Computer Networks 54 (2010) 2453–2467

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

AS-level source routing for multi-provider connection-oriented services

Stefano Secci [a,*], Jean-Louis Rougier [a], Achille Pattavina [b]

[a] Institut Télécom, Télécom ParisTech, LTCI CNRS, 23 Avenue d'Italie – CS 51327, 75214 Paris, Cedex 13, France
[b] Dip. Elettronica e Informazione, Politecnico di Milano, via Ponzio 34/5, 20133 Milano, Italy

ARTICLE INFO

Article history:
Received 14 September 2009
Received in revised form 24 February 2010
Accepted 1 April 2010
Available online 10 April 2010
Responsible Editor: T. Korkmaz

Keywords:
Inter-domain routing
Inter-AS MPLS
QoS routing
Multipoint routing
AS-level routing

ABSTRACT

In this paper, we study the inter-domain Autonomous System (AS)-level routing problem within an alliance of ASs. We first describe the framework of our work, based on the introduction of a service plane for automatic multi-domain service provisioning. We adopt an abstract representation of domain relationships by means of directional metrics which are applied to a triplet (ingress point, transit AS, egress point) where the ingress and egress points can be ASs or routers. Then, we focus on the point-to-point and multipoint AS-level routing problems that arise in such an architecture. We propose an original approach that reaches near optimal solutions with tractable computation times. A further contribution of this paper is that a heavy step in the proposed heuristic can be precomputed, independently of the service demands. Moreover, we describe how in this context AS-level path diversity can be considered, and present the related extension of our heuristic. By extensive tests on AS graphs derived from the Internet, we show that our heuristic is often equal or a few percent close to the optimal, and that, in the case of precomputation, its time consumption can be much lower than with other well-known algorithms.

1. Introduction

A dynamic routing architecture suitable for inter-provider, connection-oriented, service provisioning has not been implemented yet, mainly because of privacy, billing and monitoring issues. However some important steps in this direction are being made. The Internet Engineering Task Force (IETF) has defined an extension to the (Generalized) Multi Protocol Label Switching (G-MPLS) technology, called inter-Autonomous System (AS) G-MPLS, which enables the establishment of inter-carrier, explicitly routed connections with stringent quality of service (QoS) and availability constraints [3]. Recently, the authors of [4] have proposed additional extensions to the MPLS/G-MPLS technology in a multi-AS environment, in order to enable automatic provisioning of inter-domain TE services. The idea is to introduce a distributed inter-provider *service plane*, coupled with a Path Computation Element (PCE)-based control plane, through which providers interact by discovering carrier service elements, by composing the service elements into a multi-domain service, by instantiating and enabling the service, and finally by triggering management and network plane operations to finally establish and maintain the connection. In this framework, routing is source-based at the AS-level and distributed at the router-level. Some form of cooperation among providers is needed to override privacy, billing and monitoring issues and for managing service-related data. Hence, we believe that the proposed architecture is of great interest for a provider alliance agreement.

It should be said that a provider alliance architecture is not meant to be applied to the whole Internet, but to a subset of top-tier AS providers that share common issues in offering cross-border connection-oriented services in an automated way.

In this paper, we tackle the AS-level source routing problem that arises at the service plane of the provider

* Corresponding author. Tel.: +33 145818399.
E-mail address: stefano.secci@telecom-paristech.org (S. Secci).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.04.002

alliance architecture of [4]. The routing requirements differ from those of classical multi-constraint QoS routing problems in that the routing algorithm should scale with directional metrics and should take advantage of pre-computation. In Section 2, we present the technical context. In Section 3, we define the requirement for AS-level source routing algorithms. In Section 4, we present possible algorithms and position our contribution. In Section 5, we propose our 2-step algorithm, composed of a breadth-first search part with branch pruning, and a feasible route matching part; results from simulations are reported in Section 6. In Section 7, we indicate how to adapt the algorithm in order to take into account diversity constraints, useful to offer path diversity for reliability requirements and for accelerating the inter-layer communications of the architecture of [4]; results from simulations are reported in Section 8. Finally, Section 9 concludes the paper.[1]

2. Context

The MPLS/G-MPLS architecture allows the establishment of Label Switched Paths (LSPs) within provider boundaries. The MPLS-TE/G-MPLS protocol family intrinsically includes TE features, enabling the routing of LSPs explicitly taking TE constraints into account. Further extensions, detailed below, support the configuration of inter-AS LSPs [6].

The RSVP-TE signaling protocol [7] is used to establish MPLS/G-MPLS LSPs. The inter-AS LSP signaling can be done in three different ways:

- *LSP Nesting:* An intra-domain LSP is used between domain border routers to transport inter-domain LSPs sharing a common intra-domain subpath.
- *Contiguous LSP:* A single end-to-end LSP is signaled across the domains. There is a single signaling session between the head-end and the tail-end routers.
- *LSP Stitching:* In this mode, the local intra-domain LSPs are signaled separately, and then stitched at the boundaries to form a single inter-domain LSP.

2.1. Inter-AS LSP computation

An LSP is to be signaled over a pre-computed (router-level) path. A head-end router has full topology visibility within its domain boundaries, and hence can only compute an end-to-end intra-domain path, but not an end-to-end inter-domain path. Two methods can be adopted for the inter-AS path computation:

- The per-domain path computation method. The source or ingress router determines the next domain and the ingress router in the next domain, and computes the corresponding subpath. Then the path computation is moved to the ingress router of the next domain (by the signaling protocol), and so on up to the tail-end router. This simple method does not allow the computation of a shortest inter-domain path and can lead to several crankbacks that might affect the stability of the control plane.
- The cooperative PCE-based path computation method. It takes as input the AS chain – i.e., the succession of ASs to be crossed – and relies on computation entities present in each AS, the PCEs, to collaboratively compute an inter-AS shortest path along the given AS chain.

As highlighted in [4], the cooperative PCE-based method is preferred to allow a composed end-to-end service billing. In the PCE-based architecture [8], the PCEs serve requests sent by Path Computation Clients (PCCs) – i.e., routers or switches – using information in the local TE database. A PCE can query the PCEs of other domains to collaborate in this computation, acting in turn as a PCC; a PCE communication protocol (PCEP) [9] was defined to relay these request and answer messages. In the inter-domain path computation context, the Backward Recursive Path Computation (BRPC) [16] seems to be the procedure that meets best the operator and the supplier requirements in terms of complexity and network information hiding. It consists of computing iteratively, at each PCE of the AS chain and starting from the tail-end AS, an inverse tree of constrained shortest paths, with one branch for each ingress AS Border Router (ASBR) – and toward the destination. The tree is sent back to the previous AS, which does the same, and so forth up to the source AS. Obviously, at least one PCE is required in each domain. No TE information exchange is required between PCEs.

2.2. Service plane related extensions

At the IETF, standardization efforts have led to the definition of inter-AS LSP computation and signaling protocols and policies. However, some points are missing for the deployment of inter-provider G-MPLS network services. First, for the PCE-based architecture, the standardization does not indicate how the input AS chain is calculated. Then, being the set-up of an inter-provider tunnel subject to strong business, security, and confidentiality aspects, a trusted multi-provider service architecture would be needed to ensure billing, and to manage routing and signaling requests at provider boundaries.

These procedures are beyond the scope of the IETF; they have been defined within the ACTRICE[2] project. The authors in [4] introduce the notion of a inter-provider service plane and structure the lifecycle of an inter-provider G-MPLS LSP service with seven functional steps. The service plane brings providers interested in settling inter-AS network services together; it manages inter-provider *service elements* through which each provider announces its service offer in terms of Service Level Specifications (SLSs) – e.g., bandwidth, delay, and reliability level – and potentially according to an adopted business model of monetary costs. The authors indicate the IPsphere Forum framework [17] as a potential framework implementing such a service plane.

[1] A preliminary version of the contents of this paper have been presented in the 2008 International Communication Conference (ICC 2008) [1] and QoS-IP/IT-NEWS 2008 workshop [2].

[2] ACTRICE was a project funded by ANR, the French research agency.

Since the service elements announce per-LSP resource availabilities – and not global transit resource availability, and so avoiding giving an excessive insight in the provider network – at the service plane their SLS setting is to be updated with a longer time scale than that of physical resource availability changes.

It is worth resuming the seven functional steps proposed in [4], highlighting those of interest for our paper.

