Segment Routing for Effective Recovery and Multi-domain Traffic Engineering

A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini, and P. Castoldi

***Abstract*—Segment routing is an emerging traffic engineering technique relying on Multi-protocol Label-Switched (MPLS) label stacking to steer traffic using the source-routing paradigm. Traffic flows are enforced through a given path by applying a specifically designed stack of labels (i.e., the *segment list*). Each packet is then forwarded along the shortest path toward the network element represented by the top label. Unlike traditional MPLS networks, segment routing maintains a per-flow state only at the ingress node; no signaling protocol is required to establish new flows or change the routing of active flows. Thus, control plane scalability is greatly improved. Several segment routing use cases have recently been proposed. As an example, it can be effectively used to dynamically steer traffic flows on paths characterized by low latency values. However, this may suffer from some potential issues. Indeed, deployed MPLS equipment typically supports a limited number of stacked labels. Therefore, it is important to define the proper procedures to minimize the required segment list depth. This work is focused on two relevant segment routing use cases: dynamic traffic recovery and traffic engineering in multi-domain networks. Indeed, in both use cases, the utilization of segment routing can significantly simplify the network operation with respect to traditional Internet Protocol (IP)/MPLS procedures. Thus, two original procedures based on segment routing are proposed for the aforementioned use cases. Both procedures are evaluated including a simulative analysis of the segment list depth. Moreover, an experimental demonstration is performed in a multi-layer test bed exploiting a software-defined-networking-based implementation of segment routing.**

***Index Terms*—Multi-domain traffic engineering; Restoration; Segment routing; Software-defined networking.**

I. Introduction

Segment routing (SR) has recently been proposed within the Internet Engineering Task Force (IETF) to provide traffic engineering (TE) by simplifying control plane operation [1]. SR is essentially an implementation of the source-routing paradigm. A specific header, composed of a stack of Multi-protocol Label-Switched (MPLS) labels (i.e., the *segment list*), is included in each transmitted packet by the source node so that the traffic flows are routed through the desired path. Therefore, the segment list is compatible with the standard MPLS data plane and consists of an ordered list of *segment* identifiers. For instance, a segment can identify a network node (i.e., *node segment*), specific node interface (i.e., *adjacency segment*), or service to be applied to the traffic flow (i.e., *service segment*). Considering a segment list composed of node segments only, each intermediate node processes the top label in the segment list to determine the interface to be used for forwarding the packet, i.e., the interface along the shortest path toward the node represented by the top label. Thus, the source node is enabled to route the traffic along an explicit path by properly setting the segment list.

In traditional Internet Protocol (IP)/MPLS networks, labels have a local meaning and therefore a signaling protocol (e.g., Resource Reservation Protocol with Traffic Engineering extensions, RSVP-TE) is used for label exchange every time a new traffic flow has to be activated in the network, and a detailed per-flow state is maintained in each node traversed by the established label switched path (LSP). Conversely, using SR labels may have a global meaning, and therefore a signaling protocol is not required to perform explicit routing and a per-flow state is maintained only at the ingress node where the segment list is enforced. This approach significantly simplifies the control plane operation, especially in multi-layer networks, where SR can eliminate the need to establish and maintain hierarchical instances of generalized MPLS (GMPLS) LSPs [2]. Moreover, SR natively implements equal cost multi-path (ECMP)-aware routing, i.e., in the case of multiple shortest paths toward the destination the traffic is automatically load-balanced on a per-flow basis. This characteristic also simplifies the control plane operation where complex configurations are often required to properly deploy load-balancing policies. On the other hand, SR introduces a header in every data packet (i.e., the segment list), and thus the depth of the segment list should be kept limited because it reduces the available payload area and because commercial MPLS equipment typically supports a limited number of stacked labels [3].

Besides simplification of the control plane operation, other interesting SR use cases have been recently proposed [4]. First, by changing the segment list applied to a specific traffic flow, SR can be effectively used to dynamically steer traffic on the path characterized by the most suitable traffic engineering parameters, e.g., to dynamically avoid link congestion. Second, SR enables a straightforward implementation of *service function chaining* through the stacking of the aforementioned service segments [5]. Third,

Manuscript received June 17, 2016; revised November 18, 2016; accepted December 17, 2016; published January 27, 2017 (Doc. ID 268421).

A. Giorgetti (e-mail: a.giorgetti@sssup.it), F. Paolucci, and P. Castoldi are with the TeCIP Institute, Scuola Superiore Sant'Anna, Pisa, Italy.

A. Sgambelluri is with KTH Royal Institute of Technology, Kista, Sweden.

F. Cugini is with CNIT, Pisa, Italy.

https://doi.org/10.1364/JOCN.9.00A223

centralized implementation of operations, administration, and maintenance procedures have also been proposed by exploiting SR functionalities to overcome complex RSVP-TE signaling procedures required for establishing monitoring LSPs [6]. Finally, SR can be effectively used upon network failures to perform traffic recovery without involving the controller and without requiring signaling during the recovery process, and to enable the deployment of effective inter-domain TE solutions.

This paper proposes, implements, and experimentally validates two original procedures based on segment routing for addressing dynamic traffic recovery and inter-domain TE solutions in multi-layer networks.

Regarding traffic recovery, a SR scheme is proposed to dynamically recover traffic flows disrupted by link or node failures minimizing the depth of the required segment list. In today's IP/MPLS networks, recovery is guaranteed by a *fast reroute* [7,8] that typically reroutes disrupted traffic from the node detecting the failure toward the next or next-next hop. Indeed, using fast reroute, each backup path requires explicit setup (i.e., RSVP-TE signaling) and periodic refresh. Thus, merging the protected path with a backup path as close as possible to the failure point is a good practice to reduce signaling overhead. Conversely, with the proposed SR approach, signaling and refresh of backup paths are not required, and thus traffic recovery can be enforced from the local point of failure directly to the destination node, thus achieving a more TE-effective solution.

Regarding inter-domain TE, two schemes are proposed and compared for applying the SR concept in a multi-domain network scenario where the source-routing concept is not directly applicable since network state information of remote domains is not available at the source node. Moreover, given the absence of signaling sessions in the data plane, SR enables the scalable concatenation of multi-domain paths rarely deployed in practical scenarios, overcoming the limitations and interoperability issues of RSVP-TE stitching and nesting solutions. The proposed schemes are first evaluated by means of simulations in order to evaluate their performance and scalability in terms of segment list depth. Moreover, they are experimentally validated in a test bed deploying the SR approach in a software-defined networking (SDN) environment.

II. Previous Work

Dynamic restoration is a well investigated topic in both optical and multi-layer (e.g., IP over WDM) networks [9,10]. However, most previous work considers the utilization of the distributed GMPLS control plane, which features some important drawbacks due to possible contention among different signaling sessions [11]. From this point of view, recovery schemes based on the emerging SDN control plane can provide significant benefits if the centralized controller is properly exploited [12,13].

