HAL
open science

Distributed PLR-Based Backup Path Computation in MPLS Networks

Mohand Yazid Saidi, Bernard Cousin, Jean-Louis Le Roux

To cite this version:

Mohand Yazid Saidi, Bernard Cousin, Jean-Louis Le Roux. Distributed PLR-Based Backup Path Computation in MPLS Networks. 7th International IFIP-TC6 Networking Conference, May 2008, Singapour, Singapore. pp.642-653, 10.1007/978-3-540-79549-0_56 . hal-01184150

HAL Id: hal-01184150

https://hal.science/hal-01184150v1

Submitted on 16 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed PLR-based Backup Path Computation in MPLS Networks

Mohand Yazid SAIDI*, Bernard COUSIN*, and Jean-Louis LE ROUX**

*Université de Rennes I, IRISA/INRIA, 35042 Rennes Cedex, France
**France Télécom, 2 Avenue Pierre Marzin, 22300 Lannion, France
msaidi@irisa.fr, bcousin@irisa.fr and jeanlouis.leroux@orange-ftgroup.com

Abstract. In this article, we provide mechanisms enabling the backup path computation to be performed on-line and locally by the Points of Local Repair (PLRs), in the context of the MPLS-TE fast reroute. To achieve a high degree of bandwidth sharing, the Backup Path Computation Entities (BPCEs), running on PLRs, require the knowledge and maintenance of a great quantity of bandwidth information (non aggregated link information or per path information) which is undesirable in distributed environments. To get around this problem, we propose a distributed PLR (Point of Local Repair)-based heuristic (DPLRH) which aggregates and noticeably decreases the size of the bandwidth information advertised in the network while maintaining the bandwidth sharing high. DPLRH is scalable, easy to be deployed and balances equitably computations on the network routers.
Simulations show that with the transmission of a small quantity of aggregated information per link, the ratio of rejected backup paths is low and close to the ideal.
Keywords– local protection; bandwidth sharing; path computation.

1 Introduction

The proactive protection of communication becomes increasingly important with the explosion of the number of network real time applications (voice over IP, network games, video on demand, etc). Thus, to ensure network service continuity upon a failure, the proactive protection techniques [1, 2] precompute and generally pre-establish backup paths capable to receive and reroute the traffic of the affected primary paths. Two schemes of protection exist: global (end-to-end) and local. With global scheme [2], each primary path is protected by one vertex (or link) disjoint backup path interconnecting the primary source and destination nodes. This protection scheme presents the disadvantage of increasing the recovery cycle since it requires that failure information reaches the source before the switching from the primary toward the backup path. This drawback is eliminated with the use of local protection where the recovery is achieved locally and without any control plane notification by the upstream node to the failing component.

With the advent of MPLS [3] in the last decade, the local protection is provided in efficient manner. In fact, MPLS offers a great flexibility for choosing

paths (called Label Switched Paths or LSPs) and thus, the backup paths can be determined so that bandwidth availability is maximized. Two types of backup LSP are defined for MPLS local protection [4]: Next HOP (NHOP) LSP and Next Next HOP (NNHOP) LSP. A NHOP LSP (resp. NNHOP LSP) is a backup path protecting against link failure (resp. node failure); it is setup between a primary node called Point of Local Repair (PLR) and one primary node downstream to the PLR (resp. to the PLR next-hop) called Merge Point (MP). Such backup LSP bypasses the link (resp. the node) downstream to the PLR on the primary LSP. When a link failure (resp. node failure) is detected by a node, this later activates locally all its NHOP and NNHOP (resp. its NNHOP) backup LSPs by switching traffic from the affected primary LSPs to their backup LSPs.

To ensure enough resource (particularly the bandwidth) after the recovery from a failure, the backup LSPs must reserve the resources they need beforehand. In this way, if we consider that each backup path has its own exclusive resources, the network will be overbooked rapidly since the available resources decrease quickly. Instead, with the the practical hypothesis of single failures, resource utilization can be improved by sharing the resources between some backup LSPs.

