Distributed E2E QoS-Based Path Computation Algorithm Over Multiple Inter-Domain Routes

Nabil Bachir Djarallah, Nicolas Le Sauze and Hélia Pouyllau
Alcatel-Lucent Bell Labs France
Centre de Villarceaux
91620 Nozay, France.
{nabil_bachir.djarallah, nicolas.le_sauze, helia.pouyllau}@alcatel-lucent.com

Samer Lahoud and Bernard Cousin
IRISA, University of Rennes 1
Campus de Beaulieu
35042 Rennes, France.
{samer.lahoud, bernard.cousin}@irisa.fr

***Abstract*—[1]Internet usages have changed with the emergence of value added services relying on a higher interactivity and needs for a better quality of experience (QoE). Telecommunication operators have to face a continuing growth of new types of Internet traffic (video, games, telepresence, etc.) imposing not only a more efficient utilization of their network infrastructure resources, but also the generation of new revenues to pursue investments and sustain the increasing demand. Such services generally cross multiple domains, but inter-domain routing protocols still have some limitations in terms of service assurance. For example, BGP's single route announce for a destination limits potential traffic engineering features (e.g. no quality of service price/efficiency optimisation, inter-domain shared route protection, inter-domain load balancing, etc.). In order to provision end-to-end inter-domain connections that obey to constraints such as bandwidth, delay, jitter, and packet loss for these services, an interesting approach is to compute end-to-end (e2e) paths over multiple inter-domain routes. This will allow establishing more efficiently the inter-domain connections with respect to requested QoS constraints and sharing these constraints (and associated revenues) among multiple operators to globally accept more demands in the system, while keep satisfying the customer QoE. To address these challenges, we propose an efficient distributed inter-domain algorithm that computes such constrained paths among a set of domains, exploring multiple inter-domain routes. We demonstrate that our algorithm not only increases success rate in delivering feasible paths, but also admits more connections and keeps a reasonable runtime.**

I. Introduction

The emergence of new value-added services (e.g. Gaming On Demand, VoIP, Cloud Computing, etc.) adds further Quality of Service (QoS) requirements over networks (e.g. bandwidth capacity, transmission delay, availability, security, etc.). Being deployed at a large scale, such services exceed boundaries of a single network domain and thus extend the QoS-enabling challenges across several independent domains. These challenges relate mainly to the heterogeneity of network operators, the scalability issues; refresh of the QoS resource information at large scale, confidentiality on network resource states and network topologies, network policy and economic challenges related to benefits sharing between different actors (e.g. network providers and Over The Top *"OTTs"*), agreements on QoS-based services between involved participants, etc. This paper investigates on some of these issues.

In terms of communication networks, a value-added service is seen as a network path from a source node to a destination node that satisfies one or more QoS constraints. The computation of inter-domain constrained paths requires to figure out different issues at protocol and algorithmic levels. In this paper, we focus on the algorithmic part in charge of this computation. However, the choice of the inter-domain route toward the destination has an important impact on the quality of the desired constrained paths that allow providing guaranteed services. Exploring more than one inter-domain route is one solution to overcome complexities related to the choice of the routes and to increase success rate in delivering feasible paths. In the present work, we propose to explore in parallel divers inter-domain routes which are likely to ensure the connection between the source and destination. Several applications can benefit from the fact that divers routes are explored and different paths respecting constraints are computed, especially the inter-domain load balancing, the inter-domain shared path protection, etc.

The problem of computing QoS-constrained paths is known as the Multi-Constrained Path (MCP) problem [8]. Its extension to an optimization problem, called Multi-Constrained Optimal Path (MCOP) problem has also been extensively studied in the literature [2], [16]. In this article, we are interested in solving this problem in the inter-domain context where the objective function (e.g. generated profit, path cost, etc.) would have been agreed among a set of federated domains, and multiple inter-domain routes would be explored in order to derive the optimal inter-domain path and eventually alternate ones avoiding certain domains. Furthermore, to be compliant with operators requirements on confidentiality, we intend to provide an efficient distributed algorithm that solves accurately the MCOP problem in a multi-domain context. Efficiency is strictly translated into "optimality" to denote paths that achieve efficient utilization of the network infrastructure resources.

The remainder of the paper is structured as follows: in Sec. II, we highlight the general context and assumptions related to the inter-domain issues. Sec. III details the inter-domain multi-constrained path computation over multiple domain routes

[1]This work has been partially supported by the ETICS-project, funded by the European Commission through the 7th ICT-Framework Program. Grant agreement no.: FP7-248567 Contract Number: INFSO-ICT-248567.

problem. A formal definition and complexity classification of our problem are then provided. In Sec. IV, we review the most related studies. In Sec. V, we introduce an efficient exact algorithm to compute feasible end-to-end constrained paths over multiple inter-domain routes while also incorporating the optimization of the selected feasible paths. In Sec. VI, we evaluate by simulation the performance of our proposal. Finally, our main conclusions are drawn in Sec. VII.

II. General Context and Assumptions

In the present study we are interested to constrained path computation in an inter-domain scenarios. The interconnection of several domains forms an inter-domain topology. Fig. 1 depicts a scenario of domains interconnected via specific links, named inter-domain links. As we address the problem of inter-domain multi-constrained path computation over several domain routes, we assume that the destination is reached via several **inter-domain routes**, which could possibly meet the requirements of the e2e request and thereby increase the success rate. An inter-domain route is a sequence of domains that a specific route passes through to reach one destination. For example on Fig. 1 one inter-domain route from D_0 to D_4 can include D_1 and the other one can include D_2.