Several recovery schemes have been proposed for implementation in Ethernet and IP/MPLS networks using the SDN approach and OpenFlow protocol. Specifically, the authors of [14–16] design restoration schemes for carrier-grade Ethernet networks where the switch detecting the failure notifies the controller about the topology change. Then, the controller installs the backup paths in the data plane. In Ref. [17], a recovery mechanism is proposed based on path protection where backup paths are computed and installed before the failure occurrence, thus not involving the controller upon failure. Finally, [18] proposes local traffic re-direction from the point of failure to the destination. Also, this mechanism does not involve the controller upon failure. However, all the aforementioned schemes may suffer from scalability issues. Indeed these methods, if not involving the controller in the recovery mechanism, require a number of backup flow entries in the nodes that linearly increases with the number of working flows established in the network. Conversely, by exploiting SR, local traffic recovery can be enabled using a number of backup flow entries that does not depend on the number of flows established in the network. This is an important contribution of this work; based on SR, it proposes a dynamic recovery scheme not involving the central controller upon failure occurrence and in which the number of backup flow entries only depends on the network topology.

Regarding multi-domain traffic engineering, the first solutions proposed for GMPLS-based optical and multi-layer networks were based on Border Gateway Protocol (BGP) [19,20]. However, not advertising actual resource availability, BGP does not allow effective inter-domain path computation. To address this issue, the path computation element (PCE) architecture has been extended to support inter-domain path computation using a distributed communication process among PCEs [21]. Later, the hierarchical PCE (HPCE) architecture has been introduced where a parent PCE (pPCE) coordinates the inter-domain path computation process [22]. More recently, the inclusion of intra-domain information directly at the pPCE has been proposed [23,24], where such information is provided to the pPCE by resorting to recent BGP extensions [25]. However, this solution is not easily applicable in a multi-provider scenario because of confidential information that should be shared at the pPCE.

Multi-domain TE solutions have also been proposed using the SDN control plane [26–28], showing that the utilization of SDN is able to simplify the path setup and reduce the path setup time. Indeed, after path computation, the required signaling is performed in parallel in each domain. In this sense, SR can provide additional benefits, especially in multi-domain heterogeneous networks, because traffic can be transmitted immediately after path (and segment list) computation without requiring signaling traversing multiple domains, which would be prone to interoperability issues.

Focusing on SR, standardization is rapidly evolving [1,29] and relevant research work has been conducted within the academic community. The authors of [30] proposed to combine the benefits of SR with those of a SDN control plane. The work in Ref. [31] implemented SR in carrier-grade Ethernet networks, including experimental and simulation studies. Algorithms to compute the segment list encoding a given path are proposed in

Refs. [3,32,33]. Specifically, Refs. [3] and [33] propose the utilization of an auxiliary graph model representing the available network segments for computing the segment list, whereas the work in Ref. [32] proposes a greedy algorithm to compute the segment list of minimum depth. The works in Refs. [34,35] formulate a multi-commodity flow problem to evaluate the benefits of SR. Specifically, [34] reports an achievable reduction of up to an order of magnitude in the state maintained in routers by using SR instead of RSVP-TE, whereas [35] shows that, using a segment list composed of only two labels, SR is able to provide significant benefits with respect to shortest path routing. Finally, several works, including [36–38], detail experimental implementations and evaluations of the SR architecture.

The only proposals for implementing dynamic recovery using SR are [39,40] and our previous work [41]. In the scheme proposed by the authors of [39], backup paths are computed at the controller after failure occurrence [40] provides a multi-commodity formulation of the dynamic recovery problem in SR networks and quantifies the achievable capacity benefits. The work in Ref. [41] proposes a procedure, similar to MPLS fast reroute [7,8], to locally re-route disrupted traffic flows around a faulted network element without involving the central controller. The scheme proposed in this paper enhances the one in Ref. [41] by implementing the recovery from the point of failure to the destination node, thus minimizing the segment list depth. Finally, to the best of our knowledge, no SR solution has been proposed in the literature to deploy multi-domain TE policies.

This work extends and integrates the conference papers [42,43] that respectively propose techniques for the support of recovery and multi-domain routing in SR networks. With respect to the aforementioned works, this manuscript adapts and evaluates the recovery technique to the node failure scenario and provides extended simulative and experimental results.

III. Segment Routing Operation

This section explains SR operation considering the utilization of a centralized controller (e.g., a SDN controller [30,44]). This way, when a new traffic flow has to be established, a request is issued to the controller that computes the path, encodes the path using a segment list, and properly configures the ingress node to enforce the computed segment list. However, SR can also be deployed in fully distributed networks, where path and segment list computation are performed locally at the ingress node.

In Fig. 1, each node is reported with the associated forwarding table assuming that an interior gateway protocol (IGP) is advertising the node identifiers, as proposed in Ref. [45]. When a new traffic request arrives from node A to the destination network, the controller computes the path $\bar{p}_1 = \{A, B, C, D\}$. Since $\bar{p}_1$ is the *unique shortest path* from A to I, the segment list $\overline{SL}^{\bar{p}_1}$ encoding $\bar{p}_1$ includes only the label identifying the last node in the segment routing domain (i.e., $\overline{SL}^{\bar{p}_1} = \{D\}$). Thus, the controller configures the ingress node forwarding table. This way, packets are forwarded along $\bar{p}_1$ without modifying the segment list up from node A to node D, where the label is popped and packets are forwarded on the destination network.

Alternatively, if the path computed at the controller is not the unique shortest path, a more complex segment list is required. For instance, if path $\bar{p}_2 = \{A, B, E, C, D\}$ is considered, the associated segment list is $\overline{SL}^{\bar{p}_2} = \{E, D\}$ (see last entry of A forwarding table in Fig. 1). This way, packets are forwarded up to node E without modification to the segment list. At node E the first label is popped, and traffic is forwarded to node C, i.e., along the unique shortest path to node D. Similarly, since SR natively implements ECMP-aware routing, when a strict path is desired on a topology including ECMPs, proper identifiers of intermediate nodes should be included in the segment list to avoid load balancing.

