Improved MPLS-TE LSP Path Computation using Preemption

Imène Chaieb, Jean-Louis Le Roux, Bernard Cousin

Improved MPLS-TE LSP Path Computation using Preemption

Imène Chaieb and Jean-Louis Le Roux
France Telecom R&D, 2 Avenue Pierre Marzin
Lannion 22307, France
Email: {imene.chaieb, jeanlouis.leroux}@orange-ftgroup.com

Bernard Cousin
IRISA, université de Rennes 1
Rennes 35042, France
Email: bernard.cousin@irisa.fr

Abstract— **In Multi-Protocol Label Switching-Traffic Engineering (MPLS-TE) networks with distributed tunnel path computation on head-end routers, tunnel requests are handled one by one, in an uncoordinated manner without any knowledge of future and other requests. The order in which requests are handled has a significant impact on the network optimization and blocking probability. If it is not possible to control the arrival order, in return it is possible, in some cases, to reorder requests using the preemption function. This paper evaluates the impact of the arrival order, so as to determine efficient orders. It then proposes two preemption strategies so as to reorder arrivals and evaluate these strategies applied to the shortest constrained path computation algorithm.**

I. INTRODUCTION

Traffic Engineering (TE) is required to optimize network resources utilization, that is to maximize the amount of traffic that can be transported in a network, while ensuring the quality of service (QoS). Various TE mechanisms have been proposed for packet networks, among those MPLS-TE, a connection oriented mechanism based on the MPLS forwarding paradigm, well suited to TE thanks to its Explicit Routing capabilities. The MPLS-TE approach [1] allows setting up explicitly routed Traffic Engineering-Label Switched Paths (TE-LSP) whose that satisfy a set of traffic engineering constraints, including bandwidth and delay. MPLS-TE combines explicit routing capabilities of MPLS with a Constraint Based Routing (CBR) mechanism that lies in dynamic resources discovery (ISIS-TE [2], OSPF-TE [3]), constrained path computation, and distributed LSP signalling with resources reservation (RSVP-TE) [4]. MPLS-TE ensures Traffic Engineering functions such as network resources optimization, strict QoS delivery, and fast recovery upon link or node failures. We distinguish today two main approaches for the deployment of TE-LSPs in Service Provider networks: The *tactical* approach and the *strategic* approach. The *tactical* approach consists in the deployment of a few TE-LSPs, so as to bypass some congested network segments. The *strategic* approach consists in meshing edge routers with a set of TE-LSPs. These TE-LSPs are used to carry all traffic or some specific traffic classes between edge routers. There are various MPLS-TE routing options, which differ with the location of path computation elements (distributed on edge routers or centralized on a server), the path computation time scale (offline or online) and the level of coordination (paths can be computed either one by one

independently of each other, or in a coordinated manner). Among those, the option mainly deployed today by operators is the Online Uncoordinated Distributed one [5], where the LSP setup requests are handled one by one by the edge routers in an uncoordinated manner. This approach offers better scalability, reactivity and robustness than the Offline Coordinated Centralized mode where a server is in charge of computing all the LSPs in a coordinated manner without any time limitation. In return, with the uncoordinated approach, an edge router does not have a global knowledge of all LSPs established by other edge routers and of future LSP requests, and hence the performances in terms of optimality are affected. In some cases this mode even fails to find a path for all requests while there is a feasible solution. A basic uncoordinated path computation algorithm implemented today in most of routers relies on a modified Dijkstra algorithm [6]: links that do not support the constraints are pruned from the topology and the Dijkstra SPF algorithm is run on the resulting topology. This algorithm also referred to as CSPF (Constrained Shortest Path First) is by nature heavily greedy, and rapidly leads to blocking issues. In order to overcome these CSPF limitations, a set of solutions have been proposed in the literature that reduces the blocking probability and achieves better resources optimization while keeping agility characteristics (robustness, scalability and reactivity) of the uncoordinated scheme. These solutions try to find the best weight or cost function to be used by the routing algorithm in order to minimize congestion. This includes, non exhaustively: the Widest Shortest Path Algorithm (WSP) [7], the Shortest Widest Path Algorithm (SWP) [8], and the MIRA (Minimum Interference Routing Algorithm) algorithm [9]. The key idea in MIRA consists of avoiding links that could interfere with potential future path requests so as to reduce the blocking probability of LSP requests between a specified set of ingress-egress pairs. However, MIRA is computationally expensive. Another online algorithm, called Dynamic Online Routing Algorithm (DORA) [10], is inspired from MIRA ideas and is computationally less expensive than MIRA. A Profile Based Routing (PBR) algorithm [11] was also proposed, which uses measurement based "traffic profiles" or service level agreements (SLAs) as a rough predictor for future traffic distribution. PBR is the most expensive one in terms of the information maintained.

