Traffic Engineering with Segment Routing Considering Probabilistic Failures

Ying Tian*, Zhiliang Wang*, Xia Yin*, Xingang Shi*, Jiahai Yang*, Han Zhang*, Yingya Guo†, and Haijun Geng‡
*Tsinghua University, †Fuzhou University, ‡Shanxi University, P.R. China
E-mails: y-tian18@mails.tsinguua.edu.cn, {wzl, shixg, yang}@cernet.edu.cn
{yxia, zhhan}@tsinghua.edu.cn, guoyy@fzu.edu.cn, ghj123025449@163.com

***Abstract*—Segment Routing (SR) is a source routing paradigm that routes a packet through an ordered list of instructions called segments. It is widely used in Traffic Engineering (TE) because of its simplicity and scalability. Although there are lots of research about TE with SR (SR-TE), fewer consider network failures. The reactive approaches may suffer from latency and update issues, and the proactive approaches don't perform very well because the objectives aren't carefully designed. Besides, although different types of failures are considered, the failure probabilities are ignored. In this paper, we take failure probabilities in to consideration, and propose a proactive 2-SR model 2SRPF to handle SR-TE problem with network failures, aiming at minimizing maximum link utilization (MLU). Considering that severe failures are more noteworthy, we use probability as a severity threshold, and minimize the expectation of the larger MLUs whose corresponding failure states have probabilities sum to a specific threshold value. We solve it with probabilistic risk management. Experiments show that 2SRPF performs well with one threshold setting for different topologies consistently, and gets close to optimal results when network fails.**

***Index Terms*—Traffic Engineering, Segment Routing, Network Failure**

I. Introduction

Traffic engineering (TE) [1] is often considered by Internet Service Providers (ISPs) to optimize network traffic scheduling, reduce congestion and improve network utilization. In traditional IP networks where shortest-path-first Interior Gateway Protocols (IGP) such as Open Shortest Path First (OSPF) runs, TE goals are achieved by setting the link cost properly, which is NP-hard [2] and only offers limited TE capabilities [3]. Multi-Protocol Label Switching (MPLS) and Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) provide a more flexible way by using end-to-end explicit tunnels. However, RSVP-TE requires each hop on the tunnels to maintain states and doesn't support Equal-Cost Multi-Paths (ECMP), causing scalability issues and managing difficulties.

Segment Routing (SR) [4] is a source routing paradigm for IP/MPLS or IPv6 data plane. In SR, the ingress SR node decides the traffic path by encapsulating an ordered list of segments, i.e., segment list, in the packet header. Only the ingress nodes keep per-flow states and path signaling is not needed, which is more scalable and flexible. Each segment is routed by IGP, so SR naturally supports ECMP. When facing with network failure, SR resiliency can be achieved through IGP failure recovery without additional configuration.

There have already been many works that applies SR to TE [3], [5]–[11], but few considers network failure. Topology Independent Loop Free Alternate (TI-LFA) [12] protects segments without considering TE goals. [13] computes robustly disjoint paths with SR considering latency. [14] proposes to reactively re-route the influenced traffic on alternative SR paths. Reactive approaches need re-computation and re-configuration, and may cause latency and network update issues. Thus, some work prefer proactive approaches [15]–[17]: a routing plan is computed and will not be changed when network fails, while the shortest paths used within a segment will be re-computed by IGP. However, the existing proactive works have some drawbacks: i) the optimization objective isn't well designed so the TE performance is impaired. SRR [15] minimizes the biggest maximum link utilization (MLU) over all failures but harms the TE performance of other failure states. PCA2SR [16], [17] proposes that failure states whose MLU is larger than a deterministic threshold Φ is severe and minimizes their average MLU. But the suitable Φ greatly depends on the topology and traffic matrix (TM) and needs a prior knowledge to decide; ii) they only consider different type of failures, but don't consider the probability of failures.

In this paper, we propose a proactive approach 2SRPF that addresses the SR-TE problem with probabilistic network failures, aiming at minimizing the network's MLU. The main contributions of our paper are:

- We are the first to consider probabilistic failures in SR-TE. We first decide the failure state set according to failure probabilities. Then we decide the severe failures based on the comparison of MLU among all failures. Assuming that failure states are sorted by MLU, apparently the ones with larger MLU are severer and noteworthy. 2SRPF uses a probabilistic severity threshold β to decide what proportion of the severer ones should be optimized.
- We are the first to apply probabilistic risk management and the Conditional Value-at-Risk (CVaR) method [18] into SR-TE. We discuss the characteristics of CVaR and how it benefits the TE problem with network failures. We describe how to choose the appropriate probabilistic severity threshold β. Then we apply CVaR to SR-TE and formulate 2SRPF into a CVaR problem.
- We conduct thorough experiments to evaluate 2SRPF. The results demonstrate that our probabilistic severity threshold can effectively distinguish the severe failures, and 2SRPF achieves good TE performance consistently in both topologies with one specific probabilistic severity threshold setting. And the MLU achieved by 2SRPF is close to the theoretical optimum.

978-3-903176-36-2 ©2021 IFIP

The rest of the paper is structured as follows. Section II is a brief background knowledge of SR. Section III motivates our work. Section IV presents our 2-SR TE method considering probabilistic failures 2SRPF. Section V evaluates the performance of 2SRPF. Section VI is the related work. Finally, we make a conclusion in section VII.

