioning lightpath requests according to their protection-switching-time requirements while taking into account backup sharing in Section III.

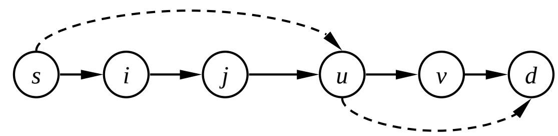
While our focus is on optical WDM network, in which the bandwidth requirement of a lightpath request is one wavelength, our approaches can also be directly applied to MPLS networks for provisioning restorable, bandwidth-guaranteed connections of differentiated bandwidth granularities with appropriate adjustments.

II. Generalized Segment Protection

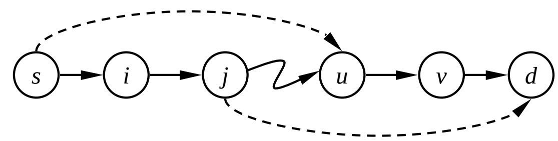
Below, we unify various forms of segment protection into generalized segment protection (GSP) in Section II-A, design an efficient heuristic in Section II-B, and demonstrate the effectiveness of GSP in Section II-C.

A. Generalized Segment Protection

Various forms of segment protection were reported in [1], [7], [9], [29], [32]. The approaches proposed



(a) Non-overlapping segment protection as in [1], [29], [32].



(b) Overlapping segment protection as in [7], [9].

Fig. 1. Various forms of segment protection. The solid lines from node s to node d represent the working path, and the dashed lines represent the backup segments. While only two segments are shown in these illustrations, in general, a path may employ many segments. Also, each backup segment may have several additional intermediate nodes, which are not shown here to avoid cluttering.

in [1], [29], [32] addressed single-link failures by dividing a working path into a sequence of non-overlapping segments and protecting each such segment individually. As shown in Fig. 1(a), the lightpath from node s to node d is partitioned into two non-overlapping segments: one with working segment $\langle s, i, j, u\rangle$ and backup segment $\langle s, u\rangle$; another with working segment $\langle u, v, d\rangle$ and backup segment $\langle u, d\rangle$. (Note that each backup segment may have several additional intermediate nodes, which are not shown in Fig. 1 to avoid cluttering.) When a failure occurs, only the affected segment performs protection switching and the other unaffected segments are oblivious to the failure. For example, in Fig. 1(a), if link $\langle i, j\rangle$ fails, the working segment $\langle s, i, j, u\rangle$ switches to its backup segment $\langle s, u\rangle$, and the other working segment $\langle u, v, d\rangle$ is unaware of the failure. Node failures are not accommodated in these approaches as consecutive non-overlapping segments share the same node failure, e.g., node u in Fig. 1(a).

Even though node failures do not occur as often as link failures, they need to be carefully treated because the impact of node failures is much more disastrous than that of link-failures. The work in [7], [9] handled single-node/link failures by dividing a working path into a sequence of overlapping segments and protecting each such segment separately. As shown in Fig. 1(b), the lightpath from node s to node d is partitioned into two overlapping segments: one with working segment $\langle s, i, j, \cdots, u\rangle$ and backup segment $\langle s, u\rangle$; and another with working segment $\langle j, \cdots, u, v, d\rangle$ and backup segment $\langle j, d\rangle$.

We unify the above approaches into generalized segment protection, which is almost the same as the overlapping segment protection shown in Fig. 1(b) except that node j and node u can be the same node. GSP differs from the previous approaches in that it can dynamically divide a working path into multiple segments while accommodating backup sharing, as will be elaborated later in Section II-B.

Similar to segment protection, GSP has a number of advantages compared to path protection.

The end-to-end protection entity is *a segment* in segment protection as opposed to *a path* in path protection. When a failure occurs along a working path (segment), the source node of that path (segment) switches to its backup. Since a segment is typically shorter than a path in terms of hop count, segment protection is expected to have shorter protection-switching time.

Meanwhile, two segments (or two lightpaths in path protection) can share backup wavelength links as long as their working segments (or working paths in path protection) do not share the same node/link failure. Since, in general, a segment is shorter than a path, the probability of two working segments sharing the same risk is typically lower than the probability of two working paths sharing the same risk. As a result, segment protection can have better backup sharing compared to shared-path protection.

Furthermore, segment protection has more flexibility in routing compared to path protection since path protection is a special case of segment protection in which every lightpath has exactly one segment. Apart from that, it is clear that the longer the working path is, the more difficult it is to find a node-disjoint backup path [18]. Later, we shall show that an improperly-selected working path can partition a network and no end-to-end (with respect to lightpath) node-disjoint backup path can be found. Thus, it is desirable to have shorter working path to achieve routing flexibility, as is the case in segment protection.

Next, we design an effective heuristic to compute a route for an incoming connection request.

B. The GSP Heuristic

Upon the arrival of a new lightpath request, the network management system needs to compute a working path l_w and a list of backup segments $\{l_b^i\}$, which divide the working path into overlapping segments $\{l_w^i\}$ such that l_w^i and l_b^i are node-/link- disjoint. New backup segments $\{l_b^i\}$ can share wavelength links with existing backup segments as well as among themselves. Unfortunately, it is NP-hard to determine if there exists an eligible solution as we have proved the NP-completeness of the existence version of shared-path protection, which is a special case of segment protection with the number of segments being one [25]. As a result, we resort to a heuristic.

Below, we first define the notations, and then present a practical heuristic which, upon the arrival of a new lightpath request, dynamically divides a judiciously-selected working path into multiple overlapped working segments and computes a backup segment for each working segment while accommodating backup sharing.

1) Notations: A network is represented as a weighted, directed graph $G = (V, E, C, \lambda)$, 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), and $\lambda : E \rightarrow Z^+$ specifies the number of wavelengths on each link (where Z^+ denotes the set of positive integers).

A conflict set is associated with a link to identify the sharing potential between backup segments[2]. The conflict set ν_e for link e defines the set of nodes traversed by such working segments whose backup segments utilize wavelengths on link e. The conflict set ν_e for link e can be represented as an integer set, $\{\nu_e^u | \ \forall u \in V, 0 \leq \nu_e^u \leq \lambda(e)\}$, where ν_e^u specifies the number of working segments which traverse node u and are protected by link e (or, in other words, their corresponding backup segments traverse link e). The number of wavelengths reserved for backup segments on link e is thus $\nu_e^* = \underset{\forall u}{max}\{\nu_e^u\}$. Clearly, the union of the conflict sets for all the links aggregates the per-segment-based information, and the size of the conflict set depends only on the number of links, not on the number of segments.

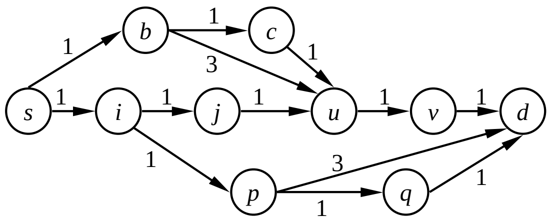
2) GSP Heuristic: The route-computation approaches in [1], [9], [29], [32] partition a working path in a fixed manner, e.g., every working path is divided into a constant number of segments or into multiple segments of equal hop count. A flexible partitioning approach in [7] dynamically divides a working path into overlapping segments, but does not take into account backup sharing. Our GSP heuristic extends the idea in [7] to incorporate backup sharing and to facilitate partitioning a working path into overlapping working segments.

Our GSP heuristic is specified in detail in Algorithm 1. In Algorithm 1, K is an input constant representing the maximum number of candidate working paths; and ϵ is a small constant such as 0.01. The value of ϵ is used to control the degree of backup sharing: smaller values encourage backup sharing and larger values discourage backup sharing [3].