Actually in the uncoordinated mode, the arrival order of the

LSP requests is critical, two distinct orders are likely to provide distinct blocking results and this is actually a major characteristic of all uncoordinated algorithms including those listed above. In this paper, we focus on the LSP request arrival order. We firstly evaluate in section II the impact of the LSP arrival order on the resource usage optimization and blocking probability and we try to identify relevant orders. In section III, we give the architectural constraints that must be respected when extending an Online Distributed Uncoordinated Routing mode. In section IV, we firstly describe the MPLS-TE preemption mechanism that allows a new LSP to delete an existing LSP which is rerouted on an alternate path, and hence can be used as a solution to reorder the LSP arrival. Then, we propose two preemption strategies to control the reordering of LSP setup, applied to the CSPF algorithm. Finally, the section V provides the evaluation of these strategies in terms of link utilization and blocking probability, when applied to CSPF algorithm.

II. IMPACT OF LSP SETUP ARRIVAL ORDER ON ROUTING PERFORMANCES

As previously discussed, in the uncoordinated MPLS-TE mode the LSP requests are computed one-by-one without any knowledge of other requests and future requests. This can lead to a sub-optimal solution, where some LSP requests can be rejected even if there is a feasible placement.

We illustrate this with a simple example. Consider the network in Figure 1. Each link is characterized by its metric (1 unit for all links) and its capacity (in Mbps). Six LSP setup requests with bandwidth size BW (in Mbps), as shown in table I, arrive at node 1 one-by-one with node 6 as destination.

LSP	L1	L2	L3	L4	L5	L6
Bandwidth (BW)	30	25	35	16	17	20

TABLE I

ILLUSTRATION: THE BANDWIDTH OF LSP REQUESTS

The Fig. 2 shows the number of rejected LSPs on the network for all possible permutations of these 6 requests, according to their bandwidth size. LSPs path computation is done using the CSPF algorithm. We see that the increasing order belongs to the optimal order set. It does not reject any LSP request. However, the non ordered case (where LSP requests are established according to their order in the table I) and the decreasing order reject one LSP request. This result can be explained as follows; since in the decreasing case, the large LSPs are established first, they may block the resources on shortest and non shortest paths. So, some of the small LSPs which arrive after may be rejected. However, in the increasing case, small LSPs are established at first. So, there is more accepted LSP requests than in the decreasing case, before reaching congestion case.

This example illustrates the limitations of the online uncoordinated distributed routing scheme in terms of routing performances. It shows the impact of the LSP requests setup order

on the routing performances; we notice that with the same topology, the same requests and the same routing algorithm (CSPF) the routing performance varies with the LSP arrival order. In this specific example, there are 6 requests, so there are 720 (6!) possible orders. The computation of the optimal order, that is the order that allows minimizing the number of rejected requests, requires knowing all orders, and this has an exponential complexity; hence it can not be performed.

