Path Selection Techniques to Establish Constrained Interdomain MPLS LSPs*

Cristel Pelsser and Olivier Bonaventure

CSE Department, Université catholique de Louvain, Belgium
{pelsser, bonaventure}@info.ucl.ac.be

Abstract. MultiProtocol Label Switching (MPLS) is used today inside most large Service Provider (SP) networks. In this paper, we analyze the establishment of interdomain MPLS LSPs with QoS constraints. These LSPs cross diverse SP networks that may belong to different companies. We show that using the standard BGP route for the establishment of such LSPs is not sufficient. We propose two path establishment techniques that rely on RSVP-TE and make use of Path Computation Elements (PCEs). Our simulations show that these techniques increase the number of constrained MPLS LSPs that can be established across domain boundaries.

1 Introduction

During the last years, MultiProtocol Label Switching (MPLS) has been deployed by most large SP networks. Initially, MPLS was offered as a replacement for ATM. However, the main driver for the current deployment of MPLS is its ability to provide new services with stringent Service Level Agreements (SLAs) such as layer-2 and layer-3 Virtual Private Networks (VPNs) as well as Voice and Video over IP. Most of these services are already deployed inside single SP networks. However, customers now require world-wide VPN and VoIP services. Therefore, SPs need to collaborate to offer these services across multiple SP networks.

Inside a single SP network, the provision of MPLS-based services with stringent bandwidth and delay requirements is typically achieved by using the Traffic Engineering (TE) extensions to the ISIS/OSPF routing protocol. These extensions enable to distribute with ISIS/OSPF the link loads and delays. Based on this information, each Label Switching Router (LSR) can use a Constrained Shortest Path First (CSPF) algorithm to find a constrained path toward any router inside the SP network. Then, it can use the Resource reSerVation Protocol with Traffic Engineering extensions (RSVP-TE) to signal the establishment of a traffic engineered MPLS Label Switched Paths (LSP) along this path. However, when traffic engineered LSPs with QoS and delay constraints must be terminated at a router in another SP network the selection of the path becomes a problem [8]. The CSPF algorithm cannot be used to find a constrained path between

* This work was partially funded by the Walloon Government (DGTRE) in the framework of the TOTEM project (http://totem.info.ucl.ac.be) and supported by the E-NEXT NoE funded by the European Commission.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 209–220, 2006.

two LSRs in different interconnected SP networks anymore. This is because the networks exchange routing information by using the Border Gateway Protocol (BGP). In contrast to OSPF-TE/ISIS-TE, BGP only provides reachability information. It does not distribute complete topology, delay and bandwidth information.

In this paper, we evaluate techniques that allow to establish traffic engineered or constrained LSPs across multiple SP networks. Our paper is organized as follows. In section 2, we introduce the issues that arise when considering TE across domain boundaries. Then, we present, in section 3, the path selection techniques that we evaluate in this paper. We propose two heuristics for the selection of the ingress node in the downstream domains and combine them with one of the techniques. Next, we evaluate the path selection techniques in section 4. Finally, we conclude the paper.

2 Interdomain Issues

BGP is the routing protocol used between SP networks, also called Autonomous Systems (ASs). As we have already mentioned, BGP only provides reachability information for the destinations. More precisely, it only provides the addresses of Next Hops (NHs), the nodes at the border of the domain, that are able to forward the packets to a given destination. The QoS properties of the paths, such as the delay and bandwidth, behind these NHs are not provided. This results in several limitations for the computation and establishment of constrained interdomain LSPs.

Firstly, inside an AS[1], all routers learn the complete topology of the AS by means of ISIS/OSPF. Thus, each router is able to compute the complete path from head-end to tail-end node for an LSP contained in the AS. However, the topology of an AS is hidden to routers outside the AS, for confidentiality purposes [10]. As a consequence, a single node is not able to compute the end-to-end path for an LSP crossing multiple ASs. Therefore, the computation of such a path has to be distributed among multiple nodes, where each node computes a segment of the path based on its knowledge of the local AS topology and the interdomain reachability information provided by BGP.

