Segment Routing in Multi-Layer Networks

P. Castoldi[1], A. Giorgetti[1], A. Sgambelluri[1], F. Paolucci[1], F. Cugini[2]
1: Scuola Superiore Sant'Anna, Pisa, Italy
2: CNIT, Pisa, Italy
e-mail: piero.castoldi@santannapisa.it

ABSTRACT

Segment Routing (SR) has been recently introduced to enable efficient traffic engineering while simplifying control plane operations. Thanks to the source routing paradigm, traffic flows can be dynamically routed along the network, effectively exploiting network resources. In this paper, dynamic SR operations for multi-layer networking are presented and experimentally demonstrated. In particular, SR-based dynamic optical bypass and effective load balancing are validated in a multi-layer network testbed, showing enhanced capabilities to achieve effective resource utilization while guaranteeing lightweight control operations.

Keywords: Segment Routing, Multi-layer, SDN, load balancing.

1. INTRODUCTION

The Segment Routing (SR) technology has been recently introduced to provide effective traffic engineering (TE) while simplifying control plane operation [1-4]. SR relies on the source-routing paradigm: a specifically designed header, composed of a stack of multi-protocol label-switched (MPLS) labels (i.e., the segment list) is enforced at the ingress node so that the traffic flows are routed through the desired path. At transit nodes, packets are forwarded along the shortest path toward the node represented by the top label in the segment list. This way, transit nodes can avoid a signalling protocol and the maintenance of flow entries, significantly simplifying control plane operations [5-9]. In case of equal cost multiple paths (ECMP), SR by default exploits all available routes, performing per-flow load balancing among the available paths [1**,** 10]. This enables automatic and effective exploitation of the network resource. However, in some networking scenarios, e.g. upon failure occurrence in multi-layer networks, it may be beneficial to enable load balancing among non ECMP routes. In this paper, we implement and experimentally demonstrate a SR-based Software Defined Network (SDN) solution enabling load balancing among non-ECMP routes in multi-layer networks including an IP/MPLS layer over an Elastic Optical Network (EON) layer. This way, dynamic and effective traffic engineering is performed, successfully exploiting available network resources also in case of non-ECMP routes.

2. PREVIOUS WORKS ON SEGMENT ROUTING

Segment Routing standardization is rapidly evolving within IETF [1, 3] and relevant research work has been conducted within the academic community. Authors of [11] proposed to combine the benefits of SR with those of a SDN control plane. The work in [9] implemented SR in carrier grade Ethernet networks including a detailed simulation studies. Algorithms to compute the segment list encoding a given path are proposed in [2, 12, 13]. Specifically, [2] and [13] propose the utilization of an auxiliary graph model representing the available network segments for computing the segment list, whereas the work in [12] proposes a greedy algorithm to compute the segment list of minimum depth. The works in [14, 15] formulate a multi-commodity flow problem to evaluate the benefits of SR. The work in [14] 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 [15] shows that using segment list composed of only two labels, SR is able to provide significant benefits with respect to shortest path routing. Then, several works including [5, 7, 8, 9, 16] detail experimental implementations and evaluations of the SR architecture. Finally, few works propose effective recovery techniques using SR [7, 17, 18].

In this work we focus on the utilization of SR concepts in multi-layer networks (e.g. IP/MPLS over EON). In particular, this work presents possible issues that may arise applying SR in multi-layer network and proposes a possible solution.

3. SEGMENT ROUTING (SR) OPERATIONS IN MULTI-LAYER NETWORKS

Fig. 1 shows a portion of a multi-layer core network scenario, composed of packet and optical nodes. Optical nodes provide the transport technology. Traffic engineering is performed at the MPLS packet level by enforcing the proper route. In particular, 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 packet node to enforce the computed segment list.

If a request arrives from node A to node F, the controller selects between two possible routes: the route A-B-C-D-E-F traversing the full sequence of packet nodes and the route A-B-E-F exploiting optical bypass (i.e., avoiding packet nodes C and D). In the context of SR, since the latter route results of shortest cost, just one label indicating node F (i.e., Segment Identifier SID F) can be enforced at node A to route the flow through the optical bypass (i.e., by node B through the adjacency B-E). Alternatively, to specifically select the full sequence of packet nodes, a segment list composed of two SIDs (i.e., labels) is required, having node C as top label and node F as bottom one. During packet forwarding, at node C, label C is popped, leaving F as the new top label along the rest of the path. Note that the selection of either paths can be performed in an extremely dynamic way by simply enforcing to each incoming packet at node A the proper segment list, without requiring complex and time consuming control plane operations [4, 5, 12].

