Performance Evaluation of Pre-computation Algorithms for Inter-Domain QoS Routing

Ahmed Frikha, Samer Lahoud

To cite this version:

Ahmed Frikha, Samer Lahoud. Performance Evaluation of Pre-computation Algorithms for Inter-Domain QoS Routing. International Conference on Telecommunications (ICT), May 2011, Ayia Napa, Cyprus. pp.327 - 332, 10.1109/CTS.2011.5898944 . hal-00642317

HAL Id: hal-00642317

https://hal.archives-ouvertes.fr/hal-00642317

Submitted on 17 Nov 2011

Performance Evaluation of Pre-computation Algorithms for Inter-Domain QoS Routing

Ahmed Frikha and Samer Lahoud
University of Rennes 1, IRISA, 35042 Rennes Cedex, France
Email:{afrikha,slahoud}@irisa.fr

Abstract—**Inter-domain QoS routing is a very challenging problem area. This problem combines the complexity of QoS routing, with the limitations of inter-domain routing, such as domain heterogeneity and information confidentiality. The pre-computation offers a very promising solution for addressing this problem. Although the pre-computation scheme has been investigated in several previous studies for a single routing domain, applying pre-computation on an inter-domain level is not straightforward and necessitates deeper investigation. In this work, we study different algorithms for QoS routing based on pre-computation. First, we investigate an exact algorithm. This algorithm provides an optimal solution for the QoS routing problem. However, its application in large scale networks is not always practical. Second, heuristic solutions are also investigated in this work. Particularly, a detailed study of the ID-MEFPA and the ID-PPPA heuristics is provided. Analytical studies and extensive simulations confirm that the exact algorithm achieves the best success rate, but has a very high computational complexity. The ID-MEFPA heuristic has a lower complexity and provides a success rate always close to the exact algorithm. When inter-domain connectivity is high, the ID-PPPA heuristic is the most appropriate with the lowest computation complexity and a success rate very close to the exact algorithm.**

I. INTRODUCTION

Diverse new applications such as multimedia data transfer, telephony, and video broadcast are becoming more common over wide area networks. These applications require more and more extensive and varied quality of service (QoS) guarantees. Providing QoS in terms of propagation time, loss rate, bandwidth or jitter to these applications over a multi-domain network is a very challenging problem area. Most research on QoS routing has focused on routing within a single domain. Addressing this problem in an inter-domain context is also strategic as it is required to ensure the end-to-end QoS guarantees. In this case the problem becomes more complex because of the domain heterogeneity and information confidentiality. Content providers, network providers or transport network operators, manage different domains or autonomous systems and implement heterogeneous policies that consolidate their economic interests and confidentiality clauses. Hence, ensuring end-to-end QoS guarantees is a very hard task. Besides, selecting a route that meets the resource needs of such applications is more difficult in an inter-domain level because of the number of nodes involved in the computation. Consequently, the route selection requires more computational time and the response time become longer.

978-1-4577-0024-8/11/$26.00 ©2011 IEEE

There are two alternatives for path computation. The first is the on-demand computation which consists in the computation of a path which satisfies the constraints upon the reception of the QoS request. The second variant is based on pre-computation. Pre-computation consists in computing a set of suitable paths a priori for a pre-determined set of possible constraints. These constraints illustrate different classes of service of application requirements. Then, when a QoS request arrives it attempts to rapidly provide a feasible solution. This alternative provides a rapid response time for the path provisioning.

In this paper, we focus on the pre-computation scheme for inter-domain QoS routing. Different QoS routing algorithms based on pre-computation are studied in this paper. First, the inter-domain pre-computation algorithm pID-MCP [2]-[3] is investigated in this paper. pID-MCP is an exact algorithm. This algorithm was reported to be the most efficient in term of success rate since it can find a feasible path for each QoS request, whenever such path exists. However, the computational complexity of this algorithm is very high. This limits its application is large scale networks. Second, heuristic solutions, such as the ID-PPPA [1] algorithm, are also investigated in this work. In our previous work [1], we showed that ID-PPPA is a very fast algorithm and has a high success rate. This algorithm is suitable for high speed networks. The MEFPA algorithm [4] was reported to be the most efficient intra-domain pre-computation algorithm. In this paper, we extend MEFPA for an inter-domain routing level; we name it ID-MEFPA. Our studies confirm that pID-MCP has the best success rate, but has a very high computational complexity compared to the heuristic algorithms. ID-MEFPA has a lower computational complexity and has a success rate very close to that of pID-MCP. ID-MEFPA always outperforms ID-PPPA. However, when inter-domain connectivity is high, ID-PPPA presents the most appropriate solution with the lowest computational complexity and a success rate very close to the exact algorithm pID-MCP.

