

# QoS-Aware Approach for Maximizing Rerouting Traffic in IP Networks

Wenyan Cui<sup>1</sup>, Xiangru Meng<sup>1</sup>, Huanhuan Yang<sup>1,2</sup>, Qiaoyan Kang<sup>1</sup> and Zhiyuan Zhao<sup>1</sup>

<sup>1</sup>College of Information and Navigation, Air Force Engineering University  
Xi'an, Shanxi 710077 - China  
[e-mail: cwy\_edu@163.com]

<sup>2</sup>Department of Electronic Engineering, Tsinghua University  
Beijing, 100084 - China

\*Corresponding author: Xiangru Meng

*Received March 21, 2016; revised June 24, 2016; accepted July 22, 2016;  
published September 30, 2016*

---

## Abstract

Network resilience provides an effective way to overcome the problem of network failure and is crucial to Internet protocol (IP) network management. As one of the main challenges in network resilience, recovering from link failure is important to maintain the constancy of packets being transmitted. However, existing failure recovery approaches do not handle the traffic engineering problem (e.g., tuning the routing-protocol parameters to optimize the rerouting traffic flow), which may cause serious congestions. Moreover, as the lack of QoS (quality of service) restrictions may lead to invalid rerouting traffic, the QoS requirements (e.g., bandwidth and delay) should also be taken into account when recovering the failed links. In this paper, we first develop a probabilistically correlated failure model that can accurately reflect the correlation between link failures, with which we can choose reliable backup paths (BPs). Then we construct a mathematical model for the failure recovery problem, which takes maximum rerouting traffic as the optimizing objective and the QoS requirements as the constraints. Moreover, we propose a heuristic algorithm for link failure recovery, which adopts the improved  $k$  shortest path algorithm to splice the single BP and supplies more protection resources for the links with higher priority. We also prove the correctness of the proposed algorithm. Moreover, the time and space complexity are also analyzed. Simulation results under NS2 show that the proposed algorithm improves the link failure recovery rate and increases the QoS satisfaction rate significantly.

---

**Keywords:** Network Resilience, Link Failure, Recovery, Reroute, Multiple Backup Paths, quality of service

---

This work was jointly supported by the National Natural Science Foundation of China (No. 61201209, 61401499), the Natural Science Foundation of Shanxi (No. 2013JQ8013, 2015JM6340).

## 1. Introduction

With the rapid development of communication technology, network bandwidth increases significantly. There are 53 wireless providers in 20 countries who have deployed or are considering deploying backbone networks with 400 Gb/s [1]. Therefore, even short-duration link failures can result in large-scale packets disruption in a high-speed backbone network, which will seriously affect the communication quality. According to the observation of the Internet service providers (ISPs), backbone network links break down with the probability of about 30% in a year, which reflects that link failure is a common problem in current networks [2]. Aiming at keeping sustainable packets transmitting when the network link breaks down, network resilience has been put forward as a fundamental ingredient of the future Internet paradigm. Therefore, reducing the discarded traffic in the event of link failure is the problem that must be solved at present.

In a network resilience environment, recovering from link failures means reducing the probability of network outages as far as possible, which is usually realized by redistributing the network routing resources when the substrate network still keeps connected. Traditional link failure recovery schemes can be divided into two totally different patterns: active and passive link failure recovery modes. The passive mode refers to dynamically and adaptively redistributing the whole network resources when links break down. However, the redistributing process is time consuming, and the NOC (network operation center) adopts some restrictions (e.g., tuning the timer and restricting the flooding range) to accelerate the convergence process, but the dynamic process may still lead to a substantial amount of dropped packets. Therefore, most works focus on active link failure recovery methods. With the active mode, network resources are configured and backups are reserved in advance. When a link breaks down, traffic on it is quickly switched to the available backups, which can keep transmitting the packets without interruption. Multi-topology routing and backup path (BP) based approaches are typical active modes. For multi-topology routing [e.g., MRC (multiple route configuration) and RRL (resilient route layer)], multiple backup topologies need to be configured, which occupy much routing table space [3]. BP-based link failure recovery approaches can provide end-to-end rerouting and distribute traffic at the global level, which is easy to implement based on the existing protocols. Therefore, a BP-based approach has higher potential, which is a research hotspot in the current link failure recovery field [4-10].

The idea of link failure recovery is to keep the traffic forwarding without interruption in the network. Thus, the problem should be understood in the traffic forwarding perspective. Most existing works adopt a single BP and do not consider the bandwidth resource problem. However, sometimes a single BP cannot satisfy the rerouting requirement, which may result in a large amount of discarded traffic due to lack of bandwidth resources. A recent work has introduced multipath routing into the failure recovery problem, which reduces discarded traffic [5]. Some works consider traffic engineering in the failure recovery process, whose objective is to balance the traffic load, and they suppose that the BPs are available [6, 7]. However, the BPs are not always available and may break down sometimes, which is a failure correlation problem. It decides whether the rerouting can succeed and the traffic carried by the failed link can be rerouted. For reliable packet transmission, a multipath-based reliable routing protocol with fast recovery of failures was proposed, which can handle the dynamic topological change of the mobile ad hoc networks (MANETs) efficiently [8]. Moreover, most existing failure recovery methods take maximum rerouting traffic as the target, but they all

ignore whether the rerouted traffic meets the requirement of the user [9-11]. Even if the traffic from the failed link is rerouted successfully, it may not satisfy the quality of service (QoS) requirements, which may mean invalid traffic to the user. Although a recent work proposed a self-adaptive multipath routing that satisfies the QoS constraints, it was not proposed for the link failure recovery problem [12]. Therefore, most existing link failure recovery methods cannot effectively ensure the QoS.

In this paper, we address the link failure recovery problem, which is the maximum rerouting problem under bandwidth and delay constraints. We propose a novel BP-based link failure recovery method that can efficiently solve the aforementioned problems. We first build a probabilistically correlated failure model based on the multiple BP strategy. With this, we can accurately calculate the failure probability of the links in which failure correlation with each other exists. Moreover, it can help us choose reliable BPs. We formulate the optimization objective on the basis of a probabilistically correlated failure model, which considers the reliability of BPs and is the most probable discarded traffic from all the failed links in the network. Then we build a mathematical model for the failure recovery problem, where we take the flow conservation as the basis and consider the reliability of BP and QoS constraints. Moreover, we design the MRT-QoS algorithm (QoS-aware approach for maximizing rerouting traffic) to solve the mathematical problem. MRT-QoS is a nested algorithm which consists of the single BP splicing algorithm and the BP selecting algorithm, where we take QoS as constraints and adopt the improved  $k$  shortest path algorithm to splice a single BP. We rank the links according to the failure probability and the carried traffic, and allocate more protection resources for links with higher priority. Also, we prove that the proposed algorithm can be executed correctly and efficiently.

The following are our major contributions in this paper:

1. We develop a probabilistically correlated failure model on the basis of multiple BPs, which does not assume that the BPs are absolutely reliable and gives the quantified impact that the failed link is on the reliability of BPs.

2. We model the link failure recovery problem with MILP (mixed integer linear programming), which takes the maximum rerouting traffic as the objective and considers the flow conservation constraint and QoS constraints. The optimization objective involves the traffic that exceeds the available bandwidth of BPs and the traffic due to BP failure.