1. *Service Discovery:* The inventory of all the service elements offered by the providers of the alliance is acquired. This allows the construction of the weighted AS graph, which therefore represents a partial vision on the real interconnection topology limited by what announced via the service elements.
2. *Service Elements Composition:* The service plane is asked for a constrained shortest AS path (for point-to-point tunnels) or AS tree (for multipoint tunnels). It consists of a composition of service elements – following the idea described in [10] – at the source AS. An example of service element composition – for a LSP from node R1 to node R2 – is depicted in Fig. 1.
3. *Service Instantiation:* The point availability of the service elements composing the AS chain is verified. A Service Identifier (SID) is generated to identify the service during its lifecycle, and distributed among the involved ASs through the service layer. Every involved AS sends back a message to grant/refuse the availability of the required service element, and to possibly negotiate some SLSs or (if allowed) the cost of the service.
4. *Service Activation:* The service establishment is triggered: an activation message is distributed within the service plane to all the ASs, including also the SID. Then, the service plane sends to the management plane the filtering policy for the SID, useful for filtering future inter-AS PCEP and RSVP-TE messages. If this is successful, the management plane configures the head-end router in the network plane, establishing the inter-AS LSP, passing the SID, the AS chain, and the TE parameters.
5. *Path Calculation:* The inter-AS path is computed at the network plane. The PCE-based architecture computes the router-level path. PCEP messages are exchanged among PCEs while executing the BRPC algorithm. PCEP signaling is extended to transport the SID to allow filtering operations at the AS borders.
6. *Service Signaling:* At this step the inter-AS LSP is contiguously signaled at the network plane. The inter-AS RSVP-TE (Path and Resv) messages are also extended to transport the SID so as to allow filtering operations.
7. *Service Maintenance:* Once in operation, the inter-AS LSP may fail or be closed. A particular protection strategy may be provided in case of failure. If a failure cannot be recovered, a status message is sent to the service plane and the source AS is notified and may proceed with a new service request.

The network plane extensions (essentially, the last three steps) have been implemented and validated in a testbed (see [5]). In the rest of the paper, we focus on the service element composition step at the service plane, also taking into account its interaction with the instantiation step to accelerate the provisioning process.

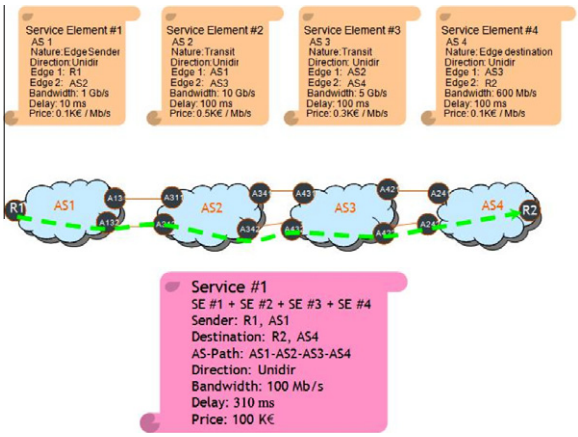


Fig. 1. Inter-provider network service composition at the provider alliance's service plane [4]. Note that the service request can ask for fewer resources than those announced via the service elements.

3. AS-level routing requirements

The provider alliance architecture gives specific routing requirements for service element composition algorithms:

1. *Policy routing:* The source AS should be able to apply local policies to influence the inter-AS route local selection, while having the highest possible visibility on inter-provider AS-level routes.
2. *Directional metrics:* A shortest path algorithm with multiple constraints should deal with a graph weighted with service elements' parameters, which are directional in the sense that they are applied to a 3-tuple ingress node – transit AS – egress node, where the nodes can represent either ASs or neighbour ASs' ASBRs or group of ASBRs.
3. *Pre-computation:* The presence of computation servers, the PCEs, may allow a reduction the online time complexity of the routing algorithm by pre-computing a part of the job.
4. *Multipoint routing:* So as to cope with the broad class of inter-provider services, the AS-level routing algorithm should encompass both point-to-point and point-to-multipoint inter-AS routes.
5. *Route diversity:* For each request, the source AS should select a set of possible routes with a certain degree of diversity for at least two reasons:
   - to decrease the request blocking probability by sequentially testing feasible routes that do not share critical common paths, so as to avoid inter-plane composition → instantiation → composition signaling loops;
   - to offer diverse paths for service requests with a certain degree of reliability so as to provide path protection mechanisms.

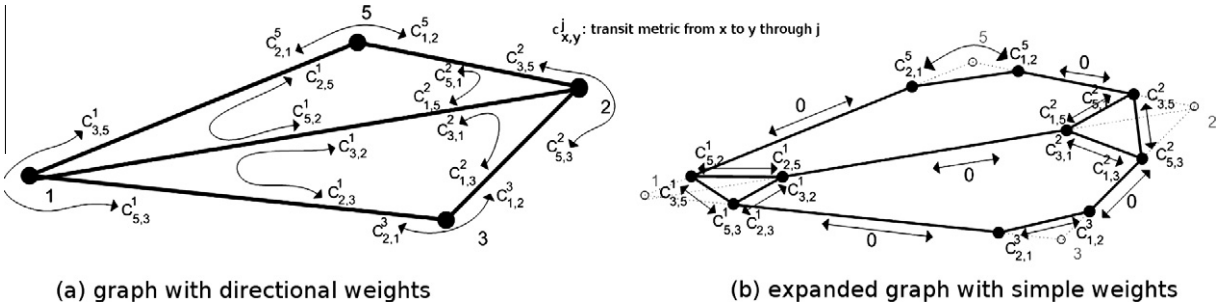


Fig. 2. Example of graph extension required to apply directional metrics to an edge-weighted graph.

In the following, we discuss the pertinence of the first four requirements (the fifth one is discussed in Section 7), and the algorithmic implications.

3.1. Policy routing

The first requirement implies that the routing decision is not distributed. This guides us toward a source-based algorithm, executed at the source AS that disposes of all the required routing information (AS connectivity graph and service elements). Based on this information, an AS might apply local policies, potentially hidden to the other ASs, and influence the routing decision by pruning the graph. The importance of the first requirement led to the definition of a functional policy architecture in the recent RFC 5394 [11] (related to the PCE architecture), which states: 'Network operators require a certain level of flexibility to shape the TE path computation process, so that the process can be aligned with their business and operational needs. Many aspects of the path computation may be governed by policies'. The idea is to let providers maintain a level of arbitrarity in the routing choice similar *yet broader* than that granted by the local preference in BGP routing.

3.2. Directional metrics

Within the service architecture described in the previous chapter, via the Service Elements each AS announces different transit costs and capabilities as function of both the entry and the exit ASs or ASBRs. Upon arrival of a request, a specific agent at each AS (called ASA) employs this information to compose the service elements.

Definition 3.1. A *directional arc* denotes a succession of two inter-AS logical arcs linking three AS-nodes.

The second requirement specifies that the adopted routing algorithms should deal efficiently with directional metrics. There are two possible ways to meet this requirement, either executing classical constrained shortest path heuristics on the pruned graph, or designing a *search algorithm* to explore the graph following the metric directions. In the first case, in order to deal with directional metrics, the original graph should be extended, as depicted in the example of Fig. 2: each AS node is to be exploded in a number of virtual nodes equal to the number of neighbors it is connected to. Then, directional metrics are to be applied to simple arcs connecting these new virtual nodes, while null metrics are to be applied to arcs connecting virtual nodes related to different originating nodes.

The AS graph having a scale-free nature (i.e., a few nodes attract most of the arcs), those few connected ASs that still occupy a key position in the graph would find in directional policies the most proper means to attract connections. We empirically discovered[3] that in a recent AS graph with n ASs, an optimistic approximation for the average degree of an AS-node can be $\sqrt[3]{n}$ (still more optimistic for those hub top-tier ASs that would be likely to participate in a provider alliance). This suggests that the aforementioned extension requires approximately $n\sqrt[3]{n}$ new nodes and arcs, which implies that classical QoS routing heuristics with at least $O(n^3)$ time complexity on normal graphs would pass to $O(n^4)$ on an AS graph with directional metrics.

3.3. Pre-computation for QoS routing

Common QoS routing algorithms minimize generic link costs while being subject to several constraints. Such algorithms are generally heuristic in that their solution is suboptimal, since the problem is NP-hard. As the number of constraints (additive, multiplicative, diagonal, etc.) used to guarantee a certain performance to QoS paths (delay, jitter, bandwidth, protection, etc.) is expected to increase (normally more than two constraints), pre-computation schemes for QoS routing are highly desirable to reduce the online computational complexity, i.e., the post-request algorithmic complexity [18].

