Traffic Engineering with Segment Routing under Uncertain Failures

Zengwei Zheng[1], Chenwei Zhao[2], Jianwei Zhang[1*], and Jianping Cai[1]
[1] School of Computer & Computing Science, Zhejiang University City College
Hangzhou, 310015, China
[Email: zhengzw@zucc.edu.cn, janyway@outlook.com, caijp@zucc.edu.cn]
[2] College of Computer Science & Technology, Zhejiang University
Hangzhou, 310027, China
[Email: chenweizhao@zju.edu.cn]
*Corresponding author: Jianwei Zhang

Received January 29, 2021; revised March 18, 2021; accepted June 23, 2021;
published July 31, 2021

Abstract

Segment routing (SR) is a highly implementable approach for traffic engineering (TE) with high flexibility, high scalability, and high stability, which can be established upon existing network infrastructure. Thus, when a network failure occurs, it can leverage the existing rerouting methods, such as rerouting based on Interior Gateway Protocol (IGP) and fast rerouting with loop-free alternates. To better exploit these features, we propose a high-performance and easy-to-deploy method SRUF (Segment Routing under Uncertain Failures). The method is inspired by the Value-at-Risk (VaR) theory in finance. Just as each investment risk is considered in financial investment, SRUF also considers each traffic distribution scheme's risk when forwarding traffic to achieve optimal traffic distribution. Specifically, SRUF takes into account that every link may fail and therefore has inherent robustness and high availability. Also, SRUF considers that a single link failure is a low-probability event; hence it can achieve high performance. We perform experiments on real topologies to validate the flexibility, high-availability, and load balancing of SRUF. The results show that when given an availability requirement, SRUF has greater load balancing performance under uncertain failures and that when given a demand requirement, SRUF can achieve higher availability.

Keywords: Segment Routing, Traffic Engineering, Value-at-Risk (VaR), Robustness, Uncertain Failures.

This work was supported by the National Natural Science Foundation of China under Grant No. 61902346 and 62072402, and the Zhejiang Provincial Natural Science Foundation of China under Grant No. LGN21F020002, and the Hangzhou Science and Technology Development Foundation under Grant No. 20191203B37.

http://doi.org/10.3837/tiis.2021.07.016 **ISSN : 1976-7277**

1. INTRODUCTION

Segment Routing (SR) is a relatively new approach that is proposed by IETF and expected to address identified limitations in Multi-Protocol Label Switching (MPLS) [1]. SR implements the source routing paradigm [2]: When a traffic packet enters the network, the ingress node assigns its forwarding path with a segment list. A segment in a segment list represents a shortest path or link. The packet with the list passes through the segments in the list one by one and finally reaches the egress node. In this way, a path can be divided into several shortest paths. The network operators can use a more flexible path selection scheme so that operators of the network can optimize the network's performance in a relatively simple deployment situation.

A centralized controller such as a Software Defined Network (SDN) controller or a Path Computation Element (PCE) that can compute the scheme of traffic distribution according to the global traffic demands can give full play to the potential of SR and can make the network more efficient [3]. Although the centralized control plane has better overall traffic distribution performance, it takes some time to measure and operate network conditions, especially for unexpected events such as network failures [4]. Considering that SR uses the Interior Gateway Protocol (IGP) protocol to route segments, a distributed IGP failure recovery mechanism can handle network failures [5]. When a node or link fails, the network quickly converges to form a new shortest path between the node pairs. Therefore, the switch can continue to forward according to the original segment list. However, this rerouting method does not consider the congestion of the network when changing the routing path, which may cause network congestion.

One way to solve this problem is to consider the failure of links/nodes in the network when the controller configures the initial flow allocation scheme. When the SDN controller or PCE allocates bandwidth, it considers various situations of network failure and reserves some capacity for network failure scenarios. Therefore, when the predictable network failure occurs, the network will neither be congested. Reserving bandwidth for failures can make the network more robust, but resulting in lower network utilization. A significant challenge faced by traffic engineering is to improve the throughput and availability of the network. In the case of limited bandwidth, these two goals are contradictory because the high availability requires the network to have enough margin to deal with failures and avoid congestion. Still, the high throughput requires the network to fulfill as much traffic demand as possible.

In this paper, we propose a method called SRUF (Segment Routing under Uncertain Failures) that optimizes traffic distribution scheme of SR based on the probability of network failure. This method can make full use of the global network information mastered by the centralized SDN controller, not only the topology, network capacity, and demand pattern, but also the failure statistics of links, nodes and Shared Risk Groups (SRG) [11]. This work uses a finance method called Value-at-Risk (VaR) to extend SR to improve network performance while considering the probability of network failure. Since every path maybe fail, we regard the forwarding of traffic on different paths as an investment with risk. For a given traffic allocation scheme, we compute the possible failure scenarios with the corresponding probability of occurrence and the Maximum Link Utilization (MLU) of the network in the scenario. Using the VaR method, we can obtain the best performance traffic distribution configuration with a given availability requirement $B(90\%, 95\%, 99\% \dots)$ .The main contributions of the paper are the following:

- As far as we know, this work is the first study of risk-aware restoration planning in segment routed networks under uncertain failures. We propose an SR-based algorithm called Segment Routing under the Worst Scenario (SRWS) to minimize the MLU of all given network scenarios with uncertain failures. When optimizing the MLU of the network, we consider all given failure scenarios and minimize the largest MLU in these scenarios. Since we reserve capacity for all given failures, the algorithm is very robust.

- In order to solve the bandwidth wasting problem and the inflexibility caused by the over-protected nature of SRWS, we propose the SRUF method leveraging the financial concept VaR. Compared with SRWS, which protects all scenarios, the SRUF algorithm allows a flexible coverage of failure scenarios. The SRUF method does not consider only one indicator of availability or MLU. Instead, it tries to achieve a tradeoff of them in optimization, so it leads to better traffic distribution with the full use of failure probability information. Besides, through the setting of a network status parameter, SRUF can be applied to more network environments with different availability requirements.
- We, for the first time, define the SR link load factor based on failure scenarios, which incorporates the Equal Cost Multipath (ECMP) routing mechanism. Further, considering that the factor is a key component of SRWS and SRUF, we develop an efficient algorithm using topological sorting for its computation.

The remainder of this paper is organized as follows. Section II discusses some related work, and Section III presents underground about SR, restoration of the network, and Value-at-Risk (VaR). Section IV introduces the proposed methods. In section V, our method is evaluated by experimental results. Section VI draws the main conclusions.

2. RELATED WORK

2.1 SR Optimization

The network optimization problem of SR is to use fewer segments to allocate paths so that the network can achieve the optimization goal. The authors in [16]propose traffic matrix aware SR, traffic matrix oblivious SR, and online SR algorithm to minimize MLU for a given demands matrix and topology. This approach uses two-segment SR, enabling the network to transmit data packets on non-shortest paths and make the network more load-balanced. The authors in [13] propose a mixed integer programming model, including adjacency segment and k -segment and simplify it.

The simplified approach can give full play to the advantages of the adjacency segment to solve the problems that cannot be solved by node segment and arbitrarily specify constant K. Because the optimization of SR only considers the overall condition of the network, it may cause the delay of the flow to be very long, which the user cannot accept.

