Abstract

The routing problem for shared path protection in multi-domain optical mesh networks is difficult due to the lack of complete and global knowledge on the network topology and bandwidth allocation. To overcome this difficulty, we propose an aggregated network modeling by overestimation and a two-step routing strategy. In the first step, a rough routing solution is sketched in a compact network which is the topology aggregation of the multi-domain network. A complete routing is then determined by solving routing problems within the original single-domain networks. Each routing step can be solved by either using an exact mathematical or a heuristic method. Computational results show the relevance of the aggregated network modeling and proves the scalability as well as the efficiency of the proposed routing for multi-domain networks.

Key Words: Multi-domain Network, Protection, Routing.

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 lightpaths (see [1]). In the single-failure scenario, two backup lightpaths could share bandwidth among them if their working lightpaths are link or node-disjoint, later called the sharing condition. SPP routing consists in finding a pair of working and backup lightpaths that satisfy the sharing condition and optimize a particular criteria such as bandwidth, number of wavelength conversions, fiber link length, etc. This paper considers the problem of dynamic routing for SPP in multi-domain optical mesh networks while minimizing the total bandwidth required by the working and backup lightpaths. Since the node-disjoint condition is equivalent to the link-disjoint condition by splitting each node into two halves with a "virtual" directed link between them (see [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 [3]). We assume also that all network nodes have OEO treatment so they can switch sub-wavelength and wavelength assignment is easy to handle. There are some static (or off-line) SPP routing approaches proposed for single domain [4] or multi-domain [5] networks. Given a network with known topology, link capacities and future requested traffic, they propose a design with 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, under the current network state, a pair of disjoint working and backup paths that minimally consume bandwidth, while satisfying the sharing condition. This problem has been proven NP-hard in [6]. An exact ILP-based solution called Sharing with Complete Information (SCI) has been proposed in [7]. Different heuristics such as Two-Step Approach (TSA) [8], Sharing with Partial Information (SPI) [9], Distributed Partial Information Management (DPIM) [2] and Active Path First-Backup Path Cost (APF-BPC) [10] have been also proposed. In all cases, the global knowledge (either partial or complete) on each network link and the complete network topology are mandatory at the route computation node.

In multi-domain networks, it is impractical to make this global information available at a node. A multi-domain network is an interconnection of several independent single-domain networks [11] (Figure 1a). To support the scalability, the routing information should not be excessively and frequently exchanged throughout the multi-domain 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 the global multi-domain 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 multi-domain networks.

A few works have been proposed on dynamic protection for multi-domain networks but none have been devoted to SPP. No-sharing path protection was proposed in [13] while no-sharing segment protection was introduced in [14]. The latter was improved in [15] to become segment-shared protection although no details on its routing model were de-
scribed. In [16], a routing for segment shared protection was proposed where a lightpath is not allowed to go through a domain that differs the source and the destination domains. In real multi-domain network, lightpaths often go through several domains. This is illustrated in Figure 1a where a lightpath from $N_1$ to $N_2$ can go through $N_0$.

This paper deals with SPP routing in multi-domain optical networks without global information knowledge. Our main idea is to transform the original multi-domain routing problem into several single-domain routing problems which are solved separately using adapted versions of existing single-domain SPP routings. We propose a two-step routing strategy. First, the multi-domain network is topologically aggregated to become a single-domain network, called inter-domain 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. Each step is performed by using an exact mathematical programming or a heuristic approach. The efficiency of the heuristic is compared to the mathematical programming approach through computational results. To evaluate the relevance of the aggregate information, the approaches are compared to SCI when the latter is applied to multi-domain networks. Also note that we try 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. This allows respecting the scalability constraint.

This paper is organized as follows: notation and the two-step routing strategy is introduced in the next section. Section 3 presents the information aggregation scenario. Section 4 describes the mathematical programming and heuristic approaches. Routing signaling and information dissemination are discussed in Section 5. Section 6 shows the experimental results. Finally, Section 7 concludes the paper.