Hence controlling the LSP setup order would allow improving the performances of the distributed uncoordinated scheme. It is not possible to control the LSP arrival order. In return it is possible in some situations, to reorder LSP setup with the MPLS-TE preemption mechanism described in more details latter.
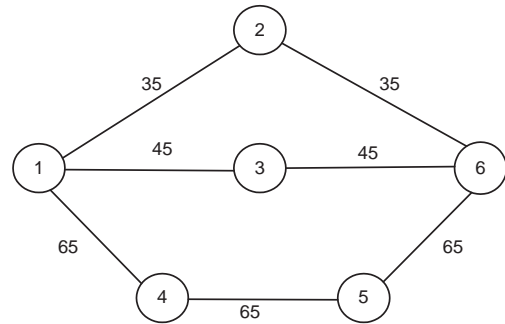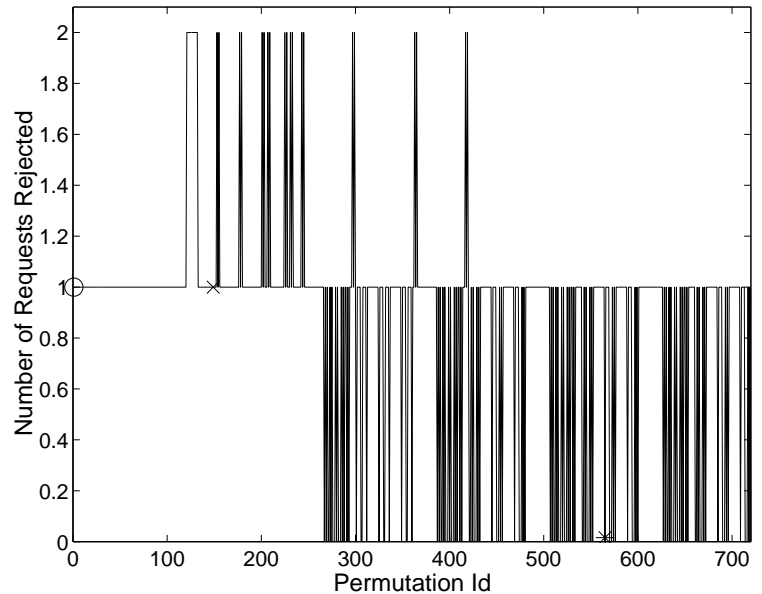

Fig. 1. Illustrative Example

Fig. 2. Illustration: Number of rejected LSPs For All permutations Requests (o : Non ordered - * : Increasing - x : Decreasing)

III. ONLINE/DISTRIBUTED/UNCOORDINATED ONLINE SYSTEM CONSTRAINTS

We have examined in the previous section the limitations of the Online Distributed Uncoordinated Routing approach

in terms of optimality. In order to keep the agility of this approach, the procedures that can be proposed to improve its performances should respect the following guidelines:

- Edge routers do not have knowledge of the LSPs established by the others edge routers in the network. They have only knowledge of the available bandwidth on each link.
- Edge routers do not have knowledge of the future LSP setup requests.
- The proposed mechanism must no impact by any mean the scalability of the IGP-TE protocol. Particularly, extensions for the advertisement of non aggregated parameters, such as for instance the advertisement of other TE-LSP so that an edge router starts to be aware of all LSPs in the networks, are not allowed. However, IGP-TE extensions which do not impact the scalability of the IGP are allowed, such as for instance, the advertisement of aggregate parameters with a controlled update frequency.
- The proposed mechanisms must not impact by any mean the scalability of the RSPV-TE protocol. Particular care should be given on minimizing the amount and rate of RSVP-TE messages.

IV. DYNAMIC REORDERING OF LSPs USING PREEMPTION

Once an efficient order is found, a solution to improve the optimality consists in trying to dynamically reorder LSPs, while respecting constraints of an Online Distributed Uncoordinated system. Our approach consists in investigating the use of MPLS-TE *preemption* so as to dynamically reorder LSPs setup.

Preemption mechanism in MPLS system

The RSVP-TE protocol [4] includes a preemption mechanism that allows an LSP with a given priority to preempt an LSP with a lower priority. The lower priority LSP is rerouted on an alternate path and all happens as if the lower priority LSP had been setup after the higher priority LSP. The RSVP-TE protocol allows specifying two priority attributes: the setup priority that specifies the capability of a LSP to pre-empt another LSP and the holding priority that specifies the capability of an LSP to resist to preemption. Both priorities have a range of 0 (highest priority) to 7 (lowest priority). An LSP with higher (numerically lower) setup priority can preempt an LSP with lower (numerically higher) holding priority. To avoid continuous preemption and oscillations, the holding priority should never be lower (numerically higher) than the setup priority. The IGP-TE advertises different "Unreserved Bandwidth" information for each priority level. So, to compute the route for an LSP with priority p, only the unreserved bandwidth for priority p has to be checked. Thus, available bandwidth is checked by considering only the LSPs with same or higher priority and as if LSPs with lower priority did not exist.

