Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks

A. Sprintson[1] M. Yannuzzi[2] A. Orda[3] X. Masip-Bruin[2]

***Abstract*—We present a distributed routing algorithm for finding two disjoint (primary and backup) QoS paths that run across multiple domains. Our work is inspired by the recent interest in establishing communication paths with QoS constrains spanning multiple IP/MPLS domains. In such settings, the routing decisions in each domain are made by the *Path Computation Element* (*PCE*). We assume that the PCEs run a joint distributed routing protocol, decoupled from the BGP, which enables them to establish efficient paths across multiple domains.**

This study makes the following contributions. First, we present an aggregated representation of a multi-domain network that is small enough to minimize the link-state overhead, and, at the same time, is sufficiently accurate, so that the PCEs can find *optimal* disjoint QoS paths across multiple domains. Second, we present a distributed routing algorithm that uses the proposed representation to find disjoint paths in an efficient manner. Finally, we consider the problem of finding two disjoint paths subject to the *export policy* limitations, imposed by customer-provider and peer relationships between routing domains. We show that this problem can be efficiently solved by employing the concept of *line graphs*. To the best of our knowledge, this is the first scheme fully decoupled from BGP that enables to establish disjoint QoS IP/MPLS paths in a multi-domain environment with provable performance guarantees.

I. Introduction

In recent years there has been a significant amount of interest in establishing reliable communication paths with QoS constraints across multiple routing domains. This effort is facilitated by the current discussion in the Internet community [1] on extending the capabilities of MPLS networks across multiple domains, so that multi-domain Label Switched Paths (LSPs) with QoS guarantees can be established.

As the reliability and resilience to failures is a key concern for many applications, several connections will require establishing *primary* and *backup* LSPs that span multiple domains. In many cases the backup LSPs need to be established together with the primary LSPs. This proactive approach enables almost instantaneous restoration in the event of a failure, which is critical for real-time applications. In order to provide end-to-end performance guarantees, both primary and backup paths have to satisfy QoS constrains.

Accordingly, in this paper we focus on the problem of establishing two disjoint QoS paths across multiple domains. This problem, referred to as *Problem 2DP*, is considered in the context of the routing model inspired by the recently proposed *Path Computation Element* (*PCE*) based architecture [2]. Our goal is to develop a distributed routing algorithm with provable performance guarantees. This will allow to overcome the limitations of coarse-grained solutions such as those that arise by iteratively solving Problem 2DP on a per-domain basis [3]. We develop a routing model where each PCE is able to compute the *optimal* primary and backup QoS paths to any destination. One of the major advantages of our approach is that it avoids the well-known *trap topology* problems [4].

To achieve scalability and due to security and administrative considerations, routing domains do not advertise their internal structure, but rather supply an *Aggregated Representation* (AR) to the outside world. Accordingly, a key aspect in the design of distributed routing algorithms is to find an adequate AR that captures the availability of diverse QoS paths across multiple domains. However, there is an inherent tradeoff between the accuracy of the representation and the size of the required data structures. In this paper we consider a setting in which a reduced set of neighboring domains are willing to extend the reachability of IP/MPLS LSPs across their boundaries. This enables each domain to provide an accurate representation of its traversal characteristics, which, in turn, enables finding optimal disjoint paths across the network. This approach is consistent with that adopted by the IETF PCE Working Group (WG). The WG has recently stated that its efforts will focus on the application of the PCE-based model within a single domain or within a small group of neighboring domains, but it is not the intention of the WG to apply this model to the greater Internet [1].

In this paper we present a novel AR for a multi-domain network which is small enough to minimize the link-state overhead, and, at the same time, is sufficiently accurate, so that the PCEs can *optimally* find disjoint QoS paths across multiple domains. Our solution guarantees that the confidentiality and administrative limits are respected between domains (e.g., neither the internal topology nor the full IGP state of the domains can be inferred from their ARs). We also present a distributed routing algorithm that uses the proposed representation to find disjoint paths in an efficient manner.

Next, we consider the problem of finding two disjoint paths subject to the *export policy* limitations [5], imposed by customer-provider and peer relationships between routing domains. It turns out that the standard approach of representing a multi-domain network by a graph is inadequate for finding disjoint paths subject to the export policies. However, we show that the export policies can be efficiently represented

[1]Department of Electrical and Computer Engineering, Texas A&M University, College Station, Texas.

[2]Advanced Networks Architectures Lab, Technical University of Catalonia (UPC), Barcelona, Spain.

[3]Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa, Israel.

This work was partially funded by the European Commission through CONTENT under contract FP6-0384239 for UPC authors.

by employing the concept of the *line graph*. We show that our distributed algorithm can be easily extended for finding optimal disjoint paths that satisfy the export policy constraints.

In summary, we present an optimal solution for the multi-domain disjoint path problem both in the general setting as well as subject to the export policy constraints. For clarity of exposition, we focus on finding link-disjoint paths. Our results can be easily extended to finding node-disjoint paths by using the standard node splitting technique (see, e.g., [6]).

II. Multi-domain Network Model

We begin with a definition of a *general* communication network. A *network* is represented by a directed graph $G(V, E)$, where V is the set of nodes and E is the set of links. Each link $e \in E$ is assigned a positive *weight* w_e, whose significance depends on the type of considered QoS requirement. For example, when the QoS requirement is an upper bound on the end-to-end delay, the link weight is its delay. In this paper we focus on additive weight metrics, i.e., the weight $W(P)$ of a path P is defined as the sum of the weights of its links, i.e., $W(P) = \sum_{e \in P} w_e$.

The goal of QoS routing is to find the best path that satisfies a QoS constraint. In this work, we accomplish this goal by finding a minimum-weight path between the source and the destination nodes. Clearly, such path has the best performance with respect to the QoS requirement that is captured by the link weight metric.

A. Multi-domain networks

We denote by $D_1, \ldots, D_k$ the set of routing domains in the network. Each routing domain D_i is a subgraph of the underlying network G. We assume that routing domains are mutually node-disjoint. The routing domains that include the source and destination nodes, s and t are referred to as D^s and D^t, respectively. A link that connects two nodes in the same domain is referred to as an *intra-domain link*. All other links connect different domains and are referred to as *inter-domain* links. We denote by E^{inter} the set of the inter-domain links in the network. A node v which is incident to an inter-domain link is referred to as a *border node*. The set of border nodes of a routing domain D_i is denoted by B_i.

In large communication networks, distributing the full link state information to every node in the network is not possible due to scalability problems. With topology aggregation, sub-networks, or *routing domains*, can limit the amount of link state information advertised throughout the network [7]. Our approach is that routing domains supply a short summary of the available (disjoint) paths that connect the border nodes of the domain. The efficiency of this approach stems from the fact that while the routing domains tend to be large, the number of border nodes in each domain is typically small.

