

Research Article

Load Balancing Routing Algorithm of Low-Orbit Communication Satellite Network Traffic Based on Machine Learning

Tie Liu , Chenhua Sun, and Yasheng Zhang

Satellite Communications and Broadcasting and Television Professional Department, China Electronics Technology Group Corporation Network Communications Research Institute, Shijiazhuang, 050011 Hebei, China

Correspondence should be addressed to Tie Liu; l970419t@st.btbu.edu.cn

Received 18 November 2021; Revised 9 December 2021; Accepted 13 December 2021; Published 31 December 2021

Academic Editor: Nima Jafari Navimipour

Copyright © 2021 Tie Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Satellite communication has become an important research trend in the field of communication technology. Low-orbit satellites have always been the focus of extensive attention by scholars due to their wide coverage, strong flexibility, and freedom from geographical constraints. This article introduces some technologies about low-orbit satellites and introduces a routing algorithm DDPG based on machine learning for simulation experiments. The performance of this algorithm is compared with the performance of three commonly used low-orbit satellite routing algorithms, and a conclusion is drawn. The routing algorithm based on machine learning has the smallest average delay, and the average value is 126 ms under different weights. Its packet loss rate is the smallest, with an average of 2.9%. Its throughput is the largest, with an average of 201.7 Mbps; its load distribution index is the smallest, with an average of 0.54. In summary, the performance of routing algorithms based on machine learning is better than general algorithms.

1. Introduction

Satellite communications have gradually gained popularity in the fields of radio and television and multimedia communications. On the one hand, people are beginning to require satellite networks to meet user requirements for delay, bandwidth, and fault tolerance. Services such as global positioning, remote communication, and telemedicine provided by satellite networks have long been closely related to people's daily lives. And it plays a huge role in promoting logistics and transportation, environmental monitoring, material exploration, navigation positioning, digital city, and other related fields. With the rapid development of satellite communication technology, satellites have obtained huge applications in acquiring and processing space information and related resources. In addition, with the development of nongeostationary orbit satellites and interstellar links, how to reduce the load of satellite networks has become one of the key factors that must be considered when designing satellite network routing algorithms. The research of machine learning in routing algorithms is still in the early stage. When it is actually used, it will bring surprises to people.

The satellite network has a highly dynamic topology. However, the distribution of ground users accessing its network is extremely uneven, and these characteristics will cause congestion in a local area of the satellite network. However, the surrounding satellite resources are vacant and wasted, causing data to be concentrated on certain paths or lost. It will increase the delay and packet loss rate of data packets. The high-speed movement of low-orbit satellites makes satellite networks have the characteristics of frequent topology changes, high possibility of link and node failure, unbalanced load distribution on the satellite, and limited resources on the satellite. These characteristics make the satellite network need better antimovement, antidestruction, and adaptive routing technology than the ground network. Therefore, it is necessary to design a routing algorithm suitable for the LEO satellite network separately. The main purpose of this paper is to verify that the traffic load balancing routing algorithm based on machine learning is superior to the traditional routing algorithm.

The innovation of this article is as follows: (1) This article introduces the testing methods of low-orbit satellite

performance parameters and introduces the general low-orbit communication satellite network traffic load balancing routing algorithm. (2) This article introduces a load balancing routing algorithm based on machine learning and compares its performance with three common routing algorithms through simulation experiments.

2. Related Work

With the continuous improvement of people's life needs, the requirements for communication satellites are getting higher and higher. Many scholars try to study more advanced satellite routing algorithms to improve the performance of passing satellites. Jiang proposed an energy consumption model based on link load, and used the bit energy consumption parameters of the network to measure the energy efficiency of the network. He also proposed an energy-saving minimum critical routing algorithm, which includes energy-efficient routing and load balancing. In order to further improve the energy efficiency of the network, Jiang et al. proposed an energy-saving multiconstraint rerouting algorithm to achieve maximum energy efficiency, but the algorithm lacks some detailed design [1]. Hui et al. proposed an energy-saving routing algorithm for wireless sensor networks based on unequal aggregation theory and connected graph theory. The new algorithm has been optimized and innovated in two aspects: cluster head election and cluster routing. The simulation results show that the new algorithm balances the energy consumption between sensor nodes and reduces the impact of the energy whole problem. It improves the link quality, greatly improves the reliability and efficiency of data transmission, and significantly extends the life of the network [2]. Kawecki and Schoeneich proposed a routing algorithm based on the mobility of nodes in a delay-tolerant network (DTN). DTN is characterized by the temporary or permanent lack of a continuous path between the source node and the destination node. Communication is accomplished through message transmission by intermediate nodes based on the store-carry-forward paradigm. This routing algorithm is based on the ability to use node mobility and its contact information. Its shortcoming is the lack of practical data support [3]. Erickson et al. rely on symmetric properties to build a single-path routing algorithm for DPillar. This algorithm can improve the average path length found, the total bottleneck throughput, and communication delay. And it emphasizes that the data center network should accept more stringent combination inspections, which can significantly improve its computing efficiency and performance. However, the algorithm is difficult to operate and not practical [4]. Fang et al. proposed a routing algorithm (GINS) based on geographic information and node selfishness. In order to select the forwarding node, GINS combines the forwarding willingness of the node with its geographic information to maximize the contact destination. GINS describe the message forwarding process as a 0-1 knapsack problem with allocation restrictions to meet the selfish needs of nodes. A large number of simulations have been carried out, and the results show that, compared with GRONE, GINS can achieve a higher transmission rate and

a lower number of hops. In addition, its management expense ratio is 25% lower than GRONE [5]. Wang et al. described the structure and function expectations of the energy router from the perspective of the network and improved the existing energy router design. They proposed a design of an electronic local area network energy routing algorithm based on graph theory. According to the characteristics of power transmission, they designed a routing algorithm with the lowest cost and proposed a power selection and routing design algorithm suitable for heavy load conditions. Both algorithms have been verified through case analysis, but the disadvantage is that there is no correlation analysis between the two algorithms [6]. Kumar and Dave proposed a beacon information-independent geographic routing algorithm called BIIR. By intelligently using the information collected by the vehicle in previous destination path discovery attempts, the algorithm reduces the number of broadcasts for forwarding data packets. Simulation results show that the algorithm is superior to the existing beacon-free routing protocol in terms of the average number of broadcasts for each packet forwarding, the packet delivery rate, and the end-to-end delay experienced by data messages. But its disadvantage is that the algorithm consumes more energy [7]. Mahalaxmi and Esther use the ant colony algorithm based on multiagent technology to improve the Internet of things routing algorithm and plan the routing algorithm to improve the packet delivery rate of the algorithm and avoid the damage of overlapping intersections by multiagent technology. With the improvement of efficiency, the delay will be reduced [8]. Kaneko and Bossard propose a method to construct $2n$ disjoint paths from a set s composed of $2n$ source nodes to a set D composed of $2n$ destination nodes. The nodes TN, K ($n \geq 1, K \geq 3$) in the n -dimensional k -element torus are formally described and evaluated. Then, the algorithm is formally described and evaluated [9].

3. Low-Orbit Satellite Network Load Balancing Routing Algorithm

This paper is aimed at proving that the traffic load balancing routing algorithm of the LEO communication satellite network based on machine learning has better performance than the traditional algorithm. Therefore, in this part, we first describe the general composition and key technologies of some LEO satellites and briefly introduce the process and evaluation indicators of the balanced routing algorithm, so as to facilitate the later experiments.