There are two types of preemption: hard preemption and soft preemption. In the hard preemption process, lower priority TE LSPs are pre-empted by higher priority LSP and rerouted in a break before make manner that is, the lower priority TE LSP is torn down and then restored on an alternate path if such a path exists, and this imply connectivity loss until the LSP is rerouted. [12]. In the soft preemption process, the point of preemption allows a brief coexistence of the two LSPs so as to let time for the lower priority LSP to be rerouted in a make before break manner, by its Ingress LSR, once notified about the ongoing preemption. In soft preemption, the new TE LSP is established before the old LSP is torn down [12]. So, the traffic is disrupted but there may be congestion during the coexistence phase.

The preemption mechanisms can be used to ensure that mission critical traffic trunks (e.g. VoIP) can always be routed through relatively favorable paths (e.g. shortest path) and can pre-empt best effort services (e.g Internet data) upon congestion or failure event. In [13], the authors propose a flexible policy to achieve various objectives when selecting the set of LSPs to be preempted. Preemption can also be used so as to dynamically reorder the LSP setup, indeed a higher priority LSP is routed as if the lower LSP did not exist. A solution to enforce a specific LSP setup order consists of assigning priorities to LSPs based on their bandwidth. For instance, if we want to apply an increasing bandwidth LSP setup order, low bandwidth LSPs should have a higher priority than high bandwidth LSPs.

As there are N LSP setup requests and only 8 priorities ($8 << N$), we cannot assign a different priority to each LSP, and hence we cannot apply an exact order (this would require N priorities). So, the problem which arises now is how to allocate the 8 priority to the N LSPs. We need to find an efficient way to allocate a priority to each LSP.

We propose here two methods: the Linear Repartition (LR) and the Non-linear Repartition (NLR).

A. Linear Repartition (LR)

This technique consists of the following steps:

- Sort the LSP requests (e.g. in increasing order)
- Equally divide the scale of the requests's bandwidth in 8 intervals, the width of each interval is $B_i = (B_{max} - B_{min})/8$ where:
  B_{max}: The bandwidth of the largest LSP demand.
  B_{min}: The bandwidth of the smallest LSP demand.
  and $0 <= i <= 7$.

- Assign a priority $Prio$ to each interval: Assign the same priority to all LSPs whose size belongs to the same interval. If an increasing order is required then:
  $Bw(lsp) \in [B_i, B_{i+1}] => Prio(lsp) = i$
  If a decreasing order is required then:
  $Bw(lsp) \in [B_i, B_{i+1}] => Prio(lsp) = 7 - i$
  Note that LSPs within the same bandwidth interval have the same priority and cannot be ordered.

With this approach, LSPs are not equally spread among all priorities; there may be a lot of LSPs with same priority,

which can not be ordered.

B. Non-Linear Repartition (NLR)

The first approach may lead to an unequal repartition of LSPs between the 8 intervals. Since, we investigate a second approach which takes into account the number of LSPs per priority level. It assigns to each set of $N/8$ LSPs the same priority (N is the number of requests). It proceeds as follow:

- Sort the LSP requests (e.g. in increasing order)
- Divide the scale of requests in 8 intervals, each interval B_i includes $n = N/8$ LSPs.
- Assign a priority $Prio$ to each interval: Allocate the priority $Prio$ to all n LSPs within this interval.

Priority configuration

In order to apply the preemption mechanism, LSP priorities should be known on the Ingress LSRs. There are two options: Priorities may be determined on a TE server and then configured on the Ingress routers, or they may be dynamically computed on the Ingress routers. In the LR case, the network administrator can determine a lower bound for Bmin and an upper bound for Bmax, which are then configured on all Ingress routers. Ingress routers can dynamically apply a priority to an LSP according to its bandwidth, following the LR formula. This allows for dynamic LSP bandwidth modification on Ingress LSRs that adapt LSP priorities accordingly. In the NLR case, priority allocation requires knowledge of all LSPs and their bandwidth and hence cannot be performed on Ingress routers (see section III). In this case priorities must be allocated by the TE server that then configures LSPs with theirs priorities on Ingress LSRs. This approach does not allow dynamic LSP bandwidth modification on the Ingress LSR, as the Ingress LSR has not enough information to modify the LSP priority accordingly. In a nutshell, the LR approach is well suited to an online distributed mode while the NLR approach better fits in with an offline centralized approach.