2 Notation and Fundamental Equations

The multi-domain 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 set $V$ (resp. $V_i$) and $L$ (resp. $L_i$) are respectively the set of nodes and the set of links of $\mathcal{N}$ (resp. $\mathcal{N}_i$). The set of border nodes of domain $\mathcal{N}_i$ is $V_i^\text{BORDER} \subset V_i$. The set of links $L$ decomposes into intra-domain links and inter-domain links $L^\text{INTER}$. An inter-domain link connects two border nodes of two different domains, $L^\text{INTER} = \{(v, v') : v \in V_i^\text{BORDER}, v' \in V_j^\text{BORDER}, i \neq j\}$.

A clique mesh topology aggregation will be applied to $\mathcal{N}_i$, $i = 1, ..., M$, to obtain an aggregated graph $G_i = (V_i^\text{BORDER}, E_i^\text{VIRTUAL})$ where $E_i^\text{VIRTUAL} = \{(v, v') : v, v' \in V_i^\text{BORDER}\}$ is the set of virtual links connecting all pairs of border nodes. The resulting network $G = (V^\text{BORDER}, E)$ is a compact inter-domain network (see an illustration on Figure 1b) contains all border nodes and all virtual and inter-domain links:

$$V^\text{BORDER} = \bigcup_{i=1}^{M} V_i^\text{BORDER},$$

$$E^\text{VIRTUAL} = \bigcup_{i=1}^{M} E_i^\text{VIRTUAL},$$

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

We will denote by $e$ a link of $G$ and $\ell$ a physical link of $\mathcal{N}$. Thus an inter-domain link can be denoted by $e \ell$ or $\ell$. We define $\mathcal{P}_e$ as the set of physical paths within $\mathcal{N}_i$ between $v$ and $v'$ when $e = (v, v') \in E_i^\text{VIRTUAL}$, and $\mathcal{P}_e = \{e\}$ when $e \in L^\text{INTER}$. 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 elements of $\mathcal{P}_e$.

This link state will be disseminated to all border nodes to give them a common aggregated view of the multi-domain network. More details will be given in Sections 3 and 5.

Let us consider a request from a source border node $v_s$ to a destination border node $v_d$ with the requested bandwidth $d$. The requested bandwidth will be routed over a single path. The following notation is introduced:

$p, p'$ are respectively the complete working and backup paths in $\mathcal{N}$ that we have to find for the new request.

$c^\text{res}_\ell$ is the residual capacity on a physical link $\ell \in L$.

$a_\ell$ is the bandwidth used by the working path $p$ on the physical link $\ell \in L$. Since working paths do not share bandwidth, $a_\ell = d$.

$B_\ell$ is the amount of bandwidth used by existing backup paths.
$B'_e$ is the backup bandwidth that has been reserved on $e' \in L$ to protect $e \in L$. This bandwidth is not sharable by any new backup path which protects $e$.

$B^e_{\max}$ and $B^{q}_{\max}$ are the maximal backup bandwidth that has been reserved by a physical link to protect the physical link $e$ and the sub-path $q$ in $N'$ respectively.

$B^e_{\max} = \max_{e' \in L} B'_e$ and $B^{q}_{\max} = \max_{e' \in q} B'_e$.

$b'_e$ (resp. $b^q_e$) is the additional backup bandwidth needed beside $B'_e$ on physical link $e'$ of $p'$ to protect the physical link $e$ of $p$ (resp. the path $q$ in $N$). Observe that $b^q_e = \max_{e' \in q} b'_e$.

$b^q_e$ is the additional bandwidth needed along the path $q'$ in $N'$ to protect the path $q$ in $N$. Observe that $b^q_e = \sum_{q \in q'} b^q_e$.

$\pi$, $\pi'$ are the representations of $p$ and $p'$ in $G$, and are respectively called directive working and backup paths. They are composed of virtual and inter-domain links. Each virtual link represents a sub-path of $p$ or $p'$ in $N'$.

