

On the Practical Irrelevance of Metrics on Segment Routing Traffic Engineering Optimization

Timmy Schüller[°], Nils Aschenbruck[°], Markus Chimani[°], Martin Horneffer^{*}

[°]University of Osnabrück, Institute of Computer Science
Wachsbleiche 27, 49090 Osnabrück, Germany
Email: {tschueller, aschenbruck, markus.chimani}@uos.de

^{*}Detecon International GmbH
Sternengasse 14-16, 50676 Köln, Germany
Email: timmy.schueller@detecon.com

^{*}Deutsche Telekom Technik GmbH
Dahlweg 100, 48153 Münster, Germany
Email: martin.horneffer@telekom.de

Abstract—Segment Routing (SR) based traffic engineering can be used to avoid network congestion effectively. Most research about SR considers the routing metric as a given constant. The metric defines shortest paths. As shortest paths are essential building blocks of SR paths, the metrics heavily influence SR. In this paper, we evaluate this influence thoroughly using data from a tier one Internet Service Provider (ISP) backbone, as well as other, publicly available topologies. We conclude that metrics do in fact play a role in SR optimization. Simple metrics, however, already push SR close to the optimum with respect to the most common traffic engineering objective of minimizing maximum utilization.

I. INTRODUCTION

It is no secret that the number of devices with a network connection and, thus, the amount of data traversing the Internet is increasing rapidly [3]. We currently are beginning to experience an age where everything becomes connected. We are able to control heating, lights, and other things in our home with mobile devices or watch the latest episode of our favorite online show on the train. Managing this increasingly heavy burden on the underlying network structure is a challenging task for Internet Service Providers (ISPs). Expanding networks physically and installing new hardware in the real world is an important, yet expensive, process to facilitate the growth of the Internet. Another strategy to absorb high traffic loads is to improve the intelligence of the network itself. Through means of traffic engineering, flows are distributed onto resource efficient paths to satisfy predefined objectives. This must not be seen as an alternative to physical expansion, but as an additional option.

One relatively conservative traffic engineering approach is routing metric optimization. The most widely used Interior Gateway Protocols (IGPs) are based on shortest paths. Examples are the Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS). A shortest path between two nodes is characterized as being the path with a minimal sum of edge weights that make up the path. This means that we can control which paths become the shortest ones by selecting different metrics [1]. We can also hand-tune such a metric, which decouples it from its original semantic meaning and turns it into more of a generic parameter. While

this is simple in deployment and management, finding an optimal set of metrics is in fact a difficult task that is often made even harder by corporate agreements and policies. Therefore, in practice, metric optimization is performed heuristically or completely manually.

Another more sophisticated traffic engineering approach is Segment Routing (SR) [6]. It became popular in the research community in the last couple years, while it is currently being standardized by the Source Packet Routing in Networking (SPRING) working group of the Internet Engineering Task Force (IETF). As the group name suggests, it is based on the idea to define a packets' path from its origin or entry point in the domain that is to be optimized. An SR path is defined by a stack of segments that can be understood as check points. Each of the segments is reached by consulting the IGP, which again, is most likely a shortest paths based routine. Thus, an abstract way to imagine an SR path would be to think of it as a concatenation of shortest paths.

To date, to the best of our knowledge, all research on traffic engineering using SR took routing metrics as unmodifiable constants, even though shortest paths play an important role in SR. In this paper, we show that the choice of routing metrics can impact the quality of SR based traffic engineering in certain cases. Yet, while there are metric configurations that make SR less successful, very good results can already be achieved with simple metrics and—in contrast to standard shortest path routing—optimization is often unnecessary. We use real data from a tier one ISP, as well as topologies from the Internet Topology Zoo [12] to come to these conclusions.

The rest of this paper is structured as follows. Section II introduces the required background knowledge, covering a few possible routing metrics along with how SR works. Then, in Section III, we look at related work and point out potential problems. In Section IV our data basis and evaluation setup is described, before Section V discusses the results in detail. Finally, Section VI concludes the paper and describes potential future work.

II. BACKGROUND

Before discussing how different sets of metrics can impact SR, a few basics have to be explained. For this, three concepts will be presented in this section. First, the notion of shortest paths has to be defined. Second, we list a few examples of routing metric configurations. Third, we give a brief introduction to SR.

A. Shortest Path Routing

We define a network as a graph $G = (V, E)$ with a set of vertices $V = \{v_1, v_2, \dots, v_n\}$ and a set of connections between vertices, called edges, $E = \{e_{ij}\}$. A single edge e_{ij} is directed from vertex v_i to vertex v_j . It is labeled with a capacity $c(e_{ij})$ that describes how much traffic it can transport and a metric $m(e_{ij})$ controls the “popularity” of the edge. Both are non-negative integer values. A path p_{ij} between two vertices in the graph can be understood as an ordered list of edges that, when followed in order, form a connection between the two.

