Demonstration of Segment Routing with SDN Based Label Stack Optimization

Rabah Guedrez* Olivier Dugeon* Samer Lahoud† Géraldine Texier‡
*Orange Labs, Lannion, France,
rabah.guedrez@orange.com, olivier.dugeon@orange.com
†*IRISA/Université de Rennes 1/Adopnet Team, Rennes, France, samer.lahoud@irisa.fr
‡IRISA/Télécom Bretagne/Adopnet Team, Rennes, France, geraldine.texier@telecom-bretagne.eu

***Abstract*—Segment Routing (SR) architecture is a promising technology. It is being standardized in collaboration between vendors and service providers. Through its simplistic control plane and the reuse of existing data planes namely MPLS and IPv6, SR helps operators to reduce the Operation Expense (OpEx) and the Capital Expense (CapEx).**

In the instantiation of SR over the MPLS data plane (SR-MPLS), a SR path gets encoded as a label stack that the ingress nodes push onto the client packet. However, the longer the path in term of traversed nodes the bigger the stack gets. In this demonstration, we couple the capabilities of an SDN controller and a path encoding engine to reduce that size of the label stack to express segment routing paths.

***Index Terms*—Segment Routing, MSD, MPLS, traffic engineering.**

I. Introduction

Segment Routing (SR) architecture [1] got a lot of attention especially from Service Providers (SP) as it reduces the operation and management overhead. SR deployment is very straight forward as it can be deployed with a software upgrade. Therefore, SPs are not required to invest in new hardware. Currently, the standardization efforts are directed by the Service Provider's use cases. Specifically, an important focus on the SR-MPLS, given the amount of work conducted by the IETF working groups. The focus on SR-IPv6 [2] comes from content distribution actors such Google and Facebook [2].

SR uses the source routing forwarding paradigm. In fact, a SR Path (SRP) is encoded as a succession of segments Identifiers (SIDs). Depending on the data plane, the format of the SID and the encoding of the SRPs changes. In the case of SR-MPLS, a SID is 20 bits label and the SRP is encoded as label stack. Unfortunately, routers have a limitation on the number of labels that get pushed onto a packet, this limitation is a known as the Maximum Stack Depth (MSD).

In this demonstration, we present ELEANOR, a northbound application for the OpenDayLight (ODL) Software Defined Network (SDN) controller. Specifically, ELEANOR relies on the SRP computation and management module to enable Traffic Engineering (TE) capabilities in SR networks. ELEANOR mitigates also the impact of the MSD limitation through its label stack optimization module, by minimizing considerably the size of label stack to express SRPs. For the purpose of this demonstration, we have built a topology composed of several routers from different vendors and Quagga-SR our open source implementation of SR-MPLS based on the Quagga suite.

II. Architecture

SR-MPLS couples MPLS's robust data plane with a light distributed control. SR control plane extends already deployed protocols such as OSPF, ISIS, and BGP-LS. SP does not adopt the same SDN vision as for data centers (*i.e.*, removing the network control plane). In fact, SPs don't want to give up the distributed control plane (*i.e.*, maintain the network states in the network). Consequently, in the SDN WAN architecture, the controller synchronizes with network control plane for the discovery of the network topology and available resources. The SDN controller is used for path computation, traffic prediction heuristics, and telemetry analysis.

Segment Routing by design is SDN ready as it removes per-flow states from the network. In SR, paths and their Quality of Service (QoS) requirement are only maintained by ingress nodes. Therefore, the resources availability is not updated and advertised by the Interior Gateway Protocol (IGP). This requires using an SDN controller to deliver functionalities such as traffic engineering.

