future internet

MDPI

Article

MOLM: Alleviating Congestion through Multi-Objective Simulated Annealing-Based Load Balancing Routing in LEO Satellite Networks

Yihu Zhou [1], Haiming Chen [1,*] and Zhibin Dou [2]

[1] Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China; 2111082409@nbu.edu.cn
[2] The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, China; dzbjet@126.com
* Correspondence: chenhaiming@nbu.edu.cn

Abstract: In satellite networks, existing congestion resolution methods do not consider the predictability and stability of paths, leading to frequent path switches and high maintenance costs. In this regard, we propose a novel congestion resolution approach, named MOLM, which introduces a continuous neighbor set during path updates. This set includes nodes capable of establishing sustainable connections with the predecessors and successors of congested nodes. Combined with a multi-objective simulated annealing framework, MOLM iteratively derives an optimal selection from this set to replace congested nodes. Additionally, we employ a Fast Reroute mechanism based on backup paths (FRR-BP) to address node failures. The simulation results indicate that the optimal node endows the new path with optimal path stability and path latency.

Keywords: satellite internet; routing algorithm; path stability; latency; simulation experiments

Citation: Zhou, Y.; Chen, H.; Dou, Z. MOLM: Alleviating Congestion through Multi-Objective Simulated Annealing-Based Load Balancing Routing in LEO Satellite Networks. *Future Internet* **2024**, *16*, 109. https://doi.org/10.3390/fi16040109

Received: 7 February 2024
Revised: 14 March 2024
Accepted: 22 March 2024
Published: 25 March 2024

1. Introduction

Satellite internet, as a communication network with extensive coverage and high flexibility, is gradually becoming a crucial means of global connectivity. However, due to its unique communication characteristics and resource constraints, satellite internet often faces challenges of load imbalance when confronted with large-scale user demands. Optimizing load balance is essential for improving the performance of satellite internet and reducing communication latency. While load balancing issues have been extensively researched and applied in traditional ground communication networks, the peculiarities of satellite internet make this problem more complex and challenging.

The core of the load balancing problem lies in the proper allocation of communication loads within the satellite internet system, ensuring the maximization of resource utilization for each satellite node to enhance overall communication efficiency. However, factors such as an uneven geographical distribution of satellite nodes, variability in satellite conditions, and resource limitations pose constraints on the applicability of traditional load balancing strategies in satellite internet.

To address the load balancing issue in satellite internet, many strategies use the latency of new paths as an evaluation metric without considering path stability. However, path stability is also an important factor, as the dynamics of satellite networks and satellite movements can lead to path switching, resulting in switching and routing maintenance costs. Choosing a path with high stability can reduce costs. This paper proposes a low-maintenance, cost–load balancing routing method based on the multi-objective simulated annealing algorithm, named MOLM. Simulated annealing, as a heuristic optimization algorithm, possesses global search capabilities and adaptability to multi-objective optimization problems. When network congestion occurs, a continuous neighbor set is established, and

the multi-objective simulated annealing algorithm is used to select neighboring satellites that satisfy both strong path stability and low latency to replace congested nodes, thereby improving the overall performance of the satellite internet system.

The main contributions of the paper are embodied in the following aspects. (1) Based on the predictability of satellite networks, the concept of a continuous neighbor set is proposed, which includes nodes that can maintain continuous connections with the predecessor and successor of congested nodes. (2) The MOLM algorithm is used, which employs the multi-objective simulated annealing (MOSA) framework to iteratively derive an optimal element from the continuous neighbor set to replace congested nodes within a specific path, generating the final solution aimed at alleviating network congestion. (3) A comprehensive comparative experiment is conducted to demonstrate the performance of the MOLM algorithm, including comparative experiments under different networking methods and varying satellite constellation densities. Finally, the reliability of FRR-BP is measured using the time taken to reconstruct backup paths.

The rest of the paper is organized as follows. In Section 2, we summarize the work related to satellite network congestion resolution strategies. In Section 3, we introduce the architecture of satellite networks. In Section 4, we first introduce the process of using the MOLM algorithm to address network congestion issues. Then, we introduce the Fast Rerouting mechanism based on backup paths (FRR-BP). In Section 5, a comprehensive comparative experiment is conducted to demonstrate the performance of the MOLM algorithm. In Section 6, we give the conclusion.

2. Related Work

Load balancing methods can be categorized based on whether they have access to global information and the scope of route modifications, including global strategies, local strategies, and hybrid strategies, which combine both global and local approaches. For global strategies, decisions about on-board routing are made by incorporating information about the global network load state. In the case of local strategies, neighboring satellites are informed passively or actively about congestion states, and congestion avoidance is carried out based on the actual link conditions. Global strategies tend to have slow response times, time-sensitive global views, and high signaling overhead, while local strategies lack a global view and may fall into local optima. Hybrid strategies that combine global and local elements aim to leverage the advantages of both.

2.1. Global Strategy

Yi et al. [1] introduced an on-demand computation and cache-centric routing strategy and algorithm for satellite networks. The algorithm dynamically groups satellite network topologies, dividing routing into three stages: direction design, direction enhancement, and congestion avoidance. This algorithm boasts efficiency and flexibility, with congestion avoidance strategies enhancing the reliability and efficiency of data transmission. Li et al. [2] proposed a dynamic routing update algorithm based on real-time queue and routing state models to balance traffic loads and ensure the prompt transmission of data packets from each satellite, thereby avoiding congestion at the current node. They employ a load balancing mechanism to achieve equilibrium in information transfer across intra-network links, facilitating enhanced overall system throughput and congestion avoidance. Wang et al. [3] proposed two routing algorithms to optimize the utilization of inter-satellite links. By scheduling low-priority traffic onto links designated for high-priority services, the algorithms reduce the number of links used by low-priority traffic. Additionally, they introduce a load balancing strategy to control the aggregation of network flows, thereby minimizing the total number of utilized links. This approach enhances the resource utilization of satellite networks and contributes to energy conservation. Liu et al. [4], building upon the selective iterative Dijkstra algorithm, propose a selective diversion global load balancing strategy to address link congestion issues in low Earth orbit (LEO) satellite networks with low latitudes. Li and Tang et al. [5] introduce a mechanism for quantitatively

estimating the global network link state and dynamically adjusting queue delay weights. This mechanism efficiently and dynamically updates routing tables between two switches, significantly reducing routing overhead.

