Efficient shared-segment protection exploiting the knowledge of connection holding time

Massimo Tornatore*, Matteo Carcagni†, Canhui (Sam) Ou‡, Biswanath Mukherjee†, Achille Pattavina*

* Department of Electronics and Information, Politecnico di Milano, Via Ponzio 34-35, 20121 Milan, Italy
{tornator,pattavina}@elet.polimi.it, matteo.carcagni@gmail.com
†Department of Computer Science, University of California, Davis, CA 95616, USA, {mukherje}@cs.ucdavis.edu
‡ AT&T Services, Inc. 2600 Camino Ramon, San Ramon, CA USA, CanhuiOu@gmail.com

Abstract—Progress in network technologies and protocols is paving the road towards flexible optical transport networks, in which leasable circuits could be set up and released on a short-term basis. Thus we consider it reasonable that, at least for some types of services, the holding time of connection requests can be known in advance. In this paper, we propose to exploit the knowledge of connection-holding time to improve the performance of an algorithm for shared-segment protection (SSP). For a typical US nationwide network, we compared our approach to an holding-time-unaware, but otherwise shared segmented efficient, approach, obtaining savings on resource overbuild of up to 7% for practical scenarios.

Index Terms—Optical network, WDM, lightpath, dynamic traffic, holding time, shared-segment protection.

I. INTRODUCTION

Optical networks provide a transport infrastructure with very high capacity, thanks to wavelength-division-multiplexing (WDM) technology. The huge bandwidth of WDM also requires efficient survivability mechanisms, because the failure of a network element (usually a node or a link failure) can cause a large amount of data loss. Recently, new techniques have been proposed to efficiently deal with this problem in mesh networks. Among them, shared segmented protection (SSP) is a promising candidate because of its desirable resource efficiency and effective backup sharing [1].

Nowadays, the evolution of the prevalently static traffic towards a more dynamic traffic paradigm offers network operator new challenges, but also opportunities, to efficiently accommodate the capacity needed for protection. So far optical transport networks have been supporting connections, which are provided and leased for long period of time, e.g. a year; in the coming years, new applications are likely to ask for a more flexible bandwidth market, where limited-time leasing of bandwidth, e.g., for important sport or social events, video distribution, or massive data transfer for backup or storage purposes would be available on-demand.

Technology is developing to provide the flexible platform the new applications are asking for: new agile optical crossconnects (OXC) are emerging to create mesh-structured optical WDM backbone networks; and, in order to manage and control dynamic WDM networks, new control protocols have been proposed: ASON/ASTN and G-MPLS are protocol-independent, control-plane architectures (standardized by ITU and IETF, respectively) which allow to setup, configure, and release connections automatically. Thus, we consider it reasonable to expect that the connection holding time could be known in advance, mainly based on service-level agreements or contracts between the network operator and its customers. In this paper, we propose to exploit the knowledge of connection-holding time to design an efficient algorithm for dynamic provisioning of shared-segment-protected connections in optical mesh networks employing WDM. In particular, shared backup-channel capacity reservation can achieve significant advantage by exploiting the additional information associated with connection durations. This opportunity has been already exploited for shared-path protection in [2] achieving significant improvements in network resource utilization. We have chosen to compare our algorithm to Generalized Segment Protection (GSP) [3], which has been shown to be very efficient for shared segmented protection, but GSP is holding-time unaware. For a typical US nationwide network, we obtained savings on resource overbuild of up to 7% for various practical scenarios, compared to the holding-time-unaware, but otherwise shared-path-efficient, GSP approach.

The rest of this paper is organized as follows. Section II discusses some fundamental issues on shared-segment protection and formally states the problem. Section II describes the baseline GSP approach. Section IV presents our new algorithm, called PHOTO-GSP: the connection-holding-time aware link-cost-assignment method as an extension of the standard case is discussed. Section V evaluates by simulations the performance of PHOTO-GSP compared to the GSP algorithm. Section VI concludes the paper.

II. SHARED-SEGMENT PROTECTION

Various forms of segment protection have been proposed in Refs [4], [1], [3], [5]. The common idea of these approaches is to divide a working path (WP) into several working segments (WSs) and to protect each WS with a link-node-disjoint backup segment (BS). When a failure occurs, only the affected WS switches to its BS, and the other WSs are unaware of the failure. In addition, in shared-segment protection, two BSs can...
share backup wavelength links as long as their WSs do not traverse the same link.