Additionally to our graph, we also have a set of traffic demands $T = \{t_{ij}\}$. Each demand has a source v_i , a destination v_j , and a positive numerical value that describes the required bandwidth. With this, we can formulate a general traffic engineering problem:

Definition 1: Given a graph G and a set of demands T , find a mapping of demands to corresponding paths, such that all demands are routed and no edge exceeds its capacity.

We can compute the distance $d(p_{ij})$ or length of a path by summarizing the metrics of each of the edges it consists of:

$$d(p_{ij}) = \sum_{e \in p_{ij}} m(e)$$

A shortest path $p_{ij}^{shortest}$ can now be identified as a path in the set of all possible paths P_{ij} from vertex v_i to v_j with minimal distance:

$$p_{ij}^{shortest} \in \left\{ p_{ij} \mid d(p_{ij}) = \min_{p'_{ij} \in P_{ij}} (d(p'_{ij})) \right\}$$

In shortest path routing, each demand is routed along the corresponding shortest path. While this is a simple solution, it is not optimal in terms of avoiding congestion and relies heavily on the configured metrics. In fact, the solution may be infeasible, as edge capacities may get exceeded.

B. Static Routing Metric Configurations

At some point in network design, the question of edge metric configuration comes up. There are several different approaches to define good metrics that rely on specific use cases [1]. We discuss three possibilities here:

- 1) **Uniform metrics:** All edges are treated equally and have the same metric value, for example 1. As a result, the distance of a path is essentially the number of nodes that are visited on its way, also known as hop count. This configuration is typically not used in practice, since in

a real network, links have significant differences, e.g., in geographical length or capacity. These factors are not accounted for by a uniform metric, while they do have a significant impact in practice.

- 2) **Geographic distance:** This directly addresses one of the weaknesses of the previous approach. Delay is primarily based on the length of a connection. Thus, when configuring metrics, we can use the physical length of an edge to account for delay. This is also one of the most intuitive metrics, because the length of a path is now equal to its actual physical length. This approach may not be applicable for all networks, however, as they have to be large enough for edges to have a significantly different delay.
- 3) **Inverse capacity:** While the previous approach seems intuitive, the capacity of an edge is often considered more important than its length. Thus, another possible configuration sets the metric of an edge proportionally to the inverse of its capacity. This favors edges that can transport much traffic over smaller edges with low capacity.

C. Segment Routing

SR is a source routing architecture that can be used in several use cases [6]. The main idea is to split up the path of a packet into multiple parts or segments. A segment can be defined by a node, adjacency, or network service. Segments are identified by segment IDs (SID). If we look at a simple network, for example, we can assign unique node SIDs to each router. We are then able to control the path of an incoming packet at its source by adding a stack of SIDs to it. This can be done using Multiprotocol Label Switching (MPLS) labels or a dedicated IPv6 extension. The SIDs are processed from top to bottom until the destination is reached. One of the big advantages of SR over similar technologies, such as MPLS tunnels, is the fact that not every hop has to be specified. Rather, the next SID on the stack can be seen as the next checkpoint that the packet has to pass through. The path it uses to satisfy this temporary goal is managed by the IGP, which, in most use cases, operates through shortest paths.

For our purposes, we are going to use this technology to perform traffic engineering. The minimization of the maximum link utilization is most commonly viewed as the main objective, because it minimizes congestion. In [2], a linear programming formulation was proposed. The authors show that 2SR, a variant where at most two shortest paths may be concatenated, is sufficient to gain very good results. We confirmed this finding in [14], where we used a real ISP network. The latter publication also adds two requirements to the formulation that an ISP might need. This is done by adding a mixed integer program that uses the result of the basic formulation in a second step. This extension is called the Tunnel Limit Extension (TLE). In the context of 2SR it is also called 2TLE. It minimizes the number of SR paths to simplify deployment and maintenance. Additionally, arbitrarily splitting traffic demands is prohibited, as this is not supported

min θ

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

$$\sum_{ij} \sum_k g_{ij}^k(e) x_{ij}^k \leq \theta c(e) \quad \forall e \quad (2)$$

$$x_{ij}^k \geq 0 \quad \forall ij \quad (3)$$

Problem 1: Basic 2SR linear programming formulation following [2].

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

$$\text{s.t.} \quad \sum_k u_{ij}^k = 1 \quad \forall ij \quad (4)$$

$$\sum_{ij} \sum_k g_{ij}^k(e) u_{ij}^k t_{ij} \leq \theta' c(e) \quad \forall e \quad (5)$$

$$\theta' \leq \lambda\theta \quad (6)$$

$$u_{ij}^k \in \{0, 1\} \quad \forall ijk \quad (7)$$

Problem 2: Tunnel Limit Extension (TLE) as second optimization step to Problem 1 that minimizes the number of SR tunnels and restricts the number of tunnels per demand to 1.

by current router hardware. 2SR is also implemented in [8]. The authors use an alternative formulation that they claim to be more efficient. They also avoid arbitrary splitting of traffic demands, which is similar to TLE.

