Received 29 October 2023, accepted 18 November 2023, date of publication 23 November 2023, date of current version 30 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3335988

RESEARCH ARTICLE

Fast Reroute Algorithms for Satellite Network With Segment Routing

XUN CHEN[1], ZHENGJIAN CHEN[2], XIAOLEI CHANG[3], TIAN JI[3,4], ZHENZHOU WU[4], AND CHENXI LI[4]
[1]Shenzhen Polytechnic University, Shenzhen 518000, China
[2]Shenzhen Energy Group Company Ltd., Shenzhen 518000, China
[3]Tsinghua University, Beijing 100000, China
[4]Research Institute of Tsinghua University in Shenzhen, Shenzhen 518000, China

Corresponding author: Zhenzhou Wu (wuzhenzhou@qzcloud.com)

This work was supported in part by the Shenzhen Science and Technology Program under Grant SGDX20190917160803729.

ABSTRACT Satellite networks, which have wide coverage and high throughput, are more and more important for the Internet. Low earth orbit (LEO) satellites have been more widely used than other orbiting satellites because they have lower unidirectional links (UDL) transmission delay. However, due to the variability and instability of satellite links, leading to relatively higher UDL transmission delay and more link failures for satellite networks. Thus, fast reroute (FRR) schemes, which can bypass the failure links and improve network performance, are badly needed in satellite networks. However, FRR need relatively large computing resources while satellites have limited resources. Thus, traditional routing technologies cannot be directly applied to satellite network due to frequent topology changes and poor computing performance. To meet the challenges, we propose a hybrid FRR schemes for satellite networks that combines the centralized computing with distributed segment routing (SR). With the scheme, satellites with relatively more computing resources can pre-compute the reroute paths, and distribute the routing rules to satellites that have more forwarding resources. We formulate the problem, propose a classification algorithm that can distinguish satellites according to their computing and forwarding resources. Additionally, we also propose a real-time backup path maintenance algorithm in the hybrid rerouting scheme. Finally, we conduct comprehensive simulations to evaluate the performance of the proposed algorithms, and the results show that under extreme conditions, the proposed LFR algorithm is 30.9% better than traditional algorithms in storage resource utilization; and the proposed LFA+ algorithm is 74.3% better than traditional algorithms in update time comparison.

INDEX TERMS Satellite routing, fast reroute, segment routing, centralized routing, hybrid routing.

I. INTRODUCTION

With the fast development of the Internet, satellite networks become a promising technology that can provide more room for the future. For example, satellite networks can cover more places that traditional networks cannot, such as oceans and foresst. With nearly 40% of the world's population still having no access to the Internet, satellite networks are not subject to terrain constraints and make the Internet easier to access. This could be particularly useful in some important applications, e.g., emergency communication under natural disasters, and ocean network communication for oil platforms in the ocean.

Depending on the altitude from the ground, satellites can be classified into LEO, medium earth orbit satellites (MEO), and geosynchronous earth orbit satellites (GEO). GEO has a higher altitude and wider coverage; however, the UDL transmission delay of GEO satellites is too high to satisfy many delay-sensitive services. Recently, LEO satellites have brought new opportunities for satellite networks and they have been more widely used because of the lower UDL transmission delay and larger bandwidth [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Vittorio Camarchia.

Although satellite networks, especially LEO satellites, have wide coverage and high throughput [2], current routing technologies cannot be directly applied without changing satellite networks [3]. The movements of satellites in orbit lead to frequent changes in link connection and network topology. Although many new satellite routing schemes have been designed to improve network performance based on movement patterns or periodicity [4], routing interruptions still occur now and then due to the higher failure rate of satellite communication links within the current satellite network.

FRR is seen as a promising technology to tackle communication interruption by pre-computing an alternate path towards the destination [5]. Recently, combined with Software Defined Networks (SDN) [6], FRR can be applied in larger networks where the controller can collect network information and compute the re-route rules.

Satellite routing can be divided into centralized routing, distributed routing, and hybrid routing [7], [8], [9]. Among them, hybrid routing combines the advantages of centralized and distributed routing by classifying nodes into two types: 1) computing nodes that have more computing resources; 2) forwarding nodes that have more forwarding resources. In this way, hybrid routing can achieve the best of two worlds, where computing nodes can compute the alternate paths to protect the original path, and forwarding nodes can forward packets according to pre-computed rules.

In this paper, to address the challenges faced by satellite networks, such as link failures, limited computing resources, and large transmission delays, we propose a FRR scheme in satellite networks, by integrating segment routing into hybrid routing. In the new fast re-routing scheme, hybrid routing is used to realize dynamic backup path updates when satellite links change, thus ensuring the timeliness and effectiveness of path protection. Besides, we propose a solution for fast reroute in the LEO satellite network based on segment routing, and design a backup path maintenance algorithm.

Finally, we conduct comprehensive simulations using different network topologies with different sizes and degrees. The results show that the proposed fast re-routing scheme and algorithm outperform traditional algorithms, especially under link failures. Besides, the computing overheads brought by the new routing scheme are acceptable for most computing satellite nodes.

The main contributions of this paper are as follows:

1. We proposed a FRR scheme for satellite networks that integrates segment routing in a centralized routing mode, addressing challenges such as high failure rates, low stability, and large delays in transmission.
2. We introduced a division of relay satellites into computing and forwarding satellites, optimizing resource utilization, and enabling backup path generation without relying on ground control during link failures.
3. We developed algorithms for efficient node partitioning based on relay satellites' locations and resources, as well as a backup path maintenance algorithm for independent calculation of paths in hybrid routing, improving performance and reliability while considering limited computing resources.

The structure of this paper is as follows: Section II introduces the related research work, including performance evaluation of fast reroute and fast reroute in segmented routing. Section III and Section IV study the protection scheme in centralized routing mode and the backup path maintenance algorithm in hybrid routing mode respectively, and the performance of the algorithm is verified by experiments. Finally, Section V summarizes the paper.

II. RELATED WORK

A. PERFORMANCE EVALUATION OF FAST REROUTE

FRR holds significant application prospects in the field of image and multimedia [10], [11], including image transmission and processing, video streaming and conferencing, multimedia content distribution and storage, as well as virtual reality and augmented reality applications. Rapid failure recovery is an important topic in the research field [12], [13], and relies on the preparation of a backup plan. However, due to loop problems, storage space limitations, etc., protection based on backup paths cannot cover all links. That is to say, the actual efficiency of fast reroute depends on the protection coverage, which is also the focus of early research. Xu et al. provided a tunnel-based IPFast reroute algorithm, which uses minimum weight protection trees to provide higher protection coverage [14], while Enyedi et al. defined the low-point problem by using the maximum redundancy tree [15], and realized fast forwarding in the maximum redundancy tree, ensuring 100% recovery of a single fault in a network with 2 connections. These algorithms provide higher protection coverage based on different data structures, that is to say, earlier studies have achieved 100% link protection based on backup paths.

Early research on fast reroute mainly focused on how to achieve a higher protection rate. After these studies achieved good results, the focus of research has gradually shifted to a more in-depth exploration of the performance of fast reroute. In some networks, backup paths can be calculated in advance for future use upon link failure. However, when protecting multiple links at the same time, it may not be possible to calculate and store the backup paths of multiple link combinations in advance (for addressing multiple link failures) as that would consume a large amount of storage space without obtaining enough gains in return. Therefore, a better way to solve multi-link failure is to update the backup path immediately after the failure occurs, so as to prepare a new backup path before the next failure occurs. This method has higher requirements on the efficiency of path find calculation. If link failures or state changes occur frequently, new backup paths need to be calculated in real-time. In order to improve the efficiency of this calculation, Jarry et al. proposed a series of algorithms to efficiently find the initial paths and backup paths in fast reroute, including the Dijkstra algorithm with memory [16] that showed good performance in terms of path calculation efficiency.