SSP can be classified as overlap SSP, if the WS are allowed to overlap on some links, and no-overlap SSP, if WSs are strictly link-disjoint (see Fig. 1). SSP 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. Since a segment is typically shorter than a path in terms of hop count, segment protection is expected to have shorter protection-switching time and 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 path-shared path protection.

Depending on the strategy used to partition a WP, segment protection can be classified into predetermined partitioning, postdetermined partitioning, and integrated partitioning [6]. In this paper, we will focus on the postdetermined partitioning strategy proposed in [3], leaving for further works the analysis of predetermined and integrated partitioning cases.

A. Problem Statement

We consider dynamic traffic. Upon the arrival of a new connection request, the network management system needs to compute a working path \( l_w \) and a list of backup paths \( \{l_b^k\} \), which divide the working path into overlapping segments \( \{l_b^k\} \) such that \( l_w \) and \( l_b^k \) are node-/link- disjoint. New backup segments \( \{l_b^k\} \) 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: in [7], the authors 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. As a result, we need to resort to heuristics.

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 a function that maps the elements in \( E \) to positive real numbers representing the link costs, and \( \lambda : E \rightarrow Z^+ \) specifies the number of wavelengths on each link (where \( Z^+ \) denotes the set of positive integers). \( \lambda^e_f \) denotes the number of free wavelengths on link \( e \in E \).

A conflict set is associated with a link to identify the sharing potential between backup segments. The conflict set \( \nu_e \) for link \( e \) defines the set of nodes traversed by such working segments whose backup segments utilize wavelengths on link \( e \). The conflict set \( \nu_e \) for link \( e \) can be represented as an integer set \( \{\nu_e^e \in E, 0 \leq \nu_e^e \leq \lambda(e')\} \), where \( \nu_e^e \) specifies the number of working paths that traverse node \( u \) and are protected by link \( e \) (i.e., their corresponding backup paths traverse link \( e \)). The number of wavelengths to be reserved for backup paths on link \( e \) is thus \( \nu_e = \max_{e \in V} \{\nu_e^e\} \).

III. GSP: A Solution Without Holding-Time Consideration

GSP is a practical heuristic which, upon the arrival of a new connection 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.

With respect to previous approaches that partition the working path in a fixed manner [1], GSP is able to flexibly choose the segments to be protected by extending an idea first proposed in [8] to incorporate backup sharing.

The details of the algorithm can be found in [3]. In the following, due to the space limit of this article, we provide only the basic steps of the GSP algorithm:

1) select candidate working paths: compute up to \( K \) admissible minimal-cost paths \( L_w = \{l_w^k \mid 1 \leq k \leq K\} \) in \( G \) based on the K-shortest-paths algorithm.

2) compute backup segments for each candidate working path \( l_w^k \) in \( L_w \) as follows:

a) define new link-cost function \( C''(e) \) for \( e \in E \):

\[ C''(e) := \begin{cases} +\infty & \text{if } e \in l_w^k \lor (\lambda^e_f = 0 \land (\exists u \in l_w^k, \nu_u^e = \nu_e)) \\ \epsilon \times C(e) & \text{if } \forall (u \neq s, d) \in l_w^k, \\ \nu_u^e \leq \nu_e^e \text{ \( C(e) \) Otherwise} \\ \end{cases} \]

b) transform the original graph \( G = (V,E,C,\lambda) \) in \( G' = (V,E',C',\lambda) \) according to a set of rules reported in [3]

c) compute a least-cost \( l_b^k \) in \( G' \)

d) map \( l_b^k \) in \( G \) and decompose \( l_b^k \) into a list of backup segments \( \{l_b^{k,i}\} \)

3) select the pair \( \{l_w^k, \{l_b^{k,i}\}\} \) of minimal cost.

