Backup Path Re-optimizations for Shared Path Protection in Multi-domain Networks

B. Jaumard
Concordia Institute for Information Systems Engineering
Concordia University, Canada
Email: bjaumard@ciise.concordia.ca

D. L. Truong
Computer Science and Operations Research
Université de Montréal, Canada
Email: truongtd@iro.umontreal.ca

***Abstract*— Within the context of dynamic routing models for shared path protection in multi-domain networks, we propose a backup path re-optimization phase with possible rerouting of the existing backup paths in order to increase the bandwidth sharing among them while minimizing the network backup cost. The re-optimization phase is activated periodically or when routing a new connection fails because of insufficient capacity. Three re-optimization models are discussed: i) Global rerouting where the re-optimization is performed once for the entire network ii) Local rerouting where the re-optimization is serially performed on one domain at a time or on selected domains, and iii) Local rerouting with least effort, i.e., where the smallest possible number of backup path reroutings is performed in order to be able to handle new connection requests. The first model offers the best resource savings while the two others are more scalable in multi-domain networks. Comparative performance of the three models are conducted and numerical results are presented.**

I. Introduction

Shared Protection [1] has been widely studied in the literature. It allows bandwidth sharing amongst backup paths leading to some bandwidth savings while continuing to guarantee 100% failure recovery. Within the single-failure context, 100% failure recovery condition is expressed with the condition that the working paths of the backup paths that share bandwidth must be disjoint. Routing for shared protection aims to identify the working and backup paths that minimize the total bandwidth consumption. We consider the problem for the networks with bandwidth guaranteed connections such as MPLS, ATM or Optical network. The later should be equipped with Multi-service Provisioning Platforms (MSPP) [2] with bandwidth grooming and wavelength conversion capacity at every node. The wavelength assignment problem and wavelength continuity constraint are thus relaxed. Existing solutions follow two paradigms: static routing (off-line) and dynamic routing (on-line). In static routing, the network traffic, i.e. requests for connections, are assumed to be stable and are given as input to the routing model. The working and backup capacities are then optimized for every network links, see, e.g., in [3]–[6]. Conversely, dynamic routing is proposed for dynamically changed traffic and requests for connections are routed one at a time without taking into account any information on the future requests, see, e.g., [7]–[9]. As the time goes, the total allocated bandwidth will be larger (less optimized) than as if routing policy with a global view on the arriving connections or at least a forecast about them had been applied.

It is known and has been already studied in [10], [11] that, if we use dynamic routing but reorganize the existing paths in the network, working bandwidth could be freed and increased bandwidth sharing for the backup bandwidth can be obtained leading to a greater resource saving. The reorganization includes finding alternate paths for the existing working and backup paths and then rerouting some working and backup paths. Moving the traffic of a connection on a new working path implies service interruption, and therefore a disorder for the user, that is to be avoided as much as possible. However, backup paths are generally inactive until a failure occurs. They can be replaced by new ones without any impact on service availability. Therefore, a reorganization scheme in which only backup paths are rerouted offer a good mean to answer to the drawback of the possible bandwidth waste involved in dynamic routing due to the uncertainty about estimating and anticipating the future connection requests.

Few publications exist on rerouting algorithms in the context of dynamic routing. A reference to it can be found in [11], but no detailed algorithm is provided. We propose thus solutions for rerouting backup paths with objective to seize the backup capacity. The solutions differ from the backup path reroute solutions, see, e.g., [12], which aim to improve the service availability at dual failures.

A multi-domain network (see Fig.1) is composed of multiple single-domain networks interconnected amongst them by inter-domain links going from border nodes of some domains to border nodes of another. Multi-domain networks are characterized by the *scalability constraint*, defined in [13], that no global information is available centrally and limited routing information is exchanged in small scope [14]. In a previous paper [13], we have proposed two dynamic routing models, called WPF (Working Path First) and JDP (Joint Directive Path), for Shared Path Protection for multi-domain networks. In these models, a request is routed a time with the objective of minimizing its total requested working and backup bandwidth. Although the models satisfy the scalability issue, it suffers from some drawbacks regarding unnecessary bandwidth waste for the backup paths. In this paper, we propose a rerouting model to enhance WPF and JDP. An additional phase will be trigger after WPF and JDP which reroutes existing backup paths in order to re-optimize the network backup capacity. It is called alternatively re-optimization or rerouting phase;