In terms of performance analysis of fast reroute, researchers have been trying to find a set of unified and standardized evaluation criteria, without which it may be impossible to study various algorithms and optimization methods. Specifically, Yang et al. provided a valuable reference for the evaluation of fast reroute by comprehensively converging convergence time, message cost, connection setup delay and other factors to determine the performance and efficiency of fast reroute [17]. There are also many researchers who tried to design a standard fast reroute architecture system to improve its deployment compatibility with different networks. For example, Andersen et al. defined a new RON architecture [18] that enables distributed Internet applications to detect and recover from path outages and performance degradation within seconds. Their research initially caused a lot of repercussions, including subsequent research and development of applications. However, with the emergence of subsequent research results and newer network technologies, this architecture has been abandoned.

B. FAST REROUTE AND SGEMENT ROUTING FOR SATELLITE NETWORK

Segment routing is a technology born from software-defined networking. It features low latency and can be used to achieve much-simplified network protocols and better traffic engineering control. The network application layer above the control plane of segment routing can be used to contain applications designed to control the lower layer.

In today's traffic-first era, the emergence of segment routing allows us to achieve a high degree of network control and simple operation simultaneously. There have been many studies on the application of fast reroute in segment routing. Firstly, in 2016, the patent applied by Filsfils from the United States proposed a device and method [19] for FRR of local segment routing traffic, which can be used to achieve fast local fault detection and traffic recovery. As for dynamic traffic recovery, Giorgetti proposed a procedure for dynamically recovering traffic interruption caused by a single fault in a segment routing network [20]. They found that, when using configured labels to restore traffic, in most cases, no more than two labels were required to successfully restore valid traffic.

In segment routing, after the device detects a link or node failure, it needs to send relevant information to the controller to recalculate the route. This process consumes a certain amount of response time and can be optimized for faster error detection and rerouting. Xhonneux et al. deployed a flexible detection method on Linux routers and servers using the extended Berkeley packet filter (eBPF) [21] and segment routing [22], which is better than Bidirectional Forwarding Detection (BFD) in terms of performance and detection speed.

For fast reroute in case of multi-link failure in segment routing, Foersterd et al. proposed a fast reroute algorithm based on hypercube topology in segment routing [23]. Its solution time can reach the level of the elastic polynomial. This method preserves the original path and does not require packets to carry failure-related information. Schüller et al. proposed an optimization model to make it easier to find a single-segment routing configuration with multiple benefits [24].

FRR and SR are widely studied in satellite networks [25]. Firstly, many previous studies use FRR as an important mechanism to improve the reliability of satellite communication [26]. Secondly, segment routing is usually combined with SDN to improve the flexibility of satellite routing [27].

In conclusion, previous research in fast reroute has primarily focused on achieving higher protection coverage, followed by more recent efforts to delve deeper into the performance evaluation and optimization of fast reroute mechanisms. Researchers have also strived to establish standardized evaluation criteria and develop compatible architectural systems. These insights collectively lay the foundation for the study and development of fast reroute solutions in various networking scenarios.

However, the previous works either focus on centralized satellite routing mechanism, which consumes lots of computing resources of the centralized controller and faces single-point failure; or distributed satellite routing mechanism, which cannot provide enough control flexibility for network management. In this paper, we adopt hybrid routing that combines the advantages of computing and forwarding nodes and achieve the best of two worlds.

III. RESEARCH ON PROTECTION SCHEME IN CENTRALIZED ROUTING MODE

A. SCHEME AND ARCHITECTURE DESIGN

Centralized routing requires a central node which can be a ground control center or a satellite. The central node is responsible for collecting link state information, calculating routes, and storing routing tables, while other satellite nodes cannot calculate routes and can only forward data by using the routing tables delivered by the central node. This process is in line with the paradigm of software-defined networking. When it is used in combination with segment routing and multi-protocol label switching technology, the central node can serve as a controller to issue paths for all packets entering the domain, and other satellite nodes perform transparent transmission after checking the label stack in the header of the received data packet. When the network topology changes, fast reroute enables the central node to collect in advance global network link information and calculate the backup path so that it can directly deploy the backup path and update the routing table. Segment routing can reduce overhead and delay by simplifying the routing protocol and eliminating the need for flow table updates between satellites by delivering the routes. The difficulty in implementing

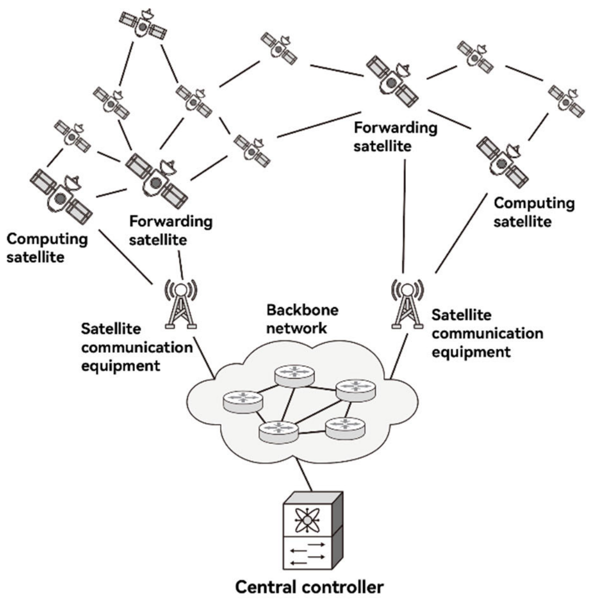


FIGURE 1. Architecture of the centralized routing link protection scheme.

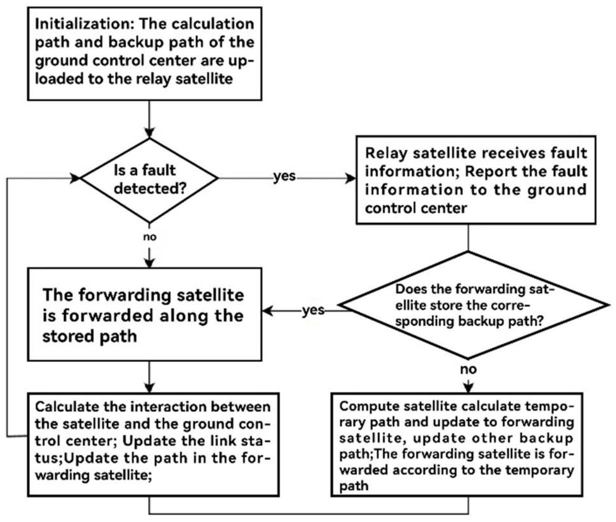

Initialization: The calculation path and backup path of the ground control center are uploaded to the relay satellite
Is a fault detected?
yes
Relay satellite receives fault information; Report the fault information to the ground control center
no
The forwarding satellite is forwarded along the stored path
yes
Does the forwarding satellite store the corresponding backup path?
no
Calculate the interaction between the satellite and the ground control center; Update the link status;Update the path in the forwarding satellite;
Compute satellite calculate temporary path and update to forwarding satellite, update other backup path;The forwarding satellite is forwarded according to the temporary path

FIGURE 2. Flowchart of the centralized routing link protection scheme.

fast reroute is that the ground device cannot detect satellite link changes in time, and there is a large delay in the transmission of the backup path. Therefore, in this paper, we propose a segment routing-based fast reroute (SS-FFR) scheme that allows data packets to be forwarded through the backup path promptly when a centralized routing link fails.

