Dynamic routing for shared path protection in multidomain optical mesh networks

D. L. Truong

Department of Computer Science and Operational Research, Université de Montréal, C.P. 6128, Succ. Centre-Ville Montréal (QC) H3C 3J7, Canada

truongtd@iro.umontreal.ca

B. Thiongane

Centre de Recherche, Institute des Science de l'Ingénieur, Dakar, Sénégal

babacar.thiongane@isci.sn

RECEIVED 14 JULY 2005;
ACCEPTED 26 OCTOBER 2005; PUBLISHED 04 JANUARY 2006

The routing problem for shared path protection in multidomain optical mesh networks is more difficult than that in single-domain mesh networks due to the lack of complete and global knowledge of the network topology and bandwidth allocation. To overcome this difficulty, we propose an aggregated network modeling by underestimation with a two-step routing strategy. In the first step, a rough routing solution is sketched in a virtual network that is the topology aggregation of the multidomain network. A complete routing is then determined by solving routing problems within the original single-domain networks. The first step can be solved by either using an exact mathematical program or a heuristic, whereas the second step is always solved by heuristics. Computational results show the relevance of the aggregated network modeling. They also prove the scalability of the proposed routing for multidomain networks and its efficiency in comparison with the optimal solution obtained by use of the complete information scenario. In addition, we believe that short working paths lead to a higher possibility of sharing backup resources between backup paths. Our mathematical program model minimizes the total requested resources and at the same time provides a short working path, resulting in a further overall saving of resources.

OCIS codes: 060.4250, 060.0060, 060.4510.

1. Introduction

It has been recognized that shared path protection (SPP) both protects against link and node failures and saves resources thanks to bandwidth sharing among backup light paths (see Ref. [1]). In the single-failure scenario, two backup light paths can share bandwidth between them if their working light paths are link or node disjoint, later called the sharing condition. SPP routing consists of finding a pair of working and backup light paths that satisfy the sharing condition and optimize a particular criterion, such as requested bandwidth capacity, number of wavelength conversions, fiber link length, etc. This paper considers the problem of dynamic routing for SPP in multidomain optical mesh networks while minimizing the total bandwidth required by the working and backup light paths. Since the node-disjoint condition can be made equivalent to the link-disjoint condition by splitting each node into two halves with a virtual directed link between them (see Ref. [2]), the focus will be on the link-disjoint condition. We assume that links are not bundled together and thus a failure affects at most one link (which is not the case in Ref. [3]). We assume also that

every network node has optical–electrical–optical (OEO) treatment so that subwavelength switching and wavelength assignment are easy to handle.

There are some static (or off-line) SPP routing approaches proposed for single domain [4] or multidomain [5] networks. Given a network with known topology, link capacities, and future requested traffic, these approaches define fixed working and backup capacities for each link. Since network traffic changes unpredictably and frequently, a dynamic (on-line) routing without *a priori* knowledge of the network traffic is necessary.

Dynamic SPP routing identifies a pair of disjoint working and backup paths that minimally consume bandwidth according to the current network state while satisfying the sharing condition. This problem was proved to be NP hard in Ref. [6]. An exact ILP-based solution called sharing with complete information (SCI) was proposed in Ref. [7], in which the total bandwidth consumed by the working and backup paths is minimized. The ILP formulation requires detailed and global information on the entire network topology and the bandwidth allocation history for each network link. The two-step approach (TSA) [8] minimizes the working and backup bandwidths separately but computes them in the same way as SCI, leading to the same information requirement as SCI. To reduce the per-link information, sharing with partial information (SPI) [9] and distributed partial information management (DPIM) [2] were introduced. They overestimate the working and backup bandwidth consumption in comparison with SCI and apply the same ILP to minimize the total overestimated bandwidth. Later, active path first-backup path cost (APF-BPC), a heuristic-based routing using partial information scenarios of DPIM and SPI, was proposed in Ref. [10]. In all cases, the global knowledge (either partial or complete) of each link and the complete network topology are mandatory at the network ingress nodes.

In multidomain networks, it is impractical to make this global information available at a node. A multidomain network is an interconnection of several independent single-domain networks [11] [Fig. 1(a)]. To support the scalability, the routing information should not be excessively and frequently exchanged throughout the multidomain network [12]. The detailed connectivity and bandwidth allocation of a domain is limited within itself, and only aggregated information can be exposed to external domains. As a result, no node is aware of either the global multidomain network topology or the bandwidth allocation on all network links. We call this constraint the scalability constraint. It makes the above-listed solutions inapplicable to multidomain networks.

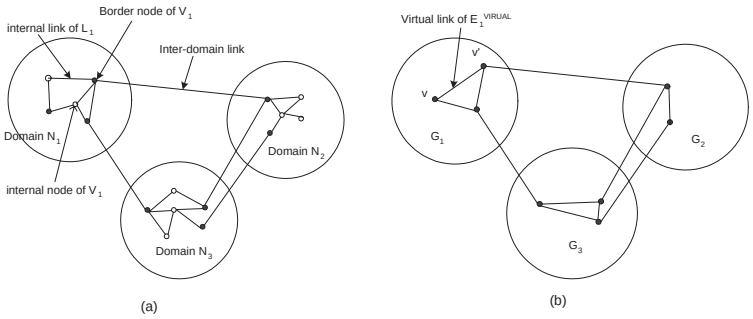

Fig. 1. (a) Multidomain network and (b) its interdomain network obtained by topology aggregation.

A few works have been proposed on dynamic protection for multidomain networks, but none have been devoted to SPP. No-sharing path protection was proposed in Ref. [13], and no-sharing segment protection was introduced in Ref. [14]. The latter was improved in Ref. [15] to become segment-shared protection although no details on its routing model were described. In Ref. [16], a routing for segment shared protection was proposed in which a

light path is not allowed to pass through any domain. In a real multidomain network, light paths often pass through many domains. This is illustrated in Fig. 1(a) in which a light path from domain $\mathcal{N}_1$ to $\mathcal{N}_3$ can pass through $\mathcal{N}_2$.

This paper deals with SPP routing in multidomain networks without global information knowledge. Our main idea is to transform the original multidomain routing problem into several single-domain routing problems that are solved separately by using adapted versions of existing single-domain SPP routings on underestimated information. We propose a two-step routing strategy. First, the multidomain network is topologically aggregated to become a single-domain network called an interdomain network, in which a rough routing is sketched out. A detailed routing is then determined within each original single-domain network. The use of aggregate information at the first step removes the global information requirement and thus preserves scalability. We propose two approaches to realize the routing strategy. The two approaches are compared through computational results. To evaluate the relevance of the aggregate information, the approaches are compared to SCI when the latter is executed on multidomain networks. Also note that, although DPIM and SPI try to reduce the amount of required per-link information, we concentrate on reducing the details of the information to be advertised from a domain and the frequency of information exchanged between domains as well as within domains. The final objective is to respect the scalability constraint.

This paper is organized as follows: Section 2 introduces the notation and the two-step routing strategy. In Section 3, the cost functions are defined using aggregate information. The two approaches to realizing the two-step routing strategy are presented in Section 4. Section 5 presents the routing signaling that coordinates the two routing steps and the routing information update. Section 6 shows the computational results on a multidomain network built from real single-domain networks. Finally, Section 7 concludes the paper.