II. Segment Routing Background

In SR, a segment is associated with a topological or service-based instruction, represented as Segment Identifier (SID). In this paper, we only consider one kind of topological segment, node segment. By using node segment, each node is assigned a global unique SID, and the packet is steered through an ordered list of middlepoints encapsulated in the segment list. Between two middlepoints, the packet is routed by IGP along the shortest paths. Fig. 1 is an illustration where all links have identical cost. A packet from A to H is routed by SR. Due to space limitation, we omit the detailed description of the routing process. In this example, one middlepoint is specified and two node segment are used, which is called 2-Segment Routing (2-SR) [5]. Compared with using more segment, 2-SR can provide enough TE capability while reducing packet overhead [8] [15]. Thus, this paper focus on 2-SR TE.

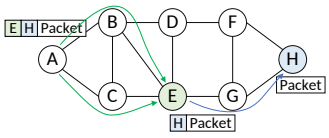


Fig. 1. SR with node segment.

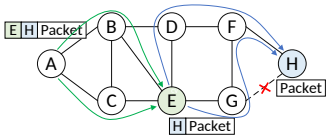


Fig. 2. SR facing a link failure.

When facing network failure, a key feature of SR is it can rely on IGP failure recovery to keep connectivity within a segment. Fig. 2 is an example. Link G-H fails and the shortest path between E and H, i.e., E-G-H, no longer exists. IGP detects the failure and updates the link state database. After IGP convergence, when packet arrives node E and top segment E is popped, the packet still is sent to node H along the shortest path(s) between E and H: not E-G-H, but E-D-F-H and E-G-F-H. Failure recovery is all done by IGP without extra configuration, and SR routing is not affected.

III. Motivation

[15]–[17] use 2-SR routing and handle failures proactively, and take MLU as the TE metric to optimize, which is a widely used metric in ISP networks. SRR [15] uses an approximation algorithm to minimize the maximum MLU over all failures. PCA2SR [16], [17] uses linear programming (LP) to minimize the average value of MLU that is larger than Φ over all failures, and constrains that the non-failure MLU is less than Θ (Φ and Θ are parameters set artificially). In our opinion, these works have some limitations.

1) Minimizing maximum MLU may harm the performance of other failure states: SRR focuses on the worst failure state and minimizes the maximum MLU, but may harm the performance of other failure states. To display this, we conduct

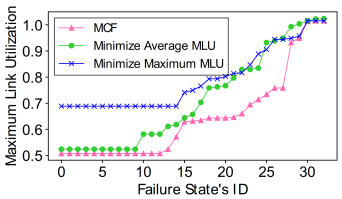
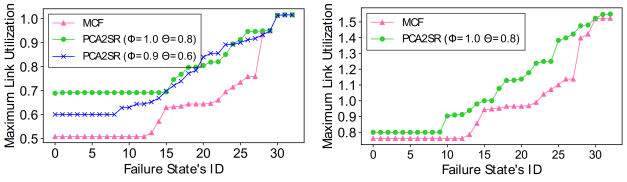


Fig. 3. The MLU of all single link failures with different method in CERNET topology. Failure states are sorted by MLU.



(a) Traffic Scale = 1 (b) Traffic Scale = 1.5

Fig. 4. The performance of [16] in CERNET topology with different threshold settings and traffic scales. Failure states are sorted by MLU.

a small experiment using CERNET topology (14 nodes, 32 links) and a corresponding TM [19]. Considering single link failures, two method are tested: i) minimizing the maximum MLU among all failures; ii) minimizing the average MLU of all failures. Fig. 3 shows the results. Minimizing the maximum MLU achieves the same MLU with MCF's for the max value (1.015), and minimizing the average MLU causes only slightly difference for the max value (1.024). Although minimizing the average MLU may sound naive, it performs better than minimizing the maximum in most of the states, especially in the ones with lower MLU. And it only performs a little worse in a few states.

2) The thresholds Φ and Θ is deterministic and need a prior knowledge to decide: PCA2SR uses Φ to separate the severer states and only optimize MLU of them. But either Φ or the no-failure MLU threshold Θ is a deterministic MLU value, and need a priori knowledge to decide: what MLU value means that the network congestion situation is bad or worse and needs to be optimized? what MLU value should no-failure MLU be lower than? To display this, we test the performance of PCA2SR in CERNET topology. Results with the original traffic scale are shown in Fig. 4. The setting Φ = 1.0, Θ = 0.8 which is used by PCA2SR for its topology doesn't perform well in CERNET when MLU is not that high, and Φ = 0.9, Θ = 0.6 performs better on the whole. However, when the traffic is scaled by 1.5×, Φ = 1.0, Θ = 0.8 performs good, but the LP with Φ = 0.9, Θ = 0.6 becomes infeasible because the no-failure MLU computed by MCF, i.e., the theoretical optimal, is larger than 0.6.

3) The probability of failures are not considered: Although various failure types are considered by SRR and PCA2SR, including link failures, SRLG failures and node failures, none of them considered the probability of failures. The latest work considering failure probability is TeaVaR [20]. Using probability as a measure of availability, TeaVaR maximizes the minimum throughput of all states with a probability sum

like 99%, 99.9%, etc. As far as we are concerned, failure probability is useful in two ways. Firstly, it helps decide the failure state set. Instead of trivially including all single link failures, double link failures etc., it is the failures with considerable probabilities (e.g., with probabilities larger than a value) that should be considered. Secondly, probability can be used as an indication of failure importance. Intuitively, failures with higher probabilities are more important and noteworthy.

