



HAL
open science

Distributed Inter-Domain Multi-Constrained Routing

Mohand Yazid Saidi

► **To cite this version:**

Mohand Yazid Saidi. Distributed Inter-Domain Multi-Constrained Routing. 2018 International Conference on Smart Communications in Network Technologies (SaCoNeT), Oct 2018, El Oued, France. pp.243-248, <10.1109/SaCoNeT.2018.8585445>. <hal-04018744>

HAL Id: hal-04018744

<https://hal.science/hal-04018744v1>

Submitted on 8 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Distributed Inter-Domain Multi-Constrained Routing

Mohand Yazid SAIDI

Université Paris 13 Sorbonne Paris Cité, 99 Av. Jean Baptiste Clément, 93430 Villetaneuse, France
Email: saidi@univ-paris13.fr

Abstract—With recent advances in the communication technologies (5G, LTE, etc.), we are witnessing a deployment of a wide range of network real-time applications like telemedicine, VoIP, remote control applications for drones and cars, e-health, etc. These applications often require the verification or optimization of various quality of service (QoS) parameters, such as the delay, the error rate, the failure probability, the energy, etc. Though several algorithms have been developed to provide QoS in intra-domain networks, there are few works which are devoted to the inter-domain multi-constrained (or multi-criteria) routing. This is essentially due to the difficulty to deal with the two inter-domain major issues which are: scalability and confidentiality.

In this paper, we propose an efficient and distributed multi-constrained routing algorithm for multi-operator or inter-domain networks. Our algorithm guarantees the determination of paths ϵ -close to the optimums. It is fully polynomial time and it respects the inter-domain constraints. Besides, it reduces the message exchanges by aggregating the path weights.

Index Terms—QoS, multi-criteria, multi-constrained, routing, inter-domain network, fully polynomial time algorithm.

I. INTRODUCTION

Today, we are witnessing a wide deployment of connected objects like drones, autonomous cars and sensors. For their operation, these objects often desire to establish with a control station routes verifying multiple constraints and optimizing various criteria like the delay, the failure probability, the energy, the error rate, etc. For instance, sensors which transmit periodically data to a given sink should prefer routes minimizing the energy and failure probability. For drones, the route to the control station should verify delay constraints and optimize the failure probability and energy.

In addition of connected objects, the renewed interest for the *Multi-Constrained Routing* (MCR) can be explained by the widespread deployment of network real-time applications (VoIP, IPTV, teleconference, telemedicine, etc.). MCR aims to provide QoS for the precedent applications by selecting routes which optimize or verify multiple additive QoS criteria.

Due to the NP-hardness of MCR [1], substantial researches have been conducted resulting in many QoS routing approaches [2]. These approaches can be grouped into three categories: exact algorithms, heuristics and guaranteed approximation algorithms. Exact algorithms permit to determine optimal QoS paths at the expense of a high worst case time complexity. Contrarily to the exact algorithms, heuristics are fast but they do not ensure neither the finding of optimal solutions, nor the determination of paths satisfying the QoS criteria. Thus, in order to achieve a tradeoff between the (worst case) execution time and the quality of computed paths,

guaranteed approximation algorithms are developed. These algorithms, referred here as fully polynomial time schemes (FPTASs), ensure the determination of ϵ -approximate paths with fully polynomial time complexity. To solve the MCR problem, FPTASs first discretizes the solution space by choosing a set of samples. Then, at each step of path computation, the real weights of the intermediary paths are approximated by the samples which are the most close to them.

Though multiple algorithms and heuristics were been developed for intra-domain networks, none of them could be applied directly for the inter-domain. Indeed, for efficiency considerations, almost all the QoS path computation algorithms are centralized and assume the knowledge of the network topology. This is not valid in inter-domain networks where the overall network topology is not known, due to confidentiality and scaling constraints imposed by the inter-domain.

Moreover, efficiency of inter-domain MCR algorithms depends closely from the following criteria: (1) the computation quickness, (2) the ability to parallelize the computations and (3) the amount of information to be exchanged for the computations. Generally, the two first criteria are determined by the convergence time and the type of intra-domain path computation algorithms whereas the last criterion depends on the message exchange method and the number of sub-paths that should be transmitted between the Path Computation Elements (PCEs). For instance, the recursive FPTASs in [3] [4] which are fast for the intra-domain routing are inefficient for the inter-domain since they require several round trips to accomplish the path computation.

In this paper, we investigate the MCR problem in the inter-domain context. In order to reduce the end-to-end computation time and to respect the confidentiality and scaling constraints imposed by the inter-domain, we propose a *Distributed Inter-domain Multi-constrained Routing* (DIMR) to provide QoS in inter-domain networks.

