This paper was presented as part of the High-Speed Networks 2011 (HSN 2011) Workshop at IEEE INFOCOM 2011

Efficient PCE-Based Survivable Path Computation in Multi-Domain Networks

Qiong Zhang[1], M. Masud Hasan[2], Xi Wang[1], Paparao Palacharla[1], and Motoyoshi Sekiya[1]
[1]Fujitsu Laboratories of America, Inc., Richardson, TX
Email: {Qiong.Zhang, Xi.Wang, Paparao.Palacharla, Motoyoshi.Sekiya}@us.fujitsu.com
[2]Dept. of Math & Computer Science, Elizabeth City State University, NC
Email: mmhasan@mail.ecsu.edu

***Abstract*—We consider the problem of finding end-to-end shortest disjoint paths for a given sequence of domains in multi-domain networks. A Path Computation Element (PCE) serves as a computing entity in each domain, specializing in path computation and optimization. We propose a novel PCE-based scheme that computes the shortest path over multiple domains in the forward direction and computes the disjoint path in the backward direction. The proposed scheme has linear-time computation and message overhead in regard to the number of border nodes. Simulation results show that our proposed scheme can result in an optimal solution, while significantly reducing computation time compared to existing algorithms.**

***Index Terms*—Multi-domain networks, Optimal disjoint path computation, PCE**

I. INTRODUCTION

Networks are partitioned into multiple domains due to factors, such as administrative boundaries, locality, and underlying technologies. Domains are connected via inter-domain links. In such multi-domain networks, topology and resource information exchanged among domains is restricted because of scalability and privacy issues. Topology aggregation, which encodes the original topology of a domain with a compact auxiliary graph, is often used to reduce messages overhead, as well as to efficiently hide confidential details of each domain. One disadvantage of topology aggregation is the loss of information, which may significantly impact the results of optimal path computation. Hence, it is a challenging task in protocol design to balance the amount of information exchanged and the accuracy of path computation results [1, 2].

Recently, many emerging applications, such as Cloud computing and Datacenter networks, require agile, high-bandwidth, and fault-tolerant connections across multiple transport networks that are often managed as individual domains. Therefore, establishing MPLS/GMPLS Label Switched Paths across multi-domain networks has attracted much attention in the Telecommunication industry. Current MPLS/GMPLS control plane consists of components that perform functions such as signaling, routing, and path computation. This architecture is efficient for a single domain, but becomes very complex and inefficient when multiple domains are involved, especially when considering Traffic Engineering characters. To address this problem, the Internet Engineering Task Force (IETF) has been developing a PCE-based architecture that treats the path computation function as a separate and distinct component from the control plane. One major advantage is that the PCE-based architecture simplifies the control plane and enables CPU and memory intensive path computation, which may not be possible to perform at a single network element [3].

In this paper, we consider the problem of finding shortest disjoint working and backup paths (i.e., the total cost of two paths is minimum) over multiple domains. PCEs are used for exchanging topology aggregation and for optimal path computation. We focus on shortest disjoint path computation with a pre-determined sequence of domains, which can be a basic procedure for survivable path computation in multi-domain networks as adopted in RFC 5441 [4].

The sequence of domains can be administratively provisioned, dynamically discovered using inter-domain routing protocols, or dynamically computed using PCE architectures [5, 6]. In the case that the domain sequence is unknown, one solution as proposed in [5] can be to evaluate multiple candidate domain sequences using the proposed procedure for shortest disjoint path computation. Another case as proposed in [7] is that the working and backup paths follow different domain sequences. However, the optimal inter-domain disjoint path pairs may not be guaranteed in this case.

The main contribution of our paper is to propose novel signaling procedures and an efficient solution for survivable path computation with regards to computation time and message overhead. In the case that multiple candidate domain sequences are offered,

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

improving the efficiency of path computation becomes even more important. Our proposed schemes are aligned with protocol standards defined by IETF RFCs, which makes them practical for implementation and deployment in multi-domain networks.