The rest of the paper is organized as follows. In section 2, a detailed description of the inter-domain QoS routing problem is presented. Section 3 presents several pre-computation algorithms and gives a detailed analysis of their complexities. Simulation results are presented in detail in section 4. Finally, conclusions are given in section 5.

II. THE INTER-DOMAIN QOS ROUTING PROBLEM

Inter-domain QoS routing, also known as inter-domain multi-constraint routing, consists of computing a path subject to multiple QoS constraints between a source and a destination node of a multi-domain network. Computing such a path requires knowledge of the topology of each domain in the network, as well as the QoS metrics on network links. As the operators can be in competition, information about the internal topology or the available resources in the network is confidential. This complicates the computation especially when using a centralized method.

Let us introduce the requisite notation to formally define the problem. Let $G(N, E, D)$ denote a network of D domains, N is the set of nodes and E the set of links. Let m be the number of QoS constraints. Note that, in our study, we consider only additive metrics, such as cost and delay, without loss of generality. In fact, multiplicative metrics, such as loss rate, can be translated into additive metrics using the logarithm function; and bottleneck metrics, such as bandwidth, can be resolved by omitting all links which violate the constraints and then computing the path on the residual graph. An m-dimensional weight vector is associated with each link $e \in E$. This vector consists of m non-negative QoS weights $w_i(e)$, $i = 1..m$. Let p be a path in the graph $G(N, E, D)$ and $w_i(p)$ be the weight of p corresponding to the metric i. As metrics are additive, $w_i(p)$ is given by the sum of the i^{th} weights of the component links of path p: $w_i(p) = \sum_{e_j \in p}(w_i(e_j))$. Let $\overrightarrow{W}(p) = (w_1(p), w_2(p), ..., w_m(p))$ denote the weight vector of the path p.

Definition 1
Given a source node s and a destination node d and a set of constraints given by the constraint vector $\overrightarrow{C} = (c_1, c_2, .., c_m)$, the Inter-Domain Multi-Constraint Path (ID-MCP) computation problem consists of finding a path p which satisfies $w_i(p) \leq c_i, \forall i \in 1..m$. Such a path p is called a feasible path.

The ID-MCP problem is $\mathcal{NP}$-hard [5] and may have zero, one, or multiple solutions. A feasible path p is called non-dominated if there does not exist a path p' which satisfies: (1) $w_i(p') \leq w_i(p), \forall i \in 1..m$ and (2) $\exists j \in 1..m$ for which $w_j(p') \prec w_j(p)$. To speed up the computation, dominated paths can be discarded from the computation search space of the QoS routing algorithms without affecting their performance, according to [6].

Currently, the inter-domain routing protocol is BGP. This protocol cannot solve the ID-MCP problem since it does not take into account QoS constraints. Many extensions for BGP are proposed to support QoS routing [7]-[8]. However, the QoS capabilities of these propositions remain limited. Solving the ID-MCP problem using a centralized method is a very complex problem, for the aforementioned reasons. Therefore, the research community has recently been exploring the use of distributed architectures to solve this problem, such as the PCE (Path Computation Element) architecture [9]. Distributing the computation over domains preserves confidentiality across domains and solves the scaling problem. To our knowledge, few works have been proposed to solve the ID-MCP problem using distributed methods. ID-MCP [10] is a distributed algorithm which extends the exact algorithm SAMCRA [6] to an inter-domain level to solve the ID-MCP problem. The drawback of this algorithm is its high complexity. Also, work in [11] proposes a promising distributed solution with crankback mechanisms for inter-domain routing. However this solution cannot take into account more than one QoS metric. Besides, these solutions are based on an on-demand computation scheme which presents some serious limitations with the emerging applications in the Internet. As explained, the ID-MCP problem is $\mathcal{NP}$-hard, consequently, the performance of the on-demand routing algorithms in terms of response time are severely affected.