We denote by A_i the *Aggregated Representation* (*AR*) of the routing domain i. The AR captures the transitional characteristics of the network and can be implemented by a (small) graph or an array. In this paper we propose an AR that includes several arrays that summarize available routing paths between the border nodes of the routing domain.

In order to distribute the ARs of routing domains throughout the network we take advantage of the architecture recently drafted by the IETF PCE WG. Our PCE-based routing model utilizes a decoupled control plane for both the computation of the 2DP and the advertisement of routing information. This decoupling is two-fold. On the one hand, the PCEs are detached from the MPLS switch/routers forwarding the traffic. On the other hand, the aggregated topology, reachability, and path state information needed to compute the routing paths are decoupled from BGP and advertised directly between the PCEs [8]. This approach has two major advantages. First, it overcomes some of the most important limitations imposed by BGP [9]. For example, it allows to advertise multiple routes per destination prefix, and to convey path state information in the routing advertisements, which cannot be done at present with BGP-4. Overcoming these limitations is essential for the optimal computation of disjoint paths between multiple routing domains. Second, this approach can be incrementally deployed since it can coexist with the legacy IP IGP/BGP routed traffic.

The information available at the source PCE includes, the source domain D^s, a set of inter-domain links E^{inter}, and the ARs of the transit and destination domains. In practice, the routing across a multi-domain network is governed by the export policies. In particular, the export policies determine the inter-domain links that the source PCE can use while computing paths for any source-destination pair.

We consider the set of commonly used export policies as summarized in [5]. We assume that, for any two neighboring routing domains D_i and D_j, one of the three following cases hold: (i) D_i is a provider of D_j and D_j is a customer of D_i; (ii) D_j is a provider of D_i and D_i is a customer of D_j; (iii) D_i and D_j are peers.

The export policies impose the following constraints on the forwarding policy.

- Suppose that D_i is a customer of D_j. Then, D_i can forward packets received from D_j to its customers, but never to its peers or other providers.
- Suppose that D_i is a provider of D_j. Then, D_i can forward packets received from D_j to its customers, providers and peers.
- Suppose that D_i is a peer of D_j. Then, D_i can forward packets received from D_j to its customers but never to its providers or peers.

Let D_i, D_j, and D_k be three domains such that D_i is connected to D_j and D_j is connected to D_k. Table I summarizes the conditions under which D_j can forward the traffic received from D_i to D_k. As mentioned before, computing an optimal solution for Problem 2DP requires special care when the above export policies are considered.

	D_j is a customer of D_k	D_j is a provider of D_k	D_j is a peer of D_k
D_i is a customer of D_j	Yes	Yes	Yes
D_i is a provider of D_j	No	Yes	No
D_i is a peer of D_j	No	Yes	No

TABLE I
EXPORT POLICIES. THE TABLE SPECIFIES THE CONDITIONS UNDER WHICH D_j CAN FORWARD THE TRAFFIC RECEIVED FROM D_i TO D_k.

B. Problem definition

In this work we focus on finding two link-disjoint paths in a multi-domain network with topology aggregation. The first path, referred to as a *primary* path, is used during the normal operation of the network. Upon a failure of a link in the primary path, the traffic is shifted to a *backup* path. In order to satisfy the required QoS constraint, we need to minimize the weight of both primary and backup paths. Accordingly, we consider the problem of finding link-disjoint paths of minimum total weight.

Problem 2DP (2 Link Disjoint Paths): Given a source node s and a destination node t, find two link-disjoint (s,t)-paths P_1 and P_2 of minimum total weight $W(P_1)+W(P_2)$.

We can use the path with minimum weight as a primary path and the second one as a backup path. A relevant problem is to find two paths P_1 and P_2 that minimize max$\{W(P_1), W(P_2)\}$. The solution to this problem can achieve a better balance between the delay of the primary and backup path. However, this problem is $\mathcal{NP}$-hard [10].

Problem 2DP is a well studied problem. The standard algorithm used for solving this problem is due to Suurballe and Tarjan [11]. However, the existing algorithms were designed for the case in which the full topology is known to every node in the network. Accordingly, the goal of this study is to provide an efficient solution for the case in which only the aggregated representation of the network is known.

III. Related Work and Our Contributions

The problem of finding primary and backup paths subject to QoS constraints in the context of IP/MPLS networks has been widely studied at the intra-domain level. With the advent of the PCE-based architecture, a few recent works have started to extend the study of this problem to LSPs spanning multiple domains. In the current IGP/BGP routing context, a major issue is that the PCE in the source domain has to compute inter-domain LSPs based on a very limited visibility of the topology and state of the network, yielding solutions that are far from optimal. To cope with this, enriched topological and path state information needs to be aggregated and available at the PCE in the source domain [8].

In [4] the authors compare the performance of some recently proposed distributed schemes for disjoint path computation of inter-domain LSPs. They assume that the AS-level path was previously computed by BGP at the source domain and that both disjoint paths belong to the same "chain" of domains. This approach has two major limitations. First, solving problem 2DP restricted to the AS-path selected by BGP will frequently return paths that are far from optimal. This is because BGP does not offer any guarantee about the quality of the chosen AS-path. Second, when several disjoint LSPs need to be established following the same (or part of the same) AS-path, crankback [3] or even blocking might occur, even though the paths could have been established along the alternative AS-paths available at the source domain.

In this paper we study a PCE-based architecture that is completely decoupled from the BGP protocol. With this approach, the PCE at the source domain is not compelled to choose both paths along the same chain of domains. This allows the domains to use their multihomed networks more efficiently. Once we extend the computation of the paths to an expanded AS topology, i.e., not restricting our study to a chain of domains, we need to consider the export policies between domains. This, however, introduces a major challenge. Whereas the chain of domains can be aggregated and represented as a directed graph, this cannot be done in the presence of the export policies. To solve this problem we introduce an AR of the expanded topology using line graphs.

In [12] the authors propose two heuristics so that the PCEs can solve the problem of finding inter-domain LSPs with low end-to-end delay. However, this work addresses the computation of only a single path (without a disjoint counterpart). In addition, the availability of inter-domain paths is inferred directly from the BGP routing information. Accordingly, the authors do not need to address the issue of finding an AR that captures path diversity and the internal structure of the domains.

Several topology aggregation techniques have been proposed in the literature (see [7] and references therein).

Overall, to the best of our knowledge, this is the first work that optimally solves Problem 2DP in an expanded multi-domain IP/MPLS environment, subject to the common export policies. Our contributions can be summarized as follows:

- We propose an accurate AR that captures the path diversity and the internal link state of each domain.
- We introduce a distributed routing algorithm that exploits an AR of the multi-domain network in order to find an optimal pair of link-disjoint paths between the source and the destination in an efficient manner.
- We provide an efficient method for finding link-disjoint paths subject to the common export policies imposed by customer-provider and peer relationships between routing domains.