V. EVALUATION

In this section, we numerically evaluate our approach. All the simulations shown in the remainder of the paper are carried out by using the network topology that was proposed in [9], see Fig. 3. This topology includes 15 nodes and 28 bidirectional links. The capacity of the light links is $12 * 100$ units and that of the dark links is $48 * 100$ units (taken to model the capacity ratio of OC-12 and OC-48 links and scaled by 100). We show the performances of our approach using the Constrained Shortest Path First algorithm (CSPF). All experiments are made under the following assumptions:

- We assume that all LSPs are long lived ("static" case).
- We construct a full mesh of LSPs between edge routers, by loading the network with 840 LSPs $= 15 * 14 * 4 = N_{er} * (N_{er} - 1) * N_l$, with N_{er} is the number of edges routers and N_l is the number of established LSPs between each edge routers pairs.
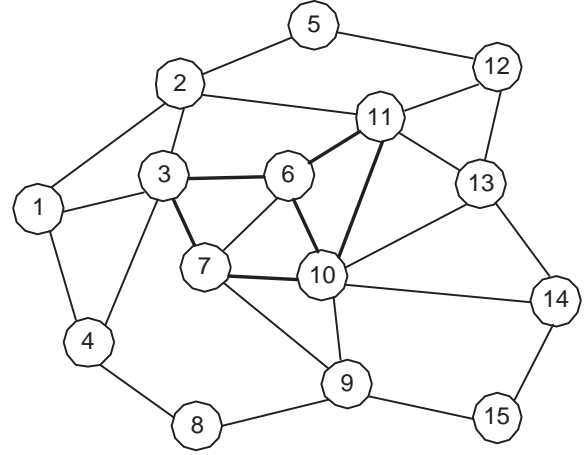

Fig. 3. The network topology

- We multiply the LSP's bandwidth by an increasing Traffic Scale factor k to vary the network load conditions.
- For each value of k, we conduct 100 trials by generating randomly 840 requests with bandwidth demands uniformly distributed between 1 and 50 Mbps.

The following metrics are used to evaluate our approach:

- *Rejected LSPs ratio*: The percentage of requests which are rejected due to insufficient resources.
- *Maximum link load*: $\max_i BW_i/C_i$, where C_i is the capacity of a link i and BW_i is the amount of traffic carried on this link.

A. Order Impact

The following notation is used in the remainder of the paper:

- CSPF: CSPF without LSPs reordering
- CSPF-IB-EO: CSPF with increasing bandwidth exact order
- CSPF-DB-EO: CSPF with decreasing bandwidth exact order
- CSPF-IB-LR-PO-CSPF: CSPF with increasing bandwidth linear repartition preemption based order
- CSPF-IB-NLR-PO: CSPF with increasing bandwidth Non linear repartition preemption based order.

Figure 4 shows the average maximum link load in the network after establishing 840 LSPs in random, increasing and decreasing bandwidth orders with CSPF algorithm. We see that IB-EO-CSPF outperforms the CSPF and the DB-EO-CSPF. This result can be explained as follows; In the decreasing case, the large LSPs are setup on shortest paths, the small LSPs requests arrive after and they fill the remaining bandwidth on shortest paths, thus increasing the maximum link load. However, in the increasing case, small LSPs are established at first and are routed through shortest paths and there is no longer enough bandwidth on shortest paths to route larger LSPs which are routed on non shortest paths. These non shortest paths are usually larger than the optimal paths after routing small LSPs. Thus, the residual bandwidth on shortest paths is larger in the increasing case (bandwidth which cannot

fit large LSPs) than in the decreasing case (bandwidth which cannot fit small LSPs).