2. Notation and Two-Step Routing Strategy

The multidomain network is represented by a graph $\mathcal{N} = (V, L)$ composed of M connected single-domain networks $\mathcal{N}_i = (V_i, L_i)$, $i = 1, .., M$. The sets V (V_i) and L (L_i) are respectively the set of nodes and the set of links of $\mathcal{N}$ ($\mathcal{N}_i$). Each single-domain set of nodes V_i decomposes into the border nodes V_i^{BORDER} and the core nodes V_i^{CORE}. Moreover, note that L decomposes into L^{INTRA} and L^{INTER}. $L^{\text{INTER}} = \left\{(v, v') \in L : v \in V_i^{\text{BORDER}}, v' \in V_j^{\text{BORDER}} \neq v_i^{\text{BORDER}}\right\}$ is the set of interdomain links where an interdomain link connects two border nodes of two different domains. On the other hand, $L^{\text{INTRA}} = \cup_{i=1..M} (L_i)$ is the set of links within domains.

A clique mesh topology aggregation will be applied to $\mathcal{N}_i$, $i = 1, .., M$, to obtain an aggregated graph $G_i = \left(V_i^{\text{BORDER}}, E_i^{\text{VIRTUAL}}\right)$ containing only border nodes of $\mathcal{N}_i$ and the set of virtual links connecting all pairs of border nodes $E_i^{\text{VIRTUAL}} = \left\{(v, v') : v, v' \in V_i^{\text{BORDER}}\right\}$. The resulting network $G = \left(V^{\text{BORDER}}, E\right)$ is a compact interdomain network [see Fig. 1(b)] where V^{BORDER} contains all the border nodes of $\mathcal{N}$ and E contains all the virtual links E^{VIRTUAL} and interdomain links L^{INTER}:

$$V^{\text{BORDER}} = \underset{i=1..M}{\cup} V_i^{\text{BORDER}},$$

$$E^{\text{VIRTUAL}} = \underset{i=1..M}{\cup} E_i^{\text{VIRTUAL}},$$

$$E = E^{\text{VIRTUAL}} \cup L^{\text{INTER}}.$$

We denote by e an edge of G and ℓ a fiber link of $\mathcal{N}$. Thus an interdomain link can be denoted by e or ℓ.

When e is a virtual link between v and $v' \in \mathcal{N}_i$, we define $\mathcal{P}_e$ as the set of physical paths within $\mathcal{N}_i$ between v and v', and $\mathcal{P}_e = \{e\}$ when e is an interdomain link. An element of $\mathcal{P}_e$ is an instance of e. A link e will be associated with a link state representing some routing information obtained from all the elements of $\mathcal{P}_e$. This link state will be disseminated to all multidomain network border nodes. Thus, these border nodes have a common aggregated view of the multidomain network. More details are given in Sections 3 and 5.

Let us consider a new request of bandwidth d from a source border node v_s to a destination border node v_d. The requested bandwidth will be routed over a single path. The following notation is introduced where Roman letters are for the original network $\mathcal{N}$ and Greek letters are for the aggregated network G:

p and p' are respectively the complete working and backup paths in $\mathcal{N}$ to be found for the new request.

c_ℓ^{res} is the residual bandwidth capacity on physical link $\ell \in L$.

a_ℓ is the bandwidth that will be consumed by physical link $\ell \in L$ of the working path p. Evidently, $a_\ell = d$ if there is sufficient residual capacity on ℓ.

$B_{\ell'}$ is the bandwidth reserved on physical link $\ell' \in L$ by existing backup paths.

$B_{\ell'}^{\ell}$ is the bandwidth reserved on physical link $\ell' \in L$ by existing backup paths that protect the working paths passing through link $\ell \in L$. This bandwidth is not sharable for protecting any new working path containing ℓ.

$B_{\max}^{\ell}$ is the maximal backup bandwidth reserved on a network link for protecting the working paths that pass through link $\ell \in L$. Indeed, $B_{\max}^{\ell} = \max_{\ell' \in L} B_{\ell'}^{\ell}$.

$B_{\max}^{q}$ and $B_{\max}^{p}$ are also defined as $B_{\max}^{q} = \max_{\ell \in q} B_{\max}^{\ell}$ and $B_{\max}^{p} = \max_{\ell \in p} B_{\max}^{\ell}$, respectively.

$b_{\ell'}^{\ell}$, $b_{\ell'}^{q}$, and $b_{\ell'}^{p}$ are respectively the additional backup bandwidths to be reserved besides $B_{\ell'}$ on physical link ℓ' to protect the new working path p against single failures on link ℓ, subpath q, and the entire p. Observe that $b_{\ell'}^{q} = \max_{\ell \in q} b_{\ell'}^{\ell}$ and $b_{\ell'}^{p} = \max_{\ell \in p} b_{\ell'}^{\ell}$.

$b_{q'}^{\ell}$, $b_{q'}^{q}$, and $b_{q'}^{p}$ are the overall additional backup bandwidths to be reserved along subpath q' to protect the new working path p against single failures on link ℓ, subpath q, and the entire p. Hence, $b_{q'}^{\ell} = \sum_{\ell' \in q'} b_{\ell'}^{\ell}$, $b_{q'}^{q} = \sum_{\ell' \in q'} b_{\ell'}^{q}$, and $b_{q'}^{p} = \sum_{\ell' \in q'} b_{\ell'}^{p}$.

π and π' are the representations of p and p', repsectively, in G. They are composed of virtual and interdomain links. They are called the directive working and backup paths.

$\mathcal{P}_\pi$ ($\mathcal{P}_{\pi'}$) is the set of physical paths obtained by substituting all virtual links of π (π') by their instances. Clearly, $p \in \mathcal{P}_\pi$ and $p' \in \mathcal{P}_{\pi'}$.

α_e is the total bandwidth that p will consume along its subpath represented by virtual or interdomain link $e \in E$. Thus, $\alpha_e = \sum_{\ell \in q} a_\ell$, where q is the subpath.

$\beta_{e'}^{e}$ ($\beta_{e'}^{\pi}$) is the overall additional backup bandwidth to be reserved along the subpath represented by link $e' \in E$ to protect p against single failures on its subpath represented by $e \in E$ (on the entire p). Thus, $\beta_{e'}^{e} = b_{q'}^{q}$ and $\beta_{e'}^{\pi} = b_{q'}^{p}$, where q and q' are the subpaths in $\mathcal{N}$ represented by e and e', respectively.

γ_e^{res} is the maximum bandwidth that can be routed over an instance of $e \in E$. $\gamma_e^{\text{res}} = \max_{q \in \mathcal{P}_e} \min_{\ell \in q} c_\ell^{\text{res}}$ is the residual capacity on e.

$\| e \|$ is the length of the shortest instance of $e \in E$, and $\| e \| = \min_{q \in \mathcal{P}_e} \| q \|$, where $\| q \|$ is the length of q in number of hops.

The parameters a, α and b, β with different indexes are also called working and backup costs.

