A distributed exact solution to compute inter-domain multi-constrained paths

Gilles Bertrand, Samer Lahoud, Géraldine Texier, Miklos Molnar

A Distributed Exact Solution to Compute Inter-domain Multi-constrained Paths

Gilles Bertrand[1], Samer Lahoud[2], Géraldine Texier[1], and Miklós Molnár[3]

[1] Institut TELECOM, TELECOM Bretagne, RSM
{gilles.bertrand,geraldine.texier}@telecom-bretagne.eu
[2] IRISA - Université de Rennes 1
samer.lahoud@irisa.fr
[3] IRISA - INSA de Rennes
molnar@irisa.fr

Abstract. The fundamental Quality of Service (QoS) routing problem, which consists in determining paths subject to multiple QoS constraints, has been extensively investigated in the intra-domain context. However, few solutions exist for the inter-domain case, despite the importance of this problem to enable the delivery of services with QoS across domain boundaries. We propose a method that distributes the operations to compute inter-domain constrained paths. This method relies on a per-domain formulation that is compatible with the path computation element framework. It enables us to propose the first algorithm that guarantees to find optimal paths subject to an arbitrary number of constraints. These paths ensure the best QoS performance with respect to the constraints.

1 Introduction

The computation of constrained inter-domain paths is a highly investigated topic: it enables the operators to take control over the routing and facilitates the delivery of Quality of Service (QoS) across domain boundaries. Inter-domain applications typically impose constraints on several QoS metrics (for example, delay, jitter, packet losses). Nevertheless, computing the corresponding inter-domain paths subject to multiple QoS constraints has long been considered as impracticable because of the required communication overhead and because of confidentiality constraints. Thus, in the current Internet, inter-domain routing is based solely on basic connectivities and operator policies. However, recent initiatives propose connection-oriented solutions based on multiprotocol label switching to allow the computation of constrained inter-domain paths without breaking the confidentiality constraints of the domains [1]. In particular, the Path Computation Element (PCE) framework [2] enables the collaboration of multiple domains to setup inter-domain constrained paths.

The present paper investigates the problem of computing inter-domain Multi-constrained Paths (MCPs) in the PCE framework. As the MCP problem has many important applications, the literature describes several solutions for the *intra-domain* case [3, 4, 5, 6]. However, these solutions cannot be used to solve

the *inter-domain* MCP problem because they rely on centralized computations and break the confidentiality constraints of the domains. Alternatively, Saad and others [7] propose an approximate solution to the inter-domain MCP problem. Nevertheless, their study is limited to the problem with two constraints.

We analyze the required inter-domain information exchanges to solve the inter-domain MCP problem exactly. This study enables us to present a method to distribute the path computation. In particular, we describe a per-domain formulation that is compatible with the PCE framework. This formulation allows us to introduce the first solution that guarantees to find a path satisfying the considered QoS constraints if such a path exists in the network. Moreover, the computed paths are the furthest from the constraints, and thus, ensure the best resistance to changes in the QoS conditions.

The remainder of the paper is organized as follows. In Section 2, we present the requirements for inter-domain MCP computations and related work. The proposed path computation method is described in Section 3 and followed by a discussion on its performance in Section 4.

2   Background

2.1   Requirements for Inter-domain Path Computations

The Internet relies on the inter-connection of networks administrated by various network operators and called *domains*, as illustrated in Fig. 1. The domains inter-connect through domain border routers, which exchange mainly connectivity information to preserve the scalability of the routing protocols.

The domains represent network operators, which are bound by competitive relationships. In particular, the operators preserve the confidentiality of their network state and topology. As a result, in every domain, only internal entities possess enough information about the network state of the domain to allow the computation of constrained paths inside this domain. Moreover, the domains (*e.g.,* BGP autonomous systems) have a wide autonomy. To summarize, there is no central coordination entity with global state and topology information on the traversed domains.

