Live Long and Prosper – On the Potential of Segment Routing Midpoint Optimization to Improve Network Robustness

Alexander Brundiers°, Timmy Schüller•, Nils Aschenbruck°

°Osnabrück University, Institute of Computer Science, Germany
Email: {brundiers, aschenbruck}@uos.de

•Deutsche Telekom Technik GmbH, Germany
Email: timmy.schueller@telekom.de

***Abstract*—Increasing the robustness of networks against failures or traffic changes is an important objective for many operators. Instead of only adapting network configurations to the new conditions in a reactive fashion *after* a critical event has already occurred, proactively hedging networks against those events has gained increasing popularity over the recent years. A variety of approaches have been proposed that implement this concept of proactive robustness using Segment Routing (SR). However, all of those focus solely on the use of conventional end-to-end SR. In this paper, we show that this can be a limiting factor regarding the achievable level of robustness and that utilizing the Midpoint Optimization concept for SR, instead, facilitates (sometimes considerably) more robust durable Traffic Engineering configurations. Furthermore, it also allows to reduce the number of SR policies required to implement the respective configurations. This is not only discussed from a theoretical perspective but also confirmed by evaluations on real-world data from the backbone network of a Tier-1 Internet Service Provider.**

I. Introduction

Increasing a networks robustness against traffic changes or failure scenarios and ensuring proper service in such events is a crucial objective for operators. A traditional way to address this issue are *reactive* approaches. Those aim to quickly bring the network back into an operable state after a problematic failure or considerable traffic changes occurred by adapting the network configuration to the new conditions [5], [8], [20], [23], [29]. However, an inherent problem of these approaches is that they only react *after* the issue is detected and a possible performance degradation already occurred. Furthermore, they require a rather dynamic and continuous re-configuration of a network, which some operators are hesitant to implement, since configuration changes generally carry at least a small risk of introducing other problems (i.e. misconfigurations or hardware issues). Instead, more stable and long living network configurations that are inherently robust and, thus, applicable to a wide range of different scenarios are preferred. Therefore, various publications [6], [21], [27], [28], [30], [36], [39] aim to compute Traffic Engineering (TE) configurations that are resilient against a specified set of failures or traffic changes and, thus, do not require any configuration changes if otherwise critical events occur.

Over the recent years, Segment Routing (SR) has become the premier technology choice across many networks [32] and received a lot of attention, both from research [40] and industry. Recent works [6], [21], [36], [39] have shown that SR can be leveraged to build TE configurations that are robust against a wide set of traffic changes or failure scenarios without requiring any (reactive) reconfiguration of the network. However, all of them focus on conventional end-to-end (E2E) SR. In this paper, we show that the rather rigid and static nature of the latter can be a limiting factor when it comes to obtaining truly robust TE configurations, and how the concept of Midpoint Optimization (MO) for SR [7] can be leveraged to circumvent these issues. These theoretical considerations are further backed up by evaluations on real-world data from the backbone network of a Tier-1 Internet Service Provider (ISP) using our newly proposed Linear Program (LP)-based optimization algorithms for computing robust TE configurations for a given set of TE scenarios. We further show that the respective MO configurations require less SR policies to be implemented, thus, resulting in improved clarity and maintainability of the network and reducing overhead.

The remainder of this paper is structured as follows. First, the relevant fundamentals regarding SR and MO are introduced, followed by a discussion of related work (Sections II and III). After this, in Section IV, we elaborate on why and how the use of the MO-concept facilitates more robust TE configurations than E2E SR. In Section V, we formalize the optimization problem of finding robust TE configurations for a given set of TE scenarios and propose two LP-based approaches to solve it using E2E SR and MO. Our evaluation setup and results are presented in Section VI and further discussed in Section VII. Finally, the paper is concluded in Section VIII with a recapitulation of our key findings and contributions and a brief discussion of future work.

II. Background

Before further discussing potential benefits of MO regarding the robustness and longevity of TE solutions, we introduce the relevant fundamentals regarding MO and SR, in general.

A. Segment Routing

Segment Routing (SR) [14], [15] is a modern implementation of the source routing paradigm. As such, it facilitates controlling a packets path through the network directly at

its respective origin/ingress node. In the context of SR, this is done with so called *SR policies* that can be configured on the respective ingress node. Such a policy defines an *SR path* the packet is steered along by specifying a set of waypoints (also called *segments* or *labels*). Those have to be traversed in the given order before forwarding it to its original destination. There are various types of segments. Originally, they mostly corresponded to the nature of the related waypoint (i.e. *node*, *adjacency*, and *service* segments) but, over time, this list was extended with segments referring to more complex instructions as well (cf. [16]). Despite the plethora of available segment types, most of the SR TE literature (e.g., [6], [23], [35]) focuses solely on the use of a limited number of node segments. While this, in theory, restricts traffic steering capabilities, it has been shown that near-optimal results, in many cases, can already be achieved with just two node segments [6], [35]. Additionally, not relying on other types, like adjacency segments, yields other benefits, as well (e.g., implicit support of Equal Cost Multipath (ECMP) and generally lower optimization complexity). For these reasons, we also focus our considerations on SR with at most two node segments in the remainder of this paper. Furthermore, similar to other works [23], [35], we also prohibit arbitrary traffic splitting over multiple SR paths, as this is generally not implementable in practice due to hardware limitations [35].

A major benefit of SR is the fact that an SR policy only has to be configured on the respective ingress node. All other required information is basically carried by the packet itself. Compared to other traffic steering approaches like Multiprotocol Label Switching (MPLS) [34] with Resource Reservation Protocol (RSVP)-TE [4] this substantially reduces the resulting network overhead and, thus, facilitates improved scalability. For an overview on SR related research, see [1], [40].