In [12], the authors introduce a bounded stretch constraint to avoid too long paths. This approach limiting the shortest-path distance between intermediate nodes can improve user experience. More segments will result in greater overhead. The authors in [7] propose an efficient segment list encoding algorithm to guarantee optimal path computation and limit segment list depth. The algorithm can reduce the overhead of the network in different network scenarios. In [14], the authors propose CG4SR algorithm, which leverages column generation, a widely used technique for solving large-scale linear programs, combined with a novel dynamic program to achieve better scalability and reach near-optimal solutions with a gap guarantee. In [27], the authors propose a semi-oblivious SR algorithm that takes bounded traffic fluctuations based on an initially estimated traffic matrix. This makes the approach robust to traffic fluctuations and has a better performance compared to oblivious routing techniques. More SR applications, e.g., SRv6, multicast [26], and IoT [39], and related overview can be found in [10], [19] and [20].

2.2 Network Restoration

SR is based on the existing IGP protocol, and all can use the fast reroute mechanism. When there is a link or node failure, IGP recomputes all the shortest paths, and the SR is automatically repaired without interruption. In [5], the authors propose a method to optimize the centralized determination of connections primary paths to enable the best sharing of restoration bandwidth over non-simultaneous failures. In [3], the authors propose a robust semi-oblivious method to meet the flow demands and ensure a good network performance after link failures. In [4], the authors propose an SR method to construct pairs path to remain disjoint even after an input set of failures to be used for restoration. In [6], the authors initiate the systematic study of such local fast failover mechanisms which not only provide connectivity guarantees, even under multiple link failures but also account for the quality of the resulting failover routes for locality and congestion and propose a method called CASA providing a high degree of robustness as well as a provable quality of fast rerouting.

3. BACKGROUND

3.1 Segment Routing

SR is a source routing architecture. The source node selects paths and guides the data packet to transmit along the paths in the network by inserting a sequenced segment list in the header of the data packet [9]. On the forwarding path, the network device that receives the data packet performs processing and forwarding operations according to this segment list [15]. Other nodes except the source node do not need to store and maintain any state information of the flow, so compared to the SDN architecture, SR saves the storage hardware of the forwarding table (such as ternary content addressable memory, TCAM) and thus has better scalability [22]. SR can provide advanced traffic steering capabilities in the IP/MPLS network while maintaining scalability in the data plane and control plane.

A segment is an instruction to be executed by the node on the received data packet. The instruction includes: forwarding the data packet according to the destination node of the shortest path, sending the data packet through the designated interface and forwarding the data packet to the designated application or service instance. As is shown in **Fig. 2**, in addition to executing the instructions encoded in the segment list, the node also maintains the list itself. For this purpose, three basic operations on the segmented list are defined [21]:

- PUSH: If the header of the data packet does not have a segment list, insert the segment list, otherwise insert one or more segments in the header of the segment list, then set the first segment as the active segment.
- NEXT: The active segment has been completed; set the next segment as the active segment.
- CONTINUE: The active segment has not been completed, continue the current forwarding instruction.

There are two basic types of segments: node and adjacency. A segment is identified with a Segment Identifier (SID) which is divided into two categories [17]. An adjacency segment represents a local interface of a node, and adjacency segment IDs are typically only locally significant on each node. A node segment identifies a router node, and node segment IDs are globally unique across the domain [28]. **Fig. 2** shows a network with bidirectional links. The number next to each link is its IGP link weight. We will use an example in **Fig. 2** to illustrate the forwarding process of SR. Considering that the controller has a full understanding of the global network status. When a traffic demand from R1 to R10 reaches the network, the data plane will forward it according to the forwarding strategy set in advance by the controller. Assuming that path R1-R4-R5-R7-R8-R10 is the optimal path in the current network state, the corresponding segment list is shown in the figure.

When the traffic packet reaches R1, R1 uses the PUSH operation on the data packet, add the segment list (R4, R7, R8, R8-R10) to the header of the data packet, set the active pointer to point to R4, and send the packet to R4. Then R4 that received the packet uses the NEXT operation to make the active pointer point to R4 according to the segment list and transmit the traffic packet to R7 along the shortest path. When reaching R5, the traffic packet is executed CONTINUE operation and continues to be forwarded to R7. The traffic packet is forwarded to the destination in this way.

Generally speaking, more segments mean that the controller has stronger control over the network and achieves better performance. But for our problem, the two segmented routings (2-SR) with node segments and adjacency segments is enough and can be easily extended.

3.2 Network Restoration

As the scale of the network continues to increase, the number of network devices in the network is also increasing, and the failure of links and nodes in the network can not be avoided. In a network system, the probability of a failure in an hour is 50%, and the probability of a failure in a day is 70% [8]. In the SDN-based data center wide area network, the link utilization rate is generally close to 100%, so even if a single link fails, the impact is very serious. In the existing work, failure recovery methods are mainly divided into two categories: One is passive recovery, and the other is active recovery. The main difference between the two is whether the failure handling occurs before or after the failure occurs.

In the data plane of explicit path configuration, the passive failure recovery method is traffic redirection. The ingress node sets the weight of the failure-related path to 0, and the remaining unaffected path weights are re-divided. Traffic is quickly redirected, and network connections are quickly restored. The failure recovery mechanism of SR that makes full use of the mature IGP protocol failure recovery mechanism is relatively more advantageous. **Fig. 2** is an example of a failure recovery for SR. In the data plane of SR, segment forwarding is based on the IGP protocol. When a failure occurs in the network, the IGP protocol will automatically recalculate the shortest path between nodes, the original segment list can still be used, and the network can continue to be used uninterrupted [18]. However, the passive recovery method is determined locally by the data plane, which will cause network congestion [25].

A better passive recovery method is to request control plane intervention. The controller formulates new routing rules according to the new topology and updates each switching device. At the same time, it is obvious that this method is very time-consuming and only used in severe failure scenarios. In order to overcome the shortcomings of passive recovery, some active failure recovery method was proposed [5]. The idea is to consider possible link/node failures in advance when computing traffic distribution to ensure that the network will still not be congested when a single link/node fails.

3.2 VaR and CVaR

To better explain our method, we introduce the concept of Value-at-Risk [31], which is one of the most well-known risk measures used in robust optimization under uncertainty. Given a confidence coefficient, VaR provides an upper bound or quantile based on the probability distribution of risk. For example, in an investment that requires a confidence level of $B \in (0,1]$, The corresponding B-VaR is the minimum value β that can meet the confidence level B; the loss will not exceed β [32]. In short, VaR in economic investment is the maximum possible loss under a certain confidence coefficient. VaR is the most widely used risk management indicator because of its simple concept and easy processing into mathematical models [33]. However, VaR also has some shortcomings; adding var constraints to a solvable polynomial problem will increase discrete variables and turn the problem into an NP-hard problem. Besides, VaR is not a so-called coherent risk measure, implying among another thing that is non-convex and not sub-additive [33].

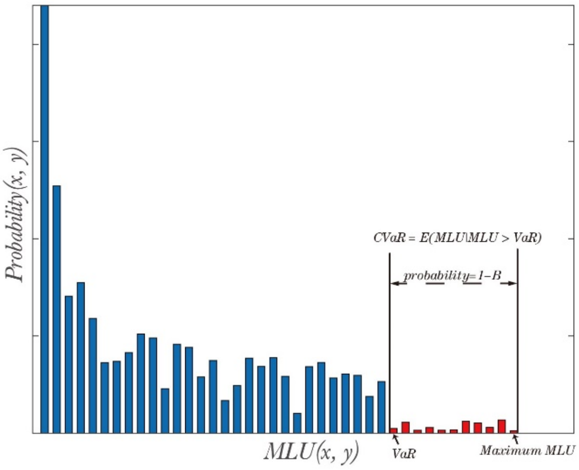

Fig. 1. Illustration of VaR and CVaR. Given a flow distribution x, then $\mathrm{MLU}(x, y)$ varies with network state y. The MLU values are sorted in ascending order. Given a probability threshold B, then the B quantile of MLU is the VaR value. The red part represents the MLU values greater than VaR, and its conditional expectation is CVaR.

