Fourth European Workshop on Software Defined Networks

Traffic Engineering with Segment Routing: SDN-based Architectural Design and Open Source Implementation

Luca Davoli(1), Luca Veltri(2), Pier Luigi Ventre(3), Giuseppe Siracusano(3), Stefano Salsano(3)
(1) Univ. of Parma / Consortium GARR - (2) Univ. of Parma - (3) Univ. of Rome Tor Vergata

***Abstract* – Traffic Engineering (TE) in IP carrier networks is one of the functions that can benefit from the Software Defined Networking paradigm. However traditional per-flow routing requires a direct interaction between the SDN controller and each node that is involved in the traffic paths. Segment Routing (SR) may simplify the route enforcement delegating all the configuration and per-flow state at the border of the network. In this work we propose an architecture that integrates the SDN paradigm with SR-based TE, for which we have provided an open source reference implementation. We have designed and implemented a simple TE/SR heuristic for flow allocation and we show and discuss experimental results.**

Keywords – Segment Routing, Software Defined Networking, Traffic Engineering, Open Source, Emulation.

I. OPEN SOURCE IMPLEMENTATION OF THE DATA AND CONTROL PLANES FOR MPLS-BASED SEGMENT ROUTING

Let us consider an ISP network managed by a (logically) centralized SDN controller, as shown in Figure 1. Network nodes are classified in Provider Edge (PE) routers and Core Routers (CR), both types are assumed to be MPLS nodes; PE nodes are connected to the Customer Edge (CE) routers, which are the external traffic sources and destinations. MPLS data plane and relative labels are used to realize Segment Routing (SR) forwarding [1].

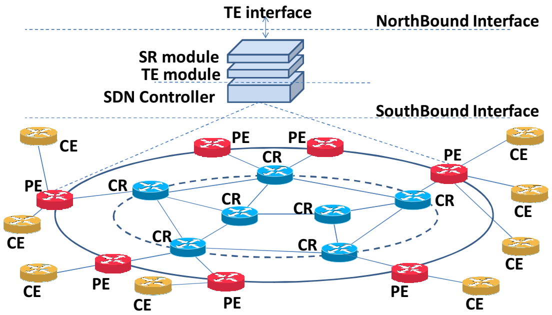

Figure 1 - Traffic Engineering in a SDN-enabled network.

Global Segment IDs (SIDs) are derived from the loopback interface address that can univocally identify a MPLS router in an ISP network. Therefore, we do not rely on the IGP (OSPF in our case) for distributing the SIDs. Packet forwarding is based on basic label switching operations and no change to the MPLS forwarding plane is required [2]. In particular, the PE and CR nodes are hybrid IP/SDN nodes [5], hereafter referred to as OSHI nodes. No MPLS control plane has to be implemented within these nodes, while the SR forwarding behavior is implemented through the Software Defined Networking paradigm [3] by means of an OpenFlow [4] controller interacting with OpenFlow capable OSHI nodes, and with the introduction of a SR daemon that interacts with IP control plane of the nodes [6]. The node architecture is explained in [7].

II. TRAFFIC ENGINEERING WITH SEGMENT ROUTING

With reference to Figure 1, we enhance a SDN controller with TE/SR modules. We assume that the SDN controller is requested to allocate a set of traffic flows with a specified bit rate, knowing the link capacity. The TE/SR modules will first allocate hop-by-hop *TE paths* solving a classical *flow assignment* problem. Then, for each TE path, it will compute a corresponding *SR path* for instructing the flow packets through the assigned TE path.