B. Midpoint Optimization

In literature, SR is almost exclusively considered in an E2E fashion, meaning that each SR policy is dedicated to route the traffic between just one pair of nodes, its respective start- and endpoint. Other demands that do not originate/end at these nodes but just visit them in transit will not be steered onto the policy. However, from a technical perspective, SR can actually be used in conjunction with other steering mechanisms, as well [16]. For example, [16, Sec. 8.7] suggest the use of *IGP Shortcut* [37] to determine whether a packet is steered onto an existing SR policy, which is already supported in the most recent hard- and software releases of some of the large vendors [12], [26]. This overall concept of stepping away from the E2E nature of conventional SR and allowing other steering mechanisms to be used is often referred to as *Midpoint Optimization (MO)* [7], [11], [12], as it allows traffic to be detoured (or "*optimized*") at arbitrary *midpoints* along its path through the network, instead of only its ingress node.

While this gives up on the fine-grained, per-flow traffic control of E2E SR, it has been shown in [7], [8], [11] that MO still allows for virtually optimal TE solutions that are on-par or even better than those of conventional E2E SR approaches. More importantly, however, it also allows for a substantial reduction in the number of SR policies that are required to implement TE solutions. Such a reduction of policy numbers results in improved clarity and maintainability of the network and also reduces the introduced overhead and, thus, is a relevant objective for many operators.

The general MO concept can be implemented based on various traffic steering mechanisms, but we will limit our considerations to the IGP Shortcut approach for the remainder of this paper. This variation is also considered in other works [7], [8], [11] and is already supported in recent router hard- and software. Its general idea is to steer a packet onto a policy, if the policy tailend is a downstream router with respect to the Interior Gateway Protocol (IGP) path from the policy headend to the packets destination. Or, in other words, if the policy tailend lies on the IGP shortest path from the policy headend to the packets destination, the latter is steered onto the policy.

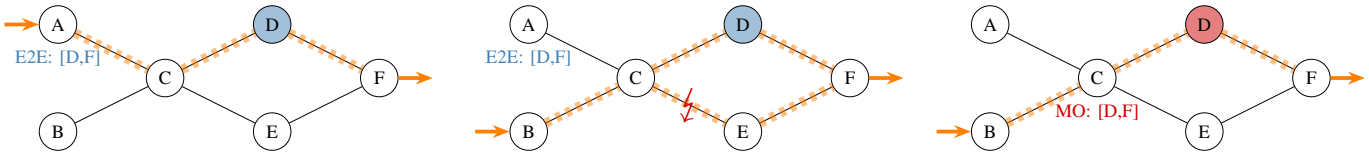
III. Related Work

As mentioned in the introduction, there is a wide range of publications addressing the topic of computing robust and long living TE configurations. With respect to traffic variations, this concept is often referred to as *traffic oblivious routing* [33]. There are multiple approaches that aim to realize it via SR (e.g., [6], [39]) but also in other TE contexts (e.g., [27], [28]). Similar holds true for failure scenarios. Here, Hao et al. [21] and Schüller et al. [36] both propose post-convergence aware optimization algorithms that facilitate the computation of configurations that are robust against a given set of failure scenarios. Their key idea can be summed up as preemptively redistributing traffic flows with SR so that, for each failure scenario, there is sufficient bandwidth available on the respective post-convergence path of the affected traffic flows. Similar proactive failure-resiliency concepts have also been proposed in the context of SDN [30] or even IGP metric optimization [17], as well.

All of the above SR-based approaches focus solely on the use of E2E SR. We are the first to consider the use of the MO concept for this purpose and show that it allows for more robust configurations than E2E SR. Furthermore, basically all approaches focus either exclusively on failures or traffic changes. However, we argue that those actually have to be considered together as they are not mutually exclusive but, in practice, often occur conjointly.

IV. Improving Robustness with MO

While it has already been shown in the literature that MO facilitates TE solutions that are on-par with E2E SR (w.r.t. Maximum Link Utilization (MLU)) but require substantially fewer policies to configure, other potential benefits of MO have not been examined yet. In the following, we show that deploying MO policies instead of conventional E2E SR can facilitate an improved robustness against both traffic changes and failures. This is due to the fact that, while E2E SR offers precise, per-flow traffic control, the rather rigid and static nature of such configurations can be a limiting factor



(a) Policy $A \rightarrow D \rightarrow F$ steers traffic with destination F that enters the network at node A away from the bottom path in order to unburden the link $C \rightarrow E$.

(b) If the ingress point of the flow changes from A to B, the dedicated E2E policy on A is bypassed. Reconfiguration is required to steer traffic away from the bottom path again.

(c) Configuring the MO policy $C \rightarrow D \rightarrow F$ on node C resolves the problem by steering traffic away from the bottom path, irrespective of node A or B being the ingress point.

Figure 1: Simplified illustration on how the use of MO instead of E2E SR results in a more "stable" TE configuration that does not require reconfiguration even if the ingress point of traffic changes. (IGP metric: hop-count)

in more dynamic networking scenarios like those mentioned above. To find a truly robust solution that applies to multiple different scenarios, it can be required to only detour traffic in a subset of those scenarios but not in all. With E2E SR, this is not really feasible (without reactive reconfiguration) since the respective demand is *always* steered onto an E2E policy configured at its ingress node. When using MO, however, policies no longer need to be installed directly at the ingress node but can be configured further down the IGP forwarding path and are not bound to a specific demand anymore. This enables a more dynamic traffic steering in which the set of demands steered over a policy can vary between scenarios without requiring any reconfiguration since a demand can only be detoured if it passes over the respective policies headend. If (i.e. due to a failure), its forwarding path changes and no longer traverses the policy headend, the demand will no longer be detoured. This allows for a more localized traffic detouring that only applies to demands if they traverse a specific slice of the network, instead of always detouring them directly at the ingress node. As a result, E2E configurations often need to be adjusted to adapt to changing conditions, while the more flexible traffic steering of MO facilitates the implementation of TE configurations that are suitable for a larger set of scenarios without requiring any reactive changes. In the following, we further illustrate this based on common scenarios from practical network operation.