In order to tackle these shortcomings of VaR, another risk measure closely related to VaR is introduced, which is the so-called Conditional Value-at-Risk (CVaR). By definition, CVaR is the conditional expectation of the loss under the condition that VaR is exceeded. Several important results regarding optimization of CVaR are proved by [35], which make this risk measure attractive from the optimization viewpoint. CVaR makes up for several deficiencies in VaR. It is coherent, which makes CVaR is easy to handle in optimization models. We can intuitively understand the relationship between CVaR and VaR from **Fig. 1**.

4 THE PROPOSED METHODS

4.1 System Model

We now describe our optimization framework in detail. The key notations in our system model formulation can be found in **Table 1**. The network is modeled as a directed graph $\mathrm{G} = (V, E)$, where the vertex set V represents switches/routers and edge set E represents links between switches/routers. Link capacities are given by $c = \big(c(1), \ldots, c(e)\big)$, and as in any Segment Routing Traffic engineering (SR-TE) formulation, the total flow on each link should not exceed its capacity. The traffic distribution of SR should meet the needs of the demand matrix D. SR-TE decisions are made at fixed time intervals, based on the estimated user traffic demands for that interval. In each time epoch, there is a set of source-destination switch pairs, where each such pair r is associated with a demand t_r and a fixed set of tunnels $tunnel(r,k)$ and $tunnel(r,m)$ on which its traffic should be routed. $tunnel(r,k)$ is a set of all routing paths using at most two segments that can transmit flow r. The amount of traffic on $tunnel(r,k)$ is represented by x_r^k. $tunnel(r,m)$ is a set of all routing paths using at most two segments with an adjacency edge m which must start from the source node of flow r and end at the destination node of r. The amount of traffic on $tunnel(r,m)$ is represented by x_r^m.

Table 1. Notations

Symbols	Description
$G(V,E)$	Network Graph with nodes V and links E.
$c(e)$	The capacity of link $e \in E$.
r	The flow r between two nodes.
r_s	The source node of flow r.
r_t	The end node of flow r.
k	The intermediate node k of SR.
m	The intermediate adjacency link m of SR.
$tunnel(r,k)$	The SR tunnel through intermediate node k for flow r.
$tunnel(r,m)$	The SR tunnel through adjacency link m for flow r.
x_r^k	The amount of forwarded traffic on $tunnel(r,k)$.
x_r^m	The amount of forwarded traffic on $tunnel(r,m)$.
$t_r \in D$	The bandwidth demand of flow r.
α	The Value-at-Risk.
B	The target availability level.
$q \in Q$	The network state corresponding to a scenario.
θ_q	The total loss in scenario q.
P_q	Probability of network state q.
$g_r^k(e,q)$	The fraction of flow r passing on link e through the $tunnel(r,k)$ in scenario q.
$g_r^m(e,q)$	The fraction of flow r passing on link e through the $tunnel(r,m)$ in scenario q.
θ_q	The maximum link utilization in scenario q.
$\widehat{\theta_q}$	Auxiliary variable used to calculate VaR.
θ	The maximum value of θ_q in all scenarios.

Understanding the concept of scenario q is key to understanding our methods. We use a general network failure model to represent each scenario in the network. A vector $q = (q_1, q_2, \ldots, q_n) \in Q$ is used to represent a scenario, where q_i is a binary random variable, indicating whether failure event i occurred or not. A failure event i represent a single SRG becoming unavailable. An SRG represents a group of logical nodes or links that will fail at the same time because of the shared infrastructure. In our method, these SRG failure events are uncorrelated. Generally speaking, a failure event i can represent a link failure, a failure of a node connected to multiple links, or failure of all links in an area. For example, for a graph with 15 edges (we assume that each edge is an SRG), we can use a vector with 15 components $q = (0,0 \ldots,0)$ to represent a network scenario where no failure occurs on all network links. Because all failure events are not related, we can use the following formula to calculate the probability of a scenario q:

$$P_q = P(q_1, \ldots, q_n) = \prod_{i=1}^{n} [p_i q_i + (1 - p_i)(1 - q_i)], \tag{1}$$

where q_i is a binary variable and p_i is the probability of occurrence of the failure event i.

In this section, we first formalize the model of SR under the worst scenario (SRWS). Based on the shortcomings of the SRWS method, we introduce the VaR method to improve the algorithm and propose SRUF method, which can make full use of probabilistic information and provide a probability guarantee for the network. Then we propose an algorithm based on variant shortest path algorithm and topological sorting algorithm to solve the important input of SRUF and SRWS methods. Finally, we introduce the changes of SID in different failure scenarios and our scenario setup.

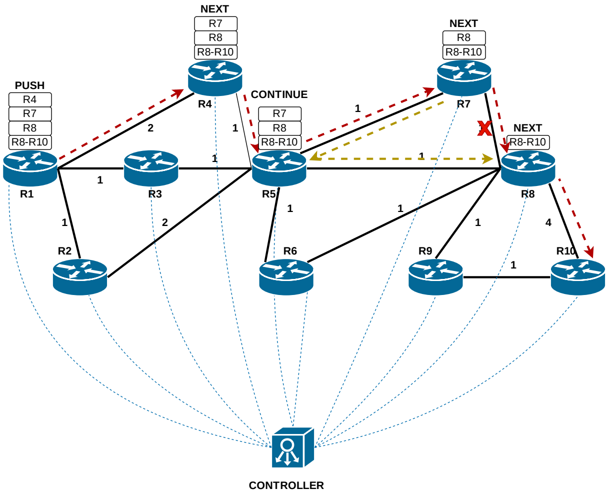

Fig. 2. Illustration of SR under uncertain failures.

4.2 Segment Routing under the Worst Scenarios

Considering that a network's failure may cause congestion on one or more links, we propose the SRWS method to avoid this problem. The core idea of SRWS is to minimize the maximum MLU in all given scenarios. SRWS ensures that no congestion occurs in given failure scenarios and can be expressed as the following linear program:

$$minimize \quad \theta$$

$$\sum_{k\in V} x_r^k + \sum_{m\in E} x_r^m \geq t_r,\ \forall r \tag{2}$$

$$\sum_r \sum_{k\in V} x_r^k g_r^k(e,q) + \sum_r \sum_{m\in E} x_r^m g_r^m(e,q) \leq c(e)\theta_q,\ \forall e,q \tag{3}$$

$$\theta_q \leq \theta,\ \forall q \tag{4}$$

$$x_r^k \geq 0, \forall r,k,\ x_r^m \geq 0, \forall r,m \tag{5}$$

Equation (2) ensures that the distribution of traffic by the SDN controller can meet all traffic demands in the network. Because our method uses a 2-SR, the task of our method is to select a suitable intermediate node or adjacency link for traffic forwarding. If one of the endpoints of flow r and intermediate node k is the same, the $tunnel(r,k)$ will use the shortest path of flow r for forwarding. For $tunnel(r,m)$, there are three situations. First, when the source node of r and the source node of m are the same node and the destination node of r and the destination node m are different, the tunnel first passes through the link m and then goes to the destination of r through the shortest paths to the destination, second, when the source node of r and the source node of m are the different node and the destination node of r and the destination node m are the same, the tunnel first passes through the shortest paths from its source to the source node of m and then goes to the destination through m, and finally, when the source node of r and m is the same, and so as the destination node of r and m, $tunnel(r, m)$ represent the link m.