Dynamic SPP routing aims to identify, for a request, a working path p and a backup path p' that are disjoint and minimize the total consumed bandwidth:

$$\min \sum_{\ell \in p} a_\ell + \sum_{\ell' \in p'} b_{\ell'}^{p}. \tag{1}$$

According to the definition of α_e and $\beta^{\pi}_{e'}$, expression (1) is equivalent to

$$\min \sum_{e\in\pi} \alpha_e + \sum_{e'\in\pi'} \beta^{\pi}_{e'}. \tag{2}$$

We propose the following two-step routing strategy:

- Interdomain routing step: This step is performed on the interdomain network. The source border node computes π and π' in G while minimizing their bandwidth consumption

$$\min \sum_{e\in\pi} \alpha_e + \sum_{e'\in\pi'} \beta^{\pi}_{e'}. \tag{3}$$

- Intradomain routing step: At this step, the virtual links of π and π' are replaced by physical paths to build the complete working and backup paths. Virtual link e is mapped with (replaced by) one of its instances in $\mathscr{P}_e$. A joint mapping of all virtual links would help to maintain the optimal bandwidth cost obtained at the interdomain step but involves many domains simultaneously and thus requires global information. Therefore, we first map the virtual links of π and then those of π'. The path instance $q \in \mathscr{P}_e$ that is mapped with the working virtual link $e \in \pi$ should minimize α_e:

$$\min_{q\in\mathscr{P}_e} \sum_{\ell\in q} a_\ell. \tag{4}$$

  The path instance $q' \in \mathscr{P}_{e'}$ that is mapped with the backup virtual link $e' \in \pi'$ should minimize $\beta^{\pi}_{e'}$, i.e.,

$$\min_{q'\in\mathscr{P}_{e'}} b^{p}_{q'} = \min_{q'\in\mathscr{P}_{e'}} \sum_{\ell'\in q'} b^{p}_{\ell'}. \tag{5}$$

  Note that the mapping of a virtual link of E_i^{VIRTUAL} involves only its instances in $\mathscr{N}_i$, which could be performed within the single-domain network $\mathscr{N}_i$ by one border node of the virtual link.

The parameters α_e, $\beta^{\pi}_{e'}$, a_ℓ, and $b^{p}_{\ell'}$ remain to be identified and the minimization problems (3), (4), (5) must be solved. Section 3 shows how the parameters are identified. Section 4 presents the algorithms for solving the minimization problems.

3. Working and Backup Costs

Until the complete working and backup paths are identified, the costs α_e and $\beta^{\pi}_{e'}$, which are used in interdomain routing, cannot be computed exactly but only estimated. To satisfy the scalability constraint, the estimation should not use the complete and detailed information on each network link. The same is true of the computation of a_ℓ and $b^{p}_{\ell'}$, which are used by the intradomain routing.

3.A. Underestimation of Working and Backup Costs for Interdomain Routing

The ultimate goal of the estimations is to relax the dependency of the exact values of α_e and $\beta^{\pi}_{e'}$ on global and detailed information about physical links inside domains. These values will be represented as functions of some domain aggregated information that will become link states of virtual or interdomain links.

We underestimate the working cost of link $e \in E$ as the minimal overall bandwidth that p should consume along e:

$$\alpha_e \simeq \min_{q\in\mathscr{P}_e} \sum_{\ell\in q} a_\ell.$$

Thus

$$\alpha_e \simeq \begin{cases} \| e \| d & \text{if } d \le \gamma_e^{\text{res}},\ e \in E^{\text{VIRTUAL}} \\ d & \text{if } d \le \gamma_e^{\text{res}},\ e \in L^{\text{INTER}} \\ \infty & \text{otherwise} \end{cases} . \tag{6}$$

The estimation of $\beta_{e'}^{\pi}$ is more complicated; let us begin with $b_{\ell'}^{\ell}$. Note that $b_{\ell'}^{\ell}$, the additional bandwidth to be reserved, is the difference between the required bandwidth and the sharable backup bandwidth on ℓ'. The sharable backup bandwidth on link ℓ' for protecting link ℓ is $B_{\ell'} - B_{\ell'}^{\ell}$ (see Ref. [7] for details). Because $b_{\ell'}^{\ell}$ must be nonnegative, we have

$$b_{\ell'}^{\ell} = \max\left\{0, B_{\ell'}^{\ell} + d - B_{\ell'}\right\}. \tag{7}$$

Here, detailed information on $B_{\ell'}^{\ell}$ is still required (as in SCI). To avoid this, $B_{\ell'}^{\ell}$ is overestimated as in Ref. [2] by $B_{\max}^{\ell}$. Note that $b_{\ell'}^{\ell}$ cannot be greater than the requested bandwidth:

$$b_{\ell'}^{\ell} \simeq \min\left\{\max\left\{0, B_{\max}^{\ell} + d - B_{\ell'}\right\}, d\right\}. \tag{8}$$

From this estimation, it can be proved that the backup cost of a virtual or interdomain link to protect a working path is not smaller than the cost of protecting any virtual or interdomain link of the path (see Appendix A):

$$\beta_{e'}^{\pi} \simeq \max_{e \in \pi} \beta_{e'}^{e}. \tag{9}$$

Now what we need to compute is $\beta_{e'}^{e}$. We underestimate $\beta_{e'}^{e}$ by the minimum backup bandwidth that p' should reserve along e':

$$\beta_{e'}^{e} \simeq \min_{q \in \mathscr{P}_e, q' \in \mathscr{P}_{e'}} b_{q'}^{q}. \tag{10}$$

The computational effort for $\beta_{e'}^{e}$ when e is a virtual link may be increasing while its value might have no effect on the maximum of expression (9) if it is not the greatest element of the maximum. Therefore, we ignore it by defining $\beta_{e'}^{e} = 0$ for all $e' \in E$ and $e \in E^{\text{VIRTUAL}}$. All that remains is to estimate the two following cases of $\beta_{e'}^{e}$: $\beta_{e'}^{\ell}$, $\ell \in L^{\text{INTER}}$, $e' \in E^{\text{VIRTUAL}}$ and $\beta_{\ell'}^{\ell}$, ℓ, $\ell' \in L^{\text{INTER}}$.

In the first case, according to expression (10), $\beta_{e'}^{\ell} \simeq \min_{q' \in \mathscr{P}_{e'}} b_{q'}^{\ell}$. Suppose that $e' \in \mathscr{N}_i$ and let $\overline{B}$ be the maximum backup bandwidth reserved on a link of the domain $\mathscr{N}_i$,

$$\overline{B} = \max_{\ell' \in L_i} B_{\ell'}. \tag{11}$$

Then, combining this with the definition of $b_{q'}^{\ell}$ and expression (8) we have

$$b_{q'}^{\ell} \ge \| q' \| \min\left\{\max\left\{0, B_{\max}^{\ell} + d - \overline{B}\right\}, d\right\},$$

$$\beta_{e'}^{\ell} \ge \min_{q' \in \mathscr{P}_{e'}} \| q' \| \min\left\{\max\left\{0, B_{\max}^{\ell} + d - \overline{B}\right\}, d\right\}.$$

Thus, $\beta_{e'}^{\ell}$ can be underestimated by

$$\beta_{e'}^{\ell} \simeq \| e' \| \min\left\{\max\left\{0, B_{\max}^{\ell} + d - \overline{B}\right\}, d\right\}. \tag{12}$$