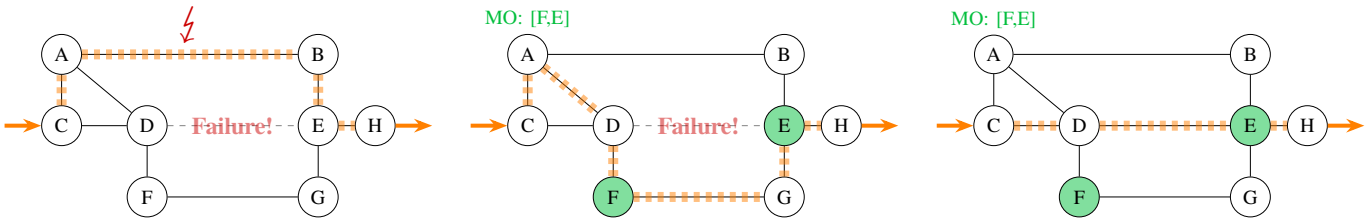
A. Traffic Changes

As most traffic traversing a carrier or ISP network is transit traffic originating and ending in other autonomous systems, it is influenced by interdomain-routing decisions (e.g., BGP or peering [2], [41]) or even by intra-domain routing in other networks [38]. These and other factors (e.g., content provider dynamically switching customer to server assignments [18], [19]), can cause significant shifts in traffic not only changing the volume of certain demands but also there ingress point into the respective ISP network. Such ingress changes, even when geographically rather minute (i.e., changing from one Point of Presence (PoP) in a city to another), can have significant impact, especially when TE is done in an E2E fashion directly at the respective ingress point. In that case, a change in the ingress point basically means that all SR policies configured on the previous ingress node become useless as the traffic no longer passes over this node. With MO, however, SR policies do not have to be configured directly at the respective ingress node anymore but can be installed further "inside" of the network. This renders traffic steering more oblivious to the exact ingress point of a traffic flow and, thus, more robust against changes of the latter, as illustrated in Figure 1.

This example also shows that, even in scenarios where E2E SR is able to match MO regarding the achievable robustness, this can require (sometimes considerably) more policies to be implemented. In Figure 1, a "robust" solution can, in fact, also be achieved with E2E SR by simply configuring a corresponding policy on both ingress nodes (A and B). This, however, would require two policies, while the MO solution achieves the same result with just one. While the difference in policy numbers in this example scenario is rather small, this quickly changes when considering larger networks with hundreds of nodes. Especially ISP backbones often feature PoP structures in which a large set of *edge routers* (possible ingress points for traffic) are connected to only a few core routers to aggregate incoming traffic (cf. [3], [11]). Instead of installing E2E policies on each of these edge routers to handle the possibility of traffic ingress varying between them, an MO policy can be installed on the related core router to cover this.

B. Failures

Another major challenge when aiming for robust TE configurations is proofing them against hardware failures. In this context, utilizing MO can lead to improved longevity by enabling the configuration of policies that a demand is only steered on in certain failure scenarios but not during normal operation (even though the policy is already configured there). It is enabled by the fact that MO policies are no longer E2E but can start at arbitrary points in the network, i.e. further down the forwarding path. This allows to install a policy on a node that the demand does not pass over in normal operation, but only in the presence of a certain (link) failure. With E2E SR, this is not possible. Due to the E2E nature of the policies, the respective demand will always be steered along the configured policy. A simplified example for such a scenario is depicted in Figure 2. Given the respective topology and a traffic flow between nodes C and H that is routed along the shortest path (based on a simple hop-count metric). Without any failures, it results in the forwarding path $C \rightarrow D \rightarrow E \rightarrow H$. However, if the



(a) Post-convergence SPR path after failure results in an overutilization of link $A \rightarrow B$.

(b) The green MO policy at node A redirects traffic flow from the overutilized edge.

(c) Traffic is only detoured in the case of failure but not during normal operation.

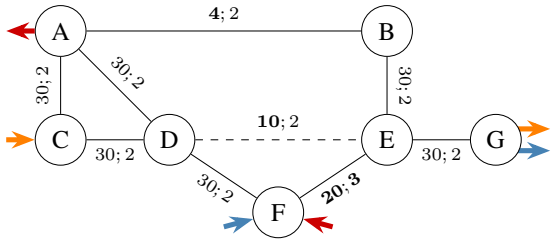
Figure 2: Simplified illustration on how a static MO configuration can be used to detour certain traffic only in the presence of a failure but not during normal operation, without requiring dynamic/reactive configuration changes.

link $D \rightarrow E$ fails, the new, post-convergence shortest path runs over link $A \rightarrow B$, which we assume to be now overutilized due to the additional traffic (cf. Figure 2a). This overutilization can be prevented by steering the traffic flow over the bottom part of the network, instead. This could, for example, be done with an E2E policy $C \rightarrow F \rightarrow H$. However, this would require a change in network configuration (i.e., the addition of the respective policy) as a reaction to the failure. Alternatively, the policy could be preemptively configured, but this would result in the demand also being detoured during normal operation, which might be suboptimal and not desired by the operator. With MO, however, a policy can be configured (already during normal operation) that only detours the demand in the case of failure. This is due to the fact that demand only traverses the policies headend (A) if the respective failure occurs, but not during normal operation (cf. Figures 2b and 2c).

Similar observations can also be made regarding node failures. In fact, our previous traffic shift example (Figure 1) can be easily modified to resemble such a scenario. The respective change of the traffic ingress point can also result from a hardware failure of the primary egress node A which forces the customer or peering partner of the ISP to send traffic to the alternative or backup ingress point B instead.