Efficiency of our algorithm is provided with the use of our (1) distributed aggregation algorithm and (2) hybrid scale based discretization [5]. The distributed aggregation algorithm, which determines the upper and lower bounds to the optimal solution, decreases the amount of information that is exchanged between the PCEs for quick computations and confidentiality preservation. The hybrid scale based discretization (that uses the upper and lower bounds) improves the time complexity by reducing the number of samples without decreasing the quality of paths.

The rest of this paper is organized as follows. In section II,

we introduce problem MCR and give some notations adopted in the paper. Some works devoted to the study of MCR problem are described in Section III. In section IV, we present the hybrid scale-based MCR algorithm that allows the computation of paths ϵ -close to the optimums in mono-domain networks. In section IV-C, we propose and describe our distributed inter-domain MCR algorithm (DIMR) which allows the computation of end-to-end paths in inter-domain networks. Section V is devoted to the performance study of our proposition. Conclusion remarks are given in section VI.

II. NOTATION

We model an inter-domain network by an edge weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \vec{w})$, where \mathcal{V} is the set of *vertices*, \mathcal{E} is the set of *edges* and $\vec{w} = (w_1, \dots, w_k)$ is an *edge weight vector* of K components. The natural constant K corresponds to the number of QoS parameters. The k^{th} $1 \leq k \leq K$ weight of edge e is a positive real that is denoted by $w_k(e)$. To model the various domains which form the inter-domain network, we consider a mathematical partition \mathcal{D} of the set of nodes \mathcal{V} , that is a division of \mathcal{V} into a finite number p of disjoint and non-empty sets D_1, D_2, \dots, D_p . For each domain D_i , we associate two functions *in* and *out* returning respectively the intra-domain and extra-domain edges:

$$\begin{aligned} in &: \mathcal{D} \rightarrow \mathcal{E} \\ D_i &\mapsto \{u \rightarrow v : u, v \in D_i\} \\ out &: \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{E} \\ (D_i, D_j) &\mapsto \{u \rightarrow v : u \in D_i \wedge v \in D_j\} \end{aligned}$$

For simplicity, we set $in(D_i) = \mathcal{E}_i$ and $out(D_i, D_j) = \mathcal{E}_{i \rightarrow j}$.

Given a loop-free sequence of domains $\mathcal{S} = (D_1, \dots, D_p)$, we define a graph G_S as follows:

$$\begin{aligned} G_S &= \bigcup_{0 < i \leq p} G_i = \left(\bigcup_{0 < i \leq p} D_i, \bigcup_{0 < i < p} (\mathcal{E}_i \cup \mathcal{E}_{i \rightarrow i+1}) \cup \mathcal{E}_p, \vec{w} \right), \\ \left\{ \begin{array}{l} \forall 1 < i \leq p : \quad bin(D_i) = \{v \in D_i : \exists u \rightarrow v \in \mathcal{E}_{i-1 \rightarrow i} \\ \quad \quad \quad bin(D_1) = \Phi \end{array} \right. \\ \left\{ \begin{array}{l} \forall 0 < i < p : \quad G_i = (D_i \cup bin(D_{i+1}), \mathcal{E}_i \cup \mathcal{E}_{i \rightarrow i+1}, \vec{w}) \\ \quad \quad \quad G_p = (D_p, \mathcal{E}_p, \vec{w}) \end{array} \right. \end{aligned}$$

We note that G_S is a reduction of the inter-domain network graph \mathcal{G} that contains the sequence of domains \mathcal{S} and the corresponding edges. This graph G_S could be divided in p sub-graphs G_i ($0 < i \leq p$). Each subgraph G_i , referred here as *visibility* of domain i , is obtained by adding to the network topology of domain i the border ingress nodes of domain $i+1$ (i.e. $bin(D_{i+1})$) and the peering edges connecting domain i to $i+1$ (i.e. $\mathcal{E}_{i \rightarrow i+1}$). For any path π in \mathcal{G} , its k^{th} weight is determined as the sum of the k^{th} weights over the edges of π , i.e. $w_k(e) = \sum_{e \in \pi} w_k(e)$. In addition, all the paths in \mathcal{G} are assumed of length less than $H_{(H \leq n-1)}$ edges.

