Traffic Engineering in Segment Routing Networks

Eduardo Moreno[a], Alejandra Beghelli[b], Filippo Cugini[c]

[a] *Faculty of Engineering and Sciences, Universidad Adolfo Ibañéz, Santiago, Chile.*
[b] *Faculty of Engineering and Sciences, Universidad Adolfo Ibañéz, Viña del Mar, Chile.*
[c] *CNIT, Pisa, Italy.*

Abstract

Segment routing (SR) has been recently proposed as an alternative traffic engineering (TE) technology enabling relevant simplifications in control plane operations. In the literature, preliminary investigations on SR have focused on label encoding algorithms and experimental assessments, without carefully addressing some key aspects of SR in terms of the overall network TE performance.

In this study, ILP models and heuristics are proposed and successfully utilized to assess the TE performance of SR-based packet networks. Results show that the default SR behavior of exploiting equal cost multiple paths (ECMP) may lead to several drawbacks, including higher network resource utilization with respect to cases where ECMP is avoided. Moreover, results show that, by properly performing segment list computations, it is possible to achieve very effective TE solutions by just using a very limited number of stacked labels, thus successfully exploiting the benefits of the SR technology.

Keywords: Segment routing, ILP, heuristic.

1. Introduction

The Segment Routing (SR) technology has been recently introduced to enable effective traffic engineering (TE) while simplifying control plane operations [1, 2]. SR can be operated in packet networks supporting Multiprotocol Label Switching (MPLS). In particular, according to SR, packet flows are enforced through a specific path by applying, at the ingress node, a specifically computed stack of segment identifiers (SIDs). The stack of SIDs, called *segment list*, corresponds to the *stack of labels* in the MPLS architecture. In principle, only the top SID in the list is considered by transit nodes to perform packet forwarding. In particular, each packet is forwarded along the shortest path toward the network element represented by the top SID. For instance, a SID can represent an Interior Gateway Protocol (IGP) prefix which identifies a specific router, such as the IGP router ID (called IGP-Node Segment in the context of SR [1]).

Differently with respect to traditional MPLS networks, SR maintains per-flow state only at the ingress node, where the segment list is applied. Therefore, no signaling protocol (e.g., Reservation Protocol with traffic engineering extensions - RSVP-TE) is required to populate the forwarding table of transit nodes. This way, a simplified control plane is employed, just relying on the IGP that is properly extended to advertise SIDs [3]. Scalability is significantly improved, also because transit nodes do not have to maintain MPLS Label Switch Paths (LSPs) state information.

To fully exploit the SR benefits, it is necessary to efficiently compute the segment list to be applied on the ingress node. Such computation, provided by a Path Computation Element (PCE) possibly located within a Software Defined Network (SDN) Controller, has to be carefully performed to achieve effective TE solutions in the whole network.

Thus, in addition to traditional objective functions and constraints that characterize current MPLS TE solutions (e.g., minimization of the maximum link utilization subject to the link capacity within the whole network), the segment list computation has to take into account specific constraints and additional objective functions.

First, each path has to be encoded as a combination of one or more shortest segments.

Second, since currently deployed MPLS equipments do not support a large stack of labels, path encoding has to consider the constraint on the maximum number of stacked SIDs, called *segment list depth* (SLD). Todays MPLS routers typically support SLD values in the range between 5 and 8 labels, determined by the internal forwarding engine (i.e., ASIC)

Third, since the segment list introduces packet overhead, path encoding has to minimize the introduced packet overhead.

Finally, as it will be detailed later, equal-cost multiple paths (ECMP) require specific treatment since, by default in the context of SR, they are exploited whenever available. However, to avoid packet misordering at the destination, packet inspection operations may be required, introducing constraints on minimum hardware requirements on SR equipments.

So far, these aspects have not been adequately investigated.

For example, the work in [4] considers the SR application in Carrier Ethernet networks. In particular, the authors propose to combine the benefits of SR with those of a software defined networking (SDN) architecture [5].

In [6], the authors proposed a SR implementation for Carrier Ethernet networks aiming at reducing the required segment list depth thorugh integrations of the segment lists at some intermediate nodes (named swap nodes).

In [7] and [8], algorithms to encode the segment lists are proposed. However, in these works, path encoding is applied only on previously identified paths and no TE solutions in the whole network are addressed.

The works in [9, 10] propose two experimental implementations of SR based on an OpenFlow-based controller and on a PCE-based controller.

Finally, the works in [11] and [12] focus on SR experimental demonstrations in the context of multi-domain and reliable scenarios, respectively.

All these studies do not address the definition and evaluation of suitable algorithms for effective TE solutions in SR networks. The only work closely related to this paper is [13]. In this valuable study, a so called traffic matrix oblivious algorithm including a game theoretic like analysis for offline and online segment routing scenario is proposed. However, the reported analysis is not suitable to drive considerations on how to efficiently exploit the SR technology.

This study proposes effective ILP models and heuristics for packet networks exploiting the segment routing (SR) technology. The TE performance of SR is then assessed over a number of different network scenarios.

Obtained results allow on the one hand to assess the possible drawbacks due to the use of SR-based ECMP and on the other hand to show that efficient segment list computation can successfully provide effective TE solutions without experiencing scalability issues.

2. Segment Routing

To clarify the SR behavior, a 2x3 reference network composed of six nodes and seven links is considered (see Fig.1). The control plane consists of an IGP routing protocol extended to advertise IGP-Node Segments (i.e., router IDs [1]). Hop count is assumed as metric. No signaling protocol is configured. The data plane consists of packet nodes supporting MPLS forwarding.

A request from node 1 to node 3 is first considered. A PCE/SDN Controller computes the segment list as a combination of shortest segments. In this case, just one (unique) shortest route exists from 1 to 3, passing through node 2. The SDN Controller then computes and configures on node 1 a segment list including a single SID (i.e., a single label) representing destination node 3. Node 1 then pushes label SID 3 and sends packets towards the shortest route, i.e. along link 1-2. Node 2, by just elaborating label SID 3, is able to forward the packet along the shortest route towards node 3, i.e. on link 2-3, successfully reaching the destination where the label is popped.