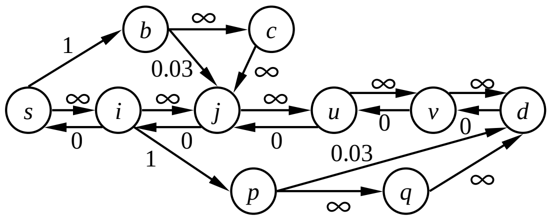
The basic ideas of our GSP heuristic are as follows.

1) Select a candidate working path l_w^k and transform the original graph based on l_w^k in a way such that any path link-disjoint to l_w^k in the transformed graph can be mapped back to the original graph and decomposed into a set of backup segments $\{l_b^{k,i}\}$, which partitions l_w^k into multiple working segments $\{l_w^{k,i}\}$ where consecutive segments

[2] The conflict set is similar to the conflict vector in [23], the aggregated square matrix in [19], and the “bucket” link metric in [31].



(a) A network state $G = (V, E, C, \lambda)$ in which two lightpaths—lightpath one with working path $\langle b, c, u\rangle$ and backup path $\langle b, u\rangle$ and lightpath two with working path $\langle p, q, d\rangle$ and backup path $\langle p, d\rangle$—are already set up. Every link has one wavelength.



(b) $G' = (V, E', C', \lambda)$ for working path $\langle s, i, j, u, v, d\rangle$ (assuming $\epsilon = 0.01$ in Algorithm 1).

Fig. 2. Illustration of the GSP heuristic. The number besides a link represents the cost of that link.

overlap by *at least* one hop and $l_w^{k,i}$ is node-/link- disjoint to $l_b^{k,i}$.

2) Consider the worst-case-scenario backup sharing when computing the backup segments but precisely allocate backup resources after the backup segments are computed. Basically, when computing the backup segments for working path l_w^k, we consider the worst case (as far as backup sharing is concerned) where l_w^k is one segment since the working segments cannot be determined without the backup segments. Later, when the list of backup segments $\{l_b^{k,i}\}$ is computed, backup sharing is performed on a per-segment basis in Steps 4 and 5 as the list of working segments $\{l_w^{k,i}\}$ can be determined.

Figure 2 highlights some distinct features of our GSP heuristic. When a new lightpath request from node s to node d arrives at the network state shown in Fig. 2(a), the only candidate working path is $\langle s, i, j, u, v, d\rangle$. We observe that shared-path protection cannot find a solution as there is no path which is end-to-end disjoint to path $\langle s, i, j, u, v, d\rangle$.

However, our heuristic can find a solution. Figure 2(b) shows the transformed graph $G' = (V, E', C', \lambda)$. Following Step 2b in Algorithm 1, the minimal-cost path $\langle s, b, j, i, p, d\rangle$ will be computed as l_b^k, which will then be mapped back to G and decomposed as two backup segments $\langle s, b, u\rangle$ and $\langle i, p, d\rangle$ (note that link $\langle b, j\rangle$ in G' was constructed from link $\langle b, u\rangle$ in G). This example highlights that our GSP heuristic can dynamically divide a working path into multiple arbitrary overlapping working

Algorithm 1 GSP

Input: $G = (V, E, C, \lambda)$, $\nu = \{\nu_e \mid e \in E\}$, s, d, K

Output: A working path l_w and a list of backup segments $\{l_b^i\}$ which partitions l_w into overlapping segments $\{l_w^i\}$ such that l_w^i & l_b^i are node-/link- disjoint; otherwise NULL if no eligible solution is found.

1) **select candidate working paths**: compute up to K minimal-cost paths $L_w = \{l_w^k \mid 1 \leq k \leq K\}$ in G from s to d based on Yen's K-shortest paths algorithm [35] subject to the constraint that every hop along a path should have at least one free wavelength; return NULL if L_w is empty
2) **compute backup segments for each candidate working path** l_w^k in L_w as follows:
   a) transform $G = (V, E, C, \lambda)$ to $G' = (V, E', C', \lambda)$:
      i) define link-cost function $C''(e)$ for $e \in E$:

$$C''(e) := \begin{cases} \infty & \text{if } l_w^k \text{ traverses link } e, \text{ or } \\ & \nu_e^* = \nu_e^u \text{ for some node } u \\ & \text{along } l_w^k \text{ and link } e \text{ does not} \\ & \text{have any free wavelength} \\ \epsilon \times C(e) & \text{if for any node } u\ (u \neq s, d) \\ & \text{traversed by } l_w^k,\ \nu_e^u < \nu_e^* \\ C(e) & \text{otherwise} \end{cases}$$

      ii) define link-set E' and $C'(e)$ for $e \in E'$:
         – $\forall \langle u, v\rangle \in E \wedge u \neq s \wedge u \neq d$, if l_w^k traverses node v but not node u, then add link $\langle u, p\rangle$ to E', where node p is v's immediate predecessor along l_w^k, and let $C'(\langle u, p\rangle) = C''(\langle u, v\rangle)$; otherwise, add $\langle u, v\rangle$ to E' and let $C'(\langle u, v\rangle) = C''(\langle u, v\rangle)$
         – $\forall \langle u, v\rangle \in E$ and l_w^k traverses $\langle u, v\rangle$, if $\langle v, u\rangle \notin E$, then add $\langle v, u\rangle$ into E' and let $C'(\langle v, u\rangle) = 0$
   b) compute a least-cost path l_b^k from s to d in G'
   c) map l_b^k back to G and decompose l_b^k into a list of backup segments $\{l_b^{k,i}\}$ which partitions l_w^k into overlapped working segments $\{l_w^{k,i}\}$
   d) $\forall i$, compute the amount of fresh wavelength links backup segment $l_w^{k,i}$ consumes: for any link e that $l_b^{k,i}$ traverses, if $\nu_e^u = \nu_e^*$ for some node u along $l_w^{k,i}$ (excluding the source and destination nodes of $l_w^{k,i}$), then increase the amount of fresh wavelength links by one
3) select the pair $\langle l_w^k, \{l_b^{k,i}\}\rangle$ of minimal cost; return NULL if no such pair exists
4) **allocate resources for** $\langle l_w^k, \{l_b^{k,i}\}\rangle$: allocate a new wavelength along l_w^k and update backup wavelengths for every $l_b^{k,i}$: for any link e that $l_b^{k,i}$ traverses, if $\nu_e^u = \nu_e^*$ for some node u along $l_w^{k,i}$ (excluding the source and the destination nodes of $l_w^{k,i}$), then reserve one more wavelength on link e as backup resources
5) **update the conflict set**: $\forall i$, update conflict set associated to links traversed by $l_b^{k,i}$: for every link e that $l_b^{k,i}$ traverses, $\nu_e^u \leftarrow \nu_e^u + 1$ for every node u along $l_w^{k,i}$ (excluding the source node and the destination node of $l_w^{k,i}$)
6) return l_w^k and $\{l_b^{k,i}\}$

segments and protect each working segment separately while accommodating backup sharing.

3) Computational Complexity: Algorithm 1 has a computational complexity of $O(K \cdot (|V|^3 + |E|))$. In particular, the complexity of Step 1 is $K \cdot |V|^3$; the complexity of Step 2 is $O(K \cdot |E|)$ (the computational complexities of Steps 2a, 2b, 2c, and 2d are $O(|E|), O(|V|^2), O(|E|)$, and $O(|E|)$, respectively); the complexities of Steps 3, 4, 5, and 6 are $O(1), O(|E|), O(|E|)$, and $O(1)$. If $K = 1$, then the complexity of Step 1 can be reduced from $K \cdot |V|^3$ to $O(|V|^2)$. Consequently, the complexity of Algorithm 1 can be reduced to $O(|V|^2 + |E|)$, which is the complexity of shortest-path algorithms.