3. We propose a novel heuristic multiple rounds iterative algorithm (MRT-QoS), with which we can obtain multiple reliable BPs for the links according to the priority in descending order. Moreover, the proposed algorithm has polynomial computation complexity.

4. We compare MRT-QoS with several existing BP-based algorithms through extensive simulations to demonstrate the effectiveness and efficiency of our proposed algorithm.

The rest of this paper is organized as follows. Section 2 presents the related work. In Section 3, we develop a correlated failure model and quantify the optimization objective. In Section 4, we develop a mathematical model for the link failure recovery problem. Our proposed MRT-QoS algorithm is described in Section 5. In Section 6, we evaluate the proposed algorithm through extensive simulations. We conclude this paper in Section 7.

## 2. Related Work

There are three kinds of existing works related to our work.

### 2.1 Backup path based link failure recovery approaches

The objective of most BP-based link failure recovery methods is to choose a connected BP that

focuses on the connectivity problem [4, 7]. However, the BP may lack adequate bandwidth resources to keep packet forwarding, which would result in traffic discarding and further cause more failures. Moreover, the delay of the BP may be too large, which can result in the rerouting traffic not satisfying the delay requirement of service. Aiming at the problem of lacking available bandwidth resources, Wang et al. [5] introduced the multiple paths technology into the failure recovery process. They adopted multiple BPs to share the traffic from the failed link and used a hash table to avoid the data packet disorder. Suchara et al. [6] proposed a new network structure that combines failure recovery and traffic engineering, which conducts traffic optimization in the multiple paths. But their objective was to balance the load rather than acquire maximum rerouting traffic that is QoS-aware. Moreover, they suppose that the bandwidth resources satisfy the requirement of failure recovery. Most existing methods take link failures as independent events and only configure one BP for each link. The failure recovery methods based on multiple BPs just configure multiple BPs for the link. They do not optimize the rerouting traffic in the whole network and also do not limit the QoS requirements.

## 2.2 Link failure correlation

Most link failure recovery methods consider the link failures as independent events. Generally, they suppose that the BP and its protected link do not fail at the same time, which means that the BP is reliable all the time [3-6]. However, there may be correlations between failures. Zheng et al. [9] point out that the group of logical links carried by the same fiber may break down simultaneously. Their method considers the correlation between physical links and logical links. Since the failure correlation is obtained by the mapping relation from the logical layer to the physical layer, it increases uncertainty and may lead to deviation. Zheng et al. [10] join the correlated failure in BP selection to overlay network survivability enhancement. However, their objective is to find a reliable BP rather than allocate the traffic load for failed links. Also, they do not rank the priority of links.

## 2.3 QoS routing and multi path routing

Multi path routing is proposed to achieve the traffic engineering target, which constructs multiple bypasses for an end-to-end path. With this, the traffic carried by a single path can be divided into several parts, which can efficiently forward traffic and ensure the QoS. To achieve this target, a new protocol called QoS routing is proposed [13, 14]. Banner et al. [7] proposed a network architecture that can acquire fast restoration when links break down. However, their aim was to design low-capacity backup networks rather than configuring the optimal BPs on the basis of existing network, which is not practical as most network topologies are stationary. Moreover, QoS routing does not consider the correlation failure problem. There is hardly any work that combines QoS routing and the link failure recovery process. Zheng et al. [9] adopt multiple BPs to recover link failure and consider the reliability of BPs, but they ignore the delay requirements. Wang et al. [5] proposed a resilient routing reconfiguration method to achieve congestion-free performance under multiple failures. However, the target is to balance traffic load rather than seek the maximal rerouting of traffic. Also, they consider the BP as reliable and ignore the fact that a failed link can result in the failure of the backup path.

Our work differs from the existing BP-based link failure recovery methods in the following three aspects. First, we adopt multiple BPs for the link failure recovery problem and propose a probabilistically correlated failure model. The model can provide adequate bandwidth resources for the links that fail easily or carry more traffic. With the correlated failure model, we can choose a more reliable BP. Second, we consider the delay constraint of the BP in the

failure recovery process to ensure that the rerouting traffic meets the delay requirement, which is important for a service with high time-sensitive requirement. It can avoid invalid traffic being rerouted to save the finite bandwidth resources. Third, we use MILP to formulate the failure recovery problem and derive the optimization objective. Different from previous works which try to recover the traffic from failed links, our objective is to maximize the rerouting traffic that is QoS-aware to avoid invalid rerouting of traffic. Besides, our heuristic multi-round algorithm has polynomial computation complexity, which solves the problem of exponential calculation complexity in the previous works [15, 16].

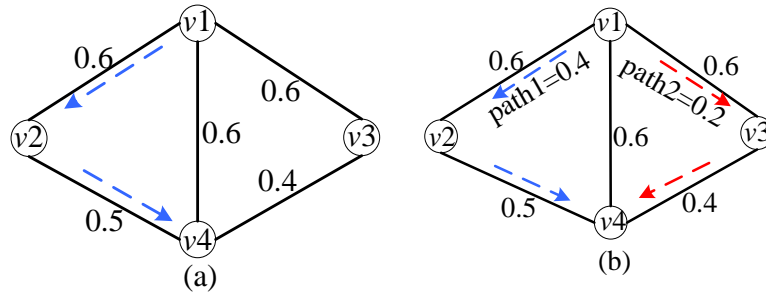
### 3. Problem Statement

In this section, we first describe the BP strategy and the probabilistically correlated failure model. Then, we present the objective of link failure recovery problem.

#### 3.1 Backup path strategy

The BP strategy is the key to the link failure recovery approach, which has great influence on the link failure recovery. The traffic load on the failed link is directly switched to the BP when the link breaks down, which can reduce the discarded traffic. Most existing link failure recovery methods adopt a single BP to recover the failed links. However, when the rerouting traffic exceeds the available bandwidth of the BP, part of traffic will be discarded, which shows that a single BP cannot satisfy the requirement. An example of the necessity for multiple BPs to protect a link is presented in Fig. 1, where the capacity of the entire link (represented by the edges) is 1, and the numbers next to the edges represent the amount of traffic load. Moreover, we use  $e_{1,4}$  to denote the link from node  $v_1$  to  $v_4$ . We adopt single backup path  $\{v_1 \rightarrow v_2 \rightarrow v_4\}$  to protect link  $e_{1,4}$ , as shown in Fig. 1(a), and the available bandwidth resources that the BP can provide are  $\min\{1 - 0.6, 1 - 0.5\} = 0.4$ . When  $e_{1,4}$  breaks down, the whole traffic load carried on the link  $e_{1,2}$  exceeds its bandwidth resources, which shows that multiple BPs are necessary for protecting the link against failures. In Fig. 1(b), we use the BP  $\{v_1 \rightarrow v_2 \rightarrow v_4\}$  and  $\{v_1 \rightarrow v_3 \rightarrow v_4\}$  together to share the traffic load from the failed link  $e_{1,4}$ , where path1 carries traffic load of 0.4 and path2 carries traffic load of 0.2. The traffic load on the failed link  $e_{1,4}$  can be forwarded without interruption based on the multiple BPs method.