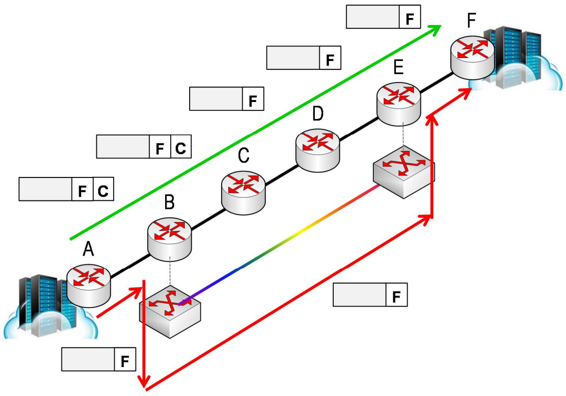


Fig. 1: Segment routing for dynamic optical bypass

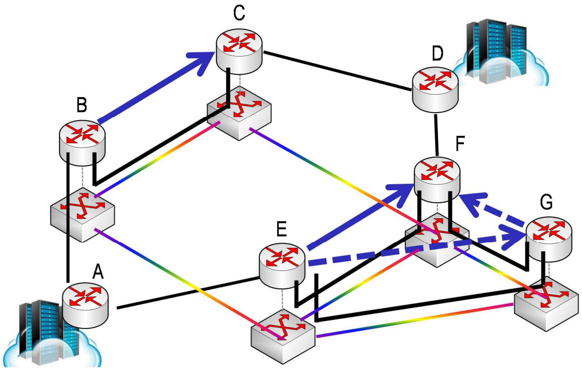


Fig. 2: reference network scenario

4. ECMP MULTI-LAYER SCENARIO

Besides dynamic optical bypass operations, the SR technology can be efficiently adopted to address a typical traffic engineering issue that affects core networks (employing either SR or traditional MPLS). Fig. 2 shows a second network scenario reproducing a typical dual-homed connectivity of node A to the transport network. For reliability purposes, two ECMP routes are activated between nodes A and D. Using either MPLS or SR the two routes can be simultaneously exploited applying per-flow load balancing. Specifically, in the SR case, node A enforces all the traffic directed to node D with a segment list including only the label of the destination node (i.e., D), then the SR agent will automatically apply load balancing on the two available ECMP [1,3].

In case of failures on the optical layer, e.g. of link E-F, route A-B-C-D becomes the unique shortest one and it is fully exploited by all A-D traffic flows. However, according to the recovery scheme, two cases may occur. In the first case, recovery is performed at the optical layer only (i.e., A-F through ROADM G). In the second case, multi-layer recovery takes place exploiting already established adjacencies at the packet level (i.e., E-G and G-F through MPLS fast reroute). Since optical restoration may take seconds, most network operators prefer to rely on the latter scheme which guarantees fast reaction time (e.g., few tens of milliseconds). However, in a dual homed scenario as the one in Fig. 2, the recovered E-G-F connection is not actually exploited by A-D traffic since it becomes at higher cost.

5. SR FOR LOAD BALANCING OVER NON-ECMP

In the scenario depicted in Fig. 2 it is desirable to continue load-balancing on the two alternate routes also if one of the two routes has higher cost.