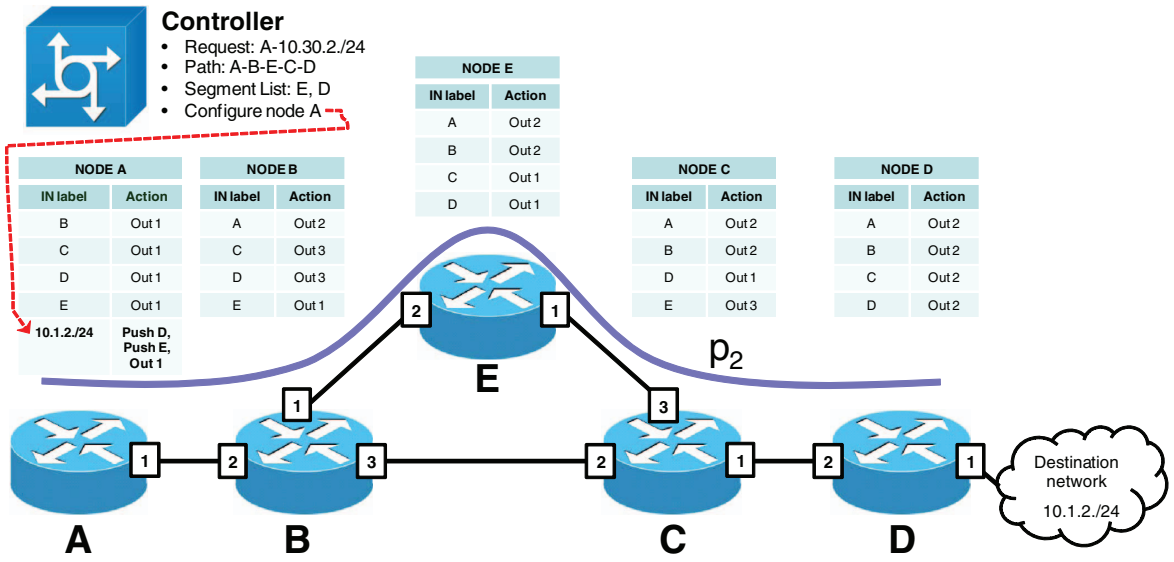


Fig. 1. SR operation example; all nodes are reported with the associated IP/MPLS forwarding table. In the figure, node A is configured by the controller to enforce $\bar{p}_2$ with the actions push E, push D, out 1; actions to enforce $\bar{p}_1$ are push D, out 1.

IV. Traffic Recovery Using Segment Routing

Current recovery solutions deployed in IP/MPLS networks (e.g., fast reroute) exploit detours from the node detecting the failure toward the next or the next-next hop. While rerouting disrupted traffic from the node detecting the failure is widely recognized as the best option to guarantee fast recovery, redirecting it directly to the destination node is a more effective solution in terms of resource utilization. Thus, the proposed SR recovery mechanism (i.e., the SR-FAILOVER scheme) implements rerouting of the traffic from the node detecting the failure up to the destination (i.e., the node indicated in the deepest label of the segment list). To this extent the SR-FAILOVER scheme proposes to use a *primary forwarding table* in each node and a number of *failover forwarding tables*; see Fig. 2. Specifically, one failover table is required for each interface of the node. Primary and failover tables include N entries, where N is the number of node segments belonging to the routing domain. Thus, the number of required entries does not depend on the number of flows established in the network.

Using the SR-FAILOVER scheme, when the output port indicated in the primary table for a specific ingress label is down, the backup actions in the primary table are executed. By applying the backup actions, the node first pops all the labels in the segment list except the deepest label (i.e., the bottom of the stack), and then the packet processing is passed on to the proper failover table; see Fig. 2(c). The actions to be executed within the failover tables depend on the value of the bottom label. Specifically, for each label, the failover table enforces the utilization of a loop-free backup path. To do this, a number of push actions may be required before forwarding the packet (e.g., in the case of port 2 failure at node B, a push action is required only for recovering traffic arriving with label D). Backup paths are computed on the updated network topology, excluding the failed link or all the links attached to the failed node. Computed backup paths are then encoded with a segment list so that traffic is routed to the destination while avoiding ECMP load balancing.

To implement the proposed scheme, each node has to be properly configured during network initialization so that, when a node physically detects a failure of a connected interface, it is able to locally redirect the traffic on the proper backup path. The network initialization can be performed in a distributed way using a properly extended IGP [1,45] or exploiting a centralized SR controller that continuously monitors the network topology and enforces the required flow table entries in the network nodes [37].

This work considers the adoption of a SR controller since it is more consistent with the SDN approach and it easily enables the configuration of multiple forwarding tables as required by the SR-FAILOVER scheme. The SR controller includes a traffic engineering database (TED), including network topology and resource utilization information that are used for path computation purposes. Specifically, the SR controller exploits the SDN approach to program the data plane using the OpenFlow protocol [37]. First, primary and failover tables are properly initialized. Then, when a new traffic flow has to be activated, the ingress node can simply apply the destination node segment or ask the SR controller. In this latter case, the controller replies with a specific segment list to enforce a path that is centrally computed considering all the required constraints.

NODE A

IN Label	Action
B	Out 1
C	Out 2
D	Out 1
E	Out 2

NODE B

IN Label	Action
A	Out 1
C	Out 3
D	Out 2
E	Out 2, Out 3

NODE C

IN Label	Action
A	Out 1
B	Out 3
D	Out 2, Out 3
E	Out 2

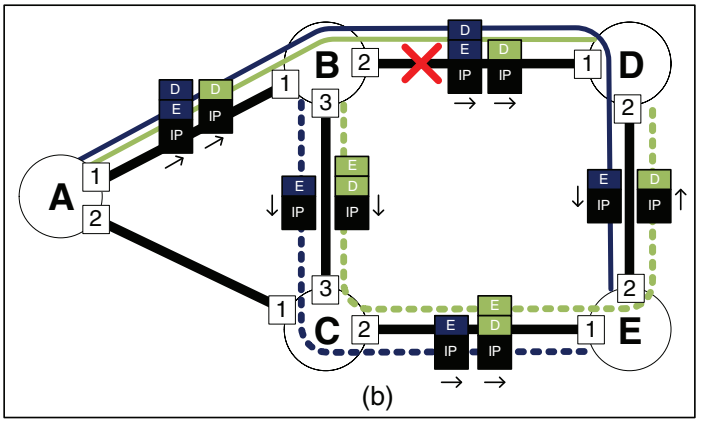
NODE D

IN Label	Action
A	Out 1
B	Out 1
C	Out 1, Out 2
E	Out 2

NODE E

IN Label	Action
A	Out 1
B	Out 1, Out 2
C	Out 1
D	Out 2

(a)



NODE B

IN Label	Action	Backup Action
A	Out 1	Pop to BOS, table fail 1
C	Out 3	Pop to BOS, table fail 3
D	Out 2	Pop to BOS, table fail 2
E	Out 2, Out 3	Pop to BOS, table fail 2 or 3

B - fail 1

IN	Action
A	Out 3
C	Out 3
D	Out 2
E	Out 3

B - fail 2

IN	Action
A	Out 1
C	Out 3
D	Push E, Out 3
E	Out 3

B - fail 3

IN	Action
A	Out 1
C	Out 1
D	Out 2
E	Out 2

(c)

Fig. 2. SR-FAILOVER scheme description for the link failure use case. (a) Primary forwarding table of each node. (b) Test bed topology with two active traffic flows; solid lines represent the working path and dashed lines represent the backup path. (c) Primary forwarding table and failover forwarding tables of node B. The first backup action in all flow entries is *Pop to BOS*, i.e., pop all the MPLS labels in the segment list except the bottom of the stack (BOS).

In any case, the SR controller is not involved in the recovery mechanism so that recovery time is minimized.

Figure 2 illustrates the required flow table configuration to deploy the SR-FAILOVER scheme in the depicted sample topology. Specifically, Fig. 2(a) illustrates all the primary tables of the nodes belonging to the network topology illustrated in Fig. 2(b). The flows installed in the primary tables simply implement the shortest path routing toward each network node considering ECMP load balancing. For instance, at node B, two ECMPs are available toward node E and therefore per-flow load balancing is applied using ports 2 and 3.