Our work 2SRPF takes both failure probability and severity into consideration. We think the severity of a failure comes from its comparison with the others. If we sort all failure states according to their MLU, apparently the ones with larger MLU are severer and more noteworthy. The question is what proportion of the severer ones should be considered severe and optimized, and we propose to use a probabilistic severity threshold β to make the decision: failure states with smaller MLU whose probability sum is approximately β are considered non-severe, and states with larger MLU whose probability sum is approximately $1-\beta$ are severe. The TE objective is to minimize the expectation of MLU of all severe failure states, so severe failures with higher probabilities are considered more. Because the severity threshold we use is not a deterministic MLU value, but a probability value indicating the proportion of failure states to be considered severe, so it saves us from having to know a prior knowledge about the topologies and TMs. Inspired by TeaVaR, we use probabilistic risk management and the CVaR method to solve the problem.

IV. 2-SR Traffic Engineering with Probabilistic Failures

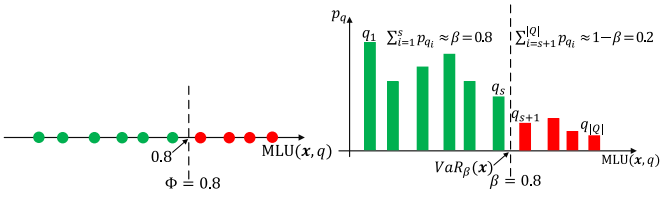
A. Probabilistic Failure Model

We first describe the probabilistic failure model. A network failure event set Z is considered, where each $z \in Z$ is a specific failure event consisting of multiple links. z can represent a single link failure, SRLG failure or node failure, depending on the links it includes, and it occurs with probability p_z.

The network state set Q consists of all network states (whether non-failure or failure) considered. A network state is represented by a 0-1 vector $q = (q_{z_1}, ..., q_{z_{|Z|}})$, where each bit q_z is a binary random variable representing whether failure event z happens ($q_z = 1$) or not ($q_z = 0$). p_q denotes the probability of state q. Knowing p_z, the probability of a specific state q' can be computed as:

$$p_{q'} = p(q_{z_1}=q'_{z_1}, ..., q_{z_{|Z|}}=q'_{z_{|Z|}}) = \prod_{z\in Z}(p_z q'_z + (1-p_z)(1-q'_z)) \quad (1)$$

Estimating failure probability is not the main focus of our paper. We rely on the network operator to provide Z and p_z based on some measurement techniques and analyses [21]–[23], and the state set Q and the probability p_q can be easily computed. $|Q|$ can be as large as $2^{|Z|}$, and increases exponentially with the network size. To deal with the state explosion problem, we use the same state pruning algorithm as [20]. The network states are searched in a depth-first manner. States with a probability p_q larger than the cutoff threshold c are recorded, and states with p_q smaller than c are pruned.



(a) Deterministic threshold Φ (b) Probabilistic threshold β

Fig. 5. Illustration of the deterministic severity threshold Φ and the probabilistic severity threshold β.

We collapse all the pruned states into a cutoff state q_c, and $p_{q_c} = 1 - \sum_{q\in Q_s} p_q$, where $Q_s = Q \backslash \{q_c\}$. In reality, c should be carefully chosen to prevent either state explosion (c is too small) or unacceptable errors (c is too large).

B. The Idea of 2SRPF and the Threshold β

As described earlier, 2SRPF uses a probability threshold β to distinguish severe states, instead of a deterministic MLU value Φ . Now we further explain the idea of 2SRPF and discuss how to choose the appropriate β.

Fig. 5a illustrates the deterministic threshold Φ. For a specific routing configuration $\boldsymbol{x}$, the horizontal axis represents the MLU of a state q, i.e., MLU$(\boldsymbol{x}, q)$, and each point represents a network state. Setting Φ to 0.8 means that states with MLU lower than 0.8 (the green ones) are non-severe, and states with MLU larger than 0.8 (the red ones) are severe. The objective is to find the best routing configuration $\boldsymbol{x}$ to minimize the average MLU of all severe states. Fig. 5b illustrates the probabilistic threshold β. Now the vertical axis represents the probability p_q of a state q, and each column represents a network state. The columns are marked as q_1 to $q_{|Q|}$ in the order of MLU$(\boldsymbol{x}, q)$. Setting β to 0.8 means that states with smaller MLU that has a probability sum approximately equaling to 0.8 are not-severe, and states with larger MLU that has a probability sum approximately equaling to 0.2 are severe[1]. The objective is to find the best routing configuration $\boldsymbol{x}$ to minimize the expectation of MLU of all severe states. Here, the MLU separatrix $\mathrm{VaR}_\beta(\boldsymbol{x})$ (we will explain this notation later) is not a deterministic MLU value, but varies with the probability setting β.

Apparently, the setting of β influences the optimization performance. And there are some constraints on β. Trivially, because β is a cumulative probability, there should be:

$$0 \le \beta \le 1 \quad (2)$$

Assume that we set the MLU of the pruned state q_c to a larger enough value l_c, so that it will not affect the judgment of the severity of the actual state, because q_c has the largest MLU and is always the most severe one. Then the rightmost column (state) in Fig. 5b, $q_{|Q|}$, is q_c, and its horizontal coordinate is l_c.

[1]We use the approximate equality ($\approx$) to imply that $\sum_{i=1}^{s} p_{q_i}$ doesn't necessarily equal to β, because MLU$(\boldsymbol{x}, q)$ is discontinuous with respect to $\boldsymbol{x}$ and q. It satisfies the requirements that $\sum_{i=1}^{s} p_{q_i} \ge \beta$ and $\sum_{i=1}^{s-1} p_{q_i} < \beta$. And $\sum_{i=s+1}^{|Q|} p_{q_i} = 1 - \sum_{i=1}^{s} p_{q_i}$.