3.1. Key Technologies of Satellite Networks

3.1.1. Satellite Communication System

(1) *Satellite Communication System Composition.* As shown in Figure 1, the satellite communication system consists of three parts: space segment, ground segment, and user segment. The space segment is a constellation of satellites, which are scattered in the orbit of the satellite according to specific rules. The ground segment refers to the control

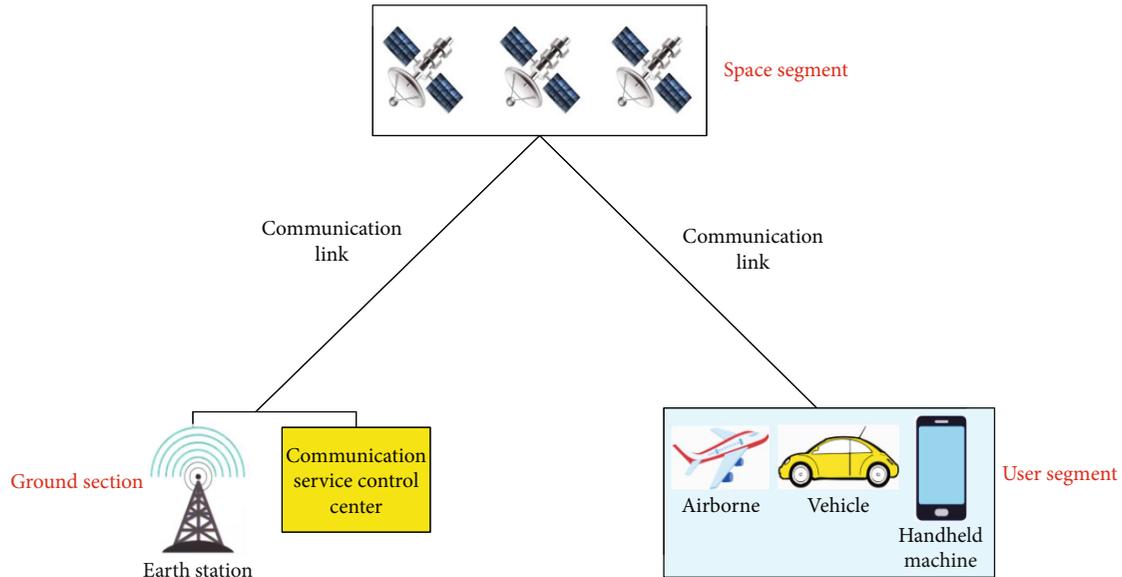


FIGURE 1: Satellite communication system composition diagram.

center, and the user segment refers to users, including various terminals.

(2) *Parameters of the LEO Satellite System.* The low-earth-orbit (LEO) satellite system can be composed of dozens to hundreds of satellites to achieve continuous global coverage of the network. Its satellites have a low orbit, and all satellite nodes fly around the earth at high speed. The LEO satellite communication system has a shorter average visibility time to the ground station, and the satellite-to-earth link communication delay established with the ground user is lower. Usually, handheld devices can be connected to the LEO satellite network. Table 1 lists typical LEO satellite systems such as Iridium, Teledesic, and Globalstar and their main system setting parameters.

3.1.2. *Satellite Constellation.* Satellite constellation is a collection of satellites designed according to the relative positions and geometric relationship rules between multiple satellites to complete complex communication tasks. Satellite constellation parameters can determine the size, shape, direction of the satellite orbit, and position of the satellite on the orbit. There are 6 main satellite constellation parameters, which can be divided into 3 categories. There are two constellation parameters that determine the size and shape of the orbit. It determines the three constellation parameters of the orbital position and one constellation parameter that determines the relative position. The schematic diagram is shown in Figure 2 [10].

The two constellation parameters for determining the size and shape of the orbit are the semimajor axis of the orbit and the eccentricity of the orbit, among which the apogee radius, apogee height, perigee radius, perigee height, and half focal length are related to the two constellation parameters; the three constellation parameters for determining the orbital position are ascending node right ascension, orbital inclination, and perigee angle; a constellation parameter that

determines the relative position of a satellite in orbit is the true near-earth angle.

3.1.3. *Satellite Network Routing.* With the development of satellite communication technology, a satellite network composed of multiple satellites or multilayer satellites has various forms of data transmission. It is particularly important to select a suitable route for data transmission from source to destination. Therefore, the design of routing algorithms is a key technology for continuous research in satellite networks. According to the different routing functions of each component in the satellite network, satellite routing can be divided into intersatellite routing, access routing, and border routing [11, 12]. Among them, due to the complexity of satellite network movement, intersatellite routing has always been a major and difficult point for many scholars. Designing a proper intersatellite routing algorithm is crucial to the routing performance of the entire satellite communication system. Figure 3 shows the composition of intersatellite routing.

3.2. LEO Satellite Network Routing Link

3.2.1. *Overview of Low-Orbit Satellite Links.* In a satellite constellation, each satellite can be abstracted as a node in the constellation, and communication between adjacent nodes is realized through a full-duplex link. This kind of link is the intersatellite link. In low-orbit satellites, intersatellite links are divided into intraorbit intersatellite links and inter-orbit intersatellite links according to whether two adjacent nodes are in the same orbital plane [13]. Figure 4 is a simple schematic diagram of the intersatellite link structure.

- (1) The intersatellite link refers to the link established by two adjacent satellites in the same orbit, and each satellite has two intraorbit links. Since the relative position of the satellites in the same orbit remains

TABLE 1: Typical LEO satellite system parameters.

System name	Iridium	Globalstar	Teledesic
Track height	778	1392	1383
Number of track planes	6	10	14
Number of satellites	69	275	51
Inter satellite link	2 in rail + 2 between rails	No	Yes
Orbit elevation	85.9°	54°	84.3°

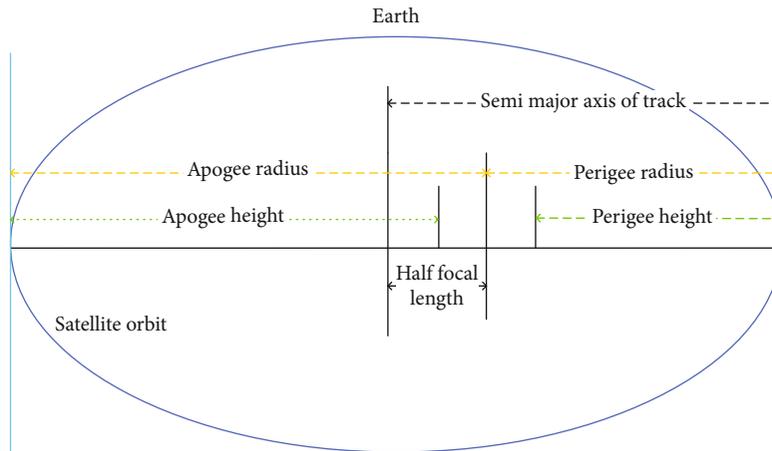


FIGURE 2: Schematic diagram of determining orbital parameters.