With distributed procedures, every domain controls its internal paths. We think that the computation of inter-domain paths must typically be distributed
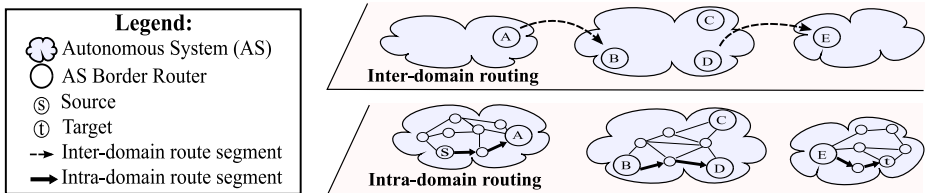
Fig. 1. Routing relies on a two-level hierarchy with a separation between intra-domain and inter-domain operations
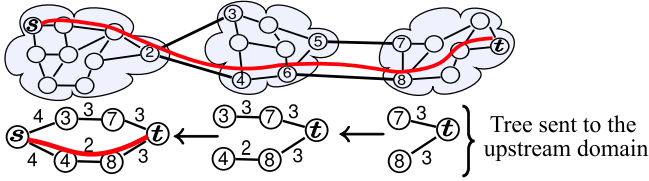
Fig. 2. Operations of the BRPC procedure to compute an inter-domain path from s to t and that traverses the lowest number of links

among the domains, to preserve both the autonomy of the domains and the scalability of the routing protocols. Therefore, we present a novel method to distribute the computation of inter-domain constrained paths.

2.2   The BRPC Procedure

Reference [1] describes a Backward Recursive PCE-based Computation (BRPC) procedure to compute constrained inter-domain traffic engineered label switched paths. BRPC uses multiple PCEs to compute these paths along a determined domain sequence and preserves the autonomy of the domains.

We illustrate the operations of BRPC in Fig. 2. Starting from the destination domain, each domain computes a shortest-path tree with respect to a specific objective function. The root of this tree is the destination node of the considered path computation request. Its leaves are the entry Border Nodes (BNs) of the domain. Then, the domain forwards this tree to the previous domain in the considered sequence of traversed domains. Note that the advertised tree does not necessarily include all the computed shortest paths: the domain selects the paths that it wants to advertise to the upstream domain. The upstream domain uses the information contained in the tree to compute a similar shortest-path tree from its entry BNs towards the destination (t) of the request. This process is repeated until the source node (s) is reached.

Note that BRPC requires that the domains exchange information about the computed paths. This information might divulge confidential details about the internal topology of the traversed domains. This problem is solved by the use of path keys [8], which enable the domains to advertise only the entry BNs and information about the path performance. The BRPC procedure specifies the protocol exchanges among the domains; however, it does not describe the algorithm that must be used in each traversed domain to compute the paths.

2.3   The MCP Problem

To define the MCP problem, we consider a network represented by a valued graph $G = (V, E, w)$. The set V of vertices represents the nodes of the considered network. The set E of edges corresponds to the network links. The function $w : E \rightarrow \mathbb{R}$, where $\mathbb{R}$ denotes the set of the real numbers, provides the value of

the considered link metrics for every link: to model multiple QoS metrics, each link l in E is associated with a vector $\boldsymbol{w}(l)$ of $K \in \mathbb{N}$ link weights. We denote the k-th weight of the link l as $w_k(l) \in \mathbb{R}^+$, with k in $[1..K]$, where $\mathbb{R}^+$ represents the set of the nonnegative real numbers. We define the weights of a path $\mathbf{p}$ as the sum $w_k(\mathbf{p}) = \sum_{l \in \mathbf{p}} w_k(l)$ of the link weights on this path. This means that we consider only additive metrics. Nevertheless, we can treat the constraints on bottleneck metrics (*e.g.,* bandwidth) by computing a path in a graph from which we have removed the links that break the constraints. In addition, positively-valued multiplicative constraints can be transformed into additive constraints by using a logarithm function.

We consider two non-empty paths $\mathbf{p}$ and $\mathbf{s}$. We say that $\mathbf{s}$ is a *suffix* of $\mathbf{p}$ if there is a path $\mathbf{q}$ such that $\mathbf{p}$ is the concatenation of $\mathbf{q}$ and $\mathbf{s}$. We denote the set of paths from a node s to a node t as $P_{s \to t}$. Similarly, we denote as $P_{D \to t}$ the set of paths whose source is in the subset D of V and whose destination is the node t in V. We consider a *path computation request*, which specifies a source s and a destination t in V, as well as K constraints W_k in $\mathbb{R}^{+*}$, with k in $[1..K]$ and $\mathbb{R}^{+*}$ representing the set of the positive real numbers. The constraints represent maximum bounds W_k on every weight $w_k(\mathbf{p})$ of the requested path: to be acceptable, a path $\mathbf{p} \in P_{s \to t}$ must satisfy $w_k(\mathbf{p}) \leq W_k$ for all k in $[1..K]$. We call *feasible path* any path that fulfills the constraints of the request; the MCP problem consists in determining a feasible path. For instance, the problem of finding a path whose end-to-end propagation delay is below fifty milliseconds and which traverses less than fifteen links is an MCP problem.