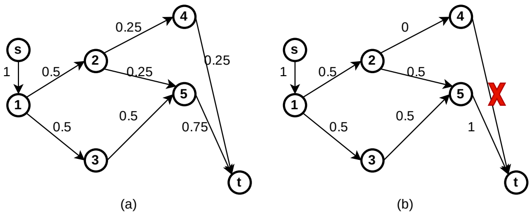

Fig. 3. Illustration of ECMP and corresponding failure recovery. (a) Fractional values of $g_r(e, q)$ with ECMP. All the links along the shortest paths between s and t are shown. The values next to the links represent the fraction of the flow on them. (b) When a failure occurs, the network status changes and $g_r(e, q)$ should be updated to restore network forwarding.

Equation (3) calculates the θ_q representing the MLU in the scenario q, and (4) calculates the MLU in the network in all scenarios. Since we need to avoid congestion in all given events of a network failure, the link utilization of all links in a scenario q needs to be computed. We have the following formula for this:

$$util(e, q) = \frac{\sum_r \sum_k x_r^k g_r^k(e,q) + \sum_r \sum_m x_r^m g_r^m(e,q)}{cap(e)} \tag{6}$$

$util(e, q)$ represents the link utilization rate caused by a specified traffic distribution scheme $\{x_r^k, x_r^m\}$ in a certain scenario q. Given the link weights in the network, one or more (in this case, we use Equal Cost Multipath (ECMP) to forward packets) shortest paths between any two nodes in the network are fixed. As is shown in **Fig. 2**, when a failure occurs in the network, the failed nodes and links will be disconnected from the network, the network will recompute the shortest path between the nodes, and the distribution of network flow will also change the amount of network flow imposed on each link. So, each scenario q corresponds to certain network topology and a set of determined shortest paths between nodes. We use $g_r(e, q)$ to represent the fraction of traffic generated on e that flows r under network state q. When ECMP is not considered, flow r only uses one shortest path, then the value of $g_r(e, q)$ of the link on the shortest path is 1, and the other links not on the shortest path are 0. When ECMP is considered since flow r is split and forwarded by multiple paths, the value of $g_r(e, q)$ can be fractional. As shown in **Fig. 3**, each shortest path value forwards part of the flow under the ECMP routing, and the situation that a huge flow has too much influence on a certain link can be reduced, which is more conducive to load balancing. When ECMP is considered, there are two types of rerouting situations when a failure occurs in the network. As shown in Figure **Fig. 3** (b), when only part of the path in ECMP has errors, the network will use the remaining paths to redistribute traffic, and when all paths in ECMP fail, the new shortest path will be automatically computed to complete the forwarding task. When we combine ECMP and 2-SR with intermediate k, the fraction on link e will be the sum of the flows from the source node of r to k and k to the destination node of r. We assume that the former is flow r_1 and the latter is flow r_2; then we define this fraction:

$$g_r^k(e,q) = g_{r_1}(e,q) + g_{r_2}(e,q) \tag{7}$$

When we combine ECMP and SR with adjacency segment m, since $tunnel(r,m)$ is made up of link m and the shortest path on which the flow is assumed to be r', then we define this fraction as:

$$g_r^m(e,q) = \begin{cases} g_{r'}(e,q) + 1, e = m \\ g_{r'}(e,q), e \neq m \end{cases} \tag{8}$$

Therefore, $g_r^k(e,q)$ is the flow that results on link e if a unit flow r is routed through $tunnel(r,k)$ in scenarios q, and $g_r^m(e,q)$ is the flow that results in one link e if a unit flow r is routed through $tunnel(r,m)$ in scenarios q.

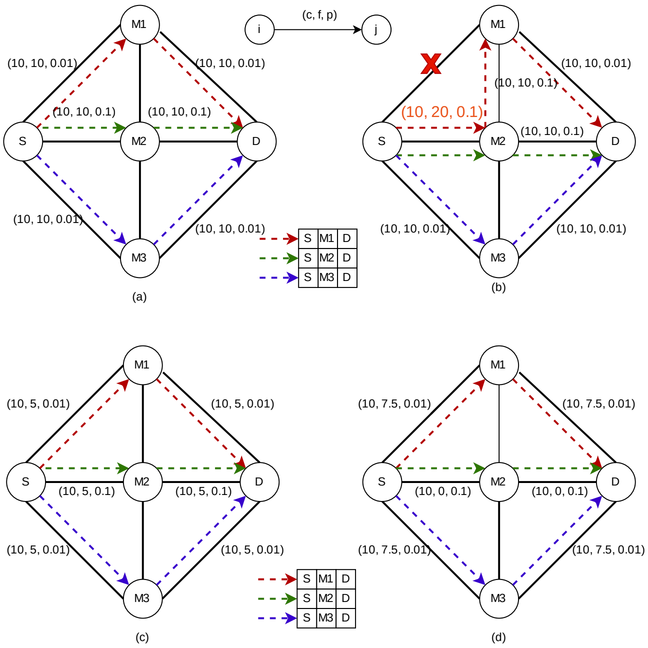

Fig. 4. Traffic configuration with different algorithms. The triple (c,f,p) represents the link capacity, the link load and the failure probability of the link, respectively. (a) Traffic configuration with general SR. (b) General SR suffers from link failures. (c) Traffic configuration with SRWS. (d) Traffic configuration with SRUF.

The idea of SRWS is to compute a traffic distribution scheme that ensures that the network will not be congested in any given scenario. This method is very effective in a network environment that cannot tolerate congestion. As is shown in **Fig. 4**, the general segment routing method only considers low MLU or high throughput when distributing traffic, without considering the occurrence of failure. But when a failure event occurs (just like **Fig. 4** (b)), the network may be congested. **Fig. 4** shows the SRWS scheme which can ensure that the network will not be congested under any single link failure scenario.

4.3 Segment Routing with Value-at-Risk

The SRWS method ensures that the SR traffic distribution scheme will not cause network congestion under a given series of failure scenarios. However, it has two shortcomings: (1)

The network must reserve a part of bandwidth resources to prevent the occurrence of failures, even in most network environments, the occurrence of failures is a small probability event [37], especially the worst scenario. Over-provisioning only results in low network performance. (2) Ignoring information about the probability of failure in the network will result in suboptimal decision making. The centralized controller can fully collect various information in the network and make full use of the failure history data of the network to make better traffic configuration. In order to overcome these difficulties, we propose our segmented routing with VaR method called SRUF. The mathematical description of the method is as follows:

$$minimize\ \alpha + \frac{1}{1-B}\sum_{q\in Q} P_q\widehat{\theta_q} \tag{9}$$

$$\sum_{k\in V} x_r^k + \sum_{m\in E} x_r^m \geq t_r,\ \forall r \tag{10}$$

$$util(e,q) - \alpha \leq \widehat{\theta_q}, \forall e, q \tag{11}$$

$$\widehat{\theta_q} \geq 0,\ \forall q \tag{12}$$

$$x_r^k \geq 0, \forall r, k,\ x_r^m \geq 0, \forall r, m \tag{13}$$

The objective function represents CVaR of the MLU in all scenarios. In order to better understand our objective function, we introduce the mathematical expressions of CVaR and VaR of MLU. Given a random variable $MLU(x,y)$ representing the MLU when given a decision vector x and a random vector y representing uncertain parameters that may affect the performance of the network system under consideration, then the $MLU(x,y)$ for each x is a random variable having a distribution induced by that of y. Therefore, the probability that $MLU(x,y)$ does not exceed some value α is defined as

$$\psi(x,\alpha)\ \coloneqq\ P\{y|MLU(x,y) \leq \alpha\} \tag{14}$$

When the value of x is given, the cumulative distribution function of the loss function associated with the decision x is represented by $\psi(x,\alpha)$ [33]. Then, B-VaR can be defined as