Secondly, we have shown in [8] that a router only possesses a subset of the possible routes for a destination. Moreover, the set of routes learned by a BGP router are not necessarily the best possible routes with regard to the end-to-end delay and the available bandwidth. The BGP routes are first selected based on local preferences and the AS path length. However, Huffaker et al. have shown in [7] that the AS path length does not reflect the delay of the path. Thus, interdomain routes with a low delay may never be learned by some routers. The diversity of the BGP routes available at each router is not sufficient to successfully compute constrained interdomain LSPs.

Extensions to BGP in order to advertise the QoS of the interdomain routes are proposed in [1]. However, such extensions have not been evaluated nor deployed. In [13] and [6], the authors define an architecture with a centralized entity inside each domain. They propose to define a new interdomain routing protocol to be used between the entities and to exchange QoS information with this routing protocol. Up to now such a routing protocol has not been defined. It is not currently possible to know a priori the

[1] We consider ASs composed of a single IGP area. This is the most common deployment today.

QoS that can be provided along an interdomain route. Thus, in this paper we rely on heuristics to estimate the QoS of a route.

3 Path Selection Techniques

In this section we present four path computation techniques for constrained interdomain MPLS LSPs. The last two techniques are based on the same principle, ERO expansion. However, they make use of two different heuristics that are proposed in this section.

3.1 Standard IP forwarding

The simplest technique to establish an interdomain MPLS LSP is to follow the same path as the normal IP packets. This path is determined by BGP for destinations outside the AS. This path would be chosen by the Label Distribution Protocol (LDP) if LDP was used between ASs.

3.2 Centralized Path Selection with CSPF

In this technique, the computation is performed by a single entity, that we name "global PCE". We assume that the global PCE learns the complete topology by receiving the ISIS/OSPF link state packets of each AS. It performs a CSPF computation for each LSP. We note that such a computation does not rely on BGP. It is not constrained by BGP peering relationships and route filtering. This computation provides an indication of the path quality that can be achieved with a centralized computation.

Such a centralized solution could be envisaged when MPLS LSPs are entirely contained inside ASs that belong to the same company. However, it is not realistic for MPLS LSPs that cross ASs from different companies as this requires the ASs to cooperate and reveal their internal topology. Moreover, this solution is not scalable in the number of nodes and links of the ASs considered by the centralized computation. We use it as a benchmark and compare it with more easily deployable techniques.

3.3 ERO Expansion

Because the use of a global PCE performing CSPF computations is not applicable in the general interdomain framework, other techniques are required. In this section, we consider the use of RSVP-TE to establish interdomain MPLS LSPs.

Inside RSVP-TE, it is feasible to indicate the path or a portion of the path to be followed by the LSP inside an object called the Explicit Route Object (ERO). The ERO expansion technique, described in [12], relies on this object. It consists in completing at the ingress router of a domain, the ingress AS Border Router, the path computation up to the last reachable hop within the downstream domain, i.e. the BGP Next-Hop (NH). The computed path segment is then stored inside the ERO of the RSVP-TE Path message. This message is forwarded along the path specified inside the ERO and requests the establishment of the LSP along the path.

In addition to RSVP-TE signalling, we assume that there is a Path Computation Element (PCE) [5] inside each domain. The PCE is responsible for the computation

of the paths on behalf of the ingress routers. It receives all the BGP routes learned inside the AS in order to improve the diversity of the routes available for the path computation [8].

Upon reception of an RSVP Path message requesting the establishment of an LSP, an AS Border Router (ASBR) sends a Path Computation Request (PCReq) to its PCE. After the completion of the computation, the PCE replies with a Path Computation Reply (PCRep) message. This message contains a path segment from the ingress ASBR to a BGP Next-Hop (NH) or indicates that there is no path segment respecting the constraints.

The ASBRs store the list of NHs that have already been tried for an LSP and lead to an infeasible path with regard to the constraints. When the PCE is not able to complete the path with a segment respecting the constraints, "crankback" is performed [4]. That is, the ASBR generates an RSVP Path Error message and sends it upstream. The upstream ASBR requests from its PCE the computation of a new segment avoiding the NHs that have already been tried.