The literature describes several brute-force algorithms [3,5] as well as heuristics [4] and approximation algorithms [6] for the *intra-domain* MCP problem. These algorithms typically assume that the network uses link-state routing (every router maintains a complete view of the network) and require centralized path computation operations. For inter-domain path computation, these assumptions imply that every domain involved in the computation operations has complete information about the state of other domains, which is usually unacceptable for confidentiality and communication overhead reasons. In addition, centralized path computation operations are incompatible with the required autonomy of the domains. Consequently, we investigate distributed methods for multi-constrained inter-domain path computation. To the best of our knowledge, our work is the first to present an exact distributed solution to the inter-domain MCP problem.

2.4   The Inter-MCP Problem

We consider a mathematical partition $\mathcal{D}$ of the set of nodes V, that is a division of V into a finite number of non-overlapping and non-empty sets. We call every element of this partition a *domain*. We define the Inter-MCP problem as follows:

Problem 1 (Inter-MCP). Given a finite loop-free sequence $\mathbf{S} = (D_1, D_2 \ldots)$ of $|S| \in \mathbb{N}$ domains, a source node $s \in D_1$, a destination node $t \in D_{|S|}$, and a set of K end-to-end constraints W_k with $k \in [1..K]$, find a *feasible* path from s to t that traverses the sequence $\mathbf{S}$ of domains.

Our definition of the Inter-MCP problem relies on the assumption that the domain sequence crossed by the path is predetermined. This assumption is commonly adopted in relevant work of the domain, as in [1], to reduce the complexity of the problem.

Wang and Crowcroft [9] have shown that the MCP problem is $\mathcal{NP}$-complete. As the MCP problem is a special case of the Inter-MCP problem, the Inter-MCP problem is $\mathcal{NP}$-complete too [10]. Thus, some of its instances cannot be solved exactly in polynomial time (if $\mathcal{P} \neq \mathcal{NP}$) [10]. However, previous work on the MCP problem shows that most instances of the MCP problem can be solved exactly in polynomial time [4, 11].

The contribution of our paper is to propose a method to distribute the computation of inter-domain MCPs. We provide two mechanisms, the first is a method to distribute end-to-end computation operations among the traversed domains, and the other one is an algorithm to perform the per-domain calculations. In particular, we propose a solution, named ID-MCP (Inter-Domain MCP), which is the first to enable the distributed exact computation of inter-domain MCPs considering an arbitrary number of constraints. We use BRPC to propagate the computation results and to permit the selection of optimal end-to-end paths.

3 Proposition of a Distributed Exact Solution

3.1 Distribution of the Computations among the Domains

Inter-domain path computations must be distributed among the traversed domains so that the domains keep their autonomy. Therefore, our solution defines the per-domain computations of the inter-domain MCP problem for every traversed domain. The result of these per-domain computations is a set of paths that our algorithm will use to find a feasible end-to-end path.

We use the concept of *dominance*, which expresses the idea that a solution is "better" than another, to reduce the complexity of the computations inside each domain. A path $\mathbf{p}$ is *dominated* if there is a path $\mathbf{p}'$, with the same source and destination, such that $w_k(\mathbf{p}') \leq w_k(\mathbf{p})$ for all considered weights $w_k, k \in [1..K]$ and such that there is a k in $[1..k]$ for which $w_k(\mathbf{p}') < w_k(\mathbf{p})$. In this case, we say that $\mathbf{p}'$ dominates $\mathbf{p}$. For instance, consider two paths $\mathbf{p}_1$ and $\mathbf{p}_2$ between the same nodes and such that $\boldsymbol{w}(\mathbf{p}_1) = (4,3)^T$ and $\boldsymbol{w}(\mathbf{p}_2) = (3,3)^T$, then, $\mathbf{p}_2$ dominates $\mathbf{p}_1$. Concretely, a dominated path is never a good solution to an MCP problem: there is a better candidate solution between the same nodes. Thus, we are interested in computing only non-dominated paths. In particular, previous work on the intra-domain MCP problem [5] shows that, to compute MCPs exactly, intermediate nodes should memorize only feasible non-dominated intermediate paths. We can apply this result: to compute inter-domain MCPs, it is sufficient that every domain computes non-dominated feasible paths from its entry BNs to the destination of the request. This method allows us to transform the Inter-MCP problem into a specific MCP problem for every traversed domain.
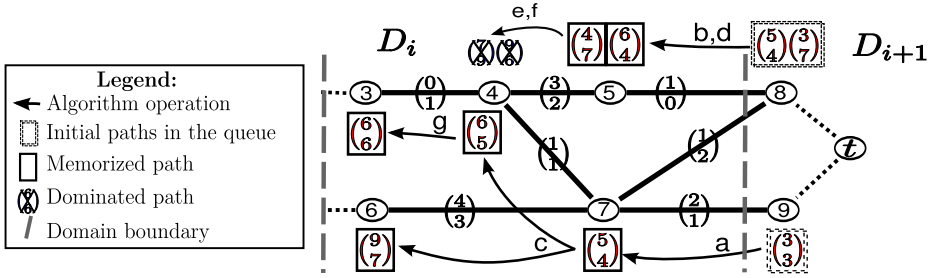
Fig. 3. Simplified operations of the extended RDA in a domain, considering the constraints $W_1 = 11$ and $W_2 = 14$