To increase the number of LSPs that can be setup in a network, the resource sharing should be taken into account when the backup LSPs are computed. Three functionalities are necessary to perform such computations in a distributed environment: information collection, information distribution and path determination. The first functionally gathers the structures and properties of the backup LSPs setup in the network. In practice, each network node stores the path links, the bandwidth and the risks protected by the backup LSPs traversing it. Such information can be obtained easily and without any additional overhead when the backup LSPs are signaled as in [4]. The second functionality reorganizes and transmits the collected information to nodes supporting the BPCEs (Backup Path Computation Entity). We note that for a same capability of bandwidth sharing, less the transmitted information is, better the functionality of distribution is. Finally, the last functionality searches for the backup LSPs providing the desired protection and verifying the bandwidth constraints.

In this article, we focus on the mechanisms allowing an efficient distribution of the bandwidth information and enabling the bandwidth guaranteed-backup LSP computation to be performed on-line and locally by the PLRs. Hence, we propose a new distributed PLR-based heuristic (DPLRH) aggregating and reducing significantly the size of the bandwidth information advertised in the network. With our heuristic, the backup LSPs are computed and configured by same nodes which correspond to the backup LSP head-end routers (PLRs). This eliminates the communication between the entities computing the backup LSPs (BPCEs) and those configuring the backup LSPs (PLRs). Besides, DPLRH is scalable, shares effectively bandwidth between the backup LSPs and capable to compute backup LSPs protecting against any type of failure risk.

The rest of this article is organized as follows: section 2 describes the three types of failure risks and gives the formulas allowing the computation of the minimal protection bandwidth to be reserved on each unidirectional link. In

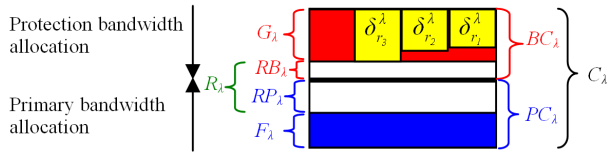

Fig. 1. Bandwidth allocation on an arc λ

section 3, we review some works related to the bandwidth sharing. Then, we explain in section 4 the principles of DPLRH. In section 5, we present simulation results and analysis. Finally, section 6 is dedicated to the conclusions.

2 Failure risks and bandwidth sharing

To deal with any single physical failure in a logical (MPLS) layer, three types of (logical) failure risks are defined: link, node and SRLG. The first type of failure risk corresponds to the risk of a logical link failure due to the breakdown of an exclusive physical component of the logical link. The second type of failure risk corresponds to the risk of a logical node failure. Finally, the third type of risk corresponds to failure risk of a common physical component (eg. optical fiber) shared by a group of logical links.

In order to ensure enough bandwidth upon a failure, minimal quantities of bandwidth must be reserved on links. To determine such quantities, we define the concept of *the protection cost* of a risk r on an arc λ (noted δ^λ_r). This later corresponds to the cumulative bandwidth of the backup LSPs which will be activated on the arc λ upon a failure of the risk r. For a SRLG risk *srlg* composed of links (l_1, l_2, .., l_n), δ^λ_{srlg} is determined as follows:

$$\delta^\lambda_{srlg} = \sum_{0<i\leq n} \delta^\lambda_{l_i} \tag{1}$$

To cope with any single failure, a minimal quantity of protection bandwidth G_λ must be reserved on the arc λ:

$$G_\lambda = Max_{r\in PFRG(\lambda)}\delta^\lambda_r \tag{2}$$

In order to control and specify the quantity of bandwidth dedicated for protection and to separate the task of primary LSP computation from that of backup LSP computation, the bandwidth capacity C_λ on arc λ can be divided in two pools: primary pool and protection pool (fig. 1). The primary pool has a capacity PC_λ and it is used to allocate bandwidth for primary LSPs. The protection pool has a capacity BC_λ and it is used to allocate bandwidth for backup LSPs.

To ensure the respect of bandwidth constraints upon a failure, the minimal protection bandwidth reserved on each arc λ must verify:

$$G_\lambda \leq BC_\lambda \tag{3}$$

To keep (3) valid after the setup of a backup LSP b of bandwidth *bw (b)* and protecting against the risks in *FR (b)*, only the arcs λ verifying the following inequality can be selected to be in the LSP b:

$$Max_{r \in FR(b)} \delta_r^\lambda \leq BC_\lambda - bw(b) \tag{4}$$

3 Related Works

Recently, a great deal of work is addressing the path protection in order to find algorithms and mechanisms allowing an on-line computation of the optimized backup paths. Several solutions are then proposed but a large number of them, like [5] and [6], copes only with failure risks of type link or node.

