A Disruption Tolerant Distributed Routing Algorithm in LEO Satellite Networks

Jifeng Jin 1,2, Feng Tian 2, Zijian Yang 1,2, Hao Di 3 and Guotong Li 1,2, *

1 University of Chinese Academy of Sciences, Beijing 100044, China; jinjifeng18@mails.ucas.ac.cn (J.J.); yangzijian18@mails.ucas.ac.cn (Z.Y.)
2 Innovation Academy for Microsatellites of CAS, Shanghai 201203, China; tianf@microsate.com
3 Shanghai Yuanxin Satellite Technology Co., Ltd., Shanghai 201600, China; dihao1022@163.com
* Correspondence: ligt@microsate.com

Abstract: The low earth orbit (LEO) constellation network has become a promising approach to provide global communication services, due to its advantages in wide global coverage, low transmission delay, and convenient networking. However, the instability of the intersatellite laser terminal and the high relative speed between adjacent satellites cause frequent network topology changing problems for data routing. In this paper, a disruption tolerant distributed routing algorithm (DTDR) is proposed, where the satellites calculate the alternate path for transmission when the network topology changes, which improves the performance of packet loss. Specifically, each satellite maintains the intersatellite link (ISL) information within a specified number of hops. When an ISL state changes within the specified number of hops, the corresponding satellite calculates and switches to the detour path. Furthermore, the traffic is balanced through the detour process. Various simulations were constructed and show that the proposed algorithm outperforms the existing algorithm in terms of packet loss ratio and transmission delay.

Keywords: LEO satellite network; distributed routing algorithm; disruption tolerant; packet drop rate; load balance

1. Introduction

Compared with GEO satellite networks, LEO satellite networks have the advantages of low communication delay, wide coverage, and low transmission power of communication equipment. With the growth of global communication services, LEO satellite networks have become more widely used in military, commercial, and civilian applications [1]. The LEO satellite communication system is composed of LEO satellite constellations, user terminals, and ground base stations. In the forward link, the traffic is uploaded from the ground base stations, then routed to the corresponding satellite, and finally transmitted to the user terminal. Correspondingly, in the reverse link, the traffic is uploaded from the user terminal, then routed to the corresponding satellite, and finally transmitted to the ground base station. This shows routing is an important issue in satellite communication systems.

Laser terminal has been widely used in intersatellite communication, due to its advantages of intensity, high transmission rate, and strong anti-interference, such as StarLink, LeoSat, and TelSate [2]. However, the laser intersatellite communication is unstable in various events such as sudden changes in channel status, satellite operating condition conversion, and on-board equipment failure, which may cause communication interruption (hereinafter referred to as link-off). Furthermore, various failures of the entire satellite can also be abstracted into corresponding link failure events that request specific consideration for the routing algorithm design [3].

In recent years, numerous routing algorithms have been designed to deal with the routing problems of LEO satellite networks. In order to solve the topology changes caused by satellite movement, there are mainly two topology discovery strategies [4]. The first
is the virtual topology strategy [5], which is suitable for complex network structures and changeable network topologies. The network topology is stored in each satellite in fragments, avoiding a large amount of routing information interaction and routing calculations, but at the same time, the satellites are required to have a certain storage capacity. The second strategy for virtual nodes [6] uses the periodicity of topology changes to avoid storage requirements, but it is only suitable for LEO satellite networks with simple topology and good coverage. The two topology discovery strategies have their own advantages and disadvantages. The virtual topology strategy divides the feeder link switching more finely, but also requires more storage space. The virtual node strategy saves storage space, but its utilization rate of the feeder link is low. At the same time, the satellite node is required to switch the IP address synchronously. However, none of these routing algorithms considers the instability of intersatellite communication.

In order to achieve network load balancing in the space segment, there are three types of solution [7]. The first type for centralized offline routing [8] solves the difficulties of real-time routing better, but it cannot solve the problem of changes in business requirements. The second type for distributed on-demand routing strategy [9] mainly saves the overhead of routing information and unnecessary on-board routing calculations. It can support on-board real-time autonomous routing calculations and has certain load balancing capabilities. This kind of scheme is suitable for sudden business access, but it is difficult to reach a global optimum. The third type for distributed multi-path routing strategy [1,3,10,11] is adapted to solve real-time link load jitter or link failure problems, avoiding link overload caused by burst traffic and resulting in packet loss, while the response speed is faster. This strategy can achieve local load balancing. However, existing research mostly uses router interface buffers to measure the degree of congestion of the link, which is impractical in satellite networks. Satellite networks encounter high propagation delay and cannot tolerate higher queuing delay, and due to the large intersatellite link bandwidth, the increase in queuing delay means that a large amount of buffer size needs to be prepared, which requires a higher performance for the space router [12]. Therefore, it is more reasonable to conduct research on this type of algorithm from the perspective of real-time link load.