The idea is to let some routing tasks be performed in advance so as to promptly provide a satisfactory path upon request. In practice, for source routing, it is possible to design an algorithm with a pre-computable initialization procedure independent of the QoS parameters of a path request. In our provider alliance architecture, route computation resources are potentially available at the PCEs,

[3] We extracted AS adjacency information dumping the data in [25] in March 2008; these adjacencies are obtained by inspecting AS paths of public some BGP routing tables, and therefore do not indicate all the real interconnections but only the visible ones (hence the following average degree approximation is optimistic).

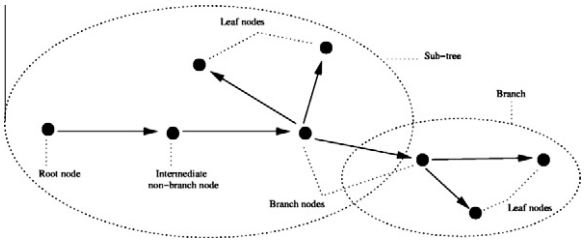


Fig. 3. Point-to-multipoint tree components.

which may be queried for such local pre-computation tasks.

3.4. Multipoint routing

Besides backhauling and inter-provider VoIP gateway interconnection, important services requiring inter-provider connection-oriented services are HD video content distribution, e.g., for VoD, Video streaming, telemedicine, teleconference. Such services require point-to-multipoint (P2MP) connections from one source to many destinations. Recent works carried within IETF have extended the MPLS-TE architecture in order to offer point-to-multipoint tunnels (i.e., P2MP TE-LSPs) [13], and have assessed the applicability of the PCE architecture for P2MP TE-LSPs [14].

An agreed taxonomy is needed to identify the elements of a P2MP path, called hereafter *AS tree* (Fig. 3):

- *Root/leaf node:* Source/destination node of a P2MP data transmission.
- *Branch node:* Node that performs data replication.
- *Intermediate node:* Non-branch and non-root node.
- *Bud node:* A leaf-and-branch node.

In a P2MP tree, a set of nodes can be classified as:

- *P2MP sub-tree:* Part of a tree such that the root or an intermediate node is connected to a subset of leaves.
- *P2MP branch:* Part of a sub-tree such that a single branch is connected to a subset of leaves.

4. Related work and our contribution

In the following, we discuss possible AS tree selection schemes that partially meet the above requirements, highlighting the open path for performance improvement.

4.1. Extensions of point-to-point algorithms

In the literature, many heuristics rely on point-to-point algorithm extensions. Most of them can be classified under the following two classes.

4.1.1. Irrespective Routes Computation with Post Merging (IRC-PM)

A simple method relies on the following steps:

- compute the shortest route subject to all constraints for each leaf AS;
- join the subroutes of the routes sharing arcs (directional arcs for our case).

We refer to this algorithm with the acronym IRC-PM. The resulting AS tree has sparse branches in sub-optimal positions. It is important to remark that resources (e.g. bandwidth) are shared on common links. Hence it is better to adopt algorithms which reduce the tree cost by encouraging arc sharing.

4.1.2. Iterative Point-to-Point selection (I-P2P)

Breaking the P2MP problem into multiple P2P route selections, inter-AS routes tend to share (directional) arcs:

- compute the shortest inter-AS route subject to all constraints from the root AS to a first leaf AS;
- assign null cost to all (directional) arcs taken by the first route and compute the inter-AS route to the second leaf;
- repeat the process for every leaf AS.

We refer to this algorithm with the acronym I-P2P. An advantage of this approach is that it still does not require the knowledge of all leaf ASs during the tree computation, while being more sensitive to link sharing than IRC-PM. However, the solution (and its optimality) strongly depends on the order in which routes to leaf nodes have been computed.

4.2. Constrained Steiner tree problem and heuristics

To avoid the dependency on leaf ordering, it is necessary to compute the optimal tree that spans all the destinations at once, i.e., the so called Steiner tree [19]. This optimization problem is known to be NP-hard, and is more complex when taking into account additive constraints. As the problem is not tractable for large instances, heuristics are needed. Heuristics for the Steiner problem have been studied extensively. A comparison of some of the main heuristics can be found in [20]; the two most promising source-based heuristics are in the sequel considered for the sake of comparison. The first one by Zhu et al. consists in an I-P2P variant, where a constrained version of Bellman-Ford algorithm is used iteratively [21]. The second is Kompella's centralized algorithm [22], which is:

- compute the all pair constrained shortest paths and build the closure graph of shortest paths from the root to the leaves;
- find the constrained spanning tree of the closure graph;
- expand the spanning tree avoiding possible loops.

The overall time complexity of Kompella's algorithm is $O(n^3D)$, where D is the integer value of the delay bound. For graphs with directional metrics, the time complexity of this heuristic after the graph expansion would therefore become $O(n^4D)$.

4.3. Motivations for improvements

We require a multipoint routing algorithm (requirement 4), which supports policy routing and is source-based (requirement 1), and which can handle multiple QoS constraints. As argued above, we need a source-based multipoint (or multicast) QoS routing algorithm.

QoS routing requires the support of multiple metrics to bound the final path solution. Some metrics are 'multiplicative' (e.g., bandwidth, class of service, etc.) and can be easily considered in source-based algorithms by pruning the graph. Other metrics are additive (e.g., secondary costs, delay, jitters, hop count, etc.) and are more complex to handle. Multi-constrained QoS routing has been an intensive research topic; several possible algorithms are well summarized and compared in [20]. The authors clearly point out that Kompella's [22] and Zhu's algorithms [21] can be considered as the two source-based multipoint QoS routing algorithms that offer the best performance, especially with respect to time complexity, optimality and QoS constraint multiplicity aspects. Both the algorithms may be adapted to solve our AS-level routing problem and will be considered in the following for performance comparison.

However, we can highlight that in our context these algorithms do not scale with directional metrics (requirement 2) and would both assume a time complexity bigger than $O(n^4)$. Moreover, they do not support pre-computation (requirement 3) and therefore can not reduce the online time complexity [18]. Finally, they do not seem to be effectively adaptable to support diversity constraints (requirement 5). In the following, we devise a novel ad-hoc routing algorithm that better meets the AS-level routing requirements. Its definition passes through the adoption of ideas of first-search approaches, namely the constrained k-shortest path A* prune [23] algorithm, and the usage of a pre-computable subalgorithm, i.e., the any-to-any unconstrained shortest path Floyd's algorithm [24].

5. The RCOM AS tree routing algorithm

To solve our specific routing problem we devise an ad-hoc heuristic called Route Collection and Optimal Matching (RCOM), composed of two steps:

1. *Route collection:* Some feasible point-to-point routes towards each leaf AS node are collected.
2. *Optimal matching:* The optimal matching of collected routes is reached minimizing the tree cost.

Unlike IRC-PM, RCOM retains a subset of feasible routes instead of only one route per destination. With respect to I-P2P, RCOM should be more flexible in branch and bud nodes placement, since it can reach a wider set of solutions.

As later discussed in Section 5.3, the more time consuming tasks can be pre-computed before the request arrivals (and independently of these requests).

Algorithm 5.1 (*ROUTE COLLECTION*).

PROCEDURE COLLECT (c,d,h,π)
[–] $\bar{f}$: per-destination vector with counters of found routes
[–] a, d_a, c_a: next directional arc, delay and cost of a
[–] M: destination group (set of leaf nodes)
[–] SPC: shortest path cost matrix

if h = H

then
- **if** $\exists!$ leaf $d \mid c + SPC(\pi[h], d) < \upsilon(d)$ **then** add π **to** ζ_{cand}
- **if** $\pi[h] \in M$ **then**
  - **if** $c < \upsilon(\pi[h])$ **then**
    - add π to ζ_{sel}
    - $f(\pi[h]) \leftarrow f(\pi[h]) + 1$
    - **if** $f(\pi[h]) \geqslant F$ **then** update $\upsilon(\pi[h])$

else
- **for** $i \leftarrow 1$ **to** N
- **do**
  - **if** i adjacent to $\pi[h]$, and $i \notin \pi$ **then**
    - $\pi[h+1] \Leftarrow i$
    - $a \leftarrow (\pi[h-1], \pi[h], \pi[h+1])$
    - **if** $h = 0$ **then** *Collect* $(c, d, h+1, \pi)$
    - **else if** $d + d_a < D$ **then** *Collect* $(c + c_a, d + d_a, h+1, \pi)$

main
$H \leftarrow 1$, $\bar{\upsilon} \leftarrow \bar{\infty}$, $\zeta_{cand} \leftarrow \{\pi_0 = (root)\}$
while $\zeta_{cand} \neq \emptyset$ **or** $H < H_m$ **do**
- extract a subroute π from ζ_{cand}
- *Collect* $(cost(\pi), delay(\pi), H-1, \pi)$
- $H \leftarrow H + 1$

5.1. Route collection

To collect the per-destination routes set, we devise an ad hoc breadth-first-search algorithm with limited depth. It starts at the root, moves to unvisited neighbors, collects the routes if a destination is attained, and so on, until no longer routes can be collected. It stops at a given number of hops or during the search by pruning branches depending on additive metric and cost bounds.