As the quality of the distributed techniques computing the backup paths depends closely on the algorithms implementing the functionality of information distribution (cf. Section 1), we will focus below on the study of these algorithms; for path computation, various variants of the Dijkstra's algorithm and ILP formulation can be applied.

In a first obvious approach [7], Kini proposes to flood within the IGP-TE protocols the topology information, the primary bandwidth, the capacities and all the protection costs of the risks on the topology arcs in the network. In this way, each node has a complete knowledge of the information necessary to the backup LSP and as a result, it can perform the backup LSP computation in an efficient manner. This distribution technique increases the bandwidth availability but it overloads the network with large and frequent messages advertising the protection costs. To scale well, [8] proposed the PCE-based MPLS-TE fast reroute technique in which no control message is necessary to compute the bandwidth-guaranteed backup LSPs. With this technique, a separate PCE (path computation element) is associated with each failure risk in order to compute the backup LSPs which will be activated at the failure of that risk. This computation technique is efficient when there are no SRLGs in the network. Otherwise, the PCE-based MPLS-TE fast reroute technique requires a mechanism distinguishing a node failure from a link failure. This increases significantly the recovery cycle of all the communications.

To offer scalability without increasing the recovery cycle, new computation heuristics which approximate and reduce the bandwidth information (protection costs especially) transmitted in the network have emerged[7, 9]. Hence, once the bandwidth information is collected, nodes aggregate it before its flooding in the network. For instance, in [7], Kini proposes to approximate the protection cost δ_r^λ of a risk r on a (unidirectional) link λ by the maximum of protection costs (G_λ) on that link. In this way, only one aggregated value per link is advertised in the network. This heuristic has the advantage of facility of its deployment. Indeed, this requires only slight modifications to IGP-TE protocols for the advertisement of the minimal quantities of protection bandwidth on links. However, this heuristic does not exploit efficiently the bandwidth sharing. As a result, the number of backup LSPs that can be built with this heuristic is low.

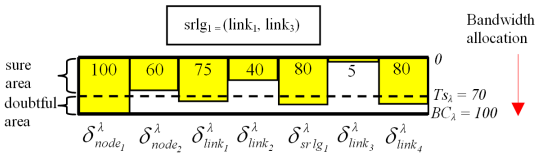

Fig. 2. Protection pool of an arc λ

4 Distribued PLR-based backup path computation heuristic (DPLRH)

4.1 DPLRH principles

DPLRH allows an efficient approximation of the protection costs on the links with the advertisement of a small quantity of aggregated protection bandwidth information. It is based on the two following principles:

- An arc λ can be used to establish a new backup LSP b requiring a quantity of bandwidth bw if and only if the protection costs of the risks protected by such LSP (on λ) are lower or equal to BC_λ - bw. As a result, the knowledge of the partial information consisting of the protection costs (and their corresponding risks) which are higher than BC_λ - bw is sufficient to decide without mistake if λ can be selected to be in the backup LSP b.
- Some values of protection cost on an arc can be very low. Aggregate and approximate these values by their maximum can decrease the quantity of protection information to be advertised in the network with slight or without deterioration of the bandwidth sharing.

To show how DPLRH exploits the two above principles, let us consider an example. In fig. 2, the protection pool of an arc λ is shown. For simplicity, we considered here that the capacity C_λ of each arc λ is divided in two disjoint pools (protection pool and primary pool). In this manner, the task which computes the backup LSPs can be dissociated from that determining the primary LSPs.

Algorithm 1 End node o_λ of an arc λ

Assign to the *new_x$_\lambda$_vector* associated to λ the x_λ *highest* protection costs and their corresponding risk identifiers
if the $(x+1)^{th}$ *highest* protection cost is higher than Ts_λ **then**
  Replace, in *new_x$_\lambda$_vector*, the identifier corresponding to the xth *highest* protection cost by the special identifier *generic_risk*
end if
if *new_x$_\lambda$_vector* $\neq$ *old_x$_\lambda$_vector* **then**
  Advertise *new_x$_\lambda$_vector*
end if
new_x$_\lambda$_vector $\leftarrow$ *old_x$_\lambda$_vector*

Algorithm 2 Each node receiving an x_λ_vector associated to λ

```
if generic_risk ∈ x_λ_vector then
    min_cost ← x_λ_vector[generic_risk]
else
    min_cost ← 0
end if
for each risk identifier id do
    if id ∈ x_λ_vector.index () then
        local_risk_protecion_cost[λ][id] ← x_λ_vector[id]
    else
        local_risk_protecion_cost[λ][id] ← min_cost
    end if
end for
```