We assume that each computing entity does not have the whole information about the inter-domain routes but only the next hops able to reach a target domain. The exploration of the routes is performed in a forward and recursive manner by sending the request to the concerned neighbor domains until it reaches the destination. Therefore, the computation process does not need to know the location of the destination in advance because the exploration is done hop-by-hop.

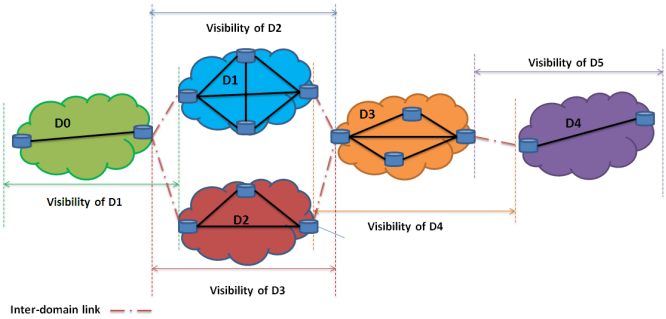

Fig. 1. Reference topology and inter-domain visibility

Confidentiality and Visibility The confidentiality of network resources and topologies between domains, which are managed by separate administrative entities, is very important and must be conserved, particularly when it is matter of inter-domain network connectivity services that require a certain level of cooperation between these domains. Thus, administrators of domains would never exchange detailed information about their own networks (e.g. capacities of links and network entities, network topology, etc.). Our model takes into account these constraints and proposes a solution that respects confidentiality on resource states and network topologies. Therefore, every domain should have a limited visibility on the whole interconnected networks. We consider that each domain has a detailed view on its own network nodes (e.g. IP address of each interface), its own links and their traffic engineering specificities and its inter-domain links (information about interfaces on entry border nodes of neighbor domains), as illustrated in fig. 1. Nevertheless, to compute an inter-domain constrained path, a minimum cooperation between the different participant domains is highly needed to compute an end-to-end optimal solution. In this case we propose to exchange abstract information about computed paths (start and end points of the segments and the corresponding cumulated QoS values) in a bilateral way toward the target domain. In this way we do not give information on intermediate nodes neither on QoS attributes associated to the intermediate links.

III. Inter-domain Multi-Constrained Path Computation over Multiple Domain Routes

A. Definitions

In order to define the computation problem of multi-constrained paths over multiple domain routes, we model the N interconnected domains as a set of valued graphs, $G_0 = (V_0, E_0) \times ... \times G_i = (V_i, E_i) \times ... \times G_N = (V_N, E_N)$, that form one global network G modeled by a valued graph $G = (V, E)$. Each graph G_i includes a vertex set $V_i(G_i)$ and a set of edges $E_i(G_i)$. The graph G includes a vertex set $V(G)$ and a set of edges $E(G)$ where $E \subseteq V \times V$. To simplify notations we use E_i, E, V_i and V instead of $E_i(G_i), E(G), V_i(G_i)$ and $V(G)$. Vertices represent network nodes, while edges represent communication links.

In the following, we only consider connected graphs without self-loops (an edge which starts and ends at the same vertex) and at most one link between a pair of nodes. A specific link in the set E between nodes u and v is denoted by $e = (u, v)$. Each link $e = (u, v) \in E$ is characterized by an m-dimensional weight vector $w_e[j] = (w_{e,0}, ..., w_{e,m})^T$, where $w_{e,j} > 0$, $\forall\ e \in E$, $j \in [0...m]$ and $m \in \mathbb{N}$. The m components of the weight vector model the m QoS metrics associated to each network communication link.

Multi-constrained inter-domain path. Is an e2e path, denoted $P_{s,t}$, between the network source node s and the destination node t, that crosses at least two domains and satisfies the constraint vector $C[j]$. The path $P_{s,t}$ is a finite sequence of path segments. A path segment can be a simple link between two adjacent nodes or an aggregate of several links within the same domain.

Non-Dominated Paths. The paths meeting a request $q(s, t, C)$ demanding an inter-domain path that respects constraints $C[j]$ where $j \in [0...m]$ between source node s and target node t in the graph G, are called **feasible paths** and denoted $P^+_{s,t}$. In order to reduce the search space and to keep only a sub-set of feasible paths, the authors of [18] used a non-dominance rule. According to this rule, a path p_1 can be discarded when there exists a path p_2 such that $W_{p_2,j} \leq W_{p_1,j}$, for all $j \in [0...m]$, except for at least one j for which $W_{p_2,j} < W_{p_1,j}$. The non-dominance rule is applied

on all nodes, during the path computation phase, to discard dominated paths. We denote $P^*_{s,t}$ the set of non-dominated paths found on the destination node.

B. Problem Statement

The multi-constrained inter-domain path computation over multiple domain routes problem can be defined as finding paths that obey to the constraints vector $C[j]$ (where $C[0]$ is the bandwidth constraint and the other weight components are the QoS additive metrics) and respect the non-dominance rule, from the source node s to the destination node t over a set of inter-domain routes S.

When an optimal path is required, an optimization can be performed in order to identify the optimal path $\hat{p}_{s,t}$. The selection of such path is done through an objective function $Z(P^*_{s,t}) = \{z(p) \setminus \forall p \in P^*_{s,t}\}$. This function can take several forms according to the policy adopted by participant domains (e.g. path cost, generated profit, remainder bandwidth, etc.). This is known as the Mult-Constrained Optimal Path problem (MCOP). In our inter-domain context we call this problem, Inter-Domain Multi-Constrained Optimal Path Over Multiple Domain Routes (ID-MCOP-MDR) problem. Consequently, the ID-MCOP-MDR problem can be expressed as follow:

$$\min_{\forall p \in P^*_{s,t}} Z(P^*_{s,t}) \tag{1}$$

subject to,

$$\forall\ p \in P^*_{s,t}, \qquad W_{P,j} = \sum_{\forall e \in p, j=1}^{m} w_{e,j} \leq C_j \tag{2}$$

$$\min_{\forall e \in p} w_{e,0} \geq C_0 \tag{3}$$

Equation (1) expresses the selection of the optimal path from the set of non-dominated paths $P^*_{s,t}$ computed over the S inter-domain routes. Equation (2) expresses the additive resource constraints on selected path segments within the different inter-domain routes. Equation (3) shows the resource constraints on local bandwidth associated to each non-dominated path.