2.2. Local Strategy

Ma et al. [6] proposed a distributed datagram routing algorithm tailored for LEO satellite networks. The algorithm considers the congestion status when selecting the next satellite node to optimize congestion handling and reduce latency. Simultaneously, it ensures data transmission efficiency and success rates in the event of node failures. Song et al. [7] proposed the traffic-light-based intelligent routing (TLR) algorithm, which utilizes traffic light indicators to sense the congestion states of the current and next-hop nodes. Through a combination of pre-planning and real-time adjustments, TLR aims to derive an approximately optimal transmission path. Tang et al. [8] studied the network coding-based multipath cooperative routing (NCMCR) algorithm, where data streams can dynamically and collaboratively transmit across multiple paths. Liu and Tao et al. [9] proposed a traffic return routing algorithm based on segmented routing. This algorithm dynamically divides the network into light-load and heavy-load zones. The light-load zone employs a pre-balanced shortest path algorithm, while the heavy-load zone utilizes a minimum congestion index path algorithm. This approach effectively enhances the load performance of a LEO satellite network. Liu and Chen [10] addressed the issue of cascading congestion in traffic regions due to the lack of a global view in distributed load balancing routing schemes. They proposed a load balancing routing scheme based on hybrid traffic diversion, where the forwarding path is determined based on prior information acquisition and real-time congestion awareness. This approach aims to alleviate cascading congestion and achieve efficient routing transmission.

2.3. Hybrid Strategy

Considering the slow response time and temporal limitations of a global approach, as well as the high signaling overhead and that the local approach lacks a global view and is prone to local optima and cascading congestion scenarios, Liu and Li et al. [11] proposed a hybrid global–local load balancing (HGL) routing algorithm for connecting the IoT through satellite networks. This scheme optimizes routing decisions by combining global and local load balancing mechanisms. At the global level, a global load balancing strategy is established by collecting the load information of various devices in the IoT, and the optimal global path is selected in the satellite network. At the local level, local load balancing and path optimization are achieved by considering direct communication between devices and the assistance of neighboring devices. Computing capability is a valuable resource in satellite internet, particularly for satellites serving as space nodes. The size, weight, and power constraints of satellites limit their computing capabilities, posing greater challenges for routing algorithms. To meet the requirements of satellite storage and processing capabilities, Yi and Quan et al. [12] proposed a routing table generation and updating algorithm that delegates the routing calculations of the satellite network to onboard and ground routers. It separately generates local network routing tables and global network routing tables. This algorithm can reduce the demands on satellite computing power, alleviate the burden on inter-satellite links, and allow for upgrades to ground routers as the satellite network expands. Furthermore, Liu and Zhu et al. [13] proposed an approach that combines precomputation with distributed onboard real-time computation. Taking into account real-time inter-satellite link states, this method calculates the next-hop routes and forwarding tables on each satellite, enhancing real-time performance and reducing the computational load on satellites. Jiang et al. [14] proposed an energy-sensitive and congestion-balanced (ESCB) routing scheme. To address the limited energy resources of satellites, they introduced a tubular sliding time window (TSTW) model. This model enables the prediction of remaining energy and employs a multi-objective algorithm to calculate routing paths. Luo et al. [15] proposed a state-aware routing algorithm based on traffic prediction to disperse network

traffic. They designed a spatial–temporal attention fusion graph neural network (STAFGNN) to predict future network states by capturing the spatial–temporal correlation of satellite network traffic. Additionally, they designed multipath routing to reduce the probability of link congestion by dispersing all end-to-end traffic.

From Table 1, it is evident that most algorithms consider latency as an evaluation metric while mitigating network congestion, highlighting the importance of latency as a crucial factor. However, there is a notable absence of algorithms that consider path stability as a factor. Due to the high dynamism inherent in satellite networks, path switching is unavoidable due to satellite movements. Therefore, the selection of more stable paths to reduce the cost of path switching is a worthwhile issue for exploration. Consequently, this paper proposes a novel congestion mitigation method that takes into account both latency and path stability as evaluation metrics. Leveraging a multi-objective simulated annealing algorithm, the approach aims to enhance path stability while maintaining a minimal latency cost.

Table 1. Comparison of three different satellite internet routing strategies.

Strategy	Algorithm	Optimization	Latency	Computing and Energy Cost	Throughput	Packet Loss Rate	Path Stability
Global	Yi's	Relieve congestion and reduce latency	✓		✓		
	Li's	Improve throughput and avoid congestion	✓		✓		
	Wang's	Improve resource utilization and save energy	✓	✓			
	Liu's	Minimum delay	✓				
	Li and Tang's	Reduce latency and save energy	✓	✓	✓		
Local	Ma's	Reduce latency and packet loss rate	✓			✓	
	Song's	Reduce latency and improve throughput	✓		✓		
	Tang's	Data transmission on multiple links	✓		✓		
	Liu and Tao's	Improve load capacity and throughput			✓		
	Liu and Chen's	Relieve congestion and achieve efficient routing transmission	✓		✓		
Hybrid	Liu and Li's	Load balancing combining global and local paths	✓		✓	✓	
	Yi and Quan's	Reduce the burden of inter-satellite links	✓	✓			
	Liu and Zhu's	Reduce onboard computing load	✓	✓			
	Jiang's	An energy-sensitive and congestion-balanced (ESCB) routing scheme	✓	✓			
	Luo's	A state-aware routing algorithm based on traffic prediction	✓	✓	✓		

3. System Architecture

The architecture of satellite networks differs significantly from that of ground networks, as shown in Figure 1. The entire network system can be divided into three segments: the space segment, the ground segment, and the user segment. The space segment comprises various types of satellites, including geostationary Earth orbit (GEO) satellites, medium Earth orbit (MEO) satellites, and low Earth orbit (LEO) satellites. The ground segment mainly consists of ground stations. The congestion mitigation method proposed in this paper primarily targets LEO satellite scenarios, employing a hybrid strategy. Global information is necessary to construct a continuously visible neighbor set, followed by local route modifications near congested nodes. In this approach, specific ground stations serve as the network control center, overseeing route control.