As we are going to use the basic 2SR optimization throughout the evaluation in Sections V-B and V-C, the problem formulation is given in Problem 1. The objective is to minimize θ , which stands for the maximum link utilization. The variables in this program, x_{ij}^k , define the absolute amount of the traffic from node i to node j that is routed through intermediate node k . As defined in Section II-A, t_{ij} are the demands given by a specific traffic matrix. They define the amount of traffic that originates at node i and is destined for node j . Equation (1) ensures that every traffic demand t_{ij} is satisfied by the sum of the traffic on all corresponding paths. Inequality (2) ensures that all sub-demands are satisfied while limiting each edge's total throughput to its capacity $c(e)$ scaled by θ . For each link, $g_{ij}^k(e)$ gives information about the share of a flow from i to j through k that is routed over edge e . These values include shortest path information, as the paths from i to k and from k to j are essentially two linked shortest paths. We can think of the values of g as Equal-Cost Multi-Path (ECMP) split-values. In a situation where there are two equally shortest paths, for example, the split-value at the node where the paths diverge would be 50% for each of the two. These values can be precomputed for all node pairs, intermediate nodes, and links and are, therefore, constant input parameters for the linear program.

With the minimization of the overall number of segments used, 2TLE introduces a second objective. It still respects the objective from basic 2SR, by using it as a constant. Thus, 2TLE can be seen as a second step in a chain of two mathematical programs. In Section V-D, we will take a short look at the impact of metrics on this multiobjective optimization. The 2TLE formulation can be found in Problem 2. It is similar to the basic problem. In order to be able to count segments and ultimately minimize them, the absolute traffic variables x_{ij}^k are replaced with relative traffic variables u_{ij}^k . As a result, constraints 4 and 5 have almost the same syntactic formulation and exactly the same semantic meaning as equations 1 and 2 in Problem 1. Yet, the objective function works a little different. The solution of the first step is added via inequality 6 and relaxed by the tradeoff coefficient λ , which effectively weights both objectives against each other. By forcing the relative traffic variables u_{ij}^k to be binary, splitting of traffic flows due to multiple different intermediate segments is prevented. Yet, minimizing the total number of SR paths would be of little use, because tunnels that follow the shortest path do not have to be deployed in practice. Shortest path tunnels should, therefore, not be counted towards the objective. The formulation implements this by excluding all variables from the sum in the objective function where the indices k and j are the same. Thus, only actual 2-SR paths are counted. The maximum link utilization θ' is added to the objective function to rate solutions with an equal number of tunnels. The constant coefficient ensures that the impact of θ' will always be lower than the impact of a single binary variable.

There also exist much faster implementations of SR that gain their efficiency by sacrificing optimality. For example, in [10] the authors follow a constraint programming approach and in [9] SR is performed using a local search algorithm.

III. RELATED WORK

In traffic engineering with metric optimization, the metric itself is mostly decoupled from its original semantic meaning. Instead of representing a physical trait, for example bandwidth or delay, it gains a more abstract, general meaning in order to fulfill a traffic engineering objective. The problem of finding a routing metric configuration that optimally routes a specific set of demands is NP-hard [7]. Thus, Fortz and Thorup propose a local search heuristic, which finds practically good solutions. This is also implemented in [8]. In real ISP networks, however, there exist use-case specific additional requirements and policies that need to be addressed. Often, they are enforced through fine tuning metrics by hand. One could, for example, apply high metric values to international or intercontinental links. This would prevent traffic originating and arriving within the same country or continent from being routed on a detour through another country or continent.

There are also routing algorithms that allow more than one metric simultaneously. For example, the Enhanced Interior Gateway Routing Protocol (EIGRP) [4], developed by Cisco, supports up to four metrics. Cisco also published a commercial planning software that is capable of performing global or

incremental routing metric optimization. Details about the algorithms behind Cisco WAN Automation Engine (WAE) (formerly Cisco MATE), however, are not publicly available.

A genetic algorithm tailored for OSPF is proposed in [5]. This strategy is further adapted in [13], where the authors use a local search heuristic to find a good starting population for the genetic algorithm. They test this hybrid genetic strategy for single metric settings, as well as multi metric scenarios. It is shown that two optimized metrics in the context of EIGRP achieve better results than a single optimized metric with OSPF. As these two works were done before the idea of SR was developed, they only focus on destination based routing, while acknowledging the possibilities of source based routing.

To the best of our knowledge, there is no research on the impact of different routing metric configurations on SR to date, even though SR relies on shortest paths.

IV. APPROACH & SETUP

The remainder of this paper aims to answer the question about the impact of metrics in SR based traffic engineering. In theory, metrics do not only have the ability to change shortest paths, but hence also SR paths. This fundamentally questions SR optimizations that take metrics for granted. The optimization could perform better or worse, depending on the underlying metrics. We test this by performing a series of experiments that compare different optimization strategies and metric configurations on multiple real-world topologies. This section introduces the algorithms and the data that these experiments are based on.