Combining this with the capacity constraint, $\beta^{\ell}_{e'}$ for $\ell \in L^{\text{INTER}}$, $e' \in E^{\text{VIRTUAL}}$ is defined as

$$\beta^{\ell}_{e'} \simeq \begin{cases} 0 & \text{if } B^{\ell}_{\max} + d \le \overline{B} \\ \| e' \| \left(B^{\ell}_{\max} + d - \overline{B} \right) & \text{if } B^{\ell}_{\max} + d > \overline{B} > B^{\ell}_{\max},\ \gamma^{\text{res}}_{e'} \ge B^{\ell}_{\max} + d - \overline{B} \\ \| e' \| d & \text{if } B^{\ell}_{\max} \ge \overline{B},\ \gamma^{\text{res}}_{e'} \ge d \\ \infty & \text{otherwise} \end{cases} . \quad (13)$$

In the second case, $\beta^{\ell}_{\ell'} = b^{\ell}_{\ell'}$ for $\ell, \ell' \in L^{\text{INTER}}$, and it is defined by expression (8) as

$$\beta^{\ell}_{\ell'} \simeq \begin{cases} 0 & \text{if } B^{\ell}_{\max} + d \le B_{\ell'},\ \ell \ne \ell' \\ B^{\ell}_{\max} + d - B_{\ell'} & \text{if } B^{\ell}_{\max} + d > B_{\ell'} > B^{\ell}_{\max},\ c^{\text{res}}_{\ell'} \ge B^{\ell}_{\max} + d - B_{\ell'},\ \ell \ne \ell' \\ d & \text{if } B^{\ell}_{\max} \ge B_{\ell'},\ c^{\text{res}}_{\ell'} \ge d,\ \ell \ne \ell' \\ \infty & \text{otherwise} \end{cases} . \quad (14)$$

Note that instead of $\overline{B}$, other estimations can be used. For example, a less-coarse estimation can be obtained by using $\overline{B}_{e'} = \max_{q' \in \mathscr{P}_{e'}} \max_{\ell' \in q'} B_{\ell'}$. It is also possible to consider $\bar{B}_{e'}$, the greatest among all the medians of $\{B_{\ell'}, \ell' \in q'\}$, $\forall q' \in \mathscr{P}_{e'}$. At that time $\beta^{e}_{e'}$ would be estimated by $\frac{1}{2} \| e' \| \min\left\{ \max\left\{0, B^{\ell}_{\max} + d - \bar{B}_{e'}\right\}, d \right\}$. In both cases, the computation effort increases while the scalability decreases.

In summary, the working and backup costs of a virtual or interdomain link in G are estimated by using only per-virtual/interdomain-link values (instead of per-link values) such as $\| e \|$, γ^{res}_{e} (or c^{res}_{ℓ}), $\overline{B}$ (or B_{ℓ}), and $B^{\ell}_{\max}$. They are defined as link-state attributes of virtual or interdomain links.

3.B. *Computation of Working and Backup Costs for Intradomain Routing*

The working cost a_{ℓ} of the link ℓ is defined as

$$a_{\ell} = \begin{cases} d & \text{if } d \le c^{\text{res}}_{\ell} \\ \infty & \text{otherwise} \end{cases} . \quad (15)$$

From expression (8) and the definition of $B^{p}_{\max}$, it is easy to deduce that $b^{p}_{\ell'} = \min\left\{ \max\left\{0, B^{p}_{\max} + d - B_{\ell'}\right\}, d \right\}$; i.e.,

$$b^{p}_{\ell'} \simeq \begin{cases} 0 & \text{if } B^{p}_{\max} + d \le B_{\ell'} \le 0 \\ B^{p}_{\max} + d - B_{\ell'} & \text{if } B^{p}_{\max} + d > B_{\ell'} > B^{p}_{\max},\ c^{\text{res}}_{\ell'} \ge B^{p}_{\max} + d - B_{\ell'} \\ d & \text{if } B^{p}_{\max} \ge B_{\ell'},\ c^{\text{res}}_{\ell'} \ge d \\ \infty & \text{otherwise} \end{cases} . \quad (16)$$

Hence, the intradomain routing requires b_{ℓ} and c^{res}_{ℓ} of every link ℓ in the domain and $B^{\ell}_{\max}$ of every link ℓ of p for computing $B^{p}_{\max}$.

4. Routing Approaches

We propose two approaches to solve the minimization problems (3), (4), (5). The intradomain step is identical but the interdomain step is different in the two approaches. The approaches are named according to their interdomain routing.

4.A. *Working Path First*

The working path is routed first. All shortest-path problems are in terms of cost.

- Interdomain routing step: Instead of minimizing expression (3), we separately minimize each term of the sum. First, the directive working path is set to the shortest path

in G between the source and the destination when the working cost α_e is assigned to each link of G. Subsequently, the backup cost $\beta_{e'}^{\pi}$ is assigned to each link of G. The directive backup path is then set to the shortest path in G between the source and the destination. Note that even when π and π' share a virtual link, their complete paths could still be link disjoint. However, π and π' must be interdomain link disjoint. This constraint is taken into account in the definition of $\beta_{\ell'}^{\ell}$.

- Intradomain routing step: First, virtual links of π are mapped one by one within their domains. For mapping the virtual link $e \in E_i^{\text{VIRTUAL}}$ between v and $v' \in \pi$, we search for the shortest path between v and v' in domain $\mathcal{N}_i$ when physical links of $\mathcal{N}_i$ are weighted by a_ℓ. Once the complete working path p is then obtained, the virtual links of π' are mapped similarly but with the backup cost $b_{\ell'}^{p}$. Again, disjointedness is taken into account through the definition of $b_{\ell'}^{p} = \infty$ for each physical link ℓ' in the working path.

The shortest-path problems are solved using Dijkstra's algorithm (see, e.g., Ref. [18]). The request is rejected if one step fails to find paths.

Note that the intradomain routing of the working path is independent of the interdomain routing of the backup path. An alternative procedure would be to route completely the working path first, then route the directive backup path and finally map the backup virtual links. In that case, $\beta_{e'}^{e} \simeq \| e' \| \min\left\{\max\left\{0, B_{\max}^{q} + d - \overline{B}\right\}, d\right\}$ for $e \in E^{\text{VIRTUAL}}$ may be obtained in a way similar to how we obtained $\beta_{e'}^{\ell}$. This routing is called complete working path first (CWPF) and will not be further developed because its experimental results are similar to those of working path first (WPF).

4.B. *Joint Computing of Directive Paths*

In this approach, the directive working and backup paths are jointly computed by mathematical programming. Here we consider each link of E as two directed arcs. However, we still keep the notations e and E, but the former represents an arc, whereas the latter denotes the set of arcs. Given $v_i \in V^{\text{BORDER}}$, $\Gamma^{+}(v_i)$ $[\Gamma^{-}(v_i)]$ denotes the set of outgoing (incoming) arcs at node v_i. We introduce the following notation: $x_e = 1$ if the directive working path π from v_s to v_d goes through arc e, 0 otherwise, and $y_e = 1$ if the directive backup path π' from v_s to v_d goes through arc e, 0 otherwise. Joint computing of direct paths (JDP) follows the procedure below:

- Interdomain routing step: We solve an ILP problem (P) defined in the interdomain network G to find π and π' for each light-path request.