We propose a PCE-based scheme called Forward Path Computation (FPC) to find the shortest path across multiple domains in the forward direction of a given sequence. Along with FPC, we use a procedure called Backward Path Computation (BPC) to search in the backward direction for the shortest disjoint path-pair given the shortest path computed by FPC. We call this scheme FPC-BPC. For disjoint path computation at each PCE, we apply Suurballe's algorithm for optimal solutions and a simpler two-step heuristic for faster solutions. Our proposed approaches require much less computation time and message overhead compared to the existing algorithms in [8, 9]. We present simulation results for both optimal and heuristic approaches. Due to space limitations, the correctness proof of FPC-BPC with Suurballe's algorithm will be presented in future publications.

The rest of this paper is organized as follows. Section II introduces the PCE-based architecture and related work. Section III includes the problem statement and describes our proposed FPC-BPC scheme. Section IV shows the simulation results. Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

In a PCE-based architecture [3], PCEs are an application that can locate at a network element or on an out-of-network server. Each domain may have a single centralized PCE or multiple distributed PCEs to compute paths. A PCE has the knowledge of detailed network topology in its local domain and is responsible for generating topology aggregation and communicating with other domains using Path Computation Element Communication Protocol (PCEP) [10]. Upon receiving a request from a Path Computation Client (PCC) that is also an application, a PCE may compute a path independently, or may initiate inter-domain communications and compute paths collaboratively with associated PCEs. The resulting path is sent back to the requesting PCC and can be used for the MPLS/GMPLS control plane to signal an inter-domain Label Switched Path (LSP).

The problem of finding shortest disjoint paths where the source and destination reside in the same domain has been solved by Suurballe [11, 12]. Similar problems of finding shortest disjoint paths where the source and destination reside in different domains are more challenging given that different domains are controlled by separate administrative entities. The problem over multiple domains has been solved in polynomial time in [8], where disjoint paths do not follow any predetermined sequence of domains. However, the solution works under an assumption that the "minimum weight path between a source node *s* and a destination node *t* traverses each routing domain at most once." Such an assumption cannot be guaranteed to be true for shortest path routing with arbitrary domain-sequences. In addition, the topology aggregation generated is quartic in terms of the number of border nodes and is flooded at the inter-domain level, which increases message overhead.

In RFC 5441, a backward-recursive PCE-Based computation (BRPC) procedure is proposed which finds the shortest (single) multi-domain path in polynomial time for a given sequence of domains [4, 13]. The basic idea is to exhaustively search for an optimal path. The search starts from the destination domain and goes backward towards the source domain. From one domain to the previous domain, a Virtual Shortest Path Tree (VSPT) is recursively sent that contains the shortest paths from all ingress border nodes to the destination.

A similar concept is extended to exhaustively search for the optimal pair of disjoint working and backup paths over a sequence of domains [9]. We term this approach BRPC-protection. In BRPC-protection, each PCE must compute shortest disjoint path-pairs from each possible pair of ingress border nodes to each possible pair of egress border nodes, which has a time complexity of $O(B^4)$, where B is the number of border nodes. Even though time complexity and message overhead are polynomially bounded with respect to border nodes, they are still costly to be implemented, especially in a dynamic centralized PCE architecture.

III. PROBLEM STATEMENT AND SOLUTION

Let D_i be the i^{th} domain in a multi-domain network. For a given sequence of domains, $D = (D_1, D_2, ..., D_i, ..., D_j, ...)$, the problem is to find a shortest disjoint path-pair (i.e., the total cost of the two paths are minimum) over multiple domains, where the source and the destination reside in different domains, and both working and backup paths follow the same domain-sequence. We assume that no domain is repeated in the domain-sequence. Links in the domains may be unidirectional or bidirectional. We assume link costs are non-negative. To calculate optimal paths across multiple domains, the cost of a link in a domain D_i and the cost of a link in another domain D_j should be comparable. If the cost metrics of the domains are different, then the PCEs can translate the cost metrics at domain boundaries.