To detail the case where equal cost multiple paths (ECMP) are present, or specific strict routes need to be selected, a second request from node 1 to node 6 is here considered. In this case, there are three equal cost routes (see Fig.1): 1-2-3-6, 1-2-5-6 and 1-4-5-6. In this case, following the default SR behavior, if a single label SID 6 is pushed at node 1, all three routes are exploited. In particular, node 1 splits the traffic between link 1-2 and 1-4. Packets reaching node 4 are then forwarded towards node 6 along the route 1-4-5-6. Instead, packets reaching node 2 are further split between link 2-3 and link 2-5, before arriving at the destination 6. In case four units of traffic are generated from node 1, given the split operated at node 1, effective load balancing is actually performed on links 1-4 and 1-2, each carrying two units of traffic. Then, load balancing is further performed at node 2, obtaining just one unit of traffic on links 2-3 and 2-5. However, given the considered topology, the traffic in the network then recombines in an unbalanced way: the traffic entering node 6 is composed by three units forwarded by node 5 and just one by node 3. That is, exploiting ECMP, traffic load in the network is then distributed in a way that strongly depends on the actual traffic matrix and topology, potentially driving to ineffective TE solutions.

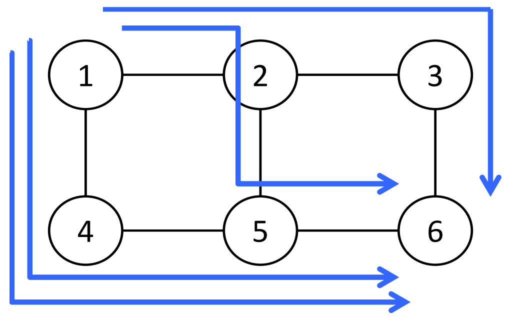
In general, the default SR behavior exploiting ECMP may need to be avoided if:

1. traffic distribution in the network leads to unbalanced situations and/or traffic congestions;
2. some routes present inadequate quality of service (e.g., excessive latency);
3. the forwarding device is not able to guarantee per-flow forwarding. Indeed, traffic split among ECMP needs packet inspection operations (e.g., at the TCP/UDP level) to perform per-flow forwarding and avoid packet mis-ordering at the destination. That is, when ECMP are exploited, the top label is not sufficient to determine the forwarding action and adequate hardware capabilities are needed to operate packet inspection at the wire speed. In case such performance is not available on some routers, it is recommended to configure just strict SR routes and avoid the use of ECMP.

In the following, the cases where ECMP is avoided are discussed with reference to the example above of Fig.1.

If path 1-4-5-6 needs to be specifically selected, a stack of labels is required having SID 4 as top label (to be popped by node 4) and SID 6 as bottom of the stack (to become top one after node 4).

In case path 1-4-5-6 needs to be specifically avoided, and ECMP can be exploited on paths 1-2-3-6 and 1-2-5-6 the stack of labels requires SID 2 as top



Segment list	Paths
6	1-2-3-6, 1-2-5-6, 1-4-5-6
2,6	1-2-3-6, 1-2-5-6,
3,6	1-2-3-6
4,6	1-4-5-6
2,5,6	1-2-5-6

Figure 1: Example of segment list (i.e., label stack) for path 1-6 over the 2x3 network topology.

label (to be popped by node 2) and SID 6 as bottom of the stack (to become top one after node 2, where split is performed).

On the other hand, if ECMP has to be completely avoided and path 1-2-5-6 needs to be specifically selected, the stack of labels requires three labels (i.e., Segment list {2,5,6}, SLD=3): SID 2 as top label (to be popped by node 2), SID 5 as second label, and SID 6 as bottom of the stack.

A further relevant aspect to be considered in SR networks is the need to guarantee the optimal path encoding (see [7, 8]). To explain this aspect, let us assume that link 2-5 is not present in the reference network shown in Fig.1. That is, only two paths are available from 1 to 6: 1-4-5-6 and 1-2-3-6. In this case, if 1-2-3-6 needs to be specifically selected, two segment list options are available. The first list includes, besides node 6, node 2. The second one, besides node 6, node 3. Both options guarantee the expected packet forwarding using a path encoding with minimum possible SLD. However, the two options lead to different packet overhead. Indeed, in the latter case, one additional label has to be transmitted over link 2-3, wasting bandwidth resources with respect to the case exploiting SID 2 as top label.

That is, in the context of segment routing, traffic engineering solutions have to target: (1) minimization of the maximum utilization among links; (2) minimization of the average utilization of links; (3) minimization of SLD; (4) minimization of the packet overhead.

3. ILP Models

In this section we present three different integer linear programming (ILP) models.

The first one, used as a first benchmark, exploits ECMP and it is called ECMP model. That is, data is transmitted along all possible shortest paths.

The second model, denoted by ShP, forces to select only one route and it is used as a second benchmark.

Finally, we exploit the full capability of SR traffic routing in the third model, called SegmR.

Let $G = (N, L)$ be the graph representing the network, with N nodes and L links. Each link $l \in L$ has a maximum capacity equal to K_l. Given a set of connections C, each of them defined by its source node $s(c) \in N$, its terminal node $t(c) \in N$ and its demand d_c. We denote by P_c the set of all shortest paths between $s(c)$ and $t(c)$. Similarly, we denote by $s(p)$ and $t(p)$ the source and terminal node of a path p.

Let x_c^p be a variable representing the fraction of the connection flow c routed through the path p, for each $p \in P_c, c \in C$, and let α be a variable representing the maximum utilization of the network, defined as the fraction between the flow in the most utilized link and its capacity. Then, the following basic model finds the routes used by each connection such that the maximal utilization is minimized.

$$\min \alpha \tag{1}$$

$$\sum_{p \in P_c} x_c^p = 1 \qquad \forall c \in C \tag{2}$$

$$\sum_{c \in C} \sum_{p \in P_c : l \in p} d_c x_c^p \leq \alpha \cdot K_l \qquad \forall l \in L \tag{3}$$

$$0 \leq x_c^p \leq 1 \qquad \forall c \in C, \forall p \in P_c \tag{4}$$

$$\alpha \geq 0 \tag{5}$$

Constraint (2) indicates that the connection demand should be routed completely through its shortest paths. Constraint (3) limits the flow over each link to be less or equal than α times its capacity, and the objective function forces that the maximum utilization α should be minimized.

In order to build the ECMP model, we need to add an extra constraint:

$$\sum_{p \in P_c : l_1 \in p} x_c^p = \sum_{p' \in P_c : l_2 \in p'} x_c^{p'} \tag{6}$$

for all $c \in C, p, p' \in P_c$ and for all outgoing links l_1, l_2 sharing one common node. That is, if there is more than one shortest path, then at each node the flow splits equally among outgoing links belonging to one of these shortest paths.