Some studies have proposed the use of relay satellites for long-distance transmission [28], but in this paper, the relay satellite is used as the relay for the ground center to control the satellite network, which solves the problem of large delays in transmission between the ground and satellite in segment routing. Due to the limited computing resources of relay satellites, some studies proposed to divide the satellite nodes in the path into computing satellites and forwarding satellites [29], thus saving a lot of computing resources. In the scheme described in this paper, relay satellites serve as control nodes in segment routing and are also divided into computing satellites and forwarding satellites to improve their processing capacity and efficiency.

As shown in Fig. 1, in this study, some stable relay satellites are selected to serve as the satellite control plane of the satellite network, while the ground control center remains the main control plan. These relay satellites are divided into different categories (including relay satellites and computing satellites) with different functions and permissions. Under normal circumstances, the forwarding satellite is responsible for storing and delivering the forwarding path, and the computing satellite is responsible for collecting nearby link state information and calculating the path, exchanging data with the ground control center regularly, and updating the routing table for the forwarding satellite after any changes in the link state. When a protected link or node fails, the forwarding satellite can directly switch to the backup path and deliver the path, while the calculation satellite calculates a new backup path.

As shown in Fig. 2, through the division of relay satellites, relay satellites become the central node in centralized routing. This reduces the workload of a single satellite and ensures that the backup route can be delivered in a timely and stable manner after a failure occurs. For the ground control center which has abundant computing resources but faces the problem of high transmission delay in satellite-to-ground link (UDL) communication, the existence of computing satellites ensures that the backup path can be delivered in time after a failure. Additionally, the path stored in the forwarding satellite can be dynamically updated, effectively enabling traffic engineering control and load balancing in segment routing. The forwarding satellites only need to label the data packets entering the domain and forward them. When a failure is detected, the forwarding satellite (as the controller) will switch to the backup path for forwarding. If the backup path is not good due to link state changes, the forwarding satellite will consult the computing satellite and wait for the updated backup path. By using satellite-to-satellite link (ISL) communication instead of time-consuming satellite-to-ground communication, we can more fully utilize the computing resources of relay satellites while greatly reducing latency.

The SS-FFR scheme proposed in this paper increases the number of relay satellites in each domain, thereby reducing the workload of each relay satellite as a controller. However, since the satellites will move frequently, and the network topology is very unstable, we have to select suitable relay satellites and partition domains appropriately for SS-FFR.

B. DEFINITION OF SYMBOLS

In view of the instability of the satellite network topology, it is necessary to select suitable relay satellites and appropriately partition the cross-orbit SR domain. In this study, the satellite

TABLE 1. Definitions of SS-FFR symbols.

Symbol	Definition
G(V，E，C)	Satellite topology graph G, which contains V nodes and E undirected edges, C is the distance set of ISL.
u	Total number of orbits
p	Number of orbits passing through polar regions
q	Number of orbits not passing through polar regions
k_i	Number of satellites on orbit
age_i	Working time of the satellite since the last failure
avg_age	Mean interval between satellite failures
UDL_i	UDL distance/length of the satellite
Δ_i	Weights assigned to candidate satellites
cpu_i	Available computing resources of satellites
mem_i	Available storage resources of satellites
α	Uptime weights of relay satellites
β	UDL distance weights of relay satellites
μ	Storage resource weights of forwarding satellites
λ	computing resources weights of forwarding satellites

topology is defined as an undirected connected graph G (V, E, C), and the number of orbits is also defined. In addition, since the ISLs in the polar regions of the earth will be turned off, satellites above these regions will not be able to transmit data. Therefore, the satellites that pass above the earth's poles are unstable. When partitioning regions, care should be taken to avoid having all unstable satellites in a region.

Since the failure of relay satellites can be very problematic, it is necessary to select some younger satellites that do not pass through the polar regions to serve as relay satellites, which helps reduce the possibility of failure. If the normal working time of a satellite from its last failure is referred to as its age (age_i), and the average interval between two consecutive failures of the satellite is referred to as its average age (avg_age), then the remaining average normal working time of the satellite is:

$$work\_time_i = avg\_age - age_i \tag{1}$$

The candidate set for selecting relay satellites should contain n satellites with the longest average normal working time. Since the cooperation of two relay satellites (one computing satellite and one forwarding satellite) is required, it is suitable to choose two satellites that are closer to each other and closer to the ground (to reduce the transmission delay of periodic updates). Firstly, each satellite in the candidate set is assigned a weight Δi which is calculated as follows:

$$\Delta_i = \alpha \times work\_time_i + \beta \times UDL_i \tag{2}$$

After sorting in descending order according to the weight, t pairs of satellites are selected from the top to serve as the relay satellites f and r of t domains, while ensuring that:

$$max \frac{\Delta_f + \Delta_r + \mu\left(mem_f + mem_r\right) + \upsilon\left(cpu_f + cpu_r\right)}{\sqrt{\left(x_f - x_r\right)^2 + \left(y_f - y_r\right)^2}} f, \quad r \in p \tag{3}$$

Then, when the remaining satellites in the collection are assigned to t domains, it is necessary to dynamically calculate the average distance avg_disi from the forwarding satellite to other nodes after a new node is added to each domain, ensuring that the selection of each node field meets the following conditions:

$$\min\ avg\_dis_{ii\in t} \tag{4}$$

$$s.t.\ mem_i > 0 \tag{5}$$

$$cpu_i > 0 \tag{6}$$

If all resources in the domain have been utilized, it's required to replace the node(s) with the longest average distance when computing and storage resources are satisfied and add the replaced node(s) to a new domain that meets the conditions.

C. ALGORITHM DESIGN

The selection of relay satellites can be done simply by using the heap sorting method, and the focus of this study is on the algorithm for the domain partition of ordinary nodes. Considering the complexity of node domain selection and node replacement process, this study proposes a planning algorithm for domain partition that aims to achieve a globally optimal solution while reducing the complexity of the algorithm.

In this study, according to the storage space that a unit in the domain may occupy in the forwarding node (including the initial forwarding path, the backup path, etc.), and the computing resources that it may occupy in the computing node (such as those for calculation of paths and backup paths, etc.), the available computing resources and storage resources are converted into the number of nodes that can be added currently. As such, each domain can accommodate a fixed number of nodes according to the working capabilities of the relay satellites. The average distance between satellites to be assigned to each domain is calculated using the forwarding node as a reference. Because the forwarding node is responsible for forwarding within the domain, the initial average distance should be the distance between the satellite and the forwarding satellite. According to the node partition algorithm (DoD) outlined in Algorithm 1, steps 1 and 2 are for assigning all unassigned nodes to nearby relay nodes based on the shortest distance principle. If there is only one relay node that meets the conditions, the node is directly assigned to the domain of that relay node. If two or more relay nodes maintain a certain node, its status is temporarily marked, and all nodes are divided into two groups (firmly allocated nodes and temporarily allocated nodes, the domain for assignment of the latter is already marked) according to the distance (from short to long). Then, step 3 is for deciding where to allocate the nodes, which is done based on their distance from the relay node, with the state of the available domains being constantly updated based on capacity status.