Problem 2. [Per-Domain Problem] Given an instance of the Inter-MCP problem, a domain D_i in the sequence $\mathbf{S} = (D_1, D_2 \ldots)$, and a set $P^* \subset P_{D_{i+1} \to t}$ of paths from the entry BNs of the next domain D_{i+1} to t, find all the non-dominated feasible paths from the entry BNs of D_i to t that have a suffix in P^*.

3.2   Solution to the Per-domain Problem

For the per-domain operations of ID-MCP, we propose a novel extended reverse Dijkstra's algorithm (RDA) [12] that memorizes all non-dominated feasible intermediate paths. Figure 3 depicts the operations of this algorithm: the purpose of the operations is to find the non-dominated feasible paths from the entry BNs 3 and 6 to the destination t of the request. To compute these paths, the algorithm uses a queue structure that contains the shortest paths from every intermediate node to t. This queue is initialized with paths from the entry-BNs of the neighboring downstream domain, nodes 8 and 9, to t. In the example, the queue starts with two different paths from 8 to t and a path from 9 to t. The calculation progresses from the right to the left of Fig. 3.

The algorithm runs a loop. During each iteration of the loop, the algorithm picks among the paths of the queue a path $\mathbf{p}$ whose weights are the furthest from the constraints. In the example, the first path selected starts from node 9 and its weights are $(3,3)^T$ because the alternative paths in the queue (two paths from node 8) are closer to the constraints. Then, the algorithm *relaxes* $\mathbf{p}$: it evaluates the weights of the paths from the neighboring nodes (node 7) of the source of $\mathbf{p}$ (node 9) that have $\mathbf{p}$ as suffix. The algorithm adds the discovered paths to the queue if they are feasible and not dominated by any path in the queue. In the example, the path from node 7 with weights $(5,4)^T$ is added to the queue. If a new path dominates one or more paths of the queue, then the dominated paths are discarded. The loop is repeated while the queue contains at least one element that has not been relaxed or discarded (operations b to g). Finally, the algorithm finds the feasible non-dominated paths from the entry BNs 3 and 6. For example, it discovers the path with weights $(6,6)^T$ from the entry BN 3 to t through the nodes 3-4-7-9. During its operations, the algorithm does not discard
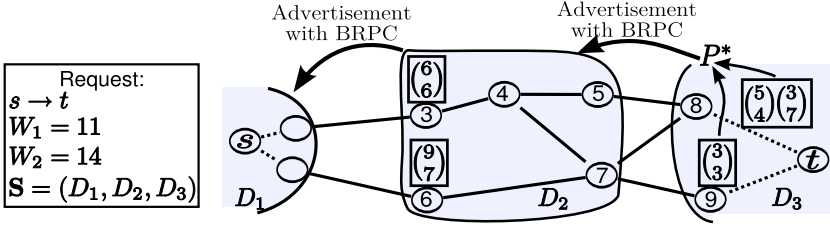
Fig. 4. Propagation of the per-domain results for an inter-domain MCP problem with two integer-valued metrics

any non-dominated feasible path. Thus, it guarantees to solve the Problem 2. The termination of the algorithm is guaranteed as it relaxes one path during each iteration of the loop and the number of paths in a domain is finite.