A. Algorithms

For our calculations, we focus on four algorithms. First, we use Shortest Path Routing (SPR) to get an approximate value for a state of the art routing strategy. While this is the state we aim to improve on, it should not be mistaken for an upper bound. Second, we use a Multicommodity Flow (MCF) formulation in the form of a linear program. MCF allows completely arbitrary paths for each individual traffic demand and, thus, can be described as optimal routing for a known set of demands. Due to its high degree of freedom, it is not feasible in practice but offers a good lower bound. It should be noted that, when optimizing congestion, MCF is independent on the chosen metrics. Third, we use the basic 2SR linear programming formulation presented in [2]. 2SR was shown to get very close to the optimal case, while maintaining a realistic level of complexity [2], [14]. In the evaluation, the closer a 2SR result is to the corresponding maximum link utilization found by MCF, the better. If the two results match, we consider the 2SR result as being optimal. Fourth, we use 2TLE [14], an extension to basic 2SR. The solutions returned by this mixed integer program are even more realistic for real-world deployments. The formulation includes additional requirements that have to be satisfied. Due to the secondary objective, 2TLE rarely finds results that are optimal in terms of the primary objective. Rather, it will find values that are slightly higher.

Table I: Overview over the selected Internet Topology Zoo networks.

Name	Nodes	Edges	Accumulated edges	Figure Nr.
Cogentco	197	490	486	8
UsCarrier	158	378	378	9
GtsCe	149	386	386	4
Colt	153	382	354	—
TataNld	145	388	372	—
Deltacom	113	166	322	—
Ntt	32	432	126	—
DialtelecomCz	138	302	302	—
Interoute	110	316	294	—
Ion	125	300	292	10

The computations are done on a Dell PowerEdge R620 server with an Intel Xeon E5-2620 CPU and 8 * 32GB DDR3 synchronous 1600 MHz RAM running a 64-bit Ubuntu 12.04.5. Linear programs were solved using CPLEX [11].

B. Data

We use two data sets for the evaluation. The first data set comes from a tier one ISP. It includes topology information and real, measured traffic matrices each quarter hour from January 2017 to September 2017. 18 peak hour matrices were selected evenly across this time period. Thus, additionally to having different traffic demands, the samples feature slight differences in topology. On average, the network spans across 109 nodes with 2085 total edges. Not taking parallel links into account, this number reduces to 663 accumulated edges and leads to a density of about 6% relative to a complete graph. The data also contains preset metrics that were optimized by the network operators. In the remainder of this paper, this topology will be referred to as the main topology.

The second data set is a subset of the data provided in a framework called REPETITA [8]. This framework aims to enable reproducible and comparable traffic engineering experiments. As such, it comes with some sample topologies, most of which are graphs from the Internet Topology Zoo [12]. We used the 10 biggest topologies¹ from the Internet Topology Zoo subset based on the sum of total number of nodes and edges. This selection is listed and characterized in Table I. REPETITA also offers five synthetic traffic matrices for each topology. Even though the demands are only synthetic, this data set gives us the opportunity to experiment with topologies that are very different from another and from our main data set. To give a feeling for how different the topologies are and how they look like, a few selected topologies from this second data set are brought into geographical and visual context in the appendix.

V. EVALUATION RESULTS

We evaluate the impact of setting different metric configurations. We start by looking at the impact on simple shortest path routing using our main topology. In this scenario, we test uniform metrics, inverse capacity metrics, and preset,

¹The KDL Topology is not featured in REPETITA and also omitted here, because it contains too many nodes.

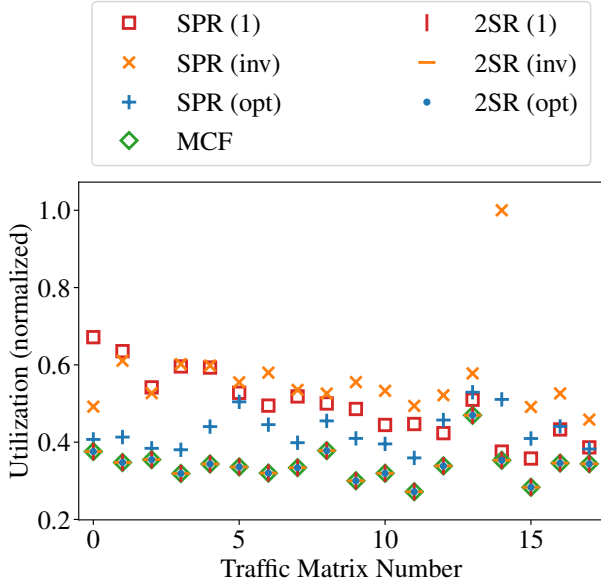


Figure 1: Influence of uniform (1), inverse capacity (inv), and optimized (opt) metric configuration on SPR and 2SR measured on 18 different peak matrices of a tier one ISP.

optimized metrics. The same configurations and topology will then be used to test the influence of metrics on basic 2SR. To support our findings, additional results with uniform and delay based metrics from 10 Internet Topology Zoo traces will be presented. We also conduct some tests with completely random metrics. Lastly, we look at results from the 2TLE optimization. With the minimization of the number of required 2SR tunnels it incorporates a secondary objective that may react differently on distinct metric configurations.