The blue solid line in Fig. 2(b) represents a working traffic flow traversing the path $\bar{p}_1 = \{A, B, D, E\}$, and this path is encoded with the segment list $\overline{SL}^{\bar{p}_1} = \{D, E\}$. Thus, node B receives the traffic with label D and forwards the traffic toward node D using interface 2. In the case of failure of this interface, label D is popped and then the switch is redirected to failover table 2, which implements the routing on a topology where the link B–D is pruned. Specifically, Fig. 2(c) illustrates all the failover tables configured at node B; in the case of $\bar{p}_1$, the backup path is $\{A, B, C, E\}$, i.e., blue dashed line in Fig. 2(b). Since the bottom of the stack is label E, traffic is simply forwarded on interface 3 without requiring further actions. Conversely, for path $\bar{p}_2 = \{A, B, D\}$ [i.e., green solid line in Fig. 2(b)], the backup path is $\{A, B, C, E, D\}$, i.e., green dashed line in Fig. 2(b). In this case the destination is D and, to assure a loop-free route, a new label is required to be pushed by node B (i.e., label E) before forwarding it on interface 3. This way, at node C, the traffic matches the desired backup path in the primary table and it is forwarded using interface 2.

The SR-FAILOVER scheme as described in the previous paragraph considers single link failures. However, it can be easily extended to node failures. Indeed, it is sufficient to initialize the failover tables computing the backup paths on the topology by pruning all the links connected to the failed node. As an example, if node A detects a failure on interface 1, the backup paths should be computed assuming the failure of node B, i.e., pruning bidirectional links A–B, C–B, and D–B from the network topology.

V. Multi-domain TE Using Segment Routing

This section proposes two schemes based on SR to enable effective TE in multi-domain networks, i.e., *end-to-end* segment routing (e2e-SR) and *per-domain* segment routing (pd-SR). Also, in this case, the proposed schemes can be implemented in a distributed way by using proper BGP extension [1] or, as considered in this work, in a scenario where each domain relies on a dedicated SR controller. However, since the advertisement of node segment identifiers is limited within the domain boundaries, the SR controller of each domain is not aware of the intra-domain topology of other domains, i.e., the stored TED includes only information of the specific domain. Therefore, a communication procedure among SR controllers should be used to perform effective inter-domain TE.

Both proposed schemes leverage a two-way communication session established among the SR controllers of the traversed domains. However, the two schemes use a different procedure to determine the segment list to be applied to data packets. In e2e-SR, a segment list with end-to-end validity is enforced by the source node and never integrated along the path to the destination node. Conversely, in the pd-SR scheme, the segment list is integrated at each ingress border node traversed by the traffic flow. In both cases, the sequence of border nodes to be traversed is considered to be known in advance, e.g., it can be provided by the network management system (NMS) or computed by a hierarchical controller such as in the standardized HPCE architecture [20].

Figure 3(a) details the e2e-SR scheme procedure in a network including three domains (i.e., $D1$, $D2$, and $D3$). When a traffic request (from node A to node F) is generated by the NMS, the SR controller of domain $D1$ forwards the request to the controller of subsequent domains through the assigned sequence of domains. When the request reaches the controller of the destination domain [i.e., SR controller of domain $D3$ in Fig. 3(a)], it computes a path from its ingress border node [i.e., E in Fig. 3(a)] to the destination node F and forwards the related segment list to the upstream SR controller. Intermediate controllers apply the same procedure where the forwarded segment list is obtained by stitching the segment list received by the downstream controller with the segment list computed locally. This way, when the SR controller of domain $D1$ receives the reply, it is able to build up the end-to-end segment list and properly configure the source node forwarding table, thus enforcing the end-to-end segment list to the incoming packets belonging to the flow.

Figure 3(b) details the pd-SR scheme procedure that is specifically designed to provide shorter segment lists and to preserve confidentiality of internal topology information. In this case, the SR controllers do not forward the computed segment list to the upstream controller; instead they just forward a *virtual* label [i.e., X and Y, respectively for $D2$ and $D3$ in Fig. 3(b)] and contextually configure the ingress border node to properly process packets that will arrive using that virtual label. In Fig. 3(b), node C is configured so that, for those packets using label X, the segment list is set to D–E–Y, whereas node E is configured so that, for packets using label Y, the segment list is set to F. This scheme preserves the intra-domain confidentiality because virtual labels assume a topological meaning only when they are expanded within the specific domain. Moreover, the pd-SR scheme also provides potential benefits in terms of fast response to network changes. For instance, if a link failure is detected inside a domain, the local controller can change the meaning of the specific virtual label without requiring any coordination with other controllers.

VI. Simulative and Experimental Validation

A. Segment Routing Recovery

To assess the scalability of the SR-FAILOVER scheme in terms of *segment list depth* (SLD), two simulation scenarios are considered: a pan-European network including 27 nodes