3.3   Propagation of the Per-domain Computation Results

Figure 4 illustrates the Inter-MCP problem on a simple example with two integer-valued link weights. We use BRPC to forward the non-dominated feasible paths computed by every traversed domain to a PCE of the previous domain in the considered sequence of crossed domains. An intermediate domain (D_2) receives a set P^* of paths from the downstream domain (D_3). P^* includes the feasible non-dominated paths from the entry-BNs (8 and 9) of the downstream domain to the destination t of the request. The intermediate domain extends its local vision of the network topology with the paths in P^* and uses this information to compute the non-dominated feasible paths from its own entry-BNs (3 and 6) to t using our algorithm. Then, it advertises these paths to its neighboring upstream domain (D_1), which is the source domain and is able to compute an end-to-end feasible path. Note that a path-key mechanism [8] can be used to preserve confidential information: it enables the BNs to advertise only path performance information and a path-key that identifies a specific path. The key is later translated into an explicit path during signaling operations.

Our algorithm requires the two following extensions of the PCE framework. First, with our method, more than one non-dominated feasible paths from the same entry BN can be advertised to the previous domain. This case was not considered for the original BRPC procedure, thus, our algorithm requires an extension of the virtual shortest-path tree structure defined in [1] to enable the forwarding of several paths with the same source. This extension of the BRPC procedure is conceptual: it does not require any modification of the PCE protocol (PCEP) [2], which already provides the objects required to carry such paths. Second, the path computation requests and replies should indicate that every traversed domain must use our algorithm. This information can be carried thanks to an objective function object [13] of PCEP [2], for example.

4   Discussion

4.1   Performance Metrics

By design, ID-MCP provides provable performance guarantees. In particular, it guarantees to find a path satisfying the request constraints if such a path exists. In addition, as ID-MCP computes all non-dominated feasible paths, it enables selecting the path that provides the largest performance margin compared to the request constraints. This feature is important to maximize the chances that a computed path can be successfully setup, as the state of the network can change between the computation of a path and its setup. Simulations provide additional information about the performance of our algorithm and the amount of signaling overhead that it introduces in the network.

We evaluate the following performance metrics. The number of feasible non-dominated paths returned by ID-MCP is denoted as NP. This number enables us to evaluate the available path diversity in the simulated scenario. Moreover, we consider the cost (C), defined as the lowest value of the function $c(\mathbf{p}) = \max_{k \in [1..K]} \left(\frac{w_k(\mathbf{p})}{W_k} \right)$ among the computed paths. Furthermore, we define a function c' as $\mu_{k \in [1..K]} \left(\frac{w_k(\mathbf{p})}{W_k} \right)$, where μ denotes the arithmetic mean operator. We call multi-dimensional cost (MC) the value of c' for the end-to-end path that has the lowest value of c' among the computed paths. MC helps to evaluate the quality of the returned paths considering all metrics, whereas C indicates their performance margin with respect to the most restrictive metric.

Previous work [4, 5] has shown that the complexity of MCP computations inside a domain depends on the maximum number α of paths memorized for a single node. Thus, we measure α to estimate the time complexity of ID-MCP. In addition, α helps us to determine the number of paths exchanged by the domains during inter-domain path computation operations, and thus, the signaling overhead.

4.2   Evaluation Scenario

We present results for topologies representing extreme cases for the algorithm, to assess its performance in a worst-case situation.[1] We consider lattice domain topologies (square grids) made up of 25 nodes, as represented in Fig. 5. We interconnect the domains to build a domain sequence along which we compute inter-domain MCPs. In FM (Full Mesh), every node of an intermediate domain is connected to every node in the next and in the previous domain of the sequence. In SL (Single Link), only the bottom-right node of every domain is connected to the top-left node of the next-domain. We generate the values of two random uniformly distributed link-weights for every simulation run. We consider the following path computation requests. The source is the top-left node of the first domain and the destination is the bottom-right node of the last domain. We compare the outputs of ID-MCP for two different set of constraints. First, we

---

[1] Simulations on real topologies provide similar results [14].
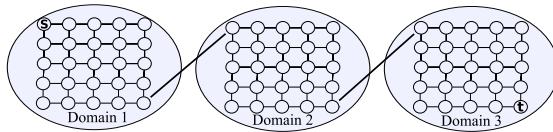
Fig. 5. The lattice topology with single-link domain inter-connections (SL)

define *strict constraints* so that, in average, there is a feasible path for less than 70% of the simulated requests. Second, we define *loose constraints* so that there is a feasible path for every simulated request.

