

Using Virtual Coordinates in the Establishment of Inter-domain LSPs*

Cristel Pelsser
CSE Department, Université catholique de Louvain, Belgium
pelsser@info.ucl.ac.be

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Packet-switching networks

General Terms: Design, Performance

Keywords: Inter-domain, MPLS, PCE, BGP, Virtual coordinates

1. INTRODUCTION

MultiProtocol Label Switching (MPLS) is used inside large ISP networks to provide services with stringent Service Level Agreements such as Virtual Private Networks (VPNs). Customers are now urging ISPs to provide such services across inter-domain boundaries. This requires the ability to establish inter-domain MPLS Label Switched Paths (LSPs) with constraints. Up to now, the literature has mostly focused on mechanisms to compute LSPs inside a single AS. In this paper, we propose two heuristics to select the downstream AS and the ingress router inside this AS for the establishment of inter-domain LSPs. Then, we evaluate them in terms of the quality of the resulting paths and the number of unsuccessful attempts.

2. ERO EXPANSION

In this short paper, we consider that a domain corresponds to an Autonomous System (AS). The ERO expansion technique is described in [6]. It consists in completing at the ingress router of the domain the path computation up to the last reachable hop within the downstream domain, i.e. the BGP Next-Hop (NH). We assume that there is a Path Computation Element (PCE) [3] inside each domain. The PCE is responsible for the computation of the paths on behalf of the ingress routers. For this purpose, it receives all the BGP routes learned inside the AS.

The AS Border Routers (ASBRs) store the list of NHs that have been tried for an LSPs 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, “crackback” is performed [2]. That is, the ASBR generates an 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 PCE usually possesses many BGP routes for a destination. The NH of each of these routes is eligible as the tail-end of the path

*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.

segment to be returned. Thus, it is important to select, among these NHs, a NH that will not require to perform crackback.

In this paper, we propose and evaluate two different selection mechanisms for LSPs subject to a maximum delay constraint. The first one is very simple. The other NH selection heuristic is more elaborated. It relies on the estimation of virtual coordinates.

2.1 Nearest NH

We call the first mechanism “nearest NH”. Among the NHs available for the destination, the PCE selects the NH for which the delay along the shortest delay path from the ASBR to the NH is the smallest. This heuristic is based on local Traffic Engineering (TE) metrics only. It does not ensure that the path segment downstream of the selected NH will have a low delay.

2.2 Vivaldi 2d coordinates - angle

Vivaldi [1] is a virtual coordinate system in which each node computes its coordinates based on RTT measurements with a limited number of other nodes. In this coordinate system, nodes connected with a low delay path have neighboring coordinates while nodes connected through a higher delay path are further apart.

Here, we propose to use the Vivaldi coordinate system in a two-dimensional Euclidean space to guide the selection of the next-hop (NH). We prefer to explore NHs that are in the direction of the tail-end D , first, in order to get closer to the destination. The nearer we get to the destination, the smaller is the delay of the path that remains to reach the destination.

Thus, for an ingress ASBR I_c inside an AS AS_c , we prefer the ingress ASBR I_d inside the downstream AS AS_d such that

$$angle(\overrightarrow{I_c I_d}, \overrightarrow{I_c D}) = \min_{I_j \in NH} angle(\overrightarrow{I_c I_j}, \overrightarrow{I_c D})$$

where NH is the set of potential NHs for destination D and $angle()$ the angle between two vectors.

3. SIMULATIONS

The topology used for the simulations is generated with the transit-stub model of the GT-ITM tool [7]. The topology is composed of 20 transit ASs like the core of the Internet. Each transit domain is composed of approximately 50 routers. The edges inside the transit domains are generated randomly with the parameters suggested in [7]. GT-ITM attaches one stub to each router in a transit AS and randomly adds one thousand extra links between the transit and the stub nodes. Each stub 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 with the same providers. We only keep one stub from each class. It results in a topology with 411 stubs. These selected stubs having been

generated by GT-ITM, they are placed in an Euclidean plane by this topology generator. This placement is then 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 our simulations, we establish a full-mesh of LSPs between the routers in the stub ASs. We establish the LSPs in one direction only. It follows that there are $411 * 410 / 2 = 84255$ LSPs for which a path has to be computed.

All LSPs are subject to the same delay constraint (3300 ms). For each LSP, 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 constraint exists in the topology.

We use the C-BGP simulator [4] 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 on behalf of the transit ASs. 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.