Objective Functions. Let us consider an ID-MCOP-MDR problem. Let $P^*_{s,t}$ be the set of feasible non-dominated paths and $Z : P^*_{s,t} \rightarrow \mathbb{R}^+$ be the objective function. The goal of the optimization problem is to find a path $\hat{p}_{s,t} \in P^*_{s,t}$ whose value assignment maximizes/minimizes the objective function Z. Solution quality refers to the objective function value $Z(p_{s,t})$ of a given non-dominated path solution $p_{s,t}$, where $p_{s,t} \in P^*_{s,t}$.

C. Problem Classification

The ID-MCOP-MDR problem is a MCOP prolem which is classified as NP-complete (e.g.[8]). The only difference between our problem and the MCOP one, is the context of solving the problem, i.e. constraints that force us to solve the problem by parts (per-domain); if there were no confidentiality constraints or management restrictions between domains and if a centralized entity (that would have a global vision of all networks resources and their states) exists, it will be exactly the same as a MCOP problem in mono-domain.

D. Business impacts on inter-domain path computation

In addition to technical constraints, business policies might influence the way that inter-domain paths are computed. Business constraints could be, for instance, the calculation of paths that pass only by neighbor domains with whom he has already economic agreements. A policy could be also related to the service pricing: some domains vary their price according to the time of a request; others do not. In this paper our focus is on technical constraints and how we can compute multiple inter-domain paths subject to these constraints, than economic and political constraints. In [15] we discussed in detail economic policies for inter-domain services.

IV. Related Work

A variety of constrained path computation problems, with multiple weights, satisfying the QoS demands have been long studied, particularly in a mono-domain context [4], [10]. In addition, different recent methods are compared both theoretically and experimentally in [21]. In order to classify different solutions, we distinguish three families of solutions : heuristic, approximate and exact solutions. 1) Heuristic solutions aim at identifying a feasible solution not necessary optimal, but simple to implement [8], [16], [20]. 2) Approximate solutions are those heuristics that additionally provide some bounds on error. In the best cases, the approximation is optimal up to a given small constant factor ϵ [17], [2]. 3) Exact approaches attempt to compute optimal solutions with the risk of possible high complexities and runtimes that could grow exponentially in the worst case. However, in practice, this time can be reduced to a finite polynomial time [13], [12], [11].

All these propositions have in common that they solve the MCP/MCOP problems only in mono-domain context. Other works have addressed the same problems within the inter-domain specificities. Authors of [5] presented a minimum price inter-domain routing algorithm based on min-plus convolutions, that selects the cheapest paths among a number of independent domains satisfying the end-to-end QoS constraints. Algorithms based on exchanging link state information are listed in [14] and [19]. These algorithms have in common that they exchange some QoS information about inter-domain links and on intra-domain links, in a special case. Two major problems are faced here : confidentiality problems between different domains when QoS informations are conveyed and the colossal amount of information that could be exchanged between routers. Another algorithm presented in [14], based on parallel probe packets sent through the network to collect routing information. Two levels of packet probing are identified ; intra-domain probe packets and the inter-domain probe packets. The intra-domain information collected probe packets are aggregate and sent to the downstream domain. In doing so, networks preserve their confidentiality. Other solutions like those in [3], [1] and [7] propose to solve the inter-domain MCP/MCOP problems using an extended SAMCRA's algorithm [13]. These solutions work on one inter-domain route assumed known in advance. In this paper, we are interested on solving the problem through different

inter-domain routes instead of only one pre-determined route. Therefore, we propose a distributed algorithm that computes constrained end-to-end paths over multiple inter-domain routes and takes into account the architecture we proposed in [6].

V. Inter-Domain Multi-Constrained Optimal Path Algorithm Over Multiple Domain Routes

This section details our proposal for a distributed algorithm, called Inter-Domain Multi-Constrained Optimal Path Over Multiple Domain Routes (ID-MCOP-MDR) algorithm. The ID-MCOP-MDR algorithm exactly computes inter-domain paths over multiple domain routes subject to m QoS constraints and optimizes an objective function over those paths. Further in this section, we only consider the minimization of an objective function, denoted $Z(P^*_{s,t})$.

A. Basic Principles

The work of Van Mieghem and Kuipers on the MCP problem, in an intra-domain context [11], has inspired our work on ID-MCOP-MDR algorithm, which is based on four key principles.

Length function. The ID-MCOP-MDR algorithm uses a non-linear length function that combines weights and constraints as defined by equation (4). This length function is more efficient and gives higher success rate to find the feasible path than a linear energy function that combines QoS metrics of paths [9]. Furthermore, when the best path is requested, the function (4) is minimized providing the optimal path.

$$l_\infty(p) = \max\left[\frac{w_1(p)}{C_1}, \ldots, \frac{w_m(p)}{C_m}\right] \tag{4}$$

k-shortest path storage. Instead of storing one shortest path, each intermediate node - including border nodes - of crossed domains stores k-shortest paths. Diversity of paths at ingress border nodes, makes it possible for network administrators to choose the input nodes offering the best response to internal policies. The k-shortest paths stored on nodes are bounded by the number of feasible paths, denoted k_{max}, as identified by equation (5) [12]. Section VI details experimental results showing that k_{max} can be reduced in practice.

$$k_{\max} = \min\left[\frac{\prod_{j=1}^{m} C_j}{\min_{1\leq j\leq m} C_j}, \lfloor e(N-2)!\rfloor\right] \tag{5}$$

Non-dominance. To reduce research space and consequently decrease computational complexity, each intermediate node keeps only non-dominated paths (Sec. III-A).