To effectively distinguish non-severe and severe states, there should be:

$$\beta \leq 1 - p_{q_c} = \sum_{q \in Q_s} p_q \tag{3}$$

We propose to set β as Eq. (4). q_{nf} denotes the non-failure state. Q_f is the set of all failure states and $Q_f = Q \backslash \{q_{nf}, q_c\}$. Considering that for any routing plan $\boldsymbol{x}$, the non-failure state q_{nf} usually has a relatively small MLU and should be considered non-severe, i.e. $\mathrm{MLU}(\boldsymbol{x}, q_{nf}) \leq VaR_\beta(\boldsymbol{x})$[2], we assume that β is at least $p_{q_{nf}}$. The factor $\gamma \in [0, 1]$ controls what percentage of failure states are considered in the optimization objective in terms of probability. Different γ values have different degrees of discrimination for the states. Large γ means that only a limited amount of states are considered severe and included in the optimization objective. Small γ means that more states are considered severe. To achieve good performance, γ should be carefully chosen based on the operator's optimization intention.

$$\beta = p_{q_{nf}} + \gamma \sum_{q \in Q_f} p_q \tag{4}$$

C. Probabilistic Risk Management

Probabilistic risk management is first studied in financial contexts [18], [24], [25], and is introduced into TE field by TeaVaR [20]. A loss is defined as a function $f(\boldsymbol{x}, \boldsymbol{y})$ of a decision vector $\boldsymbol{x} \in X \subseteq \mathbb{R}^n$ representing the decision and a uncertainty vector $\boldsymbol{y} \in Y \subseteq \mathbb{R}^m$ representing the stochastic environment. $\boldsymbol{y}$ is a random variable. Knowing the probability density function $p(\boldsymbol{y})$ of $\boldsymbol{y}$, the probability of loss $f(\boldsymbol{x}, \boldsymbol{y})$ not exceeding a threshold α is:

$$\Psi(\boldsymbol{x}, \alpha) = \int_{f(\boldsymbol{x}, \boldsymbol{y}) \leq \alpha} p(\boldsymbol{y}) d\boldsymbol{y} \tag{5}$$

Value-at-Risk (VaR) [24] is a measure of risk defined by the worst loss with a specific probability. Specifically, VaR_β is the lowest α value ensuring that the loss will not exceed α with probability $\beta (0 < \beta < 1)$, defined as:

$$\mathrm{VaR}_\beta(\boldsymbol{x}) = \min\{\alpha \in \mathbb{R} | \Psi(\boldsymbol{x}, \alpha) \geq \beta\} \tag{6}$$

Conditional Value-at-Risk (CVaR) [18] is an alternative measure that quantifies the losses in the tail. Specifically, CVaR_β is the conditional expectation of loss no-less than VaR_β, defined as:

$$\begin{aligned} \mathrm{CVaR}_\beta(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{y}) | f(\boldsymbol{x}, \boldsymbol{y}) \geq \mathrm{VaR}_\beta(\boldsymbol{x})] \\ &= (1-\beta)^{-1} \int_{f(\boldsymbol{x}, \boldsymbol{y}) \geq \mathrm{VaR}_\beta(\boldsymbol{x})} f(\boldsymbol{x}, \boldsymbol{y}) p(\boldsymbol{y}) d\boldsymbol{y} \end{aligned} \tag{7}$$

Furthermore, [18] proposes a convex and continuously differentiable function F_β on $X \times \mathbb{R}$ associating $\mathrm{VaR}_\beta(\boldsymbol{x})$ and $\mathrm{CVaR}_\beta(\boldsymbol{x})$, defined as:

$$F_\beta(\boldsymbol{x}, \alpha) = \alpha + (1-\beta)^{-1} \int_{\boldsymbol{y} \in \mathbb{R}^m} [f(\boldsymbol{x}, \boldsymbol{y}) - \alpha]^+ p(\boldsymbol{y}) d\boldsymbol{y} \tag{8}$$

[2]This inequality does not necessarily hold. It is only valid under normal circumstances.

Minimizing $\mathrm{CVaR}_\beta(\boldsymbol{x})$ over $\boldsymbol{x} \in X$ for a specific β is equivalent to minimizing $F(\boldsymbol{x}, \alpha)$ over $(\boldsymbol{x}, \alpha) \in X \times \mathbb{R}$, i.e,

$$\min_{\boldsymbol{x} \in X} \mathrm{CVaR}_\beta(\boldsymbol{x}) = \min_{(\boldsymbol{x}, \alpha) \in X \times \mathbb{R}} F(\boldsymbol{x}, \alpha) \tag{9}$$

Typically, if $(\boldsymbol{x}^*, \alpha^*)$ minimizes $F(\boldsymbol{x}, \alpha)$ over $(\boldsymbol{x}, \alpha) \in X \times \mathbb{R}$, then $\boldsymbol{x}^*$ also minimizes $\mathrm{CVaR}_\beta(\boldsymbol{x})$ over $\boldsymbol{x} \in X$, and α^* gives the value of $\mathrm{VaR}_\beta(\boldsymbol{x}^*)$.