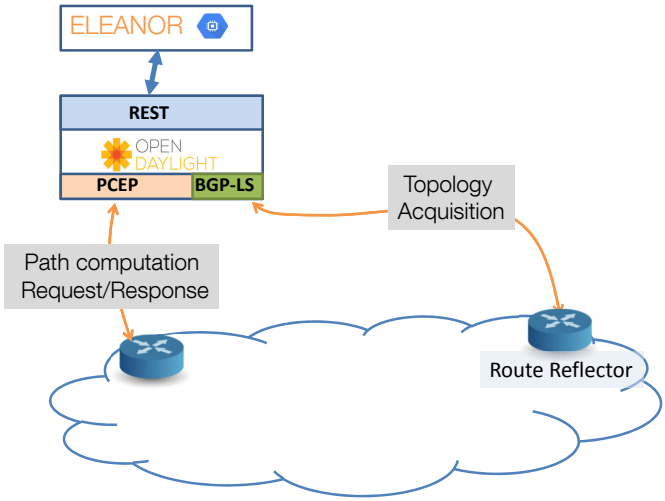

Fig. 1: Reference Architecture

978-1-5090-3672-1/17/$31.00 ©2017 IEEE

In this demonstration, we present the architecture depicted in Fig. 1. The network topology is composed of SR enabled routers, the OpenDaylight (ODL) SDN controller uses two of its southbound interfaces to communicate with the network: BGP-LS for topology acquisition and PCEP to push SRPs configuration onto the routers. ELEANOR is an application developed for SRPs computation, management and label stack optimization, it is based on the open source project Pathman-SR [3], and it communicates with ODL through its northbound REST API.

A. ELEANOR Architecture

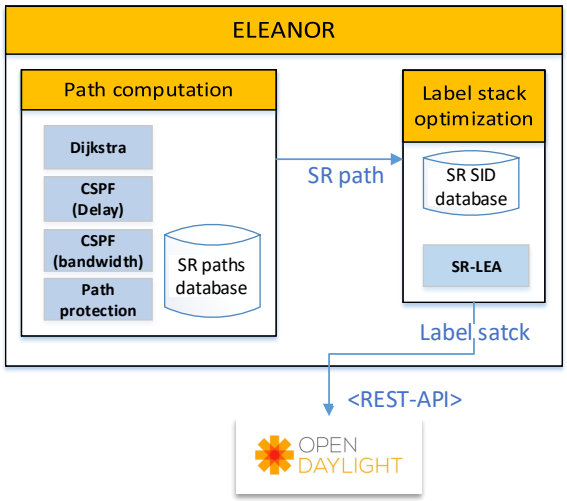

Fig. 2: ELEANOR architecture

ELEANOR's architecture as depicted in Fig. 2, has two main modules: the path computation module and the label stack optimization module:

1) Path Computation Module: The path computation module is a suite of path computation algorithms. A client's path computation request includes the QoS requirements (*e.g.*, delay, bandwidth, path protection). Accordingly, the appropriate Constraint Shortest Path First (CSPF) algorithm is called. The resulting path is then passed to the label stack optimization module, where the SR Label Encoding Algorithm (SR-LEA) is used to compute the minimum label stack to express a SRP.

In SR-MPLS, a SRP is carried in the packet's header as a label stack; this minimizes considerably the number of states core routers has to maintain. Therefore, no signaling protocols such as Resource Reservation Protocol Traffic Engineering (RSVP-TE) or Label Distribution Protocol (LDP) are required. Unfortunately, losing the signaling process means that the resources availability information is not updated hence not advertised in the network. To mitigate this problem, in ELEANOR, all the computed SRPs with their QoS requirements and priority are saved in the SRP database. This is used by CSPFs algorithm to check if a link satisfies the

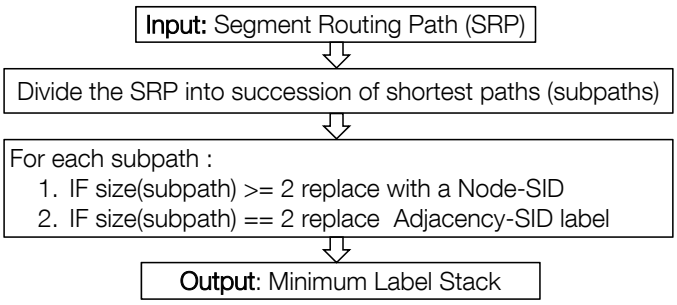

Fig. 3: SR-LEA flowchart.

requested bandwidth. It is also used for global optimization, where the administrator can schedule periodic SRPs placement to maximize the rate of acceptance of future SRP demands.