A. Problem $MCR_\epsilon(G_S, s, t, K, H, B_{inf}, B_{sup})$

Let $\epsilon \in]0, 1[$ be a real constant, $G_S = \bigcup_{0 < i \leq p} G_i = (\mathcal{V}, \mathcal{E}, \vec{w})$ be a graph, K be a number of QoS parameters, H be a maximum path length, B_{inf} a lower bound for solutions

and B_{sup} an upper bound for solutions. Solve the *tightened version* of problem MCR_ϵ which consists in determining a path π^ϵ connecting in G_S the source node s to the target node t and verifying the following formula:

$$\exists \pi^\epsilon \forall \pi : \left\{ \begin{array}{l} w_1(\pi) \leq d \\ B_{inf} \leq \max_{1 < k \leq K} w_k(\pi) \\ \leq B_{sup} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} w_1(\pi^\epsilon) \leq w_1(\pi) \\ \max_{1 < k \leq K} w_k(\pi^\epsilon) \\ \leq (1 + \epsilon) \cdot \max_{1 < k \leq K} w_k(\pi) \end{array} \right. \quad (1)$$

Formula (1) implies that for any path π verifying a given delay constraint $w_1(\pi) \leq d$, there is at least one approximate path π^ϵ (1) whose delay is lower or equal to that of π and (2) whose maximum weight on the $K - 1$ last metrics is at most ϵ -higher than the corresponding maximum weight of π . Path π^ϵ is a solution to problem MCR_ϵ .

Problem MCR_ϵ can be solved by discretizing the real solution space $[0, B_{sup} \times (1 + \epsilon)]^{K-1}$. In this paper, we will first present our hybrid scale-based solution for MCR_ϵ in the context of intra-domain networks. Then, we propose a weight aggregation-based distributed algorithm that extends the hybrid scale-based algorithm to solve MCR_ϵ in inter-domain networks.

III. RELATED WORK

Various works are proposed in literature to accommodate applications with the QoS but rare are the works [6] [7] which could be applied in the context of inter-domain networks. This can be explained essentially by the difficulty to deal with the domain topology confidentiality and the scaling constraints imposed by the inter-domain.

Two main architectures can be adopted to reduce the amount of information exchanged between domains while enabling the end-to-end path computation: hierarchical architecture and PCE-based architecture.

With the hierarchical architecture, the confidentiality and scaling constraints are tackled by aggregating the advertised information as well as we ascend in the hierarchy level [6]. This type of architecture does not suit well with the multi-constrained routing since it is inefficient to aggregate traffic flows with different QoS requirements. With the PCE-based architecture [8], a PCE is associated with one or more domains. This PCE has complete knowledge about its domain(s) and it can communicate and collaborate with other PCEs to enable distributed path computations. This second type of architecture is more flexible than the first one since the PCEs can choose the sub-paths to be advertised and the PCEs with which they communicate. With the definition of the backward-recursive PCE-Based Computation (BRPC) procedure described in [8], the determination of end-to-end paths by combining intra-domain paths becomes more easy and it is done efficiently. For instance, in [9] the BRPC procedure is combined with the SAMCRA algorithm [10] to compute optimal end-to-end paths fulfilling the QoS criteria. A similar approach reducing the number of paths which are explored is also described in [11].

In addition of the rare works devoted to the inter-domain QoS routing, various routing algorithms are proposed in literature for intra-domain networks. Some of them could be

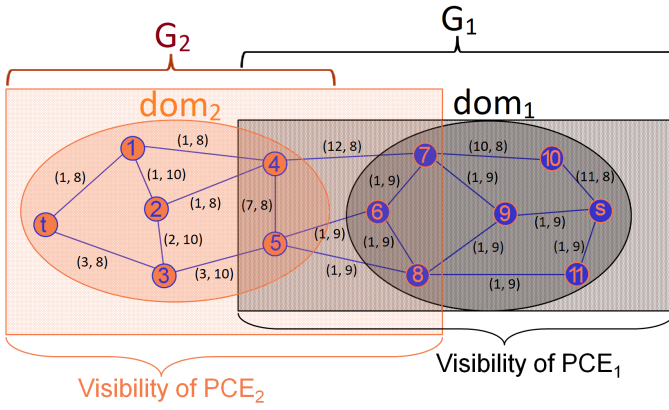


Fig. 1: PCEs and domains' visibilities

easily adapted for the inter-domain [6] [10] [12] whereas other ones cannot be applied in the inter-domain context since they overload the network or they require several round-trips to accomplish the path computations [3] [4].

Intra-domain approaches solving the MCR problem can be grouped in three main classes: exact algorithms, heuristics and FPTASs.

With exact algorithms [10] [13], the optimal solutions can be found. To reduce the running time, exact algorithms often transform the K additive metrics to one *non linear* metric. For instance, SAMCRA algorithm [10] transforms the $K-1$ last metrics into one non linear metric $(\max_{1 < k \leq K} w_k(\pi)/W)$ which is optimized with respect of the constraint $w_1(\pi) \leq d$.

Due to the NP-hardness of problem MCR, researchers explored several approximation approaches, referred here as heuristics, to solve MCR. In [2], various heuristics exploring different ideas are described. For instance, TAMCRA [12] limits the number of explored paths on each node.