$$\alpha_B(x) \coloneqq min\{\alpha \in R: \psi(x,\alpha) \geq B\} \tag{15}$$

From this, we can see that the probability that $MLU(x,y)$ exceeds $\alpha_B(x)$ is $1-B$. Due to the above concept, CVaR is the conditional expectation that the loss according to the decision vector x dominates $\alpha_B(x)$ [36]. Thus, we have B-CVaR denoted as $\phi_B(x)$ defined as

$$\phi_B(x) \coloneqq E\{MLU(x,y)|MLU(x,y) \geq \alpha_B(x)\} \tag{16}$$

To include both CVaR and VaR in the same optimization mathematical model, we characterize $\alpha_B(x)$ and $\phi_\alpha(x)$ in terms of a function F_B defined by

$$F_B(x,\alpha) \coloneqq \alpha + \frac{1}{1-B}E\{max\{MLU(x,y) - \alpha, 0\}\} \tag{17}$$

$$= \alpha + \frac{1}{1-B}\sum_q p_q max\{MLU(x,y) - \alpha, 0\}$$

It can be shown that as a function of α, $F_B(x,\ \alpha)$ is continuously differentiable and convex

[36]. For any x, $\phi_B(x) = min_\alpha F_{B(x,\alpha)}$. Furthermore, if $A_B(x) \coloneqq argmin_\alpha F_B(x, \alpha)$ is the set consisting of the values α for which F is minimized, the $A_B(x)$ is a nonempty, closed and bounded interval, and $\alpha_B(x)$ is the left endpoint of $A_B(x)$. In fact, it is always the case that $\alpha_B \in argmin_\alpha F_B(x, \alpha)$ and $\psi_B(x) = F_B\big(x, \alpha_B(x)\big)$ [33]. It has also been shown that for any probability threshold B, if (x', α') minimize F_B, then not only does x' minimize ϕ_B, but also $\phi_B(x', \alpha') = F_B(x', \alpha')$ and $\alpha_B(x') \doteq \alpha$ [36]. After the optimization of CVaR, we can get the exact solution of VaR by a simple method.

Using (17), we can process the non-smooth function which represents conditional VaR of MLU, into a linear function and obtain the VaR and CVaR corresponding to the optimization result through mathematical methods. As is shown above, (9) minimizes the CVaR of MLU under the given probability guarantee B. The auxiliary variable $\widehat{\theta_q}$ in (11) is used to process the max function in (17) into a linear function form. The optimization formula of our method is to solve the segmented routing allocation plan with the smallest condition value of the risk of MLU that meets the flow demand under the given probability guarantee. This problem is treated as a linear programming problem and there are $O(|V|^3 + |V|^2 * |E| + |Q|)$ variables and $O(|V|^2 + |Q||E|)$ constraints. We can directly solve this linear programming problem.

SRUF uses the VaR method in finance to ensure that the network will not be congested within $B(90\%, 95\%, 99\% \dots)$ of the time, and the parameter B is set by the control plane of the network according to the network environment. The use of parameter B is of great significance. On the one hand, the probability of certain errors in the network is extremely low. By setting the parameter B to ignore some scenarios with great influence and low probability of occurrence can make full use of network resources. On the other hand, The setting of parameter B allows the SRUF algorithm to adapt to the availability requirements of various network environments. Therefore, SRUF is a more general method; when the probability threshold parameter B is large enough, we almost require that the network can still work without congestion in any failure situation, and when B is small and equal to the probability of normal state without any failure, we only require the network can work without congestion when no abnormality occurs what is similar to the general SR algorithm. It must be mentioned that SRUF tends to select the paths with a lower probability of failure so it can achieve better robustness with small parameters B, which is equal to the probability of a normal state without any failure. As is shown in **Fig. 1**, SRUF not only minimizes VaR of MLU but also minimizes CVaR, which makes the conditional expectation of the MLU exceeding the VaR lowest. Our SRUF method with minimizing CVaR can effectively prevent tail risks and improve network availability.

The risk-knowledgeable SRUF method that can set parameter B according to availability requirements and effectively prevent tail risks benefits from the full use of the failure probability information of links or nodes in the network, which can be easily collected and processing by a centralized control plane. The network using SRUF forward flows in a lower-risk manner. Take **Fig. 4** as an example; when there are 15 units of traffic that need to be forwarded from node S to node D, SRWS will consider all single link failures and transmit from the three paths shown in **Fig. 4** (c). In this case, although the network is safe enough under a single link failure, when the demand from S to D increases, the network will no longer have a margin to complete the transmission. In addition, because the path *S-M2-D* has a higher failure rate, the SRWS transmission scheme will have a higher risk, causing the network to become more likely a state of high MLU. **Fig. 4** (d) is a traffic distribution scheme that considers the risk of failure. We choose two safer paths to guarantee that the maximum utilization rate of all links is 75% at least 96% of the time. SRUF uses a more secure forwarding scheme and uses the reliability of some extreme failure scenarios in exchange for higher network utilization.

4.4 SR Link Load Factor Calculation Based on Failure Scenarios

As mentioned above, $g_r^k(e,q)$ and $g_r^k(e,m)$ are key inputs for SRUF and SRWS; before we use our method to get a solution, we should first use (7) and (8) to get the value of them. Because $g_r^k(e,q)$ and $g_r^m(e,q)$ are both calculated by $g_r(e,q)$, we propose an algorithm based on a variant of the shortest path algorithm and topological sorting [24]to computing $g_r(E,q)$, which represents a set of $g_r(e,q)$ for all $e \in E$. The algorithm is described as Algorithm 1.

```
Algorithm 1: Computation of $g_r(E,q)$
Input: Graph $G = (V,E)$, flow $r$, scenario $q$
Output: $g_r(E,q)$: an array of $|E|$ elements
$\overline{G} \leftarrow error\_link(q)$
$\dot{G}(\dot{V},\dot{E}) \leftarrow G - \overline{G}$
$G_s(V_s,E_s) \leftarrow findshortestpath(\dot{G},r_s,r_t)$
$nf[i] \leftarrow 0,\ \forall i \in V_s,\ i \neq r_s$
$nf[r_s] \leftarrow 1$
$lf[e] \leftarrow 0,\ \forall e \in E_s$
$Queue\ Q\ \leftarrow \emptyset$
$indegree[i] \leftarrow in\_degree(i)$
$outdegree[i] \leftarrow out\_degree(i)$
for all $i = 1$ to $|V|$ do
    if $indegree[i] == 0$ then
        $Q.push(i)$
    end if
end for
repeat
    $x \leftarrow Q.pop()$
    for all $e \in E_s$ do
        if $from\_node(e) == x$ then
            $lf(e) = nf[x]/outdegree[x]$
            $y \leftarrow end\_note(e)$
            $nf[y] = nf[y] + nf[x]/outdegree[x]$
            $indegree[y] = indegree[y] - 1$
            if $indegree[y] == 0$ then
                $Q.push(y)$
            end if
        end if
    end for
until $Q\ ==\ \emptyset$
$greq[e] \leftarrow 0,\ \forall e \in E$
$greq[e] \leftarrow lf[e],\ \forall e \in E_s$
return $greq[1 \ldots |E|]$
```

The inputs of Algorithm 1 are graph G, flow r, and scenario q. We define the link load factor as the amount of traffic applied on each link when one unit of flow is transmitted. The main purpose of the algorithm is to calculate the link load factor of all edges in the graph G under scenario q. The algorithm can be divided into four stages. The first stage is scenario mapping, which removes all edges and nodes that have failed and generates an auxiliary graph $\dot{G}$, the second stage is the calculation of all the shortest paths of flow r in the graph $\dot{G}$ and generates the shortest path subgraph G_s, the third stage is to calculate the link load factor based on topological sorting algorithm [24] on G_s, and the fourth stage is the edge mapping between G_s and G. $nf[i]$ denotes the amount of traffic applied on node i when one unit of flow is transmitted. Similarly, $lf[i]$ represents the amount of traffic applied on link i when one unit of flow is transmitted. The output of the algorithm is $g_r(E, q)$, which means that running the algorithm once can get the load factor of all edges in the graph G.

