On the Benefits of Loops for Segment Routing Traffic Engineering

Alexander Brundiers
Institute of Computer Science
Osnabrück University
Osnabrück, Germany
brundiers@uos.de

Timmy Schüller
Deutsche Telekom Technik GmbH
Münster, Germany
timmy.schueller@telekom.de

Nils Aschenbruck
Institute of Computer Science
Osnabrück University
Osnabrück, Germany
aschenbruck@uos.de

***Abstract*—Over the recent years, Segment Routing (SR)-based Traffic Engineering (TE) received more and more attention in the research community. However, what has been mostly neglected so far is its capability to configure looping forwarding paths that visit nodes or even edges multiple times. In this paper, we show that, against intuition, the configuration of such loops can inherit (in some occasions significant) benefits with regards to common TE objectives if Equal Cost Multipath (ECMP) is used. This is not only illustrated on small theoretical examples but also confirmed for 2SR with real-world data from the backbone network of a Tier-1 Internet Service Provider, as well as other publicly available topologies.**

I. Introduction

Over the last years, global Internet traffic has increased significantly and this trend is expected to carry on in the foreseeable future [4]. To keep up with these growing demands is one of the major challenges for Internet Service Providers (ISPs) across the globe. An inevitable requirement for this is the continuous increase of capacity by physically upgrading and expanding existing infrastructure. However, this is an expensive and time-consuming task. Therefore, many ISPs complementarily deploy some form of Traffic Engineering (TE). It allows for a more efficient utilization of existing infrastructure by distributing traffic over the available resources.

There are many different approaches to TE. A fairly recent one focuses on the application of Segment Routing (SR) [6]. It is based on the idea of adding waypoints, so called segments, to a packet, that have to be visited before heading to its original destination. The path to each of these segments is determined by the used Interior Gateway Protocol (IGP). Since most IGPs deploy some form of Shortest Path Routing (SPR), an SR path can also be interpreted as a concatenation of multiple shortest paths. This allows for precise control over a packets path through the network, while introducing significantly lower overhead than comparable traffic steering approaches, like Multiprotocol Label Switching (MPLS).

If the goal is a more efficient utilization of network resources, it is self-explanatory that scenarios have to be avoided that contradict these objectives. Routing loops occupy valuable network resources and can lead to increased delay and even packet dropping [12]. For this reason, a lot of research is conducted towards their detection or prevention (e.g., [7] or [5, Ch. 9]). Most loops arise from inconsistencies between routing tables during the convergence of the routing protocol. Since those disappear automatically after convergence, they are called *temporary* or *transient* loops. An even more detrimental, but also rarer form of loops are *persistent* loops. These are, for example, caused by misconfiguration of routers and do not resolve themselves but require active countermeasures.

When using standard SPR algorithms, like Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS), there are no forwarding paths that visit a node more than once while also ensuring correct delivery of the packet. The reason for this is that a packet is forwarded based solely on its destination. Since it does not change, the packet will be forwarded in the same way every time it passes a node, unless the routing table was altered in the meantime. As a result, there are no valid routing paths that visit a node more than once. Those would always result in an infinite loop.

When deploying SR, however, this observation does not hold anymore. Here, packets are forwarded based on the (interim-)destination specified by the currently active segment. Since it can change over time, the forwarding decisions of a router for a packet can also vary. As a result, it is theoretically possible to define forwarding paths that visit some nodes and even edges multiple times, without resulting in packets being trapped in an infinite loop.

In this paper, we analyze the theoretical characteristics and practical impact of such looping SR paths. We show that when Equal Cost Multipath (ECMP) is used, they can yield significant benefits with regards to common TE objectives like the minimization of the Maximum Link Utilization (MLU) or the reduction of the number of SR policies. This is done on the basis of small theoretical examples but also confirmed on real-world data from a Tier-1 ISP and the *Repetita* dataset [9].

The remainder of this paper is structured as follows. First, required background information is introduced, followed by a discussion of related work (Sections II and III). Looping SR paths and their impact on certain TE objectives are discussed in Section IV. After this, the used datasets are described and some preliminary examinations are carried out (Section V). Results of the actual evaluations are presented in Section VI and further discussed in Section VII. Finally, the paper is wrapped up with a summary of our findings and a discussion of possible future work in Section VIII.

II. Background

Before discussing looping SR paths and their impact on different TE objectives, we first need to give some more information on SR itself and its applications for TE.

A. The Segment Routing Architecture

SR, originally proposed in [6], is a network tunneling technique that implements the source routing paradigm. The eponymous idea behind SR is the subdivison of a packet's path into different sections by defining some kind of waypoints, called segments. These waypoints have to be visited in the given order before reaching the original destination of the packet. Depending on the nature of the related waypoint (e.g., nodes, adjacencies, or services) different types of segments are used. All of them are referred to by so called segment identifiers (SIDs). Applying a list of these SIDs to a packet allows for detailed control over its path through the network. Thereby, the sup-paths to the individual segments are determined by the respective IGP which most commonly relies on some form of SPR. Hence, an SR path can often be interpreted as a concatenation of multiple shortest paths.

When compared to similar technologies with traffic steering capabilities, like MPLS with Resource Reservation Protocol (RSVP)-TE, SR offers significant benefits. The state of an RSVP-TE tunnel has to be set-up and maintained by every associated node. In contrast, all information required by SR is encoded in the packet itself and an SR tunnel or policy only needs to be configured on the respective ingress node. This also obsoletes additional protocols, like Label Distribution Protocol (LDP) or RSVP, that are normally required for MPLS. As a result, the network overhead is significantly reduced. In addition to that, contrary to RSVP-TE, SR implicitly supports load balancing over ECMP.