C. Illustrative Numerical Results

We now quantitatively evaluate GSP. We simulate a dynamic network environment with the assumptions that the lightpath-arrival process is Poisson and the lightpath-holding time follows a negative exponential distribution. In every experiment, 10^6 lightpath requests are simulated; they are uniformly distributed among all node pairs; average lightpath-holding time is normalized to unity; the cost of any link is unity; and our example network topology with 16 wavelengths per fiber is shown in Fig. 3. For the results shown in this section, $\epsilon = 0.01$ since we aim to maximize backup sharing. We remark that more results from different topologies also led to the same observations. Those results are not shown here.

Below, we compare GSP to shared-path protection, which is widely considered to be the most resource-efficient protection scheme so far. Since shared-path protection is a special case of GSP, the heuristic for shared-path protection differs from Algorithm 1 only in Step 2a and Step 2b (some steps such as Step 2c can be removed for efficiency because the number of segments in shared-path protection case is always one). Step 2a is modified as follows: temporarily remove all the nodes traversed by l_w^k (except node s and node d) and all the links sourced/sunk at the removed nodes. Step 2b is modified as follows: compute a minimal-cost path l_b^k from node s to node d in G with link-cost function C''.

1) Blocking Probability: Figure 4 compares the blocking probability of GSP to that of shared-path protection for $K = 1, 2$, and 3. We make the following observations: (a) GSP has much lower blocking probability than shared-path protection for the same K. This is because GSP can achieve better backup sharing and have more flexibility in routing, as discussed earlier in Section II-A. (b) When K increases from one to two, the reduction in blocking probability for both GSP and shared-path protection is significant while the reduction is only marginal when K further increases from two to three. This is basically the effect of alternate routing: the performance improvement is significant when the number of alternate routes increases from one to two, and the improvement

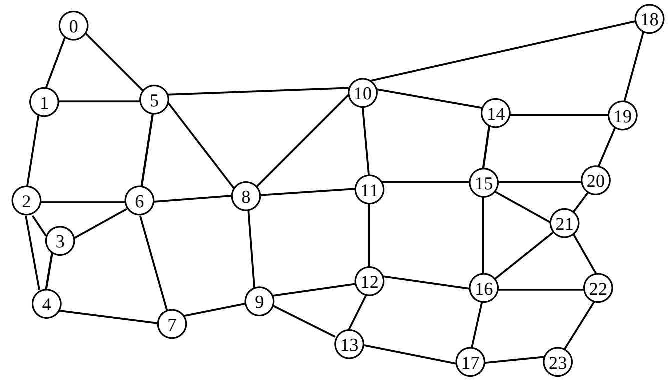


Fig. 3. A representative topology whose average hop distance is about 2.99 and average nodal degree is about 3.58.

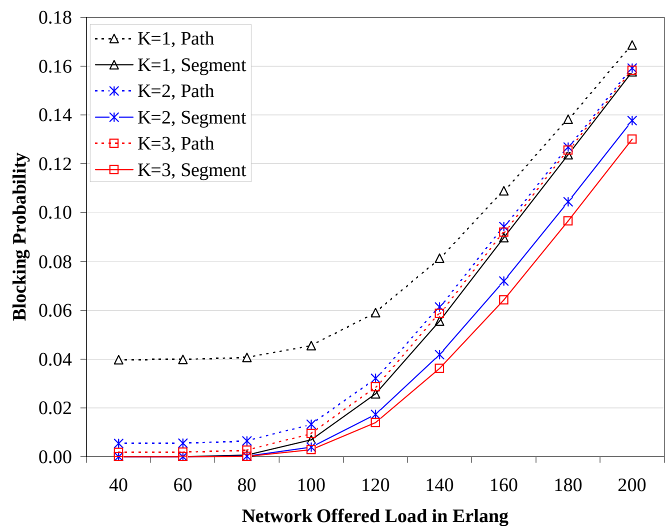


Fig. 4. Blocking probability. The average link utilization for 40 Erlangs is about 17% and for 200 Erlangs is about 65%.

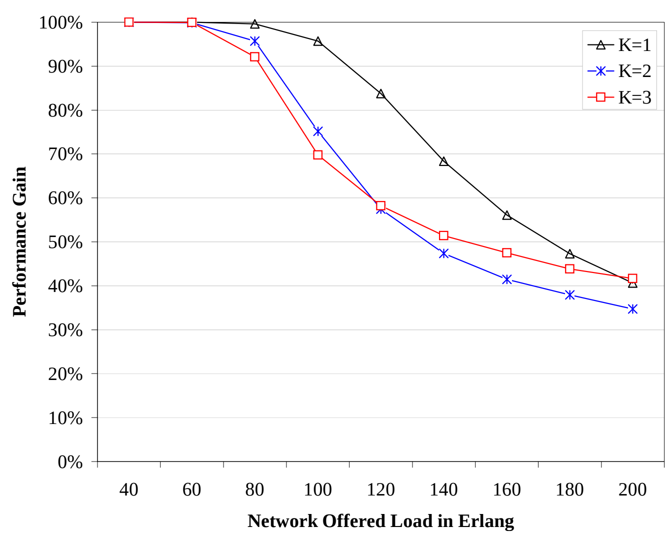


Fig. 5. Performance gain of GSP over shared-path protection.

is marginal or negligible when the number of alternate routes increases further [27]. (c) Shared-path protection can have modest blocking probability even at low load, e.g., 40 Erlangs, when $K = 1$. Similar effect was also observed in [18], [25]. The reason is that an improperly-selected working path can disconnect the network and a backup path, which should be end-to-end node-/link- disjoint to the working path, can not be found. In some cases, the least-cost paths turn out to be improper working paths. For example, the least-cost path from node 0 to node 13 in Fig. 3 is $\langle 0, 5, 8, 9, 13 \rangle$. Clearly, there is no path node-disjoint to $\langle 0, 5, 8, 9, 13 \rangle$ from node 0 to node 13. As a result, a lightpath request from node 0 to node 13 may be blocked in shared-path protection when $K = 1$. However, in GSP, backup segments $\langle 0, 1, 2, 6, 8 \rangle$ and $\langle 5, 10, 11, 12, 13 \rangle$ form a valid solution among others.

2) Performance Gain: Performance gain is defined as the percentage of lightpath requests which are blocked in shared-path protection but can be accepted by GSP. Performance gain can be calculated as follows. Whenever shared-path protection needs to block a lightpath request, we apply GSP to check whether the same lightpath request can be provisioned under the same network state (but we do not set up the lightpath request even if it can be provisioned). Figure 5 shows that GSP achieves significant performance gain over shared-path protection. This is because the routing constraint in GSP is node-/link- disjoint segment-wise; but, in shared-path protection, it is node-/link- disjoint path-wise, as discussed earlier in Section II-A.

3) Protection-Switching Time: For shared-path protection, protection-switching time for a lightpath can be calculated based on the hop count of the working and backup paths of the lightpath, as shown in [2], [24], [28]. For GSP, protection-switching time for a segment can be calculated based on the hop count of the working and backup segments using the same methodology as in [2], [24], [28] since the protection entity is a segment. Figure 6 shows that the average hop count of working and backup segments in GSP is much smaller than the average hop count of working and backup paths in shared-path protection. As a result, protection-switching for GSP is faster than that of shared-path protection.

