Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 110 (2017) 312–319

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Future Networks and Communications
(FNC-2017)

Optimizing Segment Routing using Evolutionary Computation

Vítor Pereira[a], Miguel Rocha[b], Pedro Sousa[a,*]

[a]*Centro Algoritmi / Dept. Informatics, University of Minho, Portugal*
[b]*Centre of Biological Engineering, Dept. Informatics, University of Minho, Portugal*

Abstract

Segment Routing (SR) combines the simplicity of Link-State routing protocols with the flexibility of Multiprotocol Label Switching (MPLS). By decomposing forwarding paths into segments, identified by labels, SR improves Traffic Engineering (TE) and enables new solutions for the optimization of network resources utilization. This work proposes an Evolutionary Computation approach that enables Path Computation Element (PCE) or Software-defined Network (SDN) controllers to optimize label switching paths for congestion avoidance while using at the most three labels to configure each label switching path.

Keywords: Segment Routing; Traffic Engineering; Link-State Routing; Evolutionary Computation.

1. Introduction

Traffic engineering (TE) encompasses distinct methods to efficiently allocate network resources. With emerging new technologies, TE methods evolved to meet user constraints, while increasing operators' benefits. As expected, there are a vast number of proposals which aim to achieve this goal, while preserving simplicity and scalability. Since Link-State routing (LSR) protocols were first proposed, network engineers have given them a particular attention as they intrinsically possess those two attributes. Indeed, they only require a set of weights assigned to each network link to compute the forwarding paths, shortest paths (SP), that minimize the sum of all link weights in the path.

Open Shortest Path First (OSPF)[1] and Intermediate System to Intermediate System (IS-IS)[2] are two common LSR protocols. To achieve a better usage of network resources, some OSPF/ IS-IS implementations use Equal Cost Multi-Path (ECMP)[3]. When more than one shortest path exists to the same destination, ECMP enables to evenly split traffic along the next-hops on the paths. But, although LSR with EMCP provides a better distribution of traffic, it requires a suitable link weights configuration. This configuration, usually performed by a network administrator, is not easy, notably in the case of large scale networks and heavy traffic requirements. A typical strategy consists in setting link weights inversely proportional to their capacity, InvCap, which results in links with greater capacity receiving higher volumes of traffic. This strategy, although simple, is unable to deliver an optimal traffic distribution. Fortz and Thorup[4] showed that the weight setting problem is NP-hard, using a convex continuous function to evaluate networks

---

[*] Corresponding author. Tel.: +351-253-604-470 ; fax: +351-253-604-471.
E-mail address: pns@di.uminho.pt (Pedro Sousa).

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.06.100

congestion based on link usage levels. Employing this objective function, the authors were able to obtain weights that achieve a better traffic distribution. This convex function, which penalizes overutilized links, has been used in many TE studies resorting to distinct optimization heuristics, such as Local Search[4], Evolutionary Algorithms[5], Simulated Annealing[6], or Particle Swarm Optimization[7], to optimize resource utilization and, also accommodating additional objectives such as delay, resilience to link failure and traffic mutability.

Even though LSR/ECMP with optimized weights enables a good distribution of traffic load, it can not deliver an optimal usage of network's resources, performing a few percent off[4]. The even splitting of traffic across multiple shortest-path routes performed by ECMP is an obstacle to an optimal utilization. In some cases, even with optimized configurations, some network links might not be used at all. To address this issue, unequal load balancing of traffic among outgoing links needs to be considered. Many proposals were able to implement unequal load balancing[8,9], but failed in preserving the simplicity and scalability expected from a routing protocol. The Distributed Exponentially-weighted Flow SpliTting (DEFT)[10], on the other hand, is able to forward traffic along links on non-shortest-paths and induce unequal traffic splitting, without losing in simplicity and scalability. By solely relying on link weights configuration, DEFT assigns flows to a next-hop with a probability that decreases exponentially with the extra length of the path relative to the shortest path. The authors experimentally showed that DEFT was able to offer a better utilization of resources than the one provided by optimized OSPF or IS-IS with ECMP. Additionally, the Penalizing Exponential Flow-spliTting (PEFT)[11] emerges as an evolution of DEFT. The main difference is that, in terms of flow splitting, DEFT is a link-based protocol, whereas PEFT is a path-based protocol, that is, PEFT extends DEFT by including a set of variables which encode path information on the traffic splitting ratios computation. PEFT also brought an important result, as the authors formally proved that, by forwarding a portion of traffic along links on non-shortest-paths, it is possible to achieve an optimal utilization of resources.