As illustrated in fig. 2, the protection capacity BC_λ of the arc λ is equal to 100 units. This arc λ is used in the protection of seven risks: $node_1$, $node_2$, $link_1$, $link_2$, $link_3$, $link_4$ and $srlg_1$. The protection costs associated to these risks are as follows: $\delta^\lambda_{node_1} = 100$, $\delta^\lambda_{node_2} = 60$, $\delta^\lambda_{link_1} = 75$, $\delta^\lambda_{link_2} = 40$, $\delta^\lambda_{link_3} = 5$, $\delta^\lambda_{srlg_1} = 80$, $\delta^\lambda_{link_4} = 80$.

When the maximal quantity of bandwidth max_{bw} ($max_{bw} = 30$) that a LSP can request is known, the use of the *principle 1* of DPLRH permits to deduce that all the risks whose protection costs are lower or equal to the threshold Ts_λ ($Ts_\lambda = BC_\lambda - max_{bw}$) can be ignored (approximated by zero) when a new backup LSP is computed. In fact, the selection of the arc λ to be in a new backup LSP of bandwidth bw does not cause the violation of bandwidth constraints upon a failure of a risk r if the protection cost δ^λ_r is lower or equal to the threshold Ts_λ. This results from the following inequalities:

$$\begin{cases} \delta^\lambda_r \leq Ts_\lambda = BC_\lambda - max_{bw} \\ bw - max_{bw} \leq 0 \end{cases} \Rightarrow \delta^\lambda_r + bw \leq BC_\lambda$$

Obviously, the elimination of the protection costs, which are lower or equal to the threshold Ts_λ ($Ts_\lambda = BC_\lambda - max_{bw} = 70$) from the information to be advertised in the network, does not alter the decision of excluding (or including) the arc λ in a next backup LSP computation. Typically, in fig. 2, the outgoing node o_λ to the arc λ, which is responsible[1] of the advertisement of the protection costs $\{\delta^\lambda_r\}_r$ on the arc λ, approximates the protection costs of $node_2$, $link_2$ and $link_3$ by zero. As a result, these risks ($node_2$, $link_2$ and $link_3$) and their corresponding protection costs on the arc λ are not advertised in the network.

When the value max_{bw} is high (or ignored by the nodes of the network), the quantity of bandwidth information advertised for each arc of the topology can be high and unacceptable. To avoid the flooding of the network while maintaining bandwidth sharing high, DPLRH limits the size of the protection bandwidth information that is advertised for each arc λ to a vector (called x_λ_vector) composed of x_λ elements. Each x_λ_vector component includes a couple of protection

[1] The end nodes of each arc λ know all the values of protection cost on this arc λ.

cost and its associated risk. Besides, the costs conveyed in the x_λ*_vector* of an arc λ correspond to the x_λ highest values of protection cost. In this manner, each node receiving an x_λ*_vector* of an arc λ can deduce the x_λ highest protection costs of the risks using λ for protection and it approximates all the rest of protection costs by the $(x_\lambda)^{th}$ highest protection cost (principle 2 of DPLRH). For instance, if we consider that x_λ is equal to 2 in fig. 2, the outgoing node o_λ to the arc λ will send the following x_λ*_vector*: [($node_1$, 100), (*generic_risk*, 80)] (where *generic_risk* refers to any risk different from those conveyed in the x_λ*_vector*).

4.2 DPLRH algorithm description

With the combination of DPLRH's principles 1 and 2, we obtain the algorithms Alg. 1 and Alg. 2 which specify the steps of the protection cost advertisement and collection. Thus, Alg. 1 describes the procedure of protection cost advertisement which limits the transmitted bandwidth information for each arc λ, to an x_λ*_vector*. This vector contains, at most, the x_λ highest protection costs which are larger than the threshold Ts_λ. Concerning Alg. 2, it specifies the procedures used to approximate the protection costs from the received x_λ*_vector* information.