4) Control and Management Complexity: The control and management complexity might be higher in GSP than in shared-path protection since the number of segments is typically more than the number of lightpaths. Figure 7 plots the average number of segments per lightpath, which is quite small (about 1.2) in this numerical example. As a result, the increase in control and management complexity is modest. Furthermore, by aggregating the per-segment

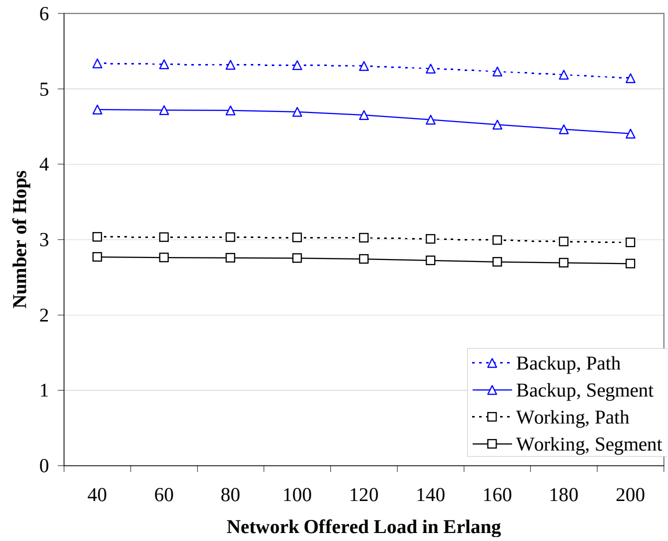


Fig. 6. Working/backup hop count ($K = 2$). The plots for $K = 1$ and $K = 3$ are similar to the one for $K = 2$ here.

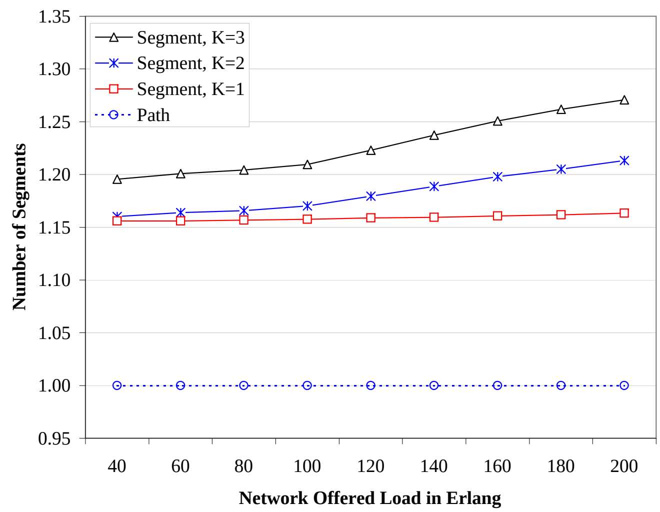


Fig. 7. Number of segments per lightpath.

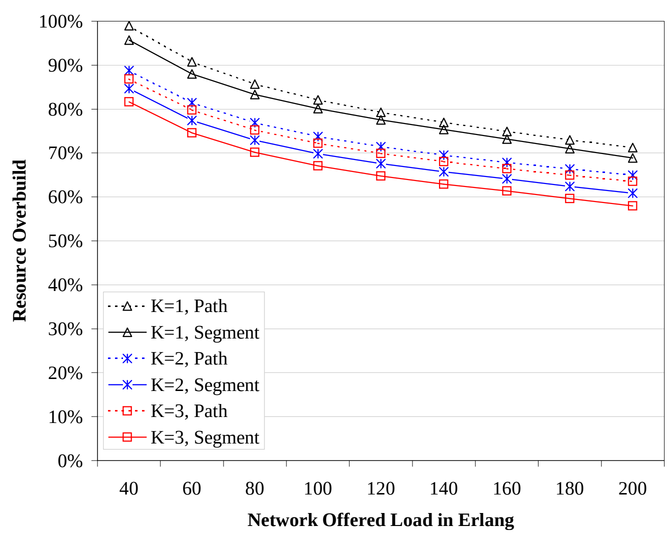


Fig. 8. Resource overbuild.

information, we find that the size of the conflict set, which was defined in Section II-B, depends only on the number of links, not on the number of segments.

5) Resource Efficiency: *Resource overbuild*, defined as the amount of wavelength links consumed by backup paths over the amount of wavelength links utilized by working paths [16], indicates the amount of extra resources needed for providing protection as the percentage of the amount of resources required without protection. Typically, it is desirable to have lower resource overbuild because lower resource overbuild implies higher backup sharing. Figure 8 shows that GSP has lower resource overbuild than shared-path protection. The fact that a segment is shorter than a path contributes to increased backup sharing, and thus, decreased resource overbuild.

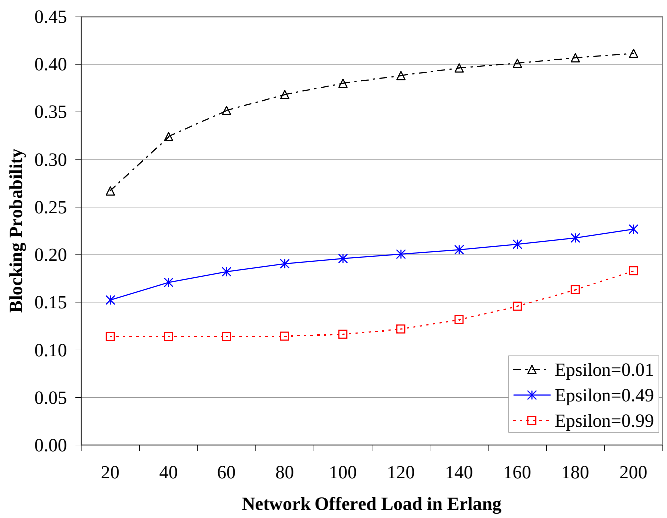


Fig. 9. Shared-path protection ($K = 2$ and $H_b = 6$).

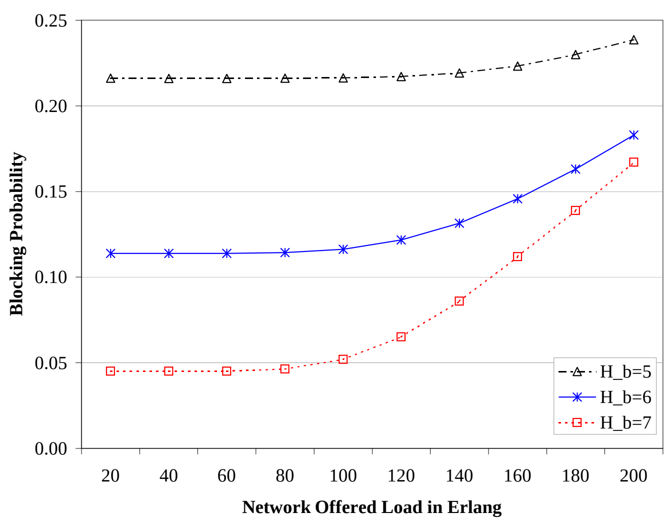


Fig. 10. Shared-path protection ($K = 2$ and $\epsilon = 0.99$).

III. Providing Differentiated Quality of Protection (QoP) Based on Generalized Segment Protection

GSP can be employed for provisioning differentiated quality of protection (QoP). Here, we focus on one of the most important QoP parameters, namely protection-switching time. The protection-switching time of a shared-path protected lightpath can be calculated from the hop count of the working/backup paths, as shown in [2], [24], [28]. Therefore, we consider QoP in terms of hop count.

Below, we first argue why new mechanisms are needed to provision differentiated QoP in Section III-A; in Section III-B, we present a new approach to provision lightpath requests according to their QoP requirements; in Section III-C, we evaluate the performance of our approach.