B. Segment Routing-based Traffic Engineering

The fact that SR allows for fine-grained control over a packet's path through the network without significant overhead renders it an interesting candidate for TE. Depending on the specific use case, a wide variety of objectives can be pursued. One of the most common ones is the minimization of the MLU which will also be the main focus of this paper.

In theory, virtually arbitrary many SIDs can be added to a packet, but in practice this number is limited by the deployed routing hardware. While this restricts the level of detail at which a packet's path can be controlled, various publications (e.g., [3] or [16]) have shown that near-optimal results can often be achieved with the minimum number of just two segments. In [3], a linear programming formulation is presented that minimizes the MLU while using a maximum of two SIDs per packet. With regards to the number of segments, this problem is also referred to as 2SR in literature.

Since the 2SR optimization will be used in the evaluation section of this paper, its formulation is depicted in Problem 1. The objective is to minimize the MLU denoted by θ. The x_{ij}^k variables indicate the percentage share of the demand t_{ij} between nodes i and j that is routed over the intermediate

$$\min \theta \tag{1}$$

$$\text{s.t.} \quad \sum_k x_{ij}^k = 1 \qquad \forall (ij) \tag{2}$$

$$\sum_{ij} t_{ij} \sum_k g_{ij}^k(e) x_{ij}^k \leq \theta\, c(e) \qquad \forall e \tag{3}$$

$$x_{ij}^k \geq 0 \qquad \forall (ij) \tag{4}$$

Problem 1: 2SR formulation (inspired by [3]).

segment k. Equation (2) ensures that each demand is satisfied. Equation (3), together with the objective function, is responsible for minimizing the MLU. For every edge e, the $g_{ij}^k(e)$ values indicate the load that is put on e if a uniform demand is routed from i to j over the intermediate segment k. These values are constants that can be efficiently precomputed. All in all, the left side of the constraint equals to the amount of traffic that is put on edge e by the SR configuration represented by the x_{ij}^k variables. This load is then limited to the edges capacity $c(e)$ scaled by θ. By minimizing this scaling factor, an SR configuration with minimal MLU is computed.

While the results presented in [3] illustrate the theoretical capabilities of SR for TE, they suffer from one crucial problem. They ignore hard- and software-imposed limitations of current routing equipment. As a result, the computed solutions are (generally) not deployable in practice. This issue is tackled in [16], where the problem formulation from [3] is extended to also adhere to specific real-world constraints. One of those is the prohibition of arbitrary traffic splitting. In [3], a demand can be distributed arbitrarily over multiple SR policies while, in practice, typical routers only support equal splitting in predefined fractions. In [16], the number of SR policies per demand is limited to just one to be independent of the routing equipment and to keep the problem formulation simple.

Another important aspect addressed in [16] is the reduction of the number of SR policies required to implement a solution. Even though virtually no network overhead is introduced by an SR policy, network operators often want to deploy SR configurations with as few policies as possible for the sake of clarity and maintainability. To take these new requirements into account, [16] presents a new mixed integer program that functions as an extension of the original 2SR Linear Program (LP). After computing an optimal MLU with the standard 2SR formulation, a second optimization step is carried out that minimizes the number of required SR policies, while not surpassing this optimal MLU by more than a user-defined margin. Since SR policies were originally referred to as tunnels, this new LP is introduced as the Tunnel Limit Extension (TLE) and in the context of 2SR it is referred to as 2TLE.

The 2TLE formulation is given in Problem 2. Equations (6) and (7) are basically identical to Equations (2) and (3) of Problem 1, respectively. The only difference is the use of binary variables u_{ij}^k instead of the x_{ij}^k variables to ensure that a demand is not split over multiple SR policies. The major

$$\min \ \theta' \frac{1}{2\,\lambda\,\theta} + \sum_{k \neq j} u_{ij}^k \tag{5}$$

$$\text{st.} \quad \sum_k u_{ij}^k \ = \ 1 \qquad \forall ij \tag{6}$$

$$\sum_{ij} t_{ij} \sum_k g_{ij}^k(e)\, u_{ij}^k \ \leq \ \theta'\, c(e) \qquad \forall e \tag{7}$$

$$\theta' \ \leq \ \lambda\theta \tag{8}$$

$$u_{ij}^k \ \in \ \{0, 1\} \qquad \forall ijk \tag{9}$$

Problem 2: 2TLE formulation [16].

difference between the two problems lies in the objective function. Instead of minimizing the MLU, the goal is to minimize the number of SR policies. This can be done by minimizing the sum over the u_{ij}^k variables. SR policies that represent normal SPR can be ignored, as they do not need to be configured in practice. This can be implemented by excluding variables with identical k and j indices. To prefer solutions with lower MLU if the policy number is equal, an additional summand is added to the objective function. Finally, Equation (8) ensures that the new MLU θ' does not surpass the objective value θ of the first optimization step by more than a given percentage. This factor λ is also referred to as a *trade-off coefficient*, which basically weighs both objectives, MLU and policy minimization, against each other. For example, $\lambda = 1.2$ would allow the optimization to surpass the optimal MLU by as much as 20% to achieve a lower number of SR policies.

Based on the example of a Tier-1 ISP backbone network, Schüller et al. show that this new LP can be used to compute virtually optimal routing configurations that require significantly less SR policies than the 2SR formulation of [3].

While LPs guarantee optimal solutions, they often suffer from poor scalability and high resource demands. Therefore, research also focuses on alternative approaches to SR TE that significantly reduce computation times and memory demands, while still providing quite good solutions. In [8] a heuristic 3-SR algorithm is presented that is designed to optimize networks in a sub-second fashion. It is based on iteratively improving an initial solution via *Local Search*. Therefore, this approach will be referred to as Segment Routing Local Search (SRLS) in the remainder of this paper. SRLS is able to compute good solutions in significantly less time than alternative LP approaches. However, this comes at the price of no guarantees on the actual solution quality.