Let us focus on the link-cost assignment for backup-segments routing in Step 2(a). Once a working path is fixed, we need to set an appropriate link-cost assignment to route the corresponding backup path. By means of the conflict set, we know the amount of backup capacity reserved on link \( e \): if the entire backup capacity is needed to protect the failure of a link along the candidate working path (i.e., \( \exists u \in l_w^k, \nu_u^e = \nu_e^e \)), then the link has to be considered as not shareable. In this case, if there is free capacity (\( \lambda^e_f > 0 \)), then a new wavelength can be allocated and link cost is the full cost \( C(e) \). Otherwise, if (\( \lambda^e_f = 0 \)), the link is not usable and its cost is set to infinite. Finally, if there are backup channels other than those reserved to protect a link along the working path (condition expressed by “if \( \forall u (u \neq s, d) \in l_w^k, \nu_u^e < \nu_e^e \)”), then the link is shareable and we don’t need to allocate a new channel on it. So, the cost is set to a very low value by multiplying \( C(e) \) with a small constant \( \epsilon \).
Actually, the state of a given link (i.e., if the link is shareable or not) will be likely to change during the holding time of the incoming connection: this happens for example when some existing connections depart or new connections arrive. So, a link, which is considered as shareable at the arrival instant of a new connection, may, during the lifetime of the incoming connection, assume a different state due to the deallocation of backup capacity because of connection departures. In the next section, we will see how to keep track of the evolution of the link state to propose a more-efficient link-cost assignment.

IV. PROVISIONING BY HOLDING-TIME OPPORTUNITY APPLIED TO GSP (PHOTO-GSP)

In this study, we have assumed that information on future connections may not be known in advance (this further information would lead to the static scheduling problem). Nevertheless, we could exploit at least the information about the departure events, which is simply retrievable from the knowledge of the connection-holding time.

By introducing holding-time-awareness in a routing algorithm, the usual assignment of link costs can be improved. Traditional approaches try to minimize the amount of additional wavelengths used by the new incoming connection; our proposal consists of minimizing the cost of additional wavelengths used by the new incoming connection; the algorithm, the usual assignment of link costs can be improved.

A. PHOTO Link-Cost Assignment

In order to follow step-by-step the changes in the link states, we introduce the new symbols \( \nu_e(\Delta t_k) \), \( \nu^u_e(\Delta t_k) \), and \( C''(e, \Delta t_k) \), which express the values of \( \nu_e \), \( \nu^u_e \), and \( C''(e) \), respectively, in the interval of time \( \Delta t_k \).

Let us define \( \Delta t_k \) first. Let us suppose, without loss of generality, that the remaining holding times (at instant \( t_a \)) of the \( n \) connections already in the network are ordered so that \( h_{t_i} \leq h_{t_{i+1}} \), \( i = 1, 2, \ldots, n \). Let us suppose also that, for a certain index \( m \), \( h_{t_m} - 1 < h_t \), but \( h_{t_m} \geq h_t \); so we can set \( h_{t_m} = h_t \). As a consequence, \( \tau = \{ \tau_0, \tau_1, \ldots, \tau_m \} = \{0, h_{t1}, h_{t2}, \ldots, h_{tm} \} \) will indicate the departure events before incoming connection departure \( t_h \), and \( \Delta t_k = \tau_k - \tau_{k-1} \) expresses the time interval between two departures. Conflict set \( \nu_e(\Delta t_k)^1 \), \( \nu^u_e(\Delta t_k) \), and associated cost \( C_h(e, \Delta t_k) \) will be updated according to the \( k \)-th connection departure. In other words, we have divided the holding time \( t_a \) into a series of time intervals \( \Delta \tau \) which express the distance between two departures. The cost of link \( e \) during the interval \( \Delta t_k \) will be re-evaluated after having considered the departure of the \( k \)-th connection, according to the following scheme:

\[
C''(e, \Delta t_k) := \begin{cases} 
+\infty & \text{if } e \in t_{h}^{t_{u}} \lor (\lambda_{\tau}^e = 0 \land \exists u \in t_{l}^{t_{u}}. \nu_u(\Delta t_k) = \nu_{e}^u(\Delta t_k)) \\
\epsilon \times C(e) & \text{if } \forall u \in t_{l}^{t_{u}}. \nu_u(\Delta t_k) < \nu_{e}^u(\Delta t_k) \\
C(e) & \text{otherwise}
\end{cases}
\]

\( ^1 \nu_e \) will be equal to \( \nu_e(\Delta t_1) \) (where \( \Delta t_1 = \tau_1 - \tau_0 = h_{t1} - 0 \)).