The third class of approaches (FPTAS [4] [5] [6]) are heuristic variants which ensure the determination of ϵ -approximate solutions in function of the problem size and $1/\epsilon$. FPTASs follow two steps: discretization and computation. In the first step, representative samples are selected in the solution space. In the second step, an exact polynomial time algorithm able to solve MCR with integer weights is applied, after approximating all the real weights by the closest samples that were previously transformed into integers.

IV. WEIGHT AGGREGATION-BASED DISTRIBUTED INTER-DOMAIN MCR (DIMR)

To determine inter-domain paths, the domains should cooperate by exchanging some information. Before describing our solution for multi-constrained path computation in the inter-domain, we first present briefly the PCE architecture we adopted in this paper. Then, we describe in details our propositions which extend the hybrid scale-based approximation algorithm [5] to allow the computation of inter-domain paths. After that, we propose an effective weight aggregation-based algorithm to determine lower and upper bounds to the

optimums. These bounds are used by our DIMR to define and reduces the number of samples in the hybrid discretization scale.

A. PCE-based architecture

The PCE architecture is proposed in [8] and [14]. With such architecture, a PCE is associated with a set of domains that are generally managed by the same infrastructure provider. The PCE, that is a centralized computation element, has complete knowledge about the costs and network topology of the domains it manages. Thus, it is able to choose and efficiently determine the paths within its domains. For the inter-domain, end-to-end path computation requires cooperation between several PCEs with the exchange of various information that is generally related to path costs (i.e., path weights).

Consider the inter-domain network depicted in Figure 1. This network is composed of two domains: $D_1 = dom_1$ and $D_2 = dom_2$ in which the path computation tasks are respectively delegated to the path computation elements PCE_1 and PCE_2 . According to [8], each PCE has a limited *visibility* about the inter-domain network. In fact, the visibility of a given PCE PCE_i associated with a domain i only includes the network topology of domain i and the peering links interconnecting domain i with its adjacent domains j . For instance, the network visibility of PCE_2 is limited to the nodes and links located in the left rectangle whereas the network visibility of PCE_1 includes only the nodes and links located in the right rectangle. Note that PCE_i 's visibility includes domain visibility G_i (see Section II and Figure 1).

B. Hybrid scale-based FPTAS for intra-domain networks

Before describing our propositions which are based on the intra-domain hybrid scale-based FPTAS (proposed in [5]), we recall the principles and operation of this last algorithm.

To determine multi-constrained paths in intra-domain networks, the hybrid scale-based FPTAS selects a set of samples in the interval $(0, B_{sup} \times (1 + \epsilon))$ according to linear and logarithmic scales. By approximating each path weight by the nearest sample that is higher than that weight in the hybrid scale (function $h_near()$), it is possible to fully explore the solution space and thus determine guaranteed approximate solutions. To store the paths, the hybrid scale-based FPTAS associates to each vertex $v \in G$ two $K-1$ -dimensional tables d and $succ$: $d[v; i_2, \dots, i_K]$ stores the least delay among $v-t$ paths for the path weight tuple $(h_sample(i_2), \dots, h_sample(i_K))$ and $succ[v; i_2, \dots, i_K]$ holds the next hop of v for the path weight tuple $(h_sample(i_2), \dots, h_sample(i_K))$. The bijective function $h_sample()$ associates the i^{th} sample with its weight (or its cost) value in the hybrid scale.

After initializing all the delay and successor tables to respectively *infinity* (except for the target node where the delay tables entries are filled with 0) and *null* (lines 1 to 6 in Algorithm 1), the hybrid scale-based FPTAS executes lines 12 to 20 of Algorithm 1 to fill the delay and successor tables. Whereas Disjkstra's algorithm keeps the least delay and successor of each node, the hybrid scale-based FPTAS holds

Algorithm 1 DIMR-MCR $_{\epsilon}(G_S = \cup G_i, s, t, K, H, W, B_{inf}, B_{sup}, num_dom)$ $\{S = (D_1, \dots, D_{last_dom})\}$