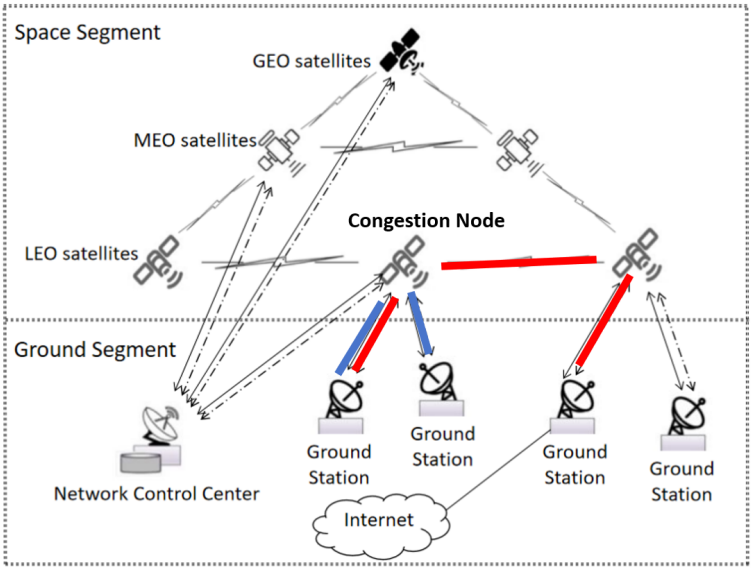


Figure 1. Satellite network architecture.

4. Algorithm Design

4.1. Overview of the MOLM Algorithm

The overview of the MOLM algorithm is shown in Figure 2. First, it will generate the initial solution using a shortest path algorithm like Dijkstra's algorithm. Dijkstra's algorithm will find paths with the minimum latency, but they are often accompanied by network congestion and path instability. Second, it will check for network congestion using Algorithm 1. If congestion occurs, it will establish the continuous neighbor set using Algorithm 2. Third, it will iterate by employing the multi-objective simulated annealing framework, with the minimum temperature and maximum iteration count as constraints. During each iteration, it will randomly select an element from the continuous neighbor set and replace the congested node to generate a new solution. Then, it will compare the new solution with the initial solution. If the new solution is better, it will accept it directly. If the initial solution is better, it will accept the new solution with a certain probability P, and with probability $1 - P$, it will stick to the initial solution. Finally, it will provide a routing solution to address network congestion.

4.2. Judgment of Network Congestion

Assuming there are two services, each with one path, as shown in Figure 1 with corresponding red and blue colors, if both of these service paths pass through a certain satellite node, then this satellite node becomes a hotspot. Calculate the total traffic at the hotspot, and if the total traffic exceeds the threshold, congestion is considered to have occurred. Regarding the threshold setting, we typically set it to 1.5 times the highest traffic volume in the service paths. The judgment of network congestion is depicted in Algorithm 1.

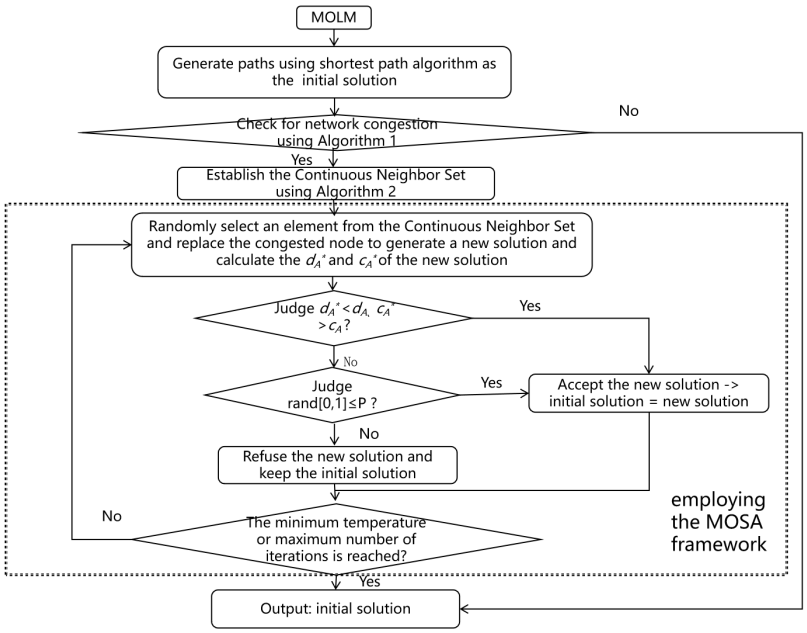


Figure 2. Overview of the MOLM algorithm.

Algorithm 1: Judgment of network congestion

Input: *path*1, *path*2, *threshold*
Output: *Congestion*

Find the shortest path from Source1 to Target1 by traditional Dijkstra algorithm, taking the geographical distance as the weight –> $(path1, length1)$;
Find the shortest path from Source2 to Target2 by traditional Dijkstra algorithm, taking the geographical distance as the weight –> $(path2, length2)$;
Obtain the traffic volume for *path*1 -> *path*1_*volume* ;
Obtain the traffic volume for *path*2 -> *path*2_*volume* ;
$threshold = max\{path1\_volume, path2\_volume\} * 1.5$;
if *path*1 *and path*2 *have a common node* **then**
    hotspot = the common node ;
    $hotspot\_volume = path1\_volume + path2\_volume$;
    if *hotspot_volume* > *threshold* **then**
        Congestion = True ;
    else
        Congestion = False ;
return *Congestion*;

4.3. Establishment of the Continuous Neighbor Set

If congestion occurs, we use the MOLM algorithm to resolve the congestion. Before that, we need to establish a continuous neighbor set, which includes nodes that can maintain continuous connections with both the predecessors and successors of the congested node. The process of establishing the continuous neighbor set can be described using Algorithm 2.

Assuming the paths for Service 1 and Service 2 are ['1','2','3','6'] and ['4','5','3','7']. The traffic volumes for path 1 and path 2 are 20 and 15, respectively. Therefore, the threshold is equal to $20 \times 1.5 = 30$. If both paths traverse satellite node 3 with cumulative traffic exceeding the threshold, network congestion occurs, with node 3 identified as the congested node. To alleviate this congestion, we intend to modify the path of Service 2. The predecessor and successor nodes for this congestion are nodes 5 and 7, respectively.