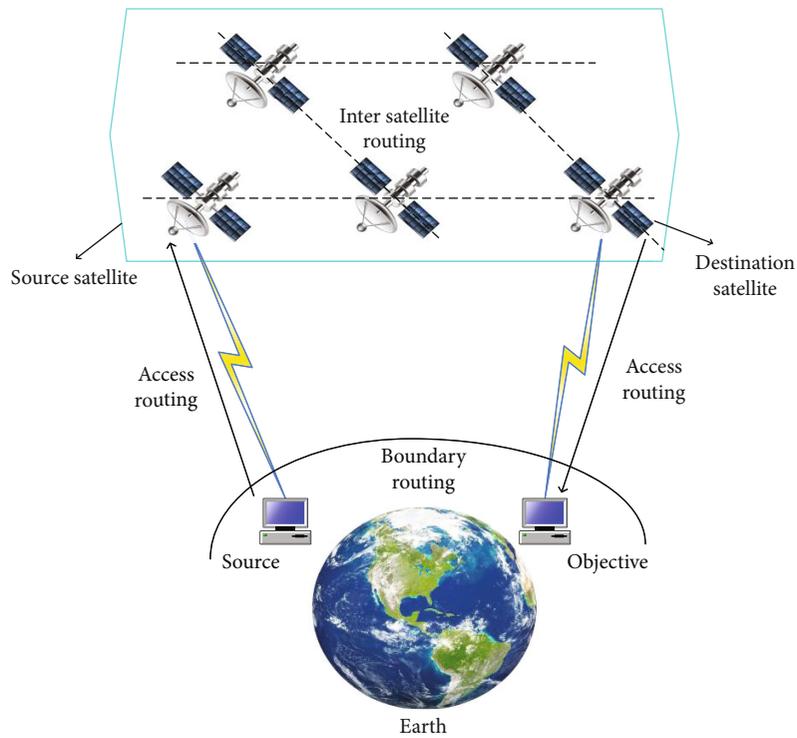


FIGURE 3: Satellite network routing composition.

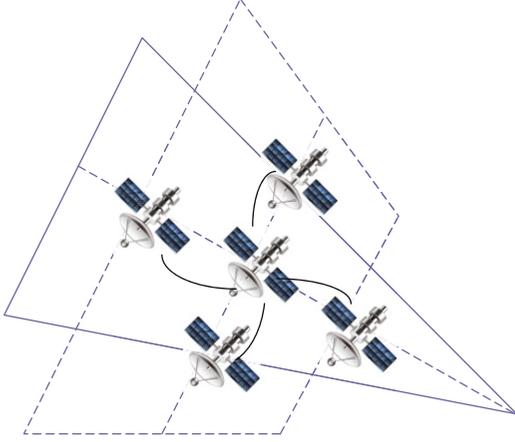


FIGURE 4: Simple structure of intersatellite link.

unchanged, the distance between adjacent satellites in the orbit will not change. The link length is expressed as follows:

$$L_{m,n} = \sqrt{2}R \sqrt{1 - \cos\left(\frac{2\pi}{M}\right)}. \quad (1)$$

Among them, $L_{m,n}$ is the distance between the m th star and the n th star in the same orbital plane. R is the orbit radius, and M is the number of satellites in the current orbit. If the orbit radius of the satellite movement and the number of satellites in the orbit are the same, then the intersatellite link distances in the orbit of the satellite are also the same [14].

(2) Interorbital intersatellite link

The interorbit intersatellite link is a link established by two adjacent satellites in two different but close orbits. When two adjacent satellites are in the polar region or on both sides of the reverse seam, it may be difficult to establish interorbit links due to unstable positions. Therefore, each satellite has 0-2 interorbit links. The expression of the link length can be divided into two cases where the orbital phase factor is equal to 0, that is, the interorbit link is parallel to the equator, and the orbital phase factor is not equal to 0, that is, the relative equator is inclined [15, 16], as shown in the following formula:

$$L_{\text{lat}} = \sqrt{2}R \sqrt{1 - \cos\left(\frac{2\pi}{N}\right)} \times \cos(\text{lat}) \quad F = 0, \quad (2)$$

$$L_{\text{lat}} = \sqrt{\left[\sqrt{2}R \sqrt{1 - \cos\left(\frac{2\pi F}{MN}\right)}\right]^2 + \left[\sqrt{2}R \sqrt{1 - \cos\left(\frac{2\pi}{N}\right)} \times \cos(\text{lat})\right]^2} \quad F \neq 0. \quad (3)$$

Among them, L_{lat} is the latitude of the satellite, R is the orbit radius, N is the number of orbits in the constellation, and F is the orbit phase factor.

3.2.2. Method for Setting Routing Link Weight of Low-Orbit Satellite Network. For low-orbit satellite networks, different satellite constellations have different network topologies. And in different time periods, the network topology of the satellite is not the same. These are the characteristics of satellite networks that are different from terrestrial networks, and these characteristics determine that the link weight calculation of satellite routing is more diverse. In order to allow satellite network routing to better adapt to the characteristics of the intersatellite network and improve the work efficiency of the satellite network, after link information is collected, these uncertain factors are considered to be optimized in the link weight. This is currently a better choice for satellite network routing. The following will introduce several different weighting schemes for intersatellite links.

(1) The intersatellite link distance is the weight

In satellite networks, the distance of intersatellite links is generally used as the link weight in satellite network routing. Such a routing algorithm is simple to implement. Through simple link information collection, the shortest path set is obtained. In such a system, an operating cycle of the satellite network is first divided into several time slices. The size of the time slice depends on the law of the dynamic topology of the satellite network. In each time slice, the system assumes that the topology of the entire satellite network is unchanged [17]. The topological structure divided in this way must take into account the visibility, direction angle, and coverage between the satellites and the ground. In this time slice, it is ensured that the topology change is maintained within a controllable range. Then, in each time slice, its topology is unchanged by default. According to the link state at the initial time of the time slice, the route is calculated at the initial time of the time slice. The distance between the satellite node m and the satellite node n is $\text{Dst}_{m,n}$, and the speed of light is V . The calculation formula for the weight $W_{m,n}$ of the intersatellite link $L_{m,n}$ is

$$W_{m,n} = \text{Del}_{m,n} = \frac{\text{Dst}_{m,n}}{V}. \quad (4)$$

When calculating the path, the total delay of the selected end-to-end link should be as small as possible. The calculation formula of the path cost is

$$\text{Metric} = \sum W_{m,n}. \quad (5)$$

The routing path calculated according to the above formula is the shortest path, but not the optimal path. The weight uses the distance of the intersatellite link, that is, to choose the closest path.

(2) Intersatellite link weight

Taking the link maintenance time into account in the routing algorithm of the satellite network, the transmission distance is still regarded as the main factor of the intersatellite link weight. Intersatellite link switching has a great

impact on routing stability. The intersatellite links between different orbital planes are regarded as temporary links. Therefore, reducing the number of intersatellite links between different orbital surfaces on the routing path can increase the continuity of the routing. The calculation formula for the weight of the intersatellite link between satellite node m and satellite node n is [18]

$$W_{m,n} = \text{Del}_{m,n} \times \frac{1}{\lambda}. \quad (6)$$

The parameter λ is used to indicate the connection capability of the intersatellite link, and the value range is 0 to 1.

(3) Routing energy consumption of the LEO satellite network

Assuming that the energy consumption of path L is E , the calculation formula is

$$E = \sum_{i=1}^n [V_i, V_{i+1}]. \quad (7)$$

That is, the energy consumption of path L is the sum of the energy consumption of the link formed between all adjacent nodes in the path. $E[V_i, V_{i+1}]$ is the link energy consumption between two adjacent nodes, and the calculation formula is deduced as

$$E[V_i, V_{i+1}] = \begin{cases} 7 \\ \min [7, 1/p^4] \end{cases}. \quad (8)$$

3.3. Routing Algorithm Performance Judgment Method

3.3.1. Traffic Load Judgment Mechanism

(1) *Factor Consideration.* When a satellite node makes a load decision, it should comprehensively consider the load status of the next-hop direction to be selected and the overall load status of the satellite node corresponding to the next-hop direction to be selected. At the same time, when calculating link load information, the long-term congestion changes in the network should be reflected as much as possible.