In this work, we focus on the pre-computation scheme to solve the ID-MCP problem while speeding up the response time. The pre-computation proceeds in two phases: It prepares in advance a set paths satisfying predetermined QoS requests in a first phase. Then, at the reception a of QoS request, it seeks to rapidly provide a feasible path among the pre-computed paths. Ideally, the second phase needs only to select one of the pre-computed solutions. However, some additional computations may be performed. For instance, when handling QoS requests with delay constraints, the first phase may pre-compute feasible paths for a wide range of possible delay constraints, while the second phase just needs to select a suitable path from the pre-computed set, *i.e.*, one that satisfies the particular delay constraints of the request. The execution time of the second phase has an immediate impact on network performance; hence, it is highly desirable to keep its computational complexity as low as possible. In the above example, the less time is consumed in finding the proper path, the less time is spent in establishing the new request.

Up to now, most research on inter-domain QoS routing are based on an on-demand computation scheme, but rarely on a pre-computation scheme. Work in [2] proposes a pre-computation based algorithm to solve the ID-MCP problem. This algorithm, named pID-MCP, relies on a distributed architecture. However, its response time remains high, and its applicability is not practical because of the high complexity of its pre-computation phase.

III. PRE-COMPUTATION ALGORITHMS FOR INTER-DOMAIN QOS ROUTING

In this section, we investigate different pre-computation algorithms for inter-domain QoS routing. Precisely, we detail the operations performed by the exact pre-computation algorithm pID-MCP [2]-[3], our algorithm ID-PPPA proposed in [1], and our novel pre-computation algorithm, named ID-MEFPA. Note that these pre-computation algorithms rely on a distributed architecture, such as the PCE architecture. These algorithms consist of two phases: an offline phase and an online phase. In the offline phase, the algorithms pre-compute a set of QoS paths in each domain separately and satisfying a set of predetermined QoS requests. In the second phase, the algorithms attempt to compute an end-to-end path by combining the paths pre-computed in each domain.

A. The offline phase

This phase is executed in each domain separately. Each algorithm pre-computes a set of paths from each border node of the considered domain toward the other nodes in the domain as well as the entry border nodes of the neighbor domains. These paths will be used later at the online phase to compute an end-to-end path. Note that after an eventual change in the network state, the pre-computed paths may be not valid. Therefore, executing the offline phase periodically or using a network state-dependent threshold is required in each domain. A low complexity of this phase enables to cope with a dynamic change in the network state information since it allows domains to rapidly pre-compute new valid paths. Therefore, in the following we investigate the operations performed by each algorithm during this phase, and compare their offline complexity.

For now, let us introduce the requisite notation. Let D_k be the considered domain, B the number of border nodes of D_k, n_1 a border node of D_k, n_2 a node of D_k or an entry border node of a neighbor domain, and m the number of the QoS metrics. Operations performed by each algorithm during this phase are detailed in the following.

1) The ID-PPPA algorithm: ID-PPPA pre-computes m shortest paths between each pair of nodes (n_1, n_2). Each shortest path minimizes a single QoS metric and is called primary path. Hence, from each border node of D_k, ID-PPPA computes m shortest path trees. Each shortest path tree, also called a primary path tree, is computed using the Dijkstra algorithm and considering a single weight component w_i, where $i \in 1..m$. Therefore, ID-PPPA executes the Dijkstra algorithm m times per border node. For one border node, ID-PPPA is in $O(m(N \log(N) + E))$. Consequently, the complexity of the offline phase of ID-PPPA is in $O(Bm(N \log(N) + E))$.

2) The ID-MEFPA algorithm: ID-MEFPA is our novel proposed algorithm. It is based on the MEFPA proposal [4]. ID-MEFPA computes multiple shortest paths between each pair of nodes (n_1, n_2). Each shortest path minimizes a different linear weighting given by $g_a(e) = \sum_{i=1}^{m} a_i w_i(e)$ where $w_i(e)$ is the i^{th} weight of link e and $\overrightarrow{a} = (a_1, a_2, .., a_m)$ is a coefficient vector. Therefore, ID-MEFPA exectutes at each entry border node n_1 the Dijkstra algorithm and considering the linear weighting g_a. An ideal algorithm would use a continuous variation of $\overrightarrow{a}$ in order to cover all the coefficient values. However, a practical algorithm cannot continuously vary $\overrightarrow{a}$, therefore ID-MEFPA uses V vectors $\overrightarrow{a}$ uniformly distributed in S^m, where $S = \{\frac{0}{(b-1)}, \frac{1}{(b-1)}, .., \frac{b-1}{(b-1)}\}$; b is a parameter of the algorithm.