It is worth noting that 1) all of the previously-cited parameters are evaluated at the instant of arrival of the incoming connection, 2) future arrivals are not known, and 3) departure events after \( t_a + t_h \) can be neglected. Overall cost \( C_h(e) \) in PHOTO will be evaluated by considering each connection departure and by summing the cost contribution due to any time interval in the following manner:

\[
C''(e) = \frac{1}{l_k} \sum_{k=1}^{m} (\Delta t_k) \cdot C''(e, \Delta t_k) \quad e \in E
\]

Hence, the new cost function to be minimized is not only the wavelength utilization at the arrival instant, but a more meaningful estimation, which considers the wavelength usage along the entire holding period.

B. A Numerical Example

As an example, we refer to the network in Fig. 2(a): the two connections \( r_1 \) and \( r_2 \) already exist at instant \( t_a = 0 \). They are characterized by holding times \( t_1 = 30, t_2 = 20 \), respectively. At instant \( t_a = 10 \), connection \( r_3 \) has to be provisioned (its working route is set on the path \( (j, k, n, x) \)): the remaining holding times for the two previous connections are \( h_{t1} = 20 \) and \( h_{t2} = 10 \). Clearly, \( h_{t1} > h_{t2} \) and routing \( r_3 \)'s backup path along links used by the backup of \( r_1 \) would lead to a longer period of backup resource sharing.

If we apply the link-cost assignment of GSP, we obtain the two alternative candidate backup paths shown in Fig. 2(b): backup path \( bp1 \) along the route \( (j, b, n, z, x) \) and backup path \( bp2 \) along \( (j, b, n, k, m, x) \). Since the cost of a new wavelength on a link has been set to 1, while the cost of a shared wavelength is \( \epsilon \), the cost of \( bp1 \) and \( bp2 \) would be the same: \( 2 + 2 \times \epsilon \). But, as previously pointed out, the \( bp2 \) would minimize the allocated resource in the network. Since the shared link \( (b, n) \) is used by both the routes, we can demonstrate this last assertion by simply observing that the behavior of backup capacity reserved on link \( m, z \) and link \( m, x \): namely, \( bp2 \) on link \( m, x \) will share backup capacity with connection \( r_1 \) for a longer time than \( bp1 \) on link \( m, z \) with connection \( r_2 \). This implies that an additional wavelength will be reserved on link \( n, z \) for a time interval equal to \( h_{t1} \). If we choose the \( bp2 \), an additional wavelength will be reserved on link \( (g, h) \) for \( h_{t2} \). Since \( t_h - h_{t1} \geq t_h - h_{t2} \), the \( bp1 \) requires additional capacity for the time duration \( h_{t1} - h_{t2} \), while \( bp2 \) does not require the additional capacity during this time interval.

Therefore, we should exploit the information provided by the holding time to modify the link-cost assignment. PHOTO evaluates the cost of link \( m, x \) and link \( m, z \) by considering the evolution of cost in three different time intervals, as follows (see Fig. 3):

- \( \Delta t_1 \) (from time 10 to 20): connection \( r_2 \) does not require a new wavelength on link \( m, z \) and link \( m, x \) for its backup path, because \( r_1 \) and \( r_2 \) are disjoint to \( r_3 \). So, the link cost during \( \Delta t_1 \) will be \( \epsilon \Delta t_1 C(e) \) (shareable state) for both link \( m, z \) and link \( m, x \).
Fig. 2. In this network example, GSP link-cost assignment does not allow to identify which of the two alternatives would be the likely more convenient choice. Holding-time-aware routing algorithm PHOTO succeeds instead.

- $\Delta \tau_2$ (from time 20 to 30): connection $r_2$ does not require an additional wavelength on link $\langle m, x \rangle$ for its backup path, because $r_2$ is still in the network, while it requires an additional wavelength on link $\langle n, z \rangle$ because $r_1$ has left the network. So, cost of link $\langle c, d \rangle$ during $\Delta \tau_2$ will be $\epsilon \Delta \tau_2 C(e)$, while the cost of link $\langle g, h \rangle$ during $\Delta \tau_2$ will be $\Delta \tau_2 C(e)$ (not-shareable state).
- $\Delta \tau_3$ (from time 30 to 40): connection $r_3$ requires an additional wavelength on both link $\langle n, z \rangle$ and link $\langle m, x \rangle$, because, after $r_2$ has also left the network, there is no backup capacity to be shared on link $\langle m, x \rangle$. So, cost of link $\langle c, d \rangle$ and link $\langle g, h \rangle$ during $\Delta \tau_3$ will be $\Delta \tau_3 C(e)$.