In the complexity analysis of the DOD algorithm, len is the number of nodes to be allocated, t is the number of

TABLE 2. **Algorithm 1: Node partition algorithm (DOD).**

```
Algorithm 1:   Node partition algorithm (DOD)
01  Input: t domains Di(fi, listi, avg_disi, capacityi), the list of
    nodes to be allocated S and the length len
02  Output: t domains Di(fi, listi, avg_disi, memi, cpui) for
    containing all nodes.
03  Initialize A[len][t], indicating whether each node is
    allocated to t, 1 means temporary allocation, 0 means
    unallocated, 2 means confirmed allocation;
04  //Step 1: Find the pair with the smallest average distance in
    each domain while ensuring capacity of the domain is not
    exceeded
05  ForEach i ∈t do
06  select for domain i capacity_i nodes that are closest to f_i and
    A[j][i]=0;
07  Set the A corresponding to these nodes and domain i to 1;
08  End For
09  // Step 2: Process temporarily assigned nodes, if there are
    unassigned nodes, return to Step 1
10  ForEach i ∈len do
11  If node i is only temporarily assigned to domain j, then
    A[i][j]=2;
12  Update D_j, if capacity_j=0 then update A[k][j]=0, k∈len;
13  End For
14      Determine whether adjustment is needed, and if so,
    assign a node to the nearest domain and update it.
15  End While
16  // Step 3: Process the remaining temporarily allocated
    nodes
17  ForEach i∈len and A[i][k]=1 do
18  Calculate the avg_dis of nodes after adding them to the
    domain, assign them to the domain j with the smallest
    weight, and update A[i][k]=0,k∈len;
19  Update D_j, if capacity_j=0 then update A[k][j]=0， k∈len;
20  End For
```

available domains, and len is greater than t. The algorithm needs to include the A array, so its space complexity is O(len×t). In terms of time complexity, in the worst case of the number of while loops, only one node to be allocated can be addressed in each loop, so a total of len loops are required. Within the loop, the complexity of step 1 is O(t×len), that of step 2 is O(len2), meaning that the complexity of the while loop is O(len3), that of step 3 is O(len2). Therefore, the time complexity of the DOD algorithm is O(len3+len2), which is still within the polynomial.

D. EXPERIMENTAL TEST

Many previous works have evaluated different FRR algorithms in satellite routing through many performance metrics, such as link failure rate [31], computing overhead [32], and path stretch ratio [33]. In this paper, we choose mean distance, storage resource utilization, computing resource utilization, etc. as the performance metrics, because these metrics can indicate the feasibility and scalability of a newly proposed re-route algorithm.

The experiment of this study is to run the DOD algorithm on several randomly generated discrete nodes and compare the results with those of the greedy algorithm (SDF) based on the average shortest distance and a local fast reroute (LFR) [33] algorithm with flow aggregation. This experiment evaluates the performance of the DOD from three aspects (average distance, storage resource utilization, and computing resource utilization). In the experiment, nodes of different scales were randomly placed into 12 domains on a plane with a size of 100 units, each of the two algorithms was run 10 times for nodes of each scale and the mean averages of the results were taken.

1) COMPARATIVE ANALYSIS BASED ON THE AVERAGE DISTANCE

Fig. 3 shows the average distance change of the three algorithms after the node size increases and the nodes are divided. The more nodes there are in a fixed plane, the smaller the average distance between nodes will be. That is to say, we can see that the overall mean distance decreases as the node size increases. Besides, overall, we can see that the mean distance of DOD is lower than that of the SDF algorithm and LFR algorithm. This proves that our DOD algorithm performs better than SDF and LFR in the mean distance. Although the SDF algorithm always chooses the closest region for a node, its problem in the case of the experiment is that, since the capacity of each region is limited, SDF is likely to cause the nodes allocated first to occupy resources that are more suitable for unallocated nodes, resulting in poor results. On the other hand, the DOD algorithm can comprehensively consider storage resources, computing resources, and average distance, thereby obtaining better results. Moreover, as the node density increases, the above-mentioned advantages of DOD will be further expanded. When 220 nodes are allocated, the average distance achieved by DOD is only 4, while that of SDF is 24 and LFR is 20. The gap between the two is quite large (in practical applications, it may mean a gap of thousands or even tens of thousands of kilometers).

2) COMPARATIVE ANALYSIS BASED ON RESOURCE UTILIZATION

In this study, resources are divided into storage resources and computing resources. The addition of each node will lengthen the corresponding storage path and backup path, increasing the topology scale and resource consumption of the computing path. Different satellites have different resources. Therefore, the relay satellites in each domain need to maintain proper storage resource utilization and computing resource utilization at the same time. Fig. 4 shows the comparison of storage resource utilization of each domain after scale increase and partition using the three algorithms. We can see that the overall storage resource utilization rate increases with the node size. Besides, we can see that the storage resource utilization of DOD is lower than that of the SDF algorithm and LFR algorithm. This proves that our DOD algorithm performs better than SDF and LFR in the storage resource utilization. In the best case, the performance of the DOD algorithm is 30.9% better than the SDF algorithm and

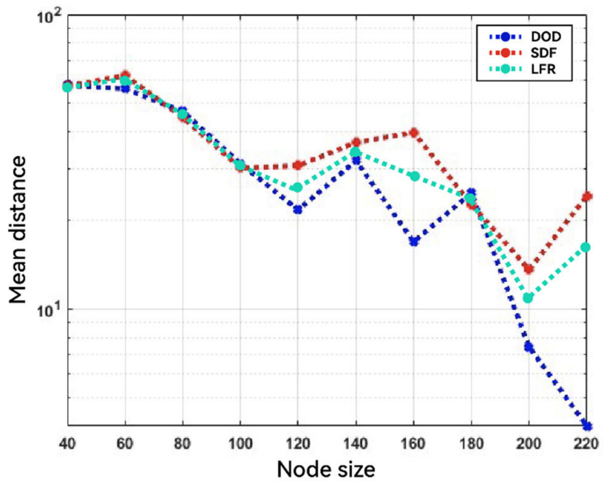


FIGURE 3. Comparison of mean distance.

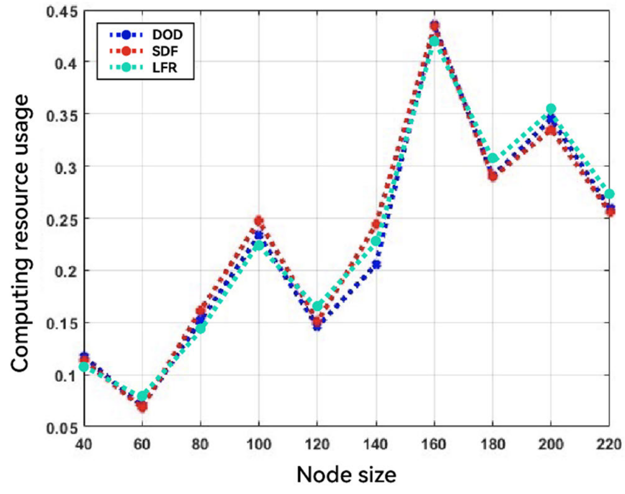


FIGURE 5. Comparison of storage resource utilization.