Parameter real: $d_constraint$

Parameter \mathcal{V} : s, t

Parameter 1 .. last_dom: num_dom

```

1: for  $(i_2, \dots, i_K) \in \{0, 1, \dots, nb\_samples - 1\}$  do
2:    $d[u; i_2, \dots, i_K] \leftarrow \infty, succ[u; i_2, \dots, i_K] \leftarrow null, \forall u \in D_{num\_dom}$ 
3:   if  $t \in D_{num\_dom}$  then
4:      $d[t; i_2, \dots, i_K] \leftarrow 0$ 
5:   end if
6: end for
7: if  $num\_dom \neq last\_dom$  then
8:   for  $v \in bin(D_{num\_dom+1})$  do
9:      $d[v] \leftarrow receive\_d\_table(v, num\_dom + 1)$ 
10:  end for
11: end if
12: for all  $(i_2, \dots, i_K) \in \{0, 1, \dots, nb\_samples - 1\}$  in increasing
lexicographic order do
13:   for each  $(u, v) \in \mathcal{E}_{num\_dom} \cup \mathcal{E}_{num\_dom \rightarrow num\_dom+1}$  so that:
 $\forall 1 < k \leq K: h\_sample(i_k) - w_k(u, v) > 0$  do
14:      $\forall 1 < k \leq K: j_k \leftarrow h\_sample^{-1}(h\_near(h\_sample(i_k) -$ 
 $w_k(u, v)))$ 
15:     if  $d[u; i_2, \dots, i_K] > d[v; j_2, \dots, j_K] + w_k(u, v)$  then
16:        $d[u; i_2, \dots, i_K] \leftarrow d[v; j_2, \dots, j_K] + w_k(u, v)$ 
17:        $succ[u; i_2, \dots, i_K] \leftarrow (u, v)$ 
18:     end if
19:   end for
20: end for
21: if  $num\_dom \neq 1$  then
22:   for  $v \in bin(D_{num\_dom})$  do
23:     send_d_table  $(v, num\_dom - 1)$ 
24:   end for
25: else
26:   % Recall that  $s \in D_1$ 
27:   if  $d\_constraint < d[s; nb\_samples - 1, \dots, nb\_samples - 1]$  then
28:     send_notification_to_all_the_domains (no_solution)
29:     return  $\phi$ 
30:   end if
31:    $u \leftarrow s$ 
32:   for  $i \leftarrow 0$  to  $nb\_samples - 1$  do
33:     if  $d[u; i, \dots, i] \leq d\_constraint$  then
34:        $(a_1, \dots, a_k) = (i, \dots, i)$ 
35:        $\pi_{\epsilon}^1 \leftarrow path(u, a_1, \dots, a_k)$ 
36:     end if
37:   end for
38:   if  $u \neq t$  then
39:     send updated tuple  $(u; a_1, \dots, a_k)$  to  $num\_dom - 1$ 
40:   end if
41:   return  $\pi_{\epsilon}^1$ 
42: end if
43: receive tuple  $(u; a_1, \dots, a_k)$  from  $num\_dom + 1$ 
44:  $\pi_{\epsilon}^{num\_dom} \leftarrow path(u, a_1, \dots, a_k)$ 
45: if  $u \neq t$  then
46:   send updated tuple  $(u; a_1, \dots, a_k)$  to  $num\_dom - 1$ 
47: end if
48: return  $\pi_{\epsilon}^{num\_dom}$ 

```

one least delay and one successor for each tuple of the $K-1$ last metrics. In this way, one approximate path is determined for each optimal path in the interval (B_{inf}, B_{sup}) .

Like in Disjkstra's algorithm, the exploration of tables $succ$ permits to determine the optimal path. After filling tables $succ$, the execution of lines from 31 to 37 in Algorithm 1 allows the selection of an approximate path for the selected constraint delay ($d_constraint$). For that, the function $path$ (see Algorithm 2) is called to determine the edges of that path (line 35). Note that the function $path$ also uses an approximation function $l_near()$ which determines for a real

Algorithm 2 Function $path$

Parameter 0.nb_samples: i_2, \dots, i_K

Parameter \mathcal{V} : s

```

1:  $\pi^{\epsilon} \leftarrow \phi$ 
2: while  $succ[s; i_2, \dots, i_K] \neq null$  do
3:   add_edge_to_path( $succ[s; i_2, \dots, i_K], \pi^{\epsilon}$ )
4:    $u \leftarrow target\_node\_of\_edge(succ[s; i_2, \dots, i_K])$ 
5:    $\forall 1 < k \leq K:$ 
 $i_k \leftarrow h\_sample^{-1}(l\_near(h\_sample(i_k) - w_k(succ[s; i_2, \dots, i_K])))$ 
6: end while
7: return  $\pi^{\epsilon}$ 

```

weight the nearest sample that is lower or equal than itself.

C. Our distributed MCR for the inter-domain

Here, we propose two extensions to the intra-domain version of the hybrid scale-based FPTAS for inter-domain paths computation that respects the scalability and confidentiality constraints imposed by the inter-domain.

Firstly, to combine the intra-domain path computations, PCEs should exchange some information related to the path weights. Such information is available in the tables d and can be transmitted to the next PCE with the use of the protocol in [14]. Secondly, for optimization and confidentiality considerations, we reduce the amount of information transmitted in the network by sending only some path weights (the path structures are not transmitted) and by limiting the information exchanges to the PCEs associated with adjacent domains.

In Algorithms 1 and 2, the operation of our DIMR algorithm is illustrated. For simplicity and ease of understand, let's explain the principles and operation of DIMR through an example.