The complexity of ID-MEFPA depends on the parameter b and the number of constraints m. For one border node, ID-MEFPA is in $O\left(V(N \log(N) + E)\right)$ corresponding to V times the complexity of a Dijkstra algorithm, where $V = C_{b+m-2}^{m-1}$. The global complexity is in $O(BV(N \log(N) + E))$.

3) The pID-MCP algorithm: pID-MCP is an exact pre-computation algorithm; it finds a feasible path, whenever such a path exists. It pre-computes all feasible paths between each pair of nodes (n_1, n_2) by executing the SAMCRA algorithm at each entry border node n_1 of D_k. This computation takes into account a set of C_v predetermined class of service expressed by a set of constraint vectors.

The drawback of this algorithm is its high computational complexity. In fact, the complexity of executing pID-MCP in each domain corresponds to the complexity of executing SAMCRA at each border node of the domain for each class of service. Hence, the complexity of pID-MCP in its offline phase is in $\mathcal{O}(BC_v(K_{max} N \log(K_{max} N) + K_{max}^3 mE))$, where $K_{max} = \min(\exp(N-2)!, \frac{\prod_{i=1}^{m} c_i}{\max_j c_j})$ [12]. This complexity is very high comparing with that of our proposed algorithms ID-MEFPA and ID-PPPA.

B. The online phase

1) Concepts: This phase is triggered upon the reception of a QoS request. The algorithms attempt to compute an end-to-end path by combining the pre-computed paths in each domain. When receiving a QoS request, the service provider computes the best domain sequence that links the source and the destination domain according to the cooperation policy [9]. The end-to-end path computation starts at the destination domain toward the source domain following the selected domain sequence. Note that, without loss of generality, we rely on backward computation according to the PCE architecture.

First, the destination domain selects the pre-computed paths that lead to the up-stream domain following to the selected domain sequence. Then, these paths will be sent to the up-stream domain using a novel compact structure named VSPH (Virtual Shortest Path Hierarchy[1]). This structure contains only the end nodes of the paths (the destination node and the entry border nodes of the up-stream domain) as well as the weight vector of each path. This structure allows the confidentiality of the domains to be preserved. When receiving a VSPH, an intermediate domain combines the paths in the VSPH with the internally pre-computed paths. Then, it selects the feasible paths by comparison with the constraint vector, computes a new VSPH and sends it to the up-stream domain. Finally, the computation is stopped when a feasible path linking the source and destination is found, or when no feasible path is found. In this case, the QoS request is rejected.

2) Complexity: Let $Seq = \{D_1, D_2, .., D_r\}$ denote the selected domain sequence, where D_1 is the destination domain and D_r the source domain. Let α be the upper bound on the number of pre-computed paths between node n_1 and node n_2 in an intermediate domain D_k, $k = 2..r-1$. Note that $\alpha \leq m$ for ID-PPPA, $\alpha \leq V$ for ID-MEFPA and all feasible paths for pID-MCP. Let B_k denote the number of border nodes

---

[1] The hierarchy is a structure which enables the storage of multiple paths between any two nodes [13].

between domains D_{k-1} and D_k, $k = 2..r-1$. At the level of domain D_k, there are at most α^{k-1} paths in the VSPH from the destination d to the node n_1 and at most αB_{k+1} pre-computed paths to reach the upstream domain D_{k+1} from n_1. Hence, the complexity of combining the pre-computed paths with the received paths at the level of the entry border node n_1 is in $O(\alpha^k B_{k+1})$. Considering all entry border nodes between D_k and D_{k-1}, the global complexity of this phase is given by $O(\alpha^k B_{k+1} B_k)$.

We note that the complexity of the online phase depends on the number of pre-computed paths α: The lower the number of pre-computed paths is, the lower is the number of combination operations. Therefore, the complexity of the ID-PPPA and ID-MEFPA algorithms in the online phase is lower than that of pID-MCP.

IV. SIMULATION AND ANALYSIS

In this section, we evaluate the performance of the pre-computation algorithms ID-PPPA, pID-MCP, and our novel algorithm ID-MEFPA. We consider the following topologies:

- LatticeFM(N, D) is a chain of D identical domains. Each domain is a grid made up of N nodes and $E = 2\sqrt{N}(\sqrt{N}-1)$ undirected links. This topology represents the worst-case for the complexity of the algorithms. Precisely, each node of an intermediate domain is connected to each node in the next and in the previous domain of the chain.
- LatticeSL(N, D) is similar to the LatticeFM topology. The only difference between these two topologies is the inter-domain connectivity. In LatticeSL, only the bottom-right node of each domain is connected to the top-left node of the next-domain.
- SYM-CORE is an inter-area topology. This topology consists of five interconnected areas and is taken from the work [14].