If we remove constraint (6) from the model and enforce variables x_c^p to be binary, then the model represents the choice of a unique route for each connection, among all shortest paths, which can be obtained using segment routing. However, at a given node, two connections with the same label should follow the same path. This can be enforced by adding the consistency constraint

$$x_c^p \geq x_{c'}^{p'} \tag{7}$$

for all $c, c' \in C$, $p \in P_c$, $p' \in P_{c'}$ such that $t(c) = t(c')$ and $p \subset p'$. That is, if two connections c and c' terminate at the same node, and their potential paths p and p' satisfy that p is a subpath of p', then if p' is selected for c', then p should be selected for c. This is because packets of both connections will arrive at node $s(p)$ with the same labels, so they should be routed through the same path. This is the ShP model.

Finally, to represent the full capability for traffic routing of SR networks, we propose the SegmR model. Let $\mathcal{P}$ be the set of *segments*, i.e. all unique shortest paths between any pair of nodes. Note that in SR networks, given the uniqueness of the paths in $\mathcal{P}$, then a packet can follow a path $p \in \mathcal{P}$ from $s(p)$ to $t(p)$ by just using the label $t(p)$. Then, we define binary variables y_c^p for all $c \in C, p \in \mathcal{P}$ representing that p is used as a subpath for connection $c \in C$.

$$\min \alpha \tag{8}$$

$$\sum_{p \in \mathcal{P}: s(p)=n} y_c^p - \sum_{p \in \mathcal{P}: t(p)=n} y_c^p = b_{c,n} \qquad \forall c \in C, n \in N \tag{9}$$

$$\sum_{c \in C} \sum_{p \in \mathcal{P}: l \in p} d_c y_c^p \leq \alpha \cdot K_l \qquad \forall l \in L \tag{10}$$

$$\sum_{p \in \mathcal{P}} y_c^p \leq \kappa \qquad \forall c \in C \tag{11}$$

$$y_c^p \in \{0, 1\} \qquad \forall c \in C, p \in \mathcal{P} \tag{12}$$

$$\alpha \geq 0 \tag{13}$$

where

$$b_{c,n} = \begin{cases} 1 & \text{if } n = s(c), \\ -1 & \text{if } n = t(c), \\ 0 & \text{otherwise.} \end{cases}$$

Constraint (9) is usually called flow conservation constraint and it ensures that the connection is routed correctly between $s(c)$ and $t(c)$. Constraint (10) is similar to previous constraint (3), and constraint (11) fixes a maximum number of subpaths (equivalently, the maximum number of labels in the stack, or SLD) for each connection up to κ. As before, the following extra consistency constraint should be included:

$$y_c^p \geq y_{c'}^p \tag{14}$$

for all $c, c' \in C$, $p \in \mathcal{P}$ such that $s(p) = s(c)$ and $t(p) = t(s)$. That is, if c' uses p as a subpath, then the connection between $s(p)$ and $t(p)$ should be routed through the same path p.

4. Heuristic approach

Due to the high computational complexity of ILP models, some instances take too much time to be solved. Therefore, we also propose a heuristic approach

Algorithm 1

Input data: A set of paths for every connection c.
P_c is the set of shortest paths between $s(c)$ and $t(c)$. The r-th alternative shortest route between $s(c)$ and $t(c)$ is denoted by p_c^r and defined by the list of nodes $n_1, n_2, \ldots, n_{|r,c|}$ where $|r, c|$ represents the number of nodes of route p_c^r.

for each connection c **do**
  if P_c has only one element **then**
    SLD=1 and the unique label is node $t(c)$
  else//search for minimum number of labels starts
    for For each alternative route for c: **do**
      Stack=0 //Boolean variable indicating whether the stack is ready or not
      //sequence of nodes of route is divided in two lists: Left List (LL) and Right List
      $LL = n_1, n_2$
      $RL = n_2, n_3, \ldots, n_{|r,c|}$
      while stack==0 **do**
        while no unique shortest route between first and last element of RL **do**
          Leftmost element in RL is eliminated
          New leftmost element in RL is copied to the rightmost part of LL
        end while
        Push rightmost element of RL in stack
        if leftmost element in RL == n_1 **then**
          stack=1 //end of stack label construction
        else
          Overwrite RL with LL (RL=LL)
          LL= first element in RL
        end if
      end while//end while (stack==0)
    end for//end for each alternative route for (s,t)
  end if
end for

to determine a unique route between each node pair that needs to transfer information as well as the segment list associated to that route. The heuristic does so whilst trying to keep the total and maximum network utilization as low as possible, whilst guaranteeing the maximum value of SLD is not exceed. As in the previous section, the link utilization is the sum of the number of connections demands passing through a link. The maximum utilization is the number of connections passing through the most utilized link.

The heuristic takes as input data:

- a set of paths between each node pair,
- the number and identification of labels required by each path, and
- the maximum SLD allowed.

Algorithm 1 was used to determine the number and identification of labels per path, so the packet overhead discussed at the end of Section 2 is kept to the minimum.

As a way of illustration, Table 1 shows the resulting data after applying Algorithm 1 to the 2x3 grid network shown in Figure 1, assuming the set of shortest paths is given as input. The left column identifies the pair source-terminal nodes. In the right column, the set of shortest routes is provided. The number in parentheses after each route is the number of elements in the stack, that is, the SLD. The elements stored in the segment list are those underlined in the route (the rightmost element is stored at the bottom of the stack and so on).

After taking the described input data and processing it, the heuristic approach selects only one path for each connection in such a way that the maximum link utilization is minimized as much as possible and the limit on SLD is not violated. The flowchart of Fig.2 shows the main steps executed by the heuristic.