Consider the inter-domain network depicted in Figure 1. At the reception of a path computation request, a sequence of domains ($S = [dom_1, dom_2]$) allowing to connect the source and target nodes is determined. Then, all the PCEs managing these domains runs Algorithms 1. PCE_2 , that manages the domain dom_2 which contains the target node, performs its path computation in the network topology limited to the visibility of domain dom_2 (i.e. G_2). For each ingress node n_{in} in domain dom_2 ($n_{in} \in bin(D_2) = \{4, 5\}$), a set of paths interconnecting the ingress nodes and the target node t is determined and stored in the tables $succ[n_{in}; -]$ while the corresponding delay weights are kept in the tables $d[n_{in}; -]$ (see lines 12 up to 20 in Algorithm 1).

In order to permit end-to-end path computations, PCE_2 transmits (see lines 22 to 24 in Algorithm 1) to its precedent PCE (PCE_1) the delay tables of its ingress nodes.

At the reception of delay tables from PCE_2 (line 9 in Algorithm 1), PCE_1 continues the execution of Algorithm 1. After reinitializing the delay tables of the external nodes 4 and 5 with the received tables, PCE_1 deduces the $delay$ and $succ$ tables for all the nodes in G_1 by running the instructions from line 12 to line 20.

The end-to-end path that is ϵ -close to the optimum is determined by combining the sub-paths computed by the various PCEs participating to the computation. In our example

in Figure 1, PCE_1 runs the instructions between lines 27 and 37 to determine the sub-path allowing to reach the next domain dom_2 . Then, it sends to PCE_2 the source node and weights of the next sub-path allowing to reach the target. At the reception of such information (line 43 in Algorithm 1), PCE_2 runs the instruction in line 44 of Algorithm 1 to determine the sub-path allowing to reach the target node t in domain dom_1 . The function *path* described previously is called.

For confidentiality and policy considerations, a PCE can decrease the number of path weights it transmits to the next PCE by increasing some values in the delay tables.

Lemma 4.1: Our DIMR allows solving the problem $MCR_\epsilon(G_S, s, t, K, H, B_{inf}, B_{sup})$ in a worst-case time complexity of $\mathcal{O}(|\mathcal{E}| \cdot (\frac{H}{\epsilon})^{K-1} \cdot (\log(\frac{B_{sup}}{B_{inf}}))^{K-1})$.

Proof The correctness of the first assertion in 4.1 is trivial and could be proved by showing that the delay tables obtained upon the running of the intra-domain version of the hybrid scale-based FPTAS on the whole inter-domain network G_S are identical to those obtained with Algorithm 1.

The complexity C of Algorithm 1 is obtained by summing the complexities of path computation in each domain.

$$C = \mathcal{O}(\sum_{i=1}^{last_dom} |\mathcal{E}_i \cup \mathcal{E}_{i \rightarrow i+1}| \cdot (\frac{H}{\epsilon})^{K-1} \cdot (\log(\frac{B_{sup}}{B_{inf}}))^{K-1})$$

$$= \mathcal{O}(|\mathcal{E}| \cdot (\frac{H}{\epsilon})^{K-1} \cdot (\log(\frac{B_{sup}}{B_{inf}}))^{K-1}).$$

D. Link weights aggregation for effective approximation of lower and upper bounds to the optimal path weights

In order to minimize the number of samples in the hybrid scale while ensuring ϵ -optimality of paths, the upper and lower bounds should be determined so that the ratio B_{sup}/B_{inf} is minimal.

An excellent ratio between the upper and lower bounds ($B_{sup}/B_{inf} < H$) is given by the congestion link weight which corresponds to the lower bound. For the multi-constrained routing problem, the weight of the congestion link corresponds to the highest link weight such that if we delete all the links whose weights are greater or equal than that weight, no path can verify the delay constraint.

Hereafter, we give a PCE-based algorithm allowing the determination of the congestion link weights. In the first step, we replace the weights of the $K-1$ last metrics by their maximums. For each different value of these maximums, the PCEs compute the shortest delay paths allowing to interconnect ingress nodes with the target node t . In Figure 1 where the link labels correspond to the links weights (the first weight corresponds to the delay whereas the second one corresponds to a metric to minimize), PCE_2 first determines in G_2 the set $(\{8, 10\})$ of maximum weights on links. Then, it computes for each maximum \max the shortest delays from ingress nodes 4 and 5 to the target node t , after pruning from the network topology all the links whose maximums are strictly greater than \max . The results which are depicted in Table Ia are sent to PCE_1 which does the same operations on G_1 . In other words, PCE_1 adds the maximums it receives from PCE_2 to its set of maximums $(\{8, 9, 10\})$ and deduces the shortest delays from the source node s to the target node t for each

max weight	delay 4-t	delay 5-t
8	2	9
10	2	6

(a) Shortest delays to target t

max weight	delay $s-t$
8	35
9	12
10	9

(b) Congestion link weights

TABLE I: Congestion link weights for different delays

maximum value. The results which are depicted in Table Ib allows to deduce the congestion link weight (first column in Table Ib) for any delay constraints. For instance, with a delay constraint in (12, 35), the congestion weight is equal to 9.