In order to increase the bandwidth sharing and to reduce the size of messages transmitting the x_λ*_vectors*, each node running Alg. 1, eliminates from its protection cost table all the entries corresponding to the risks which are included in others. In fig. 2 for instance, the SRLG risk $srlg_1$ includes two link risks: $link_1$ and $link_3$. As a result, these two last risks ($link_1$ and $link_3$) and their corresponding protection costs are deleted from all the protection cost tables before any advertisement of the x_λ*_vectors*. After this step, the outgoing node o_λ to each arc λ deduces the x_λ*_vector* containing (at max) the *x highest* protection costs on λ and their corresponding risks. For the arc in fig. 2, the corresponding x_λ*_vector* is: [($node_1$, *100*), ($srlg_1$, *80*), ($link_4$, *80*), ($node_2$, *60*), ($link_2$, *40*)].

At each change of the x_λ*_vector* corresponding to λ, node o_λ advertises its new value in the network.

When a node (different from the end nodes of λ) receives an x_λ*_vector* corresponding to an arc λ, it runs the routine shown in Alg. 2 to approximate the protection costs on the arc λ. According to the threshold value (Ts_λ) and the $(x_\lambda + 1)^{th}$ highest protection cost (denoted by x_λ*_plus_1_cost*) on the arc λ, the DPLRH's decision of excluding the arc λ in the next backup LSP computation can be "sure and correct" or "possibly wrong". Two areas are defined to measure the correctness degree of the protection cost approximation used by DPLRH: *doubtful zone* (x_λ*_plus_1_cost* $> Ts_\lambda$) and *sure zone* (x_λ*_plus_1_cost* $\leq Ts_\lambda$).

Doubtful area (x_λ*_plus_1_cost* $> Ts_\lambda$)
When the $(x_\lambda + 1)^{th}$ highest protection cost on an arc λ is in the doubtful area, the advertisement of an x_λ*_vector* whose size is equal to x_λ can be insufficient to decide without mistake if the arc λ can be selected to be in a new backup LSP.

In fig. 2 for instance, the $(x_\lambda + 1)^{th}$ highest protection cost on the arc λ is in the doubtful area if $x_\lambda \leq 2$. Thus, the advertisement of an x_λ*_vector* containing at most the two highest protection costs on λ can be insufficient. Typically, with $x_\lambda = 2$, the outgoing node o_λ to the arc λ advertises in the network this x_λ*_vector*: [($node_1$, *100*), (*generic_risk*, *80*)].

When a node n receives the x_λ*_vector* transmitted by o_λ, it updates its protection cost table by approximating the protection costs on the arc λ as follows (Alg. 2):

$$\begin{cases} \delta^\lambda_{node_1} = 100 \\ \forall r(r \ is \ a \ risk \wedge r \neq node_1) : \delta^\lambda_r = 80 \end{cases}$$

As we see here, all the risks whose protection costs on the arc λ are not transmitted within the x_λ*_vector*, are aggregated and approximated by $(x_\lambda)^{th}$ highest protection cost on this arc. As this last cost is higher than the threshold, any computation of a new backup LSP, protecting against a risk which is not conveyed in the transmitted x_λ*_vector* of λ and claiming a quantity of bandwidth *bw* ($bw > BC_\lambda - x_\lambda$*_plus_*1*_cost*), excludes by mistakes the arc λ. However, any other computation will include or exclude the arc λ without mistake.

In fig. 2 for instance, after the reception by node n of this x_λ*_vector* (x_λ*_vector* = [($node_1$, *100*), (*generic_risk*, *80*)]), node n excludes by mistake the arc λ in its next computation of a backup LSP if this last one requires a quantity of bandwidth larger than 20 ($20 = Ts_\lambda - 80$) and protects against failure risks which do not belong to $node_1$, $srlg_1$, $link_4$; otherwise, the arc λ is included or excluded without mistake.

Sure area (x_λ*_plus_*1*_cost* $\leq Ts_\lambda$)

When the $(x_\lambda + 1)^{th}$ highest protection cost on an arc λ is in the sure area, the advertisement of an x_λ*_vector* whose size is equal to x_λ is sufficient to decide without mistake if the arc λ can be selected (or not) to be in a new backup LSP (principle 1 of DPLRH).

In fig. 2 for instance, the $(x_\lambda + 1)^{th}$ highest protection cost on the arc λ is in the sure area when $x_\lambda > 2$. Thus, the advertisement of an x_λ*_vector* containing at least the three highest protection costs on λ is sufficient. Typically, with the choosing of $x_\lambda > 2$, the outgoing node o_λ to the arc λ transmits an x_λ*_vector* including (at most) the three highest protection costs which are greater than the threshold $Ts_\lambda = 70$. This x_λ*_vector* corresponds to: [($node_1$, *100*), ($srlg_1$, *80*), ($link_4$, *80*)].