4.5 Different Uncertain Failure Scenarios

Although our method can deal with all failure scenarios, the computational complexity will increase sharply as more failure scenarios increase. Consider one edge as an SRG. The complexity of the failure scenario is exponential. When the network topology has 20 links, there are 1048579 scenarios. When there are 30 links in the network, the number of scenarios is 1073741824. This complexity is unacceptable for us, and the scalability is very poor. Pruning algorithm, sampling algorithm, and appropriate scenario settings can be designed to solve this problem. There is a large body of research in restoration in networks. Single link failures, single node failures, and Shared Risk Link Group (SRLG) failures are typical scenarios in the research of failure recovery of network. An SRLG represents multiple link sharing infrastructure can fail together and include single link failures and single node failures. In this paper, we only consider the SRLG scenarios that include all single link failures and single node failures. here are two reasons for this. First, most of the network is operating in a failure-free or single-link failure state. For example, according to the data of the MWAN in [37], the failure-free and single-link failure state can occupy more than 99.99% of the network time. Second, when there are too many node and link errors in the network, the controller will repair the network and recalculate the distribution of traffic due to many requirements that cannot be fulfilled. Based on the above modeling of the failure scenarios, a vector $q = (q_1, q_2 \dots, q_n) \in Q$ will be used to represent a failure scenario, where the value of n is $|V| + |E|$, and q has only one component with a value of 1 (others are 0), which means that only one SRLG failure event will occur at the same time.

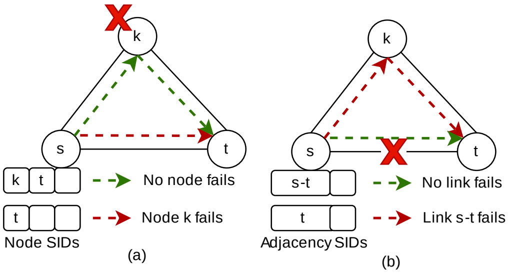

Fig. 5. Changes in SIDs in failure recovery. (a) When the intermediate node k fails, node SID k in the node SIDs list will be extracted. (b) When the link s-t fails, adjacency SID s-t in the adjacency SIDs list will be converted to node SID.

There are two more situations about failure scenarios to pay attention to in our SRUF framework (**Fig. 5**). The first situation is that a failure occurs in the intermediate node of the

SR. In this situation, the SIDs list needs to be changed, the intermediate node needs to be deleted from the SIDs list, and the network uses the shortest path from source node s to destination node t to forward the packet. The other situation is when using the adjacency SIDs to specify the link to transmit data and encountering a failure one the link, the switch has to change the adjacency SID to the node SID of the end node of the link, and use the shortest path for transmission.

5 EXPERIMENTAL RESULTS

In this section, we first describe the network setting, data setting, and the methods for comparison in our experiments, and then we evaluate the proposed algorithms SRUF and SRWS in detail.

5.1 Experimental Setting

We evaluate our method on network topology B4 [30], which is widely used in computer network traffic optimization with 12 nodes and 38 edges, see **Fig. 6**.

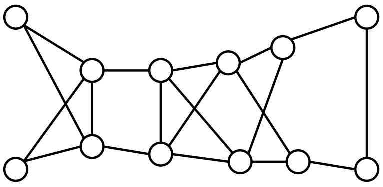

Fig. 6. The B4 topology.

We perform experiments with all links having the equal capacity as well as cases where the link capacities have uniform distribution $U[10: 100]$. We use a uniform distribution $U[1: 5]$ to set link weights to take advantage of ECMP and adjacency segment. The combination of adjacency segment and ECMP not only makes full use of the load balancing effect of ECMP, but also specifies traffic allocation for a single path of ECMP. The adjacency segment can make full use of the non-shortest links in the weight graph. The Weibull distribution which has been used in a prior study of failures in backbones [34] is used here to model failure scenarios with probability. We denote the Weibull distribution with shape parameter λ and scale parameter f by $W(\lambda, f)$. Throughout our experiments, we change the shape and scale parameters of our Weibull distribution and study the impact of the probability distribution on performance. During the experiment, we sampled the edge failure rate in the network multiple times, used the algorithm to iteratively calculate multiple times, and averaged the results. Our optimization framework uses the Gurobi LP solver and is implemented using the Julia optimization language [38].

Here, we compare the following algorithms.

- SRUF: Our SR algorithm based on CVaR. The setting of parameter B allows the SRUF algorithm to adapt to the availability requirements of various network environments.
- SRWS: Our SR algorithm considering the worst-case network performance. SRWS minimizes the worst-case MLU in consideration of all given failure conditions.
- SPR: A shortest path routing algorithm with restoration. In the flow distribution stage, the SPR algorithm uses the shortest path to transmit traffic. When the network encounters a failure, the SPR algorithm will automatically recalculate the shortest path between nodes to restore the traffic transmission.

- ECMPR: An equal cost multipath algorithm with restoration. In the flow distribution stage, the ECMPR algorithm uses multiple paths of equal cost to transmit traffic. When the network encounters a failure, the ECMPR algorithm will automatically recalculate multiple paths of equal cost between nodes to restore the traffic transmission.

5.2 Comparison of MLU

MLU is the utilization rate of the link with the largest link utilization rate in the network and a common optimization target in traffic optimization, which can improve the load balance of the network and increase the throughput of the network. Since our work is based on the network under uncertain failures, we will consider different failure scenarios and their probability of occurrence during optimization. Therefore, we calculate MLU in two ways. One is to consider the maximum value of MLU in all scenarios, which reflects the robustness of the algorithms to some extent. The other is to compute a weighted average of the MLU in various scenarios, and this method considers the probability of occurrence of each scenario.

Fig. 7 compares the MLU in the worst scenario. The x-axis represents different IDs of demand matrixes whose distribution is different. The results show that MLU of the worst scenario generated by SRWS is the smallest, which validates the load balancing effect of the SRWS algorithm in the worst scenario. The MLU produced by the SRUF algorithm with a small parameter B in the worst case is greater than that produced by SRWS but less than that produced by ECMP and SPR. As long as the value of B is large enough, SRUF can achieve the same effect as SRWS in the worst scenarios. Even if the value of B is not that large, SRUF optimizing the CVaR can do more to prevent congestion than SPR and ECMPR methods in the worst scenarios.

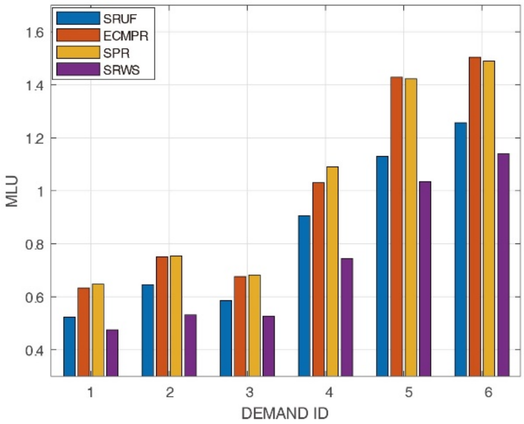

Fig. 7. Comparison of MLU in the worst scenario.