This approach was inspired by the A* prune algorithm [23], proposed to solve the constrained k-shortest paths problem. Our approach differs from it in that:

(i) since the final objective is the selection of the optimal tree, further pruning (besides that on the additive metrics) depending on the route cost is performed, giving priority to the least hop routes;
(ii) given that there is no need to sort the candidate routes (as A* prune does), the number k of shortest routes is not fixed and all the experienced (feasible) routes are collected (i.e., we do not need a best-first search approach).

5.1.1. Collection algorithm

Let $\bar{v}$ be a threshold cost vector with one entry per destination. Each entry is a threshold re-calculated for each new route collection. The starting values are infinite. Then, an entry is initialized when at least F routes have been collected for that destination; F has to be chosen conveniently (we use $F = \sqrt[3]{n}$ in our simulations). Each threshold is calculated as the average cost of a subset of the F routes. In order to avoid taking into account the routes with a too high cost with respect to the others, for the threshold computation, within the first F routes we consider only those with a variance on the average cost less than the average of this variance. In this way, the threshold has a decreasing trend, with a starting value not excessively high. Therefore, the least hop routes are privileged because the cost bound is higher in the first hops. Favoring routes of a few hops is a suitable approach for our specific problem, since long routes crossing several ASs risk having a small number of arcs joint with the previously selected ones, and tend to have very high costs. In this way we try to cut a lot of branches that would have been considered by general purpose solvers for an exhaustive optimization.

Definition 5.1. A *projected cost* of a subroute is the sum of the current subroute cost and the cost of the shortest path from the tail of the subroute towards the leaf node.

It requires a pre-computation of the shortest paths' costs from all intermediate nodes towards the leaves (see Section 5.3).

Therefore, the information required at the root node is the AS graph weighted with directional metrics, the constraints, and the shortest path cost matrix from any node to any node. The pseudo-code of the algorithm is given in Algorithm 5.1. The search starts looking for feasible routes at 2 hops, then 3, and so on. At every iteration, the search looks only at those routes with equal hop number H, up to a given bound H_m. At every iteration, the subroutes in the set ζ_{cand} are the starting point of the search. At every call of the procedure COLLECT (), c and d are the cumulative cost and delay of the route handled by the current route vector π with h hops number. When visiting the root neighbors ($h = 0$), π has only the root, and the delay is not verified. Then, the function recursively visits every neighbor of the subroute tail node, updating π, and evaluating the route feasibility on the cumulative delay. At the H^{th} hop, the route is collected in the set ζ_{sel} if a leaf is visited, if its cost is less than the threshold, and if the delay bound is respected; it is also added to ζ_{cand} for further expanding and possible selection in the next hop only if, for at least one destination, its projected cost is equal to or less than the threshold.

5.2. Route matching

The routes in ζ_{sel} define a subgraph built as a superimposition of their directional arcs. The optimal tree is therefore the least-cost composition of directional arcs linking the root to the leaves within this subgraph, solvable via Integer Linear Programming (ILP) with a low complexity given the limited size of ζ_{sel} and given that there is no need to check the additive constraints any longer. Indeed, forcing each destination to be crossed by at least one route, we assure that the leaves are reached and the constraints are satisfied.

5.3. Complexity and pre-computation

Therefore, the collection algorithm dominates the complexity of the RCOM algorithm. The majority of the time is spent in computing the (unconstrained) shortest path costs, which are needed to determine the projected costs, in the collection algorithm. We propose to pre-compute them, prior to any request, and after any topological and cost update. This can stand when costs and topology are expected to change much less frequently than the request arrivals, and this hypothesis would apply to the presented multi-provider architecture. Hence, prior to any request (characterized by root, leaves, and end-to-end constraints) a simplified version of Floyd's algorithm [24] can be used in order to pre-compute the cost of the shortest paths (SPC matrix in Algorithm 5.1) from any node to any node (A2ASP). Floyd's algorithm takes $O(n^4)$ time to compute (see the reasons in Section 4.2). The subsequent breadth-first search would have, without pruning, a time complexity of $O(n^{\frac{1}{3}H_m})$ for the worst case, approximating the base (branching factor) to $\sqrt[3]{n}$. Because of pruning, it is more efficient than that.

To improve the execution time, A2ASPs computation should be pre-computed, prior to any request, and triggered by topology and cost update. In this way the post-request worst case complexity of the collection becomes $O(n^{\frac{1}{3}H_m})$.

For the sake of comparison, the centralized heuristics proposed so far for constrained multicast routing, as those in [20], do not have a sub-algorithm independent of the constraint values. For example, Kompella's algorithm computes *constrained* A2ASPs to build the closure graph with a complexity proportional to the delay bound (see paragraph Section 4.2). Or, Zhu's algorithm [21] uses as starting point a least-delay spanning tree. Both Kompella's and Zhu's algorithms have an overall complexity equal to the post-request complexity, which is, for a graph with directional metrics, bigger than $O(n^4)$ [20].

Note 1: It is worth mentioning that given the breath-first search nature of the collection algorithm and the additive constraint transparency of the route matching, an extension of the RCOM approach to multiple additive constraints would scale with the number of constraints (besides the delay).

Note 2: A restriction to a single destination for P2P paths is straightforward and slightly decreases the RCOM complexity: the matching task is trivialised to the choice of the shortest route among those collected.

6. Performance evaluation I

We compare the described algorithms in terms of optimality and execution time, and analyze the characteristics of the selected AS trees. We chose to use realistic topologies: we dumped the AS whois database containing interconnection data available at [25]. As stated before,

our architecture is not meant to be used at Internet-wide scale (even the PCE-based one is not meant to be) but on a set of ASs collaborating to a common service plane. We use Internet topology estimations in order to be as realistic as possible. Two topologies are considered. The first one is built as following: among all the ASs, only those with at least 7 adjacencies are kept (in this way, we select those AS carriers potentially interested in inter-domain tunnel provisioning); then, only those ASs with more than 2 adjacencies with the other ASs are kept for the final topology. The final topology, called ATL7, has 643 AS-nodes. The second topology, TOP300, is similarly built with the 300 most connected nodes of ATL7.

6.1. Directional metric setting

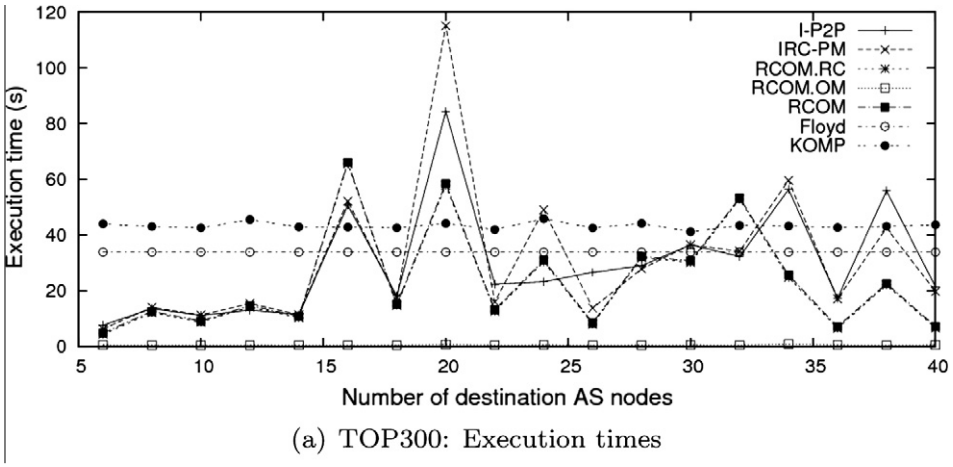
We weighted the two topologies with directional metrics for capacity, transit costs and delay bounds.