IV. General link-disjoint paths algorithm

In this section we describe a distributed algorithm for finding two link-disjoint paths in a multi-domain network with topology aggregation.

The distributed algorithm for path computation consists of the three following steps. In the first step, each routing domain D_i computes its AR A_i. This computation is performed by the PCE of the domain. In the second step, the AR A_i of each domain D_i is distributed throughout the network. In the third step, the PCE in the source domain uses the assembled representation of the network for computing two disjoint paths between the source and the destination nodes.

The rest of this section is structured as follows. In section IV-A we present our AR. In Section IV-B1 we describe an algorithm for computing the AR of a domain. Then, in Section IV-B2 we describe an algorithm for computing disjoint paths at the source PCE. Finally, in Section IV-B3 we describe an algorithm for establishing two disjoint (s,t)-paths throughout the network.

A. Aggregated representation

We begin by the description of the AR. The purpose of the AR is to summarize the traversal properties of each routing

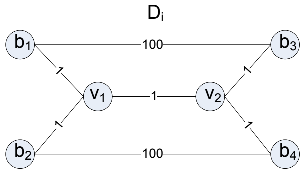

Fig. 1. An example of a routing domain. The domain has four border nodes $b_1, \ldots, b_4$. The numbers show the weights of the edges.

domain in a way that allows the source PCE to select two disjoint paths of minimum weight.

1) Aggregation scheme for minimum distances: The problem of finding a suitable AR that enables efficient computation of the minimum weight paths across the network is well studied in the literature. The natural representation of a routing domain D_i is an array that stores, for each pair of border nodes b_j and b_l of D_i, the minimum weight of a path between b_j and b_l. This representation allows the source node to find optimal paths and has the space complexity of $\Theta(|B_i|^2)$.

This representation, however, cannot be used for finding two disjoint paths across the network. To illustrate this point, consider the routing domain depicted in Fig. 1. In this domain, the minimum weight of the path between b_1 and b_3 and between b_2 and b_4 is equal to 3. However, the minimum weight of two disjoint paths, one between b_1 and b_3 and the second between b_2 and b_4 is equal to 103. This shows that additional information regarding the disjoint paths that run through the domain must be included in the aggregated representation.

2) Aggregation scheme based on the minimum weight of disjoint paths: A possible solution would be to keep, for each routing domain D_i and for each two pairs (b_j, b_l) and (b_x, b_y) of D_i, the minimum weight of two link-disjoint paths that connect b_j and b_x to b_l and b_y. In addition, we need to keep, for each routing domain D_i and for every pair (b_j, b_l) of border nodes of D_i, the minimum weight of a path between b_j and b_l. This method provides complete information about the traversal characteristics of the routing domain, under the assumption that each path enters the routing domain at most once. The main drawback of this approach is that the aggregated information does not allow the source PCE to find two disjoint paths between s and t in an efficient way. Indeed, the most effective method for finding two disjoint (s,t)-paths includes two steps, the first step finds a shortest path (s,t)-path P' and the second step finds an *augmenting* (s,t)-path P'' of P'. The augmenting path P'' may use links of P' in the reverse direction, which allows to avoid the trap topology problems [4]. This method is employed by the standard disjoint path algorithm due to Suurballe and Tarjan [11], described in detail in the next section. However, the aggregation scheme based on the minimum weight of disjoint paths inside a domain does not allow to compute the "augmenting" inter-domain path in an efficient way. In what follows, we present an alternative aggregated representation that addresses this problem.

3) Aggregation scheme based on the disjoint paths algorithm: Let $D_i(V_i, E_i)$ be a routing domain and let B_i be the set of border nodes on V_i. In this section we present the AR A_i of D_i. The main goal in the design of the AR is to allow the source PCE to find the minimum weight of disjoint paths

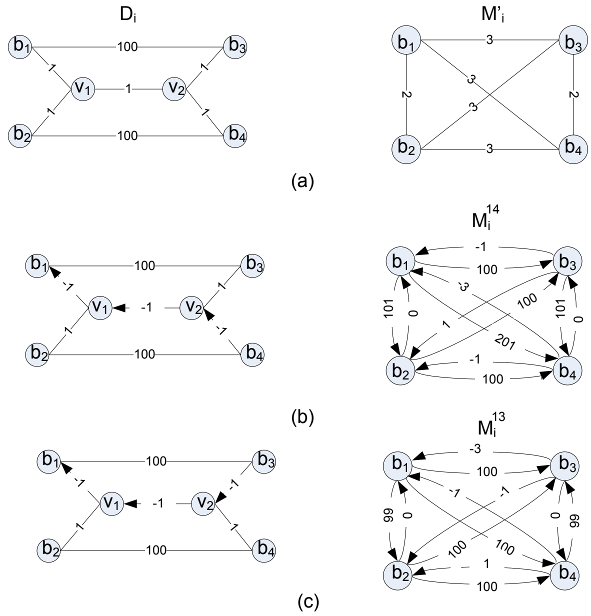

Fig. 2. The aggregated representation of a routing domain. (a) Array M'_i (b) The auxiliary graph $M_i^{1,4}$ (left) and array $M_i^{1,4}$ (right). (c) The auxiliary graph $M_i^{1,3}$ (left) and array $M_i^{1,3}$ (right).

in an efficient way. We begin by presenting the disjoint path algorithm due to Suurballe and Tarjan [11]. The algorithm receives as an input a graph $G(V,E)$, a source node s, and the destination node t. The algorithm performs the following steps:

1) Find a shortest path P' between s and t in G;
2) Reverse all links in P' and negate their weight;
3) Find an augmenting shortest path P'' in the resulting graph $\hat{G}$;
4) Remove links that appear in P' and P'' in opposite directions;
5) From the remaining links of P' and P'', form two disjoint (s,t)-paths $\hat{P}_1$ and $\hat{P}_2$.

The idea of our scheme is to allow the source PCE to compute two disjoint paths in the aggregated environment in a similar way as if the entire network topology were known. To that end, A_i includes two components. The first component allows the source PCE to find a shortest path P_1 between s and t, while the second component allows the source PCE to find the second path P_2. The paths P_1 and P_2 correspond to the paths P' and P'', used by the algorithm due to [11].

In particular, the first component of A_i includes array M'_i that contains, for each two border nodes b_j and b_l of D_i, the minimum weight of a path between b_j and b_l. The second component of A_i includes a set of $|B_i|(|B_i|-1)$ arrays $\{M_i^{j,l} \;|b_j \in B_i, b_l \in B_i, b_j \neq b_l\}$, each array containing $|B_i|(|B_i|-1)$ elements. In particular, array $M_i^{j,l}$ contains, for any two border nodes b_x and b_y of D_i, the minimum weight of a path between b_x and b_y in $D_i^{j,l}$, where $D_i^{j,l}$ is a graph formed from D_i by inverting links that belong to a minimum weight path between b_j and b_l and negating their weights.