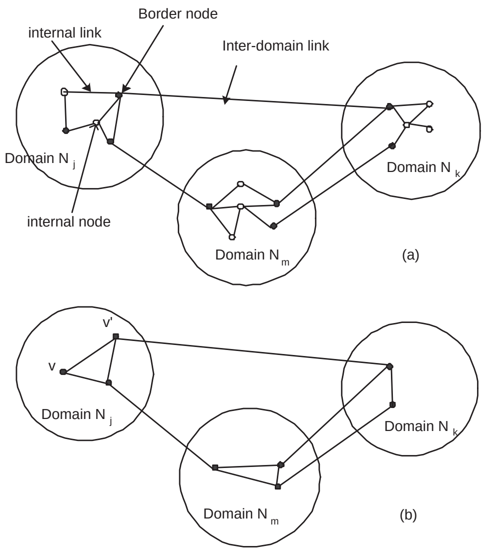

Fig. 1. Illustration of a Multi-domain network

it requires extra computation effort and network information exchange while tearing down old paths and setting up new paths. For reducing this effort and information exchange, the rerouting phase should not be activated regularly but once after a given period of time.

In the next Section, we present the backup path rerouting problem. We propose in Section III-B the *Global reroute* model, in which the end-to-end backup paths in the network are rerouted at once. Due to the global information requirements, the model is only suitable for single domain network. We next propose a *Local Reroute* model to be used for multi-domain network in Section III-C. There, each domain subsequently reroutes the segments of backup paths within it. In Section III-D, the *Least Local Reroute* model where, in each domain, only a minimal number of backup segments will be rerouted. The integration of WPF and JDP with these rerouting models will be compared with original WPF and JDP without rerouting. Numerical results are described in Section IV. The trade off of reroute phase in terms of computational effort and information exchanges is also discussed. Conclusions are drawn in the last section.

II. The Backup Path Rerouting Problem

Let us consider a multi-domain network with a set K connection requests that are already routed in the network, i.e., a working and a backup path (denoted by p_k and p'_k) has been already defined for them. Request k asks for a connection from source s_k to destination d_k for bandwidth b_k. The backup path rerouting problem is stated as follows. Let $\mathcal{R}^B \subset K$ is the index set of the requests whose backup paths might be rerouted. While all working paths should remain unchanged, we look for the set of alternative paths of the current backup paths whose indexes are in $\mathcal{R}^B$, that minimizes the overall bandwidth required for the backup paths. If changed, the backup path of the request $k \in \mathcal{R}^B$ must remain disjoint from the working path p_k so that it will not fail when p_k fails upon a single failure. The fewer backup paths are rerouted, the more scalable and practical the solution is, but may be the less bandwidth saving will be obtained. When all backup paths are allowed to be rerouted $\mathcal{R}^B = K$, the best bandwidth saving will be attained.

III. Mathematical models

A. Notations

Let us represent the multi-domain network by a directed graph $G = (E, V)$ where V is the set of nodes and E is the set of fiber links. The reversed fiber link of the link $e \in E$ is denoted by $\bar{e} \in E$. Each network link joins two nodes and is assumed to be bi-directional with two fibers, each carrying the traffic in one direction. The two fibers are assumed to be fold together in the same conduct so they share the same risk upon a single failure. Each fiber is represented by an arc and a network link is represented by a pair $(e, \bar{e})$ of fiber links. We denote by c_e the bandwidth capacity that is available on the fiber link e.

Arc $e \in E$ is associated with binary parameters δ^W_{ek} and δ^B_{ek} such that:

$$\delta^W_{ek} = \begin{cases} 1 & \text{if } e \in p_k \\ 0 & \text{otherwise} \end{cases} \quad e \in E, k \in K, \tag{1}$$

$$\delta^B_{ek} = \begin{cases} 1 & \text{if } e \in p'_k \\ 0 & \text{otherwise} \end{cases} \quad e \in E, k \in K \setminus \mathcal{R}^B. \tag{2}$$

For a given node $v \in V$, we denote by $\Gamma^+(v)$ its set of outgoing edges, and by $\Gamma^-(v)$ its set of incoming edges.