There is currently no accepted solution to the problem of intersatellite link failure and interruption. FSA [8], a centralized routing algorithm based on virtual topology, utilizes predictable topology changes and service demand changes to dynamically allocate intersatellite link bandwidth resources, but it cannot handle unexpected link failure events. According to the distributed algorithms AODV and OLSR commonly used in terrestrial wireless ad hoc networks, both the LAOR algorithm [9] and the DODR algorithm [13] for link failure scenarios are proposed in the satellite network, and the orbit speaker strategy is proposed in CEMR [14]. These algorithms mainly save routing information overhead and unnecessary on-board routing calculation, while at the same time, they can support on-board real-time autonomous calculation of routing, and have certain load balancing capabilities. However, satellite networks have higher requirements for network service quality. When a link fails, these algorithms cannot provide an instant and reliable transmission path, which is prone to packet loss. DTDR, on the other hand, utilizes the mesh characteristics and periodic rules of the network topology of the space segment, and combines static routing with dynamic algorithms, which greatly improves the efficiency of the routing calculation. The distributed multi-path routing algorithm such as ELB [10] is suitable for solving the problems of real-time link load jitter or link failure, avoiding link overload caused by burst traffic and resulting in packet loss, and the response speed is fast. However, this kind of algorithm can only achieve local load balancing, and routing loops and cascading congestion problems are easily formed. In addition to propagating link failure information to neighboring nodes, DTDR also floods the information to neighboring nodes within a specified number of hops to achieve load balancing in the area of a failed link. At the same time, a loop avoidance mechanism is formulated in combination with the network topology characteristics of the space segment. OPSPF [15] generates an instantaneous routing table through periodic routing calculation and uses an on-demand dynamic
routing mechanism to deal with the impact of link failures. The algorithm can save the storage space of the network topology, but the routing information overhead is large when the link fails, and the convergence is slow in large satellite constellations, wrong paths are easily formed. DTDR only floods the failure information to the neighboring nodes in a certain area, which reduces the routing overhead, and the algorithm converges quickly.

For LEO satellite constellation routing, the immediacy and reliability of the path is a major challenge. The bandwidth of the laser link is large and the resources on the satellite are limited. There is no large-scale buffer to support the buffering of data packets [12]. The buffer can only meet the storage requirements of a small amount of data packets during the pre-switching of the intersatellite link [16,17]. Therefore, when a link is interrupted, if there is no timely route switching strategy, a large amount of packet loss will occur. When a link interruption event occurs, there must be a backup path that can be replaced immediately, and the path should be reliable, that is, to ensure that the data is accessible under certain severe conditions. The advantage of the routing algorithm based on the link state is that it can perceive the topology of the entire network in real time, but it is more vulnerable when the link is interrupted. Before the link status of the entire network is fully synchronized, routing loops are easily formed between satellite nodes due to the lack of reliable backup paths. At the same time, the cost of broadcasting link failure information in the entire network is relatively large, and the synchronization time is long, which causes a large amount of data loss.

In response to the above problems, we designed an LEO satellite network disruption tolerant distributed routing algorithm (DTDR) based on partial link status information. This algorithm can not only provide instant alternative backup paths for space segment routing but also balance the load around the failed link, while the routing cost is limited. Since the research in this paper focuses on the routing scheme for link failure, no clear strategy is given for global or local load balancing. It was mentioned above the current solution for realizing the load balance of the space segment network, this algorithm can better combine the above three types of solutions to form a complete space segment routing system.

2. LEO Satellite Network Model

In our constellation system, the space router on each satellite communicates with neighboring satellites through four interfaces connected to laser terminals and receives traffic from a ground base router and a space base station through the other two interfaces. In addition to transmitting data packets between itself and the space router, the laser terminal can also inform the space router if the laser link is in a fault state.

As shown in Figure 1, each satellite is connected to neighboring satellites through inter-plane ISLs and intra-plane ISLs. There are two seams between the two hemispheres, the satellites on both sides of the seam move in opposite directions and do not build links. The four intersatellite link interfaces of the satellites located in the western hemisphere are counted as 0, 1, 2, 3 in the counterclockwise direction from the north direction. As shown in Figure 2, when in the western hemisphere, satellite B communicates with the hetero-orbiting satellite A through its interface 1 and the interface 3 of satellite A. When the two satellites cross the polar region and enter the eastern hemisphere, they do not adjust their movement attitudes, but constantly adjust the positions of interface 1 and interface 3. Finally, in the eastern hemisphere, satellite B still communicates with satellite A through its interface 1 and the interface 3 of satellite A, and the positions of interface 1 and interface 3 are swapped at this time. During the entire process of crossing the polar regions, the inter-plane ISL always maintains link establishment status. From the perspective of routing, the length of the inter-plane ISL is constantly changing, and the link establishment status and the corresponding relationship between the interfaces are always constant.
For a polar orbit constellation, its constellation configuration is determined by the following parameters: orbit radius $R$, orbit number $N$, number of satellites in each orbit $M$, phase difference between adjacent orbiting satellites $\varphi$, and right ascension difference of the ascending node of the adjacent orbit $RI$.