The steps shown in the flowchart work as follows:

Table 1: Shortest routes for a 2x3 grid network. Underlined nodes (SIDs) represent the associated segment lists.

s-t	SL1 (SLD1)/SL2 (SLD2)/SL3 (SLD3)		s-t	SL1 (SLD1)/SL2 (SLD2)/SL3 (SLD3)
1-2	1-$\underline{2}$ (1)		4-1	4-$\underline{1}$ (1)
1-3	1-2-$\underline{3}$ (1)		4-2	4-$\underline{1}$-$\underline{2}$ (2) / 4-$\underline{5}$-$\underline{2}$ (2)
1-4	1-$\underline{4}$ (1)		4-3	4-$\underline{1}$-2-$\underline{3}$ (2) / 4-$\underline{5}$-$\underline{2}$-$\underline{3}$ (3) / 4-5-$\underline{6}$-$\underline{3}$ (2)
1-5	1-$\underline{2}$-$\underline{5}$ (2) / 1-$\underline{4}$-$\underline{5}$ (2)		4-5	4-$\underline{5}$ (1)
1-6	1-2-$\underline{3}$-$\underline{6}$ (2) / 1-$\underline{2}$-$\underline{5}$-$\underline{6}$ (3) / 1-$\underline{4}$-5-$\underline{6}$ (2)		4-6	4-$\underline{6}$ (1)
2-1	2-$\underline{1}$ (1)		5-1	5-$\underline{2}$-$\underline{1}$ (2) / 5-$\underline{4}$-$\underline{1}$ (2)
2-3	2-$\underline{3}$ (1)		5-2	5-$\underline{2}$ (1)
2-4	2-$\underline{5}$-$\underline{4}$ (2) / 2-$\underline{1}$-$\underline{4}$ (2)		5-3	5-$\underline{2}$-$\underline{3}$ (2) / 5-$\underline{6}$-$\underline{3}$ (2)
2-5	2-$\underline{5}$ (1)		5-4	5-$\underline{4}$ (1)
2-6	2-$\underline{3}$-$\underline{6}$ (2) / 2-$\underline{5}$-$\underline{6}$ (2)		5-6	5-$\underline{6}$ (1)
3-1	3-2-$\underline{1}$ (1)		6-1	6-$\underline{3}$-2-$\underline{1}$ (2) / 6-$\underline{5}$-$\underline{2}$-$\underline{1}$ (3) / 6-5-$\underline{4}$-$\underline{1}$ (2)
3-2	3-$\underline{2}$ (1)		6-2	6-$\underline{3}$-$\underline{2}$ (2) / 6-$\underline{5}$-$\underline{2}$ (2)
3-4	3-2-$\underline{1}$-$\underline{4}$ (2) / 3-$\underline{2}$-$\underline{5}$-$\underline{4}$ (3) / 3-$\underline{6}$-5-$\underline{4}$ (2)		6-3	6-$\underline{3}$ (1)
3-5	3-$\underline{6}$-$\underline{5}$ (2) / 3-$\underline{2}$-$\underline{5}$ (2)		6-4	6-5-$\underline{4}$ (1)
3-6	3-$\underline{6}$ (1)		6-5	6-$\underline{5}$ (1)

1. Let κ be the limit on the number of labels in the stack, i.e. the maximum allowed SLD. All routes with SLD higher than κ are eliminated from further consideration.
2. As a result of Step 1, some connections might be left with no feasible routes. To those connections the route coded as -1 is allocated to signalize that no route with the required limit on SLD was available.
3. All routes with just one label are allocated.
4. For each connection without an allocated route:
   - 4.1 For each network link, its utilization U_l is calculated as the total demand already established on the link. The first time this step is executed, the utilization corresponds to the demand of 1-label segments using the link.
   - 4.2 For each connection c, the weight of each route r (W_c^r) is calculated.
   - 4.3 Among all routes (for all connections), the route with the minimum value of W_c^r is allocated next. Ties are broken arbitrarily, selecting the lowest ID of the source node and if necessary, then the lowest ID

of the terminal node.

5. After all node pairs have been allocated a route, a report is generated with:

   - the fraction of node pairs that could not be allocated a feasible route
   - the route allocated to node pairs with feasible routes
   - the utilization of all links
   - the number and identification of labels of every allocated route

The weight of a route was calculated using three different expressions, giving place to the three different heuristics:

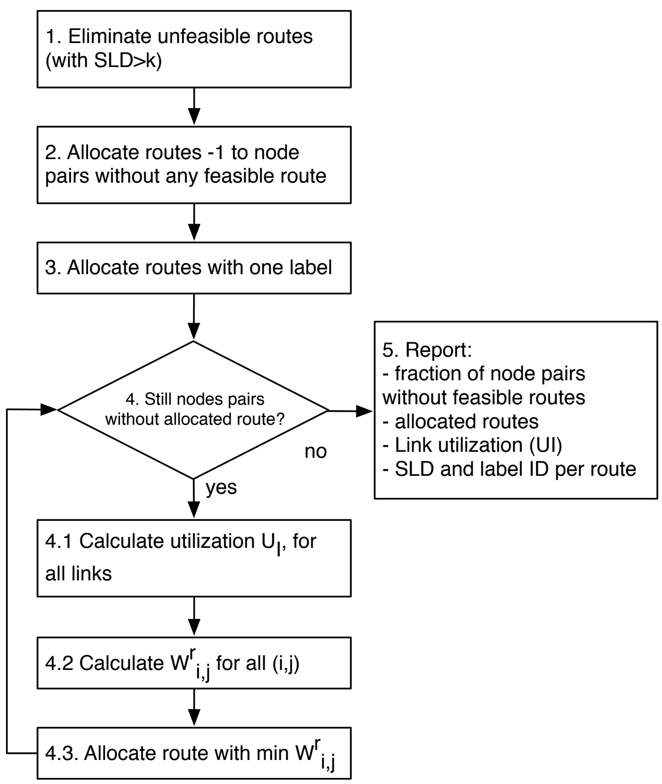


Figure 2: Flowchart of heuristic approach.

4.1. Heuristic 1 (H1):

$$W_c^r = \left[\sum_{l \in p_c^r} U_l\right] \cdot SLD_c^r \tag{15}$$

Where p_c^r is the r-th shortest route between nodes $s(c)$ and $t(c)$ and SLD_c^r is the SLD of route p_c^r. That is, Heuristic 1 selects the route whose sum of its link utilizations multiplied by the number of labels is minimum.

4.2. Heuristic 2 (H2):

$$W_c^r = \sum_{l \in p_c^r} U_l \tag{16}$$

That is, Heuristic 2 selects the route with minimum utilization, measured as the sum of the individual utilization of the links along the route.

4.3. Heuristic 3 (H3):

$$W_c^r = \max_{l \in p_c^r} U_l \tag{17}$$

That is, Heuristic 3 selects the route with the minimum maximum link utilization.