Fig. 2 graphically presents the aggregated representation of the routing domain shown in Fig. 1. The representation we

Algorithm FINDAR *(D_i, B_i):*

Input:

D_i - a routing domain,

B_i - the set of border nodes of D_i.

Output:

$A_i = \{M'_i\} \cup \{M_i^{j,l} \, | b_j \in B_i, b_l \in B_i\}$ of D_i

```
for each two border nodes b_j and b_l of D_i do
    Compute a shortest path P_i^{j,l} between b_j and b_l in D_i;
    M'_i(j,l) ← W(P_i^{j,l})
    Construct an auxiliary graph D_i^{j,l} formed from D_i by
        reversing all links of P_i^{j,l} and negating their weights
    for each two border nodes b_x and b_y of D_i do
        Compute a shortest path P_i^{j,l}(x,y) between b_x and
            b_y in D_i^{j,l}
        M_i^{j,l}(x,y) ← W(P_i^{j,l}(x,y)).
```

Fig. 3. Algorithm FINDAR

present is based on the following assumption.

Assumption 1: A minimum weight path between a source node s and the destination node t traverses each routing domain D_i at most once.

This fact significantly simplifies the construction of an aggregated representation. Our methods can be extended to deal with settings in which this assumption does not hold.

B. Disjoint path algorithm

1) First step - Computing the Aggregated Representation: The AR A_i can be efficiently computed through Algorithm FINDAR that appears in Fig. 3. The algorithm computes, for each pair of border nodes b_j, b_l of D_i, a shortest path $P_i^{j,l}$ between b_j and b_l in D_i and stores the result in array M'_i. Then, the algorithm reverses all links of $P_i^{j,l}$, negates their weights, and computes a minimum weight path between any pair of border nodes in the resulting graph. The minimum weights of these paths are stored in the array $M_i^{j,l}$. Since the resulting graph may contain negative weights, we use a modification of the Dijkstra's algorithm due to Bhandari [13]. The computational complexity of the modified algorithm is identical to that of the original Dijkstra's algorithm.

Finding a shortest path between any pair of border nodes requires $|B_i|$ invocations of the shortest path algorithm. Thus, computing the AR A_i of D_i requires $O(|B_i|^3)$ invocations of the shortest path algorithm. Therefore, the computational complexity of computing the AR is $O(|B_i|^3(|V_i| \log |V_i| + |E_i|))$. The size of the aggregated representation is $O(|B_i|^4)$.

To derive a practical estimation of the size of this AR, let us compare this latter against the number of active entries in the BGP Forwarding Information Base (FIB) of the border routers in a Tier-1 ISP. At present, these border routers have around $2\text{x}10^5$ active entries in their BGP FIB [14], and this scale does not represent an issue for the routers. In our case, an AR of 21 border routers on average per-domain (i.e. approximately $[2\text{x}10^5]^{1/4}$) represents the same load as operational routers have nowadays in a Tier-1 ISP. It is worth recalling that our proposals apply to a reduced set of neighboring domains, and that they can be incrementally deployed. In this scenario, an average of 21 IP/MPLS enabled border routers per-domain offers significant flexibility from a practical viewpoint.

2) Second step - Computing the minimum weight of shortest paths: We assume that the source PCE has a detailed topology of the source routing domain, and in addition, the ARs of the transit and destination routing domains. The source PCE uses this information in order to construct a high-level description of two disjoint paths that connect s and t.

We note that while the AR A_i of a transit domain D_i captures the path diversity and link-state information of D_i, the AR of D^t captures the same properties, but for the paths between the border nodes of D^t and the destination t. Given that the AR of D^t follows the same principle as that of any transit domain, without loss of generality, in our model the destination t is considered as a border node of the routing domain D^t. This is motivated by the fact, that in order to find an optimal pair of link-disjoint paths, the source PCE needs some information about the paths between the border nodes of D^t and the destination t.

The operations performed by the source PCE are summarized by Algorithm FIND2DP that appears in Fig. 4. Algorithm FIND2DP begins by constructing an auxiliary graph $G'(V', E')$ that includes, for each domain D_i of G, the complete graph spanned by the border nodes of D_i. In addition, G' includes the source domain D^s and the set of inter-domain links E^{inter}. The purpose of the auxiliary graph is to summarize the network information available at the source PCE.

Next, the source PCE computes the shortest path P_1 between s and t. This is accomplished by assigning for each link (b_j, b_l) that connects two border nodes of the same domain D_i, the minimum weight of a path between b_j and b_l and finding a shortest path between s and t in the resulting graph. The minimum weights of the shortest paths that run through domain D_i are available through array M'_i.

Finally, the source PCE computes the second (s,t)-path P_2. To that end, for each domain D_i traversed by P_1 (i.e., P_1 contains a link that connects border nodes of D_i) we perform the following operations. Let (b_j, b_l) be a link in P_1 that connects border nodes of D_i and let $\hat{D}_i$ be the complete graph spanned by the border nodes of D_i. Then, we set the weights of the links of the subgraph $\hat{D}_i$ of G' according to array $M_i^{j,l}$. The path P_2 is found by applying the shortest path algorithm on the resulting graph.

The computational complexity of Algorithm FIND2DP is $O(|V'| \log |V'| + |E'|)$, where V' and E' is the set of nodes of the auxiliary graph $G'(V', E')$. Again, since the auxiliary graph contains negative weights, we use the algorithm due to Bhandari [13] for finding shortest paths in G'. The set V' includes all nodes in the source routing domain and the border nodes of all transit domains and the destination domain. The set E' includes all links in the source domain, the set of inter-domain links and, in addition, a link between any two border nodes of the same domain.

3) Third step - establishing QoS paths: In the third step, the source PCE sends the paths P_1 and P_2 to every routing domain D_i traversed by these paths. At each domain, the PCE is responsible for establishing the portions of the disjoint paths

Algorithm FIND2DP *(E^{inter}, $\{A_i\}$):*

Input:

E^{inter} - a set of the inter-domain links,

For each routing domain D_i

$A_i = \{M'_i\} \cup \{M_i^{j,l} \mid b_j \in B_i, b_l \in B_i\}$ - The aggregated representation of D_i.

Output:

An auxiliary network $G'(V', E')$ and two paths P_1 and P_2 in G'.

$V' \leftarrow V(D^s) \cup \{B_i \mid D_i \in G\}$

$E' \leftarrow E(D^s) \cup E^{inter}$

for each routing domain D_i of G **do**

  for each two border nodes b_j and b_l of D_i **do**

    $E' \leftarrow E' \cup (b_j, b_l)$;

    $w_{(b_j,b_l)} \leftarrow M'_i(j, l)$

Find a shortest path P_1 between s and t in $G'(V', E')$

Reverse all inter-domain links and links that belong to D^s in P_1 and negate their weight

For each routing domain D_i of G except D^s**do**

  if P_1 contains a link (b_j, b_l) that connects border nodes of D_i **then**

    for each two border nodes b_x and b_y of D_i **do**

      $w_{(b_x,b_y)} \leftarrow M_i^{j,l}(x, y)$

Find a shortest path P_2 between s and t in $G'(V', E')$.