While traditional TE approaches, as those described earlier, focus on the improvement of routing configurations, the novel concept of Software-Defined Networking (SDN)[12] opened opportunities for devising innovative TE mechanisms. The SDN architecture decouples the network control and forwarding functions into two separate planes, a control and a data planes. This enables network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The formerly static networks, as those with a link-state routing protocol, now become intelligent, responsive, and centrally controlled. The open application programming interface (API) provided by SDN implementations, such as OpenFlow[13], simplifies network operations, as instructions and rules can directly be installed on the existing flow tables. SDN also makes possible to retrieve information about the network state and improve traffic matrix estimation procedures[14], and be more responsive to traffic variations.

Segment routing (SR)[15] has been recently proposed as a SDN technology with relevant simplifications to the data and control plane operations. SR decouples edge-to-edge routing paths into smaller paths called segments. Analogously to Multi-Protocol Label Switching (MPLS)[16], SR uses a path-label mechanism to specify the route that packets must take through a network. Thus, a route is here uniquely defined as a list of segment IDs (SIDs). This provides enhanced packet forwarding, while minimizing the need for maintaining awareness of mass volumes of network state. In fact, instead of pushing a flow entry to all the switches in a path, SR pushes a label stack, SIDs, into the packet header when it arrives at the ingress node. The problem of determining optimal configurations for SR has recently been addressed, through offline, online and traffic oblivious optimization algorithms for maximum link utilization[17], but also using an integer programming algorithm to assess the traffic engineering of packet networks[18]. In both cases, results showed that SR does not require a high-label stack depth for SR to perform well.

In this context, this work proposes an Evolutionary Computation based TE approach for Single Adjacency Label Path Segment Routing (SALP-SR) that, using at the most three labels to configure edge-to-edge routing paths, is able to optimize the utilization of network resources. Furthermore, we also explore the utilization of a split computation parameter to respond to variations on traffic requirements and avoid congestion in parts of the network.

2. Segment Routing Traffic Engineering

Segment routing, proposed by the IETF, is a simple, scalable and highly flexible platform. Based on the source routing paradigm, SR is a label-switching technique that allows edge routers to steer a packet through the network using a list of segments, each identifying a topological or service instruction. While a service instruction pinpoints a service, provided by a node, where a packet should be delivered, a topological instruction specifies a path across which a packet should be forwarded. Any edge-to-edge path can be represented as a combination of topological segments. Although SR is very similar to MPLS, it neither requires a Resource Reservation Protocol (RSVP) or a Label Distribution Protocol (LDP), making it much more flexible. The distribution of labels is assured by an extension to the Interior Gateway Protocol (IGP), IS-IS or OSPF, or centrally managed by a Path Computation Element (PCE) or SDN controller. The path information required for a packet to traverse the network is a list of segments, allocated in the packet's header at the provider's edge (PE). In our proposal, only two types of SIDs are required:

- **Node SID**: identifies a network node (assigned for each router in the network). It is a form of Prefix SID, a SID that contains an IP address prefix calculated by the IGP. A Node SID uses the node loopback address as prefix and is unique domain wide. Traffic traveling through a node segment is routed by the IGP to the destination.
- **Adjacency SID**: An Adjacency Segment represents a local segment (interface) to a specific SR node. Each router assigns a locally significant segment ID for each of its IGP adjacencies.