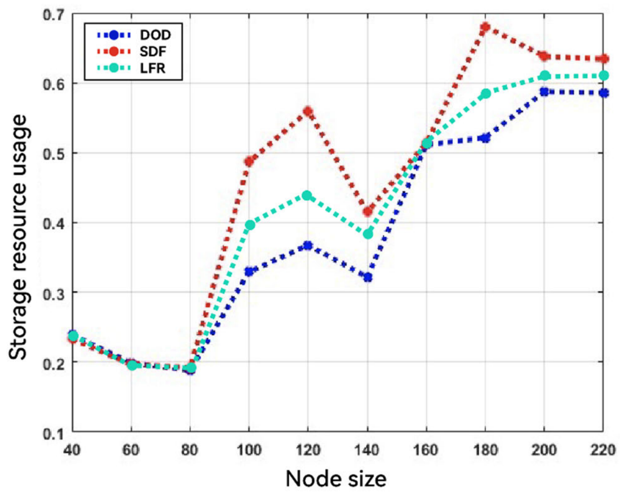


FIGURE 4. Comparison of computing resource utilization.

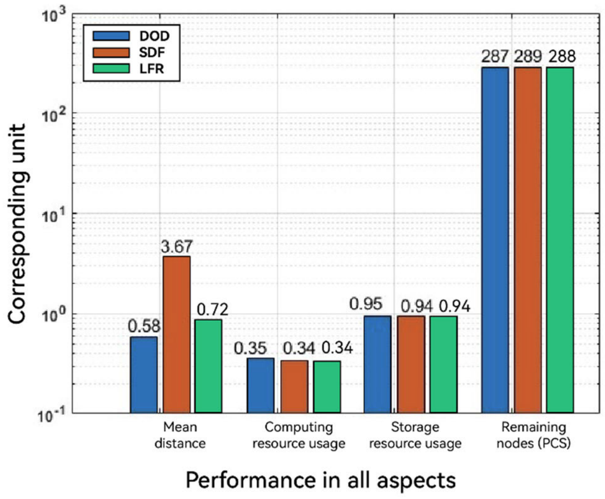


FIGURE 6. Comparison of high load performance.

13.6% better than the LFR algorithm when the node size reaches 120. The storage resource utilization of the DOD algorithm should be lower because it tends to allocate nodes to domains with sufficient resources. On the other hand, SDF is based on the principle of distance priority and does not consider the load balancing of storage resources, resulting in a rapid increase in storage resource utilization as the node size increases. Fig. 5 shows the comparison of computing resource utilization. Unlike the case of storage resources, computing resource use is reflected in the time takes for satellites to complete corresponding tasks in path calculations or node updates, etc. We can see that the overall calculating resource utilization rate increases with the node size. However, there is no big difference between the three algorithms in terms of computing resource utilization because that nodes have a small impact on computing resources and thus won't make a big difference in solving problems; on the other hand, it is rather difficult to compare the computing resource usage during the solving process (in this study, computing resource utilization is calculated based on the impact of computing resource usage on each node).

3) PERFORMANCE ANALYSIS OF HIGH LOAD NODES

The function of the two algorithms used in the experiment is to divide the nodes into different regions. If there are many nodes, the relay nodes in each region are likely to run at high load or even at full capacity. Due to the different solving methods of the algorithms, the number of nodes each area can contain is also different. This study includes high load performance comparison experiment that uses 500 random discrete nodes to compare the performance of the three algorithms in various aspects. Fig. 6 shows their performance in four aspects. From the number of remaining nodes, it can be seen that DOD can accommodate 2 more nodes than SDF and 1more node than LFR in high-load scenarios. That is to say, the DOD algorithm can absorb the nearest nodes more efficiently so that the average distance in the domain would be shorter, while the SDF algorithm and LFR algorithm are weaker in this regard and unable to effectively reduce the average distance in the domain. In terms of resource usage, DOD has no difference from SDF and LFR, and all three algorithms cannot receive more nodes due to insufficient storage resources. However, due to the absorption of more nodes, the resource utilization of DOD is 1% higher than that of SDF and LFR.

E. RESULTS ANALYSIS AND SUMMARY

This study set up a comparison experiment to compare DOD with the greedy selection algorithm and LFR algorithm. According to the analysis of experimental results, the node allocation based on DOD can achieve a shorter average distance in the domain, and the convergence speed of the average distance would increase with the increase of node density. The average computing resource utilization and storage resource utilization of the DOD are also lower, indicating that it can use the resources of each relay satellite more efficiently. In addition, DOD has a higher capacity to withstand high loads while absorbing more nodes and maintaining a shorter average distance. In summary, the DOD algorithm boasts fairly good performance.

IV. RESEARCH ON BACKUP PATH MAINTENANCE ALGORITHM IN HYBRID ROUTING MODE

A. DESCRIPTION OF BACKGROUND

In the hybrid routing mode, there are nodes in the forwarding path that can independently determine the next hop and nodes that are responsible for transparent forwarding. While having the advantages of centralized routing (high controllability and low resource consumption), it solves some problems of centralized routing, including the time delay in interaction with the central node, and a large amount of overhead caused by the signaling sending and receiving of the central node. This is why hybrid routing has become a research hotspot nowadays. Similarly, segment routing can also be implemented in a distributed manner, with the network being segmented through distributed controllers, and each controller independently calculating routes. In hybrid routing, there are only limited computing resources (independent satellites) that can be used to independently calculate the next hop in the path. As the routing decision makers in the path, independent satellites cannot complete multiple large-scale backup path calculations in real-time. Therefore, the key issue becomes how to better maintain the backup paths. In this paper, considering that independent satellites need to provide a backup path as soon as possible when a network link or node fails, we propose a backup path maintenance algorithm (BKM) that can cope with network link state changes, which can help reduce losses by quickly calculating a new backup path based on the existing backup path when the link changes.

Unlike the case of centralized routing, there are far more independent satellites in hybrid routing than relay satellites. Each of these independent satellites is only responsible for the path calculation of a small area. They collect nearby topology states and calculate paths, which in the case of fast reroute also include backup paths. All the backup paths calculated by these independent satellites in advance are stored locally so that when the protected link fails, the backup paths can be directly used without recalculation. If the link state in the backup path changes, it is necessary to find a new backup path, which requires a lot of calculations. If the paths of all protected links need to be recalculated, it will consume a lot of satellite computing resources and take a lot of time. To address this issue, this paper proposes a BKM algorithm that makes real-time adjustments according to link state changes in backup paths without re-computation.

The calculation of the backup path can be divided into the next-hop-based calculation and global calculation. The global calculation can guarantee the performance of the path (such as by using the shortest path length), but it needs to consume more computing resources and time and therefore is not suitable for satellite networks; in contrast, the next hop calculation can obtain effective results by using fewer computing resources, so it is more suitable for satellites with limited computing resources. The BKM algorithm proposed in this paper can help flexibly and correctly replace the backup path when the link state changes, and re-backup the failed link based on the next-hop mechanism, thereby avoiding large-scale path calculation, and reducing the workload of the satellite. As such, it can be deemed as a valid reference for solving the backup path-finding problem of satellite networks.

B. ALGORITHM DESIGN

This paper studies how to maintain the correctness of the backup paths after the initial paths and all backup paths have been calculated. To do that, we set an initial path set I and a backup path set B, with the former containing all the initial forwarding paths $dp_{s,d}$calculated by individual satellites based on the collected state information, the latter containing all backup paths of different nodes in each protected link for that link. As such, B includes several sets $B(e_{u,v})$, and each set includes all paths $bp_{s,d}$ corresponding to a protection link $e_{u,v}$.