B. Global reroute

1) Variables: We introduce two sets of variables. The first set, $B_e, e \in E$, defines for each B_e, the bandwidth required for backup paths going through a given arc $e \in E$. We next define variables y^k_e that are decision variables such that:

$$y^k_e = \begin{cases} 1 & \text{if } e \text{ belongs to the backup path of } k \\ & \text{after the rerouting phase} \\ 0 & \text{otherwise.} \end{cases}$$

2) Objective function: In the *Global reroute* model, the objective is to minimize the bandwidth required for all backup paths. The bandwidth required for working paths remain unchanged as no alternative path is sought for them. The objective can then be written:

$$\min \sum_{e \in E} B_e. \tag{3}$$

3) Constraints:

$$\sum_{e \in \Gamma^+(v)} y^k_e - \sum_{e \in \Gamma^-(v)} y^k_e = \begin{cases} 1 & \text{if } v = s_k \\ 0 & \text{if } v \neq s_k, d_k \\ -1 & \text{if } v = d_k \end{cases} \quad v \in V, k \in \mathcal{R}^B \tag{4}$$

$$\sum_{k\in\mathcal{R}^B} b_k(\delta^W_{ek}+\delta^W_{\bar{e}k})y^k_{e'} \le B_{e'} - \sum_{k\in K\setminus\mathcal{R}^B} b_k(\delta^W_{ek}+\delta^W_{\bar{e}k})\delta^B_{e'k} \qquad e,e'\in E \quad (5)$$

$$\delta^W_{ek}+\delta^W_{\bar{e}k}+y^k_e+y^k_{\bar{e}} \le 1 \qquad e\in E, k\in\mathcal{R}^B \quad (6)$$

$$\sum_{k\in K} b_k\delta^W_{ek}+B_e \le c_e \qquad e\in E. \quad (7)$$

Variable domains:

$$y^k_e \in \{0,1\} \qquad e\in E, k\in\mathcal{R}^B \quad (8)$$

$$B_e \ge 0 \qquad e\in E. \quad (9)$$

Constraints (4) are the flow conservation ones for the rerouted backup paths. Constraint (5) ensures that the backup bandwidth $B_{e'}$ on e' will never be smaller than the bandwidth needed to protect every working path against a single failure on the fiber pipe containing the pair $(e,\bar{e})$ of fiber links. This latter backup bandwidth is indeed the bandwidth of the working paths over e or $\bar{e}$ that are protected by the backup paths going through e'. Constraint (6) assures that p_k and p'_k are always link disjoint. Constraint (7) guarantees that the bandwidth used by both working and backup paths over a link will not exceed the link capacity. If there is a loop in p'_k the loop will be removed a posteriori.

This model provides optimal rerouting but is not scalable for multi-domain networks due to global information requirements in model building. For gathering the data of constraints (5), a central node needs to keep the routes of all the working paths in the network. It also needs the complete knowledge of the network topology and bandwidth allocation on the fiber links.

C. Local reroute

In order to overcome the drawback of the *Global reroute* model, we next propose the *Local reroute* one. Instead of rerouting the end-to-end backup paths as in the *Global reroute* model, each domain reroutes locally their inner backup segments in order to minimize its backup capacity. For each segment, the ingress and egress border nodes remain unchanged. The alternate backup paths still go through the same border nodes and inter-domain links. The model that computes the alternate backup segments for domain $D=\{E^{\text{D}},V^{\text{D}}\}$ is called RRLocal(D). Let $s^{\text{D}}_k, t^{\text{D}}_k$ be respectively the ingress and the egress border nodes of p'_k in the domain D. The model is however similar to the *Global reroute* model in respect to parameter initializations and variable domains.

1) Objective function: We minimize the backup bandwidth consumed by domain D:

$$\text{Minimize} \sum_{e\in E^{\text{D}}} B_e \quad (10)$$

2) Constraints:

$$\sum_{e\in\Gamma^+(v)} y^k_e - \sum_{e\in\Gamma^-(v)} y^k_e = \begin{cases} 1 & \text{if } v=s^{\text{D}}_k \\ 0 & \text{if } v\ne s^{\text{D}}_k, d^{\text{D}}_k \\ -1 & \text{if } v=d^{\text{D}}_k \end{cases} \qquad v\in V^{\text{D}}, k\in\mathcal{R}^B \quad (11)$$

$$\sum_{k\in\mathcal{R}^B} b_k(\delta^W_{ek}+\delta^W_{\bar{e}k})y^k_{e'} \le B_{e'} - \sum_{k\in K\setminus\mathcal{R}^B} b_k(\delta^W_{ek}+\delta^W_{\bar{e}k})\delta^B_{e'k} \qquad e\in E, e'\in E^{\text{D}} \quad (12)$$

$$\delta^W_{ek}+\delta^W_{\bar{e}k}+y^k_e+y^k_{\bar{e}} \le 1 \qquad e\in E^{\text{D}}, k\in\mathcal{R}^B \quad (13)$$

$$\sum_{k\in K} b_i\delta^W_{ek}+B_e \le c_e \qquad e\in E^{\text{D}}. \quad (14)$$