2) Label Stack Optimization Module: In order to compute the minimum label stack to express a (SRP), we have implemented the Segment Routing Label Encoding Algorithm (SR-LEA). As depicted in the Fig. 3, SR-LEA splices a SRP into multiple subpaths, a subpath is a succession of nodes. Subpaths composed of three or more nodes are the ones that follow the IGP shortest path. Those composed of only two nodes may or may not follow the IGP shortest path. Paths that are three or more nodes long are encoded using the last node's Node-SID, and to ensure that the initial SR path is respected, those that are two nodes long are encoded using the Adj-SID attributed by the First node to reach the second one.

For example, a client requests a path between *Amiens* and *Toulouse* with 100 MB of bandwidth. First, the appropriate CSPF is called to compute the path. Thus, the resulting best path is {*Amiens, Paris, Orleans, Lyon, Marseille, Toulouse*}. Second, the path is passed to SR-LEA algorithm, the SRP is spliced into three parts: {*Amiens, Paris, Orleans*}, {*Orleans, Lyon*} and {*Lyon, Marseille, Toulouse*}. Then as depicted in Fig. 3, the first subpath is encoded with the Node-SID of *Orleans*, the second subpath is encoded by the Adj-SID attributed by *Orleans* to its adjacency to *Lyon*, and the third subpath is encoded with the Node-SID of *Toulouse*. The resulting label stack is a combination of Node-SIDs and Adj-SIDs encoded in XML format, which get pushed into ODL using the POST method.

B. Network Topology

The network topology designed for this demonstration is composed of industrial routers and our SR-MPLS open source implementation: Quagga-SR routers. As can be seen in Fig. 4. The routers are mapped over the France map, each router is named based on the city it is located in.

We use OSPF protocol with SR extension enabled [4], network routers use OSPF-SR to exchange SR information such as: Node-SID, Adj-SID, and SRGB, etc. The border of the network is composed of a mix of industry routers, for the transit nodes we use Quagga-SR routers. The border

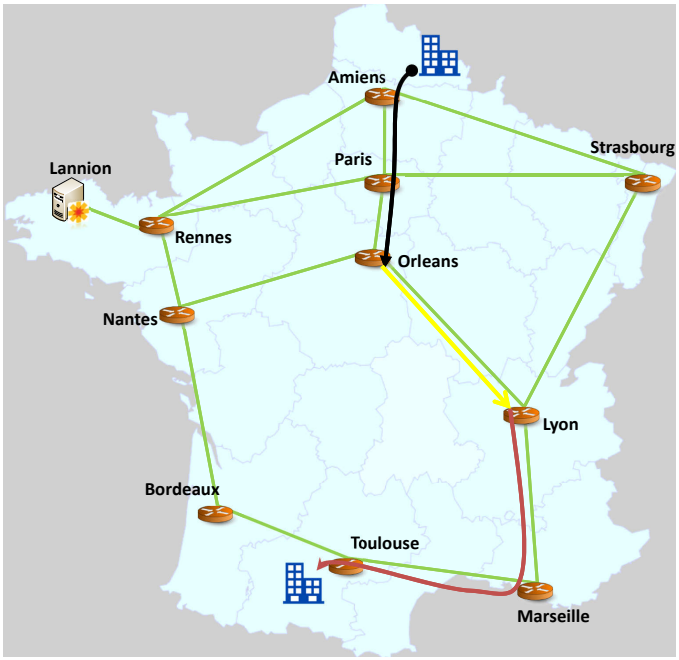

Fig. 4: Demonstration Topology

routers have Path Computation Client (PCC) enabled; each PCC establishes a PCEP [5] session with ODL's southbound interface PCEP, ODL commanded by ELEANOR uses this session for the creation and deletion of the SRPs. *Rennes* router is configured as Route Reflector (RR), this router establishes a BGP-LS session with ODL, this session is used by ODL for the acquisition of the link state topology, and this topology gets replicated in the ELEANOR application.