For a generic link $e$

$$C_e = \Delta \tau_1 C_g(e, \Delta \tau_1) + \Delta \tau_2 C_f(e, \Delta \tau_2) + \Delta \tau_3 C_f(e, \Delta \tau_3)$$

Link $(m-x)$

$C_{m-x} = 10\epsilon + 10\epsilon + 10 = 10 + 20\epsilon$

Link $(n-z)$

$C_{n-z} = 10\epsilon + 10 + 10 = 20 + 10\epsilon$

Fig. 3. Link-cost assignment for the network and traffic pattern in Fig. 2(a).

According to the new cost assignment, the algorithm will now choose the best route (e.g., $bp2$), thereby overcoming the limitation of the standard approach (GSP), which is holding-time unaware.

C. Complexity Analysis

The computational complexity of GSP is $O(K \cdot (|V|^3 + |E|))$. So, PHOTOS complexity is $O(K \times (R \times H + |V|^3 + |E|))$, where $R$ is the number of connections offered in a time unit and $H$ is holding time of the connection. Note that the algorithm is linear with respect to $R$ and $H$. For example, our heuristics in the worst case ($R \times H = 200$) only took (on a PC with a 2.4-GHz CPU and 512-MB RAM) an average processing time per connection of 1.56 milliseconds.

V. ILLUSTRATIVE NUMERICAL RESULTS

We now quantitatively evaluate the performance of our proposed PHOTO-GSP algorithm. 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. Requests are uniformly distributed among all node pairs; average connection-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. 4.

We compare our algorithm to GSP [3]. We employ two metrics to highlight the performance achievable by PHOTO-GSP: total channel consumption and resource overbuild.

A. Total Channel Consumption

Total channel consumption (TCh) is the overall number of channels needed to support all the offered traffic multiplied by the time interval these channels are actually used (note that the average holding time of a connection is normalized to unity). In Fig. 5, we show TCh for a simulation experiment with $10^6$ offered connections by PHOTO-GSP and GSP and $K=1,2,3$. PHOTO always requires fewer channels. This number tends to decrease for increasing load, because for high loads there is a higher probability to share. Note that the increase of $K$ from 1 to 2 leads to a remarkable savings in network resources, while increasing $K$ from 2 to 3 yields only a slight improvement.

B. Resource Overbuild

A figure of merit for comparing backup resource efficiency is resource overbuild (RO) [9], which indicates the amount of
extra resources needed for providing protection as the percentage of the amount of resources required without protection. Figure 6(a) shows that our heuristic has lower RO over GSP. RO assumes smaller values for increasing values of offered traffic due a wider availability of shareable capacity. The percentage difference between the two approaches is shown in Fig. 6(b): the savings in RO are remarkable especially for higher values of K and light offered traffic. We obtained savings ranging from 2% to almost 7%. Note that K affects positively the percentage difference in RO also because, when more than one path pair is evaluated, the average length of the working paths tends to increase.

PHOTO-GSP’s gain over GSP tends to decrease for high values of the arrival rate. This behavior is due to the same nature of the algorithm: if the arrival rate is high, the quality of the estimation used by the algorithm decreases which, in turn, reduces the value of PHOTO-GSP over GSP.

VI. CONCLUSION AND FUTURE WORK

We introduced a holding-time-aware, dynamic, connection-provisioning algorithm to improve sharing of backup resources in segmented protection. We observed significant savings in protection-resource usage by employing our new approach, called PHOTO-GSP, as opposed to another efficient approach that is holding-time-unaware, namely, GSP. The improvement in resource overbuild is found to be up to 7% for a US nationwide network with typical parameters. The proposed method is applicable to other contexts as well, such as MPLS networks, for bandwidth-guaranteed connections.

We are extending PHOTO to a more-efficient shared-segment-protection algorithm [6], called AGBSP: preliminary results show that a larger gain can be achieved.

REFERENCES