A. Influence on Shortest Path Routing

As explained in Section II-A, shortest paths are defined by the routing metric. Thus, we expect rapidly different results when applying different metric configurations. The maximum link utilization of shortest path routing with underlying uniform, inverse capacity, and optimized metrics on the main topology is displayed in Figure 1. Additionally to the three metrics, which are visualized by the red squares, orange crosses, and blue pluses respectively, the optimum is also displayed in the form of green diamonds. It is generated by a linear programming implementation of the MCF problem, which allows completely arbitrary paths. All values are normalized with the highest shortest path routing value, which, as a result, is now exactly 1.

As expected, the three different metric configurations lead to significant variations. One can argue that the optimized metrics were chosen well, since they yield the best result for most matrices with only few exceptions. In a similar fashion, the inverse capacity metric could be labeled worst, since in most cases it scores the highest utilization between the three. All results, however, fluctuate quite a bit across the different demand matrices. Qualitative statements are thus to be treated

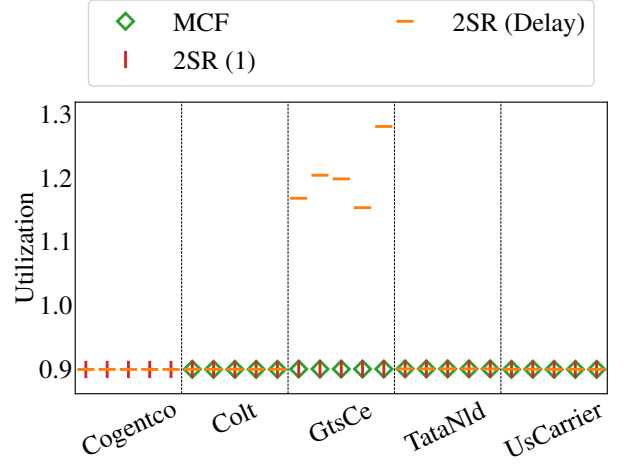


Figure 2: Influence of uniform (1) and delay based metric configuration on 2SR measured on the top 5 biggest topologies of the Internet Topology Zoo.

with caution. Another observation is the fact that the shortest path results never match the lower bound.

B. Influence on 2SR

Within the same Figure 1, the influence of the three metric configurations on 2SR becomes apparent. The red vertical lines show the utilizations of 2SR optimized traffic based on uniform metrics, the orange horizontal lines represent results based on inverse capacity metrics, and the dots stand for results based on the optimized metrics. The colors of the respective markers match the metric variation for the shortest path routing counterpart. There is no difference between the three. All of them exactly match the optimal result in all cases. Hence, the result is that already a single intermediate waypoint suffices to render the metrics essentially irrelevant.

To verify these findings, we also considered 10 topologies from the Internet Topology Zoo. Instead of small changes between the different dates in the main topology, where, for example, there is one new link or a temporarily inactive node, this time, the topologies are fundamentally different. The first half, with 5 synthetic traffic matrices for each topology, is presented in Figure 2. The second half can be found in Figure 3. Due to the lack of preset, optimized metrics, only uniform and delay based metrics could be tested. Originally we planned to also use inverse capacity metrics, but found them to be almost equal to the uniform metric configuration. Due to the public availability of the data, the maximum utilization values are not normalized for these figures.

The first topology in Figure 2, “Cogentco”, unfortunately was too big for MCF to terminate on our system before running out of memory. The overall picture, however, is consistent to our previous statement. In most cases, there are no differences between 2SR optimization results based on uniform and delay metrics. The only exception is the “GtsCe” topology. Here, the delay metric leads to significantly worse utilization values beyond 100%. Hence, in a real scenario this

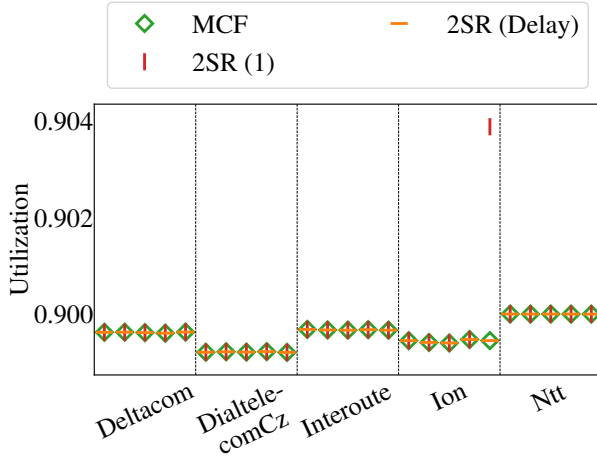


Figure 3: Influence of uniform (1) and delay based metric configuration on 2SR measured on the top 5–10 biggest topologies of the Internet Topology Zoo.