Constraints (11)-(14) are similar to constraints (4)-(7) of the *Global reroute* model except that they are applied only to the backup segments in domain D. The whole reroute process over multi-domain network follows the pseudo-code:

For all D in G RRLocal(D)

The *Local reroute* model requires smaller scope of information than *Global reroute* model. Except for the constraint (12) that requires each border node of D to keep the routes of working paths protected by an arc of D, other constraints are built with the information within the domain D. The solution is much more scalable and the resulting mathematical model, being smaller, is much easier to solve.

D. Least Local Reroute Model

The *Least Local Reroute* model (LeastRRLocal) is a further development of the *Local Reroute* model with $\mathcal{R}^B=K$. All backup paths are allowed to be rerouted with a rerouting preference level. The rerouting preference level of the backup path p_k is defined by $w_k\in[0,1]$. The smaller w_k is, the less preference is given to the rerouting p'_k. The model looks for a rerouting configuration with minimal backup capacity for the primary objective and the least weighted number of rerouted backup paths for the secondary objective.

1) Variables: A decision variable r_k is associated with each request k indicating if p'_k will be rerouted ($r_k=1$) or remain unchanged inside the domain D ($r_k=0$).

2) Objective function: A second term counting the weighted number of rerouted backup segments is added to the objective function with coefficient M_1 sufficiently large as to make the second term smaller than 1. Since the first term is integer, the second term selects the solution with the least weighted number of reroutings when breaking ties is needed.

$$\text{Minimize} \sum_{e\in E^{\text{D}}} B_e + \frac{1}{M_1}\sum_{k\in P} w_k r_k \quad (15)$$

3) Constraints: Let M_2 be a large constant.

$$\sum_{e\in\Gamma^+(v)} y^k_e - \sum_{e\in\Gamma^-(v)} y^k_e + M_2(1-r_k) \ge \begin{cases} 1, & \text{if } v=s^{\text{D}}_k \\ 0, & \text{if } v\ne s^{\text{D}}_k, d^{\text{D}}_k \\ -1, & \text{if } v=d^{\text{D}}_k \end{cases} \qquad v\in V^{\text{D}}, k\in K. \quad (16)$$

$$\sum_{e\in\Gamma^+(v)} y^k_e - \sum_{e\in\Gamma^-(v)} y^k_e - M_2(1-r_k) \le \begin{cases} 1, & \text{if } v=s^{\text{D}}_k \\ 0, & \text{if } v\ne s^{\text{D}}_k, d^{\text{D}}_k \\ -1, & \text{if } v=d^{\text{D}}_k \end{cases} \qquad v\in V^{\text{D}}, k\in K. \quad (17)$$

$$y_e^k + M_2 r_k \geq \begin{cases} 1, & \text{if } e \in p'_k \\ 0, & \text{otherwise} \end{cases} \quad e \in E^{\mathrm{D}}, k \in K \quad (18)$$

$$y_e^k - M_2 r_k \leq \begin{cases} 1, & \text{if } e \in p'_k \\ 0, & \text{otherwise} \end{cases} \quad e \in E^{\mathrm{D}}, k \in K \quad (19)$$

$$\delta_{ek}^W + \delta_{\bar{e}k}^W + y_e^k + y_{\bar{e}}^k \leq 1, \quad e \in E^{\mathrm{D}}, k \in K \quad (20)$$

$$\sum_{k \in K} b_k(\delta_{ek}^W + \delta_{\bar{e}k}^W) y_{e'}^k \leq B_{e'}, \quad e \in E, e' \in E^{\mathrm{D}} \quad (21)$$

$$\sum_{k \in K} b_k \delta_{ek}^W + B_e \leq c_e, \quad e \in E^{\mathrm{D}}. \quad (22)$$