A. Motivation

Lightpath requests may have differentiated protection-switching-time requirements. For example, lightpaths carrying voice traffic may require 50-ms protection-switching time while lightpaths carrying data traffic may have a wide range of protection-switching-time requirements. While some lightpath requests (which carry mission-critical information) can be dedicated protected, it is not economically viable to provide dedicated protection to each lightpath request due to its excessive resource requirement.

Below, we show that shared-path protection may not be able to provide the desired level of protection-switching time either. Let us consider a simple case in which the backup-path hop count of any lightpath cannot exceed a constant H_b (ignoring the constraint on working path for now). A lightpath request will be blocked if the computed backup path is longer than H_b hops. Figure 9 plots the blocking probability of shared-path protection for $H_b = 6$ for the network shown in Fig. 3 with different values of ϵ. Recall that ϵ is the parameter used by the link-cost function in computing a shared backup path. Figure 9 confirms the conclusion in [3], [34] that a larger value of ϵ leads to shorter backup path but decreased backup sharing, and a smaller value of ϵ leads to increased backup sharing but longer backup path (results for other values of ϵ follow the same trend, so they are not shown here). Please note that $H_b = 6$ for this network is reasonably large since the average backup-path hop count is about 5.2 for $\epsilon = 0.01$ as shown in Fig. 6. However, regardless of the values of ϵ, the blocking probability in Fig. 9 is quite high. The main reason is that some lightpath requests are blocked because their backup paths span more than $H_b = 6$ hops. Figure 10 shows the impact of H_b on shared-path protection. While the blocking probability drops significantly as H_b increases, the blocking for $H_b = 7$, which is quite large, is still unacceptable. (Please note that we chose $\epsilon = 0.99$ in Fig. 10 to discourage detouring of backup paths. The blocking probability of shared-path protection with hop-count constraint will be even worse if we chose a smaller value of ϵ.) As network size grows, it is clear that shared-path protection cannot achieve reasonable blocking for practical values of H_b due to its fundamental limitation: the backup path has to be end-to-end node-/link-disjoint to the working path.

Obviously, new mechanisms are needed. Below, we introduce more intelligence into GSP for provisioning lightpath requests to support differentiated QoP.

B. GSP_QoP Heuristic

We present a heuristic which applies GSP in a way such that the hop count of any backup segment is no more than H_b. For a candidate working path l_w^k, our heuristic, called GSP_QoP, performs the following recursive procedure to compute a list of eligible backup segments. 1) Starting from node s, compute a least-cost path to all the other nodes along l_w^k, where the cost function is C_1 defined in Algorithm 2. 2) Starting from node d and following the reverse direction of l_w^k, find the first node v which satisfies the constraint that the least-cost path from node s to node v is of at most H_b hops. 3) If node v is the destination node d, the heuristic succeeds and terminates; otherwise, starting from all the nodes between node s and node v (excluding nodes s and v) along l_w^k, recursively apply the above procedure (if there is no node between node s and node v, the heuristic fails).

Our GSP_QoP heuristic is specified in detail in Algorithm 2. For a node $u \in V$, $PC(u)$ denotes the cost of the least-cost path destined to node u; $HC(u)$ represents the hop count of the least-cost path; and $PH(u)$ records the previous hop along the least-cost path. For a path l_w, $Head(l_w)$ returns the first node along l_w.

We make the following remarks. 1) In Algorithm 2, the candidate working path l_w^k is given. This is just for the purpose of simplifying the presentation; in our implementation, we dynamically compute K candidate working paths as in Algorithm 1, execute Algorithm 2 for each candidate working path, and select the working path and the list of backup segments of minimal cost.

2) There are two objectives in computing a node-/link-disjoint backup segment. Objective one is to find a backup segment of hop count no more than H_b. Objective two is to select the backup segment of least cost. In general, constraint-based path selection with multiple objectives is NP-complete [15].

3) Backup sharing in this case is more tricky as the situation shown in Fig. 11 can arise. In the case of GSP (without the constraint of H_b), the path $\langle s,x,p,q,y,d\rangle$ could be a valid backup segment and this type of situation may not occur typically. In the presence of H_b, the path $\langle s,x,p,q,y,d\rangle$ is not valid when $H_b = 4$. However, the two segments $\langle s,x,p,q,u\rangle$ and $\langle j,p,q,y,d\rangle$ so formed are still valid. Our heuristic accommodates this type of backup sharing in Steps 3 and 7 since the freshly-reserved backup wavelengths for a newly-computed backup seg-

---

Algorithm 2 GSP_QoP

Input: $G = (V, E, C, \lambda)$, $\nu = \{\nu_e \mid e \in E\}, s, d$, a candidate working path l_w^k

Output: a list of backup segments $\{l_b^i\}$, each of which spans no more than H_b hops and they collectively partition l_w^k into overlapped segments $\{l_w^{k,i}\}$ such that $l_w^{k,i}$ and l_b^i are node-/link-disjoint; otherwise, NULL if no such list is found.

1) $S \leftarrow \{s\}, L_b \leftarrow \phi, l_w \leftarrow l_w^k, i \leftarrow 0$
2) $V' \leftarrow V;\ \forall u \in S, PC(u) \leftarrow 0, HC(u) \leftarrow 0, PH(u) \leftarrow NULL;\ \forall u \in V \wedge u \notin S, PC(u) \leftarrow \infty, HC(u) \leftarrow \infty, PH(u) \leftarrow NULL;\ \ i \leftarrow i+1$
3) define link-cost function $C_1(e)$, $e \in E$, with respect to l_w:

$$C_1(e) := \begin{cases} +\infty & \text{if } l_w \text{ traverses link } e\text{, or } \nu_e^* \text{ is equal to } \nu_e^u \text{ for some node } u \text{ along } l_w \text{ and link } e \text{ does not have any free wavelength} \\ \epsilon \times C(e) & \text{if for any node } u\ (u \neq s, d) \text{ traversed by } l_w,\ \nu_e^u < \nu_e^* \\ C(e) & \text{otherwise} \end{cases}$$

4) while ($V' \neq \phi$) do {
    $u \leftarrow \arg \min_{u \in V'} \{PC(u)\}, V' \leftarrow V' - \{u\}$
    if ($u = Head(l_w)$) or (l_w does not traverse u) {
        $\forall\ v \in V'$, s.t. $\langle u, v\rangle \in E$
        if $PC(v) > PC(u) + C_1(\langle u, v\rangle)$ then {
            $PC(v) \leftarrow PC(u) + C_1(\langle u, v\rangle)$
            $HC(v) \leftarrow HC(u) + 1$
            $PH(v) \leftarrow u$
        } // if
    } // if
   } // while
5) starting from node d and following the reverse direction of l_w, find the first node v which satisfies $HC(v) \leq H_b$
6) retrieve the least-cost path destined to node v by following $PH(v)$ and denote the path as l_b^i
7) allocate backup wavelengths along l_b^i: let l_w^i be the working segment starting from $Head(l_w)$ and ending at v along l_w (inclusively); for any link e that l_b^i traverses and for any node u along l_w^i (excluding the source and the destination nodes of l_w^i), $\nu_e^u \leftarrow \nu_e^u + 1$; if $\nu_e^u > \nu_e^*$, then reserve one more wavelength on link e and let $\nu_e^* \leftarrow \nu_e^u$
8) if v is d, then return $\{l_b^i\}$
9) $S \leftarrow$ all the nodes between $Head(l_w)$ and node v along l_w, excluding $Head(l_w)$ and node v; if S is empty, then undo any changes made to G in Step 7 and return NULL
10) $l_w \leftarrow$ the path starting from node v to node d along l_w
11) go to Step 2

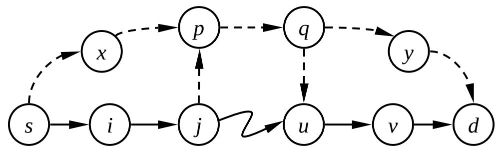


Fig. 11. Two backup segments, $\langle s, x, p, q, u\rangle$ and $\langle j, p, q, y, d\rangle$, of the same lightpath share the same wavelength link on link $\langle p, q\rangle$, assuming $H_b = 4$.

ment is used for computing later backup segments for the same lightpath request.