Another approach to heuristic online optimization is the Declarative and Expressive Forwarding Optimizer (DEFO) [10] [11], a network controller that allows for fast configuration and optimization of large networks. It is based on the concept on *Constraint Programming*. Similar to LPs, this can also be used to compute optimal solutions, but normally requires much time to do so. To speed up the solving process, the authors incorporate a technique called *Large Neighborhood Search* that allows for a fast exploration of a large solution space via heuristics. This significantly lowers the time and resource demands of the algorithm, up to a point where 3-SR configurations for networks with hundreds of nodes can be computed in a magnitude of seconds.

All of the SR TE approaches presented above rely solely on node segments. This results in a certain loss of expressiveness when compared to the use of additional adjacency segments because those allow for the definition of virtually every simple path. However, it also offers advantages, like lower problem complexity and implicit use of ECMP. In the remainder of this paper, we will also focus on SR with only node segments.

III. Related Work

There is a lot of research effort dedicated to the prevention of routing loops in different scenarios. Topology Independent Loop Free Alternate (TI-LFA) [5, Ch. 9] even utilizes SR to counteract transient loops during network convergence. However, to the best of our knowledge, there exist no publications that deal with a topic similar to our work.

There are a few publications that also acknowledge the capability of SR to specify forwarding paths that contain loops but do not focus on their relevance for TE. In [2], a network monitoring approach is presented that utilizes SR to send monitoring probes over cyclic paths, which allows for monitoring from a single vantage point. Here, SR's capability of routing packets over cycles is also exploited, but in a completely different scenario than the one we focus on.

When it comes to TE, SR's ability to specify looping paths has so far been mostly neglected. In [10] and [11] it is mentioned that cyclic paths can theoretically be defined but they are explicitly prohibited in the optimization. The only reference to a potential benefit of cyclic SR paths for certain TE objectives is given in [1, Ch. 5.3]. There, a short example is presented, in which a cyclic SR path achieves the optimal MLU. However, there also exists another, equally good acyclic solution which can replace the cyclic path. In contrast, we show that there actually are scenarios in which the optimal solution strictly requires the use of cyclic SR paths.

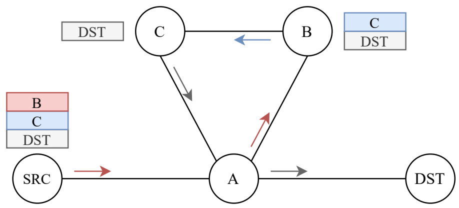
In the publications above, looping SR paths are mostly referred to as cyclic paths. We are not entirely content with this term. It might be confused with the similar sounding term "cycle" that inherits a slightly different graph-theoretic meaning. For this reason, we will instead refer to these paths as containing a Weak-Loop (WL) in the remainder of this paper.

IV. Weak-Loops

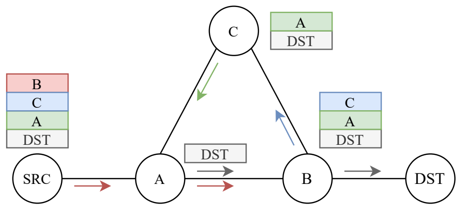
As already explained in Section I, SR allows for the definition of forwarding paths that contain WLs or, in other words, visit some nodes or even edges multiple times while still correctly delivering a packet to its destination. In the following, we further examine the properties of these WLs and illustrate that they can yield advantages with respect to certain TE objectives.

A. Definition

An SR path is said to contain a WL if it visits at least one node more than once. This definition can be further specified



(a) Node-WL: Repeatedly visited node (A).



(b) Edge-WL: Repeatedly visited edge ($A \rightarrow B$).

Figure 1: Examples for the two types of WLs.

to distinguish between two types of WLs. Those that actually visit an edge more than once and those that do not. We refer to them as *Edge*-WLs and *Node*-WLs, respectively. Figure 1 illustrates the two types of WLs and how SR can be used to configure them.

B. Impact on Traffic Engineering Objectives

If there is only a single shortest path between each pair of nodes, WLs can still be created (see Figure 1) but they will be of no benefit regarding the objective of MLU minimization. There will always be an at least equally good solution that does not incorporate a WL. This is intuitive, because every WL can be removed from an SR path without altering the utilization of any other links than those incorporated in the WL. For the latter, utilizations will only decrease.

Against intuition, however, this changes when there are multiple different shortest path between nodes and ECMP is utilized to load-balance between them. In this case, scenarios can be constructed for which the optimal MLU can only be achieved if some traffic is routed over a WL.

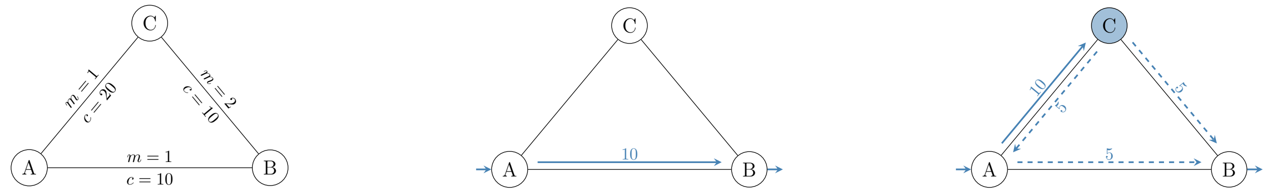
A minimalistic example for such a scenario is illustrated in Figure 2. Given the network topology depicted in Figure 2a, a demand of size 10 has to be routed from node A to node B. Edges are bidirectional with the m- and c-values denoting the metric and capacity of a link, respectively. There are basically only two ways to route the demand, either over the shortest path or via a 2SR path with intermediate node C. The first option would fully utilize the link from A to B and, hence, result in an MLU of 1.0 (see Figure 2b). The 2SR path over intermediate node C, however, would first route the whole demand to node C (indicated by the solid blue arrow in Figure 2c). There, the demand would be split by ECMP because there are two equally good paths from C to B, the direct edge and the detour back over A (indicated by the dashed blue arrows), resulting in the optimal MLU of 0.5. However, since A is visited twice, this optimal configuration features a WL. If WLs are prohibited, no MLU better than 1.0 can be achieved even if traffic can be distributed over multiple SR policies.