An important advantage of SR is that intermediate routers do not have to maintain any per-flow state. The intermediate nodes only need to know the globally distributed segment labels. Additionally, SR supports ECMP by design. Paths identified by node segments are IGP shortest-paths, and they intrinsically include all the ECMP paths to the node. These two features provide tremendous gains in network scalability.

Exploring these advantages, we propose an optimization mechanism for Segment Routing to deliver optimal traffic distribution resorting to label path configurations which contain at the most three segment IDs, with at the most one adjacency SID. This work only uses segment label paths with one, two or three SIDs to route traffic from edge-to-edge, independently of the size of the network and volume of traffic. We claim that it is possible to attain a near optimal utilization of the available resources, using only label paths with the following alternatives of configuration formats:

- **1-Segment**: The label path is configured with a unique Node SID, the SID of the destination node t, $[\mathtt{t}]$;
- **2-Segment**: The label path is configured with a Node SID and an Adjacency SID. In this case there are two possible configurations: 1) $[(\mathtt{s},\mathtt{u});\mathtt{t}]$, the adjacency segment starts at the source node s, and the Node SID identifies the destination node t; 2) $[\mathtt{u};(\mathtt{u},\mathtt{t})]$, the adjacency segment ends at the destination node t, and the Node SID identifies the start node of the adjacency segment, u.
- **3-Segment**: The label path is configured with a Node SID, an Adjacency SID, and a Node SID,$[\mathtt{u};(\mathtt{u},\mathtt{v});\mathtt{t}]$, where (u,v) is the adjacency segment.

3. Traffic Engineering with Three-Segments Routing

3.1. Three-Segments Routing

Network infrastructures are constantly challenged by a growth on traffic volume that needs to be fairly distributed. To achieve this goal, we propose an optimization procedure for Segment Routing where solutions encompass: *1)* a link weight configuration for the link-state IGP that underlies the SR network, *2)* edge-to-edge SR label paths configuration as well as *3)* the split of traffic between parallel label paths to the same destination. A conceptual representation of the SALP-SR optimization model elements is represented in Figure 1. The optimization mechanisms here proposed are implemented in a freely available framework accessible at `http://darwin.di.uminho.pt/netopt`.

As depicted in Figure 1, the optimization is performed by a northbound framework which integrates a SR/Link-State simulator and an Evolutionary Algorithm (EA) optimization engine. The northbound framework interacts with a PCE/SDN controller that collects topology and traffic related information. During the optimization process, the distribution of traffic is achieved by resorting to DEFT or PEFT traffic splitting functions (described in Section 3.2). Complementary, the SR/Link-State routing simulator is used to evaluate the quality of the solutions provided by the EA, which are encoded as a set of link weights, from which SR label paths and traffic splits can also be computed.

The SR/Link-State routing simulator models the network as a graph $G(N,A)$, where N is the set of nodes and A the set of arcs. In this work, we use the normalized sum of link cost Φ proposed in[4] as primary comparison metric and main optimization objective. The derivative of Φ is defined as:

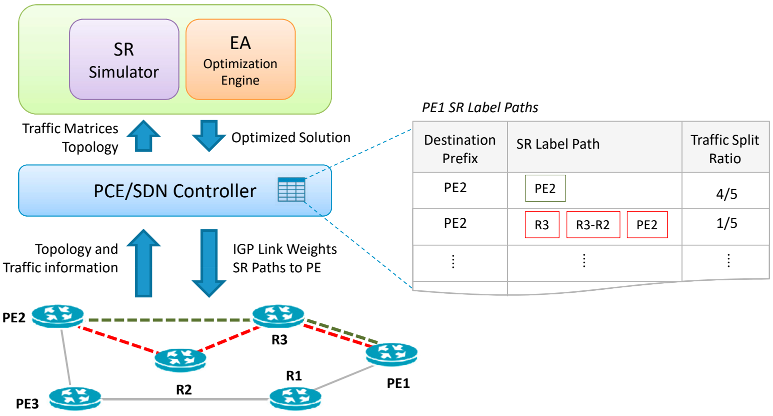


Fig. 1. Optimization Model Conceptual Architecture

$$\Phi'_a = \begin{cases} 1 & for \;\; 0 \leq u_a < 1/3 \\ 3 & for \;\; 1/3 \leq u_a < 2/3 \\ 10 & for \;\; 2/3 \leq u_a < 9/10 \\ 70 & for \;\; 9/10 \leq u_a < 1 \\ 500 & for \;\; 1 \leq u_a < 11/10 \\ 5000 & for \;\; u_a \geq 11/10 \end{cases} \tag{1}$$

where a is a topology link and u_a its utilization under traffic requirements D and a specific routing configuration.

Traffic demands are represented as a matrix (D) of edge-to-edge requirements that can be obtained from information gathered by the SDN controller[14]. We address the NP-hard problem of weight-setting with Evolutionary Algorithms (EA Optimization engine module of Figure 1). As observed, the solution attained by the northbound optimization framework will be installed in the network by the PCE/SDN controller. This involves the link weights configuration for the link-state IGP operation, and the SR label paths and traffic splits for the PE network elements operation.

3.2. DEFT/PEFT Traffic Splitting Functions

The Distributed Exponentially-weighted Flow SpliTting (DEFT) routing protocol assigns flows to a next-hop with a probability that decreases exponentially with the extra length of the path when compared with the shortest path. The distance from a node u to a node t, when traffic is routed thought a node v, can be expressed as $d^t_v + w_{u,v}$, where d^t_v is the shortest distance from the next-hop v to t, and $w_{u,v}$ is the weight of the link (u, v). The extra length of the path from u to t through v, when compared to the shortest path from u to t, is obtained by Equation 2, and denoted as $h^t_{u,v}$.

$$h^t_{u,v} = d^t_v + w_{u,v} - d^t_u \tag{2}$$

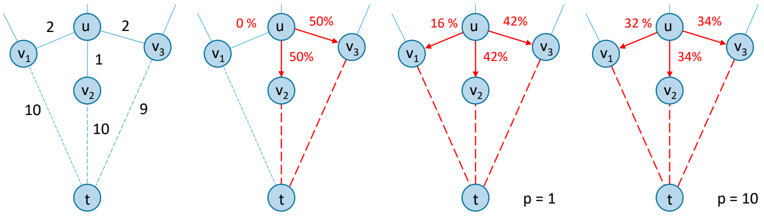
$$P\left(h^t_{u,v}\right) = \frac{\Gamma\left(h^t_{u,v}\right)}{\sum_{(u,i)\in A} \Gamma\left(h^t_{u,i}\right)}, \tag{3}$$

$$\Gamma\left(h^t_{u,v}\right) = \begin{cases} e^{-\frac{h^t_{u,v}}{p}}, & \text{if } d^t_v < d^t_u \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\Gamma_{DOWN}\left(h^t_{u,v}\right) = \begin{cases} e^{-\frac{h^t_{u,v}}{p}} \times \gamma^t_u, & \text{if } d^t_v < d^t_u \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

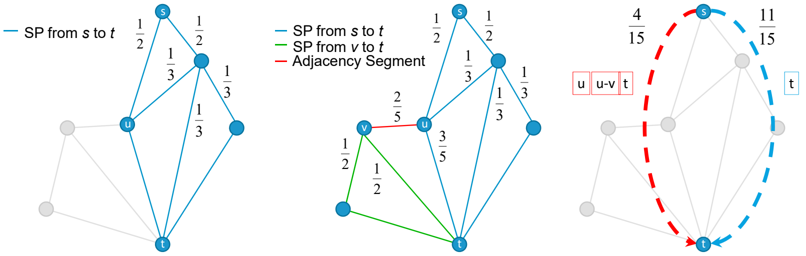
The flow proportion on the outgoing link (u, v) destined to t, at u, is computed by Equation 3, where the exponential function Γ, Equation 4, maps the extra length $h^t_{u,v}$ of a path into $[0; 1]$. For a node u in a shortest path from s to t, if all possible next-hops are also on shortest paths, all the extra lengths $h^t_{u,v}$ are equal to 0, and the portions of traffic $P\left(h^t_{u,v}\right)$ are provided by ECMP. The condition $d^t_v < d^t_u$ (Equation 4) ensures that traffic is forwarded towards the destination, preventing loops.