As shown in Algorithm 2, backup paths are updated in real-time based on the calculated initial path set I, the backup path set B, and link status. When a link failure is detected, the backup path is updated to the initial path set (step 1), at which point normal transmission can already be resumed, then the backup path set is updated to prevent faulty links from being used as backup paths for other links, which would be invalid. Below is a simple example to illustrate this situation and the general process of the algorithm.

Fig. 7 shows the situation where link a fail (a is the backup path of b). If b fails, the path from node 1 to node 2 can be switched to the backup path 1-4-2. Therefore, when updating the backup path of protected link b, it is necessary to first determine whether the backup paths of nodes 1 and 2 at both

TABLE 3. **Backup path maintenance algorithm (BKM).**

Algorithm 2: Backup path maintenance algorithm (BKM)

```
     Input: initial path set I, backup path set B, faulty link eu,v
     Output: Updated I and B
01   ////Step 1: Update the initial path set and deploy the
     corresponding backup path to I
02   ForEach $dp_{i,j} \in I$ and $e_{u,v} \in dp_{i,j}$ do
03       IF $bp_{i,j}=\varnothing$, then node i to j is unreachable, $dp_{i,j}=\varnothing$;
04       ELSE $dp_{i,j}=bp_{i,j}$, $bp_{i,j} \in B(e_{u,v})$;
05       End IF
06   End For
07   Update $B=B/B(e_{u,v})$;
08   // Step 2: Based on the next hop principle, update B with
     fault eu, v through I
09   ForEach $B(e_{i,j}) \in B$ do
10       IF $e_{u,v} \in bp_{i,j}$ Then nodes i and j need to search again
     based on the next hop principle, same for //$bp_{j,i}$
11           Search path table I, record the search path with m
12           Found path $dp_{k,j}$ and $e_{i,j} \notin dp_{k,j}$ if not, it means that
     there is no backup path between i and j
13           $bp_{i,j}=m \cup dp_{k,j}$;    //deploy the new path from i to
     j toB($e_{i,j}$)
14   End IF
15   ForEach $bp_{s,d} \in B(e_{i,j})$且 $e_{u,v} \in bp_{s,d}$ do
16           IF $bp_{i,j} \neq \varnothing$  $bp_{s,d}=bp_{s,d}/path_{i,j} \cup bp_{i,j}$;
17       ELSE $bp_{s,d}=\varnothing$;
18   EndIF
19       End FOR
20   End For
21   End
```

ends of it contain faulty links. However, when a fails, it is necessary to first find a backup path from node 1 to node 2 (the same goes for the path from node 2 to node 1). Via the initial path set I, we can find that node 3 is the closest node to node 1. Since link b is protected, all paths including this link are not considered. For example, if the initial path from node 3 to node 2 is 3-1-2, and link b is included in it, then it is necessary to continue to find node 4 that is adjacent to node 3 and its path to node 2 (4-2), meaning that the calculated path from node 1 to node 2 is 1-3-4-2. If the initial path from node 3 to node 2 is 3-4-2, the path 1-3-4-2- from node 1 to node 2 can be found faster. The backup path can be repaired more efficiently by searching the initial path set I through DFS. After the new backup path from node 1 to node 2 is found, the subsequent backup path can be found by union, that is, to use the new effective backup path from node 1 to node 2 to replace the invalid original backup path. If no path is found, it means that there is no backup path to protect the link.

The BKM algorithm proposed in this paper uses the next-hop mechanism to solve the problem that the backup path fails due to link failure. Furthermore, the algorithm also avoids a large number of path-finding computations by eliminating the need for recomputing all backup paths, and the use of initial path collection is of great help to optimize the efficiency of backup path maintenance.

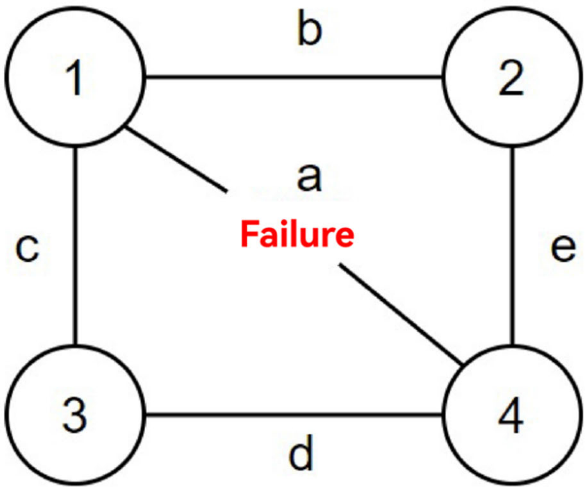


FIGURE 7. **Example of BKM algorithm.**

C. EXPERIMENTAL TEST

In order to verify the efficiency of the BKM algorithm, this study sets the Dijkstra algorithm and LFA+ algorithm [25] as a comparison. In the case of a certain number of link failures randomly generated, the backup paths were updated using the two algorithms respectively, and the resulting calculation time under link failures of various scales is shown in Fig. 8. It can be seen that the solving time for both the BKM algorithm and the LFA+ algorithm decreases as the link failure rate increases. However, the BKM algorithm shows a more significant decrease, indicating that BKM can more effectively detect situations with no backup paths, promptly reducing the search scope. In contrast, Dijkstra's algorithm is characterized by large-scale and global backup path computation and cannot dynamically adjust based on actual conditions. Thus, in terms of backup path update efficiency, the BKM algorithm demonstrates significantly higher efficiency compared to both the Dijkstra algorithm and the LFA+ algorithm. In the best case, the performance of the BKM algorithm is 74.3% better than the Dijkstra algorithm and 60.0% better than the LFA+ algorithm when the link failure rate reaches 10%.

To verify the effectiveness of the BKM algorithm, we conducted experiments using the Teledisic satellite constellation on the NS2 simulator. The constellation used in the experiment has 12 planes, and each plane has 24 low-Earth orbit satellites, one of which is an independent routing calculation satellite, and all data in the plane are forwarded through this satellite. In the experiment of this study, the capacity of the ISL is 155 Mb, the ISL between planes is turned on or off according to the direction of satellite movement, and the simulation time equals the orbit cycle (6794 seconds). To test the performance of the BKM algorithm in dealing with link failures, randomly generated failed links of gradually increasing numbers were deployed to the corresponding planes, and the average delay was calculated by sending a data packet every 30s. The results are shown in Fig. 9. It can be seen that

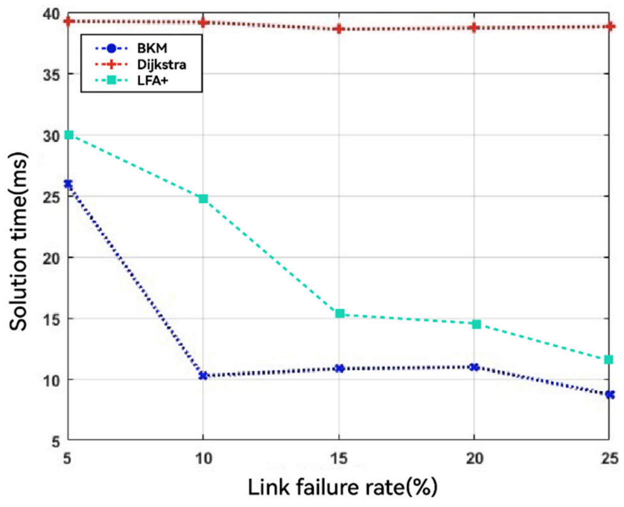


FIGURE 8. Update time comparison.