4) Sometimes, it may be desirable that the hop count of any working segment plus the hop count of its backup segment is no more than some constant H. We can modify Step 5 to cater to this constraint as follows. For any node v along path l_w, denote as $l_b^{i,v}$ the least-cost path destined to node v and denote as h_w^v the number of hops from $Head(l_b^{i,v})$ to node v along l_w. Starting from node d and following the reverse direction of l_w, find the first node v which satisfies the constraint $HC(v) + h_w^v \leq H$. Other constraints based on combinations of working and backup segment hop count also can be easily incorporated.

Computational Complexity: The computational complexity of Algorithm 2 is $O(|V|^2 + |E|)$. In particular, the computational complexities for Steps 1-11 are $O(1)$, $O(|V|)$, $O(|E|)$, $O(|V|^2 + |E|)$, $O(|V|)$, $O(|V|)$, $O(|V|)$, $O(|E|)$, $O(|V|)$, $O(|V|)$, and $O(1)$, respectively. If we compute K candidate working paths and execute Algorithm 2 for each candidate working path, then the computational complexity is $O(K \cdot (|V|^3 + |E|))$.

C. Illustrative Numerical Results

We compare our GSP_QoP approach to shared-path protection under the same network configuration as described in Section II-C. For the illustrative results shown here, we use $K = 2$ as we found the performance improvement is marginal if we increase K to any larger value.

1) Blocking probability under different values of ϵ: Figure 12 plots the blocking performance for $H_b = 6$ under $\epsilon = 0.01, 0.49$, and 0.99. Recall that ϵ is the parameter used by the link-cost function in computing a shared backup path, and smaller values of ϵ lead to better backup sharing. We observe that our GSP_QoP approach has drastically lower blocking probability than shared-path protection under the same ϵ. We further observe that large values of ϵ, e.g., $\epsilon = 0.49$ or $\epsilon = 0.99$, are preferable as both GSP_QoP and shared-path protection have significantly lower blocking when ϵ has a large value. Later, we shall use large values of ϵ.

Figure 13 shows the performance gain, defined in Section II-C.2, for $H_b = 6$ under $\epsilon = 0.01, 0.49$, and 0.99.

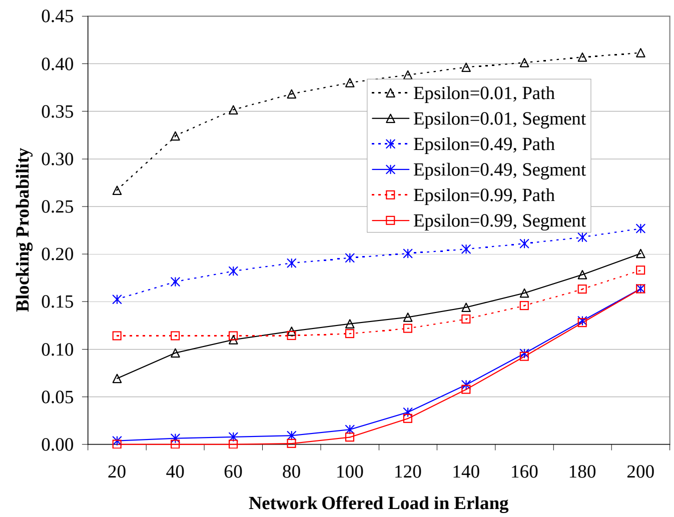


Fig. 12. Blocking probability for $H_b = 6$.

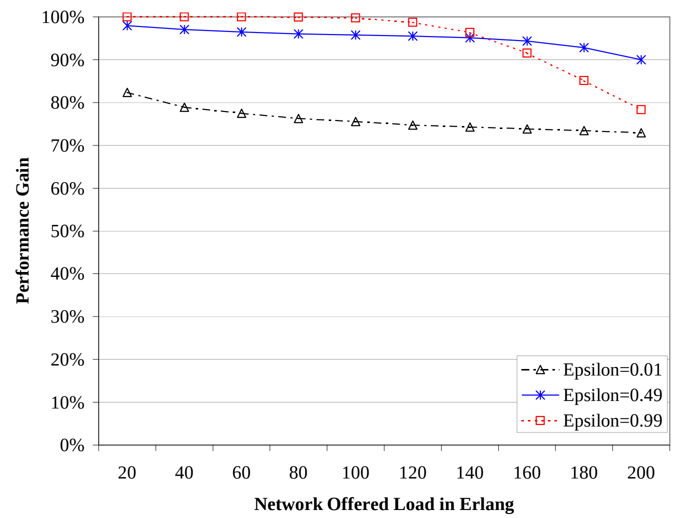


Fig. 13. Performance gain of GSP_QoP over shared-path protection for $H_b = 6$.

We observe that GSP_QoP has a remarkable performance gain (over 70% across all load regions). The huge performance gain results from the fact that GSP_QoP relaxes the path-wise node-/link- disjointness to segment-wise node-/link- disjointness and computes segments with respect to H_b.

2) Blocking probability under different values of H_b: Figure 14 examines the impact of H_b on both GSP_QoP and shared-path protection with $\epsilon = 0.99$. We observe that: (a) GSP_QoP has much lower blocking probability when the load is not very high. (b) When H_b increases from 5 to 6, GSP_QoP has noticeable reduction in blocking probability while the reduction is marginal when H_b further increases to 7. (c) As H_b increases, the blocking probability of shared-path protection drops significantly. However, shared-path protection still has remarkable blocking (above 4%) even when network offered load

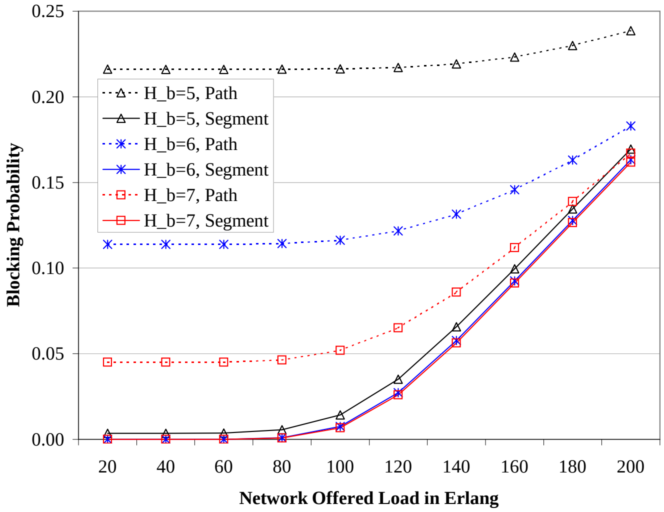


Fig. 14. Blocking probability for $\epsilon = 0.99$.