Minimizing $\mathrm{CVaR}_\beta(\boldsymbol{x})$, i.e, minimizing $F(\boldsymbol{x}, \alpha)$, is often used as a proxy for minimizing $\mathrm{VaR}_\beta(\boldsymbol{x})$, because of VaR's lacking of subadditivity and convexity. However, the characteristic of CVaR itself is overlooked. Focusing only on the expectation of losses greater than VaR_β, it distinguishes between states that are worthy of attention ($f(\boldsymbol{x}, \boldsymbol{y}) > \mathrm{VaR}_\beta(\boldsymbol{x})$) and those that are less noteworthy ($f(\boldsymbol{x}, \boldsymbol{y}) \leq \mathrm{VaR}_\beta(\boldsymbol{x})$). On the surface, it appears that $\mathrm{VaR}_\beta(\boldsymbol{x})$ is the deterministic threshold. But actually, β performs as a probabilistic threshold and makes the distinction. The states that has larger losses with probability sum 1-β is considered severe, and the states that has lower losses is ignored when computing $\mathrm{CVaR}_\beta(\boldsymbol{x})$. This exactly consists with the idea of 2SRPF.

D. 2SRPF with CVaR

Now we describe how to apply CVaR to 2SRPF. The network is represented by a graph $G = (V, E)$, where V denotes the SR router set and E denotes the directed links set. $c(e)$ denotes the capacity and $w(e)$ denotes the cost of each link $e \in E$. d is the TM, and $d_{i,j}$ denotes the aggregate amount of traffic between node i and node j. Traffic flows are 2-SR routed, i.e., each split flow has at most one middlepoint to pass through. $\boldsymbol{x}$ is the decision vector, and $x_{i,j}^k$ denotes the amount of traffic between node i and node j that goes through node k. $\boldsymbol{y}$ is the uncertainty vector, and $y_{i,j}^{k,e}$ represents the fraction of traffic on link e when the traffic is routed through shortest paths in an ECMP manner between i and k, k and j. Knowing G and $w(e)$, $y_{i,j}^{k,e}$ is easily computed. Apparently, $y_{i,j}^{k,e}$ is a random variable about network state q, so we also use the notation $y_{i,j}^{k,e}(q)$ and $\boldsymbol{y}(q)$. If there is no path between i and k or k and j in a failure state q', we assume that the traffic is routed along the shortest paths between i and j, and we set $y_{i,j}^{k,e}(q')$ to $y_{i,j}^{j,e}(q')$. If there is even no path between i and j in some state q', we just set $\boldsymbol{y}(q')$ to $\mathbf{0}$ because the traffic between i and j cannot be routed.

First we define the domain X of $\boldsymbol{x}$. We assume that all traffic flows are fully routed in the network. Trivially, $x_{i,j}^k$ is a non-negative real number. We have:

$$\sum_{k \in V} x_{i,j}^k \geq d_{i,j} \quad \forall i, j \in V \tag{10}$$

$$x_{i,j}^k \geq 0 \quad \forall i, j, k \in V \tag{11}$$

Then we define the loss function $f(\boldsymbol{x}, \boldsymbol{y})$. Like SRR and PCA2SR, we also take MLU as the TE metric, which is widely used in ISP network to balance traffic and reduce congestion. We use $\mathrm{MLU}(\boldsymbol{x}, q)$ as the loss function. The utilization of a link e is given by $\frac{\sum_{i,j \in V} \sum_{k \in V} x_{i,j}^k y_{i,j}^{k,e}(q)}{c(e)}$, the loss function is defined as the MLU among all links, i.e.:

$$f(\boldsymbol{x},\boldsymbol{y}(q)) = \mathrm{MLU}(\boldsymbol{x},q) = \max_{e\in E}\frac{\sum_{i,j\in V}\sum_{k\in V}x_{i,j}^{k}y_{i,j}^{k,e}(q)}{c(e)} \quad \forall q\in Q_s \tag{12}$$

Besides, as said in § IV-B, the pruned state q_c has the largest loss l_c that satisfies:

$$f(\boldsymbol{x},\boldsymbol{y}(q_c)) = l_c \quad \forall \boldsymbol{x}\in X \tag{13}$$

$$f(\boldsymbol{x},\boldsymbol{y}(q)) < l_c \quad \forall \boldsymbol{x}\in X, q\in Q_s \tag{14}$$

Taking minimizing $F_\beta(\boldsymbol{x},\alpha)$ as optimization objective so as to minimize $\mathrm{CVaR}_\beta(\boldsymbol{x})$, the problem can be formulated as:

$$\min \quad F_\beta(\boldsymbol{x},\alpha)$$
$$\text{s.t.} \quad \text{Eq. (8)}, \text{Eq. (10)} - \text{Eq. (12)}$$

However, this problem is non-linear and not easy to solve. Firstly, the probability distribution $p(\boldsymbol{y})$ is unknown, so the integral $\int_{\boldsymbol{y}\in\mathbb{R}^m}[f(\boldsymbol{x},\boldsymbol{y})-\alpha]^+p(\boldsymbol{y})d\boldsymbol{y}$ in Eq. (8) cannot be computed. Knowing that $\boldsymbol{y}$ is a random variable about network state q and the probability p_q, we approximate $F_\beta(\boldsymbol{x},\alpha)$ by using all states $q\in Q$ to sample the probability distribution $p(\boldsymbol{y})$, and the integral is approximated as $\sum_{q\in Q}p_q[f(\boldsymbol{x},\boldsymbol{y}(q))-\alpha]^+$. Secondly, the $[\ ]^+$ function in Eq. (8) and the $\max$ function in Eq. (12) is non-linear. We introduce two auxiliary variables s_q and $u_{e,q}$, and the problem formulation is rewritten as an linear programming problem:

$$\begin{aligned}
\min \quad & \alpha + (1-\beta)^{-1}\sum_{q\in Q}p_q s_q \\
\text{s.t.} \quad & \sum_{k\in V}x_{i,j}^{k} \geq d_{i,j} && \forall i,j\in V \\
& s_q + \alpha \geq u_{e,q} && \forall e\in E, q\in Q \\
& u_{e,q} = \frac{\sum_{i,j\in V}\sum_{k\in V}x_{i,j}^{k}y_{i,j}^{k,e}(q)}{c(e)} && \forall e\in E, q\in Q \\
& x_{i,j}^{k} \geq 0 && \forall i,j,k\in V \\
& s_q \geq 0 && \forall q\in Q
\end{aligned}$$

V. Evaluation

A. Setup

1) Topology: We use two topologies and their corresponding TMs: China Education and Research Network (CERNET, 14 nodes, 32 links) [19], and Europe Research and Education Network (GÉANT, 23 nodes, 74 links) [26]. To better display the experiment results, we scale the TMs so that the MLU is 0.8 when the traffic are routed by OSPF.