Variable domains:

$$r_k \in \{0, 1\}, \quad k \in K \quad (23)$$

$$y_e^k \in \{0, 1\}, \quad e \in E, k \in K \quad (24)$$

$$B_e \geq 0, \quad e \in E. \quad (25)$$

If a path p'_k is rerouted, the flow conservation constraints (11) must be enforced, otherwise the parameter initializations (2) must hold. Since the set of backup paths to be rerouted is still unidentified, the flow conservation constraints and parameter initializations are built in such a way that only one of them is applied for a given backup path. Inequalities (17) and (16) are flow conservation constraints for the rerouted backup paths while (18) and (19) initialize δ_{ek}^B for the unchanged backup paths. Constant M_2 enables only one of the two groups and makes the other one redundant. Indeed, M_2 is sufficiently large if it is greater than the highest incoming and outgoing degrees of a node, thus $M_2 > \max\{\max_{v,k} \sum_{e \in \Gamma^-(v)} y_e^k, \max_{v,k} \sum_{e \in \Gamma^+(v)} y_e^k\}$. The remaining constraints are similar to those of the *Local reroute* model.

When $w_k = 1, k \in K$, the same backup capacity as in the *Local reroute* model is obtained but only the minimum number of backup segments is rerouted in order to minimize the requested backup bandwidth. Less backup segments must be torn down and reserved and less information need to be exchanged in domains. For further reducing the information exchanges, the rerouting should be activated periodically after a time period, or when we reach a blocking with the routing using WPF or JDP (RRLocal-Block). In the latter case, we reroute only in the blocked domain expecting that some bandwidth could be released, then retry WPF or JDP again.

IV. Experiment results

The proposed rerouting will be evaluated on their backup bandwidth saving, blocking probability reducing and scalability. The experiment is performed on WPF (because JDP itself provides similar result as WPF) with the following schemes:

- Without reroute: WPF-noRR.
- With *Least Local reroute*, when $w_k = 1, k \in K$, after 50 or 100 requests, i.e. WPF-LeastRRLocal-50, WPF-LeastRRLocal-100.
- With *Least Local reroute* uniquely in blocked domain upon blocking: WPF-RRLocal-Block.

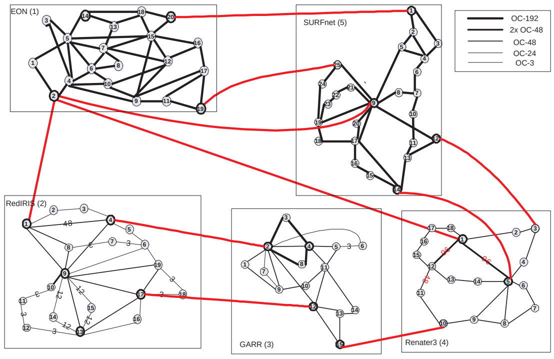

Fig. 2. Experimental multidomain network.

No experiment was conducted with the *Global reroute* model due to its high computational effort and its lack of scalability for multi-domain networks. Experiment on RRLocal will not be shown neither because when $\mathcal{R}^B = K$, the results are similar to those of LeastRRLocal whereas in the later the minimum number of backup paths are modified. The experiment with $\mathcal{R}^B \neq K$ is left for the future due to the limited space for this paper. The multi-domain network instance is composed of 5 real optical single domain networks: EONnet [15], RedIRIS [16], GARR [17], Renater3 [18], SURFnet [19] with link capacities varying from OC-3 to OC-192 (see Fig.2). Some inter-domain links of OC-192 have been added. For each experiment, a sequence of 1000 requests are sent. These requests are between randomly selected border nodes with requested bandwidth uniformly distributed in OC-{1, 3, 6, 9, 12}. Requests arrive according to Poisson process with the rate $\lambda = 0.25$ (requests/s). The request holding time is exponentially distributed with mean $h = 320(s)$. The experiment result will be shown after the 300^{th} request when the network load is stable with an average of 80 simultaneous active connections. This load is sufficient to produce blocking in the network.

CPLEX is used to solve the two rerouting models and Opnet Modeler is used for implementing WPF and simulating the network environment. It takes less than 20 seconds for a rerouting by LeastRRLocal on a Pentium IV-3Ghz. (It takes however about 6 days to solve the *Global reroute* model).

For later convenience, we consider the whole process of rerouting as a single one in RRLocal-Block or LeastRRLocal. It includes multiple simultaneous backup segment reroutings within one or different domains.