Algorithm 2: Establishment of the continuous neighbor set

Input: *path*1, *path*2
Output: *Continuous_Neighbor_Set*
if *congestion occurs on path1 and path2* **then**
  The congested node is identified as S_D;
  Find the predecessor node (S_A) and successor node (S_B) of the congested node in path2 ;
  Extract vectors from the adjacency matrix of satellite nodes at time t, representing the neighbors of S_A and S_B at time t;
  Extract vectors from the adjacency matrix of satellite nodes at time $t + \Delta t$ s, representing the neighbors of S_A and S_B at time $t + \Delta t$ s ;
  Let $Q = a_{S_A} \cap b_{S_A}$ and $R = a_{S_B} \cap b_{S_B}$, where Q and R, respectively, represent the continuous neighbors of S_A and S_B in the next Δt seconds;
  Let the *Continuous_Neighbor_Set* $= Q \cap R$;
return *Continuous_Neighbor_Set*;

To obtain the nodes that can maintain continuous connections with both node 5 and node 7, we need to know the nodes that are adjacent to nodes 5 and 7 at the current time as well as the nodes that will remain adjacent to nodes 5 and 7 after a certain period. Therefore, we require the adjacency matrix of the satellite nodes at the current time and after a certain period. Figure 3a represents the adjacency matrix of satellite nodes at the current time t, while Figure 3b represents the adjacency matrix of satellite nodes at time $t + \Delta t$ s.

Node \ Node	1	2	3	4	5	6	7
1	0	1	1	0	1	0	1
2	1	0	1	1	0	1	0
3	1	1	0	1	1	0	0
4	0	1	1	0	0	1	1
5	1	0	1	0	0	1	1
6	0	1	0	1	1	0	1
7	1	0	0	1	1	1	0

(a)

Node \ Node	1	2	3	4	5	6	7
1	0	0	1	0	1	1	1
2	0	0	1	0	0	0	1
3	1	1	0	1	1	0	0
4	0	0	1	0	1	1	1
5	1	0	1	1	0	1	0
6	1	0	0	1	1	0	1
7	1	1	0	1	0	1	0

(b)

Figure 3. Topology matrix table.

Therefore, the nodes adjacent to nodes 5 and 7 at the current time t can be represented as a_5 = {1,0,1,0,0,1,1} and a_7 = {1,0,0,1,1,1,0}, and the nodes adjacent to nodes 5 and 7 at time $t + \Delta t$ can be represented as b_5 ={1,0,1,1,0,1,0} and b_7 = {1,1,0,1,0,1,0}. Let $P = (a_5 \cap b_5) \cap (a_7 \cap b_7)$ = {1,0,0,0,0,1,0}, where P represents the continuous neighbor set. The nodes in the continuous neighbor set are node 1 and node 6.

4.4. The Design of MOLM

The MOLM algorithm first checks for network congestion using the method shown in Algorithm 1. Second, if congestion occurs, it uses the method shown in Algorithm 2 to establish the continuous neighbor set. Last, MOLM will select an element from the persistent neighbor set to replace congested nodes. The MOLM algorithm employs the multi-objective simulated annealing (MOSA) framework to iteratively select the optimal nodes from the continuous neighbor set to replace congested nodes. During each iteration, a random element is selected from the continuous neighbor set to replace the congested node, generating a temporary new path as a new solution. The stability and latency of the new path are compared with those of the original path. If the new solution is superior

to the original one, it is accepted directly. If the new solution is worse than the original one, it is accepted with a certain probability. The algorithm of MOLM can be described using Algorithm 3. The argument T means the temperature at the current moment, and min_T means the minimum temperature. The argument n means the number of iterations, and max_n means the maximum number of iterations. When the temperature reaches the minimum temperature or the number of iterations reaches the maximum number of iterations, the iteration will stop.

Algorithm 3: MOLM

Input: T, min_T, n, max_n, $path1$, $path2$
Output: $path1$, $initial\_path$

```
1  Check for network congestion using Algorithm 1;
2  if congestion occurs on path1 and path2 then
3      Establish the continuous neighbor set using Algorithm 2;
4      // Use the multi-objective simulated annealing framework to iteratively find
         the optimal nodes in the continuous neighbor set;
5      T = 1000;
6      n = 0;
7      while T > min_T and n < max_n do
8          Let initial_path = path2;
9          Calculate the d_A and c_A of the initial_path;
10         Randomly select an element from the continuous neighbor set to replace
             the congested node in the initial_path -> new_path;
11         Calculate the d*_A and c*_A of the new_path ;
12         if d*_A < d_A and c*_A > c_A then
13             Let initial_path = new_path ;
14         else
15             P = e^((d_A − d*_A) * (c_A − c*_A)/T);
16             Accept new_path with the probability P -> initial_path = new_path ;
17             Maintain the initial_path with the probability 1 − P -> initial_path;
18         T = T * cooling_rate;
19         n = n + 1;
20     return path1, initial_path;
21 else
22     return path1, path2;
```

From line 2 to line 7 in the MOLM algorithm, we show that it solves the congestion problem with dual optimization objectives related to path stability and latency, which are described as maximizing the average path duration and minimizing the average latency. The mathematical model of the multi-objective optimization problem is expressed as

$$f : L \rightarrow \begin{cases} \min d_A \\ \max c_A \end{cases} \tag{1}$$

In the given context, where f is the optimization objective function, $L = [l_{ij}[t]]$, $i, j = 1, 2, 3, 4 \ldots N$. $L = [l_{ij}[t]]$ represents the link from the starting point i to the ending point j at time t, where d_A is the average latency of these paths and c_A is the average duration of these paths. The constraints for establishing a link are as follows:

$$v_{ij} \in \{0,1\}, \quad \forall i,j \tag{2}$$
$$v^*_{ij} \in \{0,1\}, \quad \forall i,j \tag{3}$$
$$v_{ij} \geq l_{ij}, \quad \forall i,j \tag{4}$$
$$v_{ij} = v_{ji}, \quad \forall i,j \tag{5}$$

In the given context, Equations (2) and (3) involve v_{ij} and v^*_{ij}, representing whether satellite i and satellite j are visible and continuously visible, respectively. When visible, v_{ij} takes the value of 1; otherwise, it is 0. When continuously visible, v^*_{ij} takes the value of 1; otherwise, it is 0. Equation (4) indicates that a link between two satellite nodes can only be established if they are in a visible state. Equation (5) represents the symmetry constraint.

4.5. Fast Rerouting Mechanism

Fast Reroute (FRR) is a network fault recovery technique. In satellite networks, FRR can be implemented using either backup paths or local repair mechanisms. In the backup path-based FRR mechanism, when the primary path fails, data traffic is switched to the backup path. On the other hand, the local repair-based FRR mechanism involves computing multiple local paths to quickly repair the route near the point of failure. In this paper, we adopt the backup path-based Fast Reroute mechanism (FRR-BP) in the satellite network. In FRR-BP, when a node in the satellite network fails, the network control center removes the failed node from the network topology and recalculates the backup path from the source to the destination. To ensure the stable operation of the satellite network, it is essential to trigger the recalculation of the backup route and swiftly switch to the backup path when a network failure occurs, minimizing the impact of network failures on service quality and reliability. The details of FRR-BP are presented in Algorithm 4. Additionally, it is crucial to be mindful of potential network congestion when employing backup paths. If congestion occurs in the network, the resolution of congestion still necessitates the use of the MOLM algorithm.