2) Failure and Probability: Lacking of information about SRLG configuration, we assure that each failure event z consists of a single link failure. Note that a failure state q is consist of multiple failure events z, thus our states set Q still may include multiple link failure. And we make the same assumption as TeaVaR that a link's failure probability fits the Weibull distribution $W(\lambda,k)$. The shape parameter k is set to 0.8 as TeaVaR, and the scale parameters λ are set to 10^{-4}, 10^{-3} and 10^{-2} respectively to simulate different scales of failure probabilities. The cutoff threshold c is set to 10^{-6}, which is the smallest value considered by TeaVaR. In Table I, we record some related data.

TABLE I
THE PROBABILITY OF THE NON-FAILURE STATE $p_{q_{nf}}$, THE PROBABILITY SUM OF ALL STATES EXCEPT FOR THE CUTOFF STATE $\sum_{q\in Q_s}p_q$, AND THE NUMBER OF STATES EXCEPT FOR THE CUTOFF STATE $|Q_s|$ WITH DIFFERENT λ

(a) CERNET

$\boldsymbol{\lambda}$	$\boldsymbol{p_{q_{nf}}}$	$\boldsymbol{\sum_{q\in Q_s}p_q}$	$\boldsymbol{\lvert Q_s\rvert}$
$\mathbf{10^{-4}}$	0.996565	0.999995	34
$\mathbf{10^{-3}}$	0.955077	0.999897	212
$\mathbf{10^{-2}}$	0.752347	0.999081	1160

(b) GÉANT

$\boldsymbol{\lambda}$	$\boldsymbol{p_{q_{nf}}}$	$\boldsymbol{\sum_{q\in Q_s}p_q}$	$\boldsymbol{\lvert Q_s\rvert}$
$\mathbf{10^{-4}}$	0.99241	0.999971	73
$\mathbf{10^{-3}}$	0.907771	0.999413	973
$\mathbf{10^{-2}}$	0.385099	0.980444	11636

3) Algorithms: We compare the following algorithms:

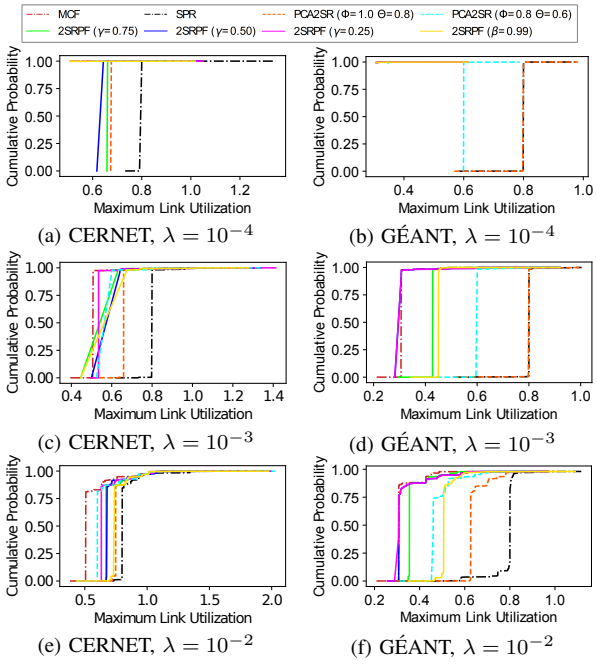
- *2SRPF*. We solve the LP in § IV-D with different β (γ) setting. When failure happens, the IGP shortest paths are re-computed but the 2-SR paths keep un-changed.
- *PCA2SR*. 2-SR traffic paths are computed using PCA2SR. When failure happens, the IGP shortest paths are re-computed but the 2-SR paths doesn't change.
- *Shortest Path Routing (SPR)*. For each network state, traffic is routed on the IGP shortest paths.
- *Multi-commodity Flow (MCF)*. For each network state, the MCF problem is solved as theoretical lower bound.

All algorithms are implemented with C++ and Gurobi [27]. Computations are performed on a server with 12 core 2.3GHz Intel CPU and 256G memory.

B. The Performance with Different Probabilistic Threshold β

We compute β with the factor γ as said in § IV-B, and three different γ values are evaluated: 0.25, 0.50 and 0.75. Knowing the data in Table I, the corresponding β values are easily computed. Besides, we also evaluate $\beta = 0.99$ like TeaVaR. For GÉANT with $\lambda = 10^{-2}$, because $\sum_{q\in Q_s}p_q < 0.99$, so we evaluate $\beta = 0.98$. We run 2SRPF, PCA2SR, SPR and MCF, and draw the CDF figure of MLU of all states $q\in Q_s$. The results are shown in Fig. 6, where some curves are close or coincide because the algorithms performs so close. In Fig. 6f, the curve "2SRPF ($\beta = 0.99$)" actually shows the result with $\beta = 0.98$.