A. Backup bandwidth saving

The ability of saving backup bandwidth in RRLocal and LeastRRLocal will be highlighted by comparing the backup capacity (backup cost) obtained in using these schemes with that of WPF-noRR. Here, link capacities are uncapacitated for getting rid of the influence of the blocking cases. Fig.3 shows backup costs of WPF-LeastRRLocal-50, WPF-LeastRRLocal-100 and WPF-noRR. The backup cost of the first two schemes reduces at each rerouting illustrating the released bandwidth thanks to backup path reroutings. We define the relative backup

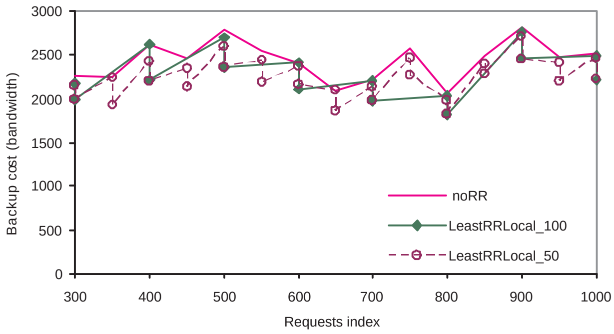

Fig. 3. Backup costs of WPF in different rerouting schemes.

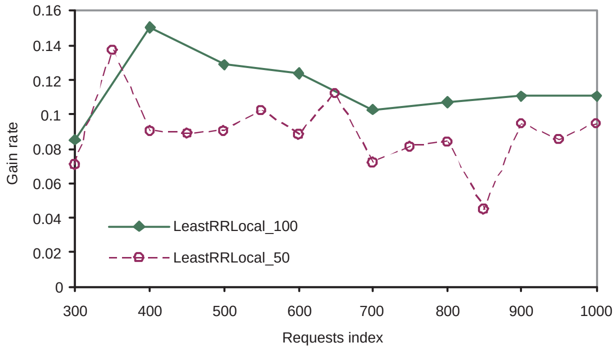

Fig. 4. Relative backup cost gains of WPF in different rerouting schemes.

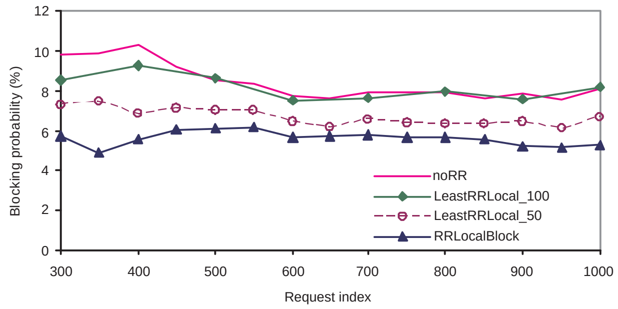

Fig. 5. Blocking probability of WPF in different rerouting schemes.

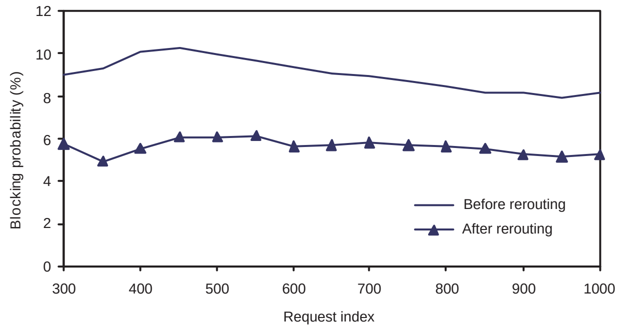

Fig. 6. Blocking probability of WPF-RRLocal-Block before and after rerouting.

cost gain as the fraction between released bandwidth and the network backup capacity before rerouting. Fig.4 depicts the gains of each rerouting scheme: for WPF-LeastRRLocal-100 it is an average of 11.5% and for WPF-LeastRRLocal-50 it is an average saving of 9.8%. Less backup bandwidth is released by WPF-LeastRRLocal-50 at each rerouting because the backup paths has been re-organized not so long before. However, WPF-LeastRRLocal-50 frees more frequently backup bandwidth than WPF-LeastRRLocal-100, after each 50 requests against 100 requests; and thus leaves more room for the requests arriving between two reroutings resulting in less blocking as we will see in the next section.