Algorithm 4: Fast Rerouting method based on backup paths (FRR-BP)

Input: $G, Source, Target, path1, length1, path2, length2, fault\_node$
Output: $path1, path2$

Find the shortest path from $Source1$ to $Target1$ by traditional Dijkstra algorithm taking the geographical distance as the weight –> $(path1, length1)$;
Find the shortest path from $Source2$ to $Target2$ by traditional Dijkstra algorithm taking the geographical distance as the weight –> $(path2, length2)$;
if $fault\_node$ *in* $path2$ **then**
    //A node failure occurred, causing the path to be unreachable;
    Remove $fault\_node$ in $path2$ from G ;
    Find the shortest path from $Source2$ to $Target2$ by traditional Dijkstra algorithm taking the geographical distance as the weight->(new_path, new_length);
    Judge network congestion by calling Algorithm 1 -> $network\_congestion$;
    if $network\_congestion==0$ **then**
        $path2 = new\_path$;
        $length2 = new\_length$;
        print$(path1, path2)$;
    else
        MOLM$(T, min_T, n, max_n, path1, new_path)$;
else
    //If there is no disconnection in the path, output the original path
    print$(path1, path2)$;
return $path1, path2$;

5. Simulation Experiment Evaluation

5.1. Simulation Environment

We utilized the SILLEO-SCNS simulator proposed by B. S. Kempton [16], which is a simulation tool designed for studying routing in large-scale satellite networks. This tool enables the simulation of various satellite network architectures and routing schemes, allowing for performance evaluation. The architecture of this simulator primarily consists of a graphical user control interface, constellation class, and simulation class, as shown in Figure 4. The constellation class serves as the core component responsible for generating satellites, ground stations, and network structures. The simulation class handles the visualization of the satellite network and all associated animations. The GUI provides a user control interface that allows users to configure constellation parameters, set the source and destination for path planning, and specify satellite connectivity options. Communication between the GUI and simulation is facilitated through interprocess communication for information exchange.

The simulation adopts a Walker constellation configuration of 100/10/1, with an orbit altitude of 1200 km and an orbital inclination of 60°. To facilitate analysis and calculations, the sampling period for the constellation data is set to 10 s. The setting of simulation parameters is shown in Table 2. In the temperature cooling schedule for multi-objective simulated annealing, the initial temperature is set to 1000 °C. Following Kirkpatrick et al.'s analysis, the Boltzmann constant during temperature cooling is set to 0.95. The iteration limit is set to 2500, and the iteration stops when the temperature cools to the minimum temperature or the iteration count reaches the upper limit.

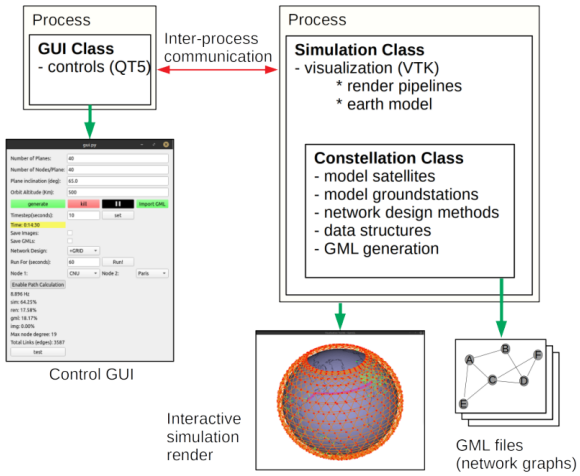


Figure 4. Simulator class diagram.

Table 2. Parameters of the simulation.

Parameter	Value	Meaning
T	1000 °C	The temperature at the current moment, with an initial value of 1000 °C.
cooling_rate	0.95	T * cooling ratio for each cooling process.
min_T	0.01 °C	Stop iteration when the temperature reaches the minimum temperature.
n	1	Current number of iterations.
max_n	2500	Stop iteration when the maximum number of iterations is reached.
step	10 s	The time interval for obtaining constellation data and routing.

5.2. Benchmarking Algorithms and Complexity Analysis

Our proposed MOLM algorithm aims to address network congestion issues, and we analyze its complexity in this part. The core idea of the MOLM algorithm is to establish a continuous neighbor set during network congestion and then utilize a multi-objective simulated annealing framework to select the best node from this set to replace congested nodes, achieving the goals of avoiding congestion while maintaining path stability and low latency.

Optimum is based on a greedy algorithm for node replacement. It employs a greedy strategy to select nodes from the continuous neighbor set, optimizing path stability and latency of the entire path. The strategy of *Shortest Path* generates the shortest path for congested traffic in the network, aiming to minimize latency. The difference between the two is that *Optimum* selects one node from a limited number of nodes, namely those in the continuous neighbor set, to minimize latency. Meanwhile, the shortest path algorithm searches for a route from *Source* to *Target*, resulting in the globally minimal latency path.

Firstly, we discuss the complexity of the MOLM algorithm. For the filtering process of the continuous neighbor set, we note that its time complexity is approximately O(n), where n represents the cardinality of the continuous neighbor set. Similarly, for the selection of the best node, we employ the Optimum method based on a greedy algorithm, with a time complexity that is also approximately O(n). This is because, under the greedy strategy, selecting nodes from the continuous neighbor set is a relatively efficient operation.

Furthermore, the complexity of the shortest path algorithm is also discussed. We use the Dijkstra algorithm to find the shortest path. Due to the +Grid networking method employed, with each satellite maintaining four connections, the topology graph is relatively sparse. Therefore, its time complexity is approximately O(nm), where m represents the number of edges in the network.

In summary, the MOLM algorithm exhibits low time complexity in addressing network congestion issues, making it capable of efficient operation in practical applications and effectively enhancing network performance.

5.3. Simulation Result

We measure path stability using the average duration of a path, which is calculated based on the consecutive occurrences of a particular path. Assuming a path persists continuously for N times, the duration of that path would be $N * step$. The comparison of different algorithms is shown in Figure 5.