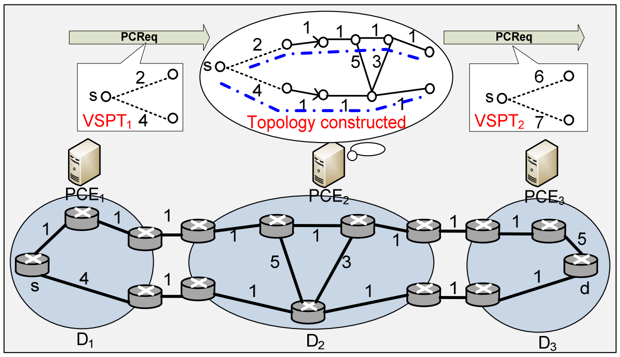


Figure 1. The FPC Scheme

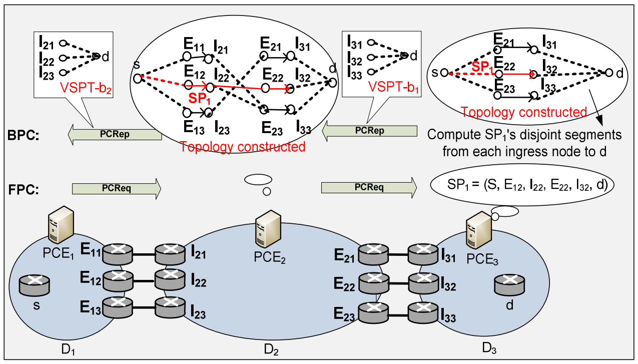


Figure 2. The FPC-BPC Scheme

We propose FPC-BPC to find shortest disjoint paths over multiple domains. FPC calculates the shortest path (SP_1) from the source to the destination using VSPT in the forward direction. BPC calculates another shortest path (SP_2) from the source to the destination using VSPT in the backward direction. Depending on the disjoint routing algorithm (Suurballe's algorithm or the two-step algorithm) used at PCEs, the path computation of SP_2 can be different. All PCEs should adopt the same algorithm for computing multi-domain paths in collaboration. FPC-BPC can be applied to both link-disjoint and node-disjoint path computation. We focus on link-disjoint computation in the rest of the paper.

A. Forward Path Computation (FPC)

FPC applies the VSPT data structure for topology aggregation and results in the shortest path that follows a pre-determined domain-sequence. FPC differs from BRPC in that FPC establishes VSPTs in the forward direction. In addition, FPC guarantees a given domain-sequence by modifying inter-domain links to be directional toward the destination.

Starting from the PCE at the source domain, VSPTs are sent (termed as VSPT-forward for FPC) to the PCE of the next domain. VSPT-forward contains shortest paths from the source to each egress border node. FPC has the following steps.

1) A PCC at the source sends PCReq to the source PCE that resides in the PCC's home domain.
2) The source PCE finds shortest paths from the source to egress nodes on the domain-sequence and builds a VSPT-forward ($VSPT_1$ in Fig. 1).
3) The source PCE sends the VSPT-forward via PCReq to a PCE of the next domain.
4) If the PCE receiving the VSPT-forward is not associated with the destination domain (i.e., it is an intermediate domain in the domain-sequence), it performs the following steps:
   a) Attach the VSPT-forward to ingress nodes using inter-domain links. Make inter-domain links directional toward the destination domain (see the topology constructed at PCE_2 in Fig. 1).
   b) Find shortest paths from the root of the VSPT-forward (i.e., the virtual source *s*) to egress nodes, and build a new VSPT-forward ($VSPT_2$ in Fig. 1). Go to Step 3.
5) Otherwise, if the PCE receiving the VSPT-forward is associated with the destination domain, it performs the following steps:
   a) Attach the VSPT-forward to ingress nodes using inter-domain links. Make inter-domain links directional toward the destination.
   b) Find a shortest path from the virtual source to the actual destination. This path is the desired shortest path, SP_1.