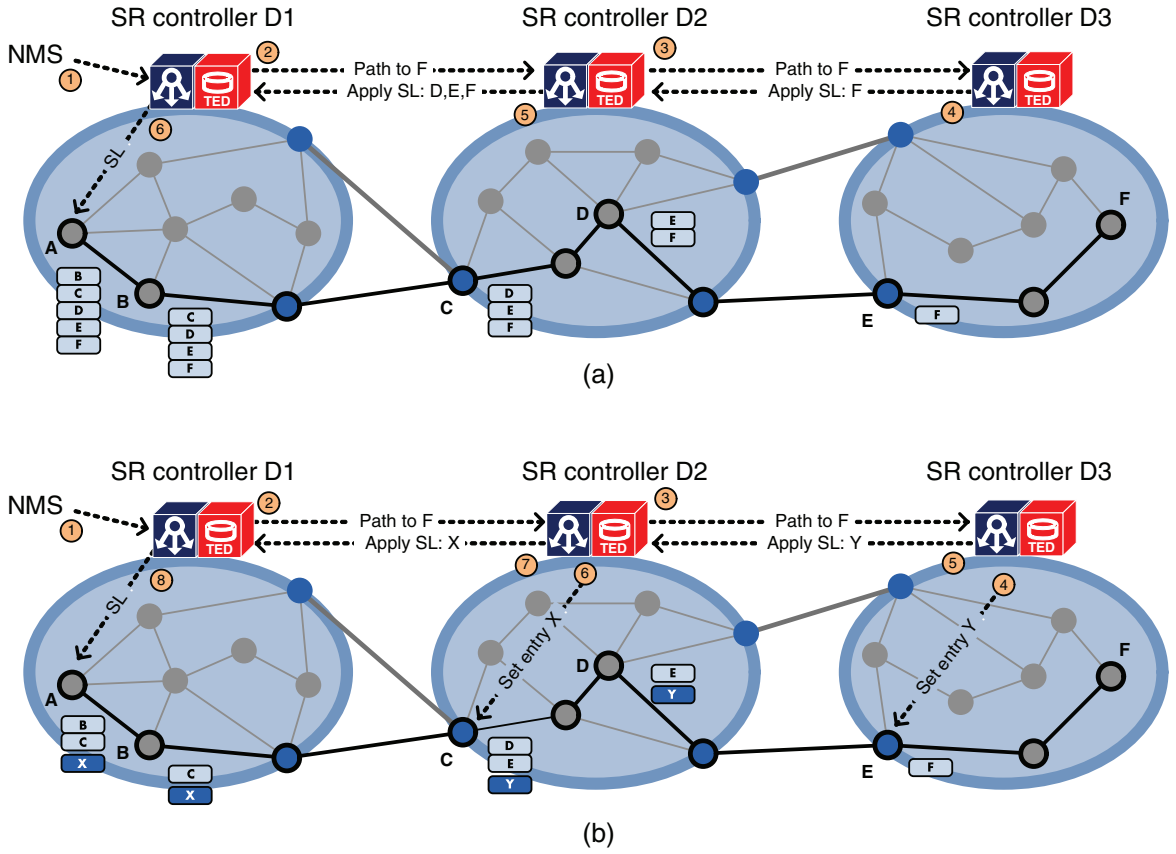


Fig. 3. Multi-domain segment routing: (a) e2e-SR procedure scheme; (b) pd-SR procedure scheme.

and 55 bidirectional links, and a set of 100 topologies generated with BRITE [46], each one including 150 nodes and 300 bidirectional links. All possible single-link and single-node failures are considered in the evaluated networks. The SLD is computed for a wide set of backup paths with the SR-FAILOVER scheme and, for comparison, with the scheme SR-DETOUR described in Ref. [41], where backup paths are computed from the point of failure to the node identified by the next label in the segment list. This comparison is therefore useful for understanding the benefit provided by SR-FAILOVER recovering the traffic directly to the destination node. A single backup path is computed for each failure that can affect every shortest path in the network. This way, in the case of link failure, 6332 and more than 20 million backup paths are respectively considered for the pan-European and BRITE topologies; in the case of node failure, a slightly smaller number of backup paths is considered because, if the failure affects the destination node, it is not possible to recover the traffic.

Figure 4 shows the obtained distributions of the SLD. In the pan-European topology [Figs. 4(a) and 4(b)], the SR-FAILOVER scheme achieves an average SLD of 1.45 and 1.68 in the cases of link and node failure, respectively. Thus, redirecting the traffic directly to the destination, the results achieved by the SR-DETOUR scheme [41] (i.e., average SLD of 2.01 and 2.53) are improved by about 30%. With the BRITE-generated networks [Figs. 4(c) and 4(d)], the average SLD achieved by the SR-FAILOVER scheme is 1.65 and 1.61 with respect to 2.19 and 2.10 obtained using the SR-DETOUR scheme [41], and thus the improvement is about 25%. It is also shown in both scenarios that, when using the SR-FAILOVER scheme, 90% of the backup paths use a segment list of 1 or 2 labels, whereas this percentage varies from 51% and 74% using the SR-DETOUR scheme.

The SR-FAILOVER scheme has been implemented in an experimental test bed using a SDN controller and five OpenFlow switches. In the switches, the primary forwarding table uses the *Group Table* OpenFlow feature to enable the monitoring of the interface status and support a backup list of actions to be applied when the primary forwarding interface is down. Nodes have been implemented within Intel Core 4 servers (CPU 2.40 GHz, Linux kernel 3.13) equipped with 4 Gb/s Ethernet interfaces and running Open vSwitch version 2.4.0, supporting MPLS-based forwarding. OpenFlow 1.3 has been utilized for the communication between the nodes and the controller. The controller has been implemented on a dedicated server by extending the SDN Ryu controller [47]. Commercially available reconfigurable optical add-drop multiplexers (ROADMs) equipped with 10 Gb/s OTN muxponders have been connected to nodes for implementing the optical transport plane of the multi-layer network.

The aforementioned hardware has been utilized to realize the network topology represented in Fig. 2(b), where the

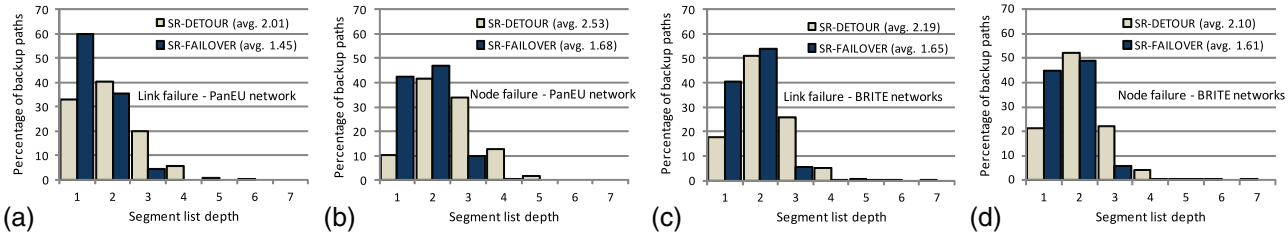


Fig. 4. Statistical distribution, percentage of backup paths encoded with a specific segment list depth: (a) link failure in pan-European topology; (b) node failure in pan-European topology; (c) link failure in 100 BRITE-generated topologies; (d) node failure in 100 BRITE-generated topologies.

two traffic flows along paths $\bar{p}_1$ and $\bar{p}_2$ are established. Figure 5(a) illustrates the test bed topology visualized by the web-based Ryu Topology Viewer. The datapath ID (i.e., *dpid*) of each node is used as a node segment identifier in the enforced segment lists. The *ping* application is used from host *A* (i.e., IP address 10.0.34.1) to both destination hosts *B* and *C* (i.e., IP addresses 10.0.35.1 and 10.0.38.1). Initially, no failure is present on the network. The two traffic flows are steered using the segment list as described in Fig. 2(b). Specifically, the Wireshark capture in Fig. 5(b) is executed on port 2 of the node with dpid 1002; it shows that packets directed to host *B* use a segment list composed of a single label, i.e., 1004. Then a failure is generated by physically disconnecting the cable from port 4 of the node with dpid 1004. The failure holds from time 12:58:05 to time 12:58:21; during this period no packets are routed on this interface. The capture in Fig. 5(c) is executed on port 3 of the node with dpid 1002; it shows that, during the failure, traffic to host *B* uses this port and includes a segment list composed of two labels, i.e., 1001, 1004.

To accurately evaluate the recovery time required by the SR-FAILOVER scheme, the aforementioned failure has been repeated more than 200 times. The *ping* has been used to generate a packet every 2 ms, so that the traffic hole generated by the occurrence of the failure can be easily measured by parsing the Wireshark capture of the packets by host *B*. Thus the measured traffic hole accounts for all the contributions of the recovery time: physical detection time, time to match the flow entry in the failover table, and increased latency along the backup path. Among these contributions, the last one is negligible in our test bed implementation and it is typically limited to a few milliseconds in core networks. With the described procedure, recovery time is measured with an error of ±2 ms. Figure 6 shows the distribution of the obtained recovery time (243 samples) where the average value is 13.1 ms.