Though this algorithm significantly reduces the search solution space with a ratio B_{sup}/B_{inf} that is lower than H , it has various drawbacks. Indeed, this algorithm does not scale because it performs a large number of computations and it also sends a large volume of information to the other domains. In addition, it suffers from confidentiality issues since it transmits real delays and link weights outside the domains.

To cope with the previous issues, we propose here to aggregate the maximum weights according to an exponential function $f_C(n) = C^n$. For instance, with an aggregation base C equal to 2, the maximums of the $K-1$ last metrics are approximated by the closest powers of 2 which are lower or equal to these maximums. In the example of Figure 1, all the maximums (i.e. $\{8, 9, 10\}$) will be approximated by 8 which is the closest power of 2 that is lower than these maximums. In this way, PCE_2 sends to PCE_1 two delay values (2 for path 4-t and 6 for path 5-t) for the unique maximum weight 8. When PCE_1 receives the delay values from PCE_2 , it deduces the delay (equal to 9) of the unique congestion link weight 8. As a result, the lower bound will be approximated by 8 whereas the upper bound will be approximated by $C \times H = 2 \times H$ for any delay constraints greater or equal to 9.

Note that our aggregation algorithm does not affect the worst case time complexity of Algorithm 1.

V. NUMERICAL RESULTS

To measure the performances of our distributed inter-domain MCR, we compared them to the linear scale-based centralized FPTAS (FSA) and logarithmic scale-based centralized FPTAS (LSA). In our simulation, we used two versions of our algorithm: (1) simple DIMR (SDIMR) that utilizes the congestion link weight to approximate the lower and upper bounds and (2) weight aggregation based-DIMR (ADIMR) that approximates the lower and upper bounds according to our aggregation algorithm described in Section IV-D. The aggregation base is equal to 2. Due to the lack of space, we show here only the important results.

The number of metrics K is fixed to 3. Two generation methods are used to set the link weights: *uniform generation* and *border generation*. With the first method, the K weights of links are randomly selected in $(10^6, 10^6 + 999)$. With the second method, about 5% of the link weights are chosen in $(1, 10)$ and the rest are selected in $(10^6, 10^6 + 999)$.

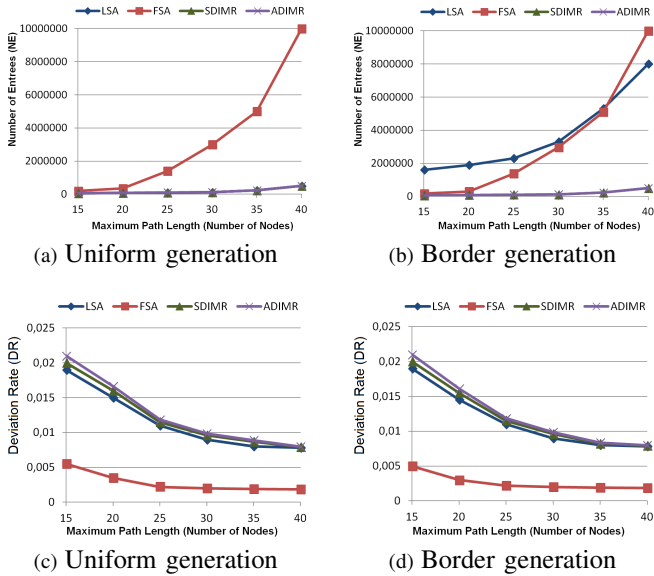


Fig. 2: Number of table entrees (NE)

For our comparison study, we chose two comparison metrics: *number of entrees (NE)* in the tables d and *deviation rate (DR)*. The first metric computes the mean number of entrees in each node table d . It is correlated with the running time and it determines the amount of information exchanged between the PCEs. The second metric measures the quality of the computed solutions. It is determined as follows:

$$DR = (\max_{i=2}^K w_i(\pi^\epsilon) - \max_{i=2}^K w_i(\pi^*)) / \max_{i=2}^K w_i(\pi^*).$$

Where π^* is the optimal path that verifies the delay constraint and minimizes the maximum weight of the $K-1$ last metrics.

All the paths have their source and target nodes in two different domains. The test results are depicted in Figure 2.