The length of the intra-plane intersatellite links is fixed, and is calculated by

$$L_\varphi = \sqrt{2R \sqrt{1 - \cos\left(\frac{360^\circ}{M}\right)}}. \quad (1)$$

The length of the inter-plane intersatellite links varies with latitude. As shown in Figure 1, if one takes a point $B'$ at the same latitude as star $B$ on the orbital plane of star $A$, and take a point $A'$ at the same latitude as star $A$ on the orbital plane of star $B$, then $AA'B'B'$ is an isosceles trapezoid.

$$AA' = \sqrt{2R \sqrt{1 - \cos(RI)} \cos(latA)} \quad (2)$$

$$BB' = \sqrt{2R \sqrt{1 - \cos(RI)} \cos(latB)} \quad (3)$$

$$AB' = \sqrt{2R \sqrt{1 - \cos(\varphi)}} \quad (4)$$

where $latA$ is the latitude of star $A$ and $latB$ is the latitude of star $B$. If $A$ and $B$ are both in the southern or northern hemisphere, $|latA - latB| = \varphi$, if $A$ and $B$ are in different hemispheres, $|latA| + |latB| = \varphi$. 

Figure 1. Polar orbiting satellite constellation configuration.

Figure 2. Satellite link establishment status in different hemispheres.
According to the isosceles trapezoidal waist length formula, the length of the inter-plane intersatellite links is

\[ L_h = \sqrt{AB''^2 + AA'B'B} = \sqrt{2R \sqrt{1 - \cos(\varphi) + (1 - \cos(RI)) \cos(latA) \cos(latB)}}. \] (5)

3. Disruption Tolerant Distributed Routing Algorithm

3.1. Algorithm Overview

Our routing algorithm is divided into four parts. The first part introduces the periodic topology discovery process of the satellite network. The second part and the third part are based on the established network topology. Each satellite node calculates its main path direction and alternate path direction to all nodes in the whole network in a distributed manner and forms the main path routing table and the alternate path routing table. In these two parts, the horizontal and vertical hop directions of each source-destination node pair are calculated first. The main path direction is to select from these two directions according to the shortest path criterion and the alternate path direction is the remaining one of the two directions. The fourth part introduces the transmission method of the link failure information, and the algorithm that the node receiving the information calculates the detour path of the failure area and forms the detour path routing table. In this part, the satellite node checks the routing information within a specified number of hops it maintains. For each failed link, it determines whether a detour is required according to whether the main path to each destination node passes through the link and forms a forwarding table. Finally, it combines all of the forwarding table to form the detour path routing table.

When the satellite node forwards data packets, it first uses the main path routing table, and when the link directly connected to it fails, it switches to the alternate path routing table. After receiving the link failure information, the satellite node switches to the detour path routing table. The details of the priority of these three routing tables is introduced in Section 4.2. “Forwarding Process and Loop Avoidance”.

3.2. Global Topology Generation

In the previous section, we introduced the constellation configuration of the LEO satellite network. According to the motion law of the constellation, the change of the space segment network topology is mainly the change of the length of the inter-plane ISLs, which affects the propagation delay in the network. The length of the inter-plane ISL depends on the latitude of the two satellites that it is connected to. Therefore, the change of the latitudes of the satellites is directly related to the change of the network topology. Learning from the idea of virtual topology [5], we divide a period of satellite motion into \( K \) time intervals of equal length. When the adjacent time interval is switched, each satellite changes the latitude information \( latC \) of the local satellite and recalculates the route.

3.3. Main Path Direction Calculation

The space segment network topology is a mesh network structure. Each satellite in orbit is identified by an orbit number and a phase number. The relative position relationship between any two satellites at any time is determined and will not change, that is, between two satellites the orbital difference and phase difference are always the same. As shown in Figure 3, the current satellite node orbit number \( c_i = 2 \), phase number \( c_j = 6 \), destination node orbit number \( d_i = 5 \), phase number \( d_j = 4 \), the orbit difference between the current node and the destination node \( |d_i - c_i| = 3 \), phase difference \( |d_j - c_j| = 2 \). Data packets from the current node to the destination node need to pass through three hops of inter-plane ISLs and two hops of intra-plane ISLs. In this article, we refer to them as horizontal hops and vertical hops for short [18]. For any two source and destination satellites in the entire network, the direction of the horizontal hops and vertical hops can be determined.
3.3.1. Horizontal Hop Direction Calculation

Let \( c_i \) represent the current satellite node orbit number, \( d_i \) represent the destination satellite node orbit number, and \( d_h \) represent the horizontal hop direction. According to the introduction in the constellation configuration of the previous section, the four intersatellite link interfaces of the satellites located in the western hemisphere are counted as 0, 1, 2, 3 in the counterclockwise direction from the north direction, so that \( d_h = 3 \) means forwarding in the direction of increasing orbit number, \( d_h = 1 \) means forwarding in the direction of decreasing orbit number, and \( d_h = -1 \) means there are no horizontal hops, that is, the current satellite node and the destination satellite node are located on the same orbital plane. The calculation method of the horizontal hop direction is as follows.

1. If \( d_i > c_i \), then \( d_h = 3 \);
2. If \( d_i < c_i \), then \( d_h = 1 \);
3. If \( d_i = c_i \), then \( d_h = -1 \).

3.3.2. Vertical Hop Direction Calculation

Let \( c_j \) represent the current satellite node phase number, \( d_j \) represent the destination satellite node phase number, and \( d_v \) represent the vertical hop direction. \( d_v = 2 \) means forwarding in the direction of increasing phase number, \( d_v = 0 \) means forwarding in the direction of decreasing phase number, \( d_v = -1 \) means there are no vertical hops, that is, the current satellite and the destination satellite are at the same phase. Because the orbit of the LEO satellite network is circular, the vertical hop direction not only considers the size relationship between the phase number of the current satellite and the destination satellite, but also considers the phase difference, and finally determines the choice of the vertical hop direction \( d_v \). The calculation method of the vertical hop direction is as follows.