When a node n receives the x_λ*_vector* transmitted by o_λ, it updates its protection cost table by approximating the protection costs on the arc λ as follows (Alg. 2):

$$\begin{cases} \delta^\lambda_{node_1} = 100 \wedge \delta^\lambda_{srlg_1} = 80 \wedge \delta^\lambda_{link_4} = 80 \\ \forall r(r \ is \ a \ risk \wedge r \neq node_1 \wedge r \neq srlg_1 \wedge r \neq link_4) : \delta^\lambda_r = 0 \end{cases}$$

Contrarily to the case x_λ*_plus_*1*_cost* $> Ts_\lambda$ where the protection costs of the risks which are not conveyed in the advertised x_λ*_vector* are approximated by

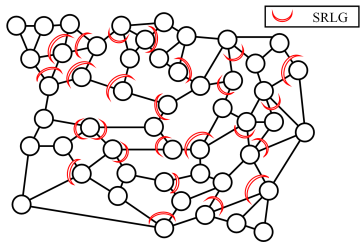

Fig. 3. Test network

the $(x_\lambda)^{th}$ highest protection cost on the arc λ, in the sure area these protection costs are aggregated and approximated by zero. As explained in the previous section, this approximation does not alter the decision of including or excluding the arc λ in the next computation of a backup LSP b. Indeed, if the new backup LSP b protects against the failure of a risk belonging to $node_1$, $srlg_1$, $link_4$, node n can deduce easily the exact value of the highest protection cost on λ of the risks protected by b. This highest protection cost corresponds to 100 units if b protects against the failure of $node_1$, 80 units otherwise. As a result, node n decides without mistake if the arc λ can be selected to be in b or not (formula (4) in section 2). When the new backup LSP b is planned to protect against the failure risks which do not belong to $node_1$, $srlg_1$, $link_4$, node n selects the arc λ, without risk of mistake, when it computes the backup LSP b (cf. *principle 1* of DPLRH in section 4.1).

5 ANALYSIS AND SIMULATION RESULTS

5.1 Simulation model

We evaluated the performance of our proposed heuristic $DPLRH$ by comparing it to the Kini's flooding-based algorithm (FBA) and heuristic (FBH) described in section 3. Our choice is motivated by two reasons:

- With the Kini's algorithm, all the protection costs are advertised in the network. In this way, the backup LSP computations could be performed efficiently and without any bandwidth waste (due to approximations). Although FBA is not practical (it floods the network), we used it in our simulation to measure the approximation quality of the compared heuristics.
- With Kini's heuristic, only the highest protection cost on a link is advertised in the network. Contrarily to FBA, FBH is practical and it was used in several backup LSP computation algorithms (like in [5, 6]).

For our simulations, we used the topology network depicted in fig. 3 and composed of 162 risks: 50 nodes, 87 bidirectional links and 25 SRLGs (represented by ellipses in fig. 3). The capacity bandwidth on each arc was separated

in two pools: protection pool of capacity equal to 100 units and primary pool of infinite capacity. The traffic matrix is generated randomly and consists of LSPs arriving one by one and asking for quantities of bandwidth uniformly distributed between 1 and 10. The ingress and egress nodes of each primary LSP are chosen randomly among the network nodes. Both the primary and backup LSP computations were based on the Dijkstra's algorithm.

Four variants of $DPLRH$ were used in simulations. The first one $DPLRH_{(\infty,90)}$ used a threshold ($\forall\lambda : Ts_\lambda = 90$) and an infinite x_λ*-vector* ($\forall\lambda : x_\lambda = \infty$) on all the arcs. The second variant $DPLRH_{(2,0)}$ used an x_λ*-vector* of maximal size equal to 2 ($\forall\lambda : x_\lambda = 2$) but it does not employed the threshold ($\forall\lambda : Ts_\lambda = 0$). The third variant $DPLRH_{(5,0)}$ used an x_λ*-vector* of maximal size equal to 5 ($\forall\lambda : x_\lambda = 5$) and a null threshold ($\forall\lambda : Ts_\lambda = 0$). Finally, the last variant $DPLRH_{(5,90)}$ uses a threshold ($\forall\lambda : Ts_\lambda = 90$) and an x_λ*-vector* of maximal size equal to 5 ($\forall\lambda : x_\lambda = 5$).