$P_\pi$ (resp. $P_{\pi'}$) is the set of all physical paths obtained by substituting all virtual links of $\pi$ (resp. $\pi'$) by their instances. Clearly, $p \in P_\pi$, $p' \in P_{\pi'}$.

$\alpha_e$ is the total bandwidth that the working path $p$ consumes along its sub-path $q \in N$ represented by $e$ in $G$. Thus $\alpha_e = \sum_{e' \in q} a_e$.

$\beta'_e$ (resp. $\beta^q_e$) is the bandwidth needed on a sub-path $q' \in N'$ of $p'$ represented by $e'$ in $G$ for protecting a sub-path $q \in N$ of $p$ represented by $e$ in $G$ (resp. for protecting $p$ represented by $\pi$ in $G$). Thus, $\beta'_e = b^q_e$ and $\beta^q_e = b'_e$.

$\gamma^\text{res}_e$ called the residual capacity on $e$, is the maximum bandwidth that can be routed over an instance of $e \in E$. $\gamma^\text{res}_e = \max_{q \in P_e, e' \in q} \gamma^\text{res}_{e'}$. Clearly, when $e = e' \in E_{\text{INTER}}$, $\gamma^\text{res}_e = \gamma^\text{res}_{e'}$.

$\|e\|$ is the length of the longest instance of $e$ when $e \in E_{\text{VIRTUAL}}$, and 1 when $e \in E_{\text{INTER}}$. $\|e\| = \max_{q \in P_e} \|q\|$ where $\|q\|$ is the length of $q$ in hops.

The parameters $a$, $\alpha$ and $b$, $\beta$ with different indexes are also called working and backup costs.

Dynamic SPP routing aims to identify for each request a pair of disjoint working and backup paths that minimize the total consumed bandwidth $\sum_{e \in p} a_e + \sum_{e' \in p'} b^q_e$.

This is equivalent to:

$$\min \sum_{e \in \pi} \alpha_e + \sum_{e' \in \pi'} \beta^q_{e'}.$$  

Therefore we propose the following two-step routing algorithm for the inter-domain network. The source border node computes $\pi$ and $\pi'$ in $G$ while minimizing

$$\min \sum_{e \in \pi} \alpha_e + \sum_{e' \in \pi'} \beta^q_{e'}.$$

Intra-domain routing step: First, virtual links of $\pi$ are mapped with physical paths and then those of $\pi'$. The mapping of a virtual link is performed within its domain. The working virtual link $e$ is mapped with the path instance in $P_e$ that minimizes $\alpha_e$:

$$\min_{q \in P_e} \sum_{e' \in q} a_e.$$  

The complete working path $p$ is found once all working virtual links have been mapped. The backup virtual link $e' \in \pi'$ is mapped with the path instance in $P_{e'}$ that minimizes $\beta^q_{e'}$:

$$\min_{q' \in P_{e'}} b^p_{e'} = \min_{q' \in P_{e'}} \sum_{e' \in q'} b^p_{e'}.$$  

All that remains is to solve the problems (1), (2) and (3). However, until the complete working and backup paths are being identified, the values of $\alpha_e$, $\beta^q_e$ and $\beta^q_{e'}$ cannot be computed exactly but only estimated. The next section presents the estimations. We describe in Section 4 the algorithms for the above three problems.

3 Information scenario by overestimation

In this section, we present how $\alpha_e$ and $\beta^q_e$ (thus $\beta^q_{e'}$) are estimated for the inter-domain routing, and how $b^q_e$ is computed for the intra-domain routing. The estimations as well as the computation should only based on aggregated information of domains in order to preserve the scalability.

3.1 Inter-domain information aggregation

The estimations will relax the dependency of the exact values of $\alpha_e$, $\beta^q_e$ on global and detailed information about physical links inside domains.