1. If \( c_j < d_j \), \( |d_j - c_j| \leq M - |d_j - c_j| \), then \( d_v = 2 \);
2. If \( c_j < d_j \), \( |d_j - c_j| > M - |d_j - c_j| \), then \( d_v = 0 \);
3. If \( c_j > d_j \), \( |d_j - c_j| \leq M - |d_j - c_j| \), then \( d_v = 0 \);
4. If \( c_j > d_j \), \( |d_j - c_j| > M - |d_j - c_j| \), then \( d_v = 2 \);
5. If \( c_j = d_j \), then \( d_v = -1 \).

3.3.3. Main Path Direction Selection

Since each satellite knows its own latitude \( \text{lat}_C \), the latitude of the destination satellite \( \text{lat}_D \) can be derived based on the phase difference with the target satellite. According to \( \text{lat}_C \) and \( \text{lat}_D \), it can be calculated whether the communication between the two needs to pass through the polar region. Let \( cpr = 0 \) represent that there is no need to pass through the polar regions, \( cpr = 1 \) represent that the communication needs to pass through the polar regions, and \( \text{lat}_C = \text{latmax} \) represent that there is no co-orbiting satellite that can be retransmitted between the current satellite and the nearest pole. The selection method of the
main path direction $rt$ is shown in Algorithm 1. This method allows the satellite to select the direction of the shortest path to the destination satellite from the horizontal hop direction and the vertical hop direction.

**Algorithm 1** Main path direction calculation.

| Variable definitions: $d_h$: horizontal hop direction, $d_v$: vertical hop direction, $rt[node]$: the main path direction of a node, $cpr$: records whether the communication needs to path through the polar region, $latC$: latitude information for current node. |
|---|
| 1: A new time interval is entered with the latitude $latC$ for current node |
| 2: for $node \in all\ nodes$ do |
| 3: calculate $d_h, d_v$ and $cpr$ |
| 4: if $d_h = -1$ then |
| 5: $rt[node] \leftarrow d_v$ |
| 6: else if $d_v = -1$ then |
| 7: $rt[node] \leftarrow d_h$ |
| 8: else |
| 9: if $cpr = 0$ then |
| 10: if $latC \geq latD$ then |
| 11: $rt[node] \leftarrow d_h$ |
| 12: else if $latC < latD$ then |
| 13: $rt[node] \leftarrow d_v$ |
| 14: endif |
| 15: else if $cpr = 1$ then |
| 16: if $latC = lat_{max}$ then |
| 17: $rt[node] \leftarrow d_h$ |
| 18: else if $latC \neq lat_{max}$ then |
| 19: $rt[node] \leftarrow d_v$ |
| 20: endif |
| 21: endif |
| 22: endif |
| 23: end for |

**Proof of Algorithm 1.** According to Equations (1) and (5), the intra-plane intersatellite link length $L_v$ is fixed and the inter-plane intersatellite link length $L_h$ increases with increasing latitude, and the total numbers of vertical hops and horizontal hops does not change. In order to get the shortest routing path, if the path between the current node and the destination node does not need to go through the polar region, then the horizontal hops should be completed at the node with higher latitude. If the path between the current node and the destination node needs to pass through the polar region, the horizontal hops should be completed at the node closest to the polar region. □

### 3.4. Alternate Path Direction Calculation

When the laser link in the direction of the main path fails, it must be switched to the alternate path immediately to avoid packet loss. Therefore, the alternate path must be calculated in advance and stored in the space router. When the main path is the horizontal hop direction, if there is a vertical hop between the current satellite node and the destination satellite node, then the alternate path is the vertical hop direction, otherwise the alternate direction is the vertical hop direction away from the polar region, which is also the vertical hop direction in which the latitude decreases. When the main path is the vertical hop direction, if there is a horizontal hop between the current satellite node and the destination satellite node, then the alternate path is the horizontal hop direction, otherwise the alternate direction is the horizontal hop direction away from the seam. The calculation method of the alternate path direction $rt1$ is as follows.

1. If $rt = d_h, d_v \neq -1$, then $rt1 = d_v$;
2. If $rt = d_h, d_v = -1$, then $rt1$ is the vertical hop direction away from the polar region;
3. If $rt = d_v, d_h \neq -1$, then $rt1 = d_h$. 


4. If \(rt = d_v, d_h = -1\), then \(rt1\) is the horizontal hop direction away from the seam.

For the alternate path direction here and the detour direction of the failure area, to be proposed in the next section, it is inevitable to encounter the problem of routing loops. For the solution of routing loop elimination, please refer to Section 4.2. “Forwarding Process and Loop Avoidance”.