B. Blocking probability

For evaluating the impact of rerouting on blocking probability, capacities are set back on fiber links. Fig.5, shows the blocking probabilities of WPF in different rerouting schemes. WPF-RRLocal-Block is the best scheme in terms of blocking probability. It reduces the blocking of WPF-noRR about 3%, note that the original blocking is between 8%-10%. WPF-LeastRRLocal-50 and WPF-LeastRRLocal-100 follow up with more modest results. This is explained by the blocking driven nature of RRLocal-Block. In RRLocal-Block, when a request is blocked, a local rerouting is activated at the domain where the blocking occurs, after that the blocked request is routed again. The rerouting has thus an immediate deblocking impact. It is easy to see in Fig.6 the deblocking capacity of WPF-RRLocal-Block through the distance between two blocking probabilities before and after rerouting. Although WPF-LeastRRLocal seizes backup bandwidth regularly, blocking may still occur at a later stage after a rerouting because of non-optimized bandwidth allocation for the subsequent requests, which are not re-organized until the next rerouting. That is why WPF-LeastRRLocal-50 and WPF-LeastRRLocal-100 have a higher blocking probability than RRLocal-Block.

Conforming with the expectation in the previous experiment, WPF-LeastRRLocal-50 blocks 0.5% less than WPF-LeastRRLocal-100 because it re-organizes more frequently backup capacity thus leaving more free capacity for the new coming requests.

C. Scalability evaluation

The scalability of LeastRRLocal and RRLocal-Block over noRR will be first of all evaluated based on the scope of the exchange of the information they require. Let us begin with the computation of rerouted paths. As discussed at the end of Section III-C, for a domain D, RRLocal, therefore LeastRRLocal and RRLocal-Block, requires that border nodes of D keeps the routes of all working paths protected by a link of D. This requirement could be easily satisfied by benefiting from the backup path reservation process of WPF, which forwards the route of a working path along its backup path (see [13] for details). Therefore, WPF-LeastRRLocal and WPF-RRLocal-Block do not require any extra information exchange in comparison with WPF-noRR, although a larger information storage is required.

TABLE I

INFORMATION EXCHANGE SCOPES

	LeastRRLocal	RRLocal-Block
Extra information exchange over WPF in path computation	no	no
Signaling scope	All domains	Blocked domain

TABLE II

NUMBER OF REROUTED BACKUP SEGMENTS

Domain (seg./rerouting)	LeastRRLocal-{f}		RRLocal-Block
	f= 50	f=100	
EON	10.75	14.2	-
RedIRIS	13.45	16.5	-
GARR	4.65	5.6	-
Renater3	9.6	8.3	-
SUFnet	5.15	5.8	-
All domains	43.6	50.4	8.92
All domains, reroutings (seg.)	872	504	375

The rerouted path computation is followed by the signaling process which is composed of i) tearing down the old backup segments, ii) reservation of the new backup segments within domains. In LeastRRLocal the signaling is needed in all domains while RRLocal-Block requires it only in one domain because backup paths are rerouted respectively in all domains and one domain. Table I summarizes the qualitatively comparison on information exchange in the two methods.

Another important factor of the scalability is the number of rerouted backup segments generated in each rerouting method. It is recommended to keep it small as the quantity of information to be exchanged and the number of operations to be performed on network nodes during the signaling process increases proportionally with the number of rerouted backup segments. Table II presents the average number of rerouted backup segments in each domain, in all domains per rerouting and the total number of rerouted segments after all reroutings in the cases of LeastRRLocal-50, LeastRRLocal-100 and RRLocal-Block. Over the entire network, LeastRRLocal-50 changes nearly as many backup segments per rerouting as LeastRRLocal-100: 43.6 versus 50.4 segments/rerouting. RRLocal-Block reroutes considerably fewer backup segments per rerouting: 8.92 segments, because it only reroutes the backup segments within blocked domains.

Although high rerouting frequency allows to reduce more blocking, it increases the overall number of rerouted backup segments. Globally, LeastRRLocal-50 involves nearly twice the number of backup segments in rerouting than LeastRRLocal-100: 872 versus 504 rerouted segments. On the other hand, RRLocal-Block reroutes only 375 segments. Note that in this experiment, RRLocal-Block re-organizes quite often backup segments, about 8 times per 100 requests, because of blocking due to high network load.

From the above analysis, we can conclude that RRLocal-Block is more scalable than LeastRRLocal-100, which is in it turn more scalable than LeastRRLocal-50.