6.1.1. Capacities and costs

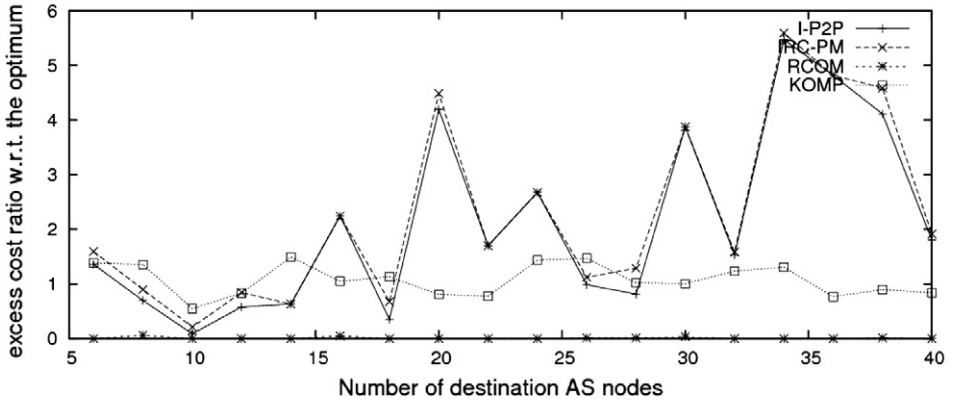
For capacity and cost assignment, we classify as Tier-3 (T3) an AS with a number of interconnections less than the average, Tier-1 (T1) one with a number of interconnections with non-T3 ASs over the average, and Tier-2 (T2) the remaining ones. This deviates from the conventional terminology, which does not apply to our framework since we overtake the BGP-policy-based peering and customer–provider relationships. Moreover, we prefer a degree-based instead of a betweeness-based ranking because for the latter we have not a convergent set of shortest (intra-alliance) routes – in fact, they are potentially computed dynamically for each new request.

Considering a T3 not able to offer as much connectivity as T2s and T1s do, and the same for T2s with respect to T1s, we assign capacities to inter-AS links normally with different averages and deviations as indicated in [2].

Moreover, since the bottleneck is not at the intra-AS but inter-AS links, and since lower transit costs come with a higher availability, we approximate the directional transit cost equal to $K\frac{\log[\beta min(C_{i,k},C_{k,j})]}{\beta min(C_{i,k},C_{k,j})}$, $K = 10^5$, for a directional arc (i,k,j) with link capacities of $C_{i,k}$ and $C_{k,j}$. We chose this function so that it decreases more than linearly as function of the product between the requested bandwidth and the minimal inter-AS capacity: the more available the transit capacity is, the less expensive the service element is; the more bandwidth is sold, the lower the per-bandwidth cost is. We halve the cost when the transit involves two AS of the same provider, and set it to zero when all three of them do, so as to try to be more realistic (the per-provider AS grouping is a publicly available information).



(a) TOP300: Execution times



(b) TOP300: Tree cost gaps w.r.t. (with respect to) the optimum

Fig. 4. Results for TOP300 topology.

6.1.2. Delay bounds

The significant factor affecting the end-to-end delay is the propagation delay [12]. According to the whois tags, we assign ASs to a country. Since providers can operate in several continents, we calculate the directional transit delay bounds independently of the geographical position of the transit nodes, but as a function of the position of edge nodes, following a normal distribution with averages and deviations chosen on the basis of experimental round trip times (see [2]).

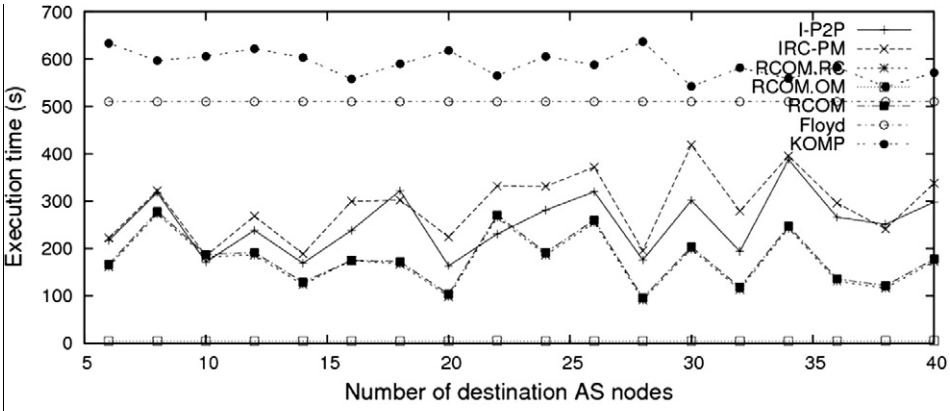
6.2. Algorithmic performance

We test the algorithms for different destination group sizes. Root and leaves are generated randomly. The delay bound is set to 1.5 s and the bandwidth to 6 Mb/s. Simulations run over a 3.4 GHz CPU, with 1 MB cache.

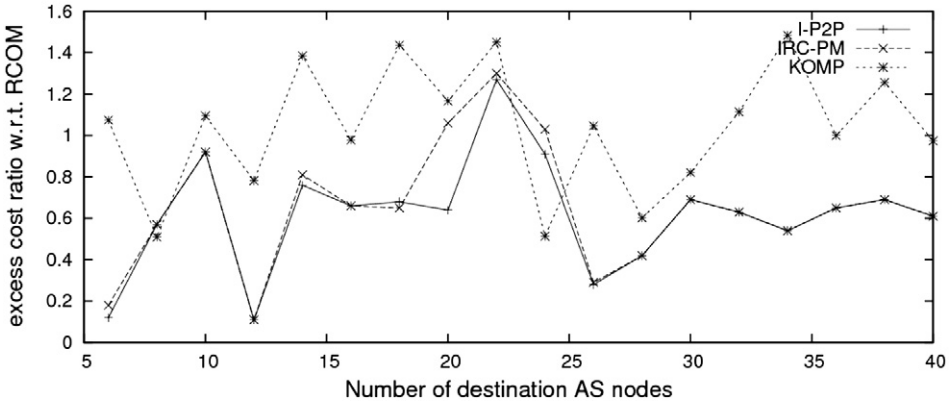
6.2.1. Execution time

Figs. 4a and 5a display the execution times obtained for the TOP300 and ATL7 topologies as function of the size of the destination group. For ATL7 H_m is set to 8 (sufficient for this topology), while for TOP300 it is set to 5 (also sufficient because of the smaller diameter). The case of the optimal approach is not plotted: it grows more than exponentially with the number of nodes. For RCOM we display: the total time ('RCOM'), the times of the collection (+ '.RC') and matching (+ '.OM') procedures. The cases of 'IRC-PM', 'I-P2P' and Kompella's ('KOMP') algorithms are also plotted. The time of the A2ASP computation ('Floyd') is separated since we assume that it can be pre-computed; in fact, it is constant since it is independent of the request parameters. We can assess that:

(i) The complexity of the RCOM route matching part becomes more negligible the more the topology grows.
(ii) As expected, KOMP is lower bounded by Floyd since it implements a constrained version of Floyd.
(iii) Including the A2ASP computation, RCOM has an execution time comparable to that of KOMP; without, it has almost always the lowest time.
(iv) I-P2P and IRC-PM have a similar behavior, and both seem to scale worse than the other algorithms with the destination group and topology sizes.
(v) Larger instances (with more AS-nodes) do not worsen the RCOM and KOMP complexity.



(a) ATL7: Execution times



(b) ATL7: Tree cost gaps w.r.t. RCOM

Fig. 5. Results for ATL7 topology.

6.2.2. Optimality

Fig. 4b displays the excess cost ratio (i.e. $1 \rightarrow 100\%$) with respect to the optimal solution for TOP300. For ATL7 this could not be computed, but Fig. 5b displays the excess cost with respect to RCOM for ATL7. We can assess that:

(i) For the TOP300 topology, RCOM yielded an optimality gap largely under 10%.
(ii) KOMP has always at least 50% excess cost with respect to RCOM.
(iii) I-P2P and IRC-PM give similar solutions.

6.3. Solution characterization

We can further characterize the solution with respect to the following aspects.