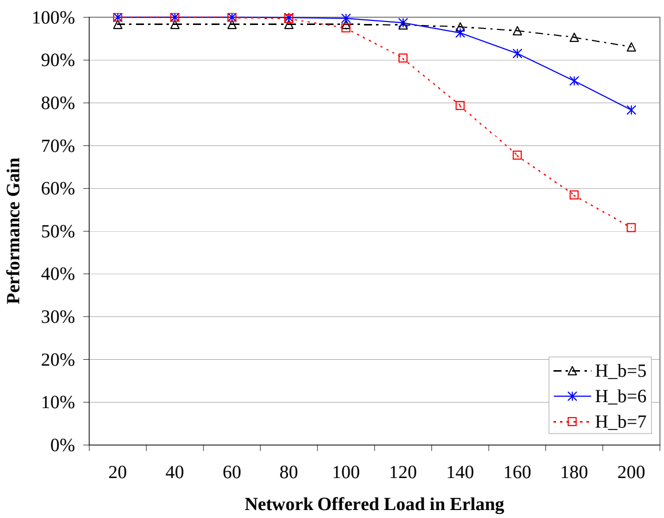


Fig. 15. Performance gain for $\epsilon = 0.99$.

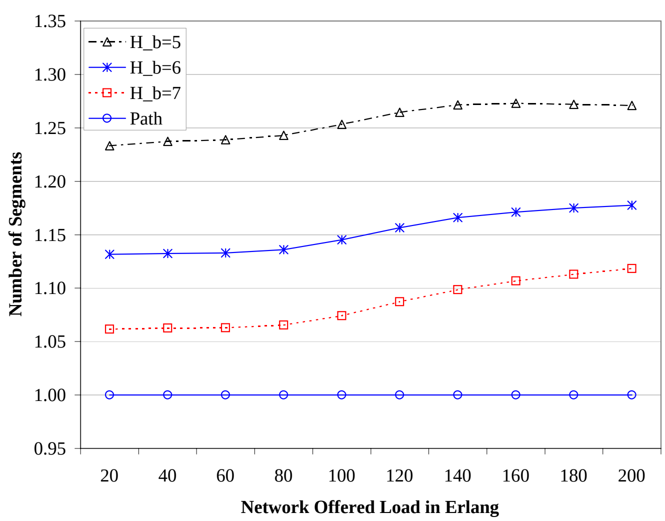


Fig. 16. Number of segments per lightpath for $\epsilon = 0.99$.

is low, e.g., 20 Erlangs which translate to about 8.5% average link utilization. This is due to the path-wise end-to-end node-/link- disjoint nature of shared-path protection.

The effectiveness of our GSP_QoP can be further observed in Fig. 15 in terms of performance gain. When load is modest or low, GSP_QoP achieves close to 100% performance gain. Even when load is high and wavelengths are heavily used, GSP_QoP still achieves more than 50% performance gain.

Figure 16 shows that more segments are needed for smaller value of H_b. However, the average number of segments per lightpath is still quite low, e.g., less than 1.3 even for $H_b = 5$. This implies that the control and management overhead due to segmentation is not very significant.

3) Blocking probability for lightpath requests with differentiated QoP requirements: Different lightpath requests may have different QoP requirements, as discussed earlier in Section III-A. Figures 17 and 18 compare the performance of GSP_QoP to shared-path protection under two types of traffic. The QoP of the lightpath requests in terms of H_b follows the distribution $5 : 6 : 7 : \infty = 30 : 20 : 10 : 40$ in Type 1 and $5 : 6 : 7 : \infty = 10 : 20 : 20 : 50$ in Type 2.

GSP_QoP has much lower blocking probability than shared-path protection, as shown in Fig. 17. For shared-path protection, the large difference between the blocking probability for the two types of traffic implies that shared-path protection cannot effectively provision lightpath requests based on their differentiated QoP requirements. In contrary, the difference between the blocking probability for the two types of traffic in GSP_QoP is very small. This indicates that GSP_QoP can properly provision lightpath requests according to their differentiated QoP requirements.

As shown in Fig. 18, when load is modest or low, GSP_QoP achieves close to 100% performance gain; even when load is high, GSP_QoP still achieves more than 35% performance gain.

4) Blocking probability for different values of H: Sometimes, it may be desirable that the hop count of any working segment plus the hop count of its backup segment be no more than some constant H. This can be easily incorporated into GSP_QoP as discussed earlier.

Figures 19 and 20 examines the impact of H on GSP_QoP and shared-path protection. The curves in Figs. 19–20 have similar trend to the ones in Figs. 14–15 and can be explained similarly. Meanwhile, since H applies to both working and backup segments, as opposed to H_b which applies only to backup segments, the H constraint is more stringent than the H_b constraint. As a result, the performance gain for different values of H is even higher—namely, above 70% across all load regions—as

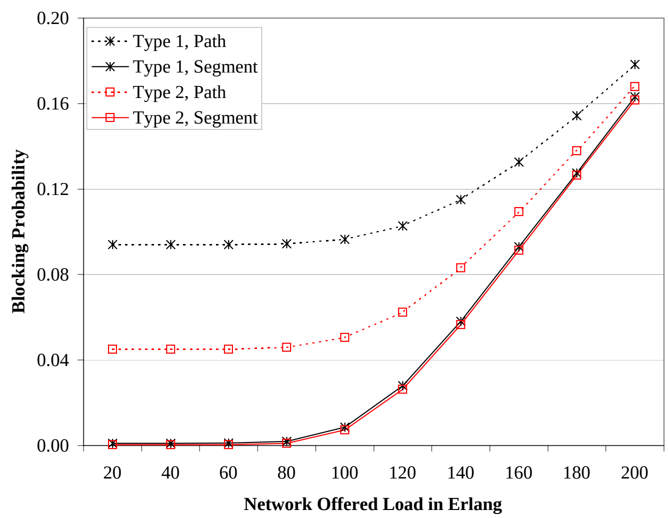


Fig. 17. Blocking probability ($\epsilon = 0.99$). In Type 1, H_b follows $5 : 6 : 7 : \infty = 30 : 20 : 10 : 40$; In Type 2, H_b follows $5 : 6 : 7 : \infty = 10 : 20 : 20 : 50$.

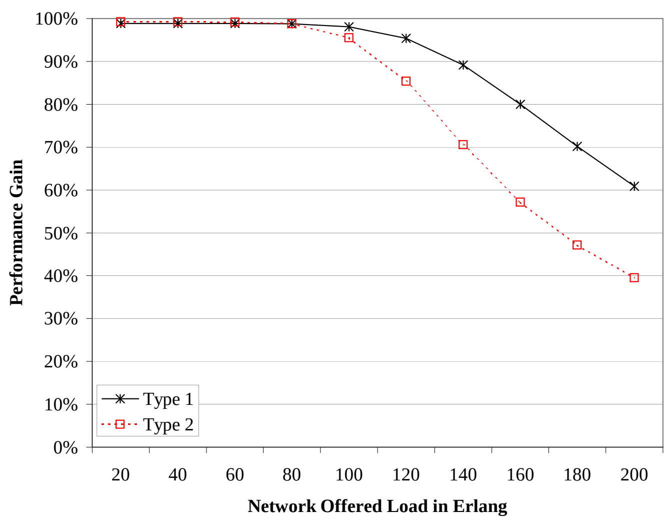


Fig. 18. Performance gain ($\epsilon = 0.99$). In Type 1, H_b follows $5 : 6 : 7 : \infty = 30 : 20 : 10 : 40$; In Type 2, H_b follows $5 : 6 : 7 : \infty = 10 : 20 : 20 : 50$.