Besides improving the MLU, configuring WLs can also yield other advantages. One of those is a reduction of the SR policies required to implement a solution. This is exemplarily illustrated in Figure 3. Given the network topology in Figure 3a, two demands need to be routed, a first one of size 10 from A to B and a second one from D to B with size 5. Using SPR would result in overutilization of the network (MLU $= 2.0$). To prevent this, the demand needs to be redirected. This can be done with an SR path over intermediate node D (red path in Figure 3b). However, this would collide with the demand from D to B, that is routed over the direct edge between them. To resolve this issue, a second SR policy needs to be installed that steers the latter demand over node A (blue path in Figure 3b). Now, the optimal MLU of 1.0 is achieved at the expense of two SR policies. It should be easily comprehensible that this is the only (and hence optimal) way to prevent overutilization in the network, if WLs and the splitting of demands over multiple tunnels are prohibited. However, if configuring WLs is allowed, the same optimal MLU can be achieved with just one SR policy for the demand from A to B that uses C as intermediate segment (see Figure 3c). In this case, the traffic is first routed to C over node D (solid red arrows) where ECMP splits it over the direct link to B and back over D (dashed red arrows). As a result, only half of the demand passes over link D$\rightarrow$B. Therefore, a second SR policy is not necessary because the other demand (D to B) can still follow its standard shortest path over this edge without increasing the overall MLU. For completeness, we note that an even better MLU of 0.75 can be achieved if arbitrary traffic splitting is allowed. However, this would also require the use of WLs.

The surprising property of WLs to enable solutions with better MLU or fewer SR policies, most likely, results from their ability to mimic or at least approximate traffic splits that would otherwise be impossible to implement. This might become clearer, when looking back at the example in Figure 2. Optimal routing would be achieved by simply routing 50% of the traffic over the direct link from A to B and the other 50% over C to B. This, however, is prevented by the given metric configuration. By routing traffic through the WL, the theoretically necessary splitting is reconstructed at the expense of some additional utilization between the nodes C and A. However, since these links have sufficient capacity, this extra utilization does not impact the overall objective. Instead, it offers the possibility to exonerate other, highly utilized links.

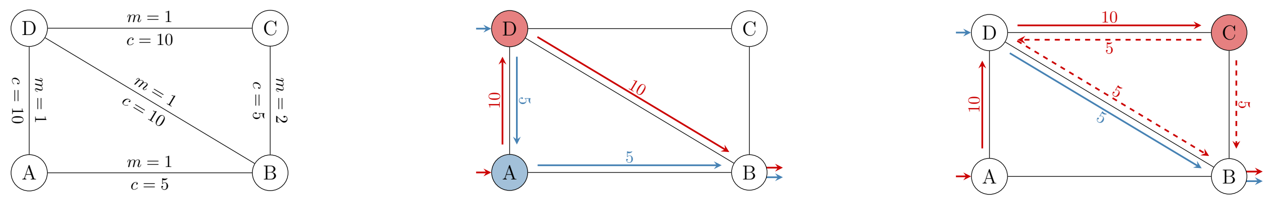
V. Data and Preliminary Examinations

While the fact that WLs can, at least in theory, be beneficial for TE is interesting in itself, it could be argued that the previous examples are carefully handcrafted and would never occur in practice. Therefore, the remainder of this paper aims at answering the question whether (or to what extent) the theoretical benefits of WLs can also be observed in practice. To evaluate this, we carry out a series of experiments on multiple



(a) Original topology. (b) Without WL: MLU = 1.0. (c) With WL: MLU = 0.5.

Figure 2: Example in which the optimal MLU cannot be achieved without a WL. A flow of size 10 needs to be routed from node A to node B.



(a) Original topology. (b) Without WL: Two policies. (c) With WL: One policy.

Figure 3: Example in which the number of required SR policies can be reduced when using WLs. A flow of size 10 needs to be routed from A to B and a flow of size 5 from D to B.

Table I: Graph properties of the topologies in the two datasets used for evaluation.

	ISP Backbone (19 Topologies)				Repetita (62 Topologies)			
	min	max	avg	stdDev	min	max	avg	stdDev
Nodes	108	186	143.11	29.90	50	197	74.31	31.60
Edges	660	1064	897.16	136.25	124	486	183.9	77.07
Density [%]	3.09	6.57	4.73	1.35	1.26	6.46	3.83	1.16
Diameter	6	8	7.32	0.58	4	35	12.57	8.196
2SR Paths with WLs [%]	65.43	79.75	72.85	5.11	62.45	97.67	83.01	8.11
2SR Paths with Edge-WLs [%]	0	0.02	0	0	0	0	0	0

real-world topologies. This section introduces the used data and also presents some preliminary examinations.

Computations are done on a Dell PowerEdge R620 with two AMD EPYC 7452 CPUs and 512GB of RAM running a 64-bit Ubuntu 20.04.1. LPs are solved using CPLEX [13].

A. Data

For our evaluations we use two sets of data. The first is provided by a globally operating Tier-1 ISP and contains real topology and traffic data collected in its backbone network between March 2017 to January 2021. For each quarter hour, a measured traffic matrix and the respective network topology snapshot is provided. Metrics are set according to a preceding metric optimization by the operator. We selected 19 sample points that are distributed about evenly across the above timeperiod and are located in the respective daily peak-hour because it is of peculiar interest for TE.