3.1 Results

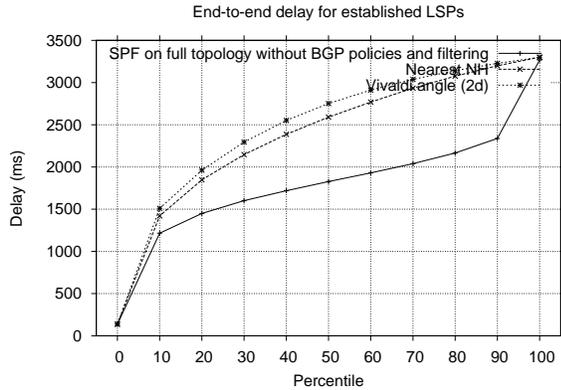


Figure 1: End-to-end delay

Figure 1 shows various percentiles for the end-to-end delay of the LSPs that are established on the topology described earlier. There are three curves in this figure. First, we computed the shortest delay paths based on the complete knowledge of the topology and without BGP policies and filtering. This curve is labeled “SPF on full topology without BGP policies and filtering”. It shows the percentiles for the smallest delay paths that are possible for the LSPs in the topology. Then, we computed the paths with the “nearest NH” and “vivaldi angle” heuristics. These heuristics rely on the BGP routes available inside the AS. We note that 6% of the LSPs cannot be established with the path computation techniques based on BGP routes because a path respecting the constraints could not be found. In addition, we see in figure 1 that the paths obtained without BGP policies and filtering have a lower delay than the paths computed based on BGP routes. There are 90% of the LSPs with a delay less than or equal to 2338ms with SPF. However, only 40% of the LSPs have a delay below or equal to 2387ms and 2552ms with the nearest NH and the vivaldi angle heuristics, respectively. This comes from the fact that some routes with a low delay may never be learned inside an AS due to the BGP routing policies and BGP route distribution. Finally, we observe that there are more paths

with a low delay with the vivaldi angle heuristic compared to the nearest NH heuristic.

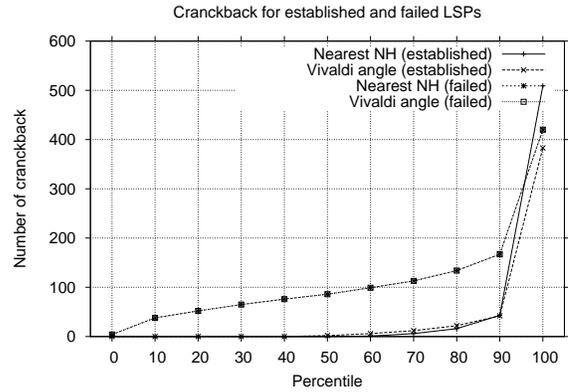


Figure 2: Number of crankedback

In figure 2, we evaluate the amount of crankedback that occurs with the “nearest NH” and “vivaldi angle” heuristics. We distinguish the LSPs that can be established on the topology, “established LSPs”, from the LSPs for which no path respecting the delay constraint could be found, “failed LSPs”. Both heuristics give the same results for “failed LSPs” because the same segments are tried. We note that the number of crankedback required to determine that an LSP cannot be established is higher than for the successful establishment of an LSP. Moreover, the “nearest NH” heuristic requires less crankedback for 90% of the LSPs compared to “vivaldi angle”.

4. CONCLUSION

In this paper, we proposed two heuristics for the establishment of delay constrained MPLS LSPs across domains. We showed that the quality of the paths is impacted by the BGP routes available for the computation. In addition, we observed that the heuristic that relies on virtual coordinates gives more paths with a lower delay than the simple heuristic. However, the number of crankedback is slightly higher than with the simple heuristic.

In the future we plan to elaborate heuristics for the support of bandwidth and delay constrained LSPs. The objective of these heuristics will be to maximize the resource usage of the network in addition to compute inter-domain paths respecting the constraints.

5. REFERENCES

- [1] 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.
- [2] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash. Crankedback signaling extensions for MPLS and GMPLS RSVP-TE. Internet draft, draft-ietf-ccamp-crankback-05.txt, work in progress, May 2005.
- [3] A. Farrel, J-P. Vasseur, and G. Ash. Path computation element (PCE) architecture. Internet draft, draft-ietf-pce-architecture-00.txt, work in progress, March 2005.
- [4] B. Quoitin. C-BGP, an efficient BGP simulator. <http://cbgp.info.ucl.ac.be/>, March 2004.
- [5] GP. Saraph and P. Singh. Traffic engineering using new VS routing scheme. In *IEEE International Conference on Communications (ICC 2004)*, Paris, France, June 2004.
- [6] 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.
- [7] 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.