(2) *Specific Implementation Method.* The load decision mechanism provides a basis for each satellite node to make a reasonable routing decision. The satellite node obtains the main route and the alternative route of the current forwarding data packet by querying the routing table and judges the current link status according to the current link load information, determining the optional next-hop direction of the current data packet according to the link status of the primary and alternative next hops, and sending data according to the optional next hop. Among them, the link state can be divided into three states: relatively idle, relatively busy, and busy [19, 20]. When the primary selected next-hop link is relatively idle, the primary selected next hop is selected as the optional next hop regardless of the status of the alternative next-hop link; when the primary selected

next-hop link is relatively busy, if the alternative next-hop link is relatively idle, select the alternative next hop as the optional next hop; otherwise, select the primary selected next hop as the optional next hop; when the primary next-hop link is busy, if the alternative next-hop link is relatively idle or busy, the alternative next hop is selected as the optional next hop. If the alternative link is also busy, it indicates that the current two links are in congestion. Start the on-demand detour route to find the temporary optional next hop as data.

By calculating the link occupancy rate L , set reasonable thresholds X , Y , and $0 < X < Y < 1$ and compare the relationship between the link occupancy rate L and the threshold X , Y to determine the state of the intersatellite link. The details are as follows:

Relative idle: when $0 < L < X$, the current link is relatively idle.

Relatively busy: when $X < L < Y$, the current link is relatively busy.

Busy: when $L > Y$, the current link is busy [21].

The calculation of the link occupancy rate is shown in the formula, which represents the ratio of the weighted load of the current link to the capacity of a given link queue:

$$L = \frac{Q_{mn}(t)}{Q}. \quad (9)$$

Among them, Q is the link queue capacity, generally a given value, and $Q_{mn}(t)$ is the weighted link load from the m th star to its neighbor's n th star at time t . The weighted link load refers to the load information of the current link and the next hop satellite node corresponding to the current link. The overall load information, weighted link load $Q_{mn}(t)$ is calculated as follows:

$$Q_{mn}(t) = k \times V_{mn}(t) + N_n(t) \times (1 - k). \quad (10)$$

Among them, k is the weighting coefficient, $Q_{mn}(t)$ is the load information of the output link direction from the m th star to its neighbor's n th star at time t , and $N_n(t)$ is the overall load information of the n th star at time t .

In order to smooth the surge of short-term captains and better reflect the recent changes of captains, $V_{mn}(t)$ calculates the exponentially weighted moving average captains of the current link direction, as shown in the formula:

$$V_{mn}(t) = \frac{\mu V_{mn}(t-1) + (1-\mu)C_{mn}(t)}{1-\mu^t}. \quad (11)$$

Among them, $V_{mn}(t-1)$ is the exponentially weighted average length of the output link from the m th star to its neighbor n th star at the previous moment. $C_{mn}(t)$ is the output link team length from the m th star to its neighbor's n th star at time t , and μ is an exponential weighting coefficient.

The overall load information of the current satellite node refers to the average queue length of all output links of the

current satellite node. The average value of the exponentially weighted average queue length of all output link directions of the current satellite node is calculated, as shown in the formula:

$$N_n(t) = \frac{\sum_{N=1}^{N_n} V_{Nn}(t)}{N_n}. \quad (12)$$

Among them, $V_{Nn}(t)$ is the exponentially weighted moving average team length in the direction of the output link from the n th star to its neighboring n th star at time t ; N_n is the number of neighbors of the n th star with inter-orbital connections.

It can be seen from the above calculation that the key factor for calculating the weighted load of the output link is the determination of the weighting coefficient k . k reflects the proportion of the team leader in the current link direction in the output weighted load. The larger the value of k , the smaller the influence of the load status of the next-hop satellite node on the weighted load of the current output link. Therefore, choosing an appropriate value of k is very important [22, 23].

3.3.2. Other Performance Indicators. To evaluate the performance of a routing algorithm, conclusions can be drawn from the calculation of delay, packet loss rate, throughput, and routing overhead. Here are some calculation methods of routing performance [24].

(1) Average Delay. The average delay refers to the average delay of all data packets in the network from the source satellite node to the destination satellite node. The calculation of the average delay is shown in the formula:

$$T_{\text{avg}} = \frac{\sum_{i=1}^M T_i}{M}. \quad (13)$$

Among them, T_{avg} is the average delay, and M is the total number of data packets transmitted in the network. T_i is the total time delay of the i th data packet transmitted in the network from the source satellite node to the destination satellite node.

(2) Packet Loss Rate. As the name implies, the packet loss rate is the packet loss rate. It refers to the ratio of the number of lost data packets to the total number of sent data packets when the source satellite node in the network sends all data packets to the target satellite node. The calculation is as follows:

$$\text{loss} = \frac{M_{\text{loss}}}{M_{\text{send}}}. \quad (14)$$

Among them, loss represents the packet loss rate, M_{loss} is the number of data packets lost during the sending process, and M_{send} is the total number of data packets sent by the network.

(3) Throughput. Throughput refers to the maximum amount of data that the network can send without losing packets. Usually measured by the amount of data normally sent per unit time, the throughput is calculated as follows:

$$\text{TL} = \frac{M_s}{t} \times L. \quad (15)$$

In the formula, TL is the throughput, M_s is the number of data packets successfully delivered, L is the length of the transmitted data packet, and t is the simulation time.

(4) Routing Cost. The routing overhead refers to the routing control packets that need to be sent to implement the routing algorithm during the operation of the constellation system. When analyzing routing performance, the number of routing control packets required to successfully transmit 1 million data packets is used to measure the routing overhead of the constellation system. The calculation formula of routing cost is as follows:

$$K = \frac{M_k}{M_s} \times 1000000. \quad (16)$$

Among them, K is the routing overhead, M_k is the total number of routing control packets, and M_s is the total number of data packets successfully transmitted in the network.

(5) Load Distribution Index. The load distribution index refers to the degree of load distribution in the satellite network, and the value range is (0,1). The larger the value, the more dispersed the load in the satellite network, and the better the balance performance. The smaller the value, the more concentrated the load in the satellite network [25, 26]. The calculation of the load distribution index is as follows:

$$F = \frac{(\sum_{i=1}^n x_i)^2}{\sum_{i=1}^n x_i^2} \times \frac{1}{n}. \quad (17)$$

Among them, F is the load distribution index, n is the total number of intersatellite links, and x_i is the total number of data packets passed by the i th intersatellite link during the simulation time.

The evaluation indexes of the above routing methods are relatively important. The average delay evaluates the load balancing time of the algorithm, the packet loss rate evaluates the reliability of the algorithm network, the throughput reflects the ability to send data, and the load distribution index reflects the balancing ability of the network. These indexes jointly evaluate the load balancing performance of the routing algorithm.

3.4. Low-Earth Satellite Network Load Balancing Algorithm. The significance of the load balancing algorithm is to allow satellite nodes to distribute data packets to adjacent lighter-loaded nodes for forwarding as much as possible when

congestion occurs. This can reduce the queuing delay of data packets at the satellite node. The algorithm adjusts the cost between links in combination with the regional traffic and uses the shortest path priority idea to find the path. The following will introduce the algorithm from several aspects such as traffic estimation, congestion judgment, and cost transformation [27].