4.4. Local Search (LS) step

The final solution found by any of the three proposed heuristics can be improved by executing the following LS step:

1. Sort the network links from highest to lowest utilization. Store them in list L
2. For each link l in L:
    - 2.1 If there is an alternative route that decreases the utilization of link l, deallocate the previous assigned route and allocate this new one. Else, end the local search step.
    - 2.2 Sort again links in list L

4.5. Time complexity

The time complexity of the heuristic approach is determined by the step 4 (4.1−4.3) of the flowchart shown in Figure 2. The time complexity of calculating the utilization of each link (step 4.1) is $O(L)$ whilst the time complexity of evaluating the weight of each route (step 4.2) is $O(L \cdot N^2)$. Finding the minimum weight (step 4.3) is $O(N^2)$. As these steps are repeated for all routes, the time complexity of the heuristic approach is $O(N^4)$. The local search step, executed at the end of the heuristic, is $O(L \cdot \log L)$ and thus, does not modify the overall time complexity of $O(N^4)$ of the heuristic approach.

5. Results

In this section we present the results obtained by the ILP models and the heuristic approach assuming that every node pair in the network requires establishing a path.

5.1. ILP models

We first compare the results of the three ILP models to compare the capability of SR traffic routing. We compare three performance indicators: the maximum utilization α obtained by each model (Table 2), the average utilization of each link and the average length of the resulting connections (Table 3).

For all instances, we fix the capacity of each link to $K_l = 1$. Note that this is not realistic because we cannot route more flow than the capacity, but in this way the maximum utilization α represents the maximum number of routes using a link. For the SegmR model, we present the results forcing a maximum limit on the SLD to $\kappa = 3$ and $\kappa = 8$.

We also solve the problem for two types of demands. First, we solve an homogeneous case where all pairs of nodes have a demand of $d_c = 1$. From that solution, we construct an heterogeneous case in the following way: we compute the ShP solution, and for each pair of nodes we make the demand equal to the maximum utilization among links on this path, normalized such that the sum

Table 2: Maximum utilization α for each model

	Homogeneous Demand				Heterogeneous Demand			
	ECMP	ShP	SegmR $\kappa = 8$	SegmR $\kappa = 3$	ECMP	ShP	SegmR $\kappa = 8$	SegmR $\kappa = 3$
Grid 2x2	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
Grid 3x3	7.00	6.00	6.00	6.00	7.00	6.00	6.00	6.00
Grid 4x4	18.63	16.00	16.00	16.00	19.36	16.52	16.52	16.52
Grid 5x5	36.75	30.00	30.00	30.00	38.53	31.16	30.96	30.96
Eurocore	4.96	4.00	4.00	4.00	5.23	4.42	4.15	4.15
NFSNET	15.33	13.00	13.00	13.00	16.25	14.46	13.38	13.38
EON	25.00	18.00	18.00	18.00	29.31	22.52	20.70	20.70
UKNET	27.63	21.00	19.00	19.00	31.73	24.01	20.59	20.59
ITALNET	48.38	33.00	28.00	28.00	57.88	42.01	32.98	32.98
Arpanet	35.88	33.00	33.00	33.00	41.45	38.31	38.14	38.14
Eurolarge	131.60	88.04	66.00	66.00	162.43	117.19	74.36	74.38

of all demands is equal to that of the homogeneous case. In this way, pair of nodes routed through a highly utilized path in the homogeneous case will have an even larger demand in the heterogeneous case.

From Table 2 it can be seen that ECMP consistently produces a higher maximum utilization for all instances, obtaining up to 2 times the utilization required by segment routing for the Eurolarge topology. Comparing ShP with SegmR for homogeneous demand, results are similar except for three instances, namely UKNET, Eurolarge and ITALNET, where SegmR outperforms the shortest path approach obtaining a maximum utilization that is 17% smaller in average. In the heterogeneous case, SegmR outperforms ShP in all instances with a reduction on the maximum utilization of 11.4% in general, and 24.7% in the three instances recently discussed.

Results of Table 3 show that, even if SegmR leads to longer routes (the combination of up to κ shortest segments may result in non-shortest routes), the average utilization per link and the average length of connections increase only by 0.6% (2% for the three instances where ShP and SegmR differs). For the heterogeneous case, where longer routes are found for almost all instances, the average utilization increase by 3.4% and the average length increase by 2.7%.

Interestingly, SegmR solution only requires a SLD of three labels to obtain these results, except for the Eurolarge instance under heterogeneous demand. These results show the impact that segment routing can have, decreasing the maximum utilization by up to 36% (11.4% in average) increasing the average utilization and length by up to 6% (3.4% and 2.7% in average, respectively).

5.2. Heuristic approach

Table 4 shows the average and maximum value for the SLD and the time, expressed in milliseconds, required to calculate the stack of labels for all the possible shortest paths for each node pair per topology using Algorithm 1. The calculation of the stacks was done in a personal computer Intel Celeron Dual

Table 3: Other performance metrics of each solution

	Homogeneous Demand					
Network	**Mean utilization**			**Mean hop number**		
	ECMP	SHP	SEGMR	ECMP	SHP	SEGMR
Grid 2x2	2.00	2.00	2.00	1.33	1.33	1.33
Grid 3x3	6.00	6.00	6.00	2.00	2.00	2.00
Grid 4x4	13.33	13.33	13.33	2.67	2.67	2.67
Grid 5x5	25.00	25.00	25.00	3.33	3.33	3.33
Eurocore	3.48	3.48	3.48	1.58	1.58	1.58
NFSNET	9.29	9.29	9.29	2.14	2.14	2.14
EON	11.51	11.51	11.51	2.36	2.36	2.36
UKNET	13.49	13.49	13.68	2.50	2.50	2.54
ITALNET	17.03	17.03	17.31	2.92	2.92	2.97
Arpanet	17.19	17.19	17.19	2.81	2.81	2.81
Eurolarge	36.03	36.03	37.04	3.59	3.59	3.69
	Heterogeneous Demand					
Network	**Mean utilization**			**Mean hop number**		
	ECMP	SHP	SEGMR	ECMP	SHP	SEGMR
Grid 2x2	2.00	2.00	2.00	1.33	1.33	1.33
Grid 3x3	6.00	6.00	6.00	2.00	2.00	2.00
Grid 4x4	13.36	13.36	13.36	2.67	2.67	2.67
Grid 5x5	25.09	25.09	25.09	3.33	3.33	3.33
Eurocore	3.53	3.53	3.69	1.58	1.58	1.65
NFSNET	9.41	9.41	10.13	2.14	2.14	2.29
EON	11.90	11.90	11.93	2.36	2.36	2.37
UKNET	13.90	13.90	14.48	2.50	2.50	2.60
ITALNET	18.09	18.09	18.70	2.91	2.91	3.00
Arpanet	17.80	17.80	18.04	2.81	2.81	2.84
Eurolarge	37.64	37.64	39.89	3.59	3.59	3.77

Table 4: Segment list depth (SLD) for the studied networks.