In addition to that, we use a subset of the network data featured in the publicly available *Repetita* [9] dataset. Repetita is a framework specifically created to enhance the reproducability of TE experiments. It contains network topologies, mainly taken from the *Internet Topology Zoo* [14], and five artificially generated traffic matrices for each network. It has to be noted that while the topological structure is taken from real-world networks, two artificial sets of metrics are used (unary and inverse capacity). For now, we limit our experiments to the unary metrics but we plan to carry out a more extensive analysis on multiple different metric sets in the future. For a more detailed insight into the Repetita framework and its data collection procedure see [9].

The Repetita dataset contains a large portion of very small topologies, mostly from fairly old networks like the Arpanet, which are of lesser interest for modern TE. For this reason and to keep the size of our evaluation basis manageable, we limit the following evaluations to networks from the Topology Zoo subset that have at least 50 nodes. Out of those, we further removed three topologies (*Telcove*, *Cudi* and *Pern*) because SPR already solves them optimally for all five traffic matrices. Hence, they are of no interest for TE. This leaves us with a total of 62 topologies with five traffic matrices each. For a small number of topologies some of these matrices are already solved optimally by SPR. In total this applies to 14 of the 310 instances. Those will be omitted in the following evaluations.

Table I provides an overview on the most important graph properties of the two datasets. When counting edges, parallel links are not taken into account and the density characterizes

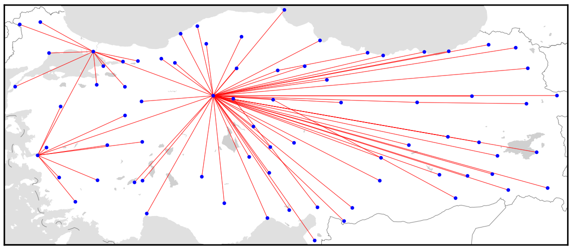


Figure 4: Ulaknet topology from the Repetita [9] dataset.

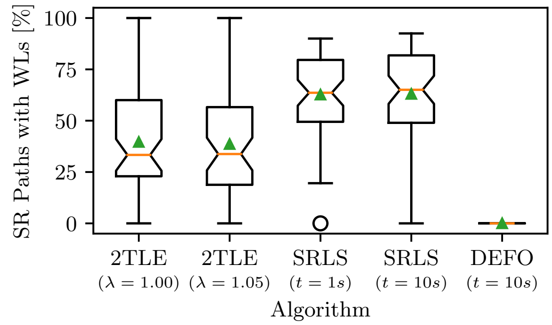


Figure 5: Percentage share of WL paths.

the ratio of numbers of (non-parallel) edges in the graph relative to a complete graph with the same number of nodes.

B. *Proportion of 2SR Paths with Weak-Loops*

In a first step, we examine how common (or uncommon) SR path with WLs really are. For this, we compute all configurable 2SR paths, excluding those that start and end at the same node, and check how many of them contain a WL. Since these results can be classified as some form of topological properties of the respective networks, they are also listed in Table I. It can be seen that the percentage share of WL paths is fairly large. For all of our evaluation instances (ISP backbone and Repetita) this number never drops below 60%, with an average of around 72% and 83%, respectively. For one topology (*Ulaknet*) it even increases to more than 97%. This latter, extraordinary high value results from the special topological structure of this network. It is basically composed out of three star-shaped networks whose centers are interconnected with each other (see Figure 4). As a result, nearly every 2SR path that uses one of the stub-nodes as intermediate segment has to pass over one of the center nodes to reach this segment and to leave it again. This, however, imposes a WL since the center node is visited multiple times.

A similar analysis was carried out for edge-WLs. These are significantly rarer than node-WLs. In the ISP backbone topologies, they only occur in significantly less than 0.1% of the total paths and not at all in the Repetita instances.

C. *Weak-Loops in current SR TE Algorithms*

Based on the previous observation that around 70–80% of all configurable 2SR paths do contain a WL, the question arises how frequently WLs are incorporated into the solutions of state-of-the-art SR TE algorithms. To answer this question, we run the 2TLE, SRLS and DEFO algorithms on our evaluation datasets and compute how many of the SR paths configured by the respective algorithm contain a WL. Since DEFO and SRLS are non-deterministic, each optimization is run five times and the median value is used. Furthermore, we run 2TLE with a time-limit of three hours.

Figure 5 illustrates the results for the Repetita instances (limited to the first of the five demand sets). The upper and lower bounds of the boxes denote the first and third quartile, respectively, meaning that 25% of all datapoints are located below, 25% above, and the remaining 50% inside the box. It can be seen that WLs are used quite frequently by 2TLE and even more by SRLS. Averaged across all instances, the percentage share of installed SR policies that contain a WL is around 40% for 2TLE and 60% for SRLS. Overall, the two algorithms use at least one WL in nearly every instance. Only less than 6% are solved without one. Contrary, DEFO does not use WLs at all. This is neither instance related nor a coincidence, but a result of its implementation. It explicitly prohibits the configuration of SR paths that contain cycles [11]. The results are fairly similar for the ISP backbone data, but are omitted for reasons of space.

It has to be remembered that the above results do not mean that WLs are strictly required to achieve solutions of this quality. There might be other, equally good (or even better) solutions that do not require any WLs. However, as long as the above algorithms are used as is, 2TLE and SRLS use WLs in more than 94% of all instances. This shows that even though WLs might, at first, seem like a rather theoretical observation with low practical relevance, the opposite is the case.