Path segmentation. Confidentiality aspects are preserved between neighboring domains and the others of the inter-domain routes. Once non-dominated paths are computed within intermediate domains, the computation unit sends abstract paths (called path segments further in this paper) instead of the explicit full paths. Path segments are delimited by the source node and the appropriate ingress node. Other information related to the path segment, such as QoS characteristics and path cost are transmitted. Then, path segments are concatenated to the domain graph once they are received.

B. Description of the ID-MCOP-MDR Algorithm

This section details the meta-code of the ID-MCOP-MDR algorithm. The request is to compute one or more paths from a source node s to a destination node t, subject to a constraint vector C_j, where $j = 0, ..., m$, that minimize a cost function $Z(P^*_{s,t})$. Other input parameters are recovered locally, such as the network graph of the source domain D_s and the set of neighboring domain's path segments $PathSegs_{i-1}$. The procedure starts initializing previous path segments to ϕ and then calls the subroutine *ID-MCOP-MDR*, described by algorithm 1, to trigger the computation over different inter-domain routes. A timer is activated until reception of potential non-dominated paths through the different inter-domain routes. Once the timer expires, subroutine *Compute_GlobalOptimumPath* (algorithm 2) performs a global optimization (in our case, minimization of $Z(P^*_{s,t})$ = minimization of $l_\infty(P^*_{s,t})$).

The subroutine *ID-MCOP-MDR* (algorithm 1) contains two different treatments: one is achieved by intermediate domains, another by the target domain D_t. Intermediate domain processing is exhibited by lines 1 to 7 of algorithm 1. It begins with the concatenation of previous path segments (initially empty) to the present network graph (line 2 of algorithm 1). Each path segment starts with the source node s and ends with a border node. The concatenation allows the computation of constrained segments, still, from the same node s to an ingress border node of a next domain. Ingress border nodes are extracted by the function *IngressNodesOfDownstreamDomains*, that uses as an input (using the function *Next(D_i, D_t, t)* in line 3) downstream domains, which are able to reach the target t. The set Ω of ingress border nodes, is then used at line 4 by subroutine *Compute_Non-dominatedPaths* to compute non-dominated path segments within the current domain, which are transmitted to neighbor domains that could reach the destination, at lines 5-7.

The second part, illustrated between lines 8 and 13, differs from the first one by replacing the ingress nodes of next neighbor domains by the target node t (line 10) and by sending back the path segments to the source domain (line 12) - instead of to neighbors - in order to perform a global optimization.

The subroutine *Compute_Non-dominatedPaths* computes intermediate path segments between the source node s and the nodes of Ω. It takes into account the requested bandwidth and other QoS constraints. The exploration of network nodes is ordered by a priority queue, *PQ*, which is initialized to ϕ at line 1 of the algorithm 2 and starts with the source node at line 2. At each step, one node u is extracted from the priority queue *PQ* (line 4) to be treated with its appropriate neighbors (i.e. neighbors that do not create loops if they are chosen (line 5)). Once appropriate neighbors are identified, the subroutine checks for the bandwidth availability between the extracted node and every neighbor node v (line 6). If there is enough bandwidth, the calculation starts (line 7 to 11) otherwise it returns failure message (not enough capacity). Another rule, detailed in sec.III-A, determines if the potential

Algorithm 1 ID-MCOP-MDR($G_i, s, t, D_t, C_j, PathSegs_{i-1}$)

1: **if** $D_i \neq D_t$ **then**
2: $G_i \leftarrow G_i \bigotimes PathSegs_{i-1}$ //graft temporary $PathSegs_{i-1}$ to G_i
3: $\Omega \leftarrow$ *IngressNodesOfDownstreamDomains(Next(*D_i, D_t, t*))*
4: $PathSegs_i \leftarrow$ *Compute_Non-dominatedPaths(*G_i, s, t, Ω, C_j*)*
5: **for** each domain $D^+ \in Next(D_i, D_t, t)$ **do**
6: *ForwardTo(*D^+, s, t, C_j, $PathSegs_i$*)*
7: **end for**
8: **else** $\{D_i = D_t\}$
9: $G_i \leftarrow G_i \bigotimes PathSegs_{i-1}$
10: $\Omega \leftarrow t$
11: $P^*_{s,t} \leftarrow$ *Compute_Non-dominatedPaths(*G_i, s, t, Ω, C_j*)*
12: *SendBackTo(*D_s, $P^*_{s,t}$*)*
13: **end if**

segment between the extracted node u and the node v is dominated, or not, using the segments stored on node v (if they exist).

Note that ID-MCOP-MDR does not generate loops. In each domain, loops are inherently prevented by the minimization objective (equation (1)) of the ID-MCOP-MDR problem: take for example one path containing a loop inside a domain; this loop can be eliminated, leading to a path with lower weight. Therefore, a looping path can not be a minimal path. However, this does not prevent loops at the inter-domain level, i.e. a path that traverses multiple times a domain. The prevention of these loops necessitates further studies and specific mechanisms.