Firstly, 2SRPF performs better than SPR with all β (γ) settings in Fig. 6, which means that 2SRPF can effectively minimize MLU and avoid congestion. Secondly, the results show that different γ makes different proportion of the failures included in the objective function, thus performs differently. To better display the result, we take GÉANT with $\lambda = 10^{-2}$ for example, and show the MLU of all network states with $\gamma = 0.25$ and $\gamma = 0.75$ in Fig. 7. We also mark the network states that are considered severe and included in the objective function. $\gamma = 0.25$ makes more failures considered severe,



(a) CERNET, $\lambda = 10^{-4}$ (b) GÉANT, $\lambda = 10^{-4}$

(c) CERNET, $\lambda = 10^{-3}$ (d) GÉANT, $\lambda = 10^{-3}$

(e) CERNET, $\lambda = 10^{-2}$ (f) GÉANT, $\lambda = 10^{-2}$

Fig. 6. The CDF of network states' MLU for CERNET and GÉNT .

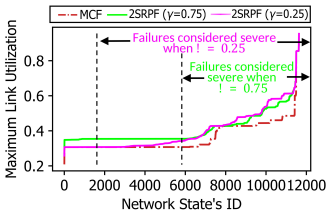


Fig. 7. The MLU of all network states in GÉANT with $\lambda = 10^{-2}$. The states are sorted by their MLU.

and it performs better in the failures whose MLU is relatively low, which is also shown in Fig. 6. However, when it comes to the failures that has larger MLU, $\gamma = 0.75$ performs better, because it only focuses on the small amount of severest failures. So, if the operator wants to only optimize the severest failure states, we recommend a higher γ value like 0.75. If the operator concerns more about the overall performance, a lower γ value like 0.25 may be more appropriate.

C. Comparison of Probabilistic and Deterministic Threshold

We compare our probabilistic threshold method 2SRPF with the deterministic threshold method PCA2SR. We use two parameter settings for PCA2SR: i) $\Phi = 1.0, \Theta = 0.8$; ii) $\Phi = 0.8, \Theta = 0.6$. The first setting is used by PCA2SR itself in its experiments. Still, Fig. 6 shows the CDF figure.

2SRPF with all γ setting outperforms PCA2SR ($\Phi = 1.0, \Theta = 0.8$) in both topologies with all λ settings. As said in § III-2, the appropriate deterministic threshold Φ and Θ vary with topologies and TMs. Although $\Phi = 1.0, \Theta = 0.8$ is suitable for the topology and TM used by PCA2SR, it performs far from good in CERNET and GÉANT. PCA2SR ($\Phi = 0.8, \Theta = 0.6$) performs well in CERNET. Its results are close to 2SRPF, and it performs a little better than 2SRPF in the network states with lower MLU when $\gamma = 10^{-2}$. But it still performs not well in GÉANT. Maybe with some carefully chosen thresholds, like by running some pre-tests, PCA2SR can performs better in GÉANT. Anyway, the deterministic thresholds needs to be carefully decided for a specific topology and TM. While our probabilistic threshold needs to be decided according to one's optimization intention, and one γ setting performs consistently well for different topologies and TMs, which helps when there is a lack of prior knowledge.

TABLE II
THE COMPUTATION TIME (S) OF 2SRPF ($\gamma = 0.25$)

Topo. \ λ	10^{-4}	10^{-3}	10^{-2}
CERNET	2.00	13.67	94.32
GÉNAT	40.48	1130.72	15584.40

D. Computation Time

We take $\gamma = 0.25$ for example and record the computation time of different topologies and λ settings in Table II. For the same λ setting, larger topology consumes more computation time. For the same topology, larger λ causes longer computation time, and the computation time is approximately proportional to the number of states $|Q_s|$. Obliviously, 2SRPF is only suitable for offline optimization, especially for large topologies or when there is a larger number of network states.

VI. Related Work

A. Traffic Engineering with Segment Routing

For SR-TE without network failures, mainly two kinds of methods are used. The first one is Linear Programming (LP) [5], [28]–[31]. LP guarantees the optimality of the results, but can be very time-consuming. The second one is heuristic algorithms, which is a lot faster. [6], [7] use Local Search. [32] uses column generation based heuristic. There are also works concerning partially deployed SR [3], [11].

B. Traffic Engineering Considering Network Failures

Works about TE with network failure considers different network scenarios: traditional IP network [33], tunnel-based routing (MPLS or SDN) network [20], [34]–[38], and of course SR network which will be introduced in the next subsection. There are three kinds of approaches used: reactive, proactive [20], [33], [36], [38], and proactive-reactive combined [34], [35], [37].

C. SR-TE Considering Network Failures

Topology Independent Loop Free Alternate (TI-LFA) [12] doesn't consider TE, it computes repair paths to solve the micro-loop problem. [7] computes the routing policy fast reactively. There are proactive-reactive combined works [13], [14]. There are also pure proactive works [15]–[17]. Our work is proactive. We take a step forward to consider failure probabilities and use a probabilistic severity threshold β, which doesn't need a prior knowledge to decide.

VII. Conclusion

In this paper, we take failure probability and severity in to consideration and propose 2SRPF to handle SR-TE problem with network failures. In the future, we consider implementing and testing our model in a testbed. And we will also study faster algorithms that can be use online.

Acknowledgment

This work is supported in part by the National Key Research and Development Program of China under Grant No. 2018YFB1800400 and the National Natural Science Foundation of China (61802092).

References

[1] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional ip routing protocols," *IEEE communications Magazine*, vol. 40, no. 10, pp. 118–124, 2002.

[2] B. Fortz and M. Thorup, "Increasing internet capacity using local search," *Computational Optimization and Applications*, vol. 29, no. 1, pp. 13–48, 2004.