In the simulations, we select a number of node N equal to 25 nodes and a number of domains D equal to 3. The LatticeFM(25,3) and LatticeSL(25,3) topologies allow us to evaluate the effect of inter-domain connectivity on our algorithms. In fact, in the LatticeFM (25,3) topology, the number of links that connect two domains is equal to 625, while in the LatticeSL (25,3) topology this number is equal to 1. The SYM-CORE topology is a realistic topology and represents an intermediate case of inter-domain connectivity degree.

We associate with each link three additive weights generated independently following a uniform distribution [10, 1023]. The QoS constraints are also randomly generated in the constraint generation space Z shown in figure 1. This space is deduced by computing p_1, p_2, and p_3 the three shortest paths which minimize the first, the second, and the third metric, respectively. The problem is not $\mathcal{NP}$-Hard outside Z, i.e. either infeasible or trivial. As shown in figure 1, we select ten zones Z_i, $i = 1..10$ from this space and we browse them from the strictest constraint zone Z_1 to the loosest constraint zone Z_{10}. Then, we assess the performance of the algorithms according to these zones.

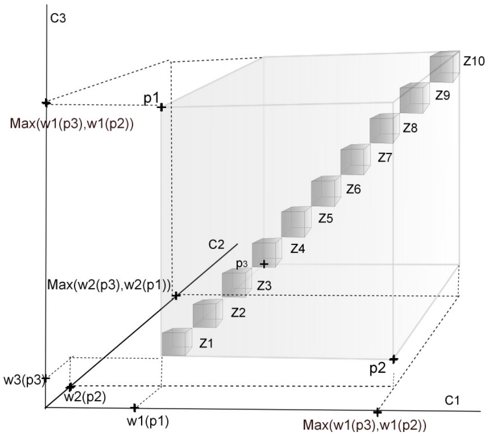

Fig. 1. Constraint generation zones for $m = 3$

In the following, we evaluate the pre-computation algorithms according to two performance criteria. First, the Success Rate (SR) given by the percentage of the requests for which a feasible path is found. Second, the Cost (C) given by the value of the path length function $c(p) = 100 * \max_{i \in 1..m}(\frac{w_i(p)}{c_i})$ for the best path computed. The cost (C) is considered only for the request for which all the algorithms find a feasible path. This performance criteria indicates the quality of the computed path since it gives a measure of the distance between the path weights and the constraints.

Figures 2, 3 and 4 illustrate the variation of the success rate according to the strictness of the QoS constraints in the LatticeSL(25,3), LatticeFM (25,3) and SYM-CORE topologies, respectively. The number of constraints m is equal to 3. The figures show that the success rate measured on the LatticeFM (25,3) topology increases faster than the one measured on the SYM-CORE and LatticeSL(25,3) topologies when the constraints are increasingly loose. As the number of border nodes depends on the inter-domain connectivity, the number of pre-computed paths by each algorithm increases when inter-domain connectivity is high. Thus, the number of feasible paths communicated among domains increases. Therefore, the probability to find an inter-domain path which satisfies the QoS constraints is higher when inter-domain connectivity is high. This is why the success rate of the ID-MEFPA and ID-PPPA algorithms is equal to the success rate of the exact algorithm pID-MCP in the LatticeFM(25,3) topology. We remark that in the LatticeSL(25,3) and SYM-CORE topologies, the success rate of ID-PPPA is lower than that of ID-MEFPA and pID-MCP. In fact, the number of paths pre-computed by ID-MEFPA and pID-MCP is higher than that of ID-PPPA. An important result deduced from these figures is that in all the used topologies the success rate of our proposed algorithm ID-MEFPA is equals or slightly lower than that of the exact algorithm. This is can be considered as a promising result since the computational comlexity of ID-MEFPA is very low comparing with that of pID-MCP.