In return, the gain remains negligible; the IB-EO-CSPF reaches an average better performance of about only 0.6% over the CSPF. We see also with the CSPF that, from $k = 0.7$ the rising scheme of the maximum link load slows down. This is due to the fact that the non shortest paths start to be used to route some LSP requests.
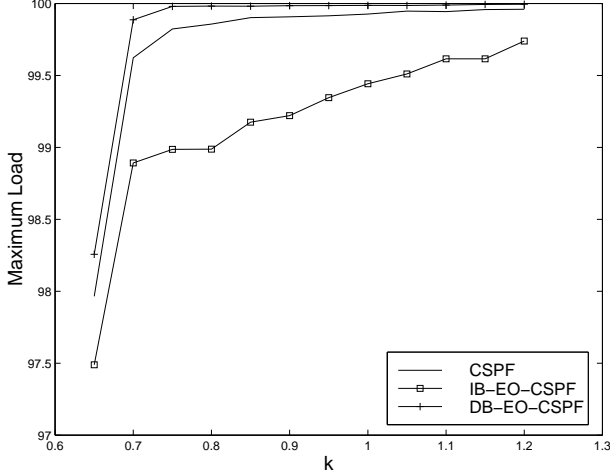

Fig. 4.   Average Maximum Link Load vs. k for CSPF without Preemption

The fig. 5 depicts the average LSP rejection ratio. It can be observed that the increasing order performs better than the decreasing and the random orders. For instance, for $k = 1.8$, the IB-EO-CSPF rejects 50% less requests than the CSPF. So, the LSPs ordering reduces significantly the number of rejected LSPs in congestion cases.
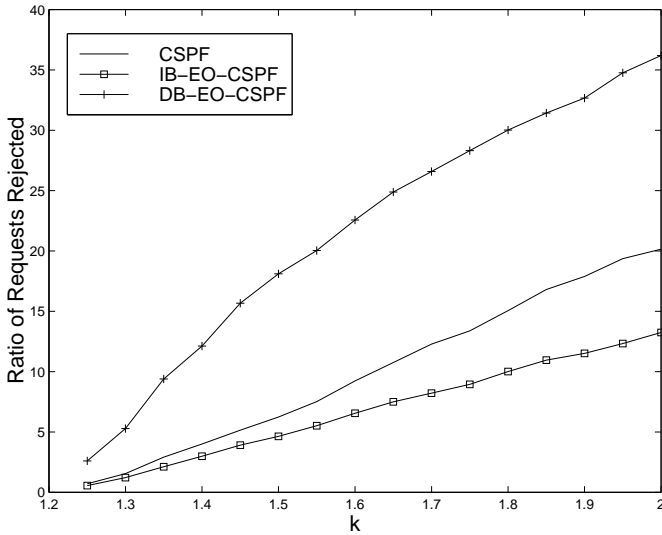

Fig. 5.   Average Rejected LSP Ratio vs. k without preemption

B. Preemption based reordering

Fig. 6 presents the performances in term of maximum link load of the CSPF when we introduce the preemption mechanism. Firstly, we can see that the performances of IB-LR-PO-CSPF and IB-NLR-PO-CSPF are close to those of IB-EO-CSPF. So, our proposed repartitions appear to be really efficient to dynamically re-order LSP requests (increasing or decreasing). The figure shows also that the IB-LR-PO-CSPF and the IB-NLR-PO-CSPF perform better that the CSPF. For instance, IB-LR-PO-CSPF reaches an average better performance of about 0.6% over the CSPF. From $k = 0.7$, the non shortest paths become to be used to route LSP requests because shortest paths are saturated. This explains the deceleration in the increase of the maximum link load from this value of k. It can be seen also that the IB-NLR-PO-CSPF is slightly better than the IB-LR-PO-CSPF and sometimes even better than the IB-EO-CSPF (for $k = 0.75$). In fact, this depends on the LSP bandwidth distribution. The optimal order is not necessary the IB order, such as for $k = 0.75$ where the LR preemption order is closer to the optimal order than the exact order.
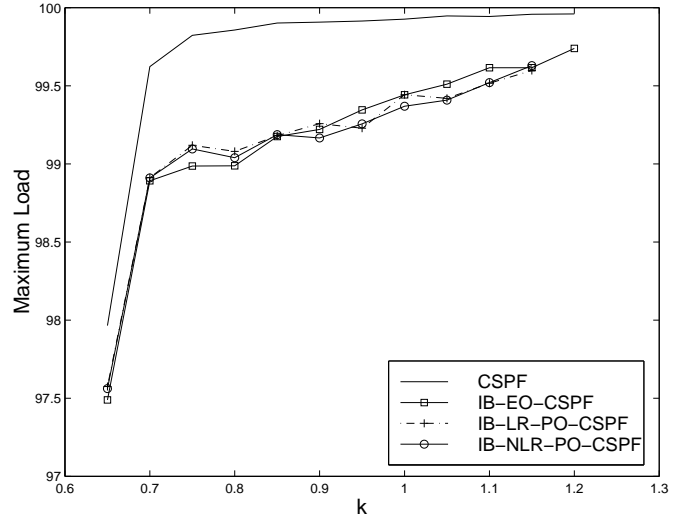