C. A More Sophisticated Example Scenario

We tried to keep the previous examples (Figures 1 and 2) as simple and comprehensible as possible while still illustrating the general concept and considerations on why and how MO can be beneficial when it comes to finding robust TE configurations. As a result, there actually also exists a sufficiently robust E2E SR configuration, for both of them. To proof that there actually are scenarios for which E2E SR does not allow for a sufficiently robust solution while MO does, we present a slightly more complex example in Figure 3. It basically is a more fleshed out version of the scenario already considered in Figure 2. This time, we combine the failure of a link with a simultaneous shift in traffic demands. Given the respective topology, during normal operation we have three demands as depicted in the included table. As second scenario, we consider the failure of link $D \rightarrow E$ with a simultaneous change in traffic that completely removes the demand between F and G but increases the size of demand $C \rightarrow G$ to 20. The latter could,



Scenario	Failed Link	Demand
I	$\emptyset$	$C \rightarrow G : 10$; $F \rightarrow G : 20$; $F \rightarrow A : 30$
II	$D \rightarrow E$	$C \rightarrow G : 20$; $F \rightarrow A : 30$

Figure 3: Example for a set of just two scenarios for which MO can provide a robust solution but E2E SR cannot. Links are annotated with their capacity and metric values (`cap;metr`).

for example, result from a failure at the network edge (or even outside of the network) that prevents some traffic to ingress at node F. For parts of it, C is chosen as alternative ingress point, resulting in the considered increase of demand $C \rightarrow G$. Even when just considering these two scenarios, there is no E2E SR configuration that can prevent overutilization in both of them. The reason for this is the fact that scenario II requires an E2E policy that sends the demand between C and G over the bottom path ($D \rightarrow F \rightarrow E$). Such a policy, however, is not feasible in scenario I since the outgoing links of F are already fully utilized by the demands ingressing at F and, thus, the respective policy would always result in overutilization. Contrary to this, configuring the MO policy $A \rightarrow F \rightarrow E$ resolves both scenarios as demand $C \rightarrow G$ will only be steered onto it in scenario II but not in scenario I. There, it simply follows the shortest path to its destination.

For completeness, we note that this example does not only hold for 2SR but for SR with an arbitrary number of intermediate segments and even the use of adjacency segments or (practically infeasible) arbitrary traffic splitting, as well.

V. Computing Robust SR Configurations

We have seen that MO, in theory, can facilitate more robust TE configurations than E2E SR. However, the scenarios presented in the previous section are of a more abstract nature and carefully handcrafted to illustrate our theoretical

considerations. Naturally, the question arises whether these observations also translate into practice and MO can yield actual benefits in real-world scenarios. In order to meaningfully examine this, we have to compare results obtained with MO to the optimal E2E SR solution. The optimality of the latter is necessary to ensure that any observed differences can, in fact, be attributed to the used SR variant. Otherwise it might just be a result of the respective algorithm not finding a better E2E SR configuration, even though it theoretically exists.

For this, we first formalize the optimization problem of finding robust TE configurations for a given set of scenarios. Based on this, we then propose LP-based optimization approaches to solve this problem with E2E SR and MO, respectively.

A. Formalizing the Optimization Problem

The overall optimization problem of finding a single TE configuration that is applicable to a set of different scenarios can be formalized as follows. Given a set $\mathcal{S}$ of (consecutive) TE scenarios from a single network, with each scenario s comprising of a topology snapshot G_s and the associated traffic matrix T_s. The topology snapshots resemble the different network states over time featuring information on available capacity and IGP metrics but also on the state of individual links (`failed` or `active`). The goal is to find a single TE/SR configuration $\mathcal{C}$ that minimizes the number of scenarios for which a given MLU threshold Φ is surpassed. In mathematical notation, this can be expressed as:

$$\min \quad \sum_{s \in \mathcal{S}} \pi_s \tag{1a}$$

$$\text{s.t.} \quad \text{load}_e(G_s, T_s, \mathcal{C}) \;\leq\; \theta_s \, c_e(G_s) \qquad \forall e \in G_s \; \forall s \in \mathcal{S} \tag{1b}$$

$$\theta_s \geq \Phi \;\Rightarrow\; \pi_s = 1 \qquad \forall s \in \mathcal{S} \tag{1c}$$

In this context, for each scenario s, $c_e(G_s)$ is the capacity of edge e, $\text{load}_e(G_s, T_s, \mathcal{C})$ denotes the traffic load that is put on edge e when the SR configuration $\mathcal{C}$ is used and θ_s resembles the respective MLU. The π_s indicate whether the respective scenarios MLU surpasses the specified threshold Φ.

Regarding problem complexity, finding an optimal SR solution for just one scenario is already NP-hard [22]. Thus, doing so for multiple scenarios simultaneously is NP-hard, as well.

B. LP-based Algorithms

Having formalized the general optimization problem, we now introduce our LP-based algorithms that solve it for E2E SR and MO, respectively.