More BPs can help the link recover from failures effectively. However, if the link is configured with too many BPs, the configuration complexity and storage cost will greatly increase. Moreover, there exists a strict limit to the number of BPs that can be configured in the network. Thus we set that each link can be configured up to  $N$  BPs. The queue delay of the congested network grows exponentially with the increase of the hop, and the signal quality in the optical network drops sharply with its increase. Therefore, to support the QoS requirement of network service in the failure recovery process, it is necessary to impose restriction on the hop of the BP. The maximum hop of the BP is limited to  $H$  in this paper.



**Fig. 1.** Example of the necessity with multiple backup paths for protecting link.  
(a) Single backup path. (b) Two backup paths

### 3.2 Probabilistically correlated failure model

Most existing link failure recovery algorithms are proposed to solve the independent link failure problem. They consider the link failures as independent and separate. However, sometimes the link failures are correlated with each other. For example, when the substrate optical fiber link breaks down, the logical links carried by it also will probably fail at the same time. When one logical link breaks down, the other logical links carried by the same optical fiber link may also break down with a certain probability, which means that there exists a probabilistic correlation between the link failures. The shared risk link group (SRLG) model can be used to represent a group of links that share the same risk, which means that the other links in the same group will break down with probability 1 when a link in the SRLG breaks down [15, 16]. However, the correlated link failures are not absolutely correlated, and the link breaks down with only a certain probability when the correlated link breaks down. Therefore, we introduce the concept of PSRLG (probabilistically shared risk link group) to represent the probabilistically correlated relation, which is based on traditional SRLG model.

**Definition 1** (PSRLG) When an arbitrary event  $r \in R$  occurs, the set involving links that the failure probability is non-zero is defined as PSRLG about the event  $r$ :

$$r_{PSRLG} = \{e_{i,j} \in E : p_r(i,j) \neq 0\} \quad (1)$$

where  $R$  is the set of SRLG events, and  $p_r(i,j)$  is the probability that the link  $e_{i,j}$  breaks down when SRLG event  $r$  occurs.

When an event  $r$  occurs, if there exists a link failure correlation between link  $e_{i,j}$  and link  $e_{u,v}$ , we can derive that  $p_r(i,j) \neq 0$  and  $p_r(u,v) \neq 0$ . However, with the traditional SRLG model, link failure correlation between link  $e_{i,j}$  and link  $e_{u,v}$  means that  $p_r(i,j) = 1$  and  $p_r(u,v) = 1$  when event  $r$  occurs. Obviously, the traditional SRLG model is only a special case of PSRLG. We can depict the characteristic of correlated link failure more accurately by adopting the PSRLG model.

When there exists a link failure correlation between link  $e_{i,j}$  and link  $e_{u,v}$ , the probabilities that they each break down are defined as  $p_{u,v}$  and  $p_{i,j}$ , respectively:

$$p_{u,v} = p_r \cdot p_{i,j} \quad (2)$$



$$p_{i,j} = p_r \cdot p_r(i, j) \tag{3}$$

where  $p_r$  is the probability that event  $r$  occurs.

According to Eqs. (2) and (3), we can derive that  $p_{u,v}$  and  $p_{i,j}$  are nonzero only when  $p_r \neq 0$ . When there exists link failure correlation between  $e_{i,j}$  and  $e_{u,v}$ , the probabilities that they each break down are determined by the event  $r$ . Thus, we can easily calculate the breakdown probability of the links where link failure correlation with each other exists. For example, as shown in Fig. 2, the logical links  $e_{a,b}$  and  $e_{c,d}$  can be carried by two substrate optical fiber paths, where the bearing paths for link  $e_{a,b}$  are path  $\{e_{A,B} \rightarrow e_{B,C} \rightarrow e_{C,D} \rightarrow e_{D,F}\}$  and path  $\{e_{A,B} \rightarrow e_{B,F}\}$ , and the bearing paths for link  $e_{c,d}$  are path  $\{e_{A,B} \rightarrow e_{B,E} \rightarrow e_{E,F}\}$  and path  $\{e_{A,B} \rightarrow e_{B,F}\}$ . When the bearing link  $e_{B,F}$  breaks down, the link  $e_{a,b}$  and  $e_{c,d}$  may be influenced by the link  $e_{B,F}$ , but they can also operate normally due to the network's self-healing mechanism. Thus logical link  $e_{a,b}$  and  $e_{c,d}$  can keep the normal state by switching the failed bearing path to the available bearing path. Therefore, we can conclude that the link failure correlation is actually a probabilistic correlation.

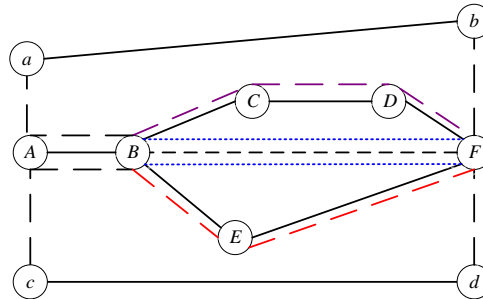


Fig. 2. Example of probabilistically correlated failure

### 3.3 Optimization objective

From the perspective that the network continuously forwards traffic, the objective is to reduce discarded traffic as much as possible.

For each link  $e_{i,j} \in E$ , we denote its bandwidth capability by  $c_{i,j}$  and the traffic load under normal state by  $l_{i,j}$ . Note that we concentrate on the backbone network. The backbone links carry a large flow and the load on them show a stable status over a long period. Thus, we take the long-term average load of links as the link load in the following formulation and assume that the load of link changes only when failures occur in the network. Similar assumption can also be found in [9]. Thus, the available bandwidth resource of link  $e_{i,j}$  is  $c_{i,j} - l_{i,j}$ , which is the maximum rerouting traffic that link  $e_{i,j}$  can bear. We denote the  $k$ th BP for link  $e_{i,j}$  by  $B_{i,j}^k$  and the bandwidth that  $B_{i,j}^k$  reserves for link  $e_{i,j}$  by  $r_{i,j}^k$ . Thus, the whole rerouting traffic load protected by  $N$  BPs is  $\sum_{k=1}^N r_{i,j}^k$ . If the  $N$  BPs are available, the discarded traffic is defined as Eq. (4), which is the difference between the traffic load on the failed link and the rerouting

traffic on the BPs. Obviously, it is better to discard less traffic.

$$TD_{i,j} = l_{i,j} - \sum_{k=1}^N r_{i,j}^k \quad (4)$$

In Eq. (4), we assume that all the  $N$  BPs are available. However, there exists probabilistic failure correlation between links according to the earlier analysis. The BP  $B_{i,j}^k$  may also break down at the same time when link  $e_{i,j}$  breaks down. If the BP  $B_{i,j}^k$  breaks down, the rerouting traffic load carried by it is discarded. We denote the probability that link  $e_{i,j}$  breaks down by  $p_{i,j}$ , and the conditional probability that the BP  $B_{i,j}^k$  breaks down when link  $e_{i,j}$  fails by  $P(B_{i,j}^k | e_{i,j})$ . Then, the discarded traffic that considers the reliability of the BP when link  $e_{i,j}$  breaks down can be formulated as follows:

$$TD_{i,j,p} = p_{i,j} (\sum_{k=1}^N P(B_{i,j}^k | e_{i,j}) r_{i,j}^k + l_{i,j} - \sum_{k=1}^N r_{i,j}^k) \quad (5)$$

where  $P(B_{i,j}^k | e_{i,j})$  is the conditional probability, which is nonzero only when link  $e_{i,j}$  and the  $k$ th backup path  $B_{i,j}^k$  are failure-correlated. We calculate  $P(B_{i,j}^k | e_{i,j})$  in the following section.