3.4.1. Flow Estimation. The distribution of ground traffic on satellite networks is very uneven. It is mainly concentrated in some important cities on the mainland but is quite small in the ocean. This leads to a situation in which the traffic is concentrated in a certain area at certain moments in the satellite network communication process, and the satellites in this area are often congested, but the surrounding satellites are not fully utilized [28]. The low-orbit satellite network load balancing algorithm can adjust the link cost of the link according to the congestion state of the satellite link and the load state of the previous time period when the satellite link is congested. This adjustment process is for the entire network. Data passing through congested areas will bypass these load areas, and a lighter-loaded path needs to be selected for data transmission, thereby balancing the flow of the entire network.

3.4.2. Congestion Judgment. In a satellite network, a satellite maintains multiple links, these links are associated with different neighboring nodes, and the congestion state is also different. In order to improve link utilization, we monitor congestion on each link of the satellite. In order to improve the accuracy of monitoring, the calculation method to determine the link load is as follows:

$$\rho = \frac{kq \times I + \lambda}{\omega \times C \times t}, \quad (18)$$

where t is the time period for link load calculation, λ is the amount of data transmitted in t , I is the average queue length of the link in t , kq is the queue reduction rate, ω is the target utilization rate of the link, and C refers to the data transmission capability of the link, setting a congestion threshold θ for each link.

3.4.3. Cost Transformation. The load cost is obtained after comprehensive evaluation of link transmission delay and link congestion. In the case of congestion encountered in actual transmission, the congested link can be bypassed to obtain network traffic balance. The calculation formula of load cost is

$$C_p = \sum_{i=1}^n (D_i + F_i). \quad (19)$$

Among them, n is the link number, D_i is the link delay factor, and F_i is the link load factor.

When $\rho \leq \theta$, $F_i = 0$.

When $\rho > \theta$,

$$F_i = u \times \frac{q(w_m + w_n)}{\bar{t}}, \quad (20)$$

q is the queue size of the link, \bar{t} is the average value of t , w_m and w_n are the estimated traffic values of the satellites at both ends of the link, and u is the reduction factor of the function.

3.4.4. Algorithm Flow. The algorithm is based on the satellite virtual topology strategy and performs low-orbit link cost conversion on the topology snapshot. According to the congestion of the link, finding a suitable path for the nodes in the satellite network to perform data distribution. Therefore, by using the predictability and periodicity of the satellite constellation operation, the system operation time is divided into several identical time periods. In each time period, n gaps are divided, and the intersatellite link is judged for link overload in each time slot [29]. The specific process is shown in Figure 5.

The flow of this algorithm is as follows: after the satellite status is updated, calculate the link overload, judge whether the load factor is greater than a , adjust the LEO link cost if yes, reset the LEO link cost if no, then plan the path, and finally update the route. When this algorithm selects the path, starting from the congestion of the entire network, the weight on the idle link is small, and the idle link is selected to transmit data during routing. The algorithm does not need to know the load of adjacent satellites, and the additional storage overhead is small.

4. Routing Algorithm Design and Simulation Experiment Based on Machine Learning

4.1. Routing Optimization Algorithm Based on Reinforcement Learning

4.1.1. Algorithm Design. This experiment will design a routing optimization algorithm based on reinforcement learning. It includes algorithm input state design, output action design, and reward function design.

(1) *Input State Design.* Considering the measurement overhead of the traffic matrix and the feasibility of the algorithm, this chapter uses the link utilization rate of the entire network as the input state of the reinforcement learning algorithm. For time t , the input state s_t is an m -dimensional vector (m represents the number of links in the network topology), and the vector represents the link utilization of each link. In addition, using the flow matrix and the link utilization rate as the algorithm, the performance difference of the routing optimization between the two input states is studied. This chapter also continues to study the reinforcement learning routing optimization method using the traffic matrix as the input. For the routing method where the input is a flow matrix, the input state s_t is a n^2 -dimensional vector (n represents the number of network nodes), and each value in the vector represents the flow of each flow.

(2) *Output Action Design.* This chapter mainly uses reinforcement learning algorithms for business routing planning. It is hoped that the algorithm model will select the best routing scheme for each flow by identifying the characteristics of the service flow at different moments. In order to enable the algorithm model to learn the direct mapping

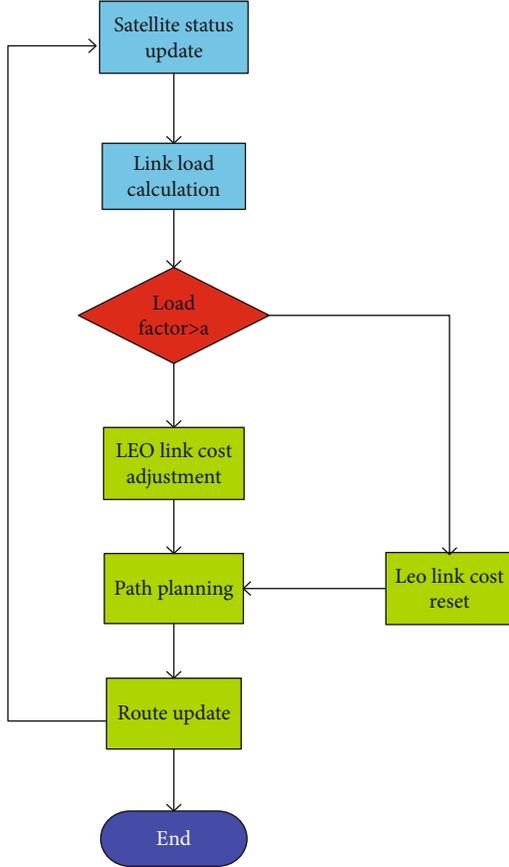


FIGURE 5: Algorithm flow chart.

relationship between traffic information and routing selection, this chapter will reinforce the learning algorithm model output action a_t to represent routing schemes of different flows. The reinforcement learning environment selects the routing scheme with the highest probability of each flow for routing according to the predicted action a_t .

(3) *Reward Function Design.* The role of the reward function in the reinforcement learning algorithm is mainly used to feed back the quality of the action taken by the agent. Then, the agent updates the neural network parameters according to the maximization reward as the optimization goal. Therefore, the quality of the constructed reward function will directly affect the performance of the reinforcement learning algorithm model for routing optimization problems. The optimization goal of this chapter is to make the network load more balanced and reduce network link congestion. Therefore, in this chapter, the reward function is set according to the link utilization, as shown in the following formula:

$$P_e = \frac{1}{C_e} \cdot \sum_i^{n^2} d_i \sigma_i, \quad (21)$$

$$\bar{P}_e^t = \frac{\sum_{e \in E} P_e^t}{N}, \quad (22)$$

$$R_t = \frac{\sum_{e \in E} (P_e^{t-1} - \bar{P}_e^{t-1})^2}{\sum_{e \in E} (P_e^t - \bar{P}_e^t)^2}. \quad (23)$$

Among them, P_e represents the link utilization rate of link e , d_i represents the size of the i th flow in the network, σ_i represents whether the i th flow passes through link e , C_e represents the capacity of link e , P_e^t represents the link utilization rate of link e in the t round of training, and \bar{P}_e^t represents the average value of P_e^t . According to the reward function A , if the variance of the link utilization rate of the entire network is lower than the variance of the link utilization rate of the entire network in the previous round of training, the agent can obtain a larger reward value. That is, the more balanced the link load, the greater the reward value obtained. Finally, the agent adjusts the parameters of the neural network according to these rewards to maximize the rewards it can obtain.