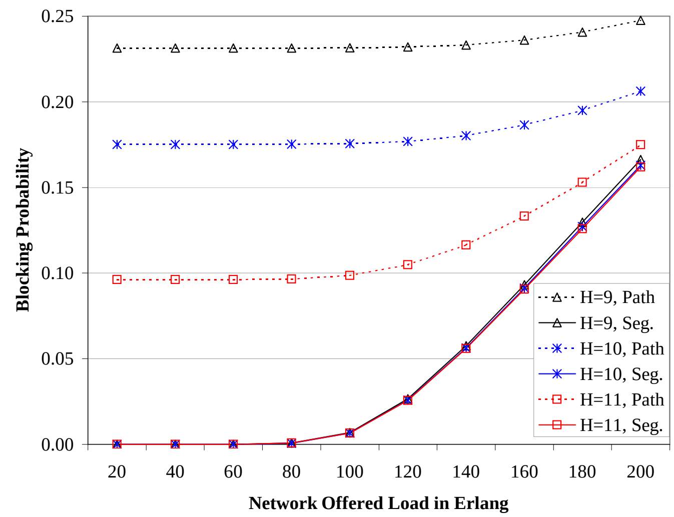


Fig. 19. Blocking probability ($\epsilon = 0.99$).

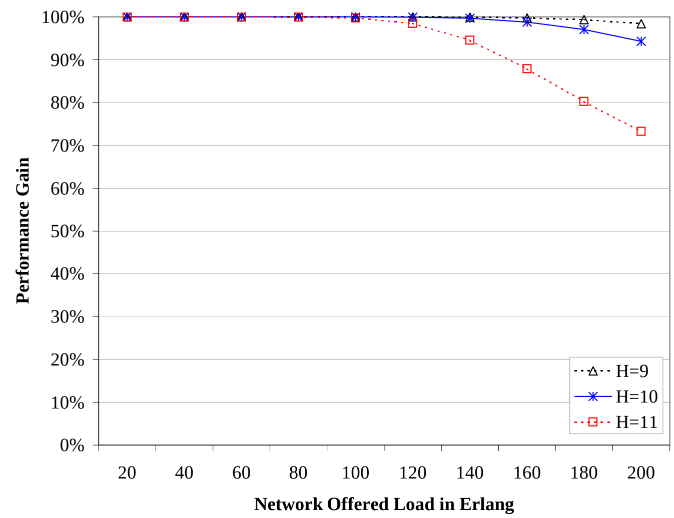


Fig. 20. Performance gain ($\epsilon = 0.99$).

shown in Fig. 20 since segmented protection leads to shorter working and backup segments as shown earlier in Fig. 6.

IV. Conclusion

This paper considered the problem of dynamic survivable lightpath provisioning against single node/link failures in optical WDM mesh networks. We unified various forms of segment protection into generalized segment protection (GSP). We designed an efficient heuristic which, upon the arrival of a lightpath request, dynamically divides a judiciously-selected working path into multiple overlapped working segments and computes a backup segment for each working segment while accommodating backup sharing. Comparison between GSP and shared-path protection demonstrated that, for a little sacrifice in control and management overhead, GSP achieves much lower blocking probability and shorter protection-switching time.

Based on generalized segment protection, we presented a new approach to provisioning lightpath requests according to their differentiated quality-of-protection (QoP) requirements. We focused on one of the most important QoP parameters—namely, protection-switching time—since lightpath requests may have differentiated protection-switching-time requirements. Numerical results showed that our approach achieves significant performance gain which leads to remarkable reduction in blocking probability.

While our focus is on WDM network, our approaches can also be applied to MPLS networks with appropriate variations, e.g., differentiated bandwidth granularities.

REFERENCES

[1] V. Anand, S. Chauhan, and C. Qiao, "Sub-path protection: A new framework for optical layer survivability and its quantitative evaluation," Dept. of CSE, State University of New York at Buffalo, Tech. Report 2002-01, Jan. 2002.

[2] C. Assi, Y. Ye, A. Shami, S. Dixit, and M. Ali, "Efficient path selection and fast restoration algorithms for shared restorable optical networks," in *Proc. IEEE ICC*, pp. 1412-1416, May 2003.

[3] 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.

[4] 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.

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

[6] A. Fumagalli, M. Tacca, F. Unghvary, and A. Farago, "Shared path protection with differentiated reliability," in *Proc. IEEE ICC*, pp. 2157–2161, May 2002.

[7] K. Gummadi, M. Pradeep, and C. S. R. Murthy, "An efficient primary-segmented backup scheme for dependable real-time communication in multihop networks," *IEEE/ACM Trans. Networking*, vol. 11, pp. 81–94, Feb. 2003.

[8] O. Hauser, M. Kodialam, and T. V. Lakshman, "Capacity design of fast path restorable optical networks," in *Proc. IEEE INFOCOM*, vol. 2, pp. 817–826, June 2002.

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

[10] P.-H. Ho and H. T. Mouftah, "A novel strategy for protection domain allocation in dynamic mesh WDM networks," in *Proc. IEEE ICC*, pp. CQ22–1, May 2003.

[11] K. Kar, M. Kodialam, and T. V. Lakshman, "Routing restorable bandwidth guaranteed connections using maximum 2-route flows," in *Proc. IEEE INFOCOM*, pp. 113–121, June 2002.

[12] 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.

[13] 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.

[14] S. Koo and S. Subramaniam, "Trade-offs between speed, capacity, and restorability in optical mesh network restoration," in *Proc. OFC*, pp. 487–489, Mar. 2002.

[15] F. Kuipers, P. V. Mieghem, T. Korkmaz, and M. Krunz, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Communication Magazine*, vol. 40, pp. 50–55, Dec. 2002.

[16] 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.

[17] J. Li, H. Park, and H. Lee, "Shared sub-path protection with overlapped protection areas in WDM networks," in *Proc. OFC*, p. FQ7, Mar. 2003.

[18] Y. Liu and D. Tipper, "Successive survivable routing for node failures," in *Proc. IEEE GLOBECOM*, vol. 4, pp. 2093–2097, Nov. 2001.

[19] 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.

[20] S. S. Lumetta, M. Medard, and Y.-C. Tseng, "Capacity versus robustness: a tradeoff for link restoration in mesh networks," *IEEE/OSA J. Lightwave Technology*, vol. 18, pp. 1765–1775, Dec. 2000.

[21] Y. Luo and N. Ansari, "Restoration with wavelength conversion in WDM networks," *Electronics Letters*, vol. 38, pp. 900–901, Aug. 2002.

[22] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: a new approach to the design of WDM-based networks," *IEEE J. Selected Areas in Communications*, vol. 20, pp. 800–809, May 2002.

[23] 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.

[24] C. Ou, H. Zang, and B. Mukherjee, "Sub-path protection for scalability and fast recovery in optical WDM mesh networks," in *Proc. OFC*, pp. 495–496, Mar. 2002. (Also, IEEE JSAC, to appear, 2004.)

[25] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, and B. Mukherjee, "New and improved approaches for shared-path protection in WDM mesh networks," *IEEE/OSA J. Lightwave Technology*, vol. 22, pp. 1223–1232, May 2004.

[26] 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.

[27] 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.

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

[29] M. Sridharan, R. Srinivasan, and A. Somani, "On improving partial information routing with segmented path protection," in *International Conference on Parallel Processing*, pp. 193–198, Aug. 2002.

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

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

[32] A. Todimala and B. Ramamurthy, "A dynamic partitioning sub-path protection routing technique in WDM mesh networks," in *Proc. International Conf. on Computer Comm.*, Aug. 2002.

[33] H. Wang, E. Modiano, and M. Medard, "Partial path protection for WDM networks: end-to-end recovery using local failure information," in *Proc. International Symposium on Computers and Communications (ISCC 2002)*, pp. 719–725, July 2002.

[34] Y. Xiong, D. Xu, and C. Qiao, "Achieving fast and bandwidth-efficient shared-path protection," *IEEE/OSA J. Lightwave Technology*, vol. 21, pp. 365–371, Feb. 2003.

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