Algorithm 2 Compute_Non-dominatedPaths(G_i, s, t, Ω, C_j)

1: PriorityQueue $PQ \leftarrow \phi$
2: *Insert(*PQ, s, 0*)*
3: **while** $PQ \neq \phi$ **do**
4: $u \leftarrow$ *Extract-Min(*PQ*)*
5: **for** each domain $v \in$ NeighborsToBeChecked(G_i, u) **do**
6: **if** *CheckForBandwidthCapacity(*u, v, C_0*)* **then**
7: $Dominated \leftarrow$ *DominanceRule(*u, v*)*
8: $Length \leftarrow l(\vec{W}_u + \vec{w}_{(u,v)})$
9: **if** not $Dominated$ AND $Length \leq 1$ **then**
10: *UpdateQueue(*PQ, u, v, $Length$*)*
11: **end if**
12: **else**
13: returns computation failure (Not enough capacity)
14: **end if**
15: **end for**
16: **end while**
17: Returns paths on all nodes in Ω

C. Worst-Case Runtime Complexity of ID-MCOP-MDR

Van Mieghem et al. [13] calculate the worst-case runtime complexity of their **one point to one point intra-domain routing algorithm** on which our ID-MCOP-MDR algorithm is based. So, the worst-case runtime complexity of Van Mieghem's algorithm has an order of $O(k.|V|.log(k.|V|) + k^2.m.|E|)$, where k is the number of feasible paths which is bounded by the equation (5), $|V|$, $|E|$ and m are respectively the number of nodes, links and metrics.

As explained before, every occurrence of our algorithm runs on a specific graph $G_i = (V_i, E_i)$. The graphs G_i are expanded by the computed path segments (except the source domain), the entry Border Nodes BN_{i+1} and their inter-domain links of the downstream domains. Therefore, the total number of nodes on which the algorithm operates is equal to the total number of nodes per-domain plus the number of entry BN_{i+1} and eventually the source node (all domains except the starting one). Moreover, the total number of edges is equal to the total number of edges per-domain plus the total number of edges within the path segments and the number of links that join BN_{i+1}. The maximal number of edges within the path segments is equal to $k.BN_{i+1}$ and the maximal number of inter-domain links that join BN_{i+1} is equal to $V_i.BN_{i+1}$.

Consequently, the worst-case runtime complexity within a domain i is in order of $O(k.(1 + |V_i| + BN_{i+1}).log(k.(1 + |V_i| + BN_{i+1})) + k^2.m.(k.BN_{i+1} + |E_i| + |V_i|.BN_{i+1}))$. The number of entry BN_{i+1} is bounded by the maximal number of nodes V_i within a network i.e. every node could be an entry border node, therefore we can replace BN_{i+1} by $|V_i|$. Furthermore, the set of nodes $(1 + |V_i| + BN_{i+1})$ can be replaced by one term $|V_i|$ representing all nodes. We can rewrite the worst-case runtime complexity as follow : $O(k.|V_i|.log(k.|V_i|) + k^2.m.(k.|V_i| + |E_i| + |V_i^2|))$. Thus the complexity of the term $k.|V_i|.log(k.|V_i|)$ is less important than the complexity of $k^2.m.(k.|V_i| + |E_i| + |V_i^2|)$, so the complexity can be reduced to $O(k^2.m.(k.|V_i| + |E_i| + |V_i^2|))$. As the ID-MCOP-MDR algorithms runs on several routes (R) and each route is a set of domains (D), the worst-case runtime complexity will be of $O(R.D.k^2.m.(k.|V'| + |E'| + |V'^2|))$, where $|V'|$ and $|E'|$ represent the biggest domain in terms of nodes and edges.

D. Worst-Case Space Complexity of ID-MCOP-MDR

The space complexity is the storage space required by the algorithm to solve the problem. Our algorithm performs the computation on each crossed domain and on inter-domain routes. Within every domain ID-MCOP-MDR computes essentially a constrained path tree. The constrained path tree of a domain i can include at most : 1) the source, all the nodes within the domain and the entry border node of the downstream domain i.e. $(1+|V_i|+BN_{i+1})$ nodes. 2) all edges of the received path tree, all edges within the domain i and the inter-domain links joining the downstream domain $i+1$, i.e. $(k.BN_{i+1}+|E_i|+|V_i|.BN_{i+1})$. Consequently, the worst-case space complexity is in order of $O(R.D.((1 + |V'| + BN') + (BN'.(k + |E'|)) + 1))$, where $|BN'|$ is the biggest set of border nodes that a domain can contain. This last one can be reduced to $O(R.D.(2 + 2.k.|V'^2| + |V'|.E'))$. In the last domain a sort is performed to select the optimal solution to return

back. Choosing the "Quick Sort", the space complexity is of $O(logP^*)$. So, the worst-case space complexity of ID-MCOP-MDR is in order of $O(R.D.(2+2.k.|V'^2|+|V'|.E')+logP^*)$.

VI. Simulation & Results

In order to evaluate feasibility and efficiency of the proposed exploration model and corresponding computation approach, we conduced several experiments using the ID-MCOPMDR algorithm on a multi-domains scenario.

A. Scenario and metrics

We evaluated the performance of the ID-MCOP-MDR algorithm by using a self-written JAVA network simulator. The network topology is generated based on a Waxman model by using the BRITE software generator. We generated five interconnected domains, exhibited by fig. 1, with 600 nodes and 1200 edges per network domain. Each edge is associated with 3, 4 or 5 QoS metrics (the first one is the bandwidth and others are additive integers), depending on the scenario, in order to observe the impact of the QoS metric number on the algorithm performance. Additive QoS metrics are positively correlated within $[1, 1000]$. The bandwidth capacity of each link is fixed to 10 Gbps. Throughout the simulations, requested bandwidth is set to 64 Mbps.