All in all, the preceding examination shows that WLs are an interesting and relevant topic, regardless of the outcome of the following experiments on their influence on solution quality in real networks. If it turns out that solution quality is not impacted by a prohibition of WLs, approaches like 2TLE and SRLS should be revised to not incorporate them in their solutions, as they would only impose additional utilization on some links. However, if our theoretical observations that WLs can be necessary to obtain an optimal solution are confirmed, this means that algorithms that explicitly prohibit them, will never be able to find the absolute best solution for those scenarios. Either way, some of the state-of-the-art SR TE algorithms will probably need to be revised and adapted.

VI. Evaluation

In this section, we evaluate the impact of WLs on the objectives of MLU minimization and policy number reduction in real-world networks. Furthermore, we also take a look at potential latency increases resulting from the use of WLs.

A. *Influence on the Maximum Link Utilization*

In Section IV, we illustrated that in some scenarios an optimal MLU can only be achieved if WLs are used. To evaluate whether these theoretic considerations carry over into

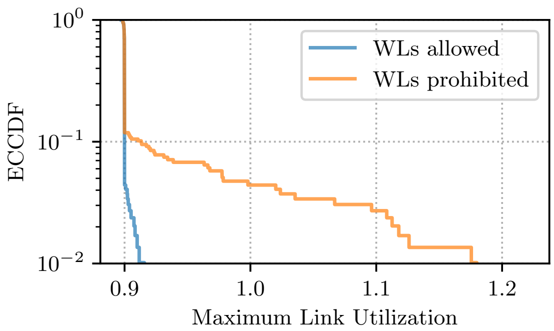


Figure 6: ECCDF of the 2SR MLUs in the Repetita instances.

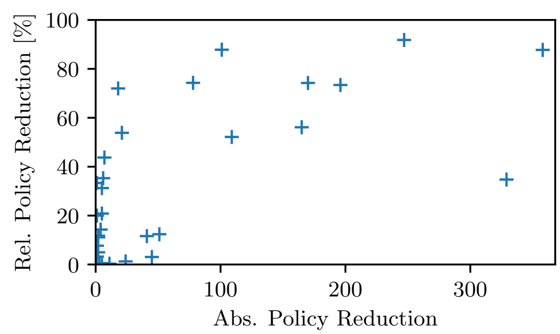


Figure 7: Reduction of the 2TLE policy number when allowing WLs in the Repetita instances.

practice, we optimize each of our evaluation instances with the 2SR algorithm (Problem 1), once with WLs and once without, and compare the achieved MLUs. The results for the Repetita dataset are depicted in Figure 6. It shows the Empirical Complementary Cumulative Distribution Function (ECCDF) of the MLU values over all instances for all five demand sets. For around 10% of all instances the MLU worsens when prohibiting the use of WLs. While some of these deteriorations are rather small, others range up to a more than 25% worse MLU. Even more important is the fact that with WLs every instance can be solved with an MLU of less than 0.93. If WLs are prohibited, however, the MLU of around 5% of all instances increases to a value higher than 1.0. Overutilization cannot be prevented anymore and a significant deterioration of network performance has to be expected.

It has to be remembered that 2SR finds the proven optimal solution for each scenario (cf. [3]). Therefore, these observations do not only show that some arbitrary algorithm performs worse when prohibiting WLs, but that solution quality will be deteriorated for every algorithm that uses two segments.

In the ISP backbone instances, the 2SR MLU does not worsen when prohibiting WLs. However, if the splitting of traffic over multiple SR policies is prohibited, results similar to the Repetita data can be observed. The only difference is the fact that while the prohibition of WLs does lead to an increased MLU, it does not result in overutilization.

B. Influence on the Number of SR Policies

To answer the question whether a reduction of the required SR policies can also be observed in real networks, we conduct a second analysis based on an adapted version of the 2TLE algorithm that uses integer variables already in the first optimization step. However, this time it is not feasible to simply run 2TLE once with WLs and once without and then compare the respective results. This is due to the previously observed fact that the MLU can worsen when prohibiting WLs. We might, for example, compare the number of policies required to reach an MLU of 0.9 to the number required to obtain an MLU of 1.1. This, however, is not a fair comparison because the latter is the easier objective and would, most likely, require less tunnels. For this reason, we first compute the optimal MLU with prohibited WLs and then compare the number of policies required to reach it once with WLs and once without.

Due to the high computation times and resource demands of 2TLE, we had to limit our evaluations of the Repetita instances to a single demand set (Set-ID 0). In addition to that, we also had to use a time-limit of five hours for each optimization. The latter results in some of the instances not being solved completely optimal. To consider this in our analysis, we compare the (potentially suboptimal) number of policies required if WLs are allowed to the CPLEX lower bound of the number of policies required if WLs are prohibited. This gives us a lower bound for the overall policy number reduction. In reality, it might be even higher. This induces some sort of inaccuracy in our evaluations, but it is basically negligible. Apart from four instances, the relative worst-case inaccuracy is always lower than 3.2% and less than 0.13% on average.

When considering policy number reduction, looking at absolute and relative values separately is not meaningful enough. For example, a relative improvement of 80% becomes less impressive if it just means that there is one policy instead of the previous five. The other way around, even an absolute reduction of 200 policies is somewhat irrelevant if there are still thousands left. To truly assess the relevance of an improvement, the absolute and relative values have to be considered in combination. This is done in Figure 7 for the Repetita dataset with a 2TLE trade-off coefficient of 0%. Each cross resembles an instance and its x- and y-values indicate the respective absolute and relative reduction. Instances for which the policy number does not improve are omitted. Overall the number of policies is reduced for around 53% of all instances when using WLs. This alone is an important finding because it confirms our theoretical considerations from Section IV for real networks. Furthermore, it also shows that this is not a rare phenomenon but occurs for every second evaluated network.

When it comes to the practical relevance of these improvements, around half of them are located around the bottom left corner. This indicates that those are of lesser relevance either due to their low relative or absolute value. The other half, however, resembles partially highly significant improvements. In one network, for example, the number of required policies can be reduced from over 400 to just 50 when allowing WLs.