The parameter p was thought as a constant to scale the penalizing function Γ to a range $[w_{\min}, w_{\max}]$ of possible link weights. In this work we consider a default p value of 1 and explicitly define an utilization threshold of 1% beyond



(a) (b) (c) (d)

Fig. 2. DEFT Traffic Splitting Example



(a) (b) (c)

Fig. 3. Three-Segments Routing Example

which a link isn't used to forward traffic. We also will consider scenarios, Sect. 4.2, where the p values are distinct variables assigned to each node u, that can be optimized to improve traffic distribution. Figure 2 provides an example for DEFT traffic splitting and also highlight the effect of distinct values of p.

In the scenario shown in Figure 2, traffic needs to be routed from node u to node t. Considering the link weights configuration from Figure 2(a), the next-hops on a shortest path from u to t are v_2 and v_3. Therefore, using ECMP, traffic will be equally split and forwarded to both next-hops (Figure 2(b)). However, DEFT enables routers to forward traffic on non-shortest paths. All adjacent nodes on non-shortest paths that are at a shortest distance less than $d(u,t)(= 11)$ of t will be considered next-hops, as it is the case with v_1, $d(v_1,t) = 10$. For $p = 1$, 16% of traffic for t, arriving at u , will be forwarded to v_1 (Figure 2(c)). For $p = 10$, this percentage increases to 32%, and v_1 receives almost one third of the traffic (Figure 2(d)). The remaining traffic is equally split between the shortest paths next-hops.

The Penalizing Exponential Flow-spliTting (PEFT) mechanism adds path information, as real values γ_u^t, to the DEFT splitting function, Equation 5. In this work we considered a Downward PEFT implementation[11], where the γ_u^t values are obtained by solving a linear equation system for each destination t.

3.3. Three-Segments Routing Traffic Engineering

Figure 3(a) presents part of a network topology, configured with OSPF or IS-IS, that corresponds to the shortest path tree from s to t, and the corresponding traffic splits. The traffic that arrives at u is $2/3$ $(1/2 + 1/2 \times 1/3)$ of the traffic with destination t, which would be forwarded along the link (u,t). If (u,t) is unable to accommodate such volume, packet losses and increased delays would occur. To prevent such scenario, and resorting to SR, a portion of the traffic can be forwarded along adjacency segments that do not belong to any shortest path from u to t, 3(a). By optimizing the distribution of traffic, using PEFT (or DEFT) and the EA optimization engine (of Figure 1), a (near) optimal solution for traffic distribution can be obtained, and translated into an IGP link weights configuration and traffic splitting ratios, Figure 3(b). To accomplish the SR TE solution, two label paths need to be configured: 1) $[t]$, where traffic is forwarded using the installed IGP, 2) $[u,(u,v),t]$, where traffic is forwarded from s to u by the IGP, across the adjacency segment (u, v), and from v to t by the IGP along the existent shortest paths. In a sense, this can be understood as a load balancing of traffic between shortest paths.