B. Backward Path Computation (BPC)

BPC aims to find an optimal disjoint path-pair given the shortest path SP_1 computed by FPC. BPC's VSPT (termed as VSPT-backward) consists of path segments disjoint from SP_1.

Two link-disjoint routing algorithms can be used at PCEs when calculating path segments for VSPT-backward.

- For finding an optimal solution, apply Suurballe's algorithm. Replace each link on SP_1 by a link directed towards the source and make the cost of those links negative as described in [12]. Compute shortest paths from all ingress nodes to the virtual destination.
- For a simple implementation, apply a two-step scheme. Remove all links that are included in SP_1. Compute shortest paths from all ingress nodes to the virtual destination.

Starting from the PCE at the destination domain, BPC has the following steps:

1) The PCE at the destination domain performs the following steps:
   a) Attach the VSPT-forward (received during FPC) to ingress nodes using inter-domain links. Make inter-domain links directional toward the destination (see the topology constructed at PCE_3 in Fig. 2).
   b) Run a link-disjoint routing algorithm (Suurballe's or the two-step algorithm) from each ingress node to the destination given SP_1, and build a VSPT-backward (*VSPT-b*$_1$ in Fig. 2).
2) The PCE sends SP_1 and the VSPT-backward via PCRep to the previous domain.
3) If the PCE receiving the SP_1 and VSPT-backward is not the source domain (i.e., it is an intermediate domain in the domain-sequence), it performs the following steps:
   a) Attach the VSPT-backward to egress nodes and attach the VSPT-forward (received during FPC) to ingress nodes using inter-domain links. Make inter-domain links directional toward the destination. (See the topology constructed at PCE_2 in Fig. 2)
   b) Run a link-disjoint routing algorithm from each ingress node to the root of VSPT-backward (i.e. the virtual destination *d*) given SP_1, and build a new VSPT-backward (*VSPT-b*$_2$ in Fig. 2). Go to Step 2.
4) If the PCE receiving the SP_1 and VSPT-backward is the source domain, it performs the following steps:
   a) Attach the VSPT-backward to egress nodes using inter-domain links and make inter-domain links directional toward destination.
   b) Run a link-disjoint routing algorithm from the source to the virtual destination to obtain SP_2.

C. Constructing Two Disjoint Paths

If the two-step disjoint routing algorithm is adopted at each PCE, the resulting SP_1 and SP_2 of FPC-BPC are the desired disjoint paths. If Suurballe's algorithm is used at each PCE, we can derive two disjoints paths from SP_1 and SP_2. If SP_1 and SP_2 do not share any segments (i.e., no overlapping), the answer is immediate. We now focus on how to remove overlapped segments between SP_1 and SP_2 in order to derive two disjoint paths.

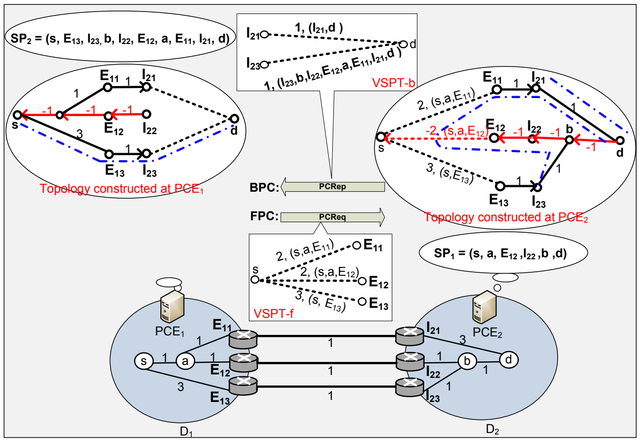


Figure 3. FPC-BPC with Suurballe's Algorithm

The path segments included in a VSPT can be either explicit (i.e., all the nodes traversed in a path segment are listed) or loose (only the border nodes traversed are listed) [4]. When explicit paths are used, the source PCE can directly calculate two disjoint paths by removing overlapped segments as described in [12]. When loose paths are used, a path-key technique in RFC 5520 [14] can be used to retrieve the explicit paths on a per-domain basis during the GMPLS/MPLS signaling process. Overlapping segments inside a domain can then be removed on a per-domain basis.