Fig. 4. Algorithm FIND2DP

that run through these domains. We consider the following cases.

1) Domain D_i is traversed by path P_1 and is not traversed by P_2. In this case, let (b_j, b_l) be the link in P_1 that connects the border nodes of D_i. Then, link (b_j, b_l) is substituted by the path $P_i^{j,l}$ computed at Line 2 of Algorithm FINDAR.
2) Domain D_i is traversed by path P_2 and is not traversed by P_1. In this case, each link $(b_j, b_l) \in P_2$ that connects the border nodes of D_i is substituted by the path $P_i^{j,l}$ computed at Line 2 of Algorithm FINDAR.
3) Domain D_i is traversed by both paths P_1 and P_2. Let (b_j, b_l) be the link in P_1 that connects the border nodes of D_i. Then, we perform the following operations. First, link (b_j, b_l) is substituted by the path $P_i^{j,l}$ computed at Line 2 of Algorithm FINDAR. Second, each link $(b_x, b_y) \in P_2$ that connects the border nodes of D_i is substituted by the path $P_i^{j,l}(x, y)$ computed at Line 6 of Algorithm FINDAR. Finally, all links of D_i that appear in $P_i^{j,l}$ and $P_i^{j,l}(x, y)$ in opposite directions are omitted from both $P_i^{j,l}$ and $P_i^{j,l}(x, y)$.

C. Illustrative Example

Fig. 5 presents an illustrative example of our algorithm. The underlying communication network, depicted in Fig. 5(a), contains source domain D^s and two transit domains, D_1 and D_2. Fig. 5(b) depicts the auxiliary network G' constructed by Algorithm FIND2DP with link weights assigned according to the values of arrays M'_1 and M'_2. This auxiliary network is used by the source PCE to compute the shortest path between the source and the destination nodes. The shortest path is marked in Fig. 5(b) by the bold lines and includes border

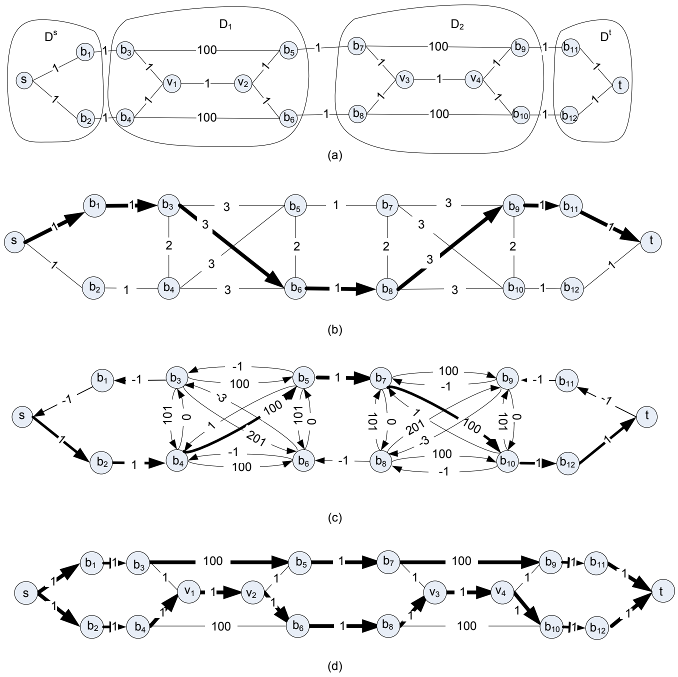

Fig. 5. An illustrative example: (a) The underlying communication network with two transit domains D_1 and D_2. (b) The auxiliary network $G'(V', E')$ with weights assigned according to arrays M'_1 and M'_2. (Two directed links with identical weights are represented by a single undirected link) (c) The auxiliary network $G'(V', E')$ with weights assigned according to arrays $M_1^{3,6}$ and $M_2^{8,9}$. (d) Two disjoint paths in the underlying network.

nodes b_3 and b_6 of routing domain D_1 and nodes b_8 and b_9 of routing domain D_2. The weight of the P_1 is 11. Next, the source PCE turns to compute path P_2. To that end, the same communication network is used, but the weights are assigned according to arrays $M_1^{3,6}$ (for D_1) and $M_2^{8,9}$ (for D_2, see Fig. 5(c)). The shortest path in this network is marked by the bold lines and includes nodes b_4 and b_5 of routing domain D_1 and nodes b_7 and b_{10} of routing domain D_2. The weight of path P_2 is 205. Finally, the source PCE sends the two disjoint paths P_1 and P_2 to the PCEs of the routing domains D_1 and D_2. The PCE of the routing domain D_1 substitutes the two links (b_3, b_6) and (b_4, b_5) of D_1 by two disjoint paths $\{b_3, b_5\}$ and $\{b_4, v_1, v_2, b_6\}$, while the two links (b_8, b_9) and (b_7, b_{10}) of D_2 are substituted by two paths $\{b_7, b_9\}$ and $\{b_8, v_3, v_4, b_{10}\}$. The two disjoint paths in the original network are depicted in Fig. 5(d).

D. Correctness proof

In this section we prove that the presented algorithm finds two optimal paths between s and t.

Theorem 1: If Assumption 1 holds, then the proposed algorithm finds two disjoint paths between s and t of minimal total weight.

Proof: Suppose that the full topology of the communication network G is known. In this case, we can apply the algorithm due to [11] (as described in Section IV-A3) to find two disjoint paths of minimal weight. Let P' and P'' be the paths identified in Lines 1 and 3 of this algorithm, respectively. The correctness of the algorithm implies that $W(P')+W(P'')$

is equal to the minimum weigh of a shortest path between s and t.

Next, we show that for paths P_1 and P_2, identified by the Algorithm FIND2DP, it holds that $W(P_1) \leq W(P')$ and $W(P_2) \leq W(P'')$. This is sufficient to prove the correctness of the algorithm. Indeed, in Step 3 (presented in Section IV-B3) of the algorithm we use P_1 and P_2 to establish two link-disjoint (s,t)-paths that will be expanded by the traversed domains. It is easy to verify that the total weight of the resulting paths is equal to the total weight of P_1 and P_2.