1) E2E SR: The respective LP formulation for the E2E SR case is given in Problem 1. It is based on the integer variant of the 2SR formulation [6] which is the de-facto standard formulation for E2E SR with at most two node segments. The binary x_{ij}^k variables resemble the respective policy configuration by indicating whether an SR policy is configured for demand $i \rightarrow j$ over intermediate segment k. Equation 2b ensures that each demand is satisfied. The left side of the capacity constraint (Equation 2c) denotes the total traffic that is put on edge e in scenario s by the SR configuration represented by the x_{ij}^k. In this context, the $g_{ij}^k(G_s, e)$ indicate

$$\min \; \sum_s \pi_s \tag{2a}$$

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

$$\sum_{ij} t_{ij}^s \sum_k g_{ij}^k(G_s, e)\, x_{ij}^k \;\leq\; \theta_s \, c_e(G_s) \qquad \forall e \; \forall s \tag{2c}$$

$$\theta_s - \Phi \;\leq\; M \, \pi_s \qquad \forall s \tag{2d}$$

$$x_{ij}^k \;\in\; \{0, 1\} \qquad \forall ijk \tag{2e}$$

$$\pi_s \;\in\; \{0, 1\} \qquad \forall s \tag{2f}$$

$$\theta_s \;\geq\; 0 \qquad \forall s \tag{2g}$$

Problem 1: LP-formulation to compute a single E2E 2SR policy configuration that minimizes the number of scenarios in which the given MLU threshold Φ is surpassed.

the load that is put on e if a uniform demand is routed from i to j over the intermediate segment k and the t_{ij}^s denote the size of the traffic demand between node i and j. This is then limited by the respective edge capacity $c_e(G_s)$ scaled by θ_s, with the latter resembling the MLU of the respective scenario. The overall objective is to minimize the number of scenarios for which the MLU θ_s surpasses the specified threshold Φ. For this, binary indicator variables π_s are introduced for each scenario that are set to 1 if the MLU threshold is surpassed. Note that, in the given problem formulation we utilize the well-known concept of a so called *big-M constraint* to model this behavior. However, since finding a suitable (and reasonably small) value for M proves to be difficult as it is heavily instance dependent, this constraint is implemented using CPLEX *indicator constraints*, in practice, which is the recommended procedure when no reasonable upper limit for M can be defined. Finally, by minimizing the sum over all π_s (Equation 2a), we obtain the policy configuration with the lowest possible number of threshold violations.

2) MO: We utilize an analogous approach to compute the respective MO solution. However, contrary to E2E SR, there is no provable optimal algorithm for computing a policy configuration that achieves the optimal MLU for a network when using MO and such an algorithms is also rather unlikely to exists [11]. Therefore, we base our LP formulation on the Shortcut 2SR (SC2SR) LP proposed in [7] that, while not guaranteeing optimality, has been shown to provide virtually optimal results for many real-world scenarios. As a result, the MO solutions computed by us might not actually correspond to the true optimum. This is no issue since guaranteed optimality of the MO solution is not strictly required to examine whether MO allows for more robust TE configurations than E2E SR. If even (potentially) non-optimal MO solutions are better than the optimal configurations obtainable with E2E SR, this observation already sufficiently confirms our theoretical considerations from Section IV. For reasons of space and because they are analogous to Problem 1, the respective LP formulation and its detailed description are omitted.

C. Minimizing Policy Numbers

It has been shown in other TE contexts that the use of MO facilitates an often substantial reduction in the number of SR policies required to implement TE solutions [8], [11]. In Section IV-A, we discussed that this probably also applies to the objective of computing robust TE configurations. In order to examine this, we extend the previously described algorithms to also minimize the number of policies required to implement the respective solutions. This is done using a concept similar to the Tunnel Limit Extension (TLE) approach proposed in [35]. After computing a solution that minimizes the number of threshold violations, we adapt the LP to carry out follow-up optimization step that minimizes the number of policies. For Problem 1, we replace the objective function by

$$\min \sum_{k \neq j} x_{ij}^{k} \tag{3}$$

and add the following constraint to the LP

$$\sum \pi_s \leq \Pi \tag{4}$$

This ensures that the number of required policies is minimized while maintaining the same number of threshold violations Π as computed in the first optimization step. The same concept is applied to the MO-based LP, as well.

D. Improving Algorithm Efficiency

Setting up capacity constraints for every edge in each considered scenario is time consuming and can result in quite large LPs of multiple hundred Gigabytes in size, especially when considering MO. However, during early testing, we discovered that actually only a small fraction of edges has to be considered *critical* and is relevant for the solution. Most others are basically always comfortably below the specified utilization threshold. This observation can be utilized to speed up the LP construction and reduce memory requirements by only including capacity constraints that are required to obtain a valid solution. Unfortunately, this set cannot be precomputed in advance, but similar can be achieved by utilizing the iterative solving and setup routine described in Algorithm 1. We start by identifying all edges whose utilization surpasses the specified MLU threshold Φ when no SR policies are configured and adding the respective capacity constraints to the LP. The latter is then solved to obtain a preliminary solution and the associated policy configuration. Based on this, we compute the resulting utilizations of the edges not yet considered in the LP. If those violate the specified threshold Φ, the respective constraints are added to the LP and it is solved again. We repeat this process until there are no threshold violations on unconsidered edges anymore, resulting in an optimal solution. This approach proves to be effective as we often only have to install considerably less than 5% of the total capacity constraints before obtaining a valid (optimal) solution, which results in a substantial reduction in computation time and memory requirements.

Algorithm 1 Iterative LP setup/solving routine.

```
Initialize LP without the capacity constraints
C ← ∅          // Start with an empty policy configuration
E ← ∅          // Keep track of edge/scenario combinations
                  for which a capacity constraint is in the LP
while true do
    isValidSolution ← true
    for s ∈ S do
        for edge e_s in G_s do
            calculate LU of e_s resulting from C
            if LU > Φ and e_s ∉ E then
                add capacity constraint for e_s to LP
                add e_s to E
                isValidSolution ← false
    if isValidSolution then
        return LP-Solution
    else
        (Re-)Solve LP to update policy configuration C
```