- Intradomain routing step: Similar to the intradomain routing of WPF.

The light-path request is rejected if a solution is not found at one of two steps. The ILP formulation (P) for the interdomain routing step is similar to the one proposed in Refs. [7] and [9]:

$$\min \sum_{e \in E} \alpha_e x_e + \sum_{e' \in E} z_{e'} + \nu \sum_{e \in E} x_e + \mu \sum_{e' \in E} y_{e'}$$

subject to

$$\sum_{e \in \Gamma^{+}(v_i)} x_e - \sum_{e \in \Gamma^{-}(v_i)} x_e = \begin{cases} 1 & v_i = v_s \\ 0 & v_i \neq v_s,\ v_d \\ -1 & v_i = v_d \end{cases}, \tag{17}$$

$$\sum_{e' \in \Gamma^{+}(v_i)} y_{e'} - \sum_{e' \in \Gamma^{-}(v_i)} y_{e'} = \begin{cases} 1 & v_i = v_s \\ 0 & v_i \neq v_s,\ v_d \\ -1 & v_i = v_d \end{cases}, \tag{18}$$

$$z_{e'} \geq \beta^{e}_{e'}\,(x_e + y_{e'} - 1), \quad e, e' \in E, \tag{19}$$

$$z_e \geq 0, \quad e \in E, \tag{20}$$

$$x_e, y_{e'} \in \{0,1\}, \quad e, e' \in E. \tag{21}$$

The first two terms of the objective function are respectively the cost of the working and backup paths. The cost of the complete paths may be far from that of the directive paths when the number of virtual links increases. Therefore, the last two terms are added to favor short directive paths among those with the same total path cost and thus to limit the number of virtual links. When costs α and β are integers and ν and μ are sufficiently small so that $\nu\sum_{e\in E} x_e + \mu\sum_{e'\in E} y_{e'} < 1$, it can be easily seen that the solution of (P) is the directive working and backup path pair with the smallest total weighted lengths among those minimizing the total consumed bandwidth. In Section 6 we study the effect of the working and backup path lengths on the cost and the blocking rate.

The two sets of constraints (17) and (18) are flow conservation constraints for the working path and the backup path, respectively. Each set represents a path from the source border node v_s to the destination border node v_d in G. The parameter z_e is in fact the backup cost β^{π}_{e} and is modeled through constraint (19).

The links with insufficient residual capacities are automatically excluded from the working and backup paths because their α_e and $\beta^{e}_{e'}$ are infinity. Once again, the disjoint constraint is taken account of by the definition of $\beta^{\ell}_{\ell'}$ as in WPF.

Besides, because a solution of (P) is defined in the directed graph, $\beta^{e}_{e'}$ is also updated according to the opposite direction of the arcs e and e'.

5. Routing Signaling and Routing Information Update

5.A. *Routing signaling*

The directive working and backup paths are both computed by the source border node. Once finished, the source node asks the border nodes along the working path to map the working virtual links with physical paths. The working segments q and their corresponding $B^{q}_{\max}$ that are found are returned to the source node. Finally the source identifies $B^{p}_{\max}$ as the maximum of all $B^{q}_{\max}$ and sends it to the border nodes along the directive backup path. These nodes use $B^{p}_{\max}$ to perform the intradomain routing for mapping their backup virtual links with physical paths.

5.B. *Routing Information Distribution*

Once the routing is completed, the paths are set up and the link states of all the physical as well as virtual or interdomain links are updated. It is worth noting that these link states are stored in a distributed way at different border nodes. A border node also keeps the link state $\left\{c^{\text{res}}_{\ell}, B_{\ell}, B^{\ell}_{\max}\right\}$ of each internal link ℓ of its domain and the link state $\left\{\| e \|, \gamma^{\text{res}}_{e}, \overline{B}\right\}$ for all adjacent virtual links e. In addition, each internal or border node keeps the set $\mathbb{B}_{\ell'} = \left\{B^{\ell}_{\ell'} : \ell \in L\right\}$ for each link ℓ' and $\mathbb{B}^{\ell} = \left\{B^{\ell}_{\ell'} : \ell' \in L\right\}$ for each link ℓ adjacent to it. The former set is necessary to compute the exact backup bandwidth to be reserved by using expression (7) if the backup path goes through ℓ'. The latter allows the computation of $B^{\ell}_{\max}$ if the working path goes through ℓ.

5.C. Routing Information Update through the Path Setup Process

The working path will be set up first, then the backup path. To set up the working path, a signaling message propagates along the working path from the source to the destination carrying the complete working and backup paths. Each node along the working path subsequently makes a cross connection and updates the set $\mathbb{B}^{\ell}$ and the link state $\left\{c_{\ell}^{\text{res}}, B_{\ell}, B_{\max}^{\ell}\right\}$, where ℓ is an adjacent working link. The new link states are collected with the signaling message until the domain's egress border node. Here these link states are forwarded to other domain border nodes to synchronize them. The number of update messages is $O\left(\mid V_i^{\text{BORDER}} \mid\right)$, where $\mathcal{N}_i$ is the current domain. The process continues until the destination is reached.

For reserving the backup path, a similar process is performed from the destination back to the source. However, no cross connection is made. The backup bandwidth is just reserved by updating $\left\{c_{\ell'}^{\text{res}}, B_{\ell'}, \mathbb{B}_{\ell'}\right\}$ on each backup link ℓ'. The number of update messages is also $O\left(\mid V_i^{\text{BORDER}} \mid\right)$.

Finally, every border node locally updates the link states $\left\{\| e \|, \gamma_e^{\text{res}}, \overline{B}\right\}$ of their virtual or interdomain links and exchanges these link states with each other. The number of exchange messages is $O\left(\mid V^{\text{BORDER}} \mid^2\right)$.

It is important to emphasize that with the exception of the flow of signaling messages, the routing information update is only performed through communication between border nodes. The overall number of update messages required after a light-path request is $O\left(\mid V^{\text{BORDER}} \mid^2 + \sum_{j=1}^{K} \mid V_i^{\text{BORDER}} \mid^2\right) = O\left(\mid V^{\text{BORDER}} \mid^2\right)$, where K is the number of domains crossed either by the working or backup path. Indeed $\mid V^{\text{BORDER}} \mid = \sum_{j=1}^{M} \mid V_i^{\text{BORDER}} \mid$; thus $\mid V^{\text{BORDER}} \mid^2 > \sum_{i=1}^{K} \mid V_i^{\text{BORDER}} \mid^2$. The size of each message is always $O(1)$.

Clearly, $O\left(\mid V^{\text{BORDER}} \mid^2\right)$ is smaller than the number of update messages in the single-domain SPP approaches, which is $O\left(|V||V^{\text{BORDER}}|\right)$, since an all-to-border node update is required. This proves that our approach is more scalable than the single-domain approaches [7–10]. The scalability of our approach can be improved if link-state updates are performed only once after several requests. In Section 6 we will analyze the effect on routing quality.