As expected, the weight aggregation, used by ADIMR, has imperceptible side effects on the performances of our distributed inter-domain MCR algorithm. As a result, for confidentiality considerations ADIMR should be preferred to SDIMR. In the following, we only focus on the comparison between SDIMR, FSA and LSA since the performances of ADIMR are similar to those of SDIMR.

Concerning the evolution of the mean number of table entrees (NE) in function of the maximum path length, the results are shown in Figures 2a and 2b for uniform and border generations respectively. Whereas SDIMR and LSA have close and smaller numbers of entrees compared to that of FSA for uniform generation, algorithm SDIMR clearly outperforms FSA and LSA with border generation. This is explained by the discretization scale of SDIMR that is better than linear and logarithmic discretization scales in one hand and to the distribution nature of SDIMR which allows the decrease of NE for the domains which are close to the target.

Figures 2 (c) and 2 (d) show the deviation rates associated with the compared methods. As shown, the four compared algorithms have similar performances which are close to the optimum values, although the theoretical maximum error rate

parameter is high ($\epsilon = 0.5$). Obviously, FSA has a deviation rate that is slightly lower than those obtained with SDIMR and LSA. Indeed, decreasing the number of samples in the discretization scale (with the use of SDIMR) leads to the increase of the deviation rate that is fortunately very small.

VI. CONCLUSION

In this paper, we proposed a novel weight aggregation-based distributed routing for end-to-end QoS path computation in inter-domain networks. With the exchange of limited weight vectors which are easy to compress, our algorithm is capable to solve the tightened version of multi-constrained routing problem in inter-domain networks. It is scalable, fast and fully polynomial with a worst case time complexity of $\mathcal{O}(|\mathcal{E}| \cdot (\frac{H}{\epsilon})^{K-1} \cdot (\log(\frac{B_{sup}}{B_{inf}}))^{K-1})$.

Besides, our weight aggregation-based distributed inter-domain multi-constrained routing verifies the confidentiality constraints imposed by the inter-domain network since no link weight and no path are exchanged between the domains to accomplish the computations. It also enables adopting domains' policies by restricting the information exchanged between the domains at the cost of small path weight increase.

REFERENCES

- [1] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.
- [2] P. Van Mieghem, F. A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-bruin, J. Solé-pareta, and S. Sánchez-López, "Quality of Service Routing," in *Quality of Future Internet Services*, ser. Lecture Notes in Computer Science. Springer, 2003, vol. 2856/2003, pp. 80–117.
- [3] D. H. Lorenz and D. Raz, "A Simple Efficient Approximation Scheme for the Restricted Shortest Path Problem," *Operations Research Letters*, vol. 28, pp. 213–219, 1999.
- [4] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman, "Polynomial time approximation algorithms for multi-constrained QoS routing," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 656–669, 2008.
- [5] M. Y. Saidi and G. Texier, "Hybrid scale-based approximation algorithm for qos routing," in *International Conference on Telecommunications (ConTEL)*, June 2011, pp. 367–374.
- [6] A. Orda and A. Sprintson, "Precomputation schemes for QoS routing," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 578–591, 2003.
- [7] T. Saad, H. T. Mouftah, and A. Nouroozifar, "Constraint-based routing across multi-domain optical WDM networks," in *Canadian Electrical and Computer Engineering*, vol. 4, May 2004, pp. 2065–2068.
- [8] J. P. Vasseur, R. Zhang, N. Bitar, and J. L. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain MPLS-TE Networks," RFC 5441, IETF, Apr. 2009.
- [9] G. Bertrand, S. Lahoud, G. Texier, and M. Molnár, "A Distributed Exact Solution to Compute Inter-Domain Multi-Constrained Paths," vol. 5733, 2009, eUNICE – The Internet of the Future.
- [10] P. Van Mieghem, H. De Neve, and F. Kuipers, "Hop-by-hop quality of service routing," *Comput. Netw.*, vol. 37, no. 3-4, pp. 407–423, 2001.
- [11] G. Bertrand, S. Lahoud, G. Texier, and M. Molnár, "Computation of Multi-Constrained Paths in Multi-Domain MPLS-TE Networks," in *Next Generation Internet Networks (NGI)*, July 2009.
- [12] H. D. Neve and P. V. Mieghem, "Tamcra: a tunable accuracy multiple constraints routing algorithm," *Computer Communications*, vol. 23, no. 7, pp. 667–679, 2000.
- [13] G. Feng, K. Makki, and C. Douligeris, "Heuristic and Exact Algorithms for QoS Routing with Multiple Constraints," *IEICE Trans. Comm.*, vol. E85-B, no. 12, pp. 2838–2850, 2002.
- [14] Vasseur, Jean Pierre and Le Roux, Jean Louis, "Path Computation Element (PCE) Communication Protocol (PCEP)," RFC 5440, IETF, 2009.