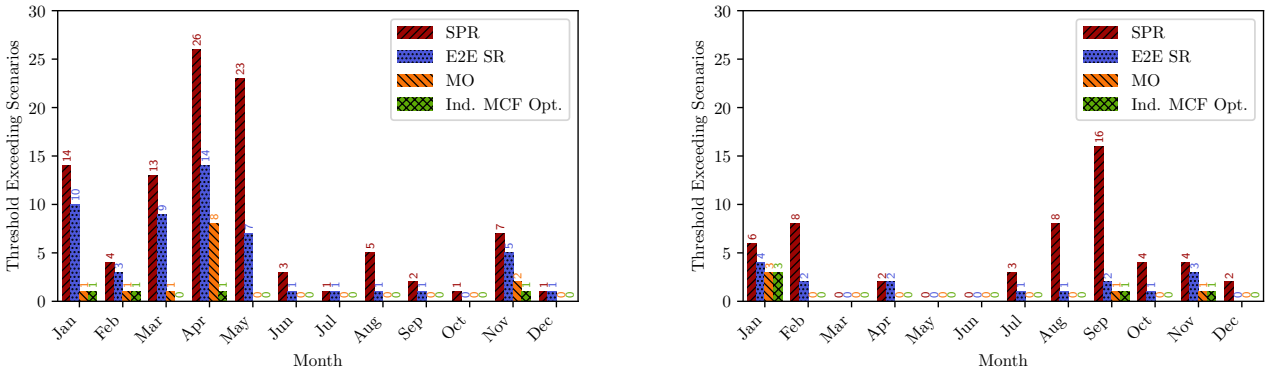
VI. Evaluation

In this section, we evaluate whether the theoretical benefits of MO regarding the robustness of TE solutions described in Section IV also translate into practice. For this, we use our proposed LP-based algorithms (cf. Section V) to compute robust SR configurations for different sets of scenarios based on real real network data from a Tier-1 ISP. All 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 version 20.1.0 [25].

A. Data

For a sensible and realistic analysis regarding the longevity and robustness of solutions, temporally contiguous data is needed that features information on network topology and traffic for multiple consecutive days. For this, we obtained real network data from the backbone network of a globally operating Tier-1 ISP. It consists of snapshots of the network topology (including information on link failures) and the associated traffic matrix out of the peak-hour for each day in 2020 and 2022, respectively. Exact properties vary between snapshots but on average a traffic matrix features 7580 demands and the network comprises of 181 nodes and either 4683 or 1052 edges, depending on whether parallel edges are counted separately or not. The average diameter is 8.44 and the average density (ignoring parallel edges) is around 3%. The used IGP link metrics correspond to those actually used in the ISP network, with the latter being´ set according to a proprietary tuning/optimization procedure.

As this dataset is based on real-world data from an in-production network, it resembles what a robustness optimization has to keep up with for it to actually be of practical use. We also looked into the data regarding the featured traffic and failure characteristics. For reasons of space, we cannot dive into detail here but, overall, results are similar to what has been reported for other ISP backbones as well (cf. e.g.,



(a) 2020.

(b) 2022.

Figure 4: Number of scenarios across the respective one month timespan that surpass the 80% MLU threshold.

[24]). Thus, while our evaluation is based only on data from a singular network, the obtained results should be transferable to other ISP networks (at least to a certain extent), as well.

B. Results – Robustness

For our evaluation, we consider the set of the daily peak-hour snapshots of each month in 2020 and 2022 as the respective set of scenarios that we want to compute a robust SR configuration for and (based on consultation with industry experts) set the MLU threshold to 80% ($\Phi = 0.8$). The results are depicted in Figure 4. For each month, it shows the number of threshold exceeding scenarios of the (optimal) solutions obtained with E2E SR and MO, respectively. The number of threshold violations resulting from plain Shortest Path Routing (SPR) and an individual Multicommodity Flow (MCF) [31, Ch. 4.4] optimization of each scenario are also included. Those function as reference values, the former resembling the default network state without any TE, and the latter a lower bound for what can be achieved when dynamically adapting the network to each scenario individually. The results show a rather high variability between the individual months. Overall, this is a result of the natural variation in traffic changes and occurrence of failure scenarios, but the largest spikes can also often be associated with specific events. For example, in September 2022 there was a larger period (over 1 week) with a higher than usual but still moderate number of failures and one day with a substantially larger spike in link failures. This, presumably, results in a more challenging set of scenarios to find a robust TE configuration for. Similar holds for the period from March to May 2020, which covers the first major peak of the Covid-19 pandemic and related lockdowns. This had significant impact on Internet traffic [13], resulting in a general increase in volume but also continuous changes regarding overall traffic characteristics (e.g., increasing voice/video traffic).

When looking at the individual results, it can be seen that MO outperforms E2E SR for basically every set of considered scenarios. In 19 of the 24 considered months, using MO allows for solutions that are better than those obtainable with E2E SR, meaning they result in fewer threshold violations. For the remaining 5 instances, E2E SR and MO both allow for optimal TE configurations with 0 violations. For some scenarios, the difference between MO and E2E SR are only moderate, with just one or two violations less for MO. However, this is a result of the respective scenarios being less demanding with low numbers in violations for E2E SR and even SPR. Thus, MO does not have that much room for improvement, even when finding the optimal (0 violations) solution. In the more difficult scenario sets for which using E2E SR results in a higher number of violations (i.e. in the first half of 2020), the advantages of MO become more apparent. Here, using MO allows for a considerable reduction in the number of threshold violations compared to E2E SR. Overall, these results impressively demonstrate the ability of MO to facilitate more robust and durable TE configurations, and confirm our theoretical considerations from Section IV.