We denote the links constituting the BP  $B_{i,j}^k$  by the set  $S(B_{i,j}^k)$ . The probability that the BP  $B_{i,j}^k$  breaks down can be formulated as follows:

$$P(B_{i,j}^k) = 1 - \prod_{e_{u,v} \in S(B_{i,j}^k)} (1 - p_{u,v}) \quad (6)$$

where the link  $e_{u,v}$  is an arbitrary element of  $S(B_{i,j}^k)$ , and  $p_{u,v}$  is the probability that link  $e_{u,v}$  breaks down. Combining with the probabilistically correlated failure model,  $P(B_{i,j}^k | e_{i,j})$  can be further formulated as follows:

$$P(B_{i,j}^k | e_{i,j}) = (1 - \prod_{e_{u,v} \in S(B_{i,j}^k)} (1 - p_r \cdot p_r(u,v))) / p_r \cdot p_r(i,j) \quad (7)$$

Therefore, the discarded traffic from all the links in the network due to link failures can be formulated as follows:

$$TD = \sum_{e_{i,j} \in E} (\sum_{k=1}^N (1 - \prod_{e_{u,v} \in S(B_{i,j}^k)} (1 - p_r \cdot p_r(u,v))) r_{i,j}^k + p_r \cdot p_r(i,j) (l_{i,j} - \sum_{k=1}^N r_{i,j}^k)) \quad (8)$$

The objective of the proposed method is to minimize  $TD$ , which is shown in Eq. (8).

#### 4. Mathematical Model for Link Failure Recovery Problem

In this section, we adopt MILP to model the link failure recovery problem, which takes the minimal discarded traffic as the objective and the QoS requirement as the constraint.

The mathematical model for link failure recovery problem is provided as follows:

1) Variables



$r_{i,j}^k$  : real variable, which is the reserved bandwidth of the  $k$  th BP  $B_{i,j}^k$  for link  $e_{i,j}$ .

$x_{u,v}^{i,j,k}$  : Boolean variable, which is 1 if link  $e_{u,v}$  constitutes the BP  $B_{i,j}^k$ , and otherwise 0.

## 2) Objective function

$$\text{Min } TD \quad (9)$$

The objective function is to minimize the discarded traffic in the whole network, which involves two parts. One part is the traffic beyond the rerouting ability of all the BPs, and the other part is the discarded traffic due to the BP failure.

## 3) Constraints

### ① Flow conservation constraint

$$\sum_{\forall u, e_{v,u} \in E} f_{e_{v,u}}^h(e_{i,j}) - \sum_{\forall u, e_{u,v} \in E} f_{e_{u,v}}^{h-1}(e_{i,j}) = 0 \quad \forall e_{i,j} \in E, \forall v \in V \setminus \{i, j\} \quad (10)$$

where constraint (10) is the flow conservation limit of a node in the network, which represents that all the traffic that enters arbitrary node  $v \in V \setminus \{i, j\}$  with backup paths of  $h-1$  hops equals the traffic that exits the node with BPs of  $h$  hops. Note that the end nodes of link  $e_{i,j}$  do not meet this constraint. We denote the sum of rerouting traffic passing from all the BPs with  $h$  hops from node  $i$  to node  $u$  by  $f_{e_{u,v}}^h(e_{i,j})$ .

### ② Capacity constraints

$$\sum_{k=1}^N r_{i,j}^k \leq l_{i,j} \quad (11)$$

$$\sum_{e_{i,j} \in E} f_{e_{u,v}}(e_{i,j}) \leq c_{u,v} - l_{u,v} \quad (12)$$

where Eq. (11) is traffic constraint for rerouting, which represents that the sum of rerouting traffic from all the backup paths is no more than the traffic load of link  $e_{i,j}$ . Equation (12) is the bandwidth capacity constraint, which means that the rerouting traffic load carried by each link is no more than its available bandwidth resource. We denote the traffic load allocated to link  $e_{u,v}$  by  $f_{e_{u,v}}(e_{i,j})$  when link  $e_{i,j}$  breaks down, which can be formulated as follows:

$$f_{e_{u,v}}(e_{i,j}) = \sum_{k=1}^N x_{u,v}^{i,j,k} r_{i,j}^k \quad (13)$$

$f_{e_{u,v}}(e_{i,j})$  is the rerouting traffic from all the backup paths from node  $i$  to node  $u$ , which can also be formulated by Eq. (14).

$$f_{e_{u,v}}(e_{i,j}) = \sum_{h=0}^H f_{e_{u,v}}^h(e_{i,j}) \quad (14)$$

### ③ Variables constraints

$$1 \leq k \leq N, 0 \leq h \leq H \quad (15)$$

$$r_{i,j}^k \geq 0 \quad (16)$$

$$x_{u,v}^{i,j,k} \in \{0,1\} \quad (17)$$

where Eq. (15) represents that each link can have no more than  $N$  BPs, and the maximum delay of each backup path is  $H$  hops. Equation (16) represents that the reserved bandwidth is nonnegative.  $x_{u,v}^{i,j,k}$  meets the integer constraint (17). Thus, the mathematical model for link failure recovery problem is NP-hard. Next, we will propose algorithms to solve this NP-hard problem.

## 5. Heuristic Algorithm Design

As pointed out earlier, the MILP model for the link failure recovery problem is NP-hard. Although we can adopt traditional linear programming methods (e.g., the simplex method) to solve it, the computational complexity increases exponentially with the increase of the problem size. Hence, traditional linear programming methods are not suitable for solving the link failure recovery problem in large-scale networks. In this section, we propose a novel heuristic algorithm (MRT-QoS) based on multiple BPs with QoS constraints, which consists of a single BP splicing algorithm and BP selecting algorithm. The optimized object of the proposed algorithm is to select no more than  $N$  BPs that are QoS-aware, and reasonably allocate resources so that the discarded traffic in the whole network is minimal.

### 5.1 Single backup path splicing algorithm

The BP for a link is selected one by one. The single BP selection is similar to the  $k$  shortest path algorithm that calculates the shortest path, which expands the BP by adding the link one by one. Suppose that link  $e_{i,j}$  has been configured with  $k-1$  BPs; then the reduced discarded traffic by adding another BP can be formulated as follows:

$$\Delta TD_{i,j} = p_{i,j} \cdot r_{i,j}^k (1 - P(B_{i,j}^k | e_{i,j})) \quad (18)$$

The objective of splicing a single BP with the MRT-QoS algorithm is to construct a BP  $B_{i,j}^k$  to maximize  $\Delta TD_{i,j}$ . We propose a single BP splicing algorithm as shown in Algorithm 1.

In the single BP splicing process, MRT-QoS adopts the  $k$  shortest path algorithm to calculate the shortest BP. Since the BPs acquired from the  $k$  shortest path algorithm do not have the bandwidth and delay constraints, part of the rerouting traffic may be invalid. Sometimes, even if the BP has available bandwidth but the delay does not meet the requirement, which would also lead to invalid rerouting traffic. Therefore, we should first choose the shortest BP that has available bandwidth resources and meets the delay requirement at the same time.