The role of crankback is crucial for the establishment of interdomain LSPs because only limited information is available concerning the paths to reach a destination outside an AS. Thus, a PCE that computes a portion of a constrained interdomain LSP must rely on heuristics to choose an appropriate BGP NH among the NHs announced for the destination. If a bad choice is performed by the heuristic at some PCE, a downstream PCE may not be able to complete the computation of the path. Crankback enables to cope with such a situation and subsequently try alternative NHs.

In this paper, we propose two heuristics for the selection of the NHs by the PCEs during the computation of LSPs. The heuristics try to determine the NHs that are along short delay paths because the LSPs considered are subject to maximum end-to-end delay constraints in addition to bandwidth reservations.

Nearest NH. We call our first NH selection heuristic "nearest NH". Two link metrics are provided with ISIS-TE/OSPF-TE : the classical IGP metric and a TE metric. The IGP metric is usually set to the link bandwidth. We propose to set the TE metric of a link to its delay. Among the NHs available for the destination, the PCE selects the NH with the shortest path, from the ASBR to the NH, with enough bandwidth to support the LSP. The TE metric is used for the computation of the shortest path.

Vivaldi $2d + h$ Coordinates. Selecting the "nearest NH" in terms of the delay, as in the first heuristic, does not ensure that the end-to-end delay of the path will be low. The path segment downstream of a NH selected with the "nearest NH" heuristic may have a long delay. Thus, the heuristic proposed in this section relies on a delay estimation of the paths through the candidate NHs up to the tail-end of the LSP.

We use a virtual coordinate system, called Vivaldi [2], to estimate the delay of a path between two nodes. In this coordinate system each node computes its coordinates based on RTT measurements with a limited number of other nodes. Nodes connected with a low delay path will have neighboring coordinates while nodes connected through a higher delay path will be further apart.

In the heuristic presented in this section, we prefer to explore NHs that are along the path with the smallest delay estimation toward the tail-end D, to minimize the delay of the remaining portion of the path to D. Thus, for an ingress ASBR I_c inside an AS

AS_c, we prefer the ingress ASBR I_d inside a downstream AS AS_d such that

$$delay(I_c, I_d) + distance(I_d, D) = \min_{I_j \in NH}(delay(I_c, I_j) + distance(I_j, D))$$

where NH is the set of potential NHs for tail-end D, $delay()$ is the delay of the ISIS/OSPF path computed with the TE metric and $distance()$ is the distance between two points in the virtual coordinate space.

In our simulations, each node computes its coordinates in a two-dimensional Euclidean space augmented with an height, noted $2d + h$, as proposed in [2]. The distance between two nodes with coordinates (x_1, y_1, h_1) and (x_2, y_2, h_2) in the $2d + h$ space is the sum of the distance of the first node to the plane (its height, h_1), the Euclidean distance between the coordinates of the two nodes in the plane ($\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$) and the distance from the plane to the second node (the height of the second node, h_2).

In order to compute the preference of the candidate NHs, the PCE needs to know the coordinates of each NH and of the LSP's tail-end. For this purpose, we assume that after the computation of its coordinates, each node stores these coordinates inside its Domain Name Server (DNS), as proposed in [3]. The PCE requests the coordinates of the candidate NHs and the destination from the DNS.

In figure 1, we illustrate the selection of the NH by the two heuristics for an LSP entering $AS2$ at router $R2$ with tail-end $R8$. There are two candidate NHs, $R5$ and $R6$, for destination $R8$. The PCE inside $AS2$ prefers $R5$ over $R6$ with the "nearest NH" heuristic because the shortest delay path from $R2$ to $R5$ is 2 and the shortest delay path from $R2$ to $R6$ is 7. With the "vivaldi" heuristic, the PCE prefers $R6$ instead of $R5$ because the delay estimation[2] of the path from $R2$ to $R8$ transiting through $R6$ is $7 + \sqrt{(37 - 34)^2 + (18 - 10)^2} = 15.5$ and the delay estimation of the path transiting through $R5$ is $2 + \sqrt{(61 - 34)^2 + (78 - 10)^2} = 75$. The path from $R1$ to $R8$ obtained with the "nearest NH" heuristic is $R1 - R2 - R4 - R5 - R7 - R6 - R8$ with delay of 44 ms. On the other hand, the path $R1 - R2 - R4 - R3 - R6 - R8$, resulting from the computation with the "vivaldi" heuristic has a shorter delay of 9 ms.