Two types of constraints are identified; loose and tight constraints. Fig. 2 gives an example of constraint space (with two additive metrics) delimited by two paths p_1 and p_2. The path p_1 minimizes the first metric and p_2 minimizes the second metric. Constraints generated within the zone $z11$ are loose and a polynomial computation algorithm can compute paths with these type of constraints. Constraints generated inside the zone $z0$ are infeasible. Contrariwise, constraints generated in the rectangle delimited by $[l_1(p_1), l_2(p_1)]$ and $[l_1(p_2), l_2(p_2)]$ are tight. This rectangle is partitioned into 10 zones ($z1$-$z10$).

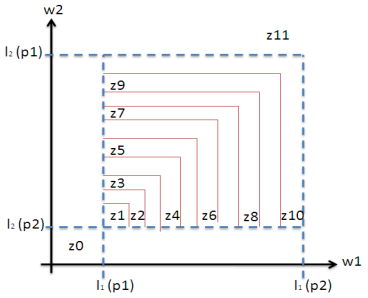

Fig. 2. Constraint generation zones

B. Performance evaluation criteria

The purpose of the simulations is: 1) to study the impact of exploring multiple routes on the number of satisfied requests and the number of solutions per-request, 2) to evaluate the tightness and the number of feasible stored paths (k) on the sucess rate, 3) to study the effect of the number of metrics and the number of feasible stored paths (k) on the generated overhead messages, and finally 4) to show the impact of the number of metrics and the number of feasible stored paths (k) on the runtime. Therefore, the parameters measured in the simulations are the following:

- *Number of paths:* the number of non-dominated paths, on one or more inter-domain routes, related to a request.
- *Average Success rate (ASR):* the average success rate to find the constrained path(s) is decided by the following formula :
$$\text{ASR} = \frac{\text{No. of successful path computation requests}}{\text{Total No. of path computation requests}}$$
- *Average message overhead (AMO):* ID-MCOP-MDR uses control messages to check the availability of resources, exchange of the request, acknowledge a path computation request or terminate a request. These control messages can be considered as network overhead since they consume some network resources. Therefore, given a set of generated requests, for the same couple source/destination, we measure the average message overhead in order to find (if it exists) the constrained path(s).
- *Mean execution time:* is the needed time to response an end-to-end constrained path computation request.

C. Simulation Results

Number of paths. Fig. 3 shows the benefit in terms of satisfied requests and number of constrained paths per-request when we explore one inter-domain route or two routes in parallel. We performed this simulation in three steps : first we launched the computation of 200 requests on the inter-domain route 1; second, the same requests are sent to be computed on the inter-domain route 2; and finally, the exploration of the two inter-domain routes simultaneously with the same requests. In the first case, we had a success of 91 requests from the 200 requests, then slightly less with the inter-domain route 2 (success of 89 requests), and finally, the success of 141 request from 200 requests when the two routes are explored in parallel. Nevertheless, these results are obvious, because the number of success request naturally increases with the increase in the number of explored inter-domain routes. We showed these results to confirm the usefulness of exploring multiple paths.

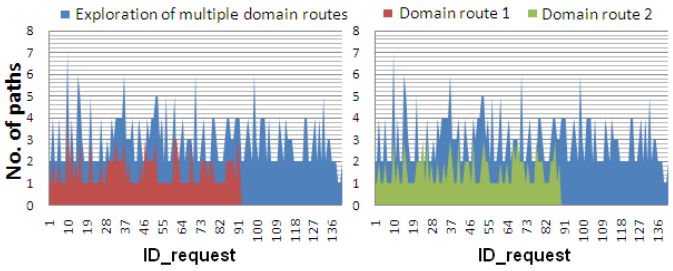

Fig. 3. Multiple domain routes vs single domain route

Average Success Rate (ASR). Fig. 4 illustrates the dependence of the average success rate of our ID-MCOP-MDR algorithm on the constraint generation zones. However, requests

lose their tightness when the constraints are generated close to zone 10. For this simulation, 100 requests were generated within each constraint zone (z0 to z10), knowing that source node is in the domain D_0 and the destination is in the domain D_4 and QoS metrics are: the bandwidth and two additive metrics (positively correlated). We note that when we approach the less restrictive zones the success rate increases, especially the zone 10 where the ASR is at its maximum (100 %).

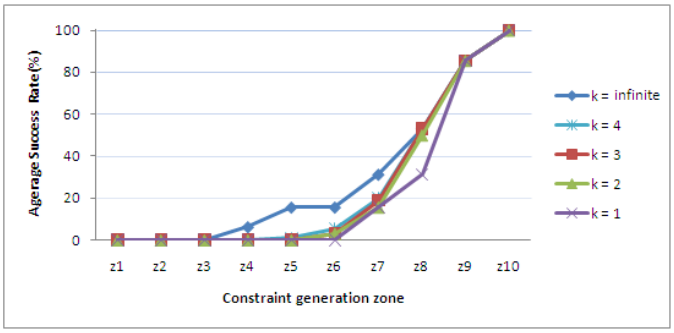

Fig. 4. Average Success Rate vs constraint generation zones

Another parameter that influences the ASR is the number of feasible stored paths k (when this number is infinite the algorithm provides exact solutions). The first satisfied requests are within the generation zone z4. Varying k between 4, 3 and 2 gives almost the same ASR when handling requests generated within the zone z8 and exactly the same ASR within requests of z9 and z10. However, decreasing the value of k to 1 allows to return the same rate as when k is equal to infinite, when requests are generated within the zone z9 and z10. Therefore, we will see later that bounds on the value of k will significantly impact the execution time and the number of message overhead generated by our algorithm.