Two metrics are used for the comparison: ratio of rejected backup LSPs (*RRL*) and mean number of parameter changes per backup LSP (*NPC*).

The first metric measures the ratio of backup LSP requests that are rejected because of the lack of protection bandwidth. It is computed as the ratio between the number of backup LSP requests that are rejected and the total number of backup LSP requests. The second metric measures the mean number of changes in the bandwidth protection parameters which allow the approximation of the different protection costs. It is computed as the ratio between the number of changes in the protection bandwidth parameters and the number of backup LSPs built in the network. For $DPLRH$ (resp. FBA), each change of the x_λ*-vectors* (resp. of any protection cost) increases the *NPC* whereas only the changes in the minimal protection bandwidth allocated on arcs ($\{G_\lambda\}_\lambda$) are counted with FBH.

At each establishment of 20 primary LSPs, the two metrics *RRL* and *NPC* are computed for each algorithm or heuristic. We note that our simulation results correspond to the average values of 1000 runs generated randomly.

5.2 Results and analysis

Fig. 4 depicts the evolution of *RRL* as a function of the number of primary LSPs setup in the network. As expected, FBA and $DPLRH_{(\infty,90)}$ have *RRL* values better than those of $DPLRH_{(5,90)}$, $DPLRH_{(5,0)}$ and $DPLRH_{(2,0)}$ which, in turn, have better *RRL* values than those of FBH. This is due to the complete knowledge of the protection costs, which are necessary for the backup LSP computation, with FBA and $DPLRH_{(\infty,90)}$ whereas only partial information consisting of the *5* (resp. *5*, *2*, *1*) highest protection costs is used for the backup LSP computation with $DPLRH_{(5,90)}$ (resp. $DPLRH_{(5,0)}$, $DPLRH_{(2,0)}$, FBH).

Due to the localization of the backup LSPs which tend to traverse arcs close to the protected risks, the probability $P\ (\delta^r_\lambda > Ts_\lambda)$ is close to zero when the distance between the risk r (its components) and the arc λ is high. As a result, the number of protection costs which are higher than the threshold on the arc λ (i.e. Protection costs required for the backup LSP computation in order to

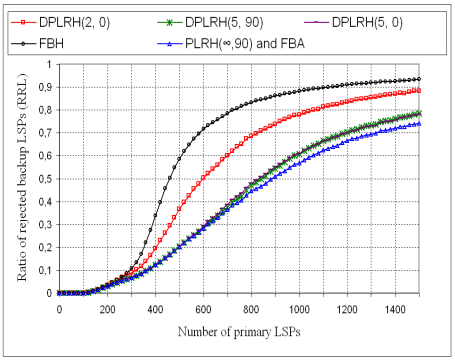

Fig. 4. Ratio of rejected backup LSPs (RRL)

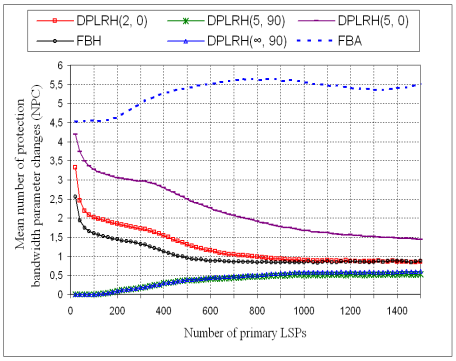

Fig. 5. Mean number of messages sent in the network per backup LSP (NPC)

avoid the rejection by mistake of λ) depends strongly from the number of risks located in the close neighborhood of λ. Typically, in the network of fig. 3 where the mean degree of nodes is *3.5* and where the number of SRLGs is equal to *25*, the advertisement of the *5* highest protection costs seems to be sufficient to avoid the blocking by mistake of protection requests when the number of primary LSPs is lower than *700*. When the number of primary LSPs is higher than *700*, the use of an x_λ*-vector* of size equal to *5* results in the rejection par mistake of some protection requests but globally, the corresponding *RRL* is a very close to the ideal case (case where all the protection costs are systematically advertised).