6. Computational Results

In this section, we evaluate the relevance of the information aggregation scenario and the efficiency of WPF compared to JDP. For the relevance of the information aggregation scenario, we compare the results of the proposed two-step routing on a multidomain network with those of the complete information scenario SCI [9] on the same network.

The computational results are conducted on a five-domain network. The five domains are real optical networks: EONet [19], RedIRIS [20], GARR [21], Renater3 [22], and SURFnet [23] with real link capacities for the last four networks. Some interdomain links have been added with OC-192 capacities (see Fig. 2) for connecting different domains. Requests are randomly generated between border nodes and the requested bandwidth is uniformly distributed among OC-{1, 3, 6, 9, 12}.

JDP (ν, μ) will be used to denote the configurations of the JDP with fixed parameters ν, μ. Configurations with shorter directive working paths are expected also to give shorter complete working paths leading to more possibility of sharing backup bandwidth.

We tested WPF when $\overline{B}$, $\overline{B}_{e'}$, and $\bar{B}_{e'}$ are used. In all cases, the results are very close. We only present the results of WPF with the $\overline{B}$ estimation.

The commercial software CPLEX and the academic version of OPNET Modeler are respectively used to implement JDP and WPF on a 1.9 GHz Pentium 4. The computational

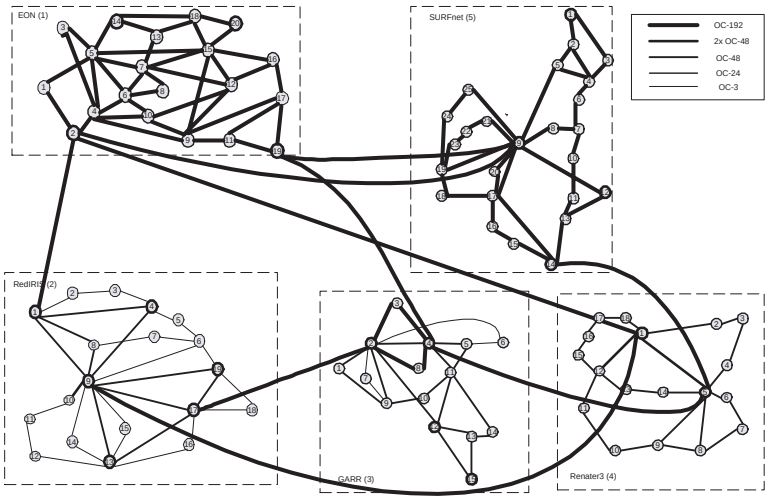

Fig. 2. Experimental network

time for routing a request is less than 16 ms for WPF and less than 1 min for JDP (ν, μ).

6.A. *Analysis of Bandwidth Costs*

To determine how WPF and JDP are far from the optimal solution in terms of bandwidth savings, we compared the total working and backup path costs found by WPF and JDP with SCI. Recall that SCI does not satisfy the scalability constraint. Let $\text{cost}^r_{\text{WPF}}$ ($\text{cost}^r_{\text{JDP}}$) be the total bandwidth cost of the complete working and backup paths in the case of WPF (JDP) and cost_{SCI} be the total cost of SCI. The relative gap between $\text{cost}^r_{\text{WPF}}$ and $\text{cost}^r_{\text{SCI}}$ is defined by

$$\text{gap}_{\text{WPF/SCI}} = \frac{\text{cost}^r_{\text{WPF}} - \text{cost}_{\text{SCI}}}{\text{cost}_{\text{SCI}}}$$

and similarly for $\text{gap}_{\text{JDP/SCI}}$. Figure 3(a) depicts the distribution of $\text{gap}_{\text{WPF/SCI}}$ and $\text{gap}_{\text{JDP/SCI}}$. In this figure, the column at abscissa 0.5, for example, represents the percentage of cases that the gap is in the range]0.25, 0.5]. Note that the gap is computed only for the requests that are successfully routed by SCI and either WPF or JDP. Figure 3(a) shows that the cost of SCI is generally smaller than that of WPF and JDP since the gap is positive most of the time. This is a natural observation since the routing in SCI is performed within a complete information scenario. Another observation is that the percentages of cases where the gap is within $]-0.5, 0.5]$ for JDP $(1/N, 1/N)$, JDP $(1/N, 1/2N)$, JDP $\left(1/N, 1/N^2\right)$, and WPF are respectively 62%, 65%, 70%, and 70%, where $N = \mid E \mid$. Thus, most of the time the real cost of the solution found by WPF and by JDP is not so far from the solution found by SCI.

The directive routing given by the interdomain routing step is accurate if the total estimated cost of the working and backup paths is closed to the total real cost obtained once the routing has been completed. Therefore, to evaluate the accuracy of the interdomain routing step of each scheme, the relative gap between the estimated and real costs is introduced. For WPF it is defined by

$$\frac{\text{cost}^e_{\text{WPF}} - \text{cost}^r_{\text{WPF}}}{\text{cost}^r_{\text{WPF}}}$$

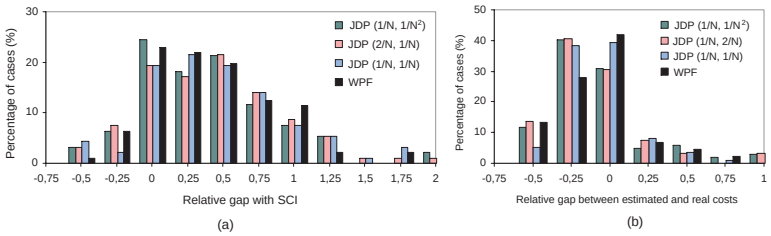

Fig. 3. Distribution of (a) the relative gap with SCI and (b) the relative gap between the estimated and the real costs for WPF and JDP.

and similarly for JDP. Figure 3(b) illustrates the distribution of the relative gap for each routing scheme. We can observe that the gap is within $]-0.5, 0.5]$ for 89%, 82%, 81%, and 81% of cases respectively for JDP $(1/N, 1/N)$, JDP $(1/N, 1/2N)$, JDP $\left(1/N, 1/N^2\right)$, and WPF. This means the estimations of WPF and JDP are mostly close to their real costs. Moreover, the advantage of shorter working paths is illustrated since JDP $(1/N, 1/N)$ gives better gaps than JDP $(1/N, 1/2N)$, which in turn gives slightly better gaps than JDP $\left(1/N, 1/N^2\right)$.

We compare JDP and WPF in frequency of finding smaller estimated and real costs. The comparisons are made with the three configurations of JDP. It should be noted that in this experiment α and β are integers and $\nu\sum_{e\in E} x_e + \mu\sum_{e\in E} y_e < 1$ for the three configurations of JDP. Therefore, the total bandwidth costs are minimized in these cases. In addition, when $(\nu, \mu) = (1/N, 1/N)$ the total length of the directive working and backup paths is minimized. When $(\nu, \mu) = (1/N, 1/2N)$, the directive working path π tends to be short. When $(\nu, \mu) = \left(1/N, 1/N^2\right)$, the shortest directive working path π and the shortest directive backup path π' among all candidates associated with π are obtained.