We proceed to show that $W(P_1) \leq W(P')$. We note that path P' can be divided into subpaths $P'_1, \ldots, P'_h$ such that P'_1 connects s to a border node of D^s, P'_h connects a border node of D^t to t and for $2 \leq i \leq h-1$, P'_i either includes an inter-domain link or a link that connects two border nodes of a routing domain. We also note that the auxiliary network $G'(V', E')$ includes all links of the subpath P'_1 and also all subpaths P'_i that include inter-domain links. Further, all links of these subpaths have the same weight in G' as in the original network. For each subpath P'_i of P' that connects border nodes b_j and b_l of a routing domain D_x, the auxiliary network $G'(V', E')$ contains a link whose weight is less than or equal to $W(P'_i)$. Since P_1 is a minimum weight path in $G'(V', E')$, it follows that $W(P_1) \leq W(P')$.

Finally, we show that $W(P_2) \leq W(P'')$. Let $\hat{G}$ be the resulting graph after executing Line 2 of the algorithm due to [11] (as presented in Section IV-B3). We note that path P'' can be divided into subpaths $P''_1, \ldots, P''_h$ such that P''_1 connects s to a border node of D^s, P''_h connects a border node of D^t to t and for $2 \leq i \leq h-1$, P'_i either includes an inter-domain link or connects two border nodes of the routing domain. We also note that the auxiliary network $G'(V', E')$ includes all links of the subpaths that traverse the source routing domains and all subpaths P''_i that include inter-domain links and these links have the same weight as in $\hat{G}$. For each subpath P''_i of P'' that connects border nodes b_j and b_l of a routing domain D_x, the auxiliary network $G'(V', E')$ contains a link whose weight is less than or equal to $W(P''_i)$. Since P_2 is a minimum weight path in $G'(V', E')$, it follows that $W(P_2) \leq W(P'')$. ■

V. EXPORT POLICIES

In this section we discuss the problem of finding two disjoint paths in the network in the presence of export policies. The main challenge posed by the export policies is that the availability of the link for a particular connection depends on the previous hop. As a result, the standard representation of the network in the form of a graph is no longer adequate for routing purposes. For example, consider the multi-domain network depicted on Fig. 6(a). In this network, D_1 is a customer of both D_2 and D_3; D_7 is a customer of D_5 and D_6; D_4 is a provider of D_3 and D_6; D_2 is a peer of D_4, and D_4 is a peer of D_5. The export policies specified in Table I allow the following paths between D_1 and D_7: (a) $D_1 \to D_2 \to D_4 \to D_6 \to D_7$; (b) $D_1 \to D_3 \to D_4 \to D_5 \to D_7$; (c) $D_1 \to D_3 \to D_4 \to D_6 \to D_7$. Note the every link of the network is included in one of these paths. Thus, pruning a link from the network will result in omitting one of the feasible paths from the network. However, the path $D_1 \to D_2 \to D_4 \to D_5 \to D_7$ that belongs to the network is not allowed by the export policies. We conclude that the graph that depicts the connectivity among multiple domains, depicted in Fig. 6(a), is not adequate for computing optimal paths subject to the export policies.

A. Line graphs

In order to efficiently find paths subject to export policies, we use the notion of the *line graph*. The rationale for this is that line graphs are able to capture the transit properties between the ingress and egress links of domains.

Definition 1 (Line Graph): Let $G(V, E)$ be a communication network, $D_1, \ldots, D_k$ be the set of routing domains in G, where $D^s = D_1$ is the source domain and $D^t = D_k$ be the destination domain, and E^{inter} be the set of inter-domain links. Then, the line graph $\hat{G}(\hat{V}, \hat{E})$ of G is a graph constructed as follows:

1) For each inter-domain link $e_i \in E^{inter}$ in G add a corresponding node $\hat{v}_i$ to $\hat{V}$.
2) For each routing domain $D_i, 1 \leq i \leq k$, and for each two inter-domain links e_j and e_l incident to D_i in G:
   – Add a link $(\hat{v}_j, \hat{v}_l)$, where $\hat{v}_j$ and $\hat{v}_l$ are nodes in $\hat{G}$ that correspond to e_j and e_l, respectively.
3) Add special nodes $\hat{s}$ and $\hat{t}$.
4) For each inter-domain link e_i incident to the source routing domain D^s add a link between $\hat{s}$ and the node $\hat{v}_i$ in $\hat{G}$ that corresponds to e_i.
5) For each inter-domain link e_i incident to the destination routing domain D^t add a link between the node $\hat{v}_i$ in $\hat{G}$ that corresponds to e_i and $\hat{t}$.

Fig. 6(b) depicts the line graph of the multi-domain network that appears in Fig. 6(a). In this figure, the node corresponding to a link between routing domains D_i and D_j in G is denoted by $\hat{v}_{i,j}$.

Let P be an (s,t)-path in G and let $D^s = D_1, D_2, \ldots, D_h = D^t$ be the set of routing domains traversed by P. We say that path $\hat{P} = \{\hat{s}, \hat{v}_{1,2}, \hat{v}_{2,3}, \ldots, \hat{v}_{h-1,h}, \hat{t}\}$ in $\hat{G}$ *corresponds* to P. The following proposition follows from the construction of the line graph $\hat{G}$.

Proposition 2: Let P_1 and P_2 be two link-disjoint paths in G. Then, the two corresponding paths $\hat{P}_1$ and $\hat{P}_2$ in $\hat{G}$ are node-disjoint.

For example, suppose that P_1 traverses routing domains D^s, D_3, D_4, D_5, D^t and P_2 traverses routing domains D^s, D_2, D_4, D_6, D^t. Then, the two corresponding paths in $\hat{G}$, $\hat{P}_1 = \{\hat{s}, \hat{v}_{1,3}, \hat{v}_{3,4}, \hat{v}_{4,5}, \hat{v}_{5,7}, \hat{t}\}$ and $\hat{P}_2 = \{\hat{s}, \hat{v}_{1,2}, \hat{v}_{2,4}, \hat{v}_{4,6}, \hat{v}_{6,7}, \hat{t}\}$ are node-disjoint.

B. Modified line graphs $\hat{G}_1$ and $\hat{G}_2$

As mentioned above, the export policies prohibit certain paths between routing domains. In order to take into account these policies, we introduce several modifications to the line graph. First, we replace each undirected link that connects nodes in $\hat{G}$ by two directed links in opposite directions. Then, for each directed link in the resulting graph we check whether it can be used under the export policies specified in Table I, and if not, the link is removed from the graph. We denote