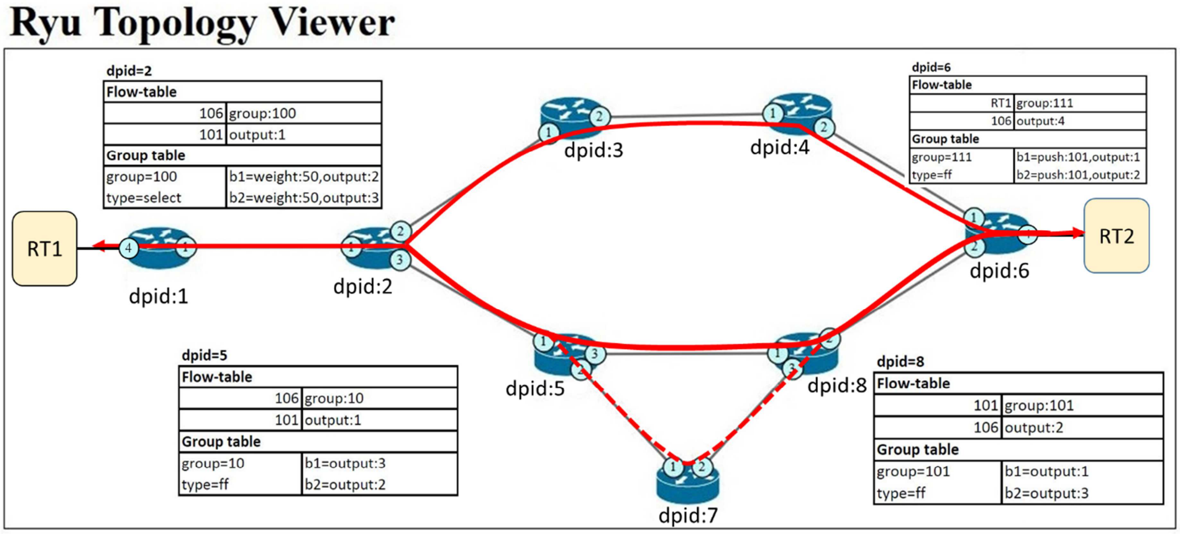
Using SR we propose two different solutions to guarantee the enforcement of load-balancing after the failure. The first solution, named CENTRALIZED-SR, involves the SDN controller upon failure occurrence, thus it is very flexible but not particularly fast. In the second solution, named PRECONFIGURED-SR, the data plane is pre-configured and the controller is not involved upon failure occurrence. This latter solution is less flexible but guarantees a faster traffic recovery.

Specifically, using the CENTRALIZED-SR scheme, when the failure is detected, the controller enforces the utilization of a new segment list to a subset of established traffic flows. This way, the flows selected to be routed along the path *A-E-G-F-B*, will be sent out toward node *E* using a segment list composed of two labels, e.g., *G-D*. This procedure is much simpler and faster than establishing a new LSP. Indeed it requires only a communication between the controller and the source node *A*, and does not involve optical layer re-configuration.

Using the PRECONFIGURED-SR the data plane is pre-configured so that also in case of failure the source node continues to effectively perform load balancing. This solution is implemented utilizing the *group table* functionality provided by the OpenFlow protocol. Instead of using a simple flow entry at the source node *A* applying the segment list composed only by the label *D*, the flow entry delegates the traffic treatment to a group entry of type *select* [19]. This type of group enforces an explicit load balancing among a set of buckets by applying local policies depending on the data plane implementation (e.g., Open vSwitch implements round-robin per-flow load balancing). Each bucket contains a list of actions. In the example in Fig. 2, the select group contains two buckets. The first bucket enforces the actions: push MPLS label *D*, output toward *B*. The second bucket enforces the actions: push MPLS label *D*, output toward *E*. In this way, also when the failure occurs, node *A* continues to apply load-balancing on the two buckets, thus node *E* continues to receive traffic with the segment list *D*. This traffic will be correctly forwarded on the path *G-F-D* because, starting from node *E*, this path is the unique shortest path to *D*. Depending on the specific network topology the two buckets may enforces different segment lists. Thus, with PRECONFIGURED-SR solution the load-balancing of the traffic can be easily applied also on NON-ECMP routes. Moreover, this solution is easily extendable to perform load-balancing on a generic number of routes [19].

6. EXPERIMENTAL VALIDATION

In order to evaluate the functional behaviour and the performance of the PRECONFIGURED-SR solution, a testbed has been prepared. The experimental testbed is composed by 1 OpenFlow Ryu SR controller version 4.11, extended with new SR functionalities, 8 OpenFlow enabled switch, running Open vSwitch software version 2.4, able to process MPLS labels, 1 traffic generator/analyser in order to generate streams of probing traffic. In *Fig.* ***3*** the network topology view exposed by the Ryu GUI is shown. RT1 and RT2 are the interfaces of the traffic generator/analyser connected respectively to port 4 of switch 1 and 4 of switch 6.



dpid=2

Flow-table	
106	group:100
101	output:1
Group table	
group=100 type=select	b1=weight:50,output:2 b2=weight:50,output:3

dpid=6

Flow-table	
RT1	group:111
106	output:4
Group table	
group=111 type=ff	b1=push:101,output:1 b2=push:101,output:2