$$F(s,t,u,v) = \begin{cases} P\left(h^t_{u,v}\right), & u = s \\ P\left(h^t_{u,v}\right) \times \sum_{p \in \mathcal{P}^{s,u}} \left(\prod_{(i,j)\in p} P\left(h^t_{i,j}\right)\right), & u \neq s \end{cases} \tag{6}$$

$$F(s,t) = 1 - \sum_{(u,v)\in\mathcal{A}^{s,t}} F(s,t,u,v) \tag{7}$$

The fraction of traffic to be assigned to each label path between s and t can be easily computed using Equations 6, for the label path containing an adjacency segment (u, v), and 7, for a label path that only contains the destination node t SID. In the equations, $\mathcal{P}^{s,u}$ represents the set of all shortest paths from s to u, and $\mathcal{A}^{s,t}$ is the set of all non shortest path adjacency segments on the shortest paths from s to t. By applying both equations to the provided example, the traffic splitting ratios are $4/15$ for the label path with the adjacency SID, and $11/15$ to the path with only the node SID of t, Figure 3(c). It is important to highlight that there are differences between the devised SR oriented proposal and the original DEFT/PEFT methods. Contrary to our proposal, the last do not ensure that a flow traverses at most one non shortest path link, and ultimately, a flow could travel solely using non-shortest paths. The solutions provided by this proposal are expected to have equivalent quality to those offered by DEFT or PEFT when the traffic necessities are well distributed. Furthermore, SR introduces advantages, such as less network state information to be maintained in intermediate routers, and the ability to perform per-flow TE without any change to the installed configuration.

4. Experiments and Results

4.1. Experiments Settings

The validation of the method used randomly generated network topologies (Table 1), varying in size and degree. For each, we produced traffic demand matrices D with distinct levels α of expected link utilization. For each pair of nodes (s, t), the traffic needs are modeled by Equations 8 and 9, where R is a random number in $[0, 1]$, $dst(s, t)$ the Euclidean distance between both nodes, $\overline{c_a}$ the average capacity of all links, $|A|$ the number of links and $hops(s, t)$ the minimum number of hops between s and t. Thus, traffic requirements are inversely proportional to distance.

Table 1. Network Topologies used in the experiments

Topology	Type	Nodes	Links
$Rand30_2$	Synthetic	30	110
$Rand30_4$	Synthetic	30	220
$Rand50_2$	Synthetic	50	194
$Rand50_4$	Synthetic	50	380

$$D(s,t) = \frac{R \times \delta}{dst(s,t)} \tag{8}$$

$$\delta = \alpha \times \overline{c_a} \times |A| \times \sum_{(s,t)\in N^2} \frac{hops(s,t)}{dst(s,t)} \tag{9}$$

The used metric, *Congestion*, is the normalized sum of link cost Φ, Equation 1. It is of notice that when this cost equals 1, all loads are below $1/3$ of the link capacity, and when all arcs are exactly full the value of congestion is 10 2/3. The link weights are integers taken from $[1; 20]$, except when PEFT is used, where weights are ranged in the $[10,001; 10,020]$ interval. All optimizations were performed by an Evolutionary Algorithm optimization engine.

4.2. Optimization Results

Representative results of the experiments are presented in Table 2. For comparison purposes, we include congestion values obtained with InvCap weight configuration, together with those from OSPF with ECMP optimized weights (OSPF Opt.) and for DEFT and PEFT routing protocols, also with EA-optimized configurations. The SALP-SR results are identified as SR-D and SR-P according to the used splitting function, respectively DEFT and PEFT.

The results show that, although the traffic distribution is not exactly the same, DEFT and SR-D solutions, as well as PEFT and SR-P, have equivalent quality. In all cases, SR-D and SR-P provide solutions with a better congestion than

OSPF/ECMP with optimized configurations. SR-D and SR-P are able to make use of all available links which does not always happen with OSPF/ECMP optimizations, and consequently provides a better resilience to traffic variations. We also observe that, in most cases, SR-D and SR-P offer configuration solutions with equivalent levels of congestion and the additional computational time required by PEFT might not be justified.