First of all, $\alpha_e$ is overestimated by the maximal total bandwidth to be consumed on an instance of $e$.

$$\alpha_e = \begin{cases} \|e\|d & \text{if } d \leq \gamma^\text{res}_e, e \in E_{\text{VIRTUAL}} \\ d & \text{if } d \leq \gamma^\text{res}_e, e \in L_{\text{INTER}} \\ \infty & \text{otherwise.} \end{cases}$$  

We next estimate $\beta^q_e$. Let us begin with $b^p_{e'}$. The sharable backup bandwidth on a link $e'$ for protecting a link $e$ is $(B'^e_e - B'_e)$ [7]. Thus $b^p_{e'}$ is the difference between the requested bandwidth and the sharable backup bandwidth on $e'$. Since $b^p_{e'}$ must not be negative, we have:

$$b^p_{e'} = \max\{0, B'_e + d - B'^e_e\}.$$  

Similar to the way that $b^p_{e'}$ is computed in [2], $B'_e$ is overestimated by $B^{\max}_e$ max. Note that $b^p_{e'}$ cannot be greater than the required bandwidth:

$$b^p_{e'} \leq \min\{\max\{0, B^{\max}_e + d - B'^e_e\}, d\}.$$  

By combination with the definition of \( b^q_{e'} \) and \( b^q_{q} \), we have:
\[
b^q_{q} \leq \sum_{e' \in q'} \min \{ \max \{ 0, B^q_{\text{max}} + d - B_{e'} \}, d \}.
\] (7)

If we redefine \( b^q_{q} \) by the right hand side of 7, readers can easily prove that
\[
\beta^q_{e'} = \max_{e \in \pi} \beta^q_{e}. \quad (8)
\]

That means the bandwidth needed on \( e' \) to protect \( \pi \) is not greater than that needed on \( e' \) to protect a virtual or inter-domain link \( e \in \pi \). We need now to estimate \( \beta^q_{e'} \). We can overestimate it as the maximum bandwidth needed on an instance of \( e' \) to protect an instance of \( e \):
\[
\beta^q_{e'} \leq \max_{q \in P_e, q' \in P_{e'}} b^q_{q'} \quad (9)
\]

Let \( B^q_{\text{max}} \) be the maximal backup bandwidth that has been reserved by any physical link \( e' \in L \) to protect an instance of link \( e \in E \):
\[
B^q_{\text{max}} = \begin{cases} 
\max_{q \in P_e} B^q_{q} & \text{if } e \in E^{\text{VIRTUAL}} \\
B^e_{\text{max}} & \text{if } e = e' \in L^{\text{INTER}}. 
\end{cases} \quad (10)
\]

Let \( B^e_{\text{res}} \) be \( B^e_{\text{res}} \) when \( e' = e' \in L^{\text{INTER}} \); and let it be the minimum backup bandwidth that has been reserved on a physical link of domain \( N_i \) when \( e' \) is a virtual link of \( N_i \):
\[
B^e_{\text{res}} = \begin{cases} 
B^e_{\text{res}} \quad & \text{if } e' = e' \in L^{\text{INTER}} \\
\min_{e' \in L_e} B^e_{\text{res}} \quad & \text{if } e' \in E^{\text{VIRTUAL}}. 
\end{cases} \quad (11)
\]

Replace (7) into (9) and combine with (10) and (11), we have
\[
\beta^q_{e'} \leq \max_{q \in P_e, q' \in P_{e'}} \sum_{e' \in q'} \min \{ \max \{ 0, B^q_{\text{max}} + d - B^e_{\text{res}} \}, d \}.
\]

Thus, \( \beta^q_{e'} \) can be overestimated by:
\[
\beta^q_{e'} \leq \|e'\| \min \{ \max \{ 0, B^q_{\text{max}} + d - B^e_{\text{res}} \}, d \}, \quad (12)
\]
i.e. for \( e, e' \in E \):
\[
\beta^q_{e'} = \begin{cases} 
0 & \text{if } \|e'\|(B^q_{\text{max}} + d - B^e_{\text{res}}) = \|e'\|d \leq B^q_{\text{max}} + d - B^e_{\text{res}} \\
\|e'\|(B^q_{\text{max}} + d - B^e_{\text{res}}) & \text{if } \|e'\|(B^q_{\text{max}} + d - B^e_{\text{res}}) > B^q_{\text{max}} + d - B^e_{\text{res}}, \\
\|e'\|d & \text{if } B^q_{\text{max}} + d - B^e_{\text{res}} \geq \|e'\|d, \\
\|e'\|d & \text{otherwise.} 
\end{cases} \quad (13)
\]