B. Segment Routing in Multi-domain Networks

The scalability of the proposed e2e-SR and pd-SR schemes has been evaluated in terms of SLD in a simulated environment. The reference multi-domain topology reported in Ref. [24] is considered including 75 nodes, 292 links, and

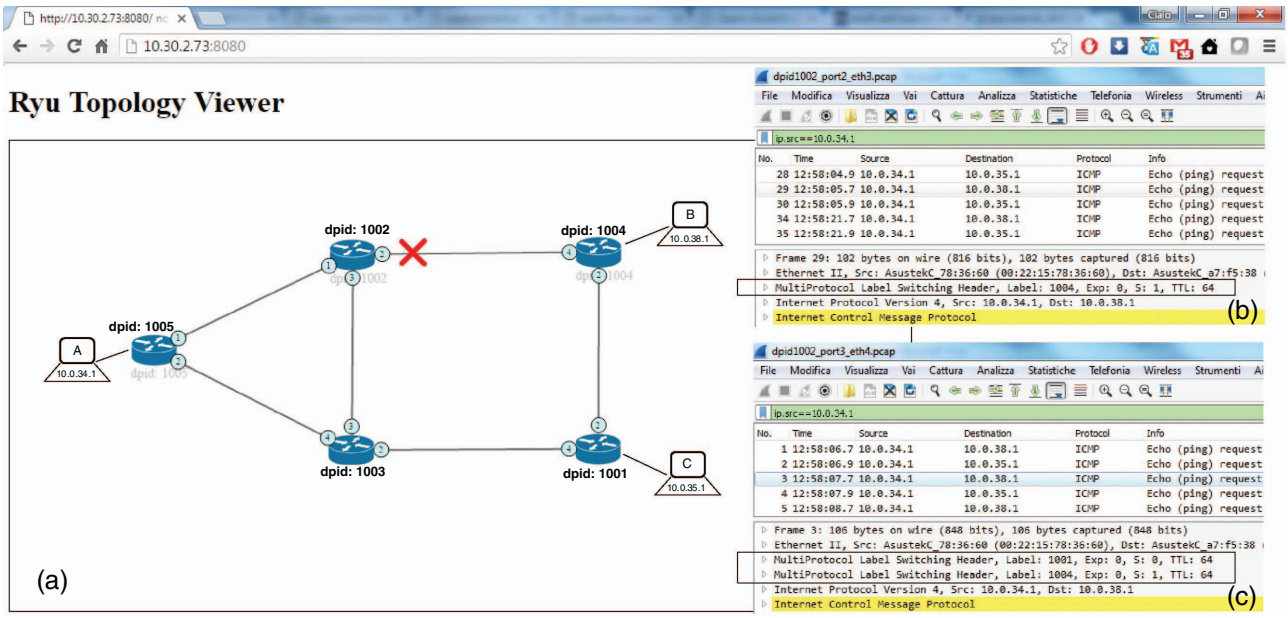


Fig. 5. SR-FAILOVER scheme test bed: (a) network topology visualized through the Ryu Topology Viewer web-based script; (b) Wireshark capture on port 2 of the node with dpid 1002; (c) Wireshark capture on port 3 of the node with dpid 1002.

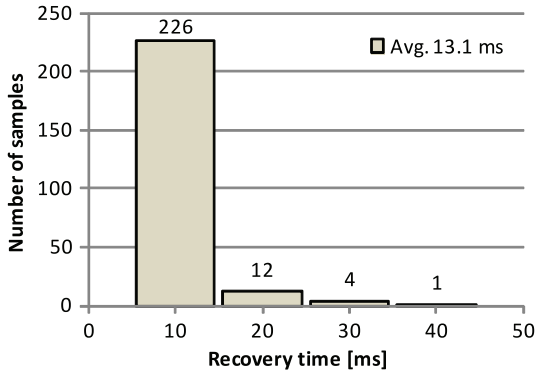


Fig. 6. Statistical distribution: recovery time [ms] measured in the experimental test bed.

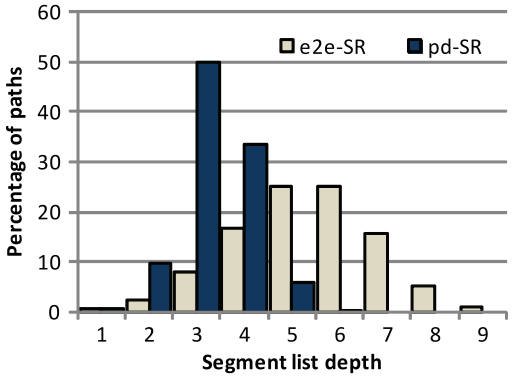


Fig. 7. Statistical distribution: maximum segment list depth values for the two schemes considering the multi-domain network topology illustrated in Ref. [24].

9 domains. The SLD is evaluated for all the paths within one hop from the shortest path for all the node pairs in the network, thus considering a total of more than 137,000 paths. For each given path, the SR controller of each domain uses the algorithm in Ref. [37] to compute the encoding segment list.

Figure 7 illustrates the achieved distribution of the maximum SLD for all the considered paths. Using the e2e-SR scheme, the maximum SLD along a given path is always registered at the source node; conversely, using the pd-SR scheme, the maximum SLD can be registered in any of the traversed border nodes that perform the mapping of virtual labels. The figure shows that the pd-SR scheme is able to encode 60% of the paths with a segment list composed of 1, 2, or 3 labels, while e2e-SR encodes less than 12% of the paths within the three labels. The achieved SLD average values are 5.34 and 3.36 for e2e-SR and pd-SR, respectively. In the example represented in Fig. 3, the e2e-SR scheme provides a maximum depth of 5 (applied at source node A), whereas the pd-SR scheme provides a maximum depth of 3 (applied at node A and at the border node of domain $D2$).

To validate the proposed pd-SR scheme, the same experimental test bed just described has been utilized but in the configuration illustrated in Fig. 8. The network includes six Open vSwitch nodes divided into two domains. Each domain includes three nodes; moreover, two optical nodes are used to implement the inter-domain link that is transparent to SR operation. SR controllers based on SDN Ryu [47] have been extended to support the proposed multi-domain schemes.

Each node in Fig. 8 is shown with the specific forwarding table, including the relevant entries. The labels are obtained from the node id adding the prefix 1000; the virtual label is randomly generated among unused labels. A traffic flow is configured from host $H2$ to host $H1$ with the pd-SR scheme. Specifically, node 5 is configured by the domain 2 controller as the ingress node, including the entry toward $H1$, with three associated actions: pushing of the assigned virtual label (push 1000632); pushing of the label of node 2, which is the ingress border node of domain 1 (push 10002); and forwarding of the packets along port 1 (out 1). The ingress border of domain 1 (node 2) is configured by the domain 1 controller with one entry matching the virtual label, where the associated actions are popping of the virtual label (pop), pushing of the node 1 label (push 10001), and forwarding of the packet along port 1 (out 1). The setup procedure has been performed 200 times with average setup time of 3.4 ms.