Fig. 6.   Average Maximum Link Load vs. k with Linear and Nolinear Repartition

The results in fig. 7 show firstly that the LR and NLR methods provide similar results in terms of rejected LSP ratio. Secondly, it can be seen that IB-LR-PO-CSPF and IB-NLR-PO-CSPF are less efficient than the IB-EO-CSPF but improve significantly, the performances of the CSPF placement. Actually, IB-LR-PO-CSPF and IB-NLR-PO-CSPF reject about 30% less than the CSPF but 13% more more than IB-EO-CSPF. Clearly, the preemption reordering applied to CSPF leads to a significant reduction of LSP rejections.

C. Network Failure Case

Now we evaluate the performances of CSPF with our approach when a link failure happens. For a given Traffic Scale Factor k, we load the network with 840 LSPs routed using the CSPF, IB-LR-PO-CSPF and IB-NLR-PO-CSPF. The
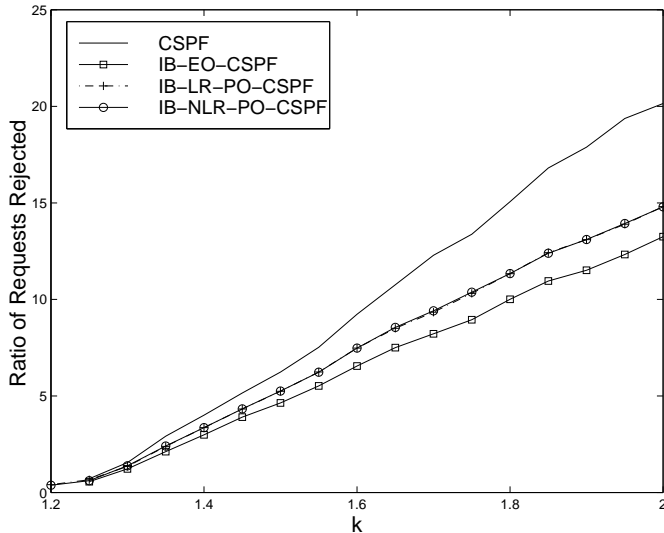
Fig. 7. Rejected LSP Ratio vs. *k* with Linear and Non Linear Repartition

network load is sufficiently low, so all LSPs are established without any rejection case. We then cut randomly a link (edge or core link) and re-route all the LSPs traversing this link. We proceed as follows 100 times, and each time, 840 LSP requests are randomly generated. The results are given in table II where the values correspond to the average number of rejected LSPs after rerouting upon one link failure. Note that the LSPs rejected when using the CSPF are a subset or all the LSPs impacted by the failure (the LSPs routed through the failing link). In return, the LSPs rejected when using IB-LR-PO-CSPF and IB-NLR-PO-CSPF may include LSPs which have not been impacted by the failure. Actually, when a failure occurs, the LSPs impacted by the failure will be rerouted by ingress routers. These impacted LSPs may preempt, on their new paths, lower priority LSPs which have not actually been directly impacted by the failure. Hence, after a network failure, such lower priority LSPs will be rerouted and potentially rejected if there are no sufficient resources. Table II shows the average number of rejected LSPs with CSPF, IB-LR-PO-CSPF and IB-NLR-PO-CSPF. It also shows the average number of LSPs impacted by the failure (Imp) as well as the average number of LSPs non impacted by the failure (NoImp), among the rejected LSPs.