For the *flow assignment* we implemented a modified version of the heuristic proposed in [6] that tries to minimize the overall network crossing time. The heuristic is divided in two phases: i) a Constrained Shortest Path First (CSPF) phase, where a first allocation of the flows is realized (flows that cannot be allocated in this phase are rejected); ii) a heuristic re-assignment phase, which tries to re-allocate all admitted flows one-by-one, in order to minimize the global network crossing time. The second phase is executed multiple times until no improvement is achieved. The SR assignment algorithm is then performed with the objective of finding the minimal-length SR paths corresponding to each TE path, i.e. the shortest list of SIDs that allows the packets to follow the assigned TE path, according to the default IP routing tables of the nodes. We propose a simple SR assignment algorithm reported with pseudo-code in Figure 2 (more details in [8]). The algorithm assumes the existence of a global SID associated to every network node. In order to work only with global SIDs, we suppose that when a direct link exists between two nodes, it matches the shortest path between those nodes. The following notation is used:

- $tep=tep(n_s,n_d)$: is a TE path that has to be set-up between the source node n_s and destination node n_d;
- $tep(n_1,n_2)$: portion of the TE path starting from node n_1 and ending with node n_2;
- $SPN(n_1,n_2)$: the number of equal-cost shortest paths from n_1 to n_2, based on the routing tables already set-up using a Shortest Path First algorithm (e.g. OSPF);
- $sp(n_1,n_2)$: if $SPN(n_1,n_2)\equiv 1$, it is the shortest path between nodes n_1 and n_2, otherwise it is not defined;
- $prec(p,n)$: the preceding node of node n along a path p;
- srp: the SR path containing the list of assigned SIDs.

```
procedure INITIALIZATION
    $n_s = R_0$,   $n_d = R_N$,   srp = { }
procedure SEGMENTALLOCATION
    BEGIN
        p = tep($n_s$, $n_d$)
        if ((SPN($n_s$, $n_d$) == 1) AND (sp($n_s$, $n_d$) == p)) then
            ADD $n_d$ to srp
            if $n_d$ == $R_n$ then goto FINISH
            else $n_s$ = $n_d$ ;  $n_d$ = $R_n$ ; goto BEGIN
        else $n_d$ = prec(p, $n_d$) ; goto BEGIN
    FINISH
```

Figure 2 – Pseudo-code representation of the developed algorithm

978-1-5090-0180-4/15 $31.00 © 2015 IEEE
DOI 10.1109/EWSDN.2015.73

III. TE/SR IMPLEMENTATION AND TESTS

The proposed solution described in the previous sections has been implemented and tested. The source code is available at [10]. Based on our implementation, we carried out an experimental analysis of the proposed architecture and algorithms, with the following two main goals: i) testing the SR assignment algorithm; ii) testing the overall implementation of the solution, from the SDN-based control plane to the MPLS-based SR forwarding in the data plane. We considered a relatively large scale topology with 153 nodes and 354 unidirectional links, the "Colt Telecom" topology included in the Topology zoo dataset [9]; we assumed that all links have the same capacity. We randomly selected 40% of the nodes to be PE, then we randomly selected 20% of the PE couples to be active source/destination of traffic flows. For each active couple of PEs, in each direction we have an average of 3.5 flows with the sum of the flow rates equal to 10% of the capacity of a link and the size of each flow follows a negative exponential distribution. We generated a total of 2460 flows, which saturate the network capacity, resulting in a total of 940 admitted flows according to the flow assignment algorithm. For each one, the SR path has been computed through the proposed SR assignment algorithm. Figure 3 reports the distribution of path lengths for the TE paths and for the *natural paths* (i.e. the shortest path from ingress PE to the egress PE).

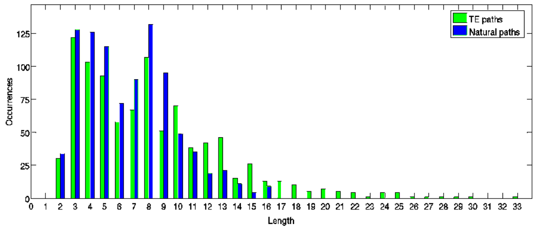

Figure 3 – Distribution of path length for TE and natural paths

The result of the SR assignment algorithm is shown in Figure 4, which provides the distribution of SR path lengths. In Figure 5 we show the mean length of TE paths, natural paths, and SR paths for different number of allocated flows.