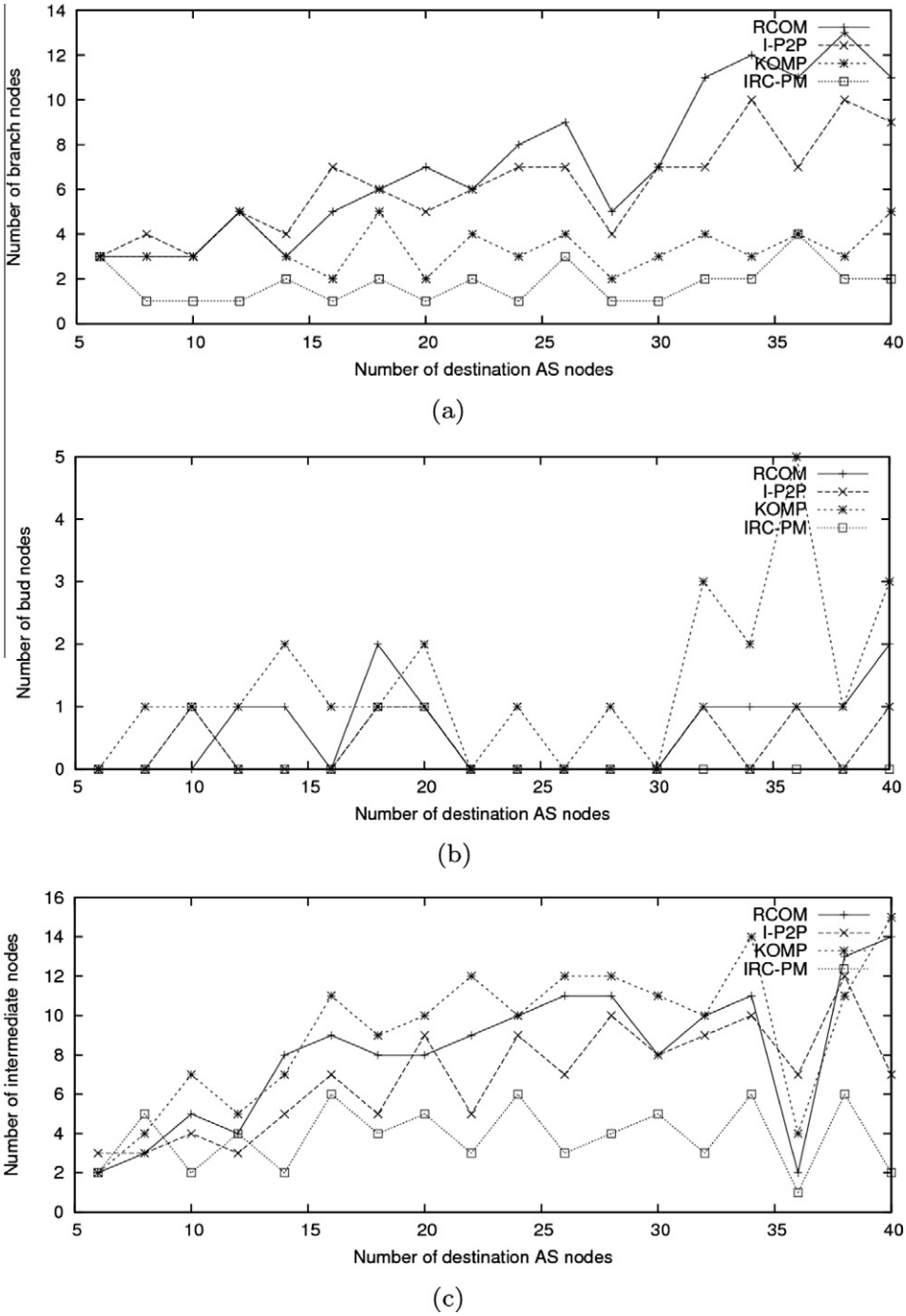


Fig. 6. Solution tree node characterization.

6.3.1. Node type

Fig. 6 displays the number of branch, bud and intermediate nodes. The ATL7 results are considered. We can assess that:

(i) for RCOM and I-P2P the number of branch nodes increases with the number of ASs;
(ii) the number of branch nodes is lower bounded by KOMP and IRC-PM;
(iii) interestingly KOMP often gives more bud nodes than the other algorithms;
(iv) on the contrary, RCOM often has more branch nodes and less bud nodes than KOMP;
(v) in terms of intermediate nodes, RCOM represents a good trade-off between I-P2P and KOMP.

(ii) and (iii) may be explained as follows. While RCOM has an unconstrained A2ASP pre-computation for projecting costs during the constrained exploration and pragmatically discarding routes, KOMP has a constrained A2ASP computation for producing a closure graph where the minimum spanning tree is computed. The KOMP algorithm seems to fall easily to local minima corresponding to longer routes. Therefore, the possibility of branching at leaves is higher; indeed, the closure graph is not sensitive to the real hop number.

6.3.2. Tree slimness

Let the utility of a directional arc be the number of destinations it serves minus one. Let the tree slimness be defined as the ratio between the sum of all these utilities and the number of directional arcs the tree is composed of. Slimness expresses how much the selected tree is exploited, or how much the selected tree has directional arcs that are much used to reach several destinations. This is not intended as an overall evaluation parameter of a tree; however, it can be deduced that the less optimal a tree is, the smaller its slimness is expected to be. We are motivated in analyzing this parameter because in a multi-layer network – major application domain of these algorithms – a computation in one layer can be followed by computations in other lower layers along the routes chosen in the upper layer. Hence, the slimmer the tree is, the simpler the under-layer path computation (and maybe signaling) might be in the case of multi-layer networks.

Fig. 7 displays the slimness of solution trees obtained for the ATL7 graph.

We can assess that:

(i) RCOM offers the best slimness, i.e. the best utility of the tree;
(ii) KOMP offers the worst slimness;
(iii) I-P2P and IRC-PM behave better than KOMP but worse than RCOM.

7. Route diversity in AS-level routing

We now deal with the last requirement in Section 3, route diversity. As previously mentioned, a set of route alternatives (P2P AS paths or P2MP AS trees) should be selected to offer enough diversity for a successful route selection, or to set-up disjoint tunnels for protection purposes. For the first case, the route alternatives should be computed and tested one after the other to avoid signaling loops between the composition and instantiation steps. For the latter case, it is possible to compute disjoint AS paths or AS trees sequentially with RCOM, by collecting in the RC step only those paths or trees that are disjoint with the first one. However, this can lead to blocking if the second path cannot be placed. Such issues can be more readily avoided if the set of route alternatives are computed in parallel [14]. In the following, we first concentrate on the P2P diverse route selection problem, the extension to P2MP routes (AS trees) being straightforward.

Definition 7.1. *Diverse routes* do not share any directional arc.

As depicted in Fig. 8, forcing a directional disjointness, two route alternatives may concern the same AS-node, but involve different intra-AS directions and different inter-AS links. Note that only one route may be instantiated because of intra-AS resource availability. Indeed, different inter-AS directions can have different intra-AS resource availability (remembering that the real intra-AS resource

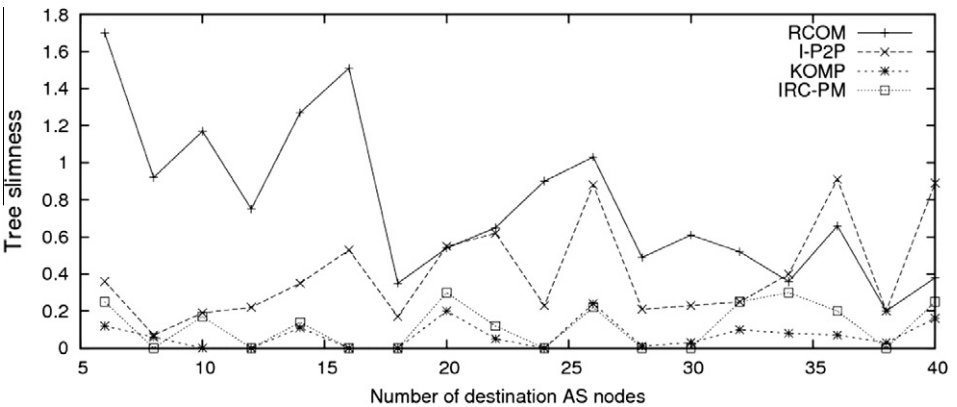


Fig. 7. Solution tree slimness as function of the destination group size.

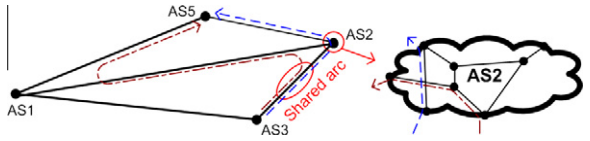


Fig. 8. Example of two diverse inter-AS routes.

availability is not visible to the other ASs, and that only per-LSP transit availabilities are announced with the service elements).

We believe that such an AS-level disjointness constraint is more pertinent than other forms, such as end-to-end disjointness. End-to-end disjointness at the AS-level would be, indeed, very hard to achieve. When two ASs are connected with a single inter-AS link, the end-to-end disjointness may not be guaranteed: this would be the case for most of the AS-node pairs in the Internet graph given its scale-free nature. In fact, the directional disjointness constraint exploits the scale-free nature of the AS graph, which presents a few AS hubs interconnecting many ASs.