4 Simulations

In this section, we present the results of simulations on two types of topologies[3]. First, we use topologies composed of 5 transit ASs to evaluate our heuristics in a small environment with MPLS deployed between the ASs. Such an environment is conceivable today. Then, we apply the path computation techniques on a larger topology composed of 20 transit ASs, as in the core of the Internet [11], to evaluate the techniques in a large scale deployment of inter-AS MPLS LSPs. We compare the four path selection techniques of section 3 in our simulations.

[2] In this example, we consider $2d$ coordinates. The delay estimation between two nodes in this $2d$ space is the Euclidean distance between the coordinates of the two nodes.

[3] The topologies and scripts used to provide the results presented in this section are available to the research community at the following URL: `http://totem.info.ucl.ac.be/tools`.

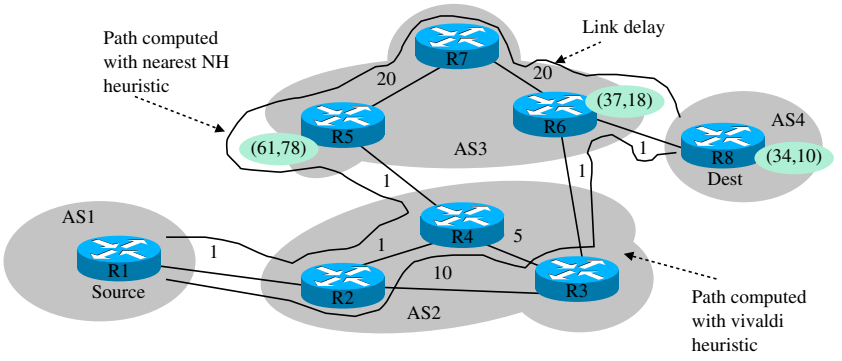

Fig. 1. Nearest NH versus vivaldi heuristics

4.1 Topologies

The topologies used for the simulations are generated with the transit-stub model of the GT-ITM tool [14]. First we generated 5 topologies each composed of 5 transit ASs. In these topologies, each transit AS is composed of approximatively 50 routers. The links inside the transit ASs are generated randomly with the parameters suggested by the authors of GT-ITM in [14]. GT-ITM attaches one stub AS to each router in a transit AS and randomly adds 250 extra links between the transit and the stub nodes. Each stub AS only contains one router. This router is the end-point of the LSPs established on the topology.

We group the stubs in classes that contain all the stubs attached to the same providers. We only keep one stub from each class to reduce the simulation time. It results in topologies with an average of 27 stubs. The nodes in these selected stubs and the nodes inside the transit ASs are placed by GT-ITM in an Euclidean plane. This placement is used to set the delay of the links. In our topology, the delay of a link is directly proportional to the Euclidean distance between its two end-points. In addition, we assign the same bandwidth to all the links.

In our simulations, we establish a full-mesh of LSPs between the routers in the stub ASs. Such a full-mesh could correspond to a very large interdomain BGP/MPLS VPN service. We establish the LSPs in one direction only. All LSPs are subject to the same bandwidth reservation (100 Mbps) and delay constraint (1900 ms). With a bandwidth reservation of 100 Mbps we can emulate the Fast-Ethernet service between Service Providers.

The delay constraint is determined as follows. For each LSP to be established, we computed the shortest path in terms of delay from the head-end to the tail-end node, on the complete topology and without BGP policies and filtering. We set the delay constraint of the LSPs to a round value just above the maximum delay of the resulting paths to ensure that, for each LSP, a path respecting the delay constraint exists in the topology.

We use the C-BGP simulator [9] to compute the BGP routing tables of the nodes. The routers inside stub ASs are configured not to advertise routes received from other ASs. Thus, stub ASs do not provide transit service. Transit ASs do not filter out the

routes advertised to neighboring ASs. This ensures that each AS receives at least one route for each destination.

The second topology is composed of 20 transit ASs as the core of the Internet. It is generated by the method described earlier for the topologies with 5 transit ASs. Again, the transit ASs are composed of 50 nodes and all links have the same capacity. This topology has 411 stub ASs. We try to establish 84255 LSPs on this topology. Again all LSPs are subject to the same bandwidth reservation (100 Mbps) and end-to-end delay constraint (3300 ms).