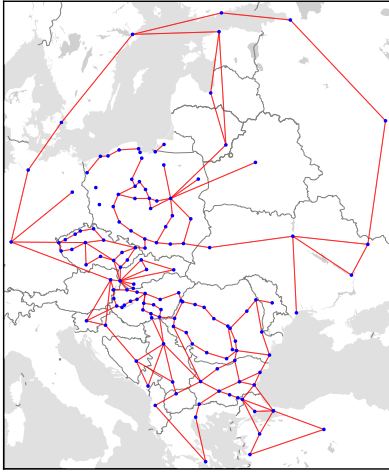


Figure 4: GtsCe Topology

configuration would lead to congestion. A possible explanation could lie in the structure of the topology itself, displayed in Figure 4. It contains long circular paths, which could be disadvantageous for the chosen strategies, leading to the observed results. Another possible reason could lie in badly measured or set delay values.

The similarity across different topologies can be explained by the synthetic traffic matrices. They were generated to mimic a congested state when applying shortest path routing that is challenging to optimize. The demands were likely constructed such that the optimal routing already results in approximately 90% maximum link utilization.

Figure 3 shows a similar picture. The y-Axis here is a bit narrower. This time, variations are only apparent in one matrix of one topology and only in the third decimal place. The exception occurs in the second to last topology “Ion”. In the

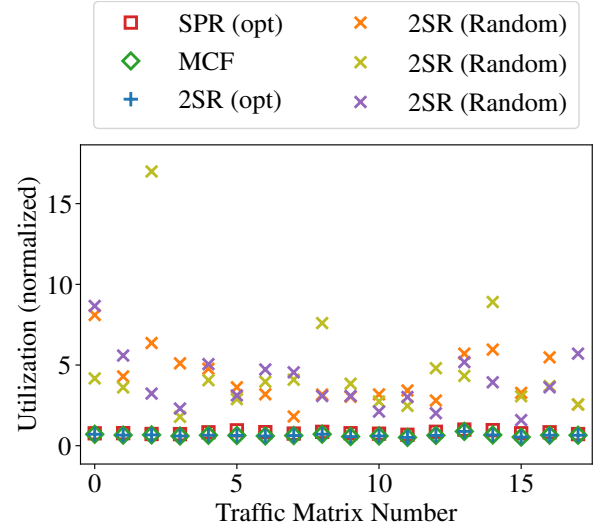


Figure 5: Influence of random metric configuration on 2SR measured on 18 different peak matrices of a tier one ISP.

last matrix 2SR based on uniform metrics performs worse. It should be noted, though, that this deviation is negligible compared to the differences of the delay metric in “GtsCe”. The latter shows different results for all five matrices, while here we only have one outlier.

To conclude, we can, again, see that 2SR seems to reach practically optimal results regardless of the underlying metric. This time, though, we see an exception in one of ten cases. The uniform metric of 1 though, proves to be surprisingly stable. Additionally, it has the unique advantage of providing a speedup in shortest path computations. It enables the use of a simple breadth-first search with a linear complexity.

C. Influence of Random Metrics

The metrics used in the previous paragraphs all follow a strategy that is more or less meaningful. To check whether the metrics have an impact on SR at all, we are now going to use completely random metrics. The results are shown in Figure 5, where we look at the main topology again. The metrics were uniformly picked from a range between 1 and 10,000. Although this seems like a wide range, it is not unrealistic. We can observe comparably inhomogeneous metric settings in the optimized metrics of our main topology that performed well.

The 2SR optimized results based on the three new sets of metrics are displayed as orange, yellow, and purple crosses. For comparison, MCF, shortest path routing, and 2SR, all based on optimized metrics, are shown as well. The maximum link utilization is, again, normalized to the highest result of shortest path routing to ensure comparability across figures. Even though the normalization is consistent, the scale goes past 1 up to 17. As such, the difference between shortest path routing and MCF appears smaller, despite being the same. This shows that the 2SR results based on random metrics are all significantly worse than everything we have seen so