Link	CSPF	IB-LR-PO		IB-NLR-PO	
		Imp	NoImp	Imp	NoImp
$14-15$	28.42	12.26		12.27	
		7.62	4.64	7.66	4.61
$12-13$	5.78	0.0		0.0	
		0.0	0.0	0.0	0.0
$8-9$	12.02	7.38		7.40	
		2.54	4.84	2.54	4.86
$2-5$	25.78	13.29		13.23	
		6.01	7.28	5.62	7.31

TABLE II
AVERAGE NUMBER OF REJECTED REQUESTS UNDER LINK FAILURE

In this experiment, requests are generated with $k = 1.05$. It can be seen that the re-ordering approach (LR or NLR) allows significant reduction in the number of rejections upon network failures. For instance, when the link $14 - 15$ is cut, there are 2.3 times less rejections with preemption based reordering, than in the random case (without reordering). $2/3$ of rejected LSP requests include LSPs impacted by the failure and $1/3$ of rejected LSP requests include LSPs non impacted by the failure. Also in some cases the reordering allows avoiding rejection (e.g. link 12-13 failure), which is a significant improvement.

VI. CONCLUSION

In this paper we have analyzed the impact of the LSP setup order on the optimization performances of an uncoordinated distributed MPLS-TE routing system. We have observed that if the LSPs are setup in increasing bandwidth order, the maximum link load is decreased and the blocking probability is reduced. We have discussed the use of preemption so as to dynamically reorder LSPs setup and we have proposed two approaches to allocate one of the eight preemption priorities to an LSP, according to its bandwidth. We have evaluated these approaches and observed that this significantly improves the performances of the CSPF algorithm in terms blocking probability. In return, as regards the maximum link load, the improvement is negligible.

As future work, we plan to evaluate the impact of the preemption on the network control plane and propose solutions to reduce this impact (in particular reduce the number of preemptions).

Finally, we plan to evaluate the impact of the LSP bandwidth distribution (e.g. linear, gaussian) on the performances of the LR and NLR approaches.

REFERENCES

[1] D. Awduche and B. Jabbari, "Internet traffic engineering using multi-protocol label switching (MPLS)," *Computer Networks*, vol. 40, no. 1, pp. 111–129, September 2002.
[2] H. Smit and T. Li, "IS-IS extensions for traffic engineering," RFC 3784, June 2004.
[3] D. Katz, K. Kompella, and D. Yeung, "Traffic engineering (TE) extensions to OSPF version 2," RFC 3630, September 2003.
[4] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP tunnels," RFC 3209, December 2001.
[5] I. Chaieb, J.L. Le Roux, and B. Cousin, "Generic architecture for MPLS-TE routing," *Fourth IASTED International Conference on Communications, Internet, and Information Technology (CIIT 2006)*, November 2006.
[6] T. Corman, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. MIT Press, 1990.
[7] R. Guerin, A. Orda, and D. Williams, "Qos routing mechanisms and OSPF extensions," *In Proceedings of 2nd Global Internet Miniconference (joint with Globecom97)*, November 1997.
[8] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal of Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.
[9] T. L. M. Kodialam, "Minimum interference routing with applications to MPLS traffic engineering," in *INFOCOM*, 2000.
[10] W. Szeto, R. Boutaba, and Y. Iraqi, "Dynamic online routing algorithm for MPLS traffic engineering," *The International Journal of Network and Systems Management*, vol. 10, no. 3, pp. 936–946, 2002.

[11] P. W. S. Suri, M. Waldvogel, "Profile-based routing: A new framework for MPLS traffic engineering," *Quality of future Internet Services*, no. 2156, pp. 138–157, 2001.

[12] D. M. C. V. M.R. Meyer, JP. Vasseur and A. Birjandi, "MPLS traffic engineering soft preemption," *draft-ietf-mpls-soft-preemption-08.txt*, October 2006.

[13] J. de Oliveira, C. Scoglio, I. Akyildiz, and G. Uhl, "A new preemption policy for diffserv-aware traffic engineering to minimize rerouting," *IEEE Infocom*, June 2002.