The details of single BP splicing algorithm are as follows. For each link in the network, we adopt the  $k$  shortest path algorithm to compute the shortest BP. Then we remove the BPs that cannot provide the available bandwidth resources and do not satisfy the delay requirement. Finally, we choose the BP that makes the discarded traffic minimal as the BP and output the BP splicing result.

---

**Algorithm 1** Single BP Splicing Algorithm
 

---

**Input:**  $G = (V, E)$  : network topology

$H$  : the maximal hops of BP

**Output:** **SINGLEBP\_SPLICING\_List**: the splicing list for single BP

1. **for** arbitrary link  $e_{i,j} \in E$  **do**
  2. Calculate set  $\Omega(BP_{i,j})$  of the shortest BP for link  $e_{i,j}$  based on  $k$  shortest path algorithm;
  3. **if** no shortest path exists **then**
  4. Return **SINGLEBP\_SPLICING\_FAILED**;
  5. **else**
  6. **if** there does not exist  $BP \in \Omega(BP_{i,j})$  that has available bandwidth resource **then**
  7. Return **SINGLEBP\_SPLICING\_FAILED**;
  8. **else**
  9. Select  $BP \in \Omega(BP_{i,j})$  to construct the set  $\Omega(BP_{i,j}^{BW}) \subseteq \Omega(BP_{i,j})$  that arbitrary  $BP \in \Omega(BP_{i,j}^{BW})$  has available bandwidth resource;
  10. **if** no BP meets the delay requirement **then**
  11. Return **SINGLEBP\_SPLICING\_FAILED**;
  12. **else**
  13. Select  $BP \in \Omega(BP_{i,j}^{BW})$  to construct the set  $\Omega(BP_{i,j}^{BW-DL}) \subseteq \Omega(BP_{i,j}^{BW})$  that arbitrary  $BP \in \Omega(BP_{i,j}^{BW-DL})$  meets the delay requirement;
  14. **for** each  $BP \in \Omega(BP_{i,j}^{BW-DL})$  **do**
  15. Calculate  $\Delta TD_{i,j}$ ;
  16. **end for**
  17. Choose the BP with the maximal  $\Delta TD_{i,j}$  as the  $k$  th BP for link  $e_{i,j}$  and save the BP selection result into **SINGLEBP\_SPLICING\_LIST**;
  18. **end if**
  19. **end if**
  20. **end if**
  21. **end for**
  22. Return **SINGLEBP\_SPLICING\_LIST**;
- 

## 5.2 Backup Path Selecting algorithm

Since the rerouting traffic of the links bearing more traffic load is larger when they break down, part of traffic may be discarded due to the lack of bandwidth resources. As the work environments of the links are different, some links that wear easily or work in a corrosive environment may easily break down. They need more protection resources. Therefore, links with high traffic load and high failure probability should be protected in priority. In this paper, we adopt the product of traffic load and failure probability to rank the link priority, which can be formulated as follows:

$$LP_{i,j} = p_{i,j} \cdot l_{i,j} \quad (19)$$

In the BP selection process, we adopt a greedy strategy to select the BPs for each link according to their ranks, which helps in protecting the links that have higher failure probability and bear more traffic load.

The details of the BP selecting algorithm are shown in Algorithm 2, which works as follows:

For each link in the network, we calculate the priority  $LP$  and rank the links according to the priority in descending order. For the link with the highest rank that has no configured BPs, we adopt the single BP algorithm to build no more than  $N$  BPs for it. If there is no more candidate BP for the link, the existing BPs are configured to the link, which will share the traffic load if the link breaks down. After selecting the BPs for the link with the highest rank, the remaining traffic load of the link and the bandwidth resources of the network are updated. The iterative selection process continues until all the links have been configured with enough BPs or there are no more available bandwidth resources.

---

**Algorithm 2** BP Selecting Algorithm

---

**Input :**  $G = (V, E)$  : network topology

**SINGLEBP\_SPLICING\_List**: the splicing list for single BP

$N$  : the maximum BPs that each link can be configured

**Output:** **BP\_SELECTNG\_List**: the selecting list for BP

1. **for** each link  $e_{i,j} \in E$  **do**
  2.   Calculate the priority  $LP_{i,j}$  of link  $e_{i,j}$ ;
  3. **end for**
  4. Rank the links according to the acquired  $LP$  in descending order and save the rank result into list **LINK\_PRIORITY\_LIST**;
  5. **for** all the links that haven't been configured BPs in the **LINK\_PRIORITY\_LIST** **do**
  6.   Choose the link with the highest rank;
  7.    $k = 1$
  8.   **while**  $k \leq N$  **do**
  9.     **Call** the single BP splicing algorithm to construct the  $k$  th BP;
  10.    **if** construction of the  $k$  th BP fails **then**
  11.     **break**
  12.    **else**
  13.     Update the rerouting traffic load for link  $e_{i,j}$  and network bandwidth resources;
  14.      $k = k + 1$ ;
  15.    **end if**
  16.   **end while**
  17. **end for**
  18. Save the BP results for each link into list **BP\_SELECTNG\_List**;
  19. Return **BP\_SELECTNG\_List**;
- 

Compared to the existing BP-based failure recovery methods, the advantages of our proposed algorithm are as follows: The MRT-QoS algorithm allocates more protection resources for the link with higher priority, which reduces the discarded traffic. Moreover, not only the bandwidth resources in each round of constructing single BP, but also the delay of BP is considered. MRT-QoS is QoS-aware and can ensure that each new BP reduces the discarded traffic as much as possible in the rerouting process.

### 5.3 Proof of Correctness

Any new algorithm should be first proved that it can work properly and efficiently, and then it can be used for solving the problem. In this section, we give the correctness proof for the MRT-QoS algorithm.

**Theorem 1** With the given network topology, suppose that we can adopt the  $k$  shortest path

algorithm to construct  $k$  shortest backup paths; then the single BP splicing algorithm can be used for constructing the BP set  $\Omega(BP_{i,j})$  such that all the BPs meet the bandwidth and delay requirements. Note that  $\Omega(BP_{i,j})$  can be a null set, which represents that there is no more BP for  $e_{i,j}$  and part of the traffic is discarded.

**Proof.** According to the algorithm described before, we can see that the extension to the  $k$  shortest path algorithm in the single BP splicing algorithm is as follows: The hop constraint is added in the process of constructing the  $k$  shortest path, which changes the judging condition that constructs the shortest path. Moreover, the BPs that have available bandwidth resources are retained. We can adopt the  $k$  shortest path algorithm to construct a  $k$  shortest path tree  $T_k$  that involves  $k$  shortest paths. Thus, single BP splicing algorithm can be realized based on  $k$  shortest path tree  $T_k$  [17].

In the follows, we adopt mathematical induction to prove the correctness of the algorithm.

When  $k = 1$ , tree  $T_k$  only involves one shortest path. If the delay exceeds when constructing the shortest path, we obtain the null set  $\Omega(BP_{i,j})$ . Otherwise, if delay meets the requirement but there are no available bandwidth resources, we also obtain the null set  $\Omega(BP_{i,j})$ . When bandwidth and delay constraints are all meet, we can acquire a BP. The proposition is true.

Suppose that the proposition is true when  $k = n$ , which means that the single BP splicing algorithm can be used for constructing the set  $\Omega(BP_{i,j})$  that meets the available bandwidth resources and delay requirement.