4.2 Evaluation of the Path Selection Techniques

In this section, we present the results of the simulations on the topologies introduced in section 4.1. We first describe the results obtained from the simulations with the topologies containing 5 transit ASs. Then, we analyze the results obtained on the larger topology. In this analysis, we focus our attention on three aspects: the end-to-end delay of the LSPs, the number of LSPs that can be supported by the network, in our case this is proportional to the total amount of traffic that can be carried on the topology, and, finally, the amount of crankback that occurs during the computation of the constrained paths.

For each topology, we performed several simulations. The link bandwidths are set to a different value in each simulation. The objective is to study the impact of various levels of congestion on the LSP's establishment techniques. In the first simulation, the bandwidth of all links is set to 10 Gbps. Then, it is set to 2400 Mbps in the second simulation. Finally, it equals 622 Mbps in the third simulation.

In the remaining of this section, we distinguish the LSPs for which a path respecting the constraints could be found in the topology, called "established LSPs", from LSPs for which no suitable path could be found, called "failed LSPs".

The curves in figure 2 are obtained from simulations on the small topologies with link bandwidths set to 2400 Mbps and 622 Mbps. The results from the simulations with 10 Gbps links are similar to the results obtained with link bandwidths equal to 2400 Mbps. This is because there is no congestion in the topology with 10 Gbps links and only a few links are congested with 2400 Mbps links. In the latter topology, only 2 links are congested with the ideal CSPF computation inside the global PCE. Moreover, 10 links are congested with the "nearest NH" heuristic and, 0 links with "vivaldi".

Figures 2 (a) and 2 (b) show the cumulative distributions of the end-to-end delay for the different path computation techniques. They show, for a given delay on the x-axis, the number of established LSPs, on the y-axis, with end-to-end delay lower or equal to the value on the x-axis. Figures 2 (a) and 2 (b) present the results for a single topology, topology 0. Only the LSPs established on the topology are considered in these figures. The simulations performed with the other topologies with 5 transit ASs provide similar results.

In figure 2 (a), we first observe that there are more paths with a low delay with the CSPF computation performed by the global PCE than with the other techniques. We note that this computation does not rely on BGP. It is not constrained by BGP policies and filtering. It is used as an upper performance bound to which the other methods are compared. Moreover, there are more IP forwarding paths with a low delay than with the "nearest NH" and "vivaldi" heuristics. The good quality of the IP forwarding paths