[3] A. Cianfrani, M. Listanti, and M. Polverini, "Incremental deployment of segment routing into an isp network: a traffic engineering perspective," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 3146–3160, 2017.

[4] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The segment routing architecture," in *Global Communications Conference (GLOBECOM), 2015 IEEE*, 2015.

[5] R. Bhatia, F. Hao, M. Kodialam, and T. Lakshman, "Optimized network traffic engineering using segment routing," in *IEEE INFOCOM*. IEEE, 2015, pp. 657–665.

[6] 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," *ACM SIGCOMM computer communication review*, vol. 45, no. 4, pp. 15–28, 2015.

[7] S. Gay, R. Hartert, and S. Vissicchio, "Expect the unexpected: Sub-second optimization for segment routing," in *IEEE INFOCOM*. IEEE, 2017, pp. 1–9.

[8] 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.

[9] M. Jadin, F. Aubry, P. Schaus, and O. Bonaventure, "Cg4sr: Near optimal traffic engineering for segment routing with column generation," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 1333–1341.

[10] X. Li and K. L. Yeung, "Traffic engineering in segment routing networks using milp," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1941–1953, 2020.

[11] Y. Tian, Z. Wang, X. Yin, X. Shi, Y. Guo, H. Geng, and J. Yang, "Traffic engineering in partially deployed segment routing over ipv6 network with deep reinforcement learning," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1573–1586, 2020.

[12] S. Litkowski, A. Bashandy, C. Filsfils, B. Decraene, P. Francois, D. Voyer, F. Clad, and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing," Internet Engineering Task Force, Internet-Draft draft-ietf-rtgwg-segment-routing-ti-lfa-04, Aug. 2020, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segment-routing-ti-lfa-04

[13] F. Aubry, S. Vissicchio, O. Bonaventure, and Y. Deville, "Robustly disjoint paths with segment routing," in *Proceedings of the 14th international conference on emerging networking experiments and technologies*, 2018, pp. 204–216.

[14] V. Pereira, M. Rocha, and P. Sousa, "Traffic engineering with three-segments routing," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1896–1909, 2020.

[15] F. Hao, M. Kodialam, and T. Lakshman, "Optimizing restoration with segment routing," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[16] T. Schüller, N. Aschenbruck, M. Chimani, and M. Horneffer, "Failure resilient traffic engineering using segment routing," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 422–429.

[17] 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*, pp. 1–13, 2020.

[18] R. T. Rockafellar, S. Uryasev *et al.*, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000.

[19] B. Zhang, J. Bi, J. Wu, and F. Baker, "Cte: cost-effective intra-domain traffic engineering," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 115–116.

[20] J. Bogle, N. Bhatia, M. Ghobadi, I. Menache, N. Bjørner, A. Valadarsky, and M. Schapira, "Teavar: striking the right utilization-availability balance in wan traffic engineering," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 29–43.

[21] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, "California fault lines: understanding the causes and impact of network failures," in *Proceedings of the ACM SIGCOMM 2010 Conference*, 2010, pp. 315–326.

[22] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: measurement, analysis, and implications," in *Proceedings of the ACM SIGCOMM 2011 Conference*, 2011, pp. 350–361.

[23] J. Meza, T. Xu, K. Veeraraghavan, and O. Mutlu, "A large scale study of data center network reliability," in *Proceedings of the Internet Measurement Conference 2018*, 2018, pp. 393–407.

[24] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk*. McGrawHill, 2001. [Online]. Available: https://books.google.com.my/books?id=S2SsFblvUdMC\&redir_esc=y

[25] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of banking & finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[26] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.

[27] Gurobi optimization. [Online]. Available: http://www.gurobi.com

[28] G. Trimponias, Y. Xiao, H. Xu, X. Wu, and Y. Geng, "On traffic engineering with segment routing in sdn based wans," *arXiv preprint arXiv:1703.05907*, 2017.

[29] E. Moreno, A. Beghelli, and F. Cugini, "Traffic engineering in segment routing networks," *Computer Networks*, vol. 114, pp. 23–31, 2017.

[30] G. Trimponias, Y. Xiao, X. Wu, H. Xu, and Y. Geng, "Node-constrained traffic engineering: Theory and applications," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1344–1358, 2019.

[31] X. Li and K. L. Yeung, "Traffic engineering in segment routing networks using milp," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1941–1953, 2020.

[32] M. Jadin, F. Aubry, P. Schaus, and O. Bonaventure, "Cg4sr: Near optimal traffic engineering for segment routing with column generation," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1333–1341.

[33] B. Fortz, M. Thorup *et al.*, "Robust optimization of OSPF/IS-IS weights," in *Proc. INoC*, vol. 20, 2003, pp. 225–230.

[34] Y. Wang, H. Wang, A. Mahimkar, R. Alimi, Y. Zhang, L. Qiu, and Y. R. Yang, "R3: resilient routing reconfiguration," in *Proceedings of the ACM SIGCOMM 2010 conference*, 2010, pp. 291–302.

[35] 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 *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018, pp. 157–170.

[36] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford, "Network architecture for joint failure recovery and traffic engineering," *ACM SIGMETRICS Performance Evaluation Review*, vol. 39, no. 1, pp. 97–108, 2011.

[37] J. Zheng, H. Xu, X. Zhu, G. Chen, and Y. Geng, "Sentinel: failure recovery in centralized traffic engineering," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1859–1872, 2019.

[38] H. H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter, "Traffic engineering with forward fault correction," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 527–538.