Next, we prove that the proposition is true when  $k = n + 1$ .

We denote two paths from node  $i$  to node  $j$  by  $p = (m_1, m_2, \dots, m_r)$  and  $q = (n_1, n_2, \dots, n_s)$ , respectively. If there exists variable  $x$  that meets the following constraints

- (1)  $x < r$  and  $x < s$
- (2)  $m_t = n_t (0 \leq t \leq x)$
- (3)  $m_{x+1} \neq n_{x+1}$

- (4)  $(n_{x+1}, n_{x+2}, \dots, n_s)$  is the shortest path from node  $n_{x+1}$  to node  $j$

then we call  $(n_x, n_{x+1})$  the deviation edge that  $q$  is relative to  $p$ , and  $(n_{x+1}, n_{x+2}, \dots, n_s)$  the shortest deviation path. To the standard shortest path algorithm, adopting the known  $n$  shortest paths to calculate  $p_{n+1}$  is as follows. It first traverses all the nodes  $m_t (0 \leq t \leq x)$  of path  $p_n$  to acquire deviation node, and acquires the shortest path from  $m_t$  to node  $j$ . Then it splices this path with the path from  $i$  to  $m_t$  on  $p_n$  to acquire the candidate paths  $p_{n+1}$ , and chooses the shortest path from candidate paths as  $p_{n+1}$ .

The main changes of the single BP splicing algorithm relative to shortest deviation path generating algorithm are as follows: In the process of traversing  $m_t$  to search the shortest path from  $m_t$  to  $j$ , the delay and bandwidth constraints are considered. The shortest paths from each node  $m_t (0 \leq t \leq x)$  to  $j$  can be calculated by the Dijkstra algorithm, and the shortest path set from node  $m_t$  to  $j$  considering delay and bandwidth constraints is the subset of the set without constraints. The acquired shortest path may be the inferior shortest path from  $m_t$  to  $j$ . Since the proposition is true when  $k = n$ , the path on  $p_n$  from  $i$  to  $m_t$  is the shortest path meeting delay and bandwidth constraints. Then, by splicing the shortest path from  $i$  to  $m_t$  and the path from  $m_t$  to  $j$ , we can acquire the candidate paths  $p_{n+1}$ . The path with the most

available bandwidth resources is selected as  $p_{n+1}$ .

Then, the proposition is true when  $k = n + 1$ .

Theorem 1 is proved by the above proof.

The BP selecting algorithm consists of multiple rounds of iterations, which is realized by nesting the single BP splicing algorithm. Therefore, under the condition that the single BP splicing algorithm can be correctly executed, the BP selecting algorithm can also be correctly executed. Moreover, the MRT-QoS algorithm is realized based on the  $k$  shortest path algorithm, which can ensure the acquired shortest path to be loop-free.

#### 5.4 Time Complexity Analysis

The single BP splicing algorithm is similar to the  $k$  shortest path algorithm, and so it has the same computation complexity  $O(|E| + |V| \log(V))$ . Since there are  $|E|$  links in the network and each link has no more than  $N$  BPs, the time complexity of the MRT-QoS algorithm is  $O(|E| + |V| \log(V) N |E|)$ .

#### 5.5 Space Complexity Analysis

The MRT-QoS algorithm adopts four vectors to store the data generated by the algorithm. One vector stores the shortest BP set  $\Omega(BP_{i,j})$  of all the links. Since each path can be represented by a series of nodes, and the needed storage space for a single BP is no more than  $V$ , the storage space is  $k|V||E|$  at most. One vector stores the single BP splicing results, and the needed storage space is  $N|V||E|$  at most. One vector stores the link priority rank results, and the needed storage space is  $|E|$ . One vector stores the BPs for links and the needed storage space is  $N|V||E|$ . Therefore, the space complexity of the MRT-QoS algorithm is  $O(N|V||E|)$ .

## 6. Performance Evaluation

In this section, we first describe the simulation environment, and then present our main evaluation results. Our evaluation focuses primarily on the comparison of MRT-QoS with several existing BP-based link failure recovery algorithms in terms of the link failure recovery rate, QoS satisfaction rate of rerouting traffic, traffic discarding rate, link overload rate, and runtime.

### 6.1 Simulation environment

NS2 is a major network simulation tool that can be used for efficiently conducting the correctness verification and performance analysis of network protocols and algorithms. Therefore, we adopt it as the simulation platform to analyze the performance of the MRT-QoS algorithm. We adopt Tier-1 backbone network as the simulation network topology, which consists of 50 nodes and 180 links. We set the link bandwidth inversely proportional to its weight, and set it to be  $5/\text{weight}$  Mbps [18]. We generate CBR data flow and randomly choose the source and destination node pair in the network to send it. The data packet in the data flow is 1 kilobytes. The transmission rate and queue length of data flow are 200 kbps and 50 data packets, respectively. The up threshold  $T_1$  and low threshold  $T_2$  are set as 10% and 100% of the queue capacity, respectively. We vary the link utilization by changing the amounts of data flow, which range from 20 to 160, corresponding to the average link utilization from 5% to

40%. Moreover, we set  $k$  of the  $k$  shortest path algorithm for calculating shortest BP to be 5.

The link failure scenarios are set as follows. We configure nine SRLG events, and each event involves 2-5 links that share the same risk. We set the probability that an SRLG event occurs as [0.05%, 0.5%] and the conditional probability for the links in the same SRLG event as [0.3, 1]. We configure 10% links of independent failures with high failure probability; the failure probability ranges from 0.1% to 0.5%, and [0.01%, 0.1%] for the other links. To achieve a relative stable performance, we configure 50 groups of probability failure scenarios, and each scenario is performed 50 times with different random seeds. We randomly select a link as failed link in each simulation, and record the average value as the final result.

Our simulation experiments evaluate the four algorithms listed in **Table 1**. MRT-QoS is the algorithm we have proposed in this paper, SelectBP is the classical BP-based algorithm proposed in [9], FR-TE [6] and R3 [5] consider the load balancing in the link failure recovery process, but they are not QoS-aware.

**Table 1.** Algorithms comparison

Notation	Description
MRT-QoS	The proposed algorithm takes QoS requirement as the constraint and considers the reliability of the BP-based on multiple BPs
SelectBP	Uses multiple rounds of iterated algorithm that considers the bandwidth resource constraints without delay constraints
FR-TE	Linear programming solving method is used that takes the load balancing as the objective joining load balancing into the link failure recovery process
R3	Linear programming solving method is utilized that takes the non-congestion of network as the objective under multiple link failures

## 6.2 Evaluation results

### 6.2.1 Link failure recovery rate and runtime

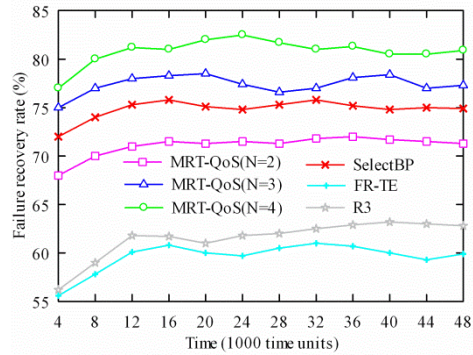
We first evaluate the link failure recovery rate and runtime of the aforementioned four BP-based algorithms. The key observations from our simulations are summarized as follows.