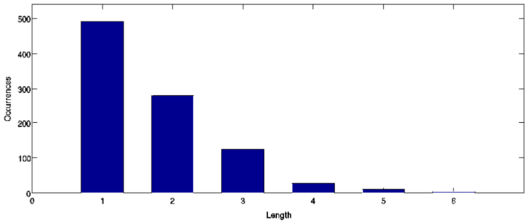

Figure 4 – Distribution of SR path length (number of SIDs in the path)

We also made a simple experimental evaluation of the processing time of the proposed TE/SR algorithms. We used a PC with an Intel Core i7 2Ghz and 6GB RAM. The processing time of the SR assignment algorithm is negligible with respect to the flow assignment heuristic. In the considered range (up to 900 admitted flows) it was possible to run both algorithms and allocate the flows in less than 8 seconds.

In the first class of experiments, we only evaluated the SR paths but did not try to allocate them. For the second class of experiments, a smaller testbed network, composed by 12 OSHI nodes and a SDN controller, has been considered and emulated on Mininet. We deployed 16 virtual circuits (VCs) through unidirectional SR paths.

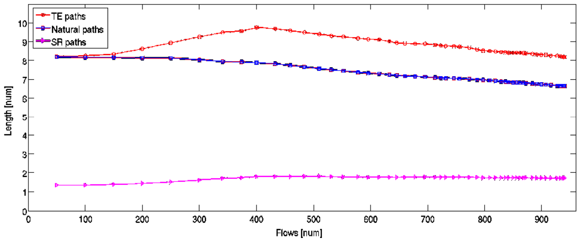

Figure 5 – Mean lengths of TE paths, natural paths, and SR paths vs. the number of allocated TE flows

Figure 6 shows the network with two bidirectional VCs between the CE nodes A and B; it also shows the allocation provided by the TE flow assignment heuristic followed by the SR allocation algorithm. The VC-1 is routed in the network with a SR path of length 1. The VC-2 in the A>B direction has an SR path of length 2, while in the B>A direction has a path of length 3.

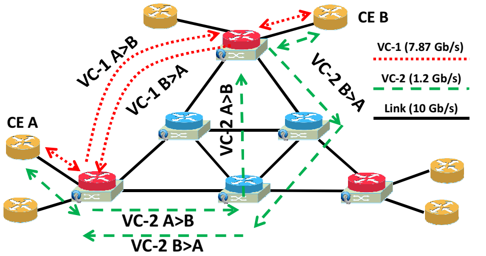

Figure 6 – Network topology with 12 nodes deployed on Mininet

For an easier reproducibility of the experiments, we have packaged all software components in a ready-to-go virtual machine (available at [5]).

ACKNOWLEDGMENTS

This work builds on the results of DREAMER project, partly funded by the EU as one of the beneficiary projects of the GÉANT Open Call research initiative.

REFERENCES

[1] C. Filsfils, S. Previdi (Eds.) et al. "Segment Routing Architecture", IETF draft-ietf-spring-segment-routing-01, Feb 2015
[2] C. Filsfils et al., "Segment Routing with MPLS data plane", IETF draft-ietf-spring-segment-routing-mpls-00, November 2014.
[3] "Software-Defined Networking: The New Norm for Networks", ONF White Paper, April 13, 2012
[4] Nick McKeown et al. "OpenFlow: Enabling Innovation in Campus Networks", ACM Communications Review, 2008
[5] OSHI homepage http://netgroup.uniroma2.it/OSHI
[6] P. L. Ventre, "Traffic engineering in OSHI networks" section 1.5 of technical report "QR.2" available at [5]
[7] S. Salsano et al. "OSHI-Open Source Hybrid IP/SDN networking and Mantoo-a set of management tools for controlling SDN/NFV experiments." arXiv preprint arXiv:1505.03579 (2015).
[8] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, S. Salsano, "Traffic Engineering with Segment Routing: SDN-based Architectural Design and Open Source Implementation", arXiv preprint arXiv:1506.05941 (2015)
[9] S. Knight et al. "The Internet Topology Zoo", IEEE Journal on Selected Areas in Communications, Vol. 29, No. 9, October 2011
[10] https://github.com/netgroup/SDN-TE-SR