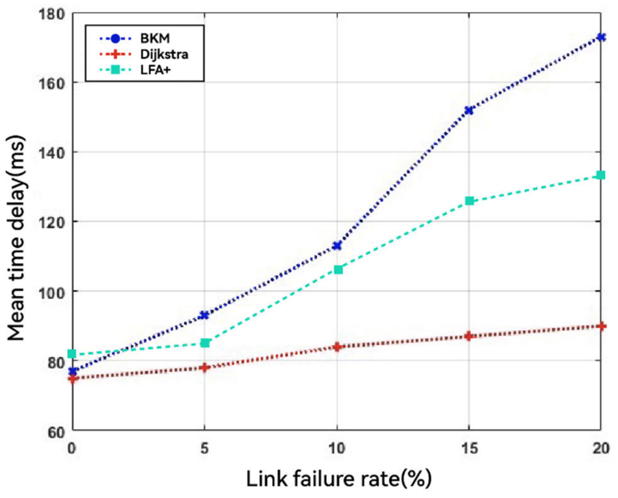


FIGURE 9. Comparison of average delays with different failure rates.

the length of the backup path updated by the BKM algorithm and the LFA+ algorithm based on the next hop mechanism is greater than that of the backup path updated by Dijkstra based on the shortest path principle, and in actual situations, some paths cannot be connected, which is related to the specific implementation of the algorithm.

D. RESULTS ANALYSIS AND SUMMY

The experimental results show that the BKM algorithm can help accelerate the update of some backup paths, but its implementation is more complicated, and the availability and length of the backup paths are affected by the implementation method. Therefore, the BKM algorithm needs further optimization. On the other hand, this paper also demonstrates that it is feasible to optimize the maintenance of hybrid routing backup paths in satellite routing based on the next-hop mechanism. Therefore, our BKM algorithm has certain advantages, this is because we adopt hybrid routing in our algorithm, and combine the advantage of centralized routing with distributed routing.

V. CONCLUSION

To address satellite link issues such as high failure rates, low stability, limited computing resources, and large delays in satellite-ground transmission, this paper analyzes different fast reroute modes in satellite networks. Aiming at centralized routing in the satellite network, this paper proposes a fast reroute scheme that integrates segment routing and divides the relay satellites into computing satellites and forwarding satellites. This scheme can make more efficient use of satellite resources and can generate backup paths without ground control in the event of a link failure. However, computing satellites still need to maintain high-frequency periodic contact with the ground. To address the selection of relay satellites and the corresponding segment routing domain in the dual relay satellite mode, this paper proposes a node partition algorithm that can efficiently allocate appropriate nodes based on relay satellites' locations and resources. Experimental results show that the algorithm can achieve effective utilization of computing/storage resources of relay satellites while maintaining a short average distance between nodes. In addition, for nodes that can independently calculate paths in hybrid routing, this paper proposes a backup path maintenance algorithm that can enable backup path updating and maintenance without requiring global recalculation of backup paths. Experiments show that this algorithm is feasible to a certain degree but still has room for improvement.

REFERENCES

[1] S. Liu, Z. Gao, Y. Wu, D. W. K. Ng, X. Gao, K.-K. Wong, S. Chatzinotas, and B. Ottersten, "LEO satellite constellations for 5G and beyond: How will they reshape vertical domains?" *IEEE Commun. Mag.*, vol. 59, no. 7, pp. 30–36, Jul. 2021.

[2] Y. Wang, W. Feng, J. Wang, and T. Q. S. Quek, "Hybrid satellite-UAV-terrestrial networks for 6G ubiquitous coverage: A maritime communications perspective," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3475–3490, Nov. 2021.

[3] S. J. Ni, Y. Yue, and Y. Zuo, "Current status and prospect of satellite network routing technology," *J. Electron. Inf. Technol.*, vol. 44, pp. 1–13, 2022.

[4] Q. Xiaogang, M. Jiulong, W. Dan, L. Lifang, and H. Shaolin, "A survey of routing techniques for satellite networks," *J. Commun. Inf. Netw.*, vol. 1, no. 4, pp. 66–85, Dec. 2016.

[5] J. Papán, P. Segec, and P. Palúch, "Analysis of existing IP Fast ReRoute mechanisms," in *Proc. Int. Conf. Inf. Digit. Technol.*, Jul. 2015, pp. 291–297.

[6] H. Hasan, J. Cosmas, Z. Zaharis, P. Lazaridis, and S. Khwandah, "Development of FRR mechanism by adopting SDN notion," in *Proc. 24th Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2016, pp. 1–7.

[7] B. Feng, Y. Huang, A. Tian, H. Wang, H. Zhou, S. Yu, and H. Zhang, "DR-SDSN: An elastic differentiated routing framework for software-defined satellite networks," *IEEE Wireless Commun.*, vol. 29, no. 6, pp. 80–86, Dec. 2022.

[8] A. Rajagopal, A. Ramachandran, K. Shankar, M. Khari, S. Jha, and G. P. Joshi, "Optimal routing strategy based on extreme learning machine with beetle antennae search algorithm for Low Earth Orbit satellite communication networks," *Int. J. Satell. Commun. Netw.*, vol. 39, no. 3, pp. 305–317, May 2021.

[9] T. Pan, T. Huang, X. Li, Y. Chen, W. Xue, and Y. Liu, "OPSPF: Orbit prediction shortest path first routing for resilient LEO satellite networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[10] W. Yan, R. T. Tan, and D. Dai, "Nighttime defogging using high-low frequency decomposition and grayscale-color networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 473–488.

[11] Y. Liu, Z. Yan, J. Tan, and Y. Li, "Multi-purpose oriented single nighttime image haze removal based on unified variational Retinex model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 4, pp. 1643–1657, Apr. 2023.

[12] T. Truong-Huu, P. Prathap, P. M. Mohan, and M. Gurusamy, "Fast and adaptive failure recovery using machine learning in software defined networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Shanghai, China, May 2019, pp. 1–6, doi: 10.1109/ICCW.2019.8757169.

[13] N. Khan, R. B. Salleh, A. Koubaa, Z. Khan, M. K. Khan, and I. Ali, "Data plane failure and its recovery techniques in SDN: A systematic literature review," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 3, pp. 176–201, Mar. 2023.

[14] M. Xu, Q. Li, L. Pan, and Q. Li, "MPCT: Minimum protection cost tree for IP fast reroute using tunnel," in *Proc. IEEE 19th IEEE Int. Workshop Quality Service*, Jun. 2011, pp. 1–3.

[15] G. Enyedi, A. Csaszar, A. Atlas, C. Bowers, and A. Gopalan, *An Algorithm for Computing IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)*, document RFC 7812, 2016.

[16] A. Jarry, "Fast reroute paths algorithms," *Telecommun. Syst.*, vol. 52, pp. 881–888, Aug. 2011.

[17] X. Yang, D. Clark, and A. W. Berger, "NIRA: A new inter-domain routing architecture," *IEEE/ACM Trans. Netw.*, vol. 15, no. 4, pp. 775–788, Aug. 2007.