V. CONCLUSION

This paper presents different backup path rerouting schemes for multi-domain networks. The experiment results demonstrate that these rerouting schemes led to an economy of up to 11.5 % backup capacity and the dropping off of until 3% blocking in comparison to the original blocking of 8%-10%.

A regular (time-driven) rerouting helps to regularly free some capacity and thus reduce the blocking probability. However, it implies extra computational effort and information exchange in rerouted path computation and signaling. The choice of the rerouting frequency is a compromise between the scalability and the blocking probability. In comparison with LeastRRLocal in different rerouting frequencies, RRLocal-Block (that is blocking-driven) provides smaller blocking probability, requires less information exchanges and less computational effort. We suggest thus RRLocal-Block as an efficient and scalable solution for multi-domain networks.

REFERENCES

[1] S. Ramamurty and B. Mukherjee, "Survivable WDM Mesh Networks, Part I Protection," in *Proc. IEEE INFOCOM*, vol. 2, New York, NY, March 1999, pp. 744–751.

[2] B. Mukherjee, *Optical WDM Networks*. Springer, 2006.

[3] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed pre-configuration: ring-like speed with mesh-like capacity for self-plaining network restoration," in *Proc. IEEE ICC*, 1998, pp. 537–543.

[4] H. Huang and J. Copeland, "Multi-domain Mesh Optical Network Protection using Hamiltonian cycles," in *Proc. High Perf. Switching and Routing, Workshop on Merging Opt. and IP Technol.*, May 2002, pp. 26–29.

[5] Y. Xiong and L. Mason, "Restoration Strategies and Spare Capacity Requirements in Self Healing ATM Networks," *IEEE/ACM Trans. Netw.*, vol. 7, no. 1, pp. 98–110, Feb. 1999.

[6] B. G. Jozsa and D. Orincsay, "Shared backup path optimization in telecommunication networks," in *Proc. of Dynamic of Reliable Com. Netw., DRCN'2001*, Oct. 2001, pp. 251–257.

[7] M. Kodialam and T. Lakshman, "Dynamic Routing of Restorable Bandwidth-Guaranteed Tunnels Using Aggregated Network Resource Usage Information," *IEEE/ACM Trans. on Netw.*, vol. 11, no. 3, pp. 399–410, 2003.

[8] J. Tapolcai and T. Cinkler, "On-line Routing Algorithm with Shared Protection in WDM Networks," in *Proc. Opt. Netw. Design and Modeling*, Budapest, Hungary, Feb. 2003, pp. 351–364.

[9] D. Xu, C. Chunming, and Y.Xiong, "An Ultra-Fast Shared Path Protection Scheme Distributed Partial Information Management, Part II," in *Proc. 10th IEEE Intl. Conf. in Netw. Protocols*, France, Nov. 2002, pp. 344–353.

[10] S. Sengupta and R. Ramamurthy, "From Network Design for Dynamic Provisioning and Restoration in Optical Cross-Connect Mesh Networks: An Architectural and Algorithmic Overview," *IEEE Network*, vol. 15, no. 4, pp. 46–54, Jul.-Aug. 2001.

[11] J.-F. Labourdette, "Shared mesh restoration in optical networks," in *Proc. Opt. Fiber Com. Conf. (OFC)*, vol. 1, Feb. 2004, pp. 23–27.

[12] C. Assi, W. Huo, A. Shami, and N. Ghani, "Analysis of Capacity Re-provisioning in Optical Mesh Networks," *IEEE Com. Letters*, vol. 9, no. 7, pp. 658–660, Jul. 2005.

[13] D. Truong and B. Thiongane, "Dynamic routing for Shared Path Protection in Multidomain optical mesh networks," *OSA J. of Opt. Netw.*, vol. 5, no. 1, pp. 58–74, Jan. 2006.

[14] G. Bernstein, V. Sharma, and L. Ong, "Interdomain Optical routing," *OSA J. of Opt. Netw.*, vol. 1, no. 2, pp. 80–92, Feb. 2002.

[15] M. O'Mahony, D. Simeonidu, A. Yu, and J. Zhou, "The Design of the European Optical Network," *J. of Ligthwave Technol.*, vol. 13, no. 5, pp. 817–828, 1995.

[16] "REDIrid," 2005, http://www.rediris.es/red/index.en.html#red%20troncal.

[17] "Consortium GARR," http://www.garr.it.

[18] "RENATER-4 network," http://www.renater.fr.

[19] "Surfnet," http://www.surfnet.nl.