dpid=5

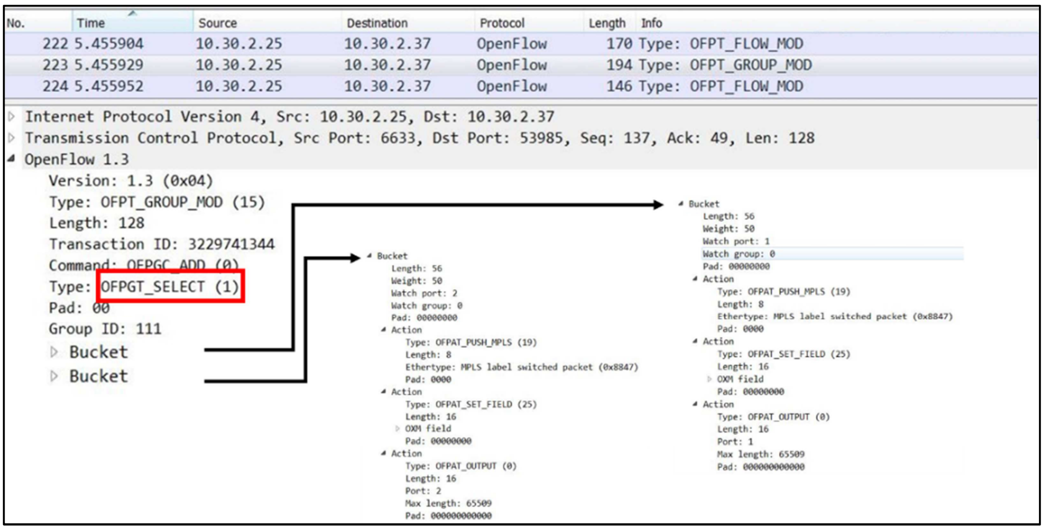
Flow-table	
106	group:10
101	output:1
Group table	
group=10 type=ff	b1=output:3 b2=output:2

dpid=8

Flow-table	
101	group:101
106	output:2
Group table	
group=101 type=ff	b1=output:1 b2=output:3

Fig. 3: Ryu network topology, with OPenFlow flow-tables of switches 2, 5, 6 8

In red solid lines, the ECMP between source and destination (i,e, RT1 and RT2) are highlighted. With dashed red line the backup path, used in case of failure on the link 5-8, is shown. Considering the switch 6, when it receives packets with destination RT1, it applies the rule present in the group 111, pushing the MPLS label 101 and performing the load balance among the two available ECMP (i.e., *4-3-2-1*, *8-5-2-1*). In *Fig.* ***4*** the capture of the OpenFlow messages required to configure the switch 6 is shown.



```
No.   Time        Source       Destination  Protocol  Length Info
  222 5.455904    10.30.2.25   10.30.2.37   OpenFlow     170 Type: OFPT_FLOW_MOD
  223 5.455929    10.30.2.25   10.30.2.37   OpenFlow     194 Type: OFPT_GROUP_MOD
  224 5.455952    10.30.2.25   10.30.2.37   OpenFlow     146 Type: OFPT_FLOW_MOD

▷ Internet Protocol Version 4, Src: 10.30.2.25, Dst: 10.30.2.37
▷ Transmission Control Protocol, Src Port: 6633, Dst Port: 53985, Seq: 137, Ack: 49, Len: 128
◢ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_GROUP_MOD (15)
    Length: 128
    Transaction ID: 3229741344
    Command: OFPGC_ADD (0)
    Type: OFPGT_SELECT (1)
    Pad: 00
    Group ID: 111
  ▷ Bucket
  ▷ Bucket
```

```
◢ Bucket
    Length: 56
    Weight: 50
    Watch port: 2
    Watch group: 0
    Pad: 00000000
  ◢ Action
      Type: OFPAT_PUSH_MPLS (19)
      Length: 8
      Ethertype: MPLS label switched packet (0x8847)
      Pad: 0000
  ◢ Action
      Type: OFPAT_SET_FIELD (25)
      Length: 16
    ▷ OXM field
      Pad: 00000000
  ◢ Action
      Type: OFPAT_OUTPUT (0)
      Length: 16
      Port: 2
      Max length: 65509
      Pad: 000000000000
```

```
◢ Bucket
    Length: 56
    Weight: 50
    Watch port: 1
    Watch group: 0
    Pad: 00000000
  ◢ Action
      Type: OFPAT_PUSH_MPLS (19)
      Length: 8
      Ethertype: MPLS label switched packet (0x8847)
      Pad: 0000
  ◢ Action
      Type: OFPAT_SET_FIELD (25)
      Length: 16
    ▷ OXM field
      Pad: 00000000
  ◢ Action
      Type: OFPAT_OUTPUT (0)
      Length: 16
      Port: 1
      Max length: 65509
      Pad: 000000000000
```

Fig. 4: OpenFlow messages capture with the highlight of the OFPT_GROUP_MOD message.