**Figs. 3** and **4** show the link failure recovery rate of the four failure recovery algorithms under low and high time-sensitive requirements, respectively. For the MRT-QoS algorithm, the amounts of BP are set as 2, 3 and 4. It can be seen that the link failure recovery rate increases as  $N$  increases, and upgrades approximately by 17.5% under the high time-sensitive requirement compared to low requirement. Therefore,  $N$  can be set with a larger value. But too many BPs can increase the router load and lead to poor practicability. Moreover, as there exists a strict limit to the maximum  $N$ , we set the value of  $N$  as 4. Note that  $N$  is set with the given network topology, but perhaps it is more appropriate with another  $N$  in another network topology. It can be seen that MRT-QoS generates the largest link failure recovery rate, which upgrades failure recovery rate by 35.1% under low time-sensitive requirements, and 48.0% under high time-sensitive requirements compared to the other three algorithms. The failure recovery rate of the MRT-QoS algorithm declines by 2.8% under high time-sensitive requirement compared to the low requirement. But the failure recovery rate of the other three algorithms declines by 19.9%. It shows that the enhancement of time-sensitive requirements has a low influence on the MRT-QoS algorithm. The main reason for this is that the MRT-QoS algorithm adopts the probabilistically correlated failure model and only chooses a reliable BP. Moreover, it gives priority to protecting the links that fail easily and bear heavy traffic. Without considering the delay constraint, the failure recovery effect under high time-sensitive requirements for SelectBP, FR-TE, and R3 algorithms are poor. FR-TE and R3 algorithms do

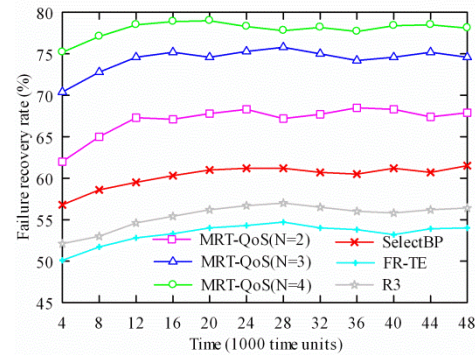


not consider the reliability of backup path, which may result in choosing failed backup paths, and thus their failure recovery rate is low.

**Table 2** shows the average runtime of the four algorithms. It can be seen that the runtimes of FR-TE and R3 are larger than the runtime of MRT-QoS and SelectBP, which means more time needed for configuring the protection resources in FR-TE and R3. The main reason is that MRT-QoS and SelectBP are multiple rounds of heuristic algorithms, which can greatly reduce the calculating time. However, FR-TE and R3 adopt the traditional linear programming method, so the problem-solving time increases exponentially with the increase of the problem size.



**Fig. 3.** Failure recovery rate with service of low time-sensitive requirement



**Fig. 4.** Failure recovery rate with service of high time-sensitive requirement

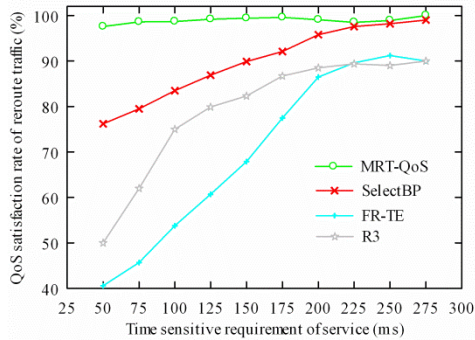
**Table 2.** Runtime comparison

Notation	MRT-QoS	SelectBP	FR-TE	R3
Runtime (s)	58.8	68.1	389.9	128.7

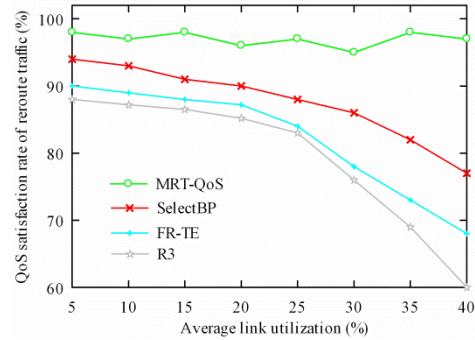
### 6.2.2 QoS satisfaction rate, traffic discarding rate, and link overload

First, we compare the QoS satisfaction rate of rerouting traffic under service of different time-sensitive requirements and different average link utilizations, respectively. QoS satisfaction rate is the proportion of the rerouting traffic that meets the QoS requirements on all the rerouting traffic. The time-sensitive requirements range from 50 to 275 ms, which is consistent with the delay requirements of most services in a real network. Since the link utilization of wireless providers is usually within 40%, we compare the performance under the link utilization from 5% to 40%, which corresponds to the amounts of CBR data flow from 20 to 160.

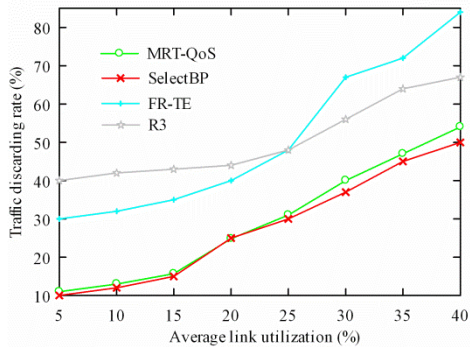
**Figs. 5** and **6** show that the QoS satisfaction rate of MRT-QoS always maintains the level above 95% and is not influenced by the variations of time-sensitive requirements and link utilization. However, the QoS satisfaction rate of rerouting traffic for SelectBP, FR-TE, and R3 rapidly declines with the enhancement of time-sensitive requirements and increase of link utilization, where the QoS satisfaction rate of FR-TE declines by 54.8% with the time-sensitive requirement of 50 ms compared to 275 ms. The main reason is that the MRT-QoS algorithm takes the QoS requirements into consideration but the other three algorithms do not consider the QoS constraints. Similarly, the available bandwidth resources decline as link utilization increases, resulting in a decline in the rerouting ability of SelectBP, FR-TE, and R3.



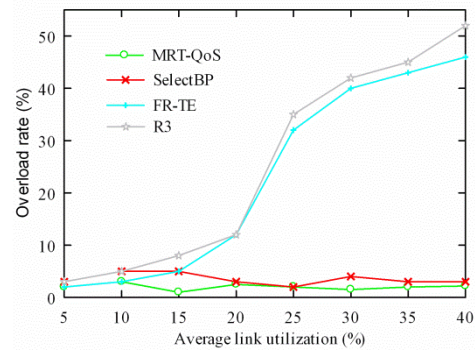
**Fig. 5.** QoS satisfaction rate of rerouting traffic under service of different time-sensitive requirements in stable state



**Fig. 6.** QoS satisfaction rate of rerouting traffic under different average link utilizations in stable state



**Fig. 7.** Traffic discarding rate under different average link utilizations in stable state



**Fig. 8.** Overload rate under different average link utilizations in stable state

Now we compare the traffic discarding rate under different average link utilizations.