Particularly, in order to avoid the situation that the working and backup paths joint at an inter-domain link, \( \beta^q_{e'} = \infty \).

In summary, the working and backup costs of a virtual or inter-domain link in \( G \) are estimated by using only per virtual/inter-domain link values (instead of per link values) such as \( \|e'\|, \gamma^e_{\text{res}}, B^e_{\text{res}} \), and \( B^q_{\text{max}} \). They are defined as link state attributes of virtual or inter-domain links.

### 3.2 Intra-domain partial information

In this section, we are going to identify \( a^p_{\ell} \) and \( b^p_{\ell} \).

The working cost \( a^p_{\ell} \) of the link \( \ell \) is the requested bandwidth if there is enough free bandwidth on \( \ell \):
\[
a^p_{\ell} = \begin{cases} 
d & \text{if } d \leq \gamma^e_{\text{res}} \\
\infty & \text{otherwise.}
\end{cases} \quad (14)
\]

From the definition of \( b^p_{\ell} \), it is easy to deduce: \( b^p_{\ell} = \min \{ \max \{ 0, B^p_{\text{max}} + d - B^0_{\ell} \}, d \} \), i.e.:
\[
b^p_{\ell} = \begin{cases} 
0 & \text{if } B^p_{\text{max}} + d - B^0_{\ell} \leq 0 \\
B^p_{\text{max}} + d - B^0_{\ell} & \text{if } B^p_{\text{max}} + d > B^0_{\ell} \geq B^p_{\text{max}}, \\
d & \text{if } B^p_{\text{max}} \geq B^0_{\ell}, \gamma^e_{\text{res}} \geq d \\
\infty & \text{otherwise.}
\end{cases} \quad (15)
\]

Hence, the intra-domain routing requires \( b^p_{e}, \gamma^e \) of every link \( \ell \) in the domain, and \( B^p_{\text{max}} \) for every link \( \ell \in p \).

### 4 Routing approaches

The two proposed routing approaches perform the intra-domain step identically but the inter-domain differently. The two approaches are then named according to their inter-domain routings.

#### 4.1 Heuristic approach

In the heuristic approach, the working path is routed first, which shall be called Working Path First (WPF). Here, all Shortest Path problem are in terms of cost. They are solved using Dijkstra’s algorithm. WPF follows the below procedure

**Inter-domain routing step:** Instead of minimizing (1), we minimize separately each term of the sum. First, \( \pi \) is set to the shortest path in \( G \) between the source and the destination when working cost \( a^e \) is assigned to each link of \( G \). Subsequently, backup cost \( \beta^q_{e'} \) is assigned to each link of \( G \). Again \( \pi' \) is set to the shortest path in \( G \) between the source and the destination.

**Intra-domain routing step:** First, virtual links of \( \pi \) are mapped one by one within their domains. The virtual link \( e \in \pi \) between \( v \) and \( v' \in N_i \) is mapped with the shortest path in \( N_i \) between \( v \) and \( v' \) when physical links of \( N_i \) are weighted by \( a^i \). The complete working path \( p \) is then obtained. Next, virtual links of \( \pi' \) are mapped similarly but with the backup cost \( b^p_{\ell} \). In the event that both the working and backup paths cross the same domain, the links used by the working path are removed before we perform the mapping of backup virtual links in order to ensure that the two paths are disjoint.