3.5. Failure Area Routing Calculation
3.5.1. Routing Information Transfer

When an ISL is disrupted, only the directly connected space router will perceive and adopt the corresponding routing strategy, and the surrounding nodes will not reduce the incoming data to the node, which will cause congestion in the links around the node. If the node disseminates link failure information to the entire network, it will cause a certain amount of overhead, and secondly, the link failure information received by the space router that is far away from the node will be lagging, and it is of little significance for the space routers far away from the failed link to take countermeasures. Considering that the intersatellite link failure time is short, and the timeliness requirements are high, it is not the best choice to spread the link failure information across the network. In our routing strategy, the router directly connected to the failed link floods the link failure notification to the satellite nodes within \(n\) hops. The satellite node that receives the failure information immediately enters the failure area state (FAS) and adopts the strategy of bypassing in advance.

3.5.2. Detour Path Direction Calculation

Since the link failure information is flooded to the satellite nodes within \(n\) hops, each satellite can obtain the real-time status of all links within \(n + 1\) hops. For a specific failed link, we adopt an early detour strategy to balance the network load. Figure 4 shows the strategy. First determine whether the main path to the destination node passes through a failed link. For the node whose main path passes through the failed link, its detour path direction is the same as the alternate path direction mentioned in Section 3.3. Since our main path calculation algorithm has a small overhead, we can calculate the first \(n + 1\) hops of the main path of any node at any time to determine whether it passes through a failed link without storing the main path for each destination node. Whenever new link failure information or link recovery information comes, the router must recalculate the routing table. When calculating the routing table, it needs to consider all failed links in the area. As long as there are failed links, the local satellite records the failure area state as 1. The steps to calculate the routing table are as follows and the calculation algorithm is described in Algorithm 2:

![Figure 4. Early detour strategy for failed links.](image-url)
1. Initialization.
   Record the forwarding interfaces for all destination nodes in the detour routing table as $-1$ and record the failure area state as 0.

2. Traverse all links within $n + 1$ hops.
   During the traversal process, if a failed link is found, the starting satellite node1 and the ending satellite node2 of the failed link are obtained, and step (3) is entered. At the same time, the failure area state is recorded as 1.

3. Calculate the forwarding table for a specific failed link.
   For a specific failed link, the starting satellite node1 and the ending satellite node2 are known. For each destination satellite node, we first determine whether its main path passes through the failed link. If it is determined to pass, then the forwarding interface for this node is recorded the same as in the alternate path routing table. If it is determined that it does not pass, then the forwarding interface of the destination node is recorded as $-1$.

4. Update the routing table for a failed link.
   According to the forwarding table calculated in (3), for each destination satellite node, if the forwarding interface in the routing table is $-1$, use the forwarding interface in the current forwarding table to update the forwarding interface in the routing table, and go back to step (2).

Algorithm 2 Failure area routing calculation.

Variable definitions: FAS: link failure state, $rt1[node]$: the alternate path direction of a node, $rt2[node]$: the detour path direction of a node, $ft[node]$: the detour path direction of a node in the current forwarding table, link_failure: records whether the link is failed, link.node1: the starting satellite of a failed link, link.node2: the ending satellite of a failed link.

1: A node receives link failure information
2: for node $\in$ all nodes do
3:   $rt2[node] \leftarrow -1$
4: end for
5: FAS $\leftarrow 0$
6: for link $\in$ links within $n + 1$ hops do
7:   if link_failure = 1 then
8:     FAS $\leftarrow 1$
9: end if
10: for node $\in$ all nodes do
11:   if node.mainpath passes link.node1 and link.node2 then
12:     $ft[node] \leftarrow rt1[node]$
13:   else
14:     $ft[node] \leftarrow -1$
15: end if
16: end for
17: for node $\in$ all nodes do
18:   if $rt2[node] = -1$ then
19:     $rt2[node] \leftarrow ft[node]$
20: end if
21: end if
22: end for

4. Detailed Algorithm Description
4.1. Algorithm Parameters and Performance Analysis
   Since the services carried by the space segment network mainly go through three to four hops, we compared the routing performance of the strategies in which the link failure information is propagated to one hop, two hops, and three hops for the V1 -> V9 data flow in a 3 x 3 network topology. As shown in Figure 5, $latV1 = latV3$, $latV1 > latV7$, $latV7 - latV4 = latV4 - latV1$. The main path of V1 -> V9 is V1-V2-V3-V6-V9, and the
failed link is V6-V9. Due to the long intersatellite link length, and the buffer size of the space router being relatively small compared to the laser link bandwidth, the transmission delay of the data packet is much smaller than the propagation delay. We take the propagation delay \( t \) and the detour distance \( d \) as the evaluation index of the detour path, the propagation delay \( t = L/c \), where \( L \) is the total length of the intersatellite link that the detour path passes through, and \( c \) is the speed of light, which is used to measure the quality of service provided by the detour path, while the detour distance \( d \) is represented by the graphic area enclosed by the detour path and the main path, which is used to measure the load balancing capacity of the detour path for the intersatellite link.