Fig. 8 compares the weighted average of MLU. We can see that the solution of the SRUF algorithm makes the network have a lower MLU, which enables the network to have better performance of load balance and greater throughput. At the same time, it is also verified that the ECMPR algorithm has better performance than the SPR algorithm in terms of load balancing. SRWS only considers the worst-case performance and therefore over-protects the network, causing performance degradation under normal conditions.

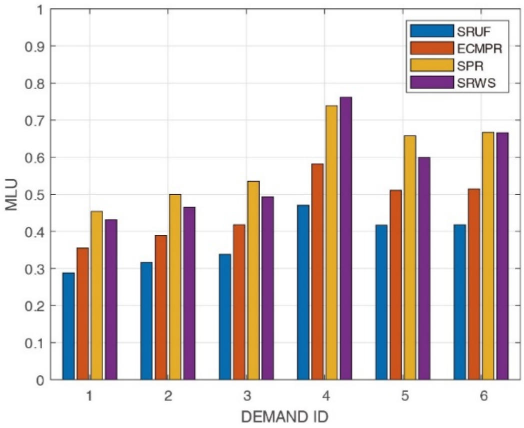

Fig. 8. Comparison of the weighted average of MLU.

5.2 Comparison of Availability

The definition of availability is the probability that a traffic distribution scheme can meet all demands. In the process of calculating availability, we first distribute traffic in different scenarios according to the allocation of each algorithm and calculate the loss representing the sum of traffic that cannot be satisfied with all flows. Then the loss in this scenario can be obtained by adding the loss of all flows. Obviously, when the demand is small, the availability of all our algorithms is 100%. We use the method of scaling the demand to continuously expand the demand to observe the changes in the availability of various algorithms in the process. **Fig. 9** shows the results. In the process of demand growth, SRUF supports higher demands for a given availability. When the availability is greater than 99%, the scaling factor of SRUF is 2.8, the scaling factor of ECMPR is 2.4, the scaling factor of SPR is 1.8, and the scaling factor of SRWS is 1.6. In the process of declining availability, although several algorithms have similar declining speeds, SRUF can support larger-scale requirements under the same availability.

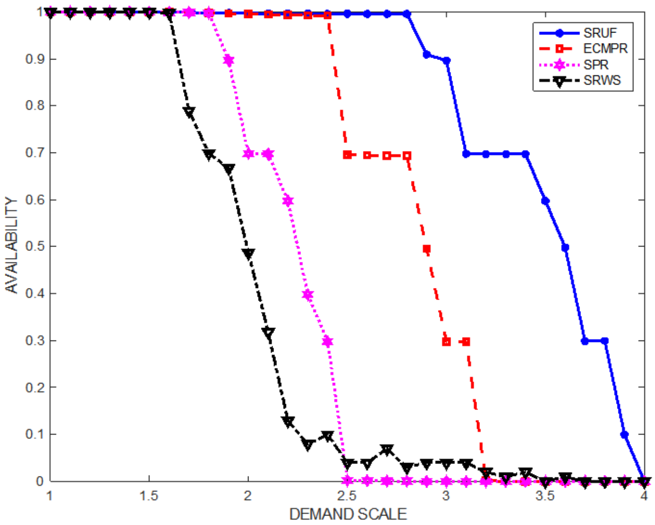

Fig. 9. Comparison of availability.

Fig. 10 shows the scaling factor of demand under higher availability. It can be seen that under the same availability level, SRUF can support more demands, and when the availability requirements increase, the total amount of demand that each algorithm can meet has been decreasing.

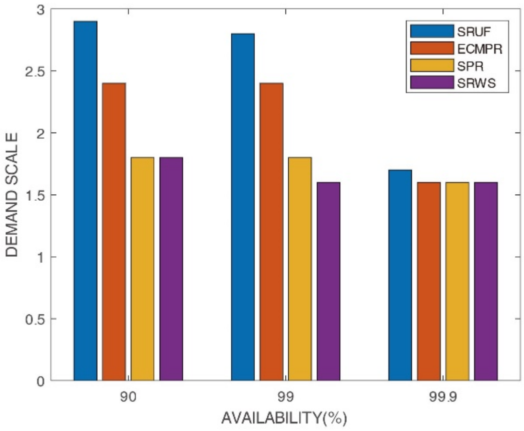

Fig. 10. Comparison of the demand scales under high availability.

6 Conclusions

Taking into account the uncertain failures of nodes and links in the network, we introduce the failure probability of the network into the calculation of traffic distribution for SR. We use the MLU of the network as the objective function and employ the VaR concept in finance to optimize the objective function. We proposed the SRUF method, which takes into account the failure probability and the process of rerouting after a failure during optimization. In SRUF, the controller can set the availability requirements of the network according to the network environment, and then minimizes the MLU of the network under the required requirements to achieve load balancing for each demand and increases the throughput upper bound of the network system. We have conducted experiments to evaluate our method. The results show that our method is able to make full use of probability information to achieve a smaller MLU, get higher availability, and support more traffic demands.

References

[1] L. Davoli, L. Veltri, P. L. Ventre, et al., "Traffic engineering with segment routing: SDN-based architectural design and open source implementation," in *Proc. of 2015 Fourth European Workshop on Software Defined Networks*, pp. 111–112, 2015. Article (CrossRef Link)

[2] Y. Wang, X, Zhang, L. Fan, et al., "Segment Routing Optimization for VNF Chaining," in *Proc. of 2019 IEEE International Conference on Communications (ICC)*, pp.1–7, 2019. Article (CrossRef Link)

[3] 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, Sept. 2020. Article (CrossRef Link)

[4] F. Aubry, D. Lebrun, Y. Deville, and O. Bonaventure, "Traffic duplication through segmentable disjoint paths," in *Proc. of 2015 IFIP Networking Conference (IFIP Networking)*, pp. 1-9, 2015. Article (CrossRef Link)

[5] F. Hao, M. Kodialam, and T. V. Lakshman, "Optimizing restoration with segment routing," in *Proc. of the 35th Annual IEEE International Conference on Computer Communications*, pp. 1-9, 2016. Article (CrossRef Link)

[6] K. Foerster, Y. Pignolet, S. Schmid, and G. Tredan, "CASA: congestion and stretch aware static fast rerouting," in *Proc. of IEEE Conference on Computer Communications*, pp. 469-477, 2019. Article (CrossRef Link)

[7] F. Lazzeri, G. Bruno, J. Nijhof, et al., "Efficient label encoding in segment-routing enabled optical networks," in *Proc. of 2015 International Conference on Optical Network Design and Modeling (ONDM)*, pp. 34-38, 2015. Article (CrossRef Link)

[8] M. Ghobadi and R. Mahajan, "Optical Layer Failures in a Large Backbone," in *Proc. of the 2016 Internet Measurement Conference*, pp. 461-467, 2016. Article (CrossRef Link)

[9] G. Trimponias, Y. Xiao, H. Xu, et al., "Centrality-based Middle-point Selection for Traffic Engineering with Segment Routing," *arXiv preprint arXiv:1703.05907*, 2017. Article (CrossRef Link)

[10] C. Filsfils, N. K. Nainar, C. Pignataro, et al., "The segment routing architecture," in *Proc. of 2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6, 2015. Article (CrossRef Link)

[11] M. Suchara, D. Xu, R. Doverspike, et al., "Network Architecture for Joint Failure Recovery and Traffic Engineering," *ACM SIGMETRICS Performance Evaluation Review*, vol. 39, no. 1, pp. 97-108, June. 2011. Article (CrossRef Link)