C. Quagga-SR

Quagga routing software suite [6] is used as an open source router, it is composed essentially of two components: an implementation of several protocols such as OSPFv2, OSPFv3, ISIS and BGP. 2). Zebra module ensures the communication between the different routing daemons and the Linux kernel.

For the purpose of this demonstration, we have extended the Quagga suite to deliver SR functionalities. This implementation requires a Linux Kernel 4.6.4. The detail of our implementation is depicted Fig. 5. Several modules and extensions had been developed in order to add the support of SR. Notably, the OSPF Daemon (OSPFD) is extended to support the encoding and decoding of the SR TLVs. SR database maintains the SR information locally configured or learned via the neighbors. Several SR specific command has been added to vtysh shell: to enable SR, SRGB configuration, Node-SID configuration, etc.

To ensure the proper functioning of Quagga-SR router, interoperability tests with routers from different vendors has been successfully performed.

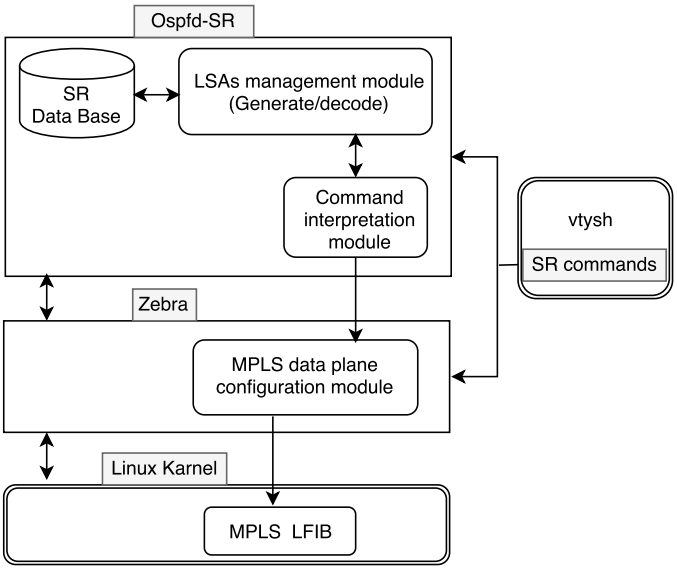

Fig. 5: Quagga-SR: Open source implementation of SR-MPLS

III. CONCLUSION

For this demo, we have presented an SDN architecture that delivers Segment Routing traffic engineering. This architecture reflects the service providers point of view of how SDN for WAN should be. We detailed ELANOR, an ODL northbound application for segment routing path computation, management, and label stack encoding optimization. The network topology for this demonstration is composed of routers from different vendors and our open source router Quagga-SR.

REFERENCES

[1] C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, "Segment Routing Architecture," Internet Engineering Task Force, Internet-Draft draft-ietf-spring-segment-routing-09, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-09

[2] S. Previdi, C. Filsfils, B. Field, I. Leung, J. Linkova, E. Aries, T. Kosugi, E. Vyncke, and D. Lebrun, "IPv6 Segment Routing Header (SRH)," Internet Engineering Task Force, Internet-Draft draft-ietf-6man-segment-routing-header-01, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-6man-segment-routing-header-01

[3] Cisco, "Pathman-sr," https://github.com/CiscoDevNet/pathman-sr, 2016.

[4] P. Psenak, S. Previdi, C. Filsfils, W. Henderickx, J. Tantsura, H. Gredler, and R. Shakir, "OSPF Extensions for Segment Routing," Internet Engineering Task Force, Internet-Draft draft-ietf-ospf-segment-routing-extensions-08, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-08

[5] S. Sivabalan, J. Medved, C. Filsfils, V. Lopez, J. Tantsura, W. Henderickx, E. Crabbe, and J. Hardwick, "PCEP Extensions for Segment Routing," Internet Engineering Task Force, Internet-Draft draft-ietf-pce-segment-routing-07, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-pce-segment-routing-07

[6] C. Networks, "quagga," https://github.com/CumulusNetworks/quagga, 2016.