4.1.2. *Algorithm Model.* This experiment selects the existing DDPG reinforcement learning algorithm to learn routing strategies under different business models as the routing optimization algorithm model in this chapter. This is a strategy learning method that integrates deep learning neural networks into DPG. It is used as a basic algorithm for routing optimization. DDPG is a strategy learning method for continuous action and high-dimensional design proposed in recent years. DDPG predicts the best routing plan in the current traffic scenario based on the input status at the current moment, thereby achieving fine-grained routing control. The training process is shown in Figure 6.

During the training process, DDPG updates the critic network parameters through the gradient of the loss function and then updates the actor network parameters through the policy gradient. After updating the critic network and actor network parameters, using the soft update method to update the parameters in the corresponding target network. It can make the target actor network and the target critic network learn slowly from the parameters in the actor network and the critic network.

4.2. Simulation

4.2.1. *Environmental Configuration.* In order to verify the effectiveness of the algorithm, the network simulation software NS2 is used for simulation and performance analysis. The simulation uses the constellation parameters shown in Table 2, and the algorithm uses the polar orbit Iridium constellation.

Some parameters in the algorithm are shown in Table 3.

4.2.2. *Comparison Algorithm Selection.* In order to better evaluate routing optimization algorithms based on reinforcement learning, commonly used DSP algorithms, LAOR algorithms, and DRA algorithms are selected for simulation analysis and comparison. The DSP algorithm adopts static routing, establishes the forwarding table of each node during system initialization, stores the shortest path information, and does not support dynamic storage management; the

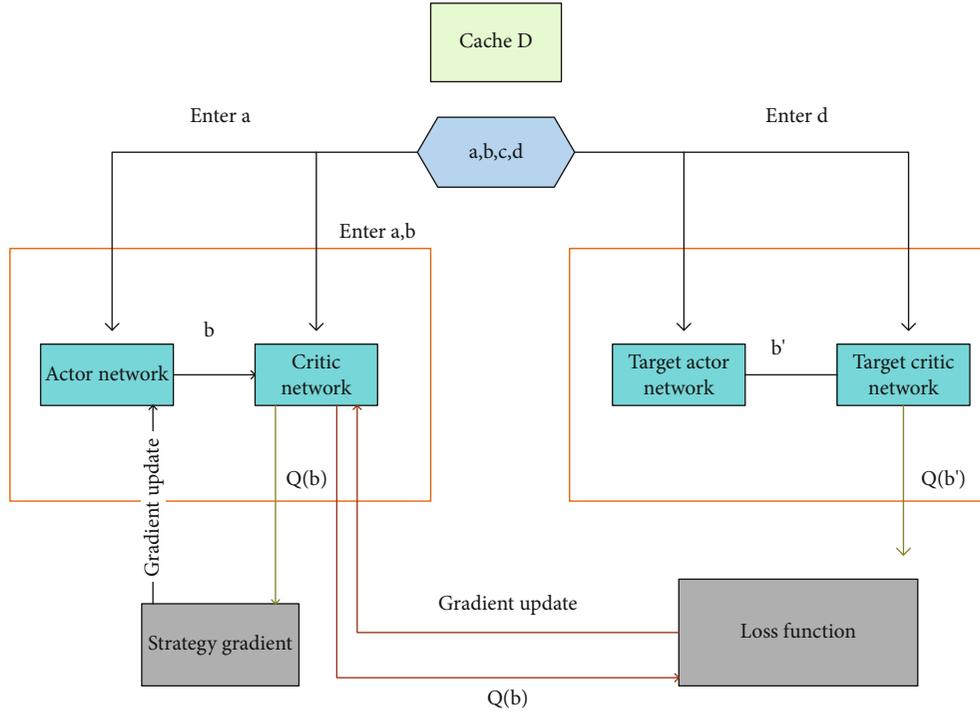


FIGURE 6: DDPG flow chart.

TABLE 2: Simulation constellation parameters.

	LEO
Model	Iridium
Number of tracks	6
Number of satellites	68
Height	810 km

TABLE 3: Design table of related parameters.

Routing topology update time	10 s
Instantaneous queue collection time	0.1 s
Queue reduction rate	0.9
Queue target utilization	0.8
Load threshold	0.8

LAOR algorithm is an algorithm formed by introducing the on-demand idea of terrestrial wireless ad hoc network into LEO satellite network. The purpose of the algorithm is to minimize the end-to-end delay and delay jitter and reduce the control overhead at the same time; the DRA routing algorithm is connectionless and distributed. It can make routing choices independently for each packet. It can avoid congested areas by making local routing decisions; the main algorithm DDPG algorithm of this experiment has been introduced above and will not be repeated here.

4.2.3. Experimental Purpose. The purpose of this experiment is to test the size of several indexes of the machine learning-

based routing algorithm, namely, the DDPG algorithm, and compare it with several other traditional algorithms to test whether it has advantages in performance.

4.3. Experimental Realization and Analysis. The overall weight W of the communication service is gradually increased from 1 to 2, that is, the data transmission rate is gradually increased, and the balanced performance of each algorithm is analyzed and compared.

4.3.1. Average Delay. In this paper, by measuring the difference between the time when the data packet arrives at the destination node and the time when it is generated in the simulation, the total delay of the current data packet transmission is obtained. The average delay is obtained by calculating the average value of the total delay of all data packets. The average delay data of the four different algorithms is shown in Figure 7.

It can be seen from Figure 7 that the average delay of the four algorithms all increases with the increase of the service weight W . When the weight is less than 1.6, the increase in the average delay along with the weight is relatively large, and when it is greater than 1.6, the increase in the average delay becomes smaller. Among the four algorithms, the DDPG algorithm based on reinforcement learning has the smallest average delay regardless of the weight. The average value of the average delay under different weights is 126 ms, and the average delay of the LAOR algorithm is the largest. This can verify that the DDPG algorithm can effectively balance the load and reduce the queuing delay.

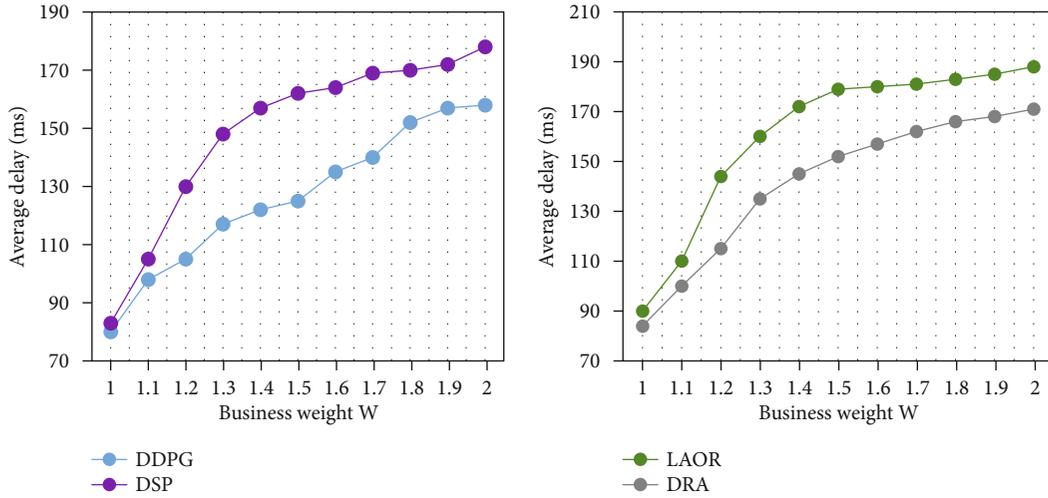


FIGURE 7: Lower average delay data of different algorithms.