Note that all virtual/inter-domain links as well as physical links with insufficient bandwidth are automatically excluded due to the definitions of working and backup costs.
As the intra-domain routing of the working path is independent of the inter-domain routing of the backup path, the former routing can be performed before the latter. In this case the working path is first routed completely. Then a better estimated backup cost \( \beta^*_p \) can be obtained with \( B_{\text{max}} \) set to \( B_{\text{max}}^p \). The computation effort for \( B_{\text{max}} \) is also reduced. This routing will be called Complete Working Path First (CWPF).

### 4.2 Mathematical programming approach

In this section we consider each link of \( E \) as two directed arcs. However we still keep the notations \( e \) and \( E \) but the former will represent an arc while the latter denotes the set of arcs. Given \( v_i \in V^\text{BORDER} \) \( \Gamma^+(v_i) \) (resp. \( \Gamma^-(v_i) \)) denotes the set of outgoing (resp. incoming) arcs at node \( v_i \). We introduce the following notation: \( x_e=1 \) if the directive working path from \( v_s \) to \( v_d \) goes through arc \( e \), 0 otherwise, \( y_e=1 \) if the directive backup path from \( v_s \) to \( v_d \) goes through arc \( e \), 0 otherwise. The two-step optimization approach is the following

**Inter-domain routing step:** We solve an ILP problem (P) defined in the inter-domain network \( G \) to find a directive working path \( \pi \) and a directive backup path \( \pi' \).

**Intra-domain routing step:** Similar to the intra-domain routing of WPF.

The ILP formulation (P) which is considered is similar to the one proposed in [7, 9]:

\[
\min \sum_{e \in E} \alpha_e x_e + \sum_{e \in E} \gamma_e y_e + \nu \sum_{e \in E} x_e + \mu \sum_{e \in E} y_e
\]

subject to:

\[
\sum_{e \in \Gamma^+(v_i)} x_e - \sum_{e \in \Gamma^-(v_i)} x_e = \begin{cases} 1 & v_i = v_s \\ 0 & v_i \neq v_s, v_d \\ -1 & v_i = v_d \end{cases} \quad (16)
\]

\[
\sum_{e \in \Gamma^+(v_i)} y_e - \sum_{e \in \Gamma^-(v_i)} y_e = \begin{cases} 1 & v_i = v_s \\ 0 & v_i \neq v_s, v_d \\ -1 & v_i = v_d \end{cases} \quad (17)
\]

\[
z_{e'} \geq \beta^*_p (x_e + y_{e'} - 1) \quad e, e' \in E \quad (18)
\]

\[
x_e, y_e \in \{0, 1\} \quad e \in E \quad (19)
\]

\[
z_e \geq 0 \quad e \in E \quad (20)
\]

The two sets of constraints (16) and (17) are respectively flow conservation constraints for the working path and the backup path. Each set represents a path from the source border node \( v_s \) to the destination border node \( v_d \) in \( G \). The backup cost is modeled through constraint (18). Moreover, as a solution of (P) is defined in the directed graph, \( \beta^*_p \) is also updated according to the opposite direction of the arcs \( e \) and \( e' \). Note that the working and backup paths can go through the same virtual link but cannot go through the same inter-domain link. This condition is taken into account in the definition of \( \beta \). 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 \( \alpha \) and \( \beta \) are integers, and \( \nu, \mu \) are sufficiently small so that \( \nu \sum_{e \in E} x_e + \mu \sum_{e \in E} y_e < 1 \), it can be easily proven 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.

### 5 Routing signaling and path setup

#### 5.1 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. The working segments \( q \) and their corresponding \( B^q \) that are found are returned to the source node. Finally the source identifies \( B_{\text{max}}^p \) as the maximum of all \( B^q \) and sends it to the border nodes of the directive backup path so that they can map the backup virtual links.