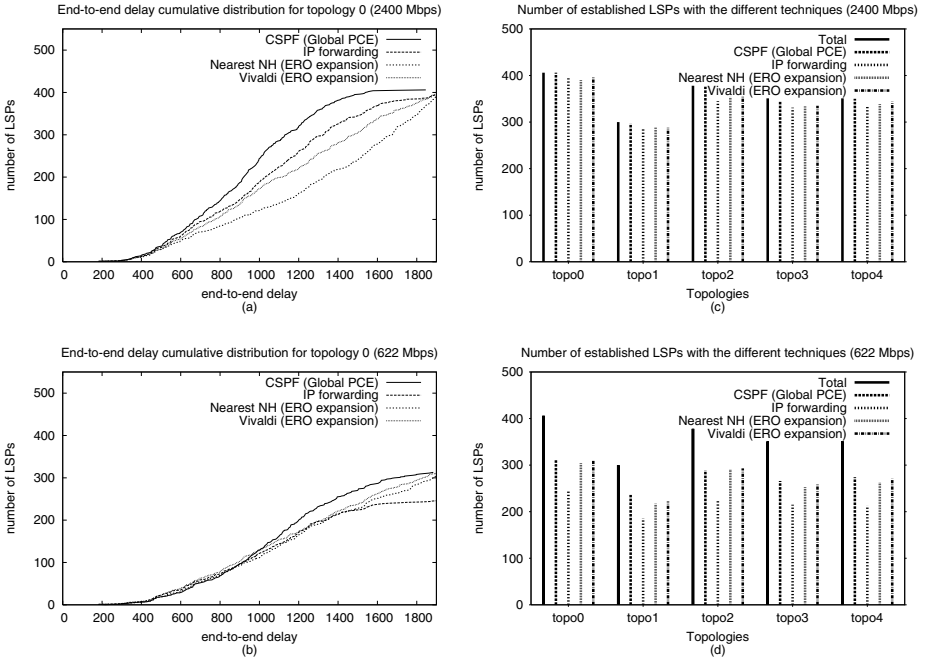

Fig. 2. Delay of LSPs established on topology with 5 transit ASs

in terms of delay comes from the fact that many BGP routes in our simulation are selected based on the IGP cost. Since we set the IGP cost of a link to its delay, the BGP selection rule based on the IGP cost prefers a route with a low delay over a route with a longer delay. Finally, the "vivaldi" heuristic provides more paths with a low delay than the "nearest NH" heuristic. This is due to the fact that "nearest NH" selects the NH only based on delay information that is local to the domain whereas the "vivaldi" NH selection is based on an estimation of the delay of the path that transits through the candidate NH.

When the bandwidth of the links is set to 622 Mbps in topology 0, congestion occurs on 4% of the links with CSPF and on 3% of the links with "nearest NH" and "vivaldi" path computation techniques. We observe in figure 2 (b) that there is not much difference between the 4 curves for the LSPs with end-to-end delay below 1000 ms. Above this value, there are more CSPF paths with a low delay compared to the other path computation techniques. Finally, we note that the total number of LSPs established along the IP forwarding paths is below the number of LSPs established with CSPF and our two heuristics. With IP forwarding, a router can only use a few outgoing interfaces for a destination. When the corresponding links are congested, the router is not able to send the path establishment request on an alternate link. Thus, the LSP establishment fails. However, the "nearest NH" and "vivaldi" heuristics rely on RSVP-TE for the establishment of the LSPs. RSVP-TE enables to avoid congested links in the establishment of an LSP by specifying the path to be followed by the LSP inside the Explicit Route Object (ERO) in order to bypass IP forwarding. Thus, techniques based on ERO expansion

are more robust to congestion than standard IP forwarding. At last, we see that there are slightly more paths with a low delay with the "vivaldi" heuristic than with "nearest NH". However this difference is not significant.

Figures 2 (c) and 2 (d) show the number of LSPs that can be successfully established on each topology, with the different path computation techniques. In figures (c) and (d), we observe that the number of LSPs established by the techniques relying on BGP routes, that is IP forwarding, "nearest NH" and "vivaldi", is lower than with CSPF. The CSPF paths are computed by a centralized entity that possesses the complete topology. With BGP, however, only a portion of the routes available for a destination is distributed. Since there are fewer routes, they become faster congested. Moreover, some of these routes are selected by BGP based on other criterion than the delay. Thus, the resulting paths learned for a destination do not necessarily have a lower delay than the maximum end-to-end delay constraint of the LSPs.

We note that the number of LSPs established with the "vivaldi" heuristic is slightly higher than this number for the simulations with "nearest NH". In both cases, the set of potential NHs depends on the BGP routes received for the destination. The set of potential NHs, inside an AS, is the same for many destinations. Among this set, the selection of the NH for a given ingress ASBR only relies on the delay of the shortest delay path with enough bandwidth for the LSP, in the "nearest NH" heuristic. Thus, the LSPs entering an AS through an ingress point incur the same delay until the shortest delay path becomes congested and a longer delay path is followed in the AS. Therefore, the delay incurred inside an AS by the LSPs entering at the same ingress ASBR increases as the LSPs are established. On the other hand, the selection of the NH by the "vivaldi" heuristic relies on the shortest delay path inside the AS and on the delay estimation from the NH to the destination of the LSP. Consequently, the delay incurred by the LSPs crossing an AS does not increase as fast as with the "nearest NH" heuristic because the LSP establishment requests entering an AS at an ingress point are distributed among multiple paths inside the AS based on the destination of the LSP. The delay of the paths computed with the "nearest NH" heuristic increases faster than with "vivaldi". As a consequence, if all LSPs are subject to the same delay constraint, this constraint will be harder to fulfill with "nearest NH" than with "vivaldi", as the LSPs are established.