Let \( L_{h1} \) represent the length of the ISL V1-V2, \( L_{h2} \) represent the length of the ISL V4-V5, \( L_{h3} \) represent the length of the ISL V7-V8, and \( L_v \) represent the length of intra-plane ISL. If the link failure information is propagated to one hop, the data flow starts to detour when it reaches V3. The detour path is V1-V2-V3-V2-V5-V8-V9. Under this circumstance, the propagation delay \( t1 = (3L_{h1} + L_{h3} + 2L_v)/c \), the detour distance \( d1 = 2 \). When the link failure information is propagated to two hops, the data flow starts to detour when it reaches V2, and the detour path is V1-V2-V5-V8-V9. In this case, the propagation delay \( t2 = (L_{h1} + L_{h3} + 2L_v)/c \), the detour distance \( d2 = 2 \). When the link failure information is propagated to three hops, the data flow starts to detour when it reaches V1, and the detour path is V1-V4-V7-V8-V9. In this case, the propagation delay \( t3 = (2L_{h3} + 2L_v)/c \), the detour distance \( d3 = 4 \).

The above only exemplifies the failure of V6-V9, and V1-V2, V2-V3, and V3-V6 will also fail with the same probability. Under the premise that each link in the network fails with equal probability, the routing performance under various strategies is comprehensively analyzed, and the calculation results are as follows:

1. Dissemination of failure information to one hop.
   \[ t1 = (4L_{h1} + 5L_{h2} + L_{h3} + 8L_v)/4c, \]
   \( d1 = 1.75 \).
2. Dissemination of failure information to two hops.
   \[ t2 = (L_{h1} + 6L_{h2} + L_{h3} + 8L_v)/4c, \]
   \( d2 = 2 \).
3. Dissemination of failure information to three hops.
   \[ t3 = (6L_{h2} + 2L_{h3} + 8L_v)/4c, \]
   \( d3 = 2.5 \).

The strategy of dissemination of failure information to one hop is obviously undesirable because of repetitive paths and poor diversion capabilities. We compare the strategies of dissemination of failure information to two hops and three hops.

Derived from the Section 2,

\[
L_{h} = \sqrt{2}R \sqrt{1 - \cos(\phi) + (1 - \cos(R1)) \cos(latA) \cos(latB)}. \tag{6}
\]
The variables in this formula are $latA$ and $latB$, in $L_{h1}$, $latA = latV1$, $latB = latV2$, in $L_{h3}$, $latA = latV7$, $latB = latV8$, since $latV1 > latV7$, $latV2 > latV8$, so $L_{h3} > L_{h1}$, $t3 > t2$.

According to the changing law of the cosine function, when the satellite is at low latitudes, the change of $L_{h}$ is relatively gentle, but at high latitudes, the change is relatively large, so that the strategy of dissemination of failure information to two hops has an obvious advantage on delay at high latitudes. At the same time, the LEO satellite network has relatively dense business at low latitudes, and relatively few at high latitudes [19], so the strategy of dissemination of failure information to three hops is more important at low latitudes. In summary, satellites at high latitudes can adopt the strategy of dissemination of failure information to two hops, and satellites at low latitudes can adopt the strategy of dissemination of failure information to three hops.

4.2. Forwarding Process and Loop Avoidance

In the routing strategy mentioned above, a total of three routing tables are involved, and their priorities are: 1. detour path routing table, 2. main path routing table, and 3. alternate path routing table. During the forwarding process, the conditions for entering the next priority are as follows:

1. The satellite is not in the state of the failure area, that is, there is no failure link in the area.
   1-> 2
2. The corresponding value of the destination address of the data packet in the detour routing table is $-1$, that is, its main path does not pass through the failed link.
   1-> 2
3. According to the current routing table, the forwarding interface is in an unestablished state (point to the seam) or the link to which it points has failed.
   1-> 2, 2 -> 3
4. According to the current routing table, the forwarding interface direction is the same as the incoming interface direction of the data packet, which means the direction leads packets to turn back.
   1-> 2, 2 -> 3

As shown in Figure 6, the data packet enters from node 4 to node 5, p1 is the main path, and p2 is the alternate path. In the case where the main path and the alternate path are both unavailable, the direction p3 opposite to the alternate path is selected for forwarding. At this time, it is no longer considered whether the forwarding interface is in the same direction as the incoming interface, that is, the data packet is allowed to turn back. If p3 is unavailable, the remaining p4 is used for forwarding at this time, and the local satellite is marked as invalid at the head of the data packet. Then, the data packet will no longer be sent to this satellite when it passes through the satellite adjacent to this satellite.

Figure 6. Forwarding interface selection.
5. Performance Analysis

5.1. Simulation Design

In order to analyze the actual performance of the algorithm, we use OPNET to build a simulation platform of the LEO satellite network to simulate the algorithm. The Walker parameter of the constellation is 128/8/3, the orbital height is 1050 km, and the orbital inclination is 89°. There is a seam between the orbits in the opposite direction, and the satellites on both sides of the seam are not linked.

We use the satellite communication equipment coverage density as the simulation input, and the traffic demand between any two satellites node \( i \) and \( j \) meets [20]:

\[
TR_{ij} = \left( \frac{SD_i \times SD_j}{d_{ij}} \right)^{\theta}.
\]  

(7)

Among them, \( SD_i \) and \( SD_j \) represent the equipment coverage density of the area served by satellite node \( i \) and satellite node \( j \), \( d_{ij} \) represent the distance between the two satellite nodes, \( \theta = 0.5, \varphi = 1.5 \), and the data flow of each node obeys the Poisson distribution.