Figure 9 shows the Wireshark capture of the OpenFlow `OFPT_FLOW_MOD` message sent by the controller of domain 2 to configure the ingress node 5. The insets in Fig. 9 highlight the message fields containing the two used labels (i.e., the virtual label 1000632 and the label 10002 of node 2).

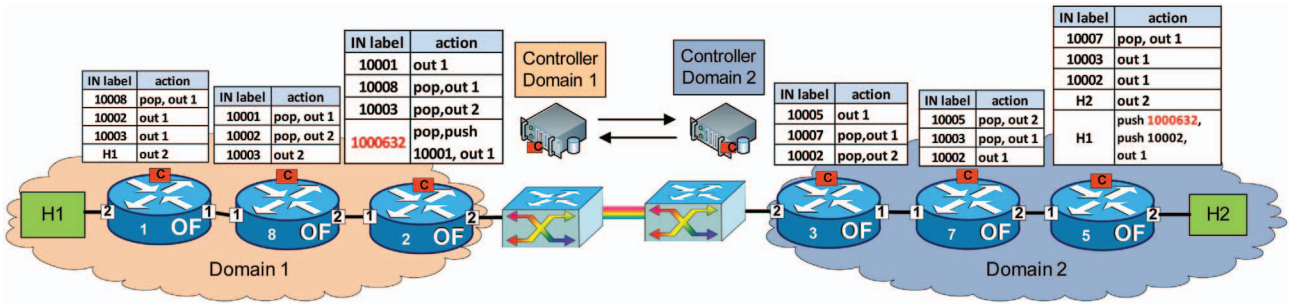


Fig. 8. Experimental validation of the pd-SR scheme.

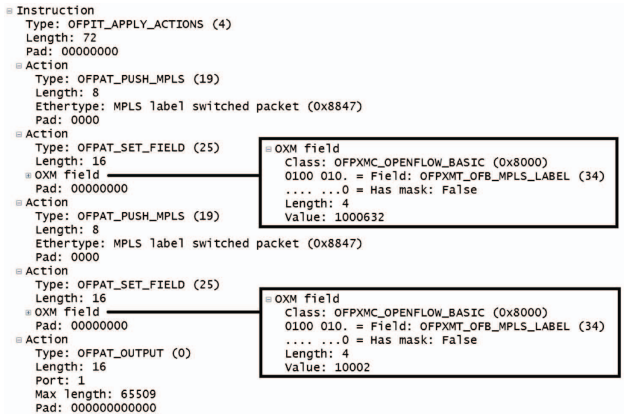


Fig. 9. Wireshark capture of the OFPT_FLOW_MOD message sent from the controller of domain *D*2 to node 5.

VII. Conclusion

Traffic recovery and multi-domain traffic engineering schemes based on segment routing have been proposed and demonstrated in a multi-layer SDN network.

Regarding the traffic recovery, simulation results showed that, using the SR-FAILOVER scheme, the wide majority of backup paths can be encoded with a segment list of one or two labels. Experimental measurements reported an average recovery time of 13 ms. Such good performance confirms that segment routing is a strong candidate technology for effective traffic recovery. Moreover, given the absence of signaling sessions, it enables the scalable implementation of failover up to the destination node, which is well known to be more effective in resource utilization with respect to schemes exploiting detour toward the next-next hop, as deployed in today's MPLS networks through the fast-reroute mechanism.

Regarding multi-domain traffic engineering, the two considered schemes have been compared in terms of provided segment list depth. Simulation results demonstrated that the pd-SR scheme is able to significantly reduce the segment list depth with respect to the e2e-SR scheme. Experimental validation of the pd-SR scheme showed a configuration time of less than 4 ms for establishing a new inter-domain traffic flow. Also in this case, such good performance confirms that segment routing can be effectively adopted for multi-domain traffic engineering. Indeed, given the absence of signaling sessions between border nodes belonging to different domains, it enables the scalable concatenation of multi-domain paths, overcoming the limitations and inter-operability issues of RSVP-TE stitching and nesting solutions, rarely deployed in practical scenarios.

Acknowledgment

Portions of this work were presented in Refs. [42,43]. This work was partially funded by EU H2020-ICT-2014 Project 5GEx (grant no. 671636).

References

[1] C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, "Segment routing architecture," draft-filsfils-spring-segment-routing-09, July 2016 [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-09.

[2] M. Vigoureux, B. Berde, L. Andersson, T. Cinkler, L. Levrau, M. Ondata, D. Colle, J. Fernandez-Palacios, and M. Jager, "Multilayer traffic engineering for GMPLS-enabled networks," *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 44–50, 2005.

[3] F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti, and P. Castoldi, "Efficient label encoding in segment-routing enabled optical networks," in *IEEE Int. Conf. on Optical Network Design and Modeling (ONDM)*, May 2015.

[4] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The segment routing architecture," in *Proc. GLOBECOM*, Dec. 2015.

[5] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu, "Research directions in network service chaining," in *Proc. Software Defined Networking for Future Networks and Services (SDN4FNS)*, Trento, Italy, Nov. 2013.

[6] N. Guilbaud and R. Cartlidge, "Localizing packet loss in a large and complex network," in *Proc. NANOG 57*, Feb. 2013.

[7] D. Wang and G. Li, "Efficient distributed bandwidth management for MPLS fast reroute," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 486–495, 2008.

[8] P. Pan, G. Swallow, and A. Atlas, "Fast reroute extensions to RSVP-TE for LSP tunnels," IETF RFC 4090, May 2005.

[9] A. Banerjee, J. Drake, J. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label switching: An overview of signaling enhancements and recovery techniques," *IEEE Commun. Mag.*, vol. 39, no. 7, pp. 144–151, 2001.

[10] J. Perello, S. Spadaro, F. Agraz, M. Angelou, S. Azodolmolky, Y. Qin, R. Nejabati, D. Simeonidou, P. Kokkinos, E. Varvarigos, and I. Tomkos, "Experimental demonstration of a GMPLS-enabled impairment-aware lightpath restoration scheme," *J. Opt. Commun. Netw.*, vol. 4, no. 5, pp. 344–355, 2012.

[11] K. Lu, G. Xiao, and I. Chlamtac, "Analysis of blocking probability for distributed lightpath establishment in WDM optical networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 187–197, 2005.

[12] A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Dynamic restoration with GMPLS and SDN control plane in elastic optical networks [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 2, pp. A174–A182, 2015.

[13] A. Aguado, M. Davis, S. Peng, M. V. Álvarez, V. López, T. Szyrkowiec, A. Autenrieth, R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, R. Martínez, N. Yoshikane, T. Tsuritani, R. Nejabati, and D. Simeonidou, "Dynamic virtual network reconfiguration over SDN orchestrated multitechnology optical transport domains," *J. Lightwave Technol.*, vol. 34, no. 8, pp. 1933–1938, 2016.

[14] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Enabling fast failure recovery in OpenFlow networks," in *IEEE 8th Int. Workshop on the Design of Reliable Communications Networks (DRCN)*, Oct. 2011.