More specifically, 2 OFPT_FLOW_MOD messages are sent in order to configure the 2 required flow-entries (i.e., dst=RT1, group:111 and MPLS=106,output:4). While, 1 OFPT_GROUP_MOD is used in order to configure the OpenFlow group. *Fig.* ***4*** shows the details related to the OFPT_GROUP_MOD message. In particular, the group type is *select* (highlighted in red), while the parameters of the two buckets used to process the packets along the two ECMP are exploded. Each bucket is assigned a weight of 50 (i.e., equal balancing among the two buckets) and includes three actions: pushing of the MPLS header, configuring the MPLS label to the value 101 (required to reach the switch connected to the destination), sending to the proper output port the packet. We tested the behaviour of the system by configuring 20 bidirectional streams of traffic between RT1 and RT2, with packet rate of 1000 packet/sec. The load balancing, based on the hash function on the header of the packets, has distributed the traffic flows among the 2 ECMP. No packet loss or packet reordering has been detected. Moreover, producing a failure on the link *5-8*, we verified that the load balancing is maintained also in the case of non-ECMP. In fact, thanks to the fast failover group configured at the switches 5 and 8, adjacent to the link failure, the traffic is recovered passing through switch 7 without involving the SR controller. Only the traffic belonging to the streams passing through the path *1-2-5-8-6* are affected by the recovery. Some packet loss has been detected, with a recovery time around 170ms dependent on the failure detection time. No corrupted and reordered packets have been verified.

7. CONCLUSIONS

The Segment Routing technology has the potential to simplify and automate provisioning and recovery operations in multi-layer packet over optical networks. In this paper, two scenarios for dynamic segment routing operations in multi-layer networks are presented and experimentally demonstrated: dynamic optical bypass and effective load balancing also among non-ECMP routes. In both scenarios, segment routing proved to be effective in guaranteeing efficient resource utilization with lightweight control operations.

ACKNOWLEDGEMENTS

This work has been partially supported by the H2020-ICT-2014 project 5GEx (Grant Agreement no. 671636), which is partially funded by the European Commission.

REFERENCES

[1] C. Filsfils, et al., “ Segment routing architecture” draft-filsfils-spring-segment-routing-11, Feb 2017

[2] F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti, P. Castoldi, “Efficient label encoding in segment-routing enabled optical networks, ” in ONDM Conf., May 2015.

[3] C. Filsfils, “The segment routing architecture” in Proc. GLOBECOM, Dec. 2015.

[4] A Sgambelluri, A Giorgetti, F Cugini, G Bruno, F Lazzeri, P Castoldi, “First demonstration of SDN-based segment routing in multi-layer networks”, OFC Conf. 2015

[5] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini and P. Castoldi, "Experimental Demonstration of Segment Routing," in IEEE/OSA Journal of Lightwave Technology (JLT), vol. 34, no. 1, pp. 205-212, Jan.1, 1 2016.

[6] E. Moreno, A. Beghelli, F. Cugini, “Traffic Engineering in Segment Routing Networks” in Journal of Computer Networks, Vol. 114, Pages 23–31, Feb 2017

[7] A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini, P. Castoldi, “Segment Routing for Effective Recovery and Multi-domain Traffic Engineering”, in Journal of Optical Communications and Networking (JOCN), Jan. 2017

[8] L. Davoli, L. Veltri, P. Ventre, G. Siracusano, and S. Salsano. "Traffic engineering with segment routing: SDN-based architectural design and open source implementation." In Software Defined Networks (EWSDN), 2015.

[9] Bidkar, S., et al, “Field trial of a software defined network (SDN) using carrier Ethernet and segment routing in a tier-1 provider”. GLOBECOM, Dec. 2014.

[10] C. Thaenchaikun et al, “Mitigate the load sharing of segment routing for SDN green traffic engineering” ISPACS 2016

[11] D. Cai, et al., “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.

[12] A. Giorgetti, P. Castoldi, F. Cugini, et. al, “Path encoding in segment routing,” in Proc. GLOBECOM, Dec. 2015.

[13] 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.

[14] 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 carriergrade networks,” in Proc. SIGCOMM, Aug. 2015.

[15] 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.

[16] 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 Tech Dig. OFC, Mar. 2015.

[17] A. Fressancourt and M. Gagnaire, “A SDN-based network architecture for cloud resiliency,” in 12th Annual IEEE Consumer Communications Networking Conf. (CCNC), Jan. 2015.

[18] 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.

[19] B. Raeisi and A. Giorgetti, "Software-based fast failure recovery in load balanced SDN-based datacenter networks," 2016 6th International Conference on Information Communication and Management (ICICM), Hatfield, 2016.