**Figs. 7** shows that MRT-QoS and SelectBP outperform the other two BP-based algorithms. The traffic discarding rate of the four algorithms increases with increase of link utilization. However, the traffic discarding rate of FR-TE and R3 are approximately 3 times higher than that of MRT-QoS and SelectBP. We can also see that the traffic discarding rate of MRT-QoS is slightly higher than that of SelectBP. The main reasons are that FR-TE and R3 do not consider the link reliability and bandwidth constraint, so that they have a higher traffic discarding rate, while MRT-QoS considers the delay constraint, so the traffic discarding rate is slightly higher than SelectBP. Besides, a slight increase of the link utilization will make rerouting more difficult. For example, the available bandwidth is 4 times the load when the link utilization is 20%, but the available bandwidth only equals its load when the link utilization is 50%.

Third, we compare the link overload rate under different average link utilizations.

**Fig. 8** shows that the link overload rates of MRT-QoS and SelectBP always maintain at the level below 5%, which means that their overload rates have no correlation with link utilization. However, large-scale links overload with the increase of link utilization for FR-TE and R3. Moreover, when the link utilization is 40%, the link overload rate of FR-TE and R3 exceed 45% of the capacity, which greatly decreases the link failure recovery performance. The main reason is that MRT-QoS and SelectBP only adopt the paths that have available bandwidth resources to recover failed links, and strictly control the rerouting traffic on each BP, which avoids overloading as far as possible.

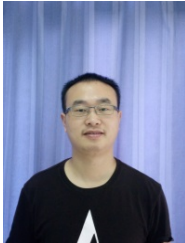
## 7. Conclusion

Link failure recovery problem is one of the main challenges in network resilience. In this paper, we developed a probabilistically correlated failure model on the basis of multiple BPs. With this, we acquired the quantified impact that the failed link is on the reliability of backup path. Moreover, we built an MILP model for the failure recovery problem and designed a novel algorithm called MRT-QoS, which is a QoS-aware approach for maximizing rerouting traffic in IP networks. Simulation results show that the proposed algorithm outperforms the existing BP-based failure recovery algorithms in terms of link failure recovery rate, QoS satisfaction rate of rerouting traffic, traffic discarding rate, link overload rate, and runtime. As links may not fail simultaneously, the capacity of the BPs can be shared to reroute multiple failed links at the same time. In future work, we will address the capacity sharing allocation problem of BPs to optimize resource utilization.

## References

- [1] G. Wellbrock and T. Xia, "How will optical transport deal with future network traffic growth?" in *Proc. of 2014 The European Conference on Optical Communication (ECOC)*, pp. 21-25, Sept. 21-25, 2014. [Article \(CrossRef Link\)](#).
- [2] D. Turner, K. Levchenko and A. Snoeren, "California fault lines: understanding the causes and impact of network failures," in *Proc. of ACM SIGCOMM*, pp. 315-326, Aug. 30 –Sept. 3, 2010. [Article \(CrossRef Link\)](#).
- [3] Y. Wang, Z. Wang and L. Zhang, "A failure recovery method for routing system based on structured backup subgraph," *Journal of Electronics and Information Technology*, vol. 35, no. 9, pp. 2254-2260, Sept. 2013. [Article \(CrossRef Link\)](#).
- [4] B. Yang, J. Liu and S. Shenker, "Keep forwarding: Towards k-link failure resilient routing," in *Proc. of INFOCOM*, pp. 1617-1625, Apr. 27 –May 2, 2014. [Article \(CrossRef Link\)](#).
- [5] Y. Wang, H. Wang and A. Mahimkar, "R3: resilient routing reconfiguration," in *Proc. of ACM SIGCOMM*, pp. 291-302, Aug. 30 –Sept. 3, 2010. [Article \(CrossRef Link\)](#).
- [6] M. Suchara, D. Xu and R. Doverspike, "Network architecture for joint failure recovery and traffic engineering," in *Proc. of ACM SIGMETRICS*, pp. 97-108, Jun. 4 – 11, 2011. [Article \(CrossRef Link\)](#).
- [7] R. Banner and A. Orda, "Designing low-capacity backup networks for fast restoration," in *Proc. of INFOCOM*, pp. 1-9, Mar. 14-19, 2010. [Article \(CrossRef Link\)](#).
- [8] H. Ngo and M. Kim, "MRFR-Multipath-based routing protocol with fast-recovery of failures on MANETs," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 12, pp. 3081-3099, Feb., 2012. [Article \(CrossRef Link\)](#).
- [9] Q. Zheng, G. Cao and F. Thomas, "Cross-layer approach for minimizing routing disruption in IP networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1659-1669, Jul., 2014. [Article \(CrossRef Link\)](#).
- [10] Q. Zheng, J. Zhao and G. Cao, "A cross-layer approach for IP network protection," in *Proc. of Dependable Systems and Networks(DSN)*, pp. 1-12, Jun. 25-28, 2012. [Article \(CrossRef Link\)](#).
- [11] Q. Zheng, G. Cao and T. Porta, "Optimal recovery from large-scale failures in IP networks," in *Proc. of IEEE ICDCS*, pp. 295-304, Jun. 18-21, 2012. [Article \(CrossRef Link\)](#).
- [12] S. Misra, G. Xue L and D. Yang, "Polynomial time approximations for multi-path routing with bandwidth and delay constrains," in *Proc. of INFOCOM*, pp. 558-566, Apr. 19-25, 2009. [Article \(CrossRef Link\)](#).
- [13] J. Tajne and V. Gulhane, "Multipath node-disjoint routing protocol to minimize end to end delay and routing overhead for MANETs," *International Journal of Engineering Research and Applications*, vol. 3, no. 4, pp. 1691-1698, Apr. 2013.

- [14] J. Liao, S. Tian, J. Wang, T. Li and Q. Qi, "Load-balanced one-hop overlay multipath routing with path diversity," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 2, pp. 443-461, Feb., 2014. [Article \(CrossRef Link\)](#).
- [15] G. Teresa, Miguel S and C. Jose, "Two heuristics for calculating a shared risk link group disjoint set of paths of min-sum cost," *Journal of Network and System Management*, vol. 37, no. 10, pp. 332-338, Oct. 2014. [Article \(CrossRef Link\)](#).
- [16] M. Johnston, H. Lee and E. Modiano, "A robust optimization approach to backup network design with random failures," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1216 - 1228, Apr. 2011. [Article \(CrossRef Link\)](#).
- [17] L. Zhou and J .Wang, "Path nodes-driven least-cost shortest path tree algorithm," *Journal of Computer Research and Development*, vol. 48, no. 5, pp. 721-728, May 2011.
- [18] K. Imura and T. Yoshihiro, "Reactive load balancing during failure state in IP fast reroute schemes," *Journal of Information Processing*, vol. 22, no. 3, pp. 527-535, Mar. 2014. [Article \(CrossRef Link\)](#).



**Wenyan Cui** is a Ph.D. candidate in Communication and Information System from Air Force Engineering University, China. His research interests include network invulnerability, data survivability and cyberspace security.



**Xiangru Meng** is a professor in Communication and Information System from Air Force Engineering University, China. His research interests include next generation Internet and broadband communication network.



**Huanhuan Yang** is a Ph.D. candidate in Communication and Information System from Air Force Engineering University and Tsinghua University, China. His research interests include network survivability and network management.



**Qiaoyan Kang** is an adjunct professor in Communication and Information System from Air Force Engineering University, China. Her research interests include network survivability and congestion control.



**Zhiyuan Zhao** is a Ph.D. candidate in Communication and Information System from Air Force Engineering University, China. His research interests include network invulnerability and network virtualization.