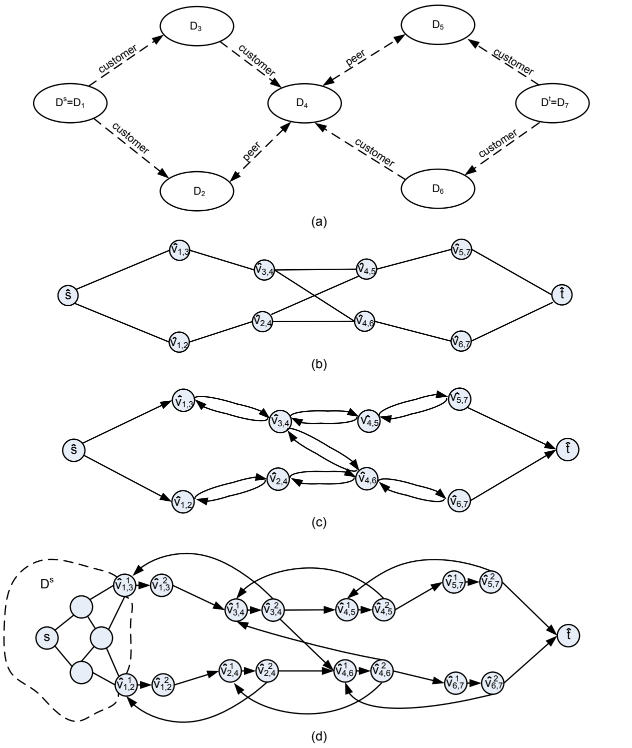

Fig. 6. (a) Multi-domain network. The lines represent the connectivity between domains. For example, D_1 is connected to D_3, and D_4 is connected to D_5. The directions of the links show the relationship between the domains. For example, D_1 is a customer of D_3, while D_4 is a peer of D_5. (b) The corresponding line graph with two special vertices $\hat{s}$ and $\hat{t}$. (c) Graph $\hat{G}_1$. (d) Graph $\hat{G}_2$.

the modified line graph by $\hat{G}_1(\hat{V}_1, \hat{E}_1)$. Fig. 6(c) depicts the modified line graph $\hat{G}_1$ that corresponds to the network depicted in Fig. 6(a). Note that link $(\hat{v}_{2,4}, \hat{v}_{4,5})$ is omitted from $\hat{G}_1$ because D_4 is a peer domain of both D_2 and D_5, hence it cannot forward packets from D_2 to D_5.

We summarize the properties of the line graph $\hat{G}_1$ in the following proposition.

Proposition 3: Let P be a (s,t)-path in G and let $\hat{P}$ be a corresponding $(\hat{s},\hat{t})$-path in $\hat{G}$. Then, if P can be used under the export policies listed in Table I, the path $\hat{P}$ also belongs to $\hat{G}_1$. Further, for each $(\hat{s},\hat{t})$-path $\hat{P}$ in $\hat{G}_1$, there exists a corresponding path P in G that satisfies the export policies.

In this section, we adapt the algorithm presented in Section IV for finding link-disjoint paths that satisfy the export policies. To that end, we take advantage of the modified line graph $\hat{G}_1$ described before.

As discussed above, for any two link disjoint paths P_1 and P_2 in G that satisfy the export policies, there exist two corresponding paths $\hat{P}_1$ and $\hat{P}_2$ in $\hat{G}_1$. Moreover, such paths are node-disjoint. Since the algorithm presented in Section IV finds two link-disjoint paths, we need to introduce the following modifications to $\hat{G}_1$ in order to take advantage of this algorithm.

1) We split each node $\hat{v}_{i,j}$ in $\hat{G}$ into two nodes $\hat{v}^1_{i,j}$ and $\hat{v}^2_{i,j}$, connected by a link $(\hat{v}^1_{i,j}, \hat{v}^2_{i,j})$, such that all links into (out of) $\hat{v}_{i,j}$ are now into $\hat{v}^1_{i,j}$ (out of $\hat{v}^2_{i,j}$). The weight of $(\hat{v}^1_{i,j}, \hat{v}^2_{i,j})$ is set to be the weight of the corresponding inter-domain link between D_i and D_j in the original network G.
2) We replace the special node $\hat{s}$ and all links $(\hat{s}, \hat{v}^1_{1,i})$ incident to $\hat{s}$ by the source routing domain D^s, such that each node $\hat{v}^1_{1,i}$ incident to $\hat{s}$ in $\hat{G}_1$ coincides with the corresponding border node of D^s (i.e., the border node of D^s incident to the inter-domain link that corresponds to $(\hat{v}^1_{1,i}, \hat{v}^2_{1,i})$).

The resulting graph is denoted by $\hat{G}_2$. Fig. 6(d) depicts graph $\hat{G}_2$ that corresponds to the multi-domain network depicted in Fig. 6(a). It is easy to verify that for any two link-disjoint paths P_1 and P_2 in G the corresponding paths $\hat{P}_1$ and $\hat{P}_2$ in $\hat{G}_2$ are also link-disjoint.

The links of $\hat{G}_2$ can be classified into three groups. The first group includes links $(\hat{v}^1_{i,j}, \hat{v}^2_{i,j})$ represent the inter-domain links in the original network G. The second category include links $(\hat{v}^2_{i,j}, \hat{v}^1_{j,l})$ that represent paths through transit routing domains. Such links are referred to as *cross-domain* links. Finally, the third group includes the links that belong to the source and destination routing domains. For example, in Fig. 6(d), links $(\hat{v}^1_{1,3}, \hat{v}^2_{1,3})$ and $(\hat{v}^1_{4,6}, \hat{v}^2_{4,6})$ correspond the the inter-domain links in G that connect domains D_1 to D_3 and D_4 to D_6, respectively. In addition, the links $(\hat{v}^2_{1,3}, \hat{v}^1_{3,4})$ and $(\hat{v}^2_{2,4}, \hat{v}^1_{4,6})$ in $\hat{G}_2$ are cross-domain links that represent paths through domains D_3 and D_4, respectively.

C. Disjoint path algorithm

The disjoint paths algorithm in the presence of export policies is an extension of the distributed algorithm presented in Section IV. In particular, the first step (Algorithm FINDAR) remains the same and only minor and straightforward modifications are needed for the third step of the algorithm. For the second step, we use Algorithm FIND2DP-EP, presented in Fig. 7 that performs operations on the modified line graph $\hat{G}_2$. Thus, the line graph $\hat{G}_2$ should be constructed prior to the application of the algorithm.

The algorithm uses the following definitions. For each cross-domain link $(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ we denote by $e'(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ and $e''(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ the inter-domain links in G that correspond to nodes $\hat{v}_{i,j}$ and $\hat{v}_{j,k}$ in the line graph, respectively. We also denote by $x(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ and $y(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ the border nodes of D_j incident to links $e'(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ and $e''(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$, respectively.

The algorithm begins by assigning to each cross-domain link $(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ of $\hat{G}_2$ the weight equal to the minimum weight of a path between the border nodes $x(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ and $y(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$ of the routing domain D_j that corresponds to $(\hat{v}^2_{i,j}, \hat{v}^1_{j,k})$. The minimum weights of the shortest paths are available through array M'_j. Next, we compute a minimum weight $(s,\hat{t})$ path P_1 in $\hat{G}_2$, which corresponds to the minimum weight of a $(s,\hat{t})$-path in G that satisfies the export policies.