Average Message Overhead (AMO). Fig. 5 shows the message overhead caused by the control messages through crossed domains D_0 to D_4 with an increasing number of feasible paths k and metrics. In Fig. 5.(a) we generate request (with 3 metrics) within zone z10 (we observed before that requests of this zone gives us a very high success rate) and we run the algorithm. Once the execution ends, we observe that the AMO is the same when the value of k is set to infinite, 4 or 3. This means that the number of paths stored on nodes do not exceed 3 paths even if the size of the queue, of each node, is set to infinite. Contrariwise, this number begins to decrease slightly when the value of k is set to 2. The decrease is more significant when the number of stored paths is limited to one. This is justified by the fact that less paths are explored, so less checks and less message overhead.

In Fig.5.(b) we evaluate the impact of the constraint's tightness and the number of metrics on the generated overhead messages. In this case we fixed the number of feasible stored paths k to infinite and we vary the number of QoS metrics in order to measure the mean number of message overhead per-domain. The number of metrics varies from three to five metrics; in each case, the first metric represents the bandwidth

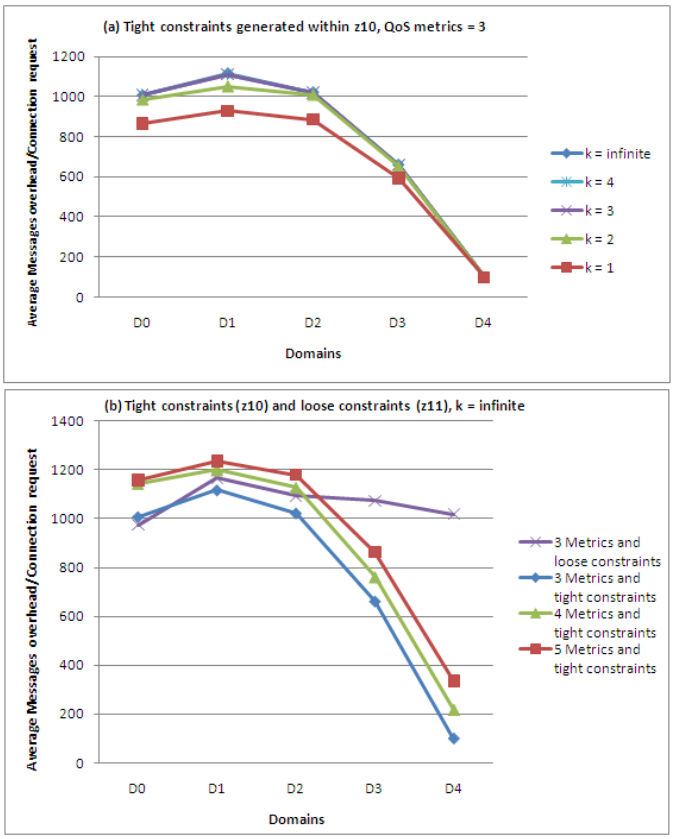

Fig. 5. Average Message Overhead across domains

and the others are the additive metrics. The first series of simulation concerns loose constraints: the AMO varies slightly from 974.61 messages to 1165.92 messages in each crossed domain. This slight variation is explained by the fact that; 1) constraints are so loose that each domain could explore all possibilities in order to calculate path segments and 2) all domains have the same size in terms of number of nodes and links. The same tests are done on tight constraints with increasing of the number of metrics: we noticed that this number tends to increase whenever we increase the number of metrics and decrease whenever we approach the destination domain (D_4), because the values of the remaining constraints are increasingly tight for the last domains. This implies that less paths would meet the constraints, so less checks and less overhead messages. Thus, we can say that reducing the number k to 2 or 1 will enable us to gain rather in terms of overhead messages while keeping almost the same success rate, if constraints are generated within z9, z10 or z11.

Mean Execution Time (MET). In Fig. 6 we evaluate the impact of the number of inter-domain routes, the feasible paths k and the number of metrics on the execution time. The first series of tests concerns the variation of the number of feasible paths k. As we can see in Fig. 6.(a), the MET increases slightly with the change of generation zone and therefore the increase of the success rate. Contrariwise, a significant difference in the execution time exists between the case where

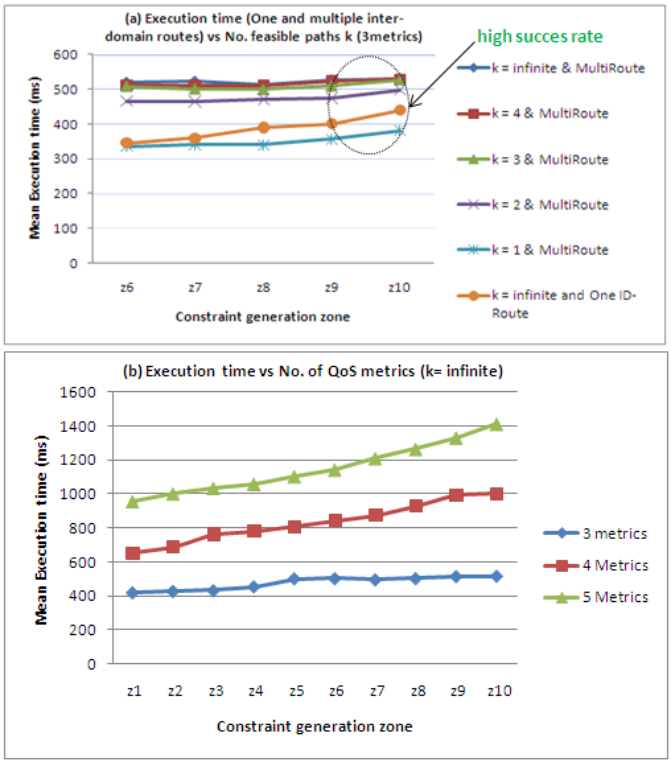

Fig. 6. Execution time evaluations

k = 1 and the other cases, knowing that the success rate is almost the same with k = 1 in zone z9 and z10 as with k = infinite (exact solution). We can see also the execution time related to the exploration case of one inter-domain route. The difference in terms of execution time is minimal compared to the obtained gain, in ASR (simulation results depicted in Fig. 3), when multiple routes are explored. The second series of tests handles the number of metrics to see the behavior of the algorithm and the time required to return a response. In Fig. 6.(b), it is obvious that the MET increases with the change of constraints generation zone because more paths are computed and found that satisfy generated requests. The MET increases also naturally with the variation of the number of metrics.