#### 5.2 Path setup

Once the routing is completed, the paths are setup and the link states of all physical as well as virtual/inter-domain links are updated. It is worth noting that these link states are stored in a distributed way at different border nodes. A border node keeps the link state \( \{e^\text{res}_q, b_t, B_{\text{max}}^q\} \) of all adjacent internal links \( \ell \) and the link state \( \{\|\ell\|, \gamma^\text{res}_\ell, B_{\text{max}}^\ell\} \) of all its adjacent virtual/inter-domain links \( e \). In addition, each internal or border node keeps the set \( B_{\text{max}}^\ell = \{B^\ell_t : \ell \in L\} \) for each link \( \ell \) and \( B^\ell = \{B^\ell_t : \ell \in L\} \) for each link \( \ell \) adjacent to it. The former set is necessary to compute the exact backup bandwidth to reserve by using (5) if the backup path goes through \( \ell \). The latter one allows the computation of \( B_{\text{max}}^\ell \) if the working path goes through \( \ell \).

For setting 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 \( B^\ell \) and the link state \( \{e^\text{res}_q, b_t, B_{\text{max}}^q\} \) where \( \ell \) is an adjacent working link. The new link states is collected with the signaling message until the domain’s egress border node. Here, they are forwarded to other domain border nodes for synchronization. The number of update messages is \( O(|V^\text{BORDER}|) \) where \( N_d \) is the current domain. The process goes on until the destination is reached.
Finally, every border node locally updates the link states (\( e_{e \to e'}^{\text{res}}, B_{e \to e'}, B_{e'}^{\text{res}} \)) of each backup link \( e' \). The number of update messages is also \( O(|V_{\text{BORDER}}|^2) \).

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 lightpath request is \( O(|V_{\text{BORDER}}|^2) \), if we assume that the working and backup paths cross the \( K < M \) first domains.

Clearly, \( O(|V_{\text{BORDER}}|^2) \) is smaller than the number of update messages in single-domain SPP approaches which is \( O(|V||V_{\text{BORDER}}|) \) since an all-to-border node update is required. Moreover, the size of each message is always \( O(1) \). This proves that our approach is more scalable than single-domain approaches.

6 Computational results

![Experimental network](image)

**Figure 2.** Experimental network.

We will evaluate the relevance of the information aggregation scenario by comparing the results of the proposed routing with the optimal solution given by SCI [9]. The efficiency of heuristic is also compared to the mathematical programming approach. The computational results are conducted on a five-domain network composed of EONet, RedIRIS, GARR, Renater3, SURFinet [17, 18] with real link capacities for the last four networks. The inter-domain links are artificially added with OC-192 capacities (see Figure 2). Requests for static traffic are randomly generated between border nodes. Requested bandwidths are uniformly distributed among OC-1, 3, 6, 9, 12.

We denote the results of WPF and CWPF by WPF-Max and CWPF-Max. Opt \((\nu, \mu)\) will be used to denote the optimization approach with fixed parameters of \((\nu, \mu)\). In this experiment, \(\alpha\) and \(\beta\) are integers because \(d\) is integer, and \((\nu, \mu)\) are chosen such that \(\nu \sum_{e \in E} x_e + \mu \sum_{e \in E} y_e < 1\). Thus the total bandwidth costs are minimized. In addition, when \((\nu, \mu) = (\frac{1}{N}, \frac{1}{N^2})\), where \(N = |E|\), the shortest direct working path \(\pi\) and the shortest direct backup path \(\pi'\) among all candidates associated with \(\pi\) are obtained.

This configuration is expected to also give shorter complete working paths leading to more possibility of sharing backup bandwidth.

The commercial software CPLEX and the academic version of OPNET Modeler are respectively used to implement the mathematical programming and the heuristic on a 1.9-GHz Pentium 4. The computational time for routing a request is less than 16 ms for both WPF-Max and CWPF-Max, and less than 1 minute for Opt.