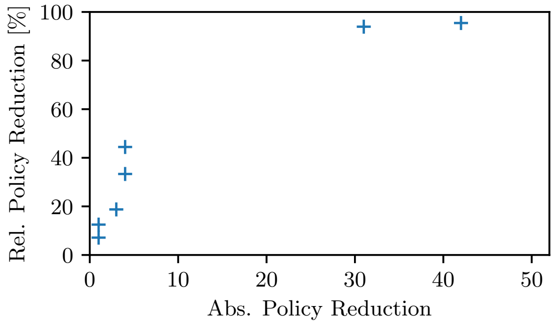


Figure 8: Reduction of the 2TLE policy number when allowing WLs in the ISP backbone instances.

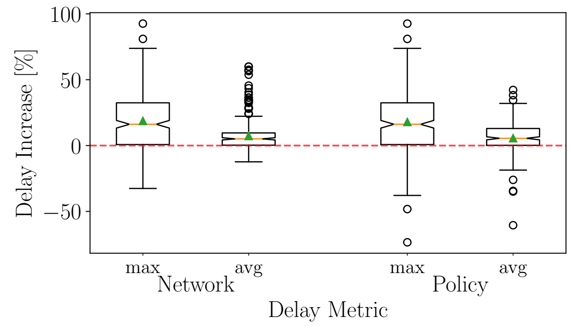


Figure 9: Percentage delay increase when allowing WLs in the 2SR optimization of the Repetita instances.

We carried out a similar analysis with a trade-off coefficient of 5% (which is closer to a practical application of 2TLE). For reasons of space, we are not able to present the detailed results, but overall they are similar to those presented in Figure 7.

Results for the ISP backbone are depicted in Figure 8. The number of instances is much smaller (cf. Section V), hence, the plot looks less crowded than the one for the Repetita dataset. It can be seen that for seven out of the 19 instances a reduction of the required number of policies can be achieved when allowing the use of WLs. Compared to the Repetita dataset, the absolute improvements are lower which is a result of generally less policies being required in the ISP backbone. Nonetheless, relative improvements of more than 90% can still be observed. In one instance, for example, the number of policies can be reduced from 44 to just two when using WLs.

To conclude, our evaluations confirm that a reduction of the required SR policies can also be observed in around half of the real-world networks. While some improvements are only of minor relevance for a practical deployment, others saved over 350 policies equaling to a reduction of more than 90%.

C. Delay

While routing traffic through WLs can be beneficial with regards to the MLU and the policy numbers, passing nodes or even edges multiple times (obviously) is not optimal in terms of delay. To evaluate the potential latency increase that results from the use of WLs, we conducted another experiment. We basically carried out the same 2SR calculations as described in Section VI-A but, this time, also calculated the average and maximum delay. This was done for all demands (the whole network) and also only for those demands that are routed via a policy. The results for the Repetita dataset are depicted in Figure 9. It shows the percentage increase of the four aforementioned delay metrics when allowing 2SR to configure WLs. The maximum delay across the whole network and the maximum policy delay are virtually identical and, hence, will not be discussed separately in the following. This also suggests that the maximum network delay often seems to be caused by an SR policy. Furthermore, it can be seen that for around 75% of all instances the delay increases when using WLs. While the maximum delay increases by roughly 20% on average, there are also instances for which it nearly doubles. The increase of the average delay, for policies as well as the whole network, is less significant. On average it lies at around 5% for both, but can also range up to more than 50% for some instances. As discussed above, this is somewhat of an expected result since routing traffic over a loop obviously introduces higher delay than using the direct, loop-free path.

However, there are also instances for which the delay can be significantly reduced when using WLs. We do not have an exhaustive explanation for this phenomenon but we suspect that it is caused by some forwarding paths only being available when using a WL. Maybe all other available paths are using some high-latency links that can only be circumvented with a WL. If this weak-loop just passes over one or two high-speed links the additional delay introduced by it might be virtually negligible. Hence, a path with such a WL might offer lower delay than the other paths without one.

The ISP backbone data does not contain delay information. Hence, we are not able to carry out a similar evaluation for it.

D. Edge-Weak-Loops

We originally planned to carry out a dedicated evaluation to assess the impact of edge-WLs. However, none of the optimal solutions for any of our experiments used edge-WLs. Hence, prohibiting them would not result in a deterioration of solution quality. Therefore, it can be concluded that they are not only significantly rarer than node WLs (cf. Table I) but also have no impact on solution quality in the evaluated networks.

VII. Discussion

The previously discussed examples and evaluations illustrate that WLs can yield benefits with regards to certain objectives. However, this often comes at the expense of increased latency. If only portions of a demand are routed over a WL, then only those are affected by these latency changes. As a result, the risk of packet reordering can increase when a round-robin-like ECMP approach is deployed. If the load-balancing is session-based, however, reordering should be no issue. Whether the benefits of using WLs are acceptable and outweigh the disadvantages depends on the respective use case and has to be ultimately decided by the network operators.

Furthermore, all of our experiments on real networks are limited to 2SR since different publications (e.g., [3] or [16] have shown that two segments are often sufficient to produce near-optimal results. In addition to that, the theoretical examples presented in Section IV show that there are scenarios in which a higher number of segments cannot resolve the necessity to configure WLs to reach an optimal solution. However, it should be examined whether our results on the impact of WLs in real networks changes when three or more segments are used. We plan to do so in the future.

Similar holds for the use of adjacency segments. Those can be used to force traffic over a link independent of its metric value. In some scenarios, this can resolve the necessity to configure WLs to obtain the optimal solution. However, this often requires a higher number of policies and also induces other potential disadvantages already discussed in Section II. Furthermore, many SR TE approaches do not use adjacency segments at all. Nonetheless, a detailed analysis on the impact of WLs when also allowing adjacency segments is an interesting research question on its own but out of the scope of this paper. We leave that for future work.