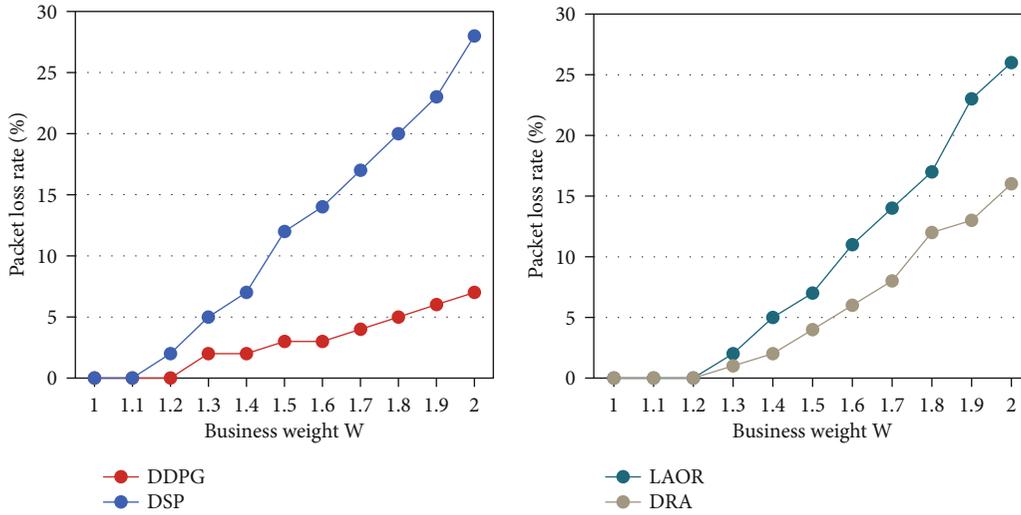


FIGURE 8: Lower packet loss rate data of different algorithms.

4.3.2. *Packet Loss Rate.* The size of the data packet loss rate shows the stability of the network to a certain extent. Figure 8 shows the packet loss rate data of each algorithm with different weights.

As can be seen from Figure 8, as the service weight increases, the service carried by the satellite increases, and the degree of unbalanced load distribution is aggravated. When the buffer queue overflows, the four algorithms will cause packet loss to a certain extent. As the weight W increases, the packet loss rate of the four algorithms also increases. Among them, the increase of DDPG is the smallest, and the average value of its packet loss rate is also the smallest. The average packet loss rate under different weights is 2.9%. Based on the reinforcement learning DDPG algorithm, because the path is adjusted in time according to the load status, the network can effectively reduce the possibility of queue overflow.

4.3.3. *Data Throughput.* Figure 9 is a comparison result of the throughput of the four algorithms under different service weights.

It can be seen from Figure 9 that as the service weight increases, the throughput of the four algorithms gradually increases. Among them, the DDPG algorithm has the highest average throughput under different weights, and its value is about 201.7Mbps. This proves that the DDPG algorithm based on reinforcement learning can sense the load status in time. And it can reasonably balance the load of each satellite node and effectively control the queuing delay of data packets. Therefore, its packet loss rate is very low, thereby improving throughput.

4.3.4. *Load Distribution.* Using the formula in 2.3 to calculate the load distribution, the obtained load distribution indexes of the four algorithms are shown in Figure 10.

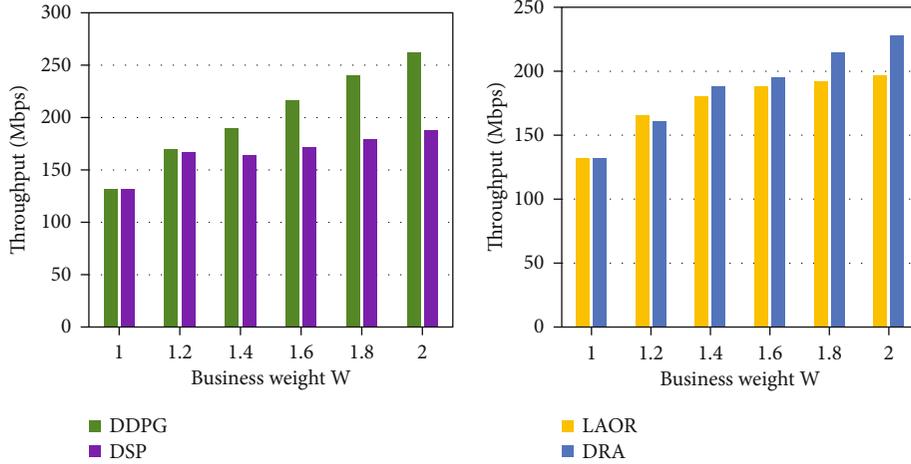


FIGURE 9: Lower throughput data of different algorithms.

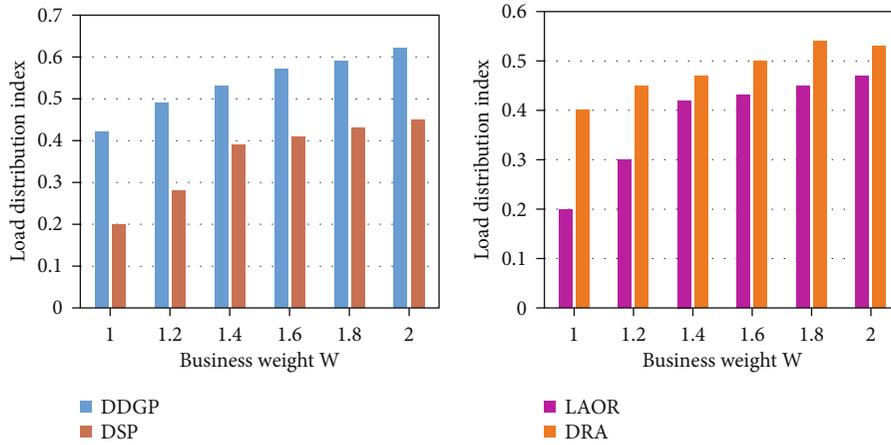


FIGURE 10: Load distribution index of different algorithms.

It can be seen from Figure 10 that the load distribution index of the DSP algorithm is the smallest, while the DDPG load distribution index is the largest, and the average load distribution index under different weights is 0.54. This is because the equalization performance of the DDPG algorithm is the best. It can adjust the path in time and spread congestion information across the entire network, reducing the traffic sent from satellites across the entire network to congested links. In addition, as the service weight increases, the load distribution index of the DDPG algorithm increases rapidly, which shows that this algorithm can distribute the load from the congested link to other links in time.

From the above four performance test experiments, we can know that the routing algorithm based on machine learning, i.e., DDPG algorithm, has the smallest average delay, the smallest packet loss rate, the highest data throughput, and the smallest load distribution index, which can verify that the performance of the DDPG algorithm is the best, and prove the superiority

of the performance of the routing algorithm based on machine learning.

4.4. Discussion. Low-orbit satellites have small time delays and low terminal energy consumption, which has great communication advantages. So in the satellite communication system with an intersatellite link, effective routing algorithm technology must be designed. At present, with the rapid development of communication technology, Internet applications, and cloud computing technology, communication networks have experienced an explosive business increase. The routing optimization algorithm based on machine learning can learn a good mapping relationship between traffic characteristics and routing strategies based on historical data. Then, use the learned knowledge to quickly make routing decisions based on changes in business characteristics. It can realize adaptive service routing scheduling and management, so the use of machine learning algorithms for network service routing optimization has attracted more and more attention.

Now, it has been extensively studied in the field of communication networks.