[18] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM Symp. Operating Syst. Princ.*, Oct. 2001, pp. 131–145.

[19] C. Filsfils and S. B. Previdi, "Fast reroute for segment routing traffic," U.S. Patent 9 485 150, Nov. 1, 2016.

[20] A. Giorgetti, A. Sgambelluri, F. Paolucci, and P. Castoldi, "Reliable segment routing," in *Proc. 7th Int. Workshop Reliable Netw. Design Model. (RNDM)*, Oct. 2015, pp. 181–185.

[21] M. A. M. Vieira, M. S. Castanho, R. D. G. Pacífico, E. R. S. Santos, E. P. M. C. Júnior, and L. F. M. Vieira, "Fast packet processing with eBPF and XDP: Concepts, code, challenges, and applications," *ACM Comput. Surveys*, vol. 53, no. 1, pp. 1–36, Jan. 2021.

[22] M. Xhonneux and O. Bonaventure, "Flexible failure detection and fast reroute using eBPF and SRv6," in *Proc. 14th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2018, pp. 408–413.

[23] K.-T. Foerster, M. Parham, S. Schmid, and T. Wen, "Local fast segment rerouting on hypercubes," in *Proc. 22st Int. Conf. Princ. Distrib. Syst. (OPODIS)*, 2018, pp. 1–16.

[24] T. Schüller, N. Aschenbruck, M. Chimani, and M. Horneffer, "Failure resiliency with only a few tunnels—Enabling segment routing for traffic engineering," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 262–274, Feb. 2021.

[25] S. Zhang, X. Li, and K. L. Yeung, "Segment routing for traffic engineering and effective recovery in Low-Earth Orbit satellite constellations," *Digit. Commun. Netw.*, Oct. 2022.

[26] L. Xu-Ming, "A new fast reroute strategy in Ka band satellite MPLS network based on cross-layer optimization," Tech. Rep., 2009.

[27] M. Ouyang, X. Duan, J. Liu, R. Zhang, T. Huang, and H. Lu, "Multi-path transmission scheme based on segment control in Low-Earth-Orbit satellite network," in *Proc. IEEE 22nd Int. Conf. High Perform. Switching Routing (HPSR)*, Jun. 2021, pp. 1–7.

[28] L. Jiang, J. Feng, Y. Shen Y, and X. Xiong, "Fast recovery routing algorithm for software defined network based operationally responsive space satellite networks," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 7, pp. 2936–2951, Jul. 2016.

[29] Z. Guo and Z. Yan, "A weighted semi-distributed routing algorithm for LEO satellite networks," *J. Netw. Comput. Appl.*, vol. 58, pp. 1–11, Dec. 2015.

[30] A. Mereu, D. Cherubini, A. Fanni, and A. Frangioni, "Primary and backup paths optimal design for traffic engineering in hybrid IGP/MPLS networks," in *Proc. 7th Int. Workshop Design Reliable Commun. Netw.*, Washington, DC, USA, Oct. 2009, pp. 273–280, doi: 10.1109/DRCN.2009.5339995.

[31] S. Antonakopoulos, Y. Bejerano, and P. Koppol, "Full protection made easy: The DisPath IP fast reroute scheme," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1229–1242, Aug. 2015.

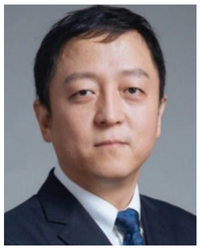
[32] K. Qiu, J. Zhao, X. Wang, X. Fu, and S. Secci, "Efficient recovery path computation for fast reroute in large-scale software-defined networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1755–1768, Aug. 2019.

[33] X. Zhang, Z. Cheng, R. Lin, L. He, S. Yu, and H. Luo, "Local fast reroute with flow aggregation in software defined networks," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 785–788, Apr. 2017, doi: 10.1109/LCOMM.2016.2638430.

XUN CHEN was born in Shandong, China, in 1985. She received the master's degree from Tsinghua University, in 2012. Since July 2021, she has been with Shenzhen Polytechnic University, where she is currently a Lecturer with the School of Communication.

ZHENGJIAN CHEN was born in Huai'an, Jiangsu, China, in 1981. He received the bachelor's degree in engineering from Xi'an Jiaotong University, in 2004, the master's degree in engineering from the Harbin Institute of Technology, in 2012, and the master's degree in economics from Peking University, in 2019.

He has been a Technician with the Shenzhen Energy Group, since 2004, where he obtained the title of Senior Engineer, in 2012. In 2020, he was the Head of smart energy technology and the President of the Smart Energy Research Institute, Shenzhen Energy Group Company Ltd. He led the research and development of integrated energy dispatching systems, intelligent power plants, virtual power plants, and other system platforms. He has participated in two monographs and published 15 articles. He holds 20 patents. His current research interests include integrated energy, microgrids, energy digitization, the application of artificial intelligence, and 5G in the energy industry.

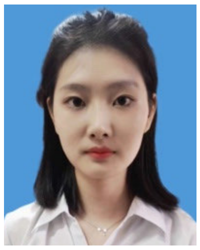


XIAOLEI CHANG was born in China, in 1975. He received the bachelor's degree in computer science and technology, the master's degree in management science and engineering, and the Ph.D. degree in engineering (electronic information) from Tsinghua University, in 1999 and 2001.

He is currently a Senior Engineer with the Information Technology Center/Network Research Institute, Tsinghua University, where he has been engaged in information technology application research and engineering practice, the transformation of scientific and technological achievements, and technology enterprise incubation for a long time. He is also the Deputy Director of the Next Generation Internet Research and Development Center, Research Institute of Tsinghua University in Shenzhen. He has completed more than 20 research projects and published more than 20 articles and two books. His current research interests include green data centers, cloud computing and edge computing, and cyberspace security.

TIAN JI was born in Jingdezhen, China, in 1989. He received the M.S. degree in big data from Tsinghua University, in 2023. He is currently a Core Member of the Future Internet Research Center, Research Institute of Tsinghua University in Shenzhen. He has participated in digital transformation projects for multiple local governments and leading state-owned enterprises, covering various aspects, including but not limited to data management and analysis, system integration, network architecture, and application development. His current research interests include the application of big data, cloud computing, and edge computing technologies in areas, such as smart cities and the industrial internet.

ZHENZHOU WU received the bachelor's degree in software engineering from Jilin University, in 2009. He was a Senior Architect, a Lisp Programmer, and the Research and Development Director of several internet companies. From 2009 to 2011, he was a Search Engineer with Avepoint, where he was responsible for the company's full-text search department. From 2011 to 2014, he was the Background Architect with Fenglin Volcano Technology Company Ltd., supporting the game platform with tens of millions of users and millions of users online at the same time. From 2014 to 2017, he was the Research and Development Director with Oudmon Technology Company Ltd., to develop the IoT platform, which has access to more than 50 million wearable devices. Since 2017, he has been the Research and Development Director of the Next Generation Internet Research and Development Center, Research Institute of Tsinghua University in Shenzhen, dedicated to the field of cloud-native, distributed networks, and distributed computing.



CHENXI LI was born in Chenzhou, Hunan, China, in 1999. She received the bachelor's degree in materials science and engineering from the China University of Mining and Technology, in 2020. Since 2021, she has been a Research Assistant with the Next Generation Internet Research and Development Center, Research Institute of Tsinghua University in Shenzhen. Her current research interests include the development and application of 5G, edge computing, and other new-generation information technologies.