Figure 3 depicts an example of constructing disjoint paths when explicit paths are used for FPC-BPC with Suurballe's algorithm. On the topology constructed at PCE_2, the disjoint path from I_{23} to d is (I_{23}, b, I_{22}, E_{12}, a, s, s, a, E_{11}, I_{21}, d). We have proved that the loop (a, s, s, a) can be removed with no impact on the optimal result. Hence, the path segment (I_{23}, d) in *VSPT-b* has the path (I_{23}, b, I_{22}, E_{12}, a, E_{11}, I_{21}, d). At the end of FPC-BPC, PCE_1 computes SP_2, removes the overlap of SP_1 and SP_2, and obtains two disjoint paths: (s, E_{13}, I_{23}, b, d) and (s, a, E_{11}, I_{21}, d). Note that, if Suurballe's algorithm is used at each PCE, a path in VSPT-backward may traverse the segments in VSPT-forward due to overlapping between SP_1 and SP_2.

D. Complexity Comparison

We compare the performance metrics such as signaling delay, the number of message exchanges, message complexity, and time complexity of different path computation schemes in Table I. Similar to BRPC-protection, FPC-BPC requires a single round trip time for signaling. Also, FPC-BPC has the same number of message exchanges as BPPC-protection.

TABLE I
A COMPARISON OF MULTI-DOMAIN PATH COMPUTATION SCHEMES

Problems	Schemes	Signaling delay	Number of message exchanges	Storage complexity per message	Time complexity
Protected (disjoint) shortest path routing without a given domain-sequence	Sprintson et al. [8]	1 round trip	$O(D^2)$	$O(D^2B^4)$	$O(D^2B^3S+D^2B^2)$
Protected (disjoint) shortest path routing with a given domain-sequence	BRPC-protection [9]	1 round trip	$O(D)$	$O(B^2)$	$O(DB^4S)$
	FPC-BPC	1 round trip	$O(D)$	$O(B)$	$O(DBS)$
Unprotected (single) shortest path routing with a given domain-sequence	BRPC [4, 13]	1 round trip	$O(D)$	$O(B)$	$O(DS)$
	FPC	1 round trip	$O(D)$	$O(B)$	$O(DS)$

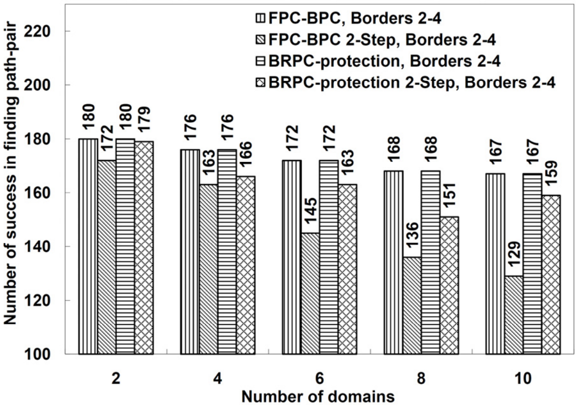
Notation: D – the number of domains in the network,
B – the maximum number of border nodes within a domain
S – the time complexity of a shortest path routing algorithm (e.g., the Dijkstra's algorithm)

However, each message of FPC-BPC contains a single VSPT whose size is linear to the number of border nodes, while the message exchanged in the backward direction of BRPC-protection contains disjoint paths from all possible pairs of ingress nodes to the destination, which is quadratic to the number of border nodes. Unlike BRPC-protection that computes all possible disjoint path-pairs, BPC computes paths disjoint from the previously-computed shortest path, resulting in linear time complexity in the number of border nodes, which significantly reduces the search space for the optimal solution.