4.3   Simulation Results

Table 1 presents the results of the simulations. For loose constraints, the performance of the optimal path computed by ID-MCP is much better than requested (C and MC are below 20%). This indicates that ID-MCP finds paths that will remain feasible even in case of variations in the network state. In addition, ID-MCP manages to find several feasible non-dominated paths (NP is greater than 3) even when the optimal performance of the network is close to the constraints (for strict constraints, C and MC are greater than 60%).

The complexity of the computations remains reasonable for the simulated scenario (α is lower than 10). However, the size of the simulated topology is relatively limited compared to real networks, which can include several hundreds of nodes. In addition, the MCP problem is $\mathcal{NP}$-complete. Thus, the complexity of the computations on larger topologies would be typically prohibitive. This scalability concern underlines the need for a trade-off between the complexity of the calculations and the performance of the computed paths. In particular, our algorithm provides the theoretical basis for future heuristics with a reduced complexity. We will study these heuristics in forthcoming work.

Table 1. Simulation results with loose and strict constraints

	C [%]		MC [%]		α		NP	
Constraints	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*
Loose	19.2	13.9	18.7	11.4	10	7	7	3
Strict	89.3	72	86.5	60.1	8	2	5	1

5   Conclusion

In this paper, we have studied the inter-domain MCP problem, whose applications are more and more important with the recent advances in inter-domain traffic engineering. The main contribution of the paper is the distribution of the MCP computation operations among the traversed domains. In particular, we

have analyzed the required information exchanges among the domains to solve the inter-domain problem. This study has enabled us to introduce ID-MCP, the first solution to the considered problem that guarantees to find a feasible path if such a path exists in the network. In particular, ID-MCP is based on distributed per-domain computations that are compatible with the PCE framework and that allow computing optimal end-to-end paths. These paths are the furthest from the constraints, thus, they ensure the best resistance to future variations of the QoS conditions. As the considered problem is $\mathcal{NP}$-complete, the complexity of the computations is prohibitive for large topologies. However, our solution provides the theoretical foundation for future heuristics with a reduced complexity.

References

1. Vasseur, J., Zhang, R., Bitar, N., Roux, J.L.: A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths. RFC 5441, IETF (2009)
2. Vasseur, J., Roux, J.L.: Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440 (2009)
3. Korkmaz, T., Krunz, M.: Multi-constrained optimal path selection. In: IEEE INFOCOM, vol. 2, pp. 834–843 (2001)
4. Yuan, X.: Heuristic algorithms for multiconstrained quality-of-service routing. IEEE/ACM Trans. Netw. 10(2), 244–256 (2002)
5. Van Mieghem, P., Kuipers, F.A.: Concepts of exact QoS routing algorithms. IEEE/ACM Trans. Netw. 12(5), 851–864 (2004)
6. Xue, G., Sen, A., Zhang, W., Tang, J., Thulasiraman, K.: Finding a path subject to many additive QoS constraints. IEEE/ACM Trans. Netw. 15, 201–211 (2007)
7. Saad, T., Mouftah, H., Nouroozifar, A.: Constraint-based routing across multi-domain optical WDM networks. In: Canadian Conference on Electrical and Computer Engineering, vol. 4, pp. 2065–2068 (2004)
8. Bradford, R., Vasseur, J.P., Farrel, A.: Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Key-Based Mechanism. draft-ietf-pce-path-key-06, work in progress, IETF (2009)
9. Wang, Z., Crowcroft, J.: Quality-of-Service Routing for Supporting Multimedia Applications. IEEE J. Sel. Areas Commun. 14, 1228–1234 (1996)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
11. Kuipers, F., Van Mieghem, P.: The impact of correlated link weights on QoS routing. In: IEEE INFOCOM, vol. 2, pp. 1425–1434 (2003)
12. Ahuja, R., Magnanti, T., Orlin, J.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., New Jersey (1993)
13. Le Roux, J.L., Vasseur, J.P., Lee, Y.: Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP). draft-ietf-pce-of, work in progress, IETF (2008)
14. Bertrand, G., Lahoud, S., Molnár, M., Texier, G.: Inter-Domain Path Computation with Multiple Constraints. Technical Report 1902, IRISA (2008), `http://hal.inria.fr`