Table 2. Optimized Congestion Values

Topology	D Level	InvCap	OSPF Opt.	DEFT	SR-D	PEFT	SR-P
$Rand30_2$	0.3	15.60	1.40	1.36	1.36	1.36	1.36
	0.5	494.92	3.77	2.94	2.94	2.74	2.74
$Rand30_4$	0.3	323.76	1.69	1.59	1.59	1.59	1.59
	0.4	717.95	5.17	2.94	2.94	2.94	2.94
$Rand50_2$	0.3	437.70	1.68	1.61	1.61	1.61	1.61
	0.4	474.44	2.13	1.89	1.89	1.88	1.88
$Rand50_4$	0.3	156.04	3.91	2.49	2.49	2.38	2.38
	0.4	486.09	34.70	22.27	22.27	21.12	21.12

Edge-to-edge traffic necessities vary over time, and although an optimized configuration provides some level of resilience to those variations, it may occur that the installed configurations are no longer suited for networks to accommodate the new traffic requirements. In some cases, by implementing per-flow TE, it is possible for a PCE to find a routing label path with sufficient available capacity to sustain and steer traffic to its destination. Another possibility is to install a new routing configuration which would imply a new link weight configuration with consequent impact on the network IGP. A better approach consists in adjusting the traffic splitting between parallel paths. Although this solution doesn't support all possible variations, it can be a good temporary fix.

DEFT and PEFT use a splitting ratios computation parameter that can be used to improve the division of traffic between label paths. In Section 3.2, we showed how different p values in Equation 4 impact the splitting of traffic. By assigning distinct p values to each node, it is possible to alter the amount of traffic to be forwarded along adjacency segments and shortest paths. For an installed configuration of link weights and SR label paths, we propose to adjust traffic splitting between label paths by only optimizing the values of p for a given new traffic matrix meanwhile computed by an SDN controller module. Results from this approach, that only consider the DEFT splitting function, are presented in Table 3 and include three metrics: congestion, the average link utilization (ALU) and the maximum link utilization (MLU). Each topology is initially optimized for a traffic matrix D_0 (Opt.), and traffic variations for two distinct scenarios are represented as traffic matrices D_1 and D_2. For each of those, we present the measures obtained with the initial configuration (Before) and those obtained after the optimization of the p values (After). For comparison, we also include the measures for D_1 and D_2 with optimized configurations. The optimization was implemented using Evolutionary Algorithms where the p values are taken within the range $[0.01, 10.0]$ of real values.

By optimizing each node p value, SALP-SR is able to address traffic demand variations. As observed in Table 3, such strategy introduces a clear improvement on the traffic splitting and congestion. In a significant number of

Table 3. p Values Optimization

Measure	$Rand30_2$							$Rand30_4$						
	D_0	D_1			D_2			D_0	D_1			D_2		
	Opt.	Before	After	Opt.	Before	After	Opt.	Opt.	Before	After	Opt	Before	After	Opt.
Cong.	1.36	23.26	1.49	1.37	15.70	3.52	1.82	1.89	14.70	2.31	1.88	22.61	3.28	2.20
ALU (%)	33	37	34	33	43	44	42	35	34	35	33	40	41	40
MLU (%)	71	153	98	88	122	110	90	90	118	102	89	125	105	101
Measure	$Rand50_2$							$Rand50_4$						
	D_0	D_1			D_2			D_0	D_1			D_2		
	Opt.	Before	After	Opt.	Before	After	Opt.	Opt.	Before	After	Opt	Before	After	Opt.
Cong.	1.69	2.42	1.83	1.78	16.39	7.57	2.00	1.90	4.35	2.32	1.78	9.48	3.25	1.80
ALU (%)	36	37	38	38	38	38	38	25	28	29	28	30	32	30
MLU (%)	94	107	98	97	144	119	100	90	123	99	91	126	109	90

scenarios the network operates now in an acceptable working region, while before it was on a congested situation (values above 10 2/3) or very close to it. Although this strategy always introduces improvements to the congestion, in some cases it's not sufficient. But, as the amount of time required to obtain optimized p values is relatively short, this approach can be evaluated in parallel with traffic necessities updates, enabling to assess if additional measures need to be implemented. For example, the p values for the 30 nodes topologies are optimized in less than one minute, in a regular personal computer, and the 50 nodes topologies p values in around two and a half minutes.