Having seen that MO consistently outperforms E2E SR, we now take a look at how close the MO solutions are to a truly optimal solution. For this, we compare them to the violation counts resulting from individual optimizations of each scenario with MCF. The latter represent a lower bound for what can at best be achieved with any kind of arbitrarily expressive TE technology. When looking at Figure 4, it can be seen that MO finds the optimal solution (by matching the lower bound of MCF) for all but three of the 24 considered month-long timeperiods. This also shows that, even though our LP-based approach to compute MO configurations is, at least in theory, not guaranteed to find optimal solutions (cf. Section V), it actually does so most of the time. For two (i.e. `Mar20` and `Nov20`) of the three sets of scenarios where MO does not match MCF, the difference is minimal with only one additional violation. Only for the third one (`Apr20`), the difference is considerably higher, probably resulting from it falling directly into the midst of the aforementioned Covid-19 related lockdown and, thereby, constituting an exceptionally challenging set of scenarios. Additionally, the individual MCF optimization is a very optimistic lower bound for which it cannot be guaranteed that there always is a single robust TE configuration that matches it.

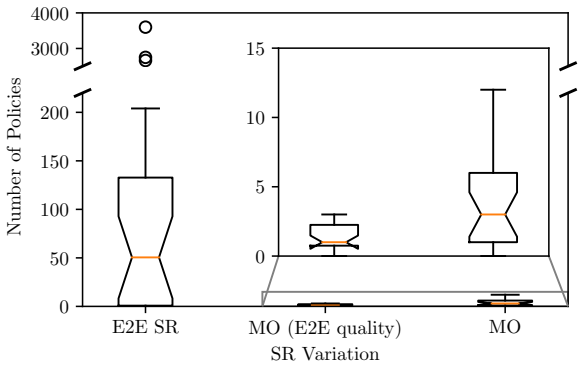


Figure 5: Number of SR policies across all 24 scenario sets.

We also repeated our experiments considering a scenario-set timescale of two instead of just one month. A detailed discussion is omitted since the results are similar to those above, with MO generally outperforming E2E SR and basically always enabling optimal or at least near-optimal solutions.

C. Results – Policy Numbers

Finally, we also look at the minimal number of policies that is required to implement the previously shown solutions. Those are depicted in Figure 5.In addition, it also shows the number of policies that are required by MO to only match the (generally worse) solution quality of E2E SR.[1] This is done to allow for a more fair comparison since better TE solutions generally require more policies. However, it becomes apparent that this is, in fact, not necessary because even the policy numbers required by MO to achieve its (sometimes far) better solutions are considerably lower than those of E2E SR. For most scenarios, only a single digit number of MO policies is needed and when only required to match the E2E solution quality, MO achieves this with less than five polices across all considered sets of scenarios. In contrast to this, multiple tens or even hundreds of policies are required when relying on E2E SR. In the worst case, these numbers can even reach multiple thousands of policies. The latter is (most likely) a result of a large number of demands needing to be detoured combined with a variation in the ingress points of the latter demands. Using E2E SR, this can require an enormous number of policies to be set up at all possible ingress points while MO often allows to implement the relevant detours with just one or two policies further inside the network (cf. Section IV-A). Overall, these results shown that MO is not only able to facilitate more robust TE configurations than E2E SR but also achieves this with a considerably less policies, resulting in less overhead and a generally easier to manage network.

VII. Discussion

The previous evaluation demonstrates the benefits of MO over E2E SR and confirms that those are of actual practical relevance. However, to a certain extend, these results are limited by the fact that they were obtained on data from just a singular network. Preliminary examinations indicate that our data is (to a reasonable extent) representative for other ISP backbones (cf. Section VI-A), it would still be interesting to repeat our evaluations on data from other networks (e.g., WANs or CDNs). This, however, is currently not possible due to the lack of publicly available data that features the required continuous information on traffic and failure scenarios.

Furthermore, in this work, we mainly focus on answering the general question whether MO can facilitate more robust solutions than E2E SR. For this, we assume a basically perfect prediction of the relevant TE scenarios in our evaluations. While recent advancements in the area of machine learning and artificial intelligence might enable increasingly accurate traffic predictions in the future, a perfect prediction remains rather unrealistic. Furthermore, there will always be scenarios that are intrinsically hard to predict (i.e. failures). Thus, when aiming to compute robust solutions in practice, one should probably consider a wider spectrum of *possible* future scenarios during optimization, as it is for example done in [36] regarding failures. Our proposed algorithms should also be applicable for this task without any adaptions and we plan to carry out respective evaluations in the future.

Finally, for reasons of space, we cannot discuss the computation times of our approaches in much detail, but those generally varied heavily between instances (ranging from a couple of minutes to multiple hours). However, since the goal is to precompute long-lasting TE configurations in an offline fashion, even these higher computation times are perfectly acceptable. In addition to that, they can probably be further reduced by utilizing certain preprocessing approaches that aim to preemptively limit the number of SR paths to consider during optimization (cf. e.g., [9]).

VIII. Conclusion

The most important contribution of this paper is the demonstration that utilizing the concept of MO for SR allows for TE configurations that are more robust and, thus, more durable than those obtainable with conventional E2E SR. Our evaluations on real network data from a Tier-1 ISP backbone show that MO consistently outperforms E2E SR for basically all of the considered instances. In most scenarios, our MO-based optimization algorithm is even able to match the theoretically lower bound achievable with individual MCF optimizations, thus providing optimal solutions. Furthermore, MO facilitates this improved robustness while simultaneously requiring substantially fewer SR policies to be configured in the network than E2E SR, resulting in less overhead and an easier to manage network. Together with other findings [7], [8], [10], [11], this demonstrates that the MO concept for SR is a promising technology worth further examination. A first step in this direction is the extension of our evaluation to consider a larger set of *possible* future TE scenarios to hedge the network against, in order to further improve the practical usability of the solutions. We expect the advantages of the more versatile TE capabilities of MO to show there, as well.

[1]The "0" values result from scenario sets where either SPR already results in zero threshold violations or E2E SR is not able to improve upon SPR.

REFERENCES

[1] Z. N. Abdullah, I. Ahmad, and I. Hussain, "Segment Routing in Software Defined Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 464–486, 2019.