Figure 2 (d) shows that in a less provisioned network, compared to the results in figure 2 (c), the number of LSPs that can be established along IP forwarding paths drops, as mentioned earlier. In addition, the number of LSPs established with our two ERO expansion heuristics is not far below the number of LSPs established along the CSPF paths computed by the global PCE. This is mostly due to the use of the RSVP-TE ERO expansion technique and the crankback mechanism.

The crankback mechanism enables to inform an upstream node of the failure during the establishment of an LSP and to try the establishment of the LSP along another path. Crankback occurs at most 12 times with "vivaldi" and 25 times with "nearest NH" for established LSPs, on topology 0 with link bandwidths set to 2400 Mbps. However, there are 95% of LSPs established with "vivaldi" without the help of crankback and 73% with "nearest NH". On topology 0 with 622 Mbps links, there are 85% and 67% of the LSPs that are established without performing crankback with the "vivaldi" and "nearest NH"

heuristics, respectively. Moreover, the maximum number of crankback for established LSPs is 13 for "vivaldi" and 14 for "nearest NH". We observe that, for the congested topology, crankback enables to carry more traffic inside the congested topology than when the IP forwarding paths are used. The contribution of the crankback mechanism, in the interdomain framework where the complete topology and the traffic load is not known by a single entity, is significant.

Now, we analyze the results of the simulations performed with the topology containing 20 transit ASs and 10 Gbps links. In these simulations, there are 13% of congested links with CSPF, and only 2% with both "vivaldi" and "nearest NH" heuristics. We observe from figure 3 (b) that the amount of traffic carried inside the topology is higher with CSPF than with our heuristics. There are 19% (18%), from the total amount of LSPs, of additional LSPs established with CSPF compared to the use of "nearest NH" ("vivaldi", respectively). Moreover, there is a difference of 32%, from the total number of LSPs, of established LSPs between CSPF and IP forwarding.

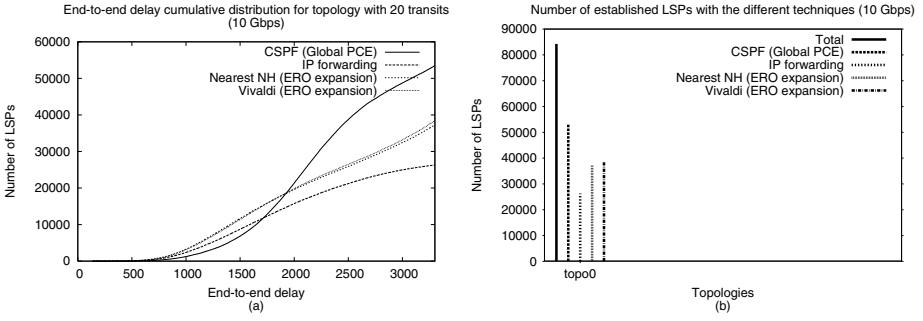

Fig. 3. Delay of LSPs established on topology with 20 transit ASs

Figure 3 (a) shows the cumulative distribution of the end-to-end delay for the LSPs established on the topology with the different path selection techniques. This distribution is almost the same for the two heuristics coupled with ERO expansion. However, we see that there are more paths with an end-to-end delay below 1927 ms, with our two heuristics than with CSPF. We assume that this is due to the higher number of LSPs established with CSPF than with the two heuristics. Some LSPs with a delay shorter than 1927 ms are established at the end of the simulation with our heuristics. However, with CSPF the low delay links are already congested. A short delay path may be found by the heuristics because there is less congestion in the topology than with CSPF due to the lower number of LSPs already supported by the topology.

On this large topology, crankback plays an important role. There are 45% of the established LSPs for which crankback occurs with "vivaldi" and 54% with "nearest NH". The maximum number of crankback for the establishment of an LSP is 199 with "vivaldi" and 283 with "nearest NH". However, there is less than 6 crankbacks for 90% of the LSPs established with "vivaldi" and less than 8 crankbacks, respectively, with "nearest NH".