Another important aspect when discussing WLs is a potential metric dependency. WLs and SR paths in general are constructed from the concatenation of shortest paths which depend on the used metrics. While the general ability to configure WLs in a network is metric-independent, this does not hold for their benefits with respect to certain objectives. In Section IV, we have shown that ECMP is necessary to construct beneficial WLs. As a result, there will be no benefits if metrics are chosen in a way that there are no ECMP paths. For the evaluations in this paper, we used either optimized preset metrics or a uniform metric. First experiments indicate that results do not change substantially when switching to an inverse capacity metric. This coincides with findings that SR in general is, to some extent, independent of the used metrics [15]. Nevertheless, a more extensive study is required to examine the influence of different metric configurations on the benefits of WLs. We leave this for future work.

VIII. Conclusion

SR can be used to define forwarding paths that visit nodes and even edges multiple times. In this paper, we made three important observations regarding these so called WLs. (1) In real-world networks, an average of 70–80% of all configurable 2SR paths contain WLs. (2) When not explicitly prohibited, common SR TE algorithms tend to use WLs rather extensively. (3) The most important finding, however, is the fact that, against intuition, WLs can inherit benefits for certain TE objectives if there are multiple shortest paths between nodes and ECMP is used to load-balance between them. We did not only show this based on theoretical examples but also confirmed it with an extensive analysis on network data from a Tier-1 ISP and the Repetita dataset [9]. For around 10% of the examined instances the MLU worsens when prohibiting WLs. For some of those, it was not possible to prevent overutilization without the use of WLs. Furthermore, in some scenarios WLs can reduce the number of SR policies required to implement an optimal solution by as much as 90%. This shows that the possibility to configure WLs should not be neglected during the conception of TE algorithms. Algorithms that prohibit them will perform significantly worse in those scenarios.

However, we suspect that the larger portion of the configurable WLs yield no benefits and could be substituted with an equally good path without a WL. We plan to examine this and whether it is possible to distinguish between beneficial and non-beneficial WLs. With this information, algorithms can be developed that rule out "useless" WLs prior to optimization and, thus, reduce solution space and computation time. Furthermore, we plan to investigate whether the use of additional node or adjacency segments impacts the benefits of WLs in real networks.

References

[1] F. Aubry, "Models and Algorithms for Network Optimization with Segment Routing," Ph.D. dissertation, Catholic University of Louvain, Belgium, 2020, http://hdl.handle.net/2078.1/237671.

[2] F. Aubry, D. Lebrun, S. Vissicchio, M. T. Khong, Y. Deville, and O. Bonaventure, "SCMon: Leveraging Segment Routing to Improve Network Monitoring," in *Proc. of the IEEE Int. Conf. on Computer Communications (INFOCOM)*, 2016.

[3] R. Bhatia, F. Hao, M. Kodialam, and T. V. Lakshman, "Optimized Network Traffic Engineering using Segment Routing," in *Proc. of the IEEE Int. Conf. on Computer Communications (INFOCOM)*, 2015, pp. 657–665.

[4] Cisco Systems Inc., "Cisco Annual Internet Report (2018–2023)," https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2020.

[5] C. Filsfils, K. Michielsen, and K. Talaulikar, *Segment Routing Part I.* CreateSpace Independent Publishing Platform, 2017.

[6] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The Segment Routing Architecture," in *Proc. of the IEEE Global Communications Conf. (GLOBECOM)*, 2015.

[7] P. Francois and O. Bonaventure, "Avoiding Transient Loops during the Convergence of Link-State Routing Protocols," *IEEE/ACM Transactions on Networking*, vol. 15, p. 1280–1292, 2007.

[8] S. Gay, R. Hartert, and S. Vissicchio, "Expect the Unexpected: Sub-Second Optimization for Segment Routing," in *Proc. of the IEEE Int. Conf. on Computer Communications (INFOCOM)*, 2017.

[9] S. Gay, P. Schaus, and S. Vissicchio, "REPETITA: Repeatable Experiments for Performance Evaluation of Traffic-Engineering Algorithms," *ArXiv e-prints*, 2017.

[10] R. Hartert, P. Schaus, S. Vissicchio, and O. Bonaventure, "Solving Segment Routing Problems with Hybrid Constraint Programming Techniques," in *Proc. of the Int. Conf. on Principles and Practice of Constraint Programming (CP)*, 2015, pp. 592–608.

[11] R. Hartert, S. Vissicchio, P. Schaus, O. Bonaventure, C. Filsfils, T. Telkamp, and P. Francois, "A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks," in *Proc. of the ACM Conf. on Special Interest Group on Data Communication (SIGCOMM)*, 2015, pp. 15–28.

[12] U. Hengartner, S. Moon, R. Mortier, and C. Diot, "Detection and Analysis of Routing Loops in Packet Traces," in *Proc. of the ACM SIGCOMM Workshop on Internet Measurment*, 2002, p. 107–112.

[13] IBM, "IBM ILOG CPLEX Optimization Studio V12.6.2," http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.2, 2014.

[14] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[15] T. Schüller, N. Aschenbruck, M. Chimani, and M. Horneffer, "On the Practical Irrelevance of Metrics on Segment Routing Traffic Engineering Optimization," in *Proc. of the IEEE Conf. on Local Computer Networks (LCN)*, 2018, pp. 640–647.

[16] T. Schüller, N. Aschenbruck, M. Chimani, M. Horneffer, and S. Schnitter, "Traffic Engineering using Segment Routing and Considering Requirements of a Carrier IP Network," *IEEE/ACM Transactions on Networking*, vol. 26, pp. 1851–1864, 2018.