5. Conclusion

In this paper, the low-orbit communication satellite network traffic load balancing routing algorithm is researched. This article introduces the meaning of low-orbit satellites and its routing algorithm, as well as the performance judgment method of the algorithm. And this article introduces a traffic load balancing routing algorithm combined with machine learning and then studies some basic performance of this algorithm, as follows: (1) The basic composition, structure, and common parameters of the LEO satellite system are introduced. (2) The intersatellite links of low-orbit satellites and their weights are introduced. (3) This article explains some methods for determining the performance of the LEO satellite routing algorithm and explains the workflow and calculation methods of the general equilibrium routing algorithm. (4) A routing algorithm based on machine learning was introduced, and simulation experiments were carried out. The performance of this algorithm was compared with the performance of three commonly used low-orbit satellite routing algorithms. (5) Through simulation experiments, this article draws a conclusion: in terms of average delay, packet loss rate, throughput, and load distribution performance, routing algorithms based on machine learning are superior to other algorithms. The routing algorithm based on machine learning has the smallest average delay, and the average value is 126 ms under different weights. Its packet loss rate is the smallest, with an average of 2.9%. The throughput is the largest, with an average of 201.7 Mbps. The load distribution index is the smallest, with an average value of 0.54. The experiment in this paper is generally successful, but the experimental software is not completely designed in the experimental design part. In the future, the experiment can be designed more completely, and several performance tests can be done in the implementation part of the experiment to improve the integrity of the experiment.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- [1] D. Jiang, P. Zhang, Z. Lv, and H. Song, "Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1437–1447, 2016.
- [2] H. Xia, R.-h. Zhang, J. Yu, and Z.-k. Pan, "Energy-efficient routing algorithm based on unequal clustering and connected graph in wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 23, no. 2, pp. 141–150, 2016.
- [3] M. Kawecki and R. O. Schoeneich, "Mobility-based routing algorithm in delay tolerant networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, 2016.
- [4] A. Erickson, A. E. Kiasari, J. Navaridas, and I. A. Stewart, "An optimal single-path routing algorithm in the datacenter network DPillar," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 3, pp. 689–703, 2017.
- [5] F. Lu, J. Li, S. Jiang, Y. Song, and F. Wang, "Geographic information and node selfish-based routing algorithm for delay tolerant networks," *Tsinghua Science & Technology*, vol. 22, no. 3, pp. 243–253, 2017.
- [6] R. Wang, J. Wu, Z. Qian, Z. Lin, and X. He, "A graph theory based energy routing algorithm in energy local area network," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3275–3285, 2017.
- [7] N. Kumar and M. Dave, "BIIR: a beacon information independent VANET routing algorithm with low broadcast overhead," *Wireless Personal Communications*, vol. 87, no. 3, pp. 869–895, 2016.
- [8] G. Mahalaxmi and R. K. Esther, "Multi-agent technology to improve the internet of things routing algorithm using ant colony optimization," *Indian Journal of Science & Technology*, vol. 10, no. 31, pp. 1–8, 2017.
- [9] K. Kaneko and A. Bossard, "A set-to-set disjoint paths routing algorithm in tori," *International Journal of Networking & Computing*, vol. 7, no. 2, pp. 173–186, 2017.
- [10] L. ZENG, T. PAN, X. JIANG, and T. WATANABE, "An efficient highly adaptive and deadlock-free routing algorithm for 3D network-on-chip," *Ieice Transactions on Fundamentals of Electronics Communications & Computer Sciences*, vol. E99.A, no. 7, pp. 1334–1344, 2016.
- [11] L. Qiao, Y. Li, D. Chen, and S. Serikawa, "A survey on 5G/6G, AI, and robotics," *Computers & Electrical Engineering*, vol. 95, article 107372, 2021.
- [12] S. Avallone and A. Banchs, "A channel assignment and routing algorithm for energy harvesting multiradio wireless mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1463–1476, 2016.
- [13] W. Wang, H. Shi, D. Wu et al., "VD-PSO: an efficient mobile sink routing algorithm in wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 537–546, 2017.
- [14] W. Jia and Z. Chen, "Sensor communication area and node extend routing algorithm in opportunistic networks," *Peer-to-Peer Networking and Applications*, vol. 11, no. 8, 2018.
- [15] T. Ren, Z. Jiang, X. Cai et al., "A dynamic routing optimization problem considering joint delivery of passengers and parcels," *Neural Computing and Applications*, vol. 33, no. 16, pp. 10323–10334, 2021.
- [16] Y. Meraihi, D. Acheli, and A. Ramdane-Cherif, "QoS multicast routing for wireless mesh network based on a modified binary bat algorithm," *Neural Computing and Applications*, vol. 31, no. 7, pp. 3057–3073, 2019.
- [17] Y. Ding, R. Chen, and K. Hao, "A rule-driven multi-path routing algorithm with dynamic immune clustering for event-driven wireless sensor networks," *Neurocomputing*, vol. 203, p. 139, 2016.
- [18] F. Li, M. Xiong, L. Wang, H. Peng, J. Hua, and X. Liu, "A novel energy-balanced routing algorithm in energy harvesting sensor networks," *Physical Communication*, vol. 27, pp. 181–187, 2018.

- [19] C. Zhang, X. Wang, F. Li, and M. Huang, "NNIRSS: neural network-based intelligent routing scheme for SDN," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6189–6205, 2019.
- [20] T. Senthil and B. Kannapiran, "ECTMRA: energy conserving trustworthy multipath routing algorithm based on cuckoo search algorithm," *Wireless Personal Communications*, vol. 94, no. 4, pp. 2239–2258, 2017.
- [21] S. Hamrioui and P. Lorenz, "Bio inspired routing algorithm and efficient communications within IoT," *IEEE Network*, vol. 31, no. 5, pp. 74–79, 2017.
- [22] B. T. Thuan, L. B. Ngoc, and K. Kaneko, "A stochastic link-fault-tolerant routing algorithm in folded hypercubes," *Journal of Supercomputing*, vol. 74, no. 10, pp. 5539–5557, 2018.
- [23] L. Wu, C.-H. Chen, and Q. Zhang, "A mobile positioning method based on deep learning techniques," *Electronics*, vol. 8, no. 1, p. 59, 2019.
- [24] S. El Alaoui and B. Ramamurthy, "EAODR: a novel routing algorithm based on the modified temporal graph network model for DTN-based interplanetary networks," *Computer Networks*, vol. 129, pp. 129–141, 2017.
- [25] Z. Zhang, W. Serwe, J. Wu, T. Yoneda, H. Zheng, and C. Myers, "An improved fault-tolerant routing algorithm for a network-on-chip derived with formal analysis," *Science of Computer Programming*, vol. 118, p. 24, 2016.
- [26] Y. R. B. Al-Mayouf, N. F. Abdullah, M. Ismail, S. M. Al-Qaraawi, O. A. Mahdi, and S. Khan, "Evaluation of efficient vehicular ad hoc networks based on a maximum distance routing algorithm," *Eurasip Journal on Wireless Communications & Networking*, vol. 2016, no. 1, 2016.
- [27] W. Xu, M. Jiang, F. Tang, and Y. Yang, "Network coding-based multi-path routing algorithm in two-layered satellite networks," *IET Communications*, vol. 12, no. 1, pp. 2–8, 2018.
- [28] J. Li, X. Hou, D. Su, and J. D. D. Munyemana, "Fuzzy power-optimised clustering routing algorithm for wireless sensor networks," *Wireless Sensor Systems, IET*, vol. 7, no. 5, pp. 130–137, 2017.
- [29] J. Zou, F. Sadiq, and A. Gulnaz, "PERA: priority-based energy-efficient routing algorithm for WBANs," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4737–4753, 2017.