Figure 4(a) shows the percentage of cases for which JDP $(1/N, 1/N)$ or WPF finds better (smaller) total estimated costs when the number of sent requests increases. Figure 4(b) depicts the percentage of cases for which JDP $(1/N, 1/N)$ or WPF finds better total real costs when the number of sent requests increases. Figures 4(c) and 4(d) show the same results for JDP $(1/N, 1/2N)$, and Figs. 4(e) and 4(f) illustrate the results for JDP $\left(1/N, 1/N^2\right)$. Note that WPF is overall slightly better than JDP $(1/N, 1/N)$ in estimated and real costs but JDP is more improved when the length of the working path is more minimized. The best result is given with JDP $\left(1/N, 1/N^2\right)$. This confirms the expectation that when there are fewer virtual links the real cost is reduced. Furthermore, when the working path is short, there is more chance to share backup bandwidth with the future light-path requests because there is less chance of violating the sharing constraint due to link-joint working paths. The overall resource utilization will be improved. In fact, WPF follows this strategy since it always looks for the shortest working path first. This explains why WPF obtains a relatively good performance even if it does not jointly compute the working and backup paths.

6.B. Blocking Probability Analysis

When the request holding time is infinite, the scheme with better resource allocation rejects less bandwidth and begins to reject later than the others. That is why we chose the bandwidth-blocking probability as an index for evaluating the resource allocation capability. This probability is defined as the ratio between the amount of accepted bandwidth and the amount of requested bandwidth. Figure 5(a) shows the bandwidth-blocking probability at the interdomain step. We can see that the blocking of JDP is better than for WPF. The blocking of JDP $(1/N, 1/2N)$ [similar to that of JDP $\left(1/N, 1/N^2\right)$] is better than the

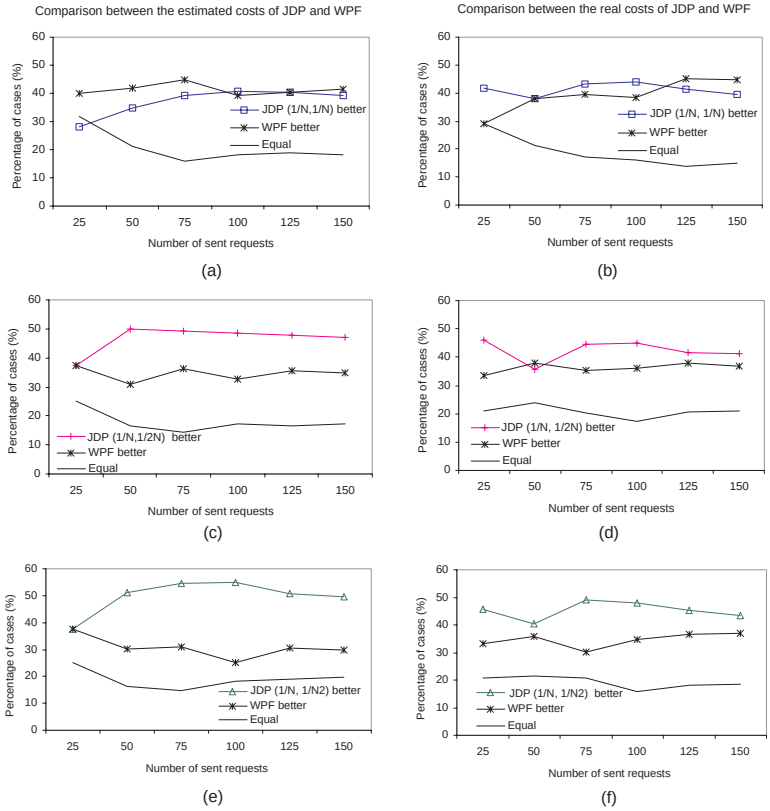

Fig. 4. Advantages of JDP and WPF (a), (c), (e) in estimated cost and (b), (d), (f) in real cost when the number of sent requests increases.

blocking of JDP $(1/N, 1/N)$. This can be explained by the fact that a shorter working path length increases the probability of finding a disjoint backup path. Although JDP is more advantageous at the interdomain step, it finally has slightly more overall blocking due to intradomain blocking [see Fig. 5(b)]. It seems that the interdomain solutions found by WPF are somewhat more realistic than those of JDP since the intradomain blocking probability is smaller. Note that in both approaches, intradomain blocking may result from the impossibility of finding an instance of a backup virtual link that is disjointed with the fixed working path when they cross the same domain. A joint mapping of working and backup virtual links could reduce the blocking. Finally, note that SCI does not block drastically less than WPF and JDP. WPF never blocks 15% more than SCI, and the difference tends to be reduced when the network is increasingly loaded.

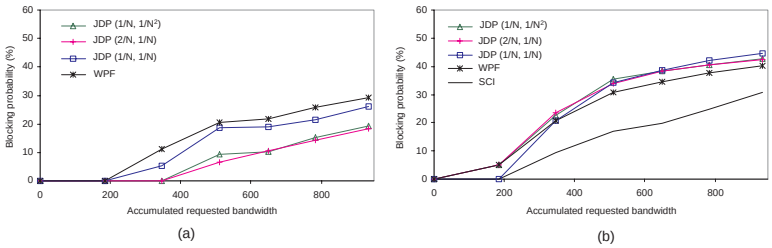

Fig. 5. (a) Bandwidth-blocking probability at the interdomain step and (b) overall bandwidth blocking probability.

6.C. *Effect of Update Frequency on Estimated Cost and Blocking Probability*

In the experiments so far, network link states are updated to border nodes immediately once a light path has been routed. Although in our solution the number of update messages is significantly reduced, this number can be further reduced by a delayed update. In other words, the updates are performed periodically at a short interval. However, the delayed update leaves the link-state information out of date, leading to inaccurate routing. To analyze the effect of short update intervals on the cost and blocking probability, we conducted experiments with WPF. The experiment on JDP is unnecessary since WPF and JDP use the same information scenario and update method. We generated 500 requests according to a Poisson process with rates of $\lambda_1 = 0.25$ (requests/s) and $\lambda_2 = 0.125$ (requests/s). The holding time is exponentially distributed with the mean $h = 160$ s.

The interdomain blocking probability [Fig. 6(a)], as well as the overall blocking probability [Fig. 6(b)], varies slightly when the update interval increases. The estimated cost, which is not shown here, is almost unchanged over different update intervals. We can conclude that short update intervals do not substantially decrease the routing quality though they make it more scalable.

On the other hand, the number of messages per update increases when the updates are more delayed (Fig. 7). However, this number increases at a slower rate than the update interval, leading to a decrease in the total number of update messages when the update interval increases. Furthermore, the number of messages per update is almost constant in the range from 128 to 256.

7. Conclusion

Existing SPP solutions require global and detailed network information. Because such information is not centrally available in multidomain networks, these solutions are no longer

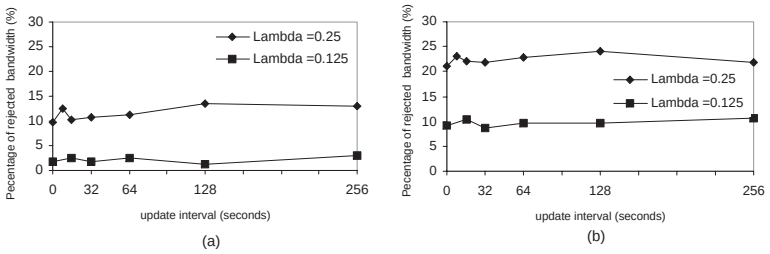

Fig. 6. (a) Bandwidth-blocking probability of WPF at the interdomain step and (b) overall bandwidth blocking probability under different update intervals.