Now we focus on the quality of the computed path by each algorithm. Figures 5, 6 and 7 illustrate the variation of the

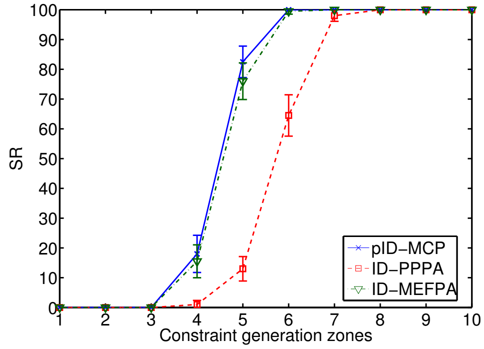

Fig. 2. The success rate according to the constraint generation zones in the LatticeSL(25,3) topology

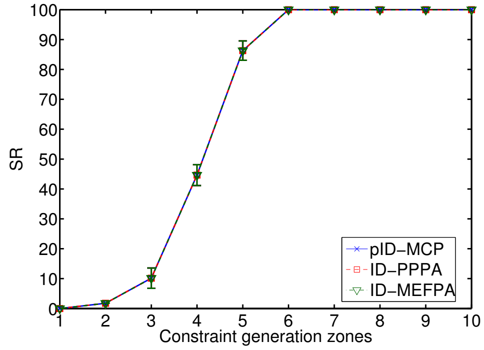

Fig. 3. The success rate according to the constraint generation zones in the LatticeFM (25,3) topology

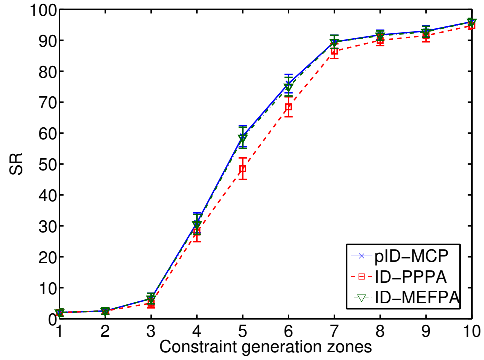

Fig. 4. The success rate according to the constraint generation zones in the SYM-CORE topology

cost C depending on the strictness of the QoS constraints in the LatticeSL(25,3), LatticeFM (25,3) and SYM-CORE topologies, respectively. The quality of paths computed by pID-MCP is almost the best in all used topologies. In fact, pID-MCP inherently minimize the cost C when pre-computing paths. Comparing these three figures illustrates the effect of inter-domain connectivity on the path quality for each algorithm. The higher the inter-domain connectivity is, the better the quality of paths is. The quality of paths is almost the same for all the algorithms in LatticeFM(25,3) topology. However, in LatticeSL(25,3) and SYM-CORE topologies the

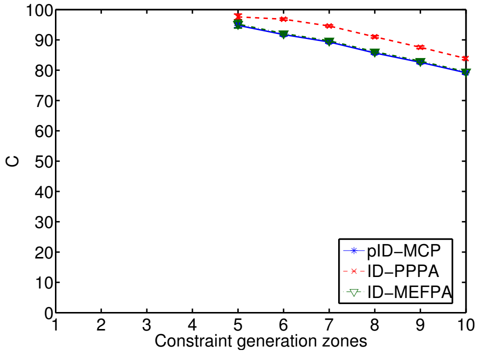

Fig. 5. The cost C according to the constraint generation zones in the LatticeSL(25,3) topology

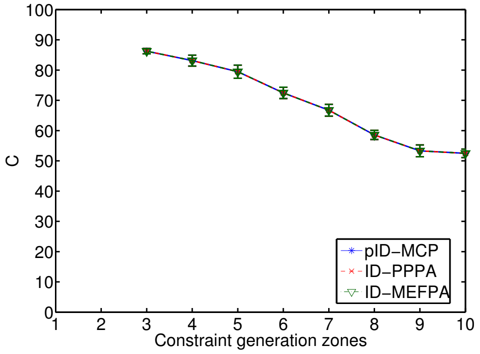

Fig. 6. The cost C according to the constraint generation zones in the LatticeFM(25,3) topology

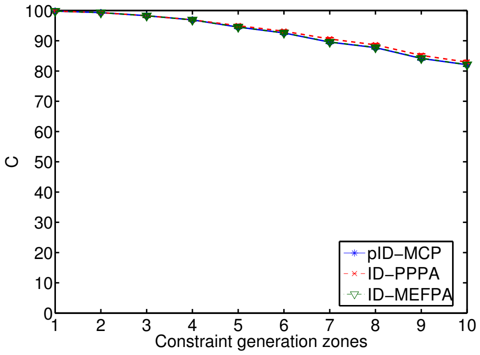

Fig. 7. The cost C according to the constraint generation zones in the SYM-CORE topology

quality of paths computed by ID-MEFPA is slightly better than that of ID-PPPA. This difference in the quality of paths between algorithms is due to the number and the quality of the pre-computed paths.