[12] T. Settawatcharawanit, V. Suppakitpaisarn, S. Yamada and Y. Ji, "Segment routed traffic engineering with bounded stretch in software-defined networks," in *Proc. of 2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pp. 477-480, 2018. Article (CrossRef Link)

[13] X. Li and K. L. Yeung, "Traffic engineering in segment routing using MILP," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1941-1953, Sept. 2020. Article (CrossRef Link)

[14] M. Jadin, F. Aubry, P. Schaus, and O. Bonaventure, "CG4SR: Near optimal traffic engineering for segment routing with column generation," in *Proc. of IEEE Conference on Computer Communications*, pp. 1333-1341, 2019. Article (CrossRef Link)

[15] G. Trimponias, Y. Xiao, X. Wu, et al., "Node-constrained traffic engineering: Theory and applications," *IEEE/ACM Transactions on Networking*, vol. 27, no.4, pp. 1344–1358, Aug. 2019. Article (CrossRef Link)

[16] R. Bhatia, F, Hao, M. Kodialam, and T.V. Lakshman, "Optimized network traffic engineering using segment routing," in *Proc. of 2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 657-665, 2015. Article (CrossRef Link)

[17] A. Cianfrani, M. Listanti, and M. Polverini, "Translating traffic engineering outcome into segment routing paths: The encoding problem," in *Proc. of 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 245-250, 2016. Article (CrossRef Link)

[18] X. Li and K. L. Yeung, "Fast Reroute in Hybrid Segment Routing Network," in *Proc. of 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1-6, 2020. Article (CrossRef Link)

[19] Z. N. Abdullah, I. Ahmady and I. Hussain, "Segment routing in software defined networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no.1, pp. 464–486, 2019. Article (CrossRef Link)

[20] P. L. Ventre, S. Salsano, M. Polverini, et al., "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. Article (CrossRef Link)

[21] X. Hou, M. Wu, and M. Zhao, "An optimization routing algorithm based on segment routing in software-defined networks," *Sensors*, vol. 19, no. 1, p. 49, Jan. 2019. Article (CrossRef Link)

[22] A. Sgambelluri, F. Paolucci, A. Giorgetti, et al., "Experimental demonstration of segment routing," *Journal of Lightwave Technology*, vol. 34, no. 1, pp. 205–212, Jan. 2016. Article (CrossRef Link)

[23] F. Aubry, S. Vissicchio, O. Bonaventure, and Y. Deville, "Robustly disjoint paths with segment routing," in *Proc. of the 14th international conference on emerging networking experiments and technologies*, pp. 204-216, 2018. Article (CrossRef Link)

[24] D. J. Pearce and P. H. J. Kelly, "A Dynamic Topological Sort Algorithm for Directed Acyclic Graphs," *ACM Journal of Experimental Algorithmics*, vol. 11, no. 1.7, pp. 1–24, Feb. 2006. Article (CrossRef Link)

[25] A. Giorgetti, A. Sgambelluri, F. Paolucci, et al., "Segment routing for effective recovery and multi-domain traffic engineering," *Journal of Optical Communications and Networking*, vol. 9, no.2, pp. A223–A232, 2017. Article (CrossRef Link)

[26] W. Wu, J. Liu, and T. Huang, "The source-multicast: A sender-initiated multicast member management mechanism in SRv6 networks," *Journal of Network and Computer Applications*, vol. 153, 2020. Article (CrossRef Link)

[27] H. Roomi, S. Hamid, and S. Khorsandi, "Semi-oblivious segment routing with bounded traffic fluctuations," in *Proc. of Electrical Engineering (ICEE)*, pp. 1670-1675, 2018. Article (CrossRef Link)

[28] T. Schüller, N. Aschenbruck, M. Chimani, et al., "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, Aug. 2018. Article (CrossRef Link)

[29] L. Luo, H. Yu, S. Luo, et al., "Scalable explicit path control in software-defined networks," *Journal of Network and Computer Applications*, vol. 141, pp. 86–103, 2019. Article (CrossRef Link)

[30] C. Hong, S. Mandal, M. Al-Fares, et al., "B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN," in *Proc. of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pp. 74–87, 2018. Article (CrossRef Link)

[31] A. Maghyereh and H. Al-Zoubi, "Value-at-risk under extreme values: the relative performance in MENA emerging stock markets," *International journal of managerial finance*, vol. 2, no. 2, 2006. Article (CrossRef Link)

[32] P. Krokhmal, J. Palmquist, and S. Uryasev, "Portfolio optimization with conditional value-at-risk objective and constraints," *Journal of risk*, vol. 4, no. 2, pp. 43–68, 2002. Article (CrossRef Link)

[33] V. L. Boginski, C. W. Commander, and T. Turko, "Polynomial-time identification of robust network flows under uncertain arc failures," *Optimization Letters*, vol. 3, no. 3, pp. 461–473, 2009. Article (CrossRef Link)

[34] A. Markopoulou, G. Iannaccone, S. Bhattacha

[35] ryya, et al., "Characterization of failures in an operational IP backbone network," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 749-762, Aug. 2008. Article (CrossRef Link)

[36] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of banking & finance*, vo. 26, no. 4, pp. 1443–1471, 2002. Article (CrossRef Link)

[37] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of risk*, no. 2, pp. 21–42, 2000. Article (CrossRef Link)

[38] J. Bogle, N. Bhatia, M. Ghobadi, et al., "TEAVAR: striking the right utilization-availability balance in WAN traffic engineering," in *Proc. of the ACM Special Interest Group on Data Communication*, pp. 29–43, 2019. Article (CrossRef Link)

[39] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," *arXiv preprint, arXiv:1209.5145*, 2012. Article (CrossRef Link)

[40] L. Xie, Y. Ding, H. Yang, and Z. Hu, "Mitigating LFA through segment rerouting in IoT environment with traceroute flow abnormality detection," *Journal of Network and Computer Applications*, vol. 164, 2020. Article (CrossRef Link)

Zengwei Zheng received the B.S. and Master degrees in Computer Science and Western Economics from Hangzhou University, China, in 1991 and 1998, respectively, and the Ph.D. degree in Computer Science and Technology from Zhejiang University, China, in 2005. He is currently a Professor with the School of Computer and Computing Science, the Director of Intelligent Plant Factory of Zhejiang Province Engineering Laboratory, the Director of the Hangzhou Key Laboratory for IoT Technology and Application, Zhejiang University City College. His research interests include wireless sensor networks, location-based service, Internet of Things, digital agriculture, and pervasive computing. He is a senior member of CCF.

Chenwei Zhao received his B.S degree in School of Instrument Science and Opto-electronics Engineering from Hefei University of Technology in 2017. He is a master candidate of College of Computer Science and Technology, Zhejiang University and is now studying at Zhejiang University City College. His research topic includes network resource allocation and optimization, network traffic scheduling and routing under software-defined networking (SDN) environment.

Jianwei Zhang received his B.E. and Ph.D. degrees in Computer Science and Technology from Zhejiang University, Hangzhou, China, in 2008 and 2013, respectively. He is currently a research associate professor with the School of Computer and Computing Science in Zhejiang University City College. His research interests include network optimization and control, network algorithm, network architecture, software-defined networking (SDN) and peer-to-peer networking. He is a member of the IEEE and the CCF.

Jianping Cai is an associate professor with the School of Computer and Computing Science in Zhejiang University City College in Hangzhou, China. He also works as a researcher in Embedded Systems Laboratory in the same university. His interests include network architecture and protocol, Internet of Things. He received his M. Sc. from Zhejiang University in 2002.