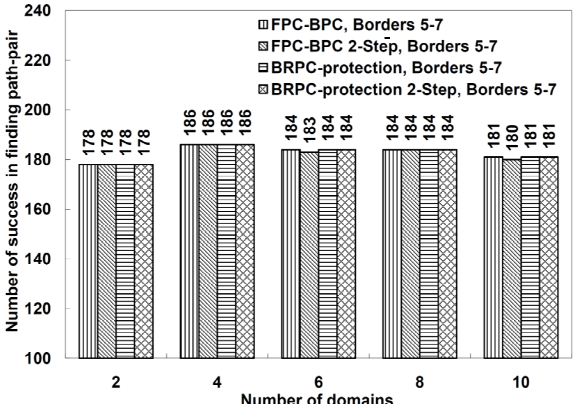
IV. NUMERICAL RESULTS

We implemented our proposed heuristics and optimal schemes for finding a disjoint shortest path-pair. Since a pre-determined domain sequence is considered, any practical multi-domain network can be treated as a network with a linear sequence of domains. Hence, we simulate a multi-domain network with a sequence of domains varying from 2 to 10 domains. Each domain includes 16 to 24 nodes. The number of ingress/egress border nodes ranges either from 2 to 4 or from 5 to 7. Links in the networks are bidirectional and have costs between 1 and 10. Each data point presented is averaged from 200 randomly generated networks. A single demand is generated for each network, where the source (or destination) is randomly chosen from the source (or destination) domain. We compare four schemes:

1) FPC-BPC: optimal FPC-BPC with Suurballe's algorithm (described in Section III).
2) FPC-BPC 2-Step: FPC-BPC with the two-step heuristic (described in Section III).
3) BRPC-protection: optimal BRPC-protection with Suurballe's algorithm.
4) BPRC-protection 2-Step: BRPC-protection with the two-step heuristic.



(a) Number of success for 2-4 border nodes



(b) Number of success for 5-7 border nodes
Figure 4. Comparison of number of success in finding path-pairs
(Total number of requests is 200)

As shown in Fig. 4, using Suurballe's algorithm, FPC-BPC and BRPC-protection have the same number of successful requests. Fig. 4 (a) shows that, when there are 2 to 4 border nodes, BRPC-protection 2-Step has higher success in finding a path-pair than FPC-BPC 2-Step, since BRPC-protection 2-step adopts an exhaustive search. The success rate of FPC-BPC 2-Step can be significantly improved if rerouting is applied in domains where the path computation fails. Thus, there is a tradeoff between the success rate and computation

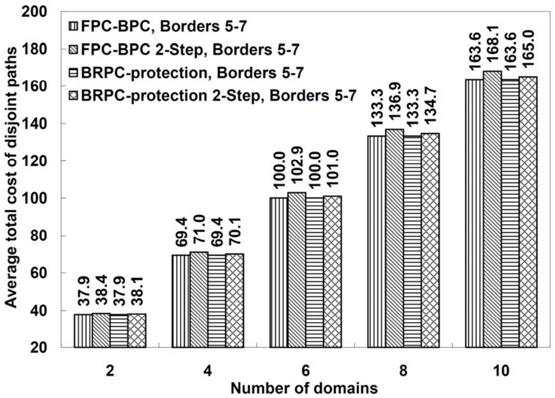


Figure 5. Comparison of average total cost per disjoint path-pair

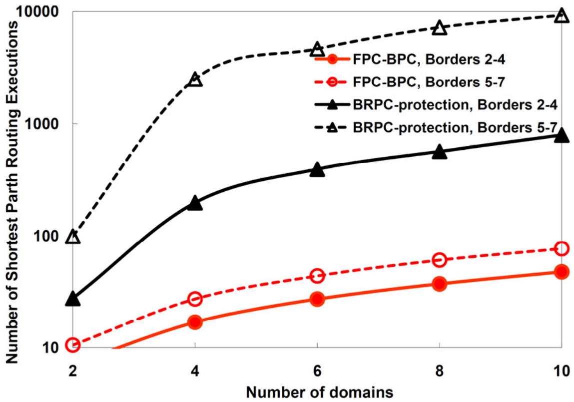


Figure 6. Number of shortest path routing executions per demand

complexity. When there are 5 to 7 border nodes, as shown in Fig. 4(b), FPC-BPC 2-Step performs close to BRPC-protection 2-Step, but has much lower computational complexity compared to BRPC-protection 2-Step, as shown in Fig. 6.