7.1. Diverse AS-level routing problem

The diverse routing problem consists in selecting the less costly set of diverse routes satisfying a given connection request. The set of feasible routes ζ_{sel} can be collected with the Route Collection algorithm presented in Section 5.1. Then, a given number of diverse routes is kept, that is, a clique of diverse route has to be selected within ζ_{sel}.

7.1.1. Optimal clique selection

The next step consists in extracting the least cost clique of a diverse (collected) routes. Every route-element of ζ_{sel} has a cost and can be included in the final clique. We can see every route as a vertex, so that the least cost clique of vertices is the solution. In Fig. 9, e.g., we have a 5-route graph from which only three cliques of three vertices can be extracted. This problem is linked to the Generalized Minimum Clique Problem (GMCP), with a fixed clique size. The routes of ζ_{sel} are considered as vertices, which are connected only if diverse.

The optimal clique selection sub-problem can be solved by ILP. The GMPC considers weighted vertices and links, and is NP-hard [15]. In our case only vertices have cost, and it becomes a node-weighted MCP, which is still not polynomial, but less complex and treatable for a few hundreds of routes. Let f_i be a binary variable equal to 1 if $i \in \zeta_{sel}$ is a clique member. Let $s_{i,j}$ be a parameter equal to 1 if route i and route j are disjoint. The formulation is:

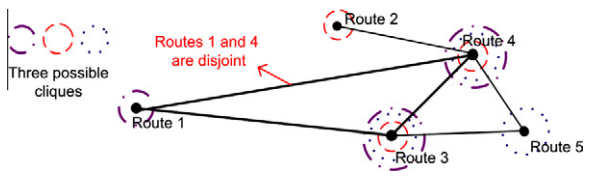


Fig. 9. Example of three possible cliques of three diverse routes in a 4-route graph.

$$\min \sum_{i\in\zeta_{sel}} f_i c_i \tag{1}$$

$$\text{s.t.} \sum_{i\in\zeta_{sel}} f_i = a \tag{2}$$

$$(a-1)f_i - \sum_{j\in(\zeta_{sel}-\{i\})} f_j s_{i,j} \leqslant 0 \quad \forall i \in \zeta_{sel} \tag{3}$$

$$f_i \in \{0,1\} \quad \forall i \in \zeta_{sel} \tag{4}$$

The objective (1) is the minimization of the clique cost. Eq. (2) sets a routes for the clique. Eq. (3) forces the clique membership. (4) sets the f binarity.

7.2. About route diversity for multipoint paths

As already mentioned, the extension of the AS-level routing algorithm to deal with the selection of several diverse AS trees is straightforward and not included. Two AS trees shall be considered as diverse if they do not share any directional arc. Please consider that, however, such a disjointness constraint may be too strict especially for small AS graphs with a few hubs. In such cases it might make more sense to consider as diverse the AS trees that do not share branch nodes, which may also decrease the computational complexity of the optimal matching step.

8. Performance evaluation II

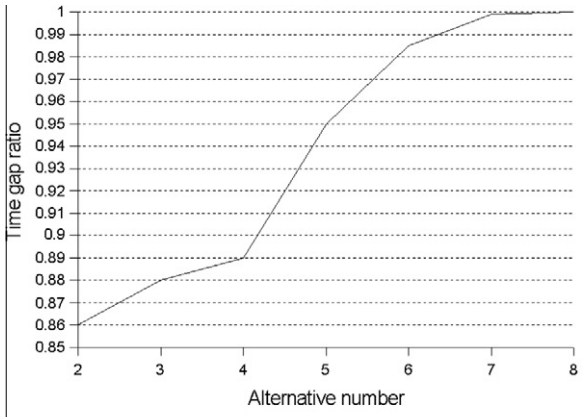
In order to test the diverse AS-level routing algorithm – nicknamed RECS (Route Enumeration and Clique Selection) in the following – on very large instances, this time the AS graph is built considering among all the ASs only those with at least 4 adjacencies (instead of 7 – then again only those with more than 2 adjacencies within the selection are kept in). The final graph has now 1716 ASs.

8.1. Algorithmic performance

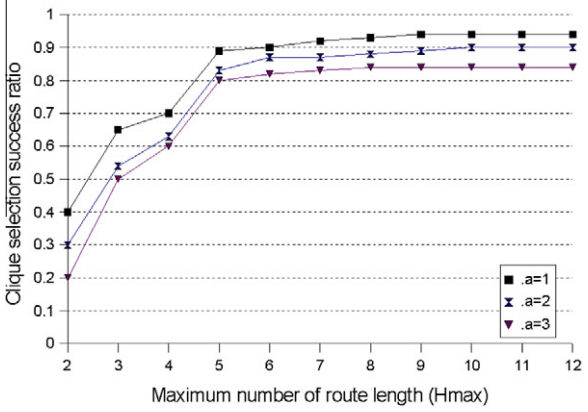
We tested the diverse routing algorithm against time complexity and optimality performance.

8.1.1. Time complexity

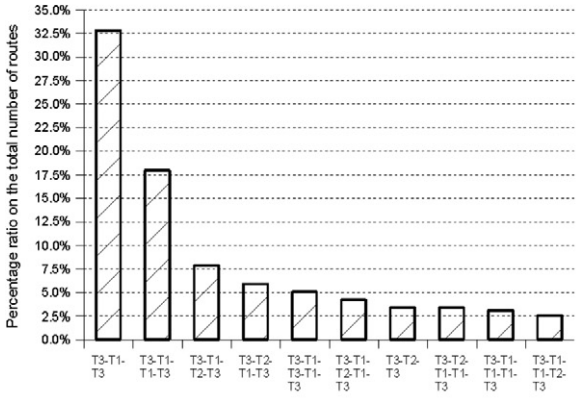
Fig. 10a displays the average execution time gap ratio between the proposed approach for diverse route selection (RECS) and the optimal result that could be obtained by ILP (CPLEX). The ratio is simply computed as $1 - t_{RECS}/t_{ILP}$, where t_{RECS} and t_{ILP} are the execution times under the two approaches; the results are displayed as function of a. Fifty successful simulations are considered. Fig. 10a displays two curves: the dotted one considers the A2ASP computation time in t_{RECS}, while the continuous one does not. Indeed, the ASAs should compute the A2ASPs off-line prior to inter-AS route requests. The higher the number of alternatives is, the harder the optimal approach: the RECS approach scales with the number of alternatives. Indeed, given that the number of collected routes remains always under 1000, the clique selection requires only a few solution searches. Obviously with the A2ASP computation time we have just a shift.



(a) Time gap ratio between RECS and the optimum (ILP)



(b) Success ratio performance



(c) Hierarchical routes ratio characterization

Fig. 10. RECS simulation results.

8.1.2. Optimality

We compare the average deviation of the selected clique cost using the RECS approach to that given by an optimal approach. Each entry of Table 1 indicates how many of the performed simulations per case produced a solution with an optimality gap within 5%, 15% or 100%. Three cases are considered for 2, 4 and 16 route alternatives in the clique, with 50 simulations per case. For each case we show how often (in percentage) RECS solutions had an optimality gap that falls in the three intervals. We can assess that:

Table 1
RECS optimality evaluation.

	<5%	<50%	<100%
$a = 2$	80%	93%	99%
$a = 4$	75%	80%	99%
$a = 16$	69%	77%	96%

(i) RECS can give a solution with an optimality gap within 5% more than once every two times;
(ii) it can guarantee a solution with an optimality gap within 100% for practically all the requests;
(iii) better optimality gaps can be obtained with a small number of route alternatives, even if large numbers of alternatives still reach an optimality gap within 50% most of the time.

8.2. Solution characterization

We can further characterize the solution with respect to the following aspects.

8.2.1. Connection admission

Fig. 10b shows the success ratio in selecting a route clique for three clique sizes ($a = 1,2,3$), as function of the upper hop bound, for 50 new simulations per case. We can assess that:

(i) the majority of ASs is attainable within 5 hops;
(ii) the exploration of the graph for more than 8 hops is not useful;
(iii) even for single-element degenerate cliques, a 100% success ratio was never reached because the bandwidth and the delay constraints limit the number of collected routes.

8.2.2. AS hierarchy

For 100 new successful simulations with a hop bound of 8, Fig. 10c reports the 10 most selected hierarchical route profiles. We can assess that:

(i) all the routes have T3s as source and destination ASs;
(ii) more than 80% of routes count less than 5 hops;
(iii) a significant percentage has only T1s transit nodes, while the others use at least one T1.

Less than 0.1% of ASs (the T1s) attract most of the traffic. Such results prove that assuming, as we did, a carrier hierarchy where top-tier ASs dispose of more resources and can apply lower prices, the economically feasible routes are attracted by top-tier ASs. This does not preclude, however, a lower-tier AS attracting more routes if it can tune transit rates efficaciously.