Next, for each cross-domain link $(\hat{v}^2_{ij}, \hat{v}^1_{jk}) \in P_1$ we perform the following operations. Let D_j be the routing domain that corresponds to $(\hat{v}^2_{ij}, \hat{v}^1_{jk})$. Note that due to Assumption 1 path P_1 traverses each domain at most once. Then, we assign the weight of each cross-domain link $(\hat{v}^2_{zj}, \hat{v}^1_{jw})$ that corresponds

Algorithm FIND2DP-EP *($\hat{G}_2$, $\{A_i\}$):*

Input:

$\hat{G}_2$ - the modified line graph of G;
For each routing domain D_i
$A_i = \{M'_i\} \cup \{M_i^{j,l} \mid b_j \in B_i, b_l \in B_i\}$ -
The aggregated representation of D_i.

Output:

Two paths P_1 and P_2 in $\hat{G}_2$.

for each cross-domain link $(\hat{v}^2_{ij}, \hat{v}^1_{jk})$ of $\hat{G}_2$ **do**
  $w_{(\hat{v}^2_{ij}, \hat{v}^1_{jk})} \leftarrow M'_j(x(\hat{v}^2_{ij}, \hat{v}^1_{jk}), y(\hat{v}^2_{ij}, \hat{v}^1_{jk}))$
Find a shortest path P_1 between s and $\hat{t}$ in $\hat{G}_2$
for each cross-domain link $(\hat{v}^2_{ij}, \hat{v}^1_{jk}) \in P_1$ **do**
  $x_1 \leftarrow x(\hat{v}^2_{ij}, \hat{v}^1_{jk})$
  $y_1 \leftarrow y(\hat{v}^2_{ij}, \hat{v}^1_{jk})$
  for each cross-domain link $(\hat{v}^2_{zj}, \hat{v}^1_{jw})$ that corresponds to domain D_j **do**
    $x_2 \leftarrow x(\hat{v}^2_{zj}, \hat{v}^1_{jw})$
    $y_2 \leftarrow y(\hat{v}^2_{zj}, \hat{v}^1_{jw})$
    $w_{(\hat{v}^2_{zj}, \hat{v}^1_{wk})} \leftarrow M_j^{x_1,y_1}(x_2, y_2)$
Find a shortest path P_2 between s and $\hat{t}$ in $\hat{G}_2$

Fig. 7. Algorithm FIND2DP-EP

to D_j (including $(\hat{v}^2_{ij}, \hat{v}^1_{jk})$) according to array $M_j^{x_1,y_1}(x_2, y_2)$, where $x_1 = x(\hat{v}^2_{ij}, \hat{v}^1_{jk})$, $y_1 = y(\hat{v}^2_{ij}, \hat{v}^1_{jk})$, $x_2 = x(\hat{v}^2_{zj}, \hat{v}^1_{jw})$, and $y_2 = y(\hat{v}^2_{zj}, \hat{v}^1_{jw})$. This operation correspond to lines 11 and 12 of Algorithm FIND2DP.

Finally, the source node sends the paths P_1 and P_2 to every routing domain D_i traversed by these paths. At each domain, the PCE expands every cross-domain link of P_1 and P_2 into an intra-domain path by using the methods described in Section IV-B3.

The following theorem summarizes the correctness of the algorithm.

Theorem 4: The proposed algorithm finds two disjoint paths between s and t that satisfy the export policies at minimum possible weight.

Proof: The proof follows the same lines as that of Theorem 1. Due to space constraints, the details are omitted. ■

The computational complexity of the algorithm presented in this section is similar to that of the algorithm presented in Section IV.

VI. Conclusions and Future Work

This paper presents a distributed routing algorithm for finding two disjoint (primary and backup) QoS paths across multiple IP/MPLS domains. Our algorithm can be employed by a PCE-based architecture and work completely decoupled from the BGP protocol. We have developed an aggregated representation of a multi-domain network that captures the path diversity and the link-state characteristics of transit paths that run across different routing domains. This representation is used by the distributed routing algorithm, allowing each PCE to efficiently compute two disjoint QoS paths (2DP) for any source-destination pair.

We present an optimal solution for the multi-domain disjoint path problem both for the general setting, as well as subject to the *export policies* imposed by customer-provider and peer relationships between routing domains. Our approach can be easily extended to accommodate changes in the existing export policies. Our algorithms can be used in many practical settings, in particular, when high-quality primary and backup QoS LSPs need to be established across a reduced set of neighboring domains.

For future work, we plan to investigate additional approaches that aim at balancing the intrinsic tradeoff between the scalability of the aggregated representation of a multi-domain network and the optimality of the resulting LSPs. We have also plans to address the load balancing and traffic engineering issues related to establishing disjoint QoS paths in a multi-domain environment.

References

[1] http://www.ietf.org/html.charters/pce-charter.html.

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

[3] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash. Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE (draft-ietf-ccamp-crankback-05.txt). Internet Draft, work in progress, May 2005.

[4] F. Ricciato, U. Monaco, and D. Ali. Distributed Schemes for Diverse Path Computation in Multi-domain MPLS Networks. *IEEE Communications Magazine, Feature Topic on Challenges in Enabling Inter-Provider Service Quality on the Internet*, 43(6):138–146, June 2005.

[5] Y. R. Yang, H. Xie, H. Wang, A. Silberschatz, A. Krishnamurthy, and L. E. Li. On Route Selection for Interdomain Traffic Engineering. *IEEE Network*, 19(6):20–27, November-December 2005.

[6] J. Suurballe. Disjoint Paths in a Network. *Networks*, 4:125–145, 1974.

[7] S. Uludag, K.S. Lui, K. Nahrstedt, and G. Brewster. Comparative Analysis of Topology Aggregation Techniques and Approaches for the Scalability of QoS Routing. In *Technical Report TR05-010, DePaul University*, Chicago, USA, May 2005.

[8] M. Yannuzzi, X. Masip-Bruin, S. Sanchez-Lopez, J. Domingo-Pascual, A. Orda, and A. Sprintson. On the Challenges of Establishing Disjoint QoS IP/MPLS Paths across Multiple Domains. *IEEE Communications Magazine*, 44(12):60–66, December 2006.

[9] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure. Open Issues in Interdomain Routing: A Survey. *IEEE Network*, 19(6):49–56, November-December 2005.

[10] C. L. Li, T. McCormick, and D. Simchi-Levi. The Complexity of Finding Two Disjoint Paths with Min-Max Objective Function. *Discrete Applied Mathematics*, 26:105–115, 1990.

[11] J. Suurballe and R. Tarjan. A Quick Method for Finding Shortest Pairs of Disjoint Paths. *Networks*, 14:325–336, 1984.

[12] C. Pelsser and O. Bonaventure. Path Selection Techniques to Establish Constrained Interdomain MPLS LSPs. In *Proceedings of IFIP Networking'06*, Coimbra, Portugal, May 2006.

[13] R. Bhandari. *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[14] http://www.cidr-report.org/.