6.1 Analysis of bandwidth costs

In order to determine how our solution is far from the ideal solution in terms of bandwidth savings, we compared the total working and backup path costs found by WPF-Max, CWPF-Max and Opt with SCI. Let \(\text{cost}_{\text{WPF-Max}}^r\) (resp. \(\text{cost}_{\text{Opt}}^r\)) be the total bandwidth cost of the complete working and backup paths in the case of WPF-Max (resp. Opt), and \(\text{cost}_{\text{SCI}}\) the total cost of SCI. The relative gap between \(\text{cost}_{\text{WPF-Max}}^r\) and \(\text{cost}_{\text{SCI}}^r\) is defined by:

\[
\text{gap}_{\text{WPF-Max/SCI}} = \frac{\text{cost}_{\text{WPF-Max}}^r - \text{cost}_{\text{SCI}}}{\text{cost}_{\text{SCI}}}
\]

and similarly for \(\text{gap}_{\text{Opt/SCI}}\). The definition holds for CostWPF-Max. Figure 3 depicts the distribution of \(\text{gap}_{\text{WPF/SCI}}\) and \(\text{gap}_{\text{Opt/SCI}}\). The 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 only computed for the requests that are successfully routed by SCI and WPF-Max, CWPF-Max or Opt. Observe that the cost of SCI is

![Distribution of the relative gap with SCI for WPF-Max, CWPF-Max and Opt](image)

**Figure 3.** Distribution of the relative gap with SCI for WPF-Max, CWPF-Max and Opt.
generally smaller than that of WPF-Max, CWPF-Max and Opt \( \frac{1}{N}, \frac{1}{N^2} \) since the gap is positive most of the time. This is a natural observation since SCI is routed according to the complete information scenario. Another observation is that the percentage of cases where the gap is within \([-0.5, 0.5]\) for Opt \( \frac{1}{N}, \frac{1}{N^2} \), WPF-Max and CWPF-Max is respectively 68, 60 and 60. Thus, roughly, the real cost of the solution found by WPF and by Opt is not so far from the solution found by SCI.

We compare Opt and WPF-Max with respect to frequency of finding smaller estimated and real path costs. Figure 4a (resp. Figure 4b) shows the percentage of cases for which Opt \( \frac{1}{N}, \frac{1}{N^2} \) or WPF-Max finds better (smaller) total estimated (resp. real) costs when the number of sent requests increases. Most of the time Opt \( \frac{1}{N}, \frac{1}{N^2} \) gives better total estimated and real costs. The real costs obtained by CWPF-Max are similar to that of WPF-Max while the estimated costs of CWPF-Max are smaller (so better) in 15% of cases (those results of CWPF-Max are not depicted). We observe that WPF-Max or CWPF-Max is overall slightly better than Opt \( \frac{1}{N}, \frac{1}{N^2} \) in estimated and real costs thanks to their shorter working paths. As we have seen, Opt is improved when the length of working path is minimized with \((\nu, \mu) = \left( \frac{1}{N}, \frac{1}{N^2} \right)\). This confirms the expectation that when there are less virtual links, the real cost is reduced. On the other hand, when the working path is short, there are more opportunities to share backup bandwidth with the future lightpath requests because there is less chance of violating the sharing constraint due to link-joint working paths. As a result, the overall resource utilization is improved. This explains why WPF obtains a relatively good performance since it always looks for the shortest working path first.

6.2 Blocking Probability Analysis

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 5a shows the bandwidth blocking probability at the inter-domain step (a) and Overall bandwidth blocking probability (b).
that of WPF-Max, CWPF-Max and Opt. In fact, CWPF-Max never blocks 20% more than SCI.

7 Conclusion

Existing SPP solutions require global and detailed network information at a central node which is not allow-
able in multi-domain networks. This paper proposes a two-step routing for SPP in multi-domain networks. The main idea is to transform the original multi-domain problem to multiple single-domain problems using a topology aggregation combined with an overestimation 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.

The proposed optimization model jointly computes the directive working and backup paths that minimize 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. In order to reduce the blocking at the intra-domain step future works will deal with the joint mapping of working and backup paths when they cross the same domain.

Acknowledgements

We would like to thank Dr. Omar Cherkaoui from UQAM for providing us with access to the Laboratory of Teleinfo.

References