5 Conclusion and Further Work

In this paper, we studied the establishment of constrained interdomain MPLS LSPs. We presented and evaluated four path computation techniques. Two of these techniques rely on heuristics proposed in this paper. The first technique tries to establish constrained MPLS LSPs along the default BGP route. The other techniques take advantage of RSVP-TE and the Path Computation Elements (PCEs) that are currently discussed within the IETF. First, we assume the existence of a global PCE that performs a CSPF computation on the complete topology for each LSP. This technique is not applicable in a general interdomain framework. It gives an upper performance bound for the other techniques. In the last two techniques, the computation of the constrained paths is distributed. Each PCE selects a NH to leave its AS based on the heuristics proposed in this paper and computes the path toward this NH.

Our simulations showed that using the default BGP route to establish constrained MPLS LSPs is not a good solution. A large amount of LSPs cannot be established. In addition, the simulations indicate that the number of constrained interdomain MPLS LSPs successfully established significantly increases with the two heuristics. Moreover, the "vivaldi" heuristic is slightly better than "nearest NH". More LSPs are established with "vivaldi" and the maximum amount of crankback is lower. However, this negligible improvement has a cost. It requires the computation of coordinates. Finally, we saw that the amount of crankback during the establishment of the LSPs with both heuristics is low for a very large portion of the LSPs. This is a strong argument in favor of ERO expansion for the current standardization work within the IETF.

In this paper, we presented and evaluated two heuristics for the "ERO expansion" architecture described in [5]. [5] also proposes another architecture that relies on communication between PCEs in order to find a constrained path for an LSP. If the list of ASs to be crossed by the LSP is not known a priori, the heuristics of this paper may be used to select a subset of the downstream ASs and, thus, of the PCEs that will contribute to the path computation. We propose to evaluate such a solution in the future.

Acknowledgements

The authors thank Steve Uhlig, Virginie Van den Schrieck and Pierre Francois for their reviews as well as Bruno Quoitin for providing the C-BGP tool. The authors thank Cédric de Launois for the code related to the computation of the vivaldi coordinates.

References

1. M. Boucadair. QoS-Enhanced Border Gateway Protocol. Internet draft, draft-boucadair-qos-bgp-spec-00.txt, work in progress, June 2005.
2. F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of the ACM SIGCOMM '04 Conference*, Portland, Oregon, August 2004.
3. C. de Launois. *Unleashing Traffic Engineering for IPv6 Multihomed Sites*. PhD thesis, Université catholique de Louvain, September 2005.

4. A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash. Crankback signaling extensions for MPLS and GMPLS RSVP-TE. Internet draft, draft-ietf-ccamp-crankback-05.txt, work in progress, May 2005.
5. A. Farrel, J-P. Vasseur, and G. Ash. Path computation element (PCE) architecture. Internet draft, draft-ietf-pce-architecture-02.txt, work in progress, September 2005.
6. N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *ACM SIGCOMM workshop on Future Directions in Network Architecture (FDNA 2004)*, August 2004.
7. Bradley Huffaker, Marina Fomenkov, Daniel J. Plummer, David Moore, and k claffy. Distance metrics in the internet. In *IEEE International Telecommunications Symposium*, 2002.
8. C. Pelsser, S. Uhlig, and O. Bonaventure. On the difficulty of establishing interdomain LSPs. In *IEEE International Workshop on IP Operations and Management (IPOM 2004)*, Beijing, China, October 11-13th 2004.
9. B. Quoitin and S. Uhlig. Modeling the Routing of an Autonomous System with C-BGP. *IEEE Network*, 19(6), November 2005.
10. N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1):2–16, February 2004.
11. L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *INFOCOM 2002*, June 2002.
12. J-P. Vasseur, A. Ayyangar, and R. Zhang. A per-domain path computation method for computing inter-domain traffic engineering (TE) label switched path (LSP). Internet draft, draft-ietf-ccamp-inter-domain-pd-path-comp-00.txt, work in progress, April 2005.
13. M. Yannuzzi, S. Sánchez-López, X. Masip-Bruin, J. Solé-Pareta, and J. Domingo-Pascual. A combined intra-domain and inter-domain qos routing model for optical networks. In *9th conference on Optical Network Design and Modelling (ONDM 2005)*, Milan, Italy, February 7-9th 2005.
14. Ellen W. Zegura, Kenneth L. Calvert, and Michael J. Donahoo. A quantitative comparison of graph-based models for Internet topology. *IEEE/ACM Transactions on Networking*, 5(6):770–783, 1997.