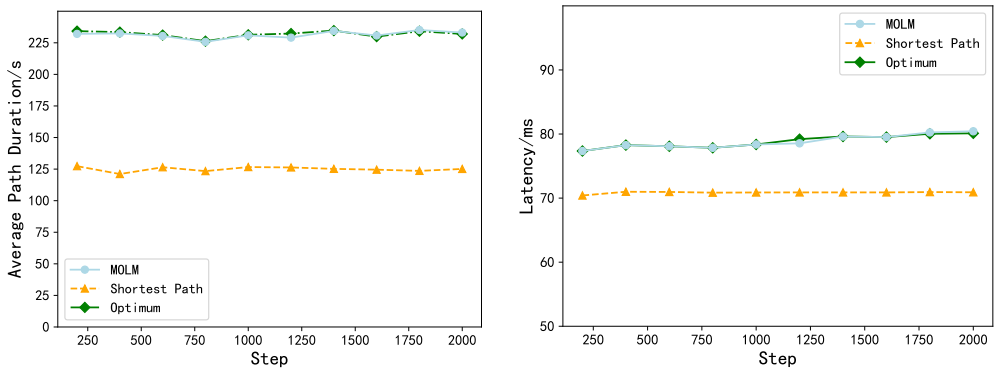
In terms of path stability, when using a shortest-path-first rerouting mechanism to address network congestion, the average path duration is approximately 125 s. By employing the MOLM algorithm in conjunction with multi-objective simulated annealing to address congestion, the average path duration can exceed 225 s. This represents an improvement of around 80% compared to the *Shortest Path* strategy.

In terms of path latency, the *Shortest Pat*h strategy achieves minimal latency, approximately around 70 ms. The MOLM algorithm filters out the continuous neighbor set from all satellites, iteratively selects an optimal candidate neighbor using multi-objective simulated annealing, and replaces congested nodes. This strategy, while enhancing path stability, maintains a relatively small latency cost. As shown in Figure 5b, the latency introduced by MOLM is around 80 ms, an increase of less than 15% compared to *Shortest Path*.

This is because the *Shortest Path* strategy prioritizes the selection of the shortest path, sacrificing path stability in pursuit of minimizing latency. In contrast, the nodes selected by the MOLM algorithm in the continuous neighbor set maintain sustained connections with the predecessor and successor nodes of congested nodes, contributing to robust path stability. After undergoing iterations of multi-objective simulated annealing, the nodes selected by MOLM are very close to the global optimum, resulting in a minimal latency cost.

Whether considering path latency or path stability, the results obtained with the MOLM algorithm approach those achieved by the *Optimum* strategy. This highlights the

characteristic of multi-objective simulated annealing of gradually converging towards the global optimum through iterations.

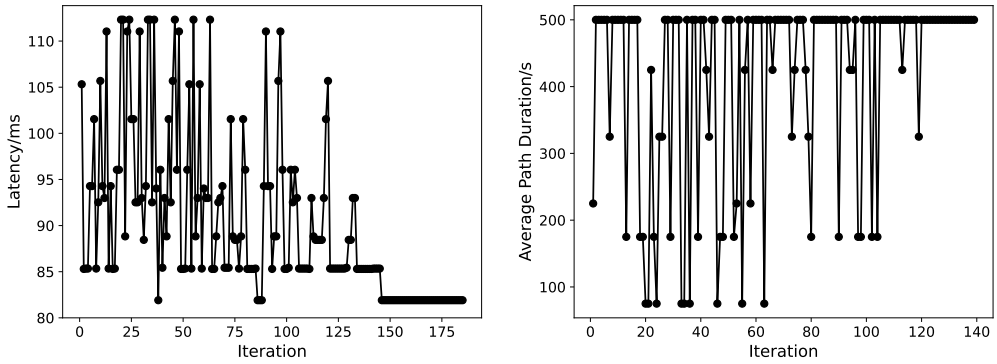


(a) (b)

Figure 5. Comparison of (**a**) path stability and (**b**) latency of different algorithms.

The convergence process of latency and path stability in multi-objective simulated annealing is shown in Figure 6. During the annealing process, when encountering a poorer new solution (i.e., a new path with higher latency or poorer path stability), the algorithm accepts this poorer solution with a probability P. However, when a better solution is encountered (i.e., a new path with both lower latency and better path stability than the original path), the algorithm directly accepts the new solution.

From the graph, it can be observed that the convergence characteristic curve is not strictly monotonic, indicating that the simulated annealing algorithm possesses the ability to escape from local optima and converge towards global optimum solutions.



(a) (b)

Figure 6. The convergence process of multi-objective simulated annealing.

5.4. The Impact of Networking Methods on Algorithm Performance

Network topology methods refer to the schemes by which network nodes form a topology. Common satellite network topology methods include Sparse, +Grid, and Ideal. Sparse indicates a sparse topology where each satellite can maintain only two connections. Specifically, a satellite connects to the satellites immediately before and after it in the same orbit. +Grid represents a grid topology where each satellite can maintain four connections. In addition to the two connections mentioned in Sparse, a satellite can also connect to the nearest satellites in the adjacent left and right orbits. In an ideal topology, each satellite can connect to any other satellite within its visible range, and there is no limit on the number of connections.

Due to the sparsity of the topology in the Sparse case, which may hinder the establishment of a continuous neighbor set, the MOLM algorithm is tested in both +Grid and Ideal scenarios. The results are averaged over 1000 steps and compared with *Shortest Path* and *Optimum*.

The comparison results for path stability and path latency are shown in Figure 7. In terms of average path duration, MOLM performs similarly in both +Grid and Ideal scenarios and tends to approach the Optimum. In the +Grid scenario, the average path duration obtained by the MOLM algorithm is improved by 104.29 s, approximately 82.36%, compared to the *Shortest Path*. In the Ideal scenario, the average path duration achieved by MOLM is improved by 130.52 s, approximately 162.24%, compared to the *Shortest Path*.

Regarding path latency, in the +Grid scenario, MOLM's path latency increases by about 7.66 ms, approximately 10.81%, compared to the *Shortest Path*. In the Ideal scenario, MOLM's path latency increases by about 7.22 ms, approximately 20.36%, compared to the *Shortest Path*. In the Ideal scenario, the path latency for all three strategies is significantly lower than the path latency in the +Grid scenario. This is because the *Shortest Path* strategy prioritizes the shortest path, and in the Ideal scenario, where the topology has high connectivity and numerous selectable neighbors, it is easier to find paths with lower latency. However, the trade-off is more frequent path switches and worse path stability. Therefore, in the Ideal scenario, the MOLM algorithm exhibits a particularly noticeable improvement in path stability.