Figure 5 shows that both FPC-BPC and BRPC-protection are optimal in finding the lowest cost path-pairs. BRPC-protection 2-Step has slightly (up to 2%) lower average cost than FPC-BPC 2-Step due to a larger search space. A similar trend is observed from the simulation results for 2 to 4 border nodes, which are not shown due to space limitations.

Figure 6 compares the average number of executions of shortest path routing per demand, since the shortest path routing has the highest computation time in the implementation. We see that FPC-BPC requires significantly fewer executions of shortest path routing than BRPC-protection due to the much smaller search space. For example, when the number of domains is 4 and the number of border nodes is 5 to 7, FPC-BPC requires an average of 27 executions, while BRPC-protection requires an average of 2511 executions for a single demand, which is quartic in terms of the number of border nodes (as shown in Table I). We also see that BRPC-protection becomes computationally costly with an increasing number of border nodes. Hence, BRPC-protection may not be practical with a high number of border nodes, especially in a centralized PCE system.

V. CONCLUSION

We have proposed the FPC-BPC scheme to find shortest disjoint paths in multi-domain networks with PCEs. The time complexity and message overhead are significantly improved in FPC-BPC compared to other existing schemes. The FPC-BPC scheme with Suurballe's algorithm can achieve optimal solutions. In addition, we have included FPC-BPC with a two-step heuristic for the simplicity of implementation.

Future research area includes applying FPC-BPC under a more general assumption, where the working path can follow different domain-sequence from the backup path.

REFERENCES

[1] K. Liu, K. Nahrstedt, and S. Chen, "Routing with Topology Abstraction in Delay-Bandwidth Sensitive Networks," *IEEE/ACM Trans. Net.*, vol. 12, no. 1, pp. 17–29, Feb. 2004.

[2] F. Hao and E. Zegura, "On Scalable QoS Routing: Performance Evaluation of Topology Aggregation," in *Proc. of INFOCOM*, Tel-Aviv, Isreal, March 2000, pp. 147–56.

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

[4] J. P. Vasseur, et al., "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths," RFC 5441, April 2009.

[5] D. King and A. Farrel, "The Application of the Path Computation Element Architecture to Determination of a Sequence of Domains in MPLS & GMPLS," Internet Draft, July 2010.

[6] M. Chamania, et al., "An Adaptive Inter-domain PCE framework to Improve Resource Utilization and Reduce Inter-domain Signaling," *Optical Switching and Networking*, vol. 6, no. 4, pp. 259-267, Dec 2009.

[7] S. Secci, J.-L. Rougier, and A. Pattavina, "AS-level Source Routing for Multi-Provider Connection-Oriented Services," *Computer Networks*, vol. 54, no. 14, pp. 2453-2467, October 2010.

[8] A. Sprintson, et al., "Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks," in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, May 2007, pp. 1820–1828.

[9] I. Nishioka and D. King, "The Use of SVEC (Synchronization VECtor) List for Synchronized Dependent Path Computations," Internet Draft, June 2010. Available:www.ietf.org/id/draft-ietf-pce-pcep-svec-list-05.txt

[10] J. P. Vasseur and J. L. Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)," RFC 5440, March 2009.

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

[12] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. Springer, 1st Edition, Jan 15, 1999.

[13] S. Dasgupta, et al., "Path-Computation-Element-Based Architecture for Inter-domain MPLS/GMPLS Traffic Engineering: Overview and Performance," *IEEE Network*, vol. 21, no. 4, pp. 38–45, Jul-Aug 2007.

[14] R. Bradford, J. P. Vasseur, and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism," RFC 5520, April 2009.