[2] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "The Impact of BGP Dynamics on Intra-Domain Traffic," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 319–330, 2004.

[3] D. Alderson, L. Li, W. Willinger, and J. Doyle, "Understanding Internet Topology: Principles, Models, and Validation," *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1205–1218, 2005.

[4] D. O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, 2001.

[5] A. Bashandy, S. Litkowski, C. Filsfils, P. Francois, B. Decraene, and D. Voyer, "Topology Independent Fast Reroute using Segment Routing," Internet Draft draft-ietf-rtgwg-segment-routing-ti-lfa-13, 2024.

[6] 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.

[7] A. Brundiers, T. Schüller, and N. Aschenbruck, "Midpoint Optimization for Segment Routing," in *Proc. of the IEEE Int. Conf. on Computer Communications (INFOCOM)*, 2022, pp. 1579–1588.

[8] ——, "Tactical Traffic Engineering with Segment Routing Midpoint Optimization," in *Proc. of the IFIP Netw. Conf. (NETWORKING)*, 2023, pp. 1–9.

[9] ——, "Preprocess your Paths – Speeding up Linear Programming-based Optimization for Segment Routing Traffic Engineering," in *Proc. of the IFIP Netw. Conf. (NETWORKING)*, 2024, pp. 1–10.

[10] ——, "Combining Midpoint Optimization and Conventional End-to-End Segment Routing for Traffic Engineering," in *Proc. of the IEEE Conf. on Local Computer Networks (LCN)*, 2023, pp. 1–9.

[11] ——, "An Extended Look at Midpoint Optimization for Segment Routing," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1447–1468, 2024.

[12] Cisco Systems, "Cisco WAE Design 7.6.x User Guide," 2022.

[13] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "The Lockdown Effect: Implications of the COVID-19 Pandemic on Internet Traffic," in *Proc. of the ACM Internet Measurement Conf. (IMC)*, 2020, pp. 1–18.

[14] 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.

[15] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, "Segment Routing Architecture," RFC 8402, 2018.

[16] C. Filsfils, K. Talaulikar, D. Voyer, A. Bogdanov, and P. Mattes, "Segment Routing Policy Architecture," RFC 9256, 2022.

[17] B. Fortz and M. Thorup, "Robust Optimization of OSPF/IS-IS Weights," in *Proc. of the Int. Network Optimization Conf. (INOC)*, 2003, pp. 225–230.

[18] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann, "Content-Aware Traffic Engineering," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, pp. 413–414, 2012.

[19] ——, "Content-Aware Traffic Engineering," *ArXiv e-prints*, 2012.

[20] 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.

[21] F. Hao, M. Kodialam, and T. V. Lakshman, "Optimizing Restoration with Segment Routing," in *Proc. of the IEEE Int. Conf. on Computer Communications (INFOCOM)*, 2016, pp. 1–9.

[22] 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.

[23] 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.

[24] G. Iannaccone, C.-n. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of Link Failures in an IP Backbone," in *Proc. of the ACM SIGCOMM Workshop on Internet Measurement*, 2002, pp. 237–242.

[25] IBM, "IBM ILOG CPLEX Optimization Studio 20.1.0," https://www.ibm.com/docs/en/icos/20.1.0, 2020.

[26] Juniper Networks, "Junos OS IS-IS User Guide," 2021.

[27] M. Kodialam, T. V. Lakshman, and S. Sengupta, "Traffic-Oblivious Routing for Guaranteed Bandwidth Performance," *IEEE Communications Magazine*, vol. 45, pp. 46–51, 2007.

[28] ——, "Traffic-Oblivious Routing in the Hose Model," *IEEE/ACM Transactions on Networking*, vol. 19, pp. 774–787, 2011.

[29] P. Kumar, Y. Yuan, C. Yu, N. Foster, R. Kleinberg, P. Lapukhov, C. L. Lim, and R. Soulé, "Semi-Oblivious Traffic Engineering: The Road Not Taken," in *Proc. of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2018, pp. 157–170.

[30] H. H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter, "Traffic Engineering with Forward Fault Correction," in *Proc. of the ACM Conf. on Special Interest Group on Data Communication (SIGCOMM)*, 2014, pp. 527–538.

[31] D. Medhi and K. Ramasamy, *Network Routing: Algorithms, Protocols, and Architectures*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2017.

[32] R. Mota, "Segment Routing Survey," ACG Research, White Paper, 2022.

[33] H. Racke, "Minimizing Congestion in General Networks," in *Proc. of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002, pp. 43–52.

[34] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, 2001.

[35] 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, no. 4, pp. 1851–1864, 2018.

[36] T. Schüller, N. Aschenbruck, M. Chimani, and M. Horneffer, "Failure Resiliency With Only a Few Tunnels – Enabling Segment Routing for Traffic Engineering," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 262–274, 2021.

[37] J. Shen and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels," RFC 3906, 2004.

[38] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of Hot-Potato Routing in IP Networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, pp. 307–319, 2004.

[39] U. Usubütün, M. Kodialam, T. V. Lakshman, and S. Panwar, "Oblivious Routing Using Learning Methods," in *Proc. of the IEEE Global Communications Conf. (GLOBECOM)*, 2023, pp. 5226–5231.

[40] P. L. Ventre, S. Salsano, M. Polverini, A. Cianfrani, A. Abdelsalam, C. Filsfils, P. Camarillo, and F. Clad, "Segment Routing: A Comprehensive Survey of Research Activities, Standardization Efforts, and Implementation Results," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 182–221, 2021.

[41] J. Wu, Z. M. Mao, J. Rexford, and J. Wang, "Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network," in *Proc. of the Symposium on Networked Systems Design & Implementation (NSDI)*, 2005, pp. 1–14.