9. Summary

In this paper, we proposed heuristics for the AS-level source-based routing and diverse routing problems. The context of our work is a provider alliance architecture in

which routing is source-based at the AS-level and distributed at the router-level. In this context, we highlighted the peculiar AS-level routing requirements and positioned our contribution with respect to the state of the art.

We have showed that with our heuristics, pre-computation of some tasks can be performed, which drastically speeds up subsequent routing computations at tunnel request arrivals. All in all, by means of extensive simulations, we argued that:

(i) exploiting pre-computation, our approaches are faster than the well-known algorithms;
(ii) multiple additive constraints do not affect the asymptotic time complexity;
(iii) they often reach optimality, and have an optimality always largely under 10% on realistic AS graphs;
(iv) they produce efficient trees with respect to underlayer computation issues;
(v) AS-level diversity constraints can be included in the routing algorithm, and their consideration does not decrease the optimality and computational performance.

As a further work, in the framework of the European FP7 ETICS (Economics and Technologies for Inter-Carrier Services) integrated project, we are currently working to an evaluation of the algorithm in a testbed implementation coupling the service plane and the network plane extensions. Moreover, we are currently studying how joint static off-line reservation schemes should be implemented to allow a seamless instantiation of the Provider Alliance's service elements and how to motivate such a collaboration.

Acknowledgments

The authors thank the annonymous reviewers for their excellent suggestions for improvement, and Prof. Peter Weyer-Brown for his detailed review of the English. This work was funded by the European FP7 ETICS (Economics and Technologies for Inter-Carrier Services) project.

References

[1] S. Secci, J.-L. Rougier, A. Pattavina, AS tree selection for inter-domain multipoint MPLS tunnels, in: Proc. of 2008 IEEE International Conference on Communications (ICC 2008), Beijing, China, 19–23 May 2008.
[2] S. Secci, J.-L. Rougier, A. Pattavina, On the selection of optimal diverse AS-paths for inter-domain IP/MPLS tunnel provisioning, in: Proc. of 2008 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks (QoS-IP/IT-NEWS 2008), 13–15 Feb. 2008, Venezia, Italy.
[3] A. Farrel, J.-P. Vasseur, A. Ayyangar, A framework for inter-domain multiprotocol label switching traffic engineering, RFC 4726, Nov. 2006.
[4] R. Douville, J.-L. Le Roux, J.-L. Rougier, S. Secci, A service plane over the PCE architecture for automatic multidomain connection-oriented services, IEEE Communications Magazine 46 (6) (2008).
[5] A.P. Bianzino et al., Testbed implementation of control plane extensions for inter-carrier G-MPLS LSP provisioning, in: Proc. of 2009 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2009), Washington, USA, 6–8 April 2009.
[6] A. Farrel, A. Ayyangar, J.-P. Vasseur, Inter-domain MPLS and G-MPLS traffic engineering – resource reservation protocol-traffic engineering (RSVP-TE) extensions, RFC 5151, Feb. 2008.
[7] L. Berger, Generalized multi-protocol label switching (G-MPLS) signaling resource reservation protocol-traffic engineering (RSVP-TE) extensions, RFC 3473, Jan. 2003.
[8] A. Farrel, A. Vasseur, J. Ash, A path computation element (PCE) based architecture, RFC 4655, Aug. 2006.
[9] J.-P. Vasseur, J.-L. Le Roux, Path computation element (PCE) communication protocol (PCEP), RFC 5440, Mar. 2009.
[10] J. Xiao, R. Boutaba, QoS-aware service composition and adaptation in autonomic communication, IEEE Journal on Selected Areas in Communications 23 (2005).
[11] I. Bryskin, D. Papadimitriou, L. Berger, J. Ash, Policy-enabled path computation framework, RFC 5394, Dec. 2008.
[12] B. Choi, S. Moon, Z. Zhang, K. Papagiannaki, C. Diot, Analysis of point-to-point packet delay in an operational network, in: Proc. of INFOCOM 2004.
[13] S. Yasukawa, Signaling requirements for point-to-multipoint traffic engineered MPLS LSPs, RFC4461, Apr. 2006.
[14] S. Yasukawa, A. Farrel, Applicability of the path computation element (PCE) to point-to-multipoint (P2MP) multiprotocol label switching (MPLS) and generalized MPLS (G-MPLS) traffic engineering (TE), RFC 5671, Oct. 2009.
[15] A. Koster, S.P.M. Van Hoesel, A.W.J. Kolen, The partial constraint satisfaction problem: facets and lifting theorems, Operations Research Letters 23 (1998).
[16] J.-P. Vasseur et al., A BRPC procedure to compute optimal inter-domain TE label switched paths, RFC 5441, Apr. 2009.
[17] IP sphere framework tech. specification. Available from: <http://www.tmforum.org/ipsphere> (visited in Feb. 2010).
[18] A. Orda, A. Sprintson, Precomputation schemes for QoS routing, IEEE/ACM Transactions on Networking 11 (4) (2003).
[19] S. Chopra, M.R. Rao, The Steiner tree problem I: formulations, compositions and extension of facets, Mathematical Programming 64 (1994).
[20] H.F. Salama, D.S. Reeves, Y. Viniotis, Evaluation of multicast routing algorithms for real-time communication on high-speed networks, IEEE Journal on Selected Areas in Communications 15 (3) (1997).
[21] Q. Zhu et al., A source-based algorithm for delay-constrained minimum-cost multicasting, in: Proc. of INFOCOM 1995.
[22] V.P. Kompella, J.C. Pasquale, G.C. Polyzos, Multicast routing for multimedia communication, IEEE/ACM Transactions on Networking 1 (3) (1993).
[23] G Liu, KG Ramakrishnan, A*Prune: an algorithm for finding K shortest paths subject to multiple constraints, in: Proc. of INFOCOM 2001.
[24] R.W. Floyd, Algorithm97: shortest path, Communications of the ACM 5 (6) (1962).
[25] The CIDR report. Available from: <http://www.cidr-report.org> (visited in Feb. 2010).

Stefano Secci is currently a postdoctoral research associate at Télécom ParisTech – ENST, France, and George Mason University, USA. He obtained a Ph.D. in Computer Science and Networking from the same school, and a Ph.D. in ICT networking from Politecnico di Milano, Italy, in 2009. He received a M.Sc. in telecommunications engineering from Politecnico di Milano, in 2005. His Ph.D. and M.Sc. research activities mainly concerned Internet routing, Network Design and Optimization, and Future Internet architectures. He worked as research associate in 2005 at the Research Center in Decision Analysis (GERAD) in the Ecole Polytechnique de Montreal, Canada, and in 2006 at CNIT-DEI in Politecnico di Milano. He also worked as xDSL service engineer for Fastweb Italia. He has been visiting researcher at Warsaw University of Technology, Poland, and Q2S Center, NTNU, Norway. His research interests and activities currently cover Future Internet routing and traffic engineering and network neutrality. He is recipient of the NGI 2009 Best Paper Award.

Jean-Louis Rougier received his engineering diploma in 1996 and his Ph.D. in 1999 from Télécom ParisTech (formerly called Ecole Nationale Supérieure des Télécommunications). He joined the computer science and networks department of Télécom ParisTech in 2000 as an associate professor. He has been working ever since on routing and traffic engineering for networks in different context (Optical, Wireless Mesh networks, IP, Post-IP). He has been involved in several national and European projects and industrial collaborations. He is a technical advisor for several companies. His main research interest is currently on the evolution of architectures for the future Internet.

Achille Pattavina (M'85/SM'93) received the Dr.Eng. degree in electronic engineering from University La Sapienza of Rome (Italy) in 1977. He was with the same university until 1991 when he moved to the Politecnico di Milano, Milan (Italy), where he is now a Full Professor. He has been the author of more than 100 papers in the area of communications networks published in leading international journals and conference proceedings. He has been guest or coguest editor of special issues on switching architectures in IEEE and non-IEEE journals. He has been engaged in many research activities, including European Union-funded projects. Dr. Pattavina has authored two books, Switching Theory, Architectures and Performance in Broadband ATM Networks (Wiley, 1998) and Communication Networks (McGraw-Hill, 1st ed., 2002, 2nd ed., 2007, in Italian). He has been an Editor for Switching Architecture Performance of the IEEE Transactions on Communications since 1994 and Editor-in-Chief of the European Transactions on Telecommunications since 2001. He is a Senior Member of the IEEE Communications Society. His current research interests are in the areas of optical switching and networking, traffic modeling, and multilayer network design.