Through the MOLM algorithm, nodes that can replace congested nodes and maintain longer connections with predecessors and successors are added to the continuous neighbor set. The multi-objective simulated annealing algorithm selects the optimal node from this set to replace the congested node, reducing path switches and significantly improving path stability. As the multi-objective simulated annealing algorithm considers both path latency and path stability, it enhances path stability while maintaining a relatively low latency cost.

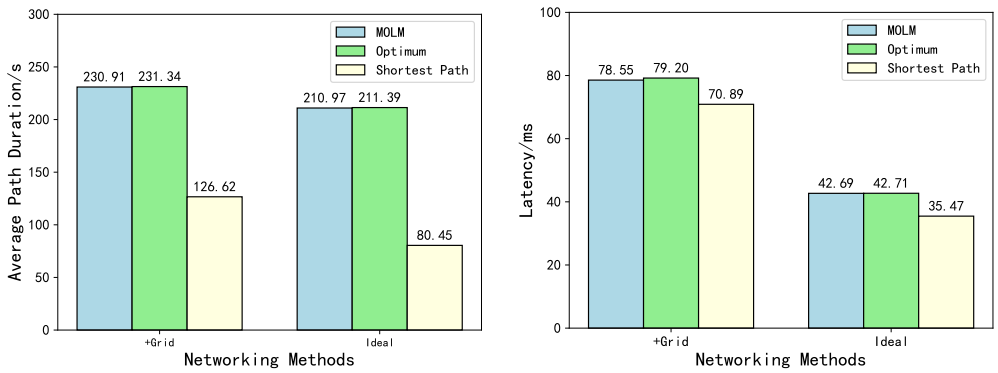
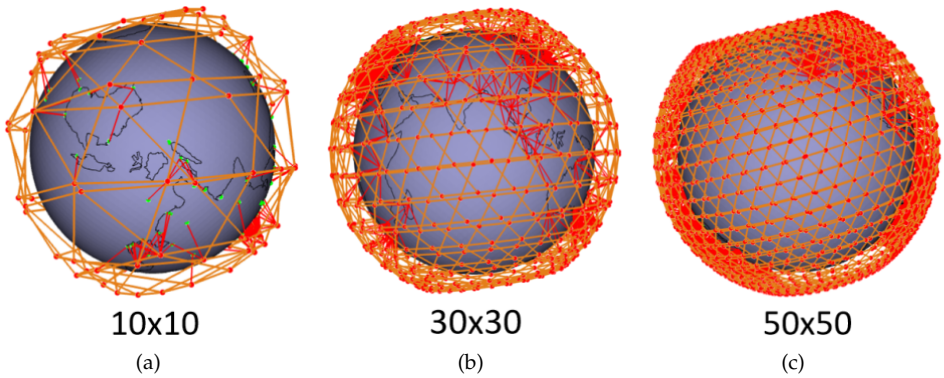


Figure 7. The impact of networking methods on algorithm performance.

5.5. The Impact of Constellation Density on Algorithm Performance

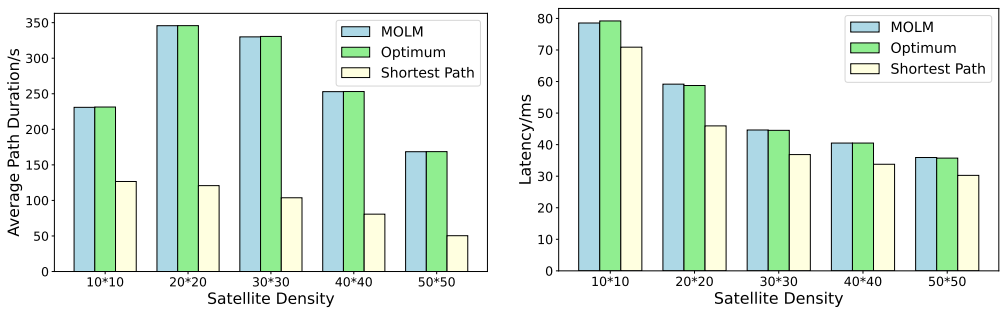
In general, a higher constellation density provides more options for selecting the next-hop neighbor when establishing paths using the shortest path algorithm. This increases the likelihood of finding low-latency paths. However, due to the increased number of choices, path stability may decrease. Figure 8 shows the constellation topologies for $10 \times 10, 30 \times 30$, and 50×50, and the +Grid scheme is used for network formation.



(a) (b) (c)

Figure 8. Constellation topology with different densities under the +Grid scheme.

We utilize satellite constellations of sizes $10 \times 10, 20 \times 20, 30 \times 30, 40 \times 40$, and 50×50 to investigate the impact of different constellation densities on the performance of the routing algorithm. The simulation results are shown in Figure 9.



(a) (b)

Figure 9. The impact of constellation density on algorithm performance.

From the perspective of path stability, the stability obtained by the *Shortest Path* algorithm decreases as the constellation density increases. In contrast, the path stability obtained by the MOLM algorithm does not exhibit a monotonic decrease but rather shows two significant peaks at constellation densities of 20×20 and 30×30. In terms of latency, all three strategies adhere to the rule that "higher constellation density leads to lower path latency". Therefore, this implies that in certain constellations with specific simulation characteristics (orbit height of 1200 km, inclination of 60 degrees, and constellation density of 20×20 or 30×30), when network congestion occurs, the new paths obtained using the MOLM algorithm can achieve a good trade-off between stability and latency.

5.6. Evaluation of FRR-BP

We integrate the MOLM algorithm with a Fast Reroute mechanism based on backup paths. After using multi-objective simulated annealing to select continuous neighbors for replacing congested nodes, if any node along the newly selected path experiences a failure, we promptly initiate the Fast Reroute based on backup paths. When establishing backup paths, two considerations are essential: first, avoiding the failed node, and second, steering clear of congested nodes to prevent potential congestion on the newly created backup path. Additionally, if the newly generated backup path still encounters network congestion with other existing traffic paths, we further employ the MOLM algorithm to alleviate network congestion.