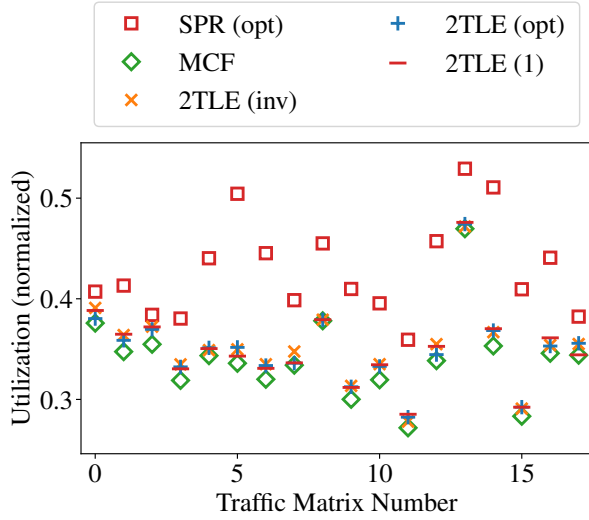


Figure 6: Maximum link utilization as optimized by 2TLE ($\lambda = 5\%$). Based on the main topology with uniform (1), inverse capacity (inv), and optimized (opt) metric configuration. Compared with lower bound MCF and shortest path routing.

far. This observation stands in sharp contrast to our previous findings, where the metrics seemed to be mostly irrelevant to the 2SR optimization. Here we can see that this is definitely not generally valid and that metrics should be selected with some thought rather than completely arbitrarily.

D. Influence on the Number of Required SR Tunnels

So far, we have seen the influence of different metrics on the maximum link utilization optimization of 2SR. 2TLE is an extension to 2SR and adds the minimization of the number of required 2SR tunnels as a secondary objective. We test both objectives as optimized by 2TLE using our main data set and metric settings as seen in the preceding evaluation.

First, Figure 6 shows a familiar picture. It visualizes the first objective, the maximum link utilization. The tradeoff parameter of 2TLE was set to 5%. This means that in order to achieve a small number of 2SR tunnels, the maximum link utilization reached by basic 2SR may be deteriorated by at most 5%. As a result, the orange crosses, blue pluses, and red lines, standing for 2TLE results based on optimized, uniform, and inverse capacity metrics, respectively, find values a little above the lower bound. Between the three, no significant difference can be observed.

The metrics do, however, have an impact on the minimized number of SR tunnels. This is visualized in Figure 7. The boxplots depict the secondary objective of 2TLE with respect to the same parametrization as before. While we arguably can not determine a statistical significance between the results based on optimized and uniform metrics, the third option stands out. With optimized metrics we only need 3 SR tunnels on average to get within 5% of the optimal maximum link utilization. With uniform metrics we need about 5 and with

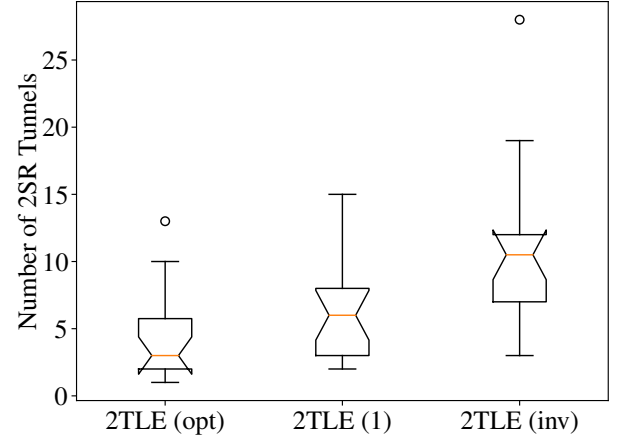


Figure 7: Number of SR tunnels needed as optimized by 2TLE ($\lambda = 5\%$). Based on the main topology with uniform (1), inverse capacity (inv), and optimized (opt) metric configuration.

inverse capacity metrics we need twice as much. Yet, these numbers are all quite low. To put them into perspective: With basic 2SR instead of 2TLE, about 3000 tunnels would be used. We also observed significantly longer computation times. This suggests the following theory: There are several solutions for minimal link utilizations. The number of possible optimal solutions, however, is larger for better metric configurations. The addition of a secondary objective now greatly reduces this number. This results in an observable differences in performance based on metric configurations for the secondary objective.

VI. CONCLUSION

In this paper, we shared three core findings. (1) The routing metric configuration influences shortest path based traffic engineering approaches such as SR. We found that randomly selected metrics can lead to a substantial performance loss of 2SR optimizations. (2) As long as the configured metric follows a reasonable pattern, it does *not* play a role in SR based traffic engineering. A simple uniform metric or a metric set proportional to the inverse capacity, for example, works as well as a complex optimized metric. A uniform metric could even increase the performance of SR optimizations, as it eases shortest path computation. There is, therefore, no need to do complicated optimizations when SR is used. In one of 11 test scenarios, however, the delay based metric produced worse link utilizations. This shows that these theories are not universal but topology dependent. (3) If we extend our scope to a secondary objective besides minimizing maximum link utilization, such as minimization of the number of 2SR tunnels, the choice of metric becomes more relevant. We found a significant difference in the secondary objective after optimization based on optimized and uniform metrics versus optimization based on inverse capacity metrics. As more requirements and

objectives are added, using some sort of metric optimization beforehand seems to become more beneficial.