We compared the disruption tolerant distributed routing algorithm (DTDR) based on local link information in this article with the DRA algorithm [6], OPSPF algorithm [15], and DODR algorithm [13]. The DRA algorithm statically calculates the route in each virtual node switching cycle, and when the link fails, the satellite directly connected to it switches to the backup path. OPSPF forms the network topology of the space segment according to the position of the local satellite and the relative position of satellites during topology switching. When the link fails, it advertises the failure information globally, and uses the shortest path algorithm to calculate the route. The DODR algorithm is based on the topology discovery strategy of the virtual nodes. In the process of path discovery, the routing information overhead is saved by restricting the access area. When the link fails, the area is expanded to establish a new path. The path selection follows the shortest propagation delay criterion. Common algorithm performance indicators include end-to-end delay, throughput, packet loss rate, link utilization, etc. Here we use end-to-end delay and packet loss rate to compare the performance of different algorithms [21].

5.2. Simulation Result

For different link failure scenarios, we mainly focus on two characteristics of the scenario. One is the probability of link failure per unit time. This feature determines the severity of the network environment and is used to measure the ability of the algorithm to withstand link failures. The second is the average load of the entire network link. This feature reflects the amount of data transmitted in the network and is used to measure the strength of the service provided by the algorithm. In the simulation, we set the reference value of the link failure probability to 2.5‰ and the reference value of the average network load to 24%, fix the value of one of the features, and analyze the influence of the other feature on the performance of the algorithm.

5.2.1. Performance Comparison under Different Link Failure Probabilities

In this part, we fixed the average load of the entire network link to 24% and set the link failure probability from 0.3‰ to 8‰ for simulation. Figure 7 shows the simulation results. In terms of packet loss rate, DTDR is always better than other algorithms, while OPSPF has the worst performance. This is because the route convergence in OPSPF is slow, and the response to short-term link failure is slow, so it is easy to generate the wrong paths which leads to routing loops, so packet loss occurs. When the link failure probability is low, the packet loss rate of the DRA algorithm is lower than that of the DODR algorithm, and when the link failure rate is high, its packet loss rate is higher than that of the DODR algorithm. It is more efficient to find a reliable backup path for DRA, but when the network environment is poor, a real-time path discovery strategy is required to deal with link failures. In terms of
latency, DODR is the best, DTDR and OPSPF are about the same, and DRA is the worst. The high delay of OPSPF is mainly caused by routing loops, while the high delay of DRA is due to the lack of advance awareness of failed links in the routing strategy, resulting in unnecessary duplicate paths and congestion. The delay of DTDR is also slightly higher than that of DODR when the link failure rate is high. The main reason is that the link failure information of DTDR is only transmitted within a limited number of hops, so unnecessary detours may occur at the boundary of the route convergence area. This phenomenon is inevitable. DTDR has formulated a loop avoidance strategy to reduce the possibility of packet loss at the boundary of the convergence area to a detour. In general, although the delay of DTDR is slightly higher than that of DODR when the link failure rate is high, DTDR is still optimal compared to its obvious advantage in packet loss rate.

Figure 7. (a) Packet loss rate under different link failure probabilities; (b) average end-to-end delay under different link failure probabilities.

5.2.2. Performance Comparison under Different Average Link Loads

In this part, we fixed the link failure probability as 2.5‰, and changed the rate of upstream traffic of each node without changing its proportion, so that the average load of the entire network link was from 0.16 to 0.4. As shown in Figure 8, DTDR still consistently outperforms other algorithms in terms of packet loss rate. In terms of delay, when the average link load is lower than 28% or higher than 36%, the delay of the DRA algorithm is significantly higher than other algorithms, and the reasons were analyzed above. However, the delays of DTDR, OPSPF, and DODR algorithms are not much different, and DODR is slightly better. Compared with the delay changes under different loads, the difference of the performance on delay between algorithms is insignificant. This result shows that the main advantage of the DTDR algorithm lies in its low packet loss rate, which is obvious when the link load is at a medium-low level.
6. Conclusions

A disruption tolerant distributed routing algorithm (DTDR) is proposed in this paper, which resolves the unstable intersatellite link issue to improve the performance of the data loss ratio. Considering the characteristics of the LEO satellite network, our proposed algorithm only requests that each satellite maintains limited information and performs distributed routing computing. DTDR floods the link failure information to neighboring nodes within a specified number of hops to achieve load balancing in the area of the failed link. It utilizes the mesh characteristics and periodic rules of the network topology of the space segment, and combines static routing with dynamic algorithms, which greatly improves the efficiency of the routing calculation. In addition, a loop avoidance mechanism is formulated in combination with the network topology characteristics of the space segment. Various simulations are constructed, and the results show that the proposed algorithm outperforms the existing algorithms in terms of transmission delay and packet loss ratio.