Network	N	L	**SLD**		Execution time (ms)
			Average SLD	**Max SLD**	
2x2 grid	4	8	1.5	2	0.2
3x3 grid	9	24	2	4	2.1
4x4 grid	16	48	2.6	6	16
5x5 grid	25	80	3.4	8	112
6x6 grid	36	120	4	9	346
7x7 grid	49	168	4.2	9	777
Eurocore	11	50	1.7	3	3.4
NSFNet	14	42	1.4	2	2.7
EON	20	78	2.1	5	33
UKNet	21	78	2.0	4	31
ItalNet	21	72	2.0	4	19
ARPANet	20	62	1.6	3	9.8
Eurolarge	43	180	2.4	5	118

Core 2GHz, with 4 GB RAM. N and L denote the number of nodes and unidirectional links of each network, respectively.

Table 5 shows the results on the maximum link utilization for the three versions of the heuristic approach, before and after the local search step, for the homogeneous traffic case. The results obtained after the local search step are denoted as H1-LS, H2-LS and H3-LS. The maximum value of SLD was set to 3. The set of paths used as input for the heuristics were: the set of shortest paths ($e = 0$), the set of shortest paths plus all the paths with 1 extra hop ($e = 1$) and the set of shortest path plus all the paths with 2 extra hops ($e = 2$). For the sake of space, we include the results for values of $e = 1, 2$ only when better results than $e = 0$ are obtained. As a way of comparison, the results obtained with the ILP model SegmR for the segment routing case with $\kappa = 3$ are also included. Numbers in brackets correspond to the running time in seconds. It must be noted that the heuristic was executed in personal computer Intel Celeron Dual Core 2GHz with 4 GB RAM whilst the ILP model was solved in a cluster of computers. The lowest utilization obtained for each topology is highlighted in bold.

Table 6 shows the same results as Table 5 for the heterogeneous traffic case

Table 5: Maximum utilization achieved by the heuristic and the ILP approaches and their execution times for the homogeneous traffic case

Homogeneous demand								
Network	e	H1	H1-LS	H2	HS-LS	H3	H3-LS	ILP
Grid 2x2	0	**2** [0.0008]	**2** [0.00085]	**2** [0.0008]	**2** [0.00085]	**2** [0.0008]	**2** [0.00085]	2 [0.053]
Grid 3x3	0	7 [0.0068]	**6** [0.007]	7 [0.0066]	**6** [0.0068]	**6** [0.0063]	**6** [0.0064]	6 [0.52]
Grid 4x4	0	18 [0.10]	**16** [0.104]	**16** [0.10]	**16** [0.105]	18 [0.089]	17 [0.091]	16 [21]
Grid 5x5	0	34 [1]	**31** [1.03]	**31** [1]	**31** [1.006]	33 [0.88]	**31** [0.90]	30 [154]
Grid 6x6	0	58 [6.4]	**54** [6.5]	55 [6.4]	**54** [6.42]	58 [5.6]	**54** [5.7]	—
Grid 7x7	0	152 [24.5]	152 [24.5]	152 [24.5]	152 [24.5]	152 [21.4]	152 [21.4]	—
	1	97 [39.5]	87 [40.3]	94 [39.4]	**85** [40]	92 [36.3]	**85** [36.7]	
Eurocore	0	5 [0.009]	5 [0.0093]	5 [0.0093]	5 [0.0094]	**4** [0.0086]	**4** [0.0087]	4 [9.3]
NSFNet	0	14 [0.012]	**13** [0.0129]	14 [0.013]	**13** [0.014]	14 [0.012]	**13** [0.013]	13 [26]
EON	0	22 [0.13]	**18** [0.14]	22 [0.13]	**18** [0.14]	20 [0.13]	20 [0.14]	18 [16]
UKNet	0	24 [0.15]	24 [0.151]	23 [0.15]	23 [0.151]	22 [0.14]	22 [0.142]	19 [191]
	1	25 [1.7]	21 [1.71]	24 [1.7]	21 [1.71]	21 [1.4]	**20** [1.4]	
ItalNet	0	35 [0.15]	34 [0.153]	36 [0.15]	34 [0.156]	37 [0.14]	34 [0.145]	28 [208]
	1	33 [2.2]	29 [2.21]	34 [2.2]	29 [2.22]	32 [2.0]	**28** [2.01]	
ARPANet	0	35 [0.056]	**33** [0.059]	35 [0.057]	**33** [0.060]	36 [0.054]	**33** [0.060]	33 [142]
Eurolarge	0	100 [5.1]	90 [5.5]	99 [5.0]	89 [5.4]	92 [4.4]	90 [4.6]	
	1	93 [97]	76 [97.4]	91 [97]	75 [97.4]	81 [75]	70 [75.2]	66 [22191]
	2	92 [518]	71 [518.9]	90 [513]	74 [513.5]	71 [389]	**67** [389.2]	

Table 6: Maximum utilization achieved by the heuristic and the ILP approaches and their execution times for the heterogeneous traffic case.