Concerning the second metric, fig. 5 shows that FBA has higher *NPC* values than those of $DPLRH$ and FBH. This is due to the systematical advertisement of protection costs used with FBA. In fact, at each establishment of a new backup LSP, FBA advertises the new values of protection costs (on the backup LSP links) whereas only changes in the x highest protection costs which are larger than the threshold (resp. *1* highest protection cost) on the backup LSP links require new x_λ*-vector* advertisements with $DPLRH$ (resp. require the advertisement of the new highest protection costs with FBH).

Concerning the comparison between the *NPC* values of $DPLRH$ and FBH, we observe in fig. 5 that the *NPC* values of $DPLRH_{(\infty,90)}$ and $DPLRH_{(5,90)}$ are lower than those of FBH, $DPLRH_{(2,0)}$ and $DPLRH_{(5,0)}$. This comes from the use, in $DPLRH_{(\infty,90)}$ and $DPLRH_{(5,90)}$, of a high threshold Ts_λ ($Ts_\lambda/BC_\lambda = 0.9$) which eliminates the flooding of a large number of x_λ*-vectors*. Thus, when the threshold value is known and high, setting x_λ to the infinity could be a very promising choice which decreases the blocking probability without the flooding of the network.

The other important observation, in fig. 5, is related to the similarity between the *NPC* values of $DPLRH_{(\infty,90)}$ and those of $DPLRH_(5, 90)$. This means that for such network load, the x_λ*-vectors* transmitted with $DPLRH_{(\infty,90)}$ are nearly the same as those advertised with $DPLRH_{(5,90)}$. This explains the similarity between the *RRL* values of $DPLRH_{(\infty,90)}$ and $DPLRH_{(5,90)}$ in fig. 4.

6 Conclusion

In this article, we proposed a completely distributed heuristic, called DPLRH, for backup LSP computation. Our heuristic allows a high-quality approximation of the protection information necessary for the backup LSP computation with the advertisement of small and size-limited vector (x_λ_vector) per network arc λ. This x_λ_vector is formed of the x_λ ($x_\lambda > 0$) highest protection costs which are higher than a threshold Ts_λ ($Ts_\lambda \geq 0$), and does not change at each backup LSP establishment.

DPLRH has several advantages. Firstly, it is symmetrical and it balances equitably computations on the network nodes. Secondly, DPLRH does not involve a communication between the entities computing the backup LSP and those configuring them since such tasks are performed by the same nodes (PLRs). Thirdly, DPLRH is scalable and it reaches a high degree of bandwidth sharing with the advertisement of a limited quantity of protection information (x_λ_vector information). Finally, our heuristic is easy to be deployed since it requires only very slight extensions to the IGP-TE protocols.

Simulation results show that DPLRH decreases significantly the number of rejected backup LSPs and the frequency of advertisements when the threshold and the highest sizes of $x_\lambda_vectors$ are well chosen.

References

1. Meyer, P., Van Den Bosch, S., Degrande, N.: High Availability in MPLS-based Networks. Alcatel telecommunication review, Alcatel (4th Quarter 2004)
2. Ramamurthy, S., Mukherjee, B.: Survivable WDM Mesh Networks (Part 1 - Protection). In IEEE INFOCOM **2** (1999) 744–751
3. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC 3031 (January 2001)
4. Pan, P., Swallow, G., Atlas, A.: Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090 (May 2005)
5. Kodialam, M.S., Lakshman, T.V.: Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels using Aggregated Link Usage Information. In IEEE INFOCOM (2001) 376–385
6. Balon, S., Mélon, L., Leduc, G.: A Scalable and Decentralized Fast-Rerouting Scheme with Efficient Bandwidth Sharing. Computer Networks **50**(16) (November 2006) 3043–3063
7. Kini, S., Kodialam, K., Lakshman, T.V., Sengupta, S., Villamizar, C.: Shared Backup Label Switched Path Restoration. Internet Draft draft-kini-restoration-shared-backup-01.txt, IETF (May 2001)
8. Vasseur, J.P., Charny, A., Le Faucheur, F., Achirica, J., Le Roux, J.L.: Framework for PCE-based MPLS-TE Fast Reroute Backup Path Computation. Internet Draft draft-leroux-pce-backup-comp-frwk-00.txt, IETF (July 2004)
9. Saidi, M.Y., Cousin, B., Le Roux, J.L.: A Distributed Bandwidth Sharing Heuristic for Backup LSP Computation. In IEEE GlobeCom (November 2007)