We verified these conclusions using an extensive data set from a tier one ISP and ten other publicly available topologies from the Internet Topology Zoo. The former includes real traffic measurements, while the latter uses synthetic traffic generated by [8].

We suspect that the high deterioration of 2SR with random metrics is based on the variety of metrics and not the potentially high distance between possible values. In other words, it may be a bad idea to give each link a unique metric value, while it may be okay to give some links a metric of 1, others a metric of 600, and the rest a metric of 10 000. Verifying this, however, would require further evaluation.

Another potential topic of future work could be to look at scalability of 2SR. We captured the current state of real networks and optimization within them. In the future, they are likely to increase in size. This could lead to a point where 2SR is not expressive enough to result in sufficiently good optimizations. Before adding more layers of complexity by using 3SR or even n SR instead, a preceding metric optimization might be a viable alternative. It is also possible that the applicability of 2SR without optimized metrics is dependent on certain graph properties, which remain to be identified.

REFERENCES

- [1] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel, "A Survey on Routing Metrics," *TIK report*, vol. 262, pp. 1–53, 2007.
- [2] R. Bhatia, F. Hao, M. Kodialam, and T. V. Lakshman, "Optimized Network Traffic Engineering using Segment Routing," in *Proc. of the 2015 IEEE Conf. on Computer Communications (INFOCOM)*, 2015, pp. 657–665.
- [3] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2016–2021," <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>, 2017.
- [4] Cisco, "Enhanced Interior Gateway Routing Protocol," White Paper, September 2017.
- [5] M. Ericsson, M. Resende, and P. Pardalos, "A Genetic Algorithm for the Weight Setting Problem in OSPF Routing," *Journal of Combinatorial Optimization*, vol. 6, pp. 299–333, 2002.
- [6] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The Segment Routing Architecture," in *Proc. of the 2015 IEEE Global Communications Conf. (GLOBECOM)*, 2015, pp. 1–6.
- [7] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 756–767, 2002.
- [8] S. Gay, P. Schaus, and S. Vissicchio, "REPETITA: Repeatable Experiments for Performance Evaluation of Traffic-Engineering Algorithms," *ArXiv e-prints*, Oct. 2017.
- [9] S. Gay, R. Hartert, and S. Vissicchio, "Expect the Unexpected: Sub-Second Optimization for Segment Routing," 2017.
- [10] R. Hartert, P. Schaus, S. Vissicchio, and O. Bonaventure, "Solving Segment Routing Problems with Hybrid Constraint Programming Techniques," in *Proc. of the 2015 International Conf. on Principles and Practice of Constraint Programming (CP)*, 2015, pp. 592–608.
- [11] IBM, "IBM ILOG CPLEX Optimization Studio V12.6.0," http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.0, 2014.
- [12] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1765–1775, 2011.
- [13] A. Riedl, "A Hybrid Genetic Algorithm for Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics," in *IEEE Workshop on IP Operations and Management*. IEEE, 2002, pp. 166–170.
- [14] T. Schüller, N. Aschenbruck, M. Chimani, M. Horneffer, and S. Schnitter, "Traffic Engineering using Segment Routing and Considering Requirements of a Carrier IP Network," in *IFIP Networking*, 2017, pp. 1–9.

APPENDIX

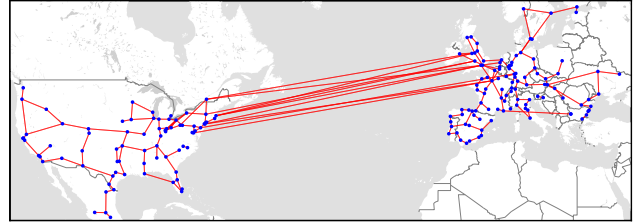


Figure 8: Cogentco Topology

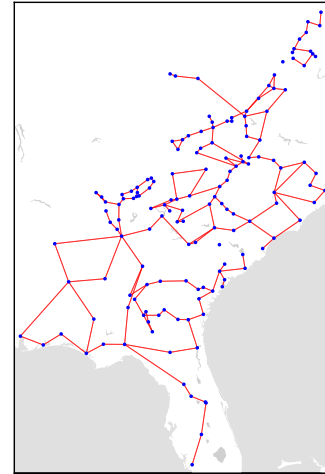


Figure 9: UsCarrier Topology

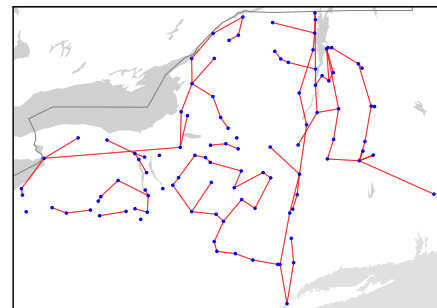


Figure 10: Ion Topology