Heterogeneous demand								
Network	e	H1	H1-LS	H2	HS-LS	H3	H3-LS	ILP
Grid 2x2	0	**2** [0.0009]	**2** [0.00098]	**2** [0.0009]	**2** [0.00098]	**2** [0.0009]	**2** 0.00099]	2 [0.01]
Grid 3x3	0	7 [0.007]	**6** [0.0074]	7 [0.007]	**6** [0.0073]	**6** [0.0069]	**6** [0.0071]	6 [0.63]
Grid 4x4	0	18.45 [0.1]	**16.71** [0.106]	17.37 [0.1]	16.77 [0.104]	17.88 [0.092]	17.01 [0.098]	16.52 [TLE]
Grid 5x5	0	35.04 [1.02]	31.60 [1.1]	32.20 [1.01]	**31.57** [1.02]	32.55 [0.98]	32.30 [0.99]	30.96 [TLE]
Eurocore	0	5.25 [0.010]	4.84 [0.011]	5.25 [0.010]	4.84 [0.011]	4.97 [0.095]	**4.42** [0.096]	4.15 [TLE]
NSFNet	0	15.21 [0.014]	14.24 [0.015]	15.22 [0.014]	14.24 [0.015]	15.22 [0.014]	**14.22** [0.015]	13.38 [1421.36]
	0	26.57 [0.14]	21.95 [0.16]	26.23 [0.14]	21.14 [0.16]	22.38 [0.13]	21.9 [0.131]	
EON	1	24.31 [1.4]	21.03 [1.41]	24.04 [1.4]	20.88 [1.41]	22.84 [1.3]	20.95 [1.305]	20.70 [TLE]
	2	23.9 [4.7]	21 [4.72]	24.62 [4.6]	**20.87** [4.61]	22.06 [4.5]	21.02 [4.51]	
	0	27.16 [0.15]	26.74 [0.151]	26.38 [0.14]	25.75 [0.142]	26.72 [0.14]	24.36 [0.141]	
UKNet	1	27.49 [1.7]	23.92 [1.72]	26.15 [1.7]	22.98 [1.73]	23.10 [1.4]	22.61 [1.406]	20.59 [3755.88]
	2	27.20 [5.9]	23.00 [5.93]	26.12 [5.9]	22.56 [5.93]	24.23 [5.2]	**22.35** [5.22]	
	0	44.52 [0.16]	42.15 [0.168]	44.91 [0.16]	42.22 [0.168]	42.99 [0.16]	42.11 [0.162]	
ItalNet	1	39.28 [2.2]	33.62 [2.24]	40.99 [2.2]	33.72 [2.23]	34.39 [2.2]	33.42 [2.21]	32.98 [550.59]
	2	40.1 [8.2]	34.1 [8.24]	39.91 [8.2]	33.61 [8.24]	33.44 [8.0]	**33.33** [8.008]	
ARPANet	0	40.65 [0.060]	38.87 [0.066]	39.45 [0.060]	38.87 [0.064]	41.47 [0.057]	38.45 [0.065]	38.14 [1200.95]
	1	41.48 [0.6]	38.43 [0.605]	40.39 [0.6]	**38.38** [0.607]	43.44 [0.55]	38.54 [0.57]	
	0	123.7 [5.1]	117.19 [5.13]	120.55 [5.0]	117.19 [5.02]	117.19 [4.6]	117.19 [4.605]	
Eurolarge	1	106.8 [95]	92.19 [95.4]	107.2 [94]	92.54 [94.4]	93.45 [87]	86.26 [87.2]	74.38 [TLE]
	2	108 [514]	82.3 [515.4]	104.9 [509]	86.7 [509.7]	76.67 [467]	**75.39** [467.4]	

[TLE] : Time limit of 86400 seconds exceeded. The solution provided is the best solution found by ILP at this time.

using the same traffic matrices (generated from the ILP solutions) described in section 5.1.

Before applying the local search step (LS), it can be seen that H3 performs the better with its maximum link utilization being in average 8% higher than the utilization obtained by solving the ILP model in the homogeneous case (17% and 14.5% higher in the cases of H1 and H2, respectively) and 7.3% in the heterogeneous case (18% and 15.8% higher in the cases of H1 and H2, respectively). This is an expected result, as H3 aims at selecting the routes with minimum maximum utilization and thus it should exhibit a better performance. In second place comes H2, as it aims at decreasing the whole route utilization. Lastly, H1 aims at decreasing the route utilization and the number of labels and thus, it achieves a lower number for the average SLD at expense of higher link utilization.

After applying the LS step a significant improvement on the maximum link utilization is achieved: H3-LS now obtains an average maximum link utilization of just 2.6% higher than the ILP model for the homogeneous case (6.7% and 7.3% for H1-LS and H2-LS, respectively) and 2.7% for the heterogeneous case (4.6% and 4.7% for H1-LS and H2-LS), without a significant increase in the execution time of the same instance.

Generally speaking, taking into account the results obtained by the heuristic with the lowest link utilization in each case, its performance is very good: in average, 0.8% and 2.6% higher than those obtained by solving the ILP model in the homogeneous and heterogeneous cases, respectively; with maximum differences of just 5.3% and 8.5% (UKNet, both types of traffic).

A significant advantage of the heuristic approach is its low execution time, with a minimum difference of two orders of magnitude with respect to the ILP model, even when the heuristic was executed in a personal computer (as opposed to a cluster in the case of the ILP model). In fact, for 6x6 and 7x7 grid networks the ILP model did not obtain results because of memory exhaustion. This difference is more significant in the heterogeneous case, where the ILP model stops after a time limit of 24 hours in five instances whilst the heuristic execution

Table 7: Mean link utilization and mean hop number for H1/H2/H3-LS for homogeneous and heterogeneous demand.

Homogeneous demand									
		Mean link utilization				**Mean hop number**			
Network	e	H1-LS	H2-LS	H3-LS	ILP	H1-LS	H2-LS	H3-LS	ILP
Grid 2x2	0	2	2	2	2	1.33	1.33	1.33	1.33
Grid 3x3	0	6	6	6	6	2	2	2	2
Grid 4x4	0	13.33	13.33	13.33	13.33	2.67	2.67	2.67	2.67
Grid 5x5	0	25	25	25	25	3.33	3.33	3.33	3.33
Grid 6x6	0	41.67	41.67	41.67	—	3.98	3.98	3.98	—
Grid 7x7	1	61.9	61.92	61.96	—	4.51	4.52	4.52	—
Eurocore	0	3.48	3.48	3.48	3.48	1.58	1.58	1.58	1.58
NSFNet	0	9.29	9.29	9.29	9.29	2.14	2.14	2.14	2.14
EON	0	11.51	11.51	11.51	11.51	2.36	2.36	2.36	2.36
UKNet	1	13.78	13.81	15.24	13.68	2.56	2.56	2.83	2.54
ItalNet	1	17.5	17.48	18.63	17.31	2.99	3.00	3.19	2.97
ARPANet	0	17.19	17.19	17.19	17.19	2.81	2.81	2.81	2.81
Eurolarge	2	37.82	37.29	42.57	37.04	3.77	3.72	4.24	3.77
Heterogeneous demand									
		Mean link utilization				**Mean hop number**			
Network	e	H1-LS	H2-LS	H3-LS	ILP	H1-LS	H2-LS	H3-LS	ILP
Grid 2x2	0	2	2	2	2	1.33	1.33	1.33	1.33
Grid 3x3	0	6	6	6	6	2	2	2	2.00
Grid 4x4	0	13.36	13.36	13.36	13.36	2.67	2.67	2.67	2.67
Grid 5x5	0	25.09	25.09	25.09	25.09	3.33	3.33	3.33	3.33
Eurocore	0	3.53	3.53	3.53	3.69	1.58	1.58	1.58	1.65
NSFNet	0	9.51	9.51	9.51	10.13	2.14	2.14	2.14	2.29
EON	2	12.48	12.43	14.21	11.93	2.47	2.46	2.82	2.37
UKNet	2	14.69	14.63	15.62	14.48	2.64	2.63	2.81	2.60
ItalNet	2	19.22	19.27	22.61	18.70	3.10	3.10	3.63	3.00
ARPANet	1	17.85	17.88	18.36	18.04	2.81	2.82	2.9	2.84
Eurolarge	2	40.51	39.65	44.4	39.89	3.83	3.75	4.22	3.77