V. CONCLUSION

In this paper, we have investigated and evaluated performance of several pre-computation algorithms for inter-domain QoS routing. These algorithms rely on a distributed architecture, such as the PCE architecture, which allows domain confidentiality to be preserved. The pID-MCP is an exact algorithm and has the best success rate. However, the complexity of this algorithm in its offline phase is very high. Thus, the time required for pre-computing new paths in each domain is high and this limits the application of pID-MCP in a very dynamic traffic load. In contrast with pID-MCP, ID-PPPA has a low complexity in its offline phase and this allows pre-computations to be executed rapidly to deal with eventual changes in the network load. However, the success rate of ID-PPPA is lower than that of pID-MCP especially when the inter-domain connectivity is very low. The ID-MEFPA algorithm has a success rate very close to that of the exact algorithm pID-MCP in all the considered topologies. Moreover, its offline complexity is very low comparing with pID-MCP. Besides, the complexity of the online phase of ID-PPPA and ID-MEFPA is lower than that of pID-MCP. Thus the response time of ID-MEFPA and ID-PPPA is lower than that of pID-MCP.

REFERENCES

[1] A. Frikha, and S. Lahoud, Pre-computation Based Heuristic for Inter-Domain QoS Routing, *IEEE Advanced Networks and Telecommunication Systems (ANTS), 2010. Accepted for publication*

[2] G. Bertrand, S. Lahoud, M. Molnar and G. Texier, Inter-Domain Path Computation with Multiple QoS Constraints, *Recent Advances in Providing QoS and Reliability in the Future Internet Backbone, Nova Science*, Hauppauge, NY, USA, 2010.

[3] G. Bertrand, S. Lahoud, M. Molnar, and G. Texier, Inter-Domain Path Computation with Multiple Constraints. *In IRISA,* Tech. Rep. 1902, 2008, http://hal.inria.fr/inria-00319401.

[4] C. Yong, X. Ke, W. Jianping: Precomputation for multiconstrained qos routing in high-speed networks. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, IEEE Societies,* vol. 2, pp. 1414-1424 vol.2 (2003).

[5] Z. Wang and J. Crowcroft, Quality-of-Service Routing for Supporting Multimedia Applications, *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228-1234, 1996.

[6] P. Van Mieghem and F.A Kuipers, Concepts of exact QoS routing algorithms, *IEEE/ACM Transactions on Networking*, vol. 12, no. 5, pp. 851-864, October 2004.

[7] T. Knoll, BGP Extended Community Attribute for QoS Marking, *draft-knoll-idr-qos-attribute-02, work in progress, IETF*, 2009.

[8] D. Griffin, J. Spencer, J. Griem, M. Boucadair, P. Morand, M. Howarth, N. Wang, G. Pavlou, A. Asgari, and P. Georgatsos, Interdomain routing through QoS-class planes, *IEEE Commun. Mag.*, vol. 45, no. 2, pp. 88-95, February 2007.

[9] A. Farrel, J.P. Vasseur, J.A Ash, Path Computation Element (PCE)-Based Architecture, *IETF RFC 4655*, August 2006.

[10] G. Bertrand, S. Lahoud, G. Texier, and M. Molnar, A Distributed Exact Solution to Compute Inter-domain Multi-constrained Paths, *The Internet of the Future, Lecture Notes in Computer Science*, vol. 5733, no. ,pp. 21-30, 2009.

[11] M. Esmaeili, F. Xu, M. Peng, N. Ghani, A. Gumaste and J. Finochietto, Enhanced crankback signaling for multi-domain IP/MPLS networks, *Computer Communications*, vol. 33, no. 18, pp. 2215-2223, 2010.

[12] Van Mieghem, P. , De Neve, H., and Kuipers, F. A., *Hop-by-hop quality of service routing*, Computer Networks, 2001, 37, 407-423

[13] M. Molnar, Hierarchies for Constrained Partial Spanning Problems in Graphs, *IRISA*, Tech. Rep. 1900, 2008.

[14] K.I. Calvert, M.B. Doar, and E.W. Zegura, Modelling Internet Topology. *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160-163, June 1997.