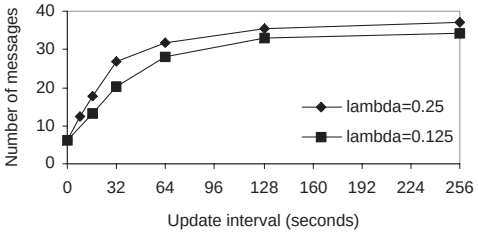

Fig. 7. Number of update messages received by each border node under different update intervals.

applicable. In this paper, we have proposed an information aggregation scenario by underestimation and a two-step routing strategy for SPP in multidomain networks. The main idea is to transform the original multidomain problem into multiple single-domain problems using a topology aggregation combined with the proposed information scenario. Each single-domain problem is solved by using adapted versions of known single-domain SPP algorithms. The computational results show that our solution is not far from the ideal solution obtained using a complete information scenario. In other words, the proposed scheme is efficient and adequately respects the scalability constraint in the same time. Furthermore, we show that a short update interval does not significantly reduce the routing quality but makes the routing more scalable.

The proposed mathematical programming model with the coefficient $\left(1/N, 1/N^2\right)$ jointly computes the directive working and backup paths that minimize the total resource costs. In addition, it finds the shortest directive working path among those minimizing the costs and the shortest directive backup path among those with the same directive working path length. The experimental results show that such a scheme leads to a smaller overall resource cost, followed by more efficient resource utilization thanks to a greater possibility of sharing backup bandwidth.

To reduce the blocking at the intradomain step (and thus the overall blocking), especially when single-domain networks are slightly meshed, future works will concern the joint routing of working and backup paths when they cross the same domain.

A. Appendix A

Proposition: $\beta_{e'}^{\pi} \simeq \max_{e \in \pi} \beta_{e'}^{e}$

Proof:

By combination the definitions of $b_{q'}^{q}, B_{\max}^{q}, b_{\ell'}^{q}$, and the approximation of b_{ℓ}^{ℓ} by expres-

sion (8), we have

$$b^q_{q'} \simeq \sum_{\ell' \in q'} \min\left\{\max\left\{0, B^q_{\max} + d - B_{\ell'}\right\}, d\right\}.$$

Similarly, $b^p_{q'} \simeq \sum_{\ell' \in q'} \min\left\{\max\left\{0, B^p_{\max} + d - B_{\ell'}\right\}, d\right\}$.

We use $q \subset p$ to denote that q is a subset of p. It is clear that $B^p_{\max} = \max_{q \subset p} B^q_{\max}$; then $b^p_{q'} \simeq \max_{q \subset p} b^q_{q'}$.

Combining this with the definition of $\beta^\pi_{e'}$ and $\beta^e_{e'}$ we conclude that $\beta^\pi_{e'} \simeq \max_{e \in \pi} \beta^e_{e'}$.

Acknowledgments

We thank Brigitte Jaumard from Concordia University (Canada) for providing us with access to the ORC laboratory and for precious comments.

References and Links

[1] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," J. Lightwave Technol. **21**, 870–883.

[2] C. Qiao and D. Xu, "Distributed partial information management (DPIM) schemes for survivable networks—Part I," in *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies* (IEEE, 2001), pp. 302–311.

[3] E. Bouillet and J.-F. Labourdette, "Distributed computation of shared backup path in mesh optical networks using probabilistic methods," IEEE/ACM Trans. Netw. **12**, 920–930 (2004).

[4] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-plaining network restoration," in *1998 IEEE International Conference on Communications* (IEEE, 1998), pp. 537–543.

[5] H. Huang and J. A. Copeland, "Multi-domain mesh optical network protection using Hamiltonian cycles," *High Performance Switching and Routing, Workshop on Merging Optical and IP Technologies* (IEEE, 2002), pp. 26–29.

[6] C. Li, S. T. McCornick, and D. Simchi-Levi, "Finding disjoint paths with different path costs: complexity and algorithms," Networks **22**, 653–667 (1992).

[7] M. Kodialam and T. V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (IEEE, 2000), pp. 902–911.

[8] J. Tapolcai and T. Cinkler, "On-line routing algorithm with shared protection in WDM networks," presented at Optical Network Design and Modeling, Budapest, Hungary, 3–5 February 2003, pp. 351–364.

[9] M. Kodialam and T. V. Lakshman, "Dynamic routing of restorable bandwidth-guaranteed tunnels using aggregated network resource usage information," IEEE/ACM Trans. Netw. **11**, 399–410 (2003).

[10] D. Xu, C. Chunming, and Y. Xiong, "An ultra-fast shared path protection scheme distributed partial information management, Part II," in *10th IEEE International Conference on Network Protocols* (IEEE, 2002), pp. 344–353.

[11] G. Bernstein, V. Sharma, and L. Ong, "Interdomain optical routing," J. Opt. Netw. **1**, 80–92 (2002).

[12] J. L. Le Roux, J. P. Vasseur, and J. Boyle, "Requirements for inter-area MPLS traffic engineering," IETF Internet draft (2004), available at draft-ietf-tewg-interareampls-te-req-02.txt.

[13] A. D'Achille, M. Listanti, U. Monaco, F. Ricciato, and V. Sharma, "Diverse inter-region path setup/establishment," IETF Internet draft (2004), available at draft-dachille-diverse-inter-region-path-setup-00.txt.

[14] C. Ou, H. Zang, and B. Mukherjee, "Sub-path protection for scalability and fast recovery in WDM mesh networks," in *Optical Fiber Communication Conference (OFC)*, Postconference Digest, Vol. 54 of OSA Trends in Optics and Photonics Series (Optical Society of America, 2001), ThO6.

[15] A. A. Akyamac, S. Sengupta, and J.-F. Labourdette, "Reliability in single-domain vs. multi-domain optical mesh networks," presented at the National Fiber Optic Engineers Conference, Dallas, Texas, September 15–19, 2002.
[16] T. Miyamura, T. Kurimoto, M. Aoku, and A. Misawa, "An inter-area SRLG-disjoint routing algorithm for multi-segment protection in GMPLS networks," presented at the International Conference on Broadband Networking (ICBN), Kobe, Japan, April 7–9, 2004.
[17] F. Hao and E. W. Zegura, "On scalable QoS routing: performance evaluation of topology aggregation," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (IEEE, 2000), pp. 147–156.
[18] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, And Applications* (Prentice-Hall, 1993).
[19] M. J. O'Mahony, D. Simeonidu, A. Yu, and J. Zhou, "The design of the European optical network," J. Lightwave Technol. **13**, 817–828 (1995).
[20] http://www.rediris.es.
[21] http://www.garr.it.
[22] http://www.renater.fr.
[23] http://www.surfnet.nl.