We consider the time required to generate the final feasible backup path as the evaluation metric for rerouting. Tests were conducted on constellations with densities of $10 \times 10, 20 \times 20, 30 \times 30, 40 \times 40$, and 50×50 under both +Grid and Ideal networking meth-

ods. In general, higher constellation density leads to increased traversal time, resulting in longer rerouting times. The test results, as shown in Figure 10, indicate that under both +Grid and Ideal networking methods, higher constellation density corresponds to longer backup path reconstruction times.

In the +Grid scenario, even with a constellation density of 50 × 50 (2500 satellites in total), the reconstruction time remains within 1000 ms, which aligns with expectations. However, in the Ideal scenario, where there are no restrictions on the number of connections between satellites, the increase in connection density with rising constellation density leads to a rapid increase in backup path reconstruction time. This is a normal occurrence. Nevertheless, at lower constellation densities, the backup path reconstruction time remains relatively low, within 1000 ms.

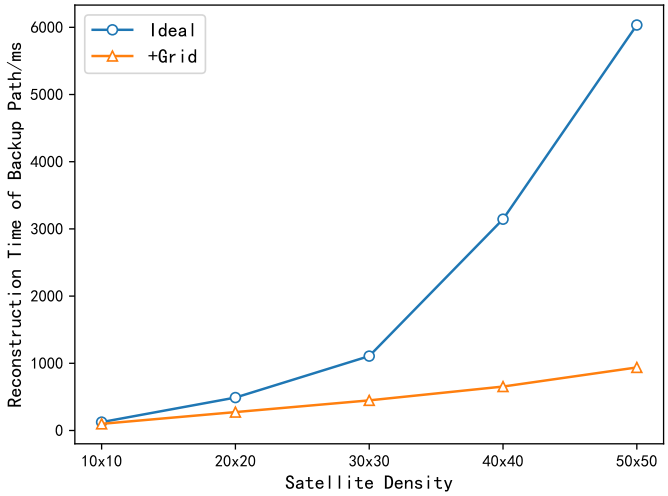


Figure 10. The impact of constellation density on the reconstruction time of backup paths.

6. Conclusions

Based on the predictability of LEO satellite networks, we propose a novel congestion resolution method named MOLM. Other congestion resolution methods focus on reducing path latency or increasing throughput while addressing network congestion. In contrast, MOLM considers path stability because it affects the cost of path switching and routing table maintenance. The main advantage of our proposed algorithm over others lies in simultaneously addressing network congestion, enhancing path stability, reducing path switching costs and routing table maintenance costs while maintaining relatively low latency costs. The approach constructs a continuous neighbor set when updating paths, consisting of nodes that can establish sustainable connections with both the predecessor and successor of congested nodes, thereby enhancing path stability. Integrated with a multi-objective simulated annealing algorithm, we iteratively derive a node from this set to replace the congested node. As the solutions generated by multi-objective simulated annealing approach global optimality, the selected node facilitates optimal path stability and latency for the new path. Furthermore, we employ the FRR-BP mechanism to address node failures, which can maintain a short backup path rebuilding time in almost all network settings (except for the Ideal networking method with high constellation density).

Author Contributions: Conceptualization, Y.Z. and H.C.; Methodology, Y.Z. and H.C.; Software, Y.Z.; Validation, Y.Z.; Resources, Z.D.; Writing—original draft, Y.Z.; Writing—review & editing, H.C. and Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Zhejiang Province (No. LY24F020001), Ningbo Municipal Commonweal S&T Project (No. 2022S005), and Major S&T Projects of Ningbo High-tech Zone (No. 2022BCX05001).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Dr. Zhibin Douwas employed by the company The 54th Research Institute of China Electronics Technology Group Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Yi, X.; Sun, Z.; Yao, F.; Miao, Y. Satellite constellation of MEO and IGSO network routing with dynamic grouping. *Int. J. Satell. Commun. Netw.* **2013**, *31*, 277–302. [CrossRef]
2. Li, H.; Zhang, H.; Qiao, L.; Tang, F.; Xu, W.; Chen, L.; Li, J. Queue state based dynamical routing for non-geostationary satellite networks. In Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, 16–18 May 2018; pp. 1–8.
3. Wang, F.; Jiang, D.; Qi, S. An adaptive routing algorithm for integrated information networks. *China Commun.* **2019**, *16*, 195–206. [CrossRef]
4. Liu, J.; Luo, R.; Huang, T.; Meng, C. A load balancing routing strategy for LEO satellite network. *IEEE Access* **2020**, *8*, 155136–155144. [CrossRef]
5. Li, X.; Tang, F.; Chen, L.; Li, J. A state-aware and load-balanced routing model for LEO satellite networks. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
6. Ma, Y.; Peng, W.; Yu, W.; Su, J.; Wu, C.; Zhao, G. A distributed routing algorithm for LEO satellite networks. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, VIC, Australia, 16–18 July 2013; pp. 1367–1371.
7. 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]
8. Tang, F.; Zhang, H.; Yang, L.T. Multipath cooperative routing with efficient acknowledgement for LEO satellite networks. *IEEE Trans. Mob. Comput.* **2018**, *18*, 179–192. [CrossRef]
9. Liu, W.; Tao, Y.; Liu, L. Load-balancing routing algorithm based on segment routing for traffic return in LEO satellite networks. *IEEE Access* **2019**, *7*, 112044–112053. [CrossRef]
10. 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]
11. 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]
12. Yi, Z.; Quan, Z.; Jun, L.; Wei, L. The generation and update algorithm of routing table in satellite network. In Proceedings of the 2015 IEEE International Conference on Communication Problem-Solving (ICCP), Guilin, China, 16–18 October 2015; pp. 619–622.
13. Liu, Z.; Zhu, J.; Zhang, J.; Liu, Q. Routing algorithm design of satellite network architecture based on SDN and ICN. *Int. J. Satell. Commun. Netw.* **2020**, *38*, 1–15. [CrossRef]
14. Jiang, Y.; Wu, S.; Mo, Q.; Liu, W.; Wei, X. An Energy Sensitive and Congestion Balance Routing Scheme for Non-Terrestrial-Satellite-Network (NTSN). *Remote Sens.* **2023**, *15*, 585. [CrossRef]
15. Luo, Y.; Ning, Q.; Chen, B.; Zhou, X.; Huang, L. Software defined network-based multipath state-aware routing with traffic prediction in satellite network. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4819. [CrossRef]
16. Kempton, B.S. *A Simulation Tool to Study Routing in Large Broadband Satellite Networks*; Christopher Newport University: Newport News, VA, USA, 2020.