times did not increase significantly with respect to the homogeneous case.

Regarding the mean link utilization and the average number of hops per connection, results are shown in Table 7. For the heuristic approach, only the results obtained after executing the LS step are shown, for the value of e for which the lowest maximum link utilization was obtained. The ILP results for $\kappa = 3$ are included as a way of comparison.

From the table it can be seen that the better performance of H3-LS in terms of the maximum link utilization comes at expense of longer routes and higher mean link utilization. In fact, for the homogeneous case H3-LS achieves a 2.8% longer routes and 5.7% higher mean link utilization than the ILP, compared to just 0.13/2.8% and 0.28/2.7% of H1-LS and H2-LS, respectively. A similar behaviour is observed in the heterogeneous case.

6. Conclusions

In this study, effective ILP models and heuristics are proposed and utilized to assess the traffic engineering performance of packet networks exploiting the segment routing (SR) technology. ILP models provided optimal results for low to medium-size networks. For larger and highly-mesh networks, feasible solutions were achieved only through heuristics.

Interesting considerations have been derived by the analysis of the obtained results.

First, we focused on the default SR behavior of exploiting equal cost multiple paths (ECMP). Such default behavior may drive the use of routes presenting inadequate quality of service (e.g., excessive latency) or may require more expensive hardware capabilities since it is necessary to guarantee per-flow forwarding to avoid packet misordering. Moreover, we showed that SR performance is highly dependent on the considered traffic scenario and topology, generally determining higher network resource utilization with respect to cases where ECMP is avoided and just single shortest routes are selected. For these reasons, the use of the default SR behavior has to be carefully considered.

On the other hand, when the default behavior of SR is avoided, and only strict routes are selected, it is a common understanding that significant scalability issues may occur in SR due to the use of larger segment list depth (SLD) values, i.e. the number of required labels to be stacked at ingress nodes. Quite surprisingly, the analysis carried out in this study has not confirmed such drawback. Indeed, by properly computing the segment lists to configure, it is possible to achieve very effective TE solutions just using very limited values of SLD. For example, with a maximum SLD value equal to three (a value already typically supported by most of the commercially available MPLS routers), it was possible to achieve, for all considered networks, the overall optimal resource utilization. Indeed, no improvements were experienced by relaxing such constraint and enabling larger SLD values. The possibility to rely on low SLD values also drives the additional positive effect of introducing very limited packet overhead with respect to traditional MPLS deployments, which typically exploit just a single label.

Acknowledgments

We are grateful for the time devoted by the anonymous reviewers to read and provide thoughtful and insightful comments on our work, which improved its quality and the applicability of results. Financial support from Fondecyt project 1130388 is also gratefully acknowledged.

[1] C. Filsfils, et al., Segment routing architecture, draft-filsfils-spring-segment-routing.

[2] C. Filsfils, et al., Segment routing with MPLS data plane, draft-filsfils-spring-segment-routing-mpls.

[3] P. Psenak, et al., OSPF extensions for segment routing, draft-psenak-ospf-segment-routing-extensions-05.

[4] D. Cai, A. Wielosz, S. Wei, Evolve carrier Ethernet architecture with SDN and segment routing, in: 2014 IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014, pp. 1–6.

[5] G. Swallow, From tag switching to SDN and segment routing MPLS: an enduring architecture, in: Proc. MPLS SDN World Congress, 2014.

[6] S. Bidkar, A. Gumaste, P. Ghodasara, A. Kushwaha, J. Wang, A. Somani, Scalable segment routing a new paradigm for efficient service provider net-

working using carrier ethernet advances, IEEE/OSA Journal of Optical Communications and Networking 7 (5) (2015) 445–460.

[7] F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti, P. Castoldi, Efficient label encoding in segment-routing enabled optical networks, in: 2015 International Conference on Optical Network Design and Modeling (ONDM), 2015, pp. 34–38.

[8] A. Giorgetti, P. Castoldi, F. Cugini, J. Nijhof, F. Lazzeri, G. Bruno, Path encoding in segment routing, in: 2015 IEEE Global Communications Conference (GLOBECOM), 2015, pp. 1–6.

[9] A. Sgambelluri, A. Giorgetti, F. Cugini, G. Bruno, F. Lazzeri, P. Castoldi, First demonstration of SDN-based segment routing in multi-layer networks, in: Optical Fiber Communication (OFC) Conference, 2015, pp. Th1A–5.

[10] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, P. Castoldi, Experimental demonstration of segment routing, J. Lightw. Technol. 34 (1) (2016) 205–212.

[11] A. Sgambelluri, A. Giorgetti, F. Paolucci, F. Cugini, P. Castoldi, Experimental demonstration of multi-domain segment routing, in: 2015 European Conference on Optical Communication (ECOC), 2015, pp. 1–3.

[12] A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini, P. Castoldi, Demonstration of dynamic restoration in segment routing multi-layer SDN networks, in: Optical Fiber Communication Conference, 2016, pp. Th4G–4.

[13] R. Bhatia, F. Hao, M. Kodialam, T. Lakshman, Optimized network traffic engineering using segment routing, in: 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 657–665.