[15] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takacs, and P. Sköldström, "Scalable fault management for OpenFlow," in *IEEE Int. Conf. on Communications (ICC)*, June 2011.

[16] S. S. W. Lee, K. Y. Li, K. Y. Chan, G. H. Lai, and Y. C. Chung, "Software-based fast failure recovery for resilient OpenFlow

networks," in *7th Int. Workshop on Reliable Networks Design and Modeling (RNDM)*, Oct. 2015.

[17] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "OpenFlow: Meeting carrier-grade recovery requirements," *Comput. Commun.*, vol. 36, no. 6, pp. 656–665, 2013.

[18] A. Sgambelluri, A. Giorgetti, F. Cugini, F. Paolucci, and P. Castoldi, "OpenFlow-based segment protection in Ethernet networks," *J. Opt. Commun. Netw.*, vol. 5, no. 9, pp. 1066–1075, 2013.

[19] A. Manolova and S. Ruepp, "Export policies for multi-domain WDM networks," in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, Mar. 2010.

[20] A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Hierarchical PCE in GMPLS-based multi-domain wavelength switched optical networks," in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, Mar. 2011.

[21] F. Paolucci, F. Cugini, L. Valcarenghi, and P. Castoldi, "Enhancing backward recursive PCE-based computation (BRPC) for inter-domain protected LSP provisioning," in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, Feb. 2008.

[22] D. Siracusa, S. Grita, G. Maier, A. Pattavina, F. Paolucci, F. Cugini, and P. Castoldi, "Domain sequence protocol (DSP) for PCE-based multi-domain traffic engineering," *J. Opt. Commun. Netw.*, vol. 4, no. 11, pp. 876–884, 2012.

[23] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray, "North-bound distribution of link-state and traffic engineering (TE) information using BGP," IETF RFC 7752, Mar. 2016.

[24] A. Giorgetti, "Proactive H-PCE architecture with BGP-LS update for multidomain elastic optical networks [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 11, pp. B1–B9, 2015.

[25] O. Gonzalez de Dios, R. Casellas, R. Morro, F. Paolucci, V. Lopez, R. Martinez, R. Muñoz, R. Vilalta, and P. Castoldi, "First multi-partner demonstration of BGP-LS enabled inter-domain EON control with H-PCE," in *Optical Fiber Communication Conf. (OFC)*, 2015, paper Th1A.4.

[26] L. Liu, "SDN orchestration for dynamic end-to-end control of data center multi-domain optical networking," *China Commun.*, vol. 12, no. 8, pp. 10–21, 2015.

[27] Y. Yu, J. Zhang, Y. Zhao, Y. Lin, J. Han, H. Zheng, Y. Cui, M. Xiao, H. Li, Y. Peng, Y. Ji, and H. Yang, "Field demonstration of multi-domain software-defined transport networking with multi-controller collaboration for data center interconnection [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 2, pp. A301–A308, 2015.

[28] Z. Zhu, C. Chen, X. Chen, S. Ma, L. Liu, X. Feng, and S. J. B. Yoo, "Demonstration of cooperative resource allocation in an OpenFlow-controlled multidomain and multinational SD-EON testbed," *J. Lightwave Technol.*, vol. 33, no. 8, pp. 1508–1514, 2015.

[29] P. Francois, S. Previdi, B. Decraene, and R. Shakir, "Resiliency use cases in SPRING networks," draft-ietf-spring-resiliency-use-cases-08, Oct. 2016 [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-resiliency-use-cases-08.

[30] D. Cai, A. Wielosz, and S. Wei, "Evolve carrier Ethernet architecture with SDN and segment routing," in *IEEE 15th Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2014.

[31] S. Bidkar, A. Gumaste, P. Ghodasara, A. Kushwaha, J. Wang, and A. Somani, "Scalable segment routing: A new paradigm for efficient service provider networking using carrier Ethernet advances," *J. Opt. Commun. Netw.*, vol. 7, no. 5, pp. 445–460, 2015.

[32] A. Giorgetti, P. Castoldi, F. Cugini, J. Nijhof, F. Lazzeri, and G. Bruno, "Path encoding in segment routing," in *Proc. GLOBECOM*, Dec. 2015.

[33] A. Cianfrani, M. Listanti, and M. Polverini, "Translating traffic engineering outcome into segment routing paths: The encoding problem," in *IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2016.

[34] R. Hartert, S. Vissicchio, P. Schaus, O. Bonaventure, C. Filsfils, T. Telkamp, and P. Francois, "A declarative and expressive approach to control forwarding paths in carrier-grade networks," in *Proc. SIGCOMM*, Aug. 2015.

[35] R. Bhatia, F. Hao, M. Kodialam, and T. V. Lakshman, "Optimized network traffic engineering using segment routing," in *IEEE Conf. on Computer Communications (INFOCOM)*, Apr. 2015.

[36] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, "Traffic engineering with segment routing: SDN-based architectural design and open source implementation," in *Proc. EWSDN*, Sept. 2015.

[37] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, and P. Castoldi, "Experimental demonstration of segment routing," *J. Lightwave Technol.*, vol. 34, no. 1, pp. 205–212, 2016.

[38] A. Gumaste, S. Bidkar, P. Ghodasara, S. Hote, A. Kushwaha, R. Ambasta, and P. Agrawal, "Demonstrating a software defined network (SDN) using carrier Ethernet switch routers in a provider network," in *Optical Fiber Communication Conf. and Exhibition (OFC)*, Mar. 2015.

[39] A. Fressancourt and M. Gagnaire, "A SDN-based network architecture for cloud resiliency," in *12th Annu. IEEE Consumer Communications Networking Conf. (CCNC)*, Jan. 2015.

[40] F. Hao, M. Kodialam, and T. V. Lakshman, "Optimizing restoration with segment routing," in *IEEE 35th Annu. IEEE Int. Conf. on Computer Communications*, Apr. 2016.

[41] A. Giorgetti, A. Sgambelluri, F. Paolucci, and P. Castoldi, "Reliable segment routing," in *7th Int. Workshop on Reliable Networks Design and Modeling (RNDM)*, Oct. 2015.

[42] A. Sgambelluri, A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Experimental demonstration of multi-domain segment routing," in *European Conf. on Optical Communication (ECOC)*, Sept. 2015.

[43] A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini, and P. Castoldi, "Demonstration of dynamic restoration in segment routing multi-layer SDN networks," in *Optical Fiber Communication Conf. (OFC)*, Mar. 2016.

[44] G. Swallow, "From tag switching to SDN and segment routing MPLS: An enduring architecture," in *Proc. MPLS SDN World Congr.*, Mar. 2014.

[45] P. Psenak, S. Previdi, C. Filsfils, H. Gredler, R. Shakir, W. Henderickx, and J. Tantsuraand, "OSPF extensions for segment routing," draft-psenak-ospf-segment-routing-extensions-10, Oct. 2016 [Online]. Available: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-10.

[46] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proc. MASCOTS*, Aug. 2001.

[47] Ryu controller [Online]. Available: http://osrg.github.io/ryu/.