5. Conclusions

This work proposed a TE method, entitled as SALP-SR, able to attain near optimal resources utilization on Segment Routing networks using label path configurations which use at the most three segments. Using DEFT or PEFT splitting functions to leverage traffic distribution, SR can achieve a configuration that optimally distributes traffic on the available resources without compromising simplicity and scalability. By providing an SR network with an optimized congestion, additional TE tasks can be implemented without the burden of performing complex management tasks. We also proposed a method to address traffic variations when the network is no longer able to assure a good performance. By optimizing parameters assigned to each node, it is possible to make adjustments to the traffic splitting between parallel paths and respond to a number of new traffic conditions, while maintaining label path configurations.

Acknowledgments

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.

References

1. Moy, J.: OSPF Version 2. RFC 2328 (Standard). Updated by RFC 5709 (1998)
2. Gredler, H., Goralski, W.: The Complete IS-IS Routing Protocol, Springer (2005)
3. Hopps, C.: Analysis of an Equal-Cost Multi-Path Algorithm. IETF RFC 2992, https://tools.ietf.org/html/rfc2992 (2000)
4. Fortz, B.,Thorup, M.: Internet Traffic Engineering by Optimizing OSPF Weights. In Proceedings of IEEE INFOCOM, pages 519-528 (2000)
5. Pereira, V., Rocha. M.,Cortez, P., Rio, M., Sousa, P.: A Framework for Robust Traffic Engineering using Evolutionary Computation, 7th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2013), Barcelona, Spain, Springer, LNCS 7943, pp. 2-13 (2013)
6. Ben-Ameur,W., Gourdin, E., Liau, B., Michel, N.: Optimizing administrative weights for efficient single-path routing. In Proc. of networks (2000)
7. Mohiuddin, M., Khan, S., Engelbrecht, A.: Fuzzy particle swarm optimization algorithms for the open shortest path first weight setting problem. Applied Intelligence 45, 3, 598-621 (2016)
8. Srivastava, S., Agrawal, G., Pioro, M., Medhi, D.: Determining link weight system under various objectives for OSPF networks using a Lagrangian relaxation-based approach. IEEE Trans. Netw. Serv. Manage., vol. 2, no. 1, pp. 918, (2005)
9. Sridharan, A., Guerin, R., Diot, C.: Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks. IEEE/ACM Trans. Netw., vol. 13, no. 2, pp. 234247, (2005)
10. Xu, D., Chiang, M., Rexford,J.: DEFT: Distributed exponentially-weighted flow splitting. *Proc. IEEE Conf. Comput. Commun.*, pp. 7179 (2007).
11. Xu, D., Chiang, M., Rexford, J.: Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering. IEEE/ACM Trans. Netw. 19, 6, 1717-1730 (2011)
12. Feamster, N., Rexford, J., Zegura, E.: The road to SDN. *Queue*, 11(12):20:20–20:40 (2013)
13. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38, 2, 69-74 (2008)
14. Hark, R., Stingl, D., Richerzhagen, N., Nahrstedt, K., Steinmetz, R.: DistTM: Collaborative traffic matrix estimation in distributed SDN control planes," 2016 IFIP Networking Conference (IFIP Networking) and Workshops, Vienna, pp. 82-90 (2016)
15. Filsfils, C., Nainar, N. K., Pignataro, C., Cardona, J. C., Francois, P.: The Segment Routing Architecture. IEEE Global Communications Conference (GLOBECOM), San Diego, CA, pp. 1-6 (2015)
16. Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., McManus, J.: Requirements for Traffic Engineering Over MPLS. RFC 2702 (Informational) (1999).
17. Bhatia, R., Hao, F., Kodialam, M. S., Lakshman, T. V.: Optimized network traffic engineering using segment routing. INFOCOM pp. 657-665, : IEEE. ISBN:978-1-4799-8381-0 (2015)
18. Moreno, E., Beghelli, A., Cugini, F.: Traffic engineering in segment routing networks. Computer Networks, 114, pp. 23-31 (2017)