Author Contributions: Conceptualization, J.J. and H.D.; methodology, J.J. and F.T.; software, J.J. and Z.Y.; validation, F.T. and G.L.; formal analysis, J.J.; investigation, J.J.; resources, F.T. and G.L.; data curation, J.J.; writing—original draft preparation, J.J.; writing—review and editing, F.T.; visualization, J.J.; supervision, G.L.; project administration, G.L.; funding acquisition, G.L. and F.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National special support plan for high-level talents (grant number WRJH19DH01), Youth Innovation Promotion Association of Chinese Academy of Sciences (grant number 2021291) and Shanghai Sailing Program (grant number 19YF1446400).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Song, G.; Chao, M.; Yang, B.; Zheng, Y. TLR: A Traffic-Light-Based Intelligent Routing Strategy for NGEO Satellite IP Networks. *IEEE Trans. Wirel. Commun.* 2014, 13, 3380–3393. [CrossRef]
2. Li, J.; Zhao, S. Reviews of Space Laser Communication Research. *Electron. Test* 2013, Z2, 80–83.
3. Liu, P.; Chen, H.; Wei, S.; Li, L.; Zhu, Z. Hybrid-Traffic-Detour based load balancing for onboard routing in LEO satellite networks. *China Commun.* 2018, 15, 28–41. [CrossRef]
4. Qi, X.; Ma, J.; Wu, D.; Liu, L.; Hu, S. A Survey of Routing Techniques for Satellite Networks. *J. Commun. Inf. Netw.* 2016, 1, 66–85. [CrossRef]

5. Werner, M. A dynamic routing concept for ATM-based satellite personal communication networks. *IEEE J. Sel. Areas Commun.* 1997, 15, 1636–1648. [CrossRef]

6. Ekici, E.; Akyildiz, I.F.; Bender, M.D. A distributed routing algorithm for datagram traffic in LEO satellite networks. *IEEE/ACM Trans. Netw.* 2001, 9, 137–147. [CrossRef]

7. Cai, H.; Bian, D.; Miao, Q. Research on Routing Technology for Broadband Multimedia LEO Satellite Network. *DIGITCW* 2019, 4, 4–9.

8. Chang, H.S.; Kim, B.W.; Lee, C.G.; Min, S.L.; Choi, Y.; Yang, H.; Kim, C.S. SFSA-based link assignment and routing in low-earth orbit satellite networks. *IEEE Trans. Veh. Technol.* 1998, 47, 1037–1048. [CrossRef]

9. Papapetrou, E.; Karapantazis, S.; Pavlidou, F.-N. Distributed on-demand routing for LEO satellite systems. *Comput. Netw.* 2007, 51, 4356–4376. [CrossRef]

10. Taleb, T.; Mashimo, D.; Jamalipour, A.; Kato, N.; Nemoto, Y. Explicit Load Balancing Technique for NGEO Satellite IP Networks With On-Board Processing Capabilities. *IEEE/ACM Trans. Netw.* 2009, 17, 281–293. [CrossRef]

11. Korcak, O.; Alagoz, F. Analysis of Priority-based Adaptive Routing in Satellite Networks. In Proceedings of the 2005 2nd International Symposium on Wireless Communication Systems, Siena, Italy, 5–7 September 2005; pp. 629–633.

12. 7 Series FPGAs Data Sheet: Overview. DS180 (v2.6.1). Available online: [www.xilinx.com](http://www.xilinx.com) (accessed on 8 September 2020).

13. Ji, X.; Liu, L.; Zhao, P.; Wang, D. A destruction-resistant on-demand routing protocol for LEO satellite network based on local repair. In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 2013–2018.

14. Bai, J.; Lu, X.; Lu, Z.; Peng, W. Compact explicit multi-path routing for LEO satellite networks. In Proceedings of the HPSR. 2005 Workshop on High Performance Switching and Routing, Hong Kong, China, 12–14 May 2005; pp. 386–390.

15. Pan, T.; Huang, T.; Li, X.; Chen, Y.; Xue, W.; Liu, Y. OPSPF: Orbit Prediction Shortest Path First Routing for Resilient LEO Satellite Networks. In Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.

16. Sun, J.; Modiano, E. Routing strategies for maximizing throughput in LEO satellite networks. *IEEE J. Sel. Areas Commun.* 2004, 22, 273–286. [CrossRef]

17. Taleb, T.; Kato, N.; Nemoto, Y. REFWA: An Efficient and Fair Congestion Control Scheme for LEO Satellite Networks. *IEEE/ACM Trans. Netw.* 2006, 14, 1031–1044. [CrossRef]

18. Zhang, L.; Yan, F.; Zhang, Y.; Wu, T.; Zhu, Y.; Xia, W.; Shen, L. A Routing Algorithm Based on Link State Information for LEO Satellite Networks. In Proceedings of the 2020 IEEE Globecom Workshops, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.

19. Xia, S.; Jiang, Q.; Zou, C.; Li, G. Beam Coverage Comparison of LEO Satellite Systems Based on User Diversification. *IEEE Access* 2019, 7, 181656–181667. [CrossRef]

20. Liu, Z.; Li, J.; Wang, Y.; Li, X.; Chen, S. HGL: A hybrid global-local load balancing routing scheme for the Internet of Things through satellite networks. *Int. J. Distrib. Sens. Netw.* 2017, 13, 1550147717692586. [CrossRef]

21. Liu, Z.; Han, J.; Wang, Y.; Li, X.; Chen, S. Performance analysis of routing algorithms in satellite network under node failure scenarios. In Proceedings of the 2014 IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014; pp. 2838–2843.