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