As a conclusion, we can say that the number of feasible paths k may be reduced to 2 if requests are generated between zone z9 and z10 while maintaining a very good success rate with a reasonable execution time.

VII. Conclusion

In this paper, we presented and discussed an efficient distributed end-to-end QoS-based path computation algorithm. This algorithm has the particularity to explore several inter-domain routes in parallel instead of one pre-determined route, in order to provide exact solutions to the inter-domain constrained path computation problem while respecting the confidentiality rules between crossed domains. The major goal of this algorithm is to avoid the pre-computation of the inter-domain route, to increase the success rate and to return eventually multiple solutions that meet constraints.

We presented simulation results - evaluating success rate, execution time and overhead messages - of the proposed algorithm. These results showed that our algorithm has best performances in terms of success rate, compared to a solution that explore only one inter-domain route. Even though our algorithm takes a little longer time than an algorithm that explores one route, the difference in execution time remains reasonable given the gain in terms of success rate. In addition, we noticed that the number of shortest paths stored on nodes can be reduced in order to reduce the execution time and the overhead messages, while maintaining a good success rate.

References

[1] W. Amari, R. Gadhgadhi, A. Ben Letaifa, and S. Tabbane. Path Computation with Multiple Constraints QoS in Multi-Domain. In *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference*, pages 1–8, May 2010.

[2] O. Ariel. Routing with End-to-End QoS Guarantees in Broadband Networks. *IEEE/ACM Trans. Netw.*, Vol 7:pp 365–374, June 1999.

[3] G. Bertrand, S. Lahoud, G. Texier, and M. Molnar. Computation of multi-constrained paths in multi-domain mpls-te networks. In *Next Generation Internet Networks, 2009. NGI '09*, pages 1 –8, 2009.

[4] I. Cidon, R. Rom, and Y. Shavitt. Multi-Path Routing Combined with Resource Reservation. In *Proceedings of the INFOCOM '97*, Washington, DC, USA.

[5] D. Di Sorte and G. Reali. Minimum price inter-domain routing algorithm. *Communications Letters, IEEE*, 6(4):165 –167, April 2002.

[6] N. B. Djarallah, H. Pouyllau, N. Le Sauze, and R. Douville. Business-driven PCE for Inter-Carrier QoS Connectivity Services. Soon in the conference proceedings for Future Network and Mobile Summit 2011.

[7] A. Frikha and S. Lahoud. Hybrid Inter-Domain QoS Routing based on Look-Ahead Information. Research Report IRISA, No 1946, 2010.

[8] J. M. Jaffe. Algorithms for finding paths with multiple constraints. *Networks*, Vol 14:pp 95–116, 1984.

[9] T. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 834 –843, 2001.

[10] F. Kuipers, T. Korkmaz, M. Krunz, and P. Van Mieghem. Performance Evaluation of Constraint-Based Path Selection Algorithms. *Network, IEEE*, Vol 18:pp 16–23, 2004.

[11] P. Van Mieghem and F. Kuipers. Concepts of exact qos routing algorithms. *IEEE/ACM Trans. Netw.*, Vol 12:pp 851–864, October 2004.

[12] P. Van Mieghem and F. A. Kuipers. On the complexity of qos routing. *Computer Communications*, 26(4):376 – 387, 2003.

[13] P. Van Mieghem, H. De Neve, and F. Kuipers. Hop-by-Hop Quality of Service Routing. *Computer Networks*, Vol 37(3-4):pp 407–423, 2001.

[14] S. Norden. Inter-Domain Routing: Algorithms for QoS Guarantees. *Comput. Netw.*, Vol 49:pp 593–619, November 2005.

[15] H. Pouyllau, R. Douville, N.B. Djarallah, and N. Le Sauze. Economic and technical propositions for inter-domain services. *Bell Labs Technical Journal*, 14(1):185–202, 2009.

[16] T. Sanguankotchakorn and N. Perera. Hybrid Multi-Constrained Optimal Path QoS Routing with Inaccurate Link State. In *Networks (ICN)*, pages 321 –326, 2010.

[17] U. Shuchita and D. Gaytri. Exploring Issues for QoS Based Routing Algorithms. *International Journal on Computer Science and Engineering (IJCSE)*, Vol 02:pp 1792–1795, 2010.

[18] C. E. Leiserson T. H. Cormen and R. L. Rivest. *An Introduction to Algorithms*. Cambridge, MA: MIT Press, 1991.

[19] K. Tae-Il, J. Hae-Won, M. Young Chung, and J. Seong-Il. Inter-domain Routing Based on Link State Information for End-to-End QoS Guarantee. In *Advanced Communication Technology, ICACT*, volume 03, pages 1729 –1732, 2009.

[20] G. Zhuo, Q. Jianzhong, L. Shukuan, and C. Xueliang. A Distributed Parallel QoS Routing Algorithm with Multi-path Probing. In *Control and Decision Conference, CCDC '09. Chinese*, pages 1296 –1301.

[21] M. Ziegelmann. *Constrained Shortest Paths and Related Problems - Constrained Network Optimization*. VDM Verlag, Germany, 2007.