



A DQN-Based Routing Algorithm for Load Balancing in LEO Satellite Networks

Ziqi Sun¹, Jing Meng², Ruofei Ma¹(✉), Gongliang Liu¹(✉), Xiaoling Che³,
and Guodong Kang⁴

¹ Harbin Institute of Technology, Weihai 264200, Shandong, China
{maruofei, liugl}@hit.edu.cn

² Qian Xuesen Laboratory, CAST, Beijing 100094, China

³ DFH Satellite Computer, LTD, Beijing 100094, China

⁴ China National Space Administration Earth Observation and Data Center, Beijing 100101,
China

Abstract. With the rapid development and popularization of low orbit satellite networks, their importance in global communication coverage and data transmission continues to be highlighted. However, due to the particularity of LEO (Low Earth Orbit) satellite networks, such as highspeed operation, limited channel capacity, and large latency, poses challenges to network load balancing and routing scheduling. The current routing algorithms for low orbit satellite networks mostly use traditional methods such as Dijkstra's shortest path, which lack real-time adaptability and intelligence to the high mobility of the network. Therefore, based on the deep reinforcement learning intelligent routing algorithm of DQN (Deep Q Network) has become a new approach to solve this problem. By combining the multidimensional information of LEO satellite networks and the advantages of DQN, this study proposes a new load balancing intelligent routing algorithm that comprehensively considers inter satellite link connectivity and link load conditions, and jointly optimizes multiple objectives. The aim is to improve network performance, reduce latency, and provide technical support and solutions for the future development of low orbit satellite networks.

Keywords: LEO Network · Routing Algorithm · Load Balancing · DQN

1 Introduction

Due to the lack of real-time adaptability and intelligence to network dynamic changes in traditional satellite routing algorithms, more scholars are focusing on novel satellite routing algorithms to cope with the challenges brought by future complex network structures and the increasing amount of data in satellite communication. The routing algorithm for LEO satellite networks is mainly used to solve the problem of efficient and reliable end to end data transmission in LEO satellite networks. Routing algorithms need to plan the optimal path in satellite networks by moving data packets from the source satellite to the destination satellite in this problem. The measurement of the optimal path

can be based on various indicators, such as latency, link bandwidth, link stability, link state, satellite load, hop count, etc. Considering the high mobility in satellite network topology, traditional routing techniques have the following drawbacks: (1) They cannot fundamentally avoid link and connection switching, as well as a series of switching control and rerouting calculation problems caused by them. (2) The computational cost is relatively high and it is difficult to implement on satellite, usually requiring the assistance of ground systems for calculation. (3) To achieve integration with ground IP networks, a series of intermediate processes such as protocol conversion and data format conversion are required, which will bring additional time and processing costs, making the system implementation more complex. Therefore, current research on low orbit satellite routing protocols mainly focuses on non-connected routing algorithms.

Some scholars have proposed some novel routing algorithms from different perspectives. A novel routing algorithm was proposed in reference [1], which takes into account the mobility and periodicity of satellite constellations. The algorithm predicts the network topology structure based on ephemeris, and transforms dynamic topology into static topology within multiple time slots. Based on this, consider the current state information of the channel, perceive global network information, and calculate routing tables. Reference [2] analyzed the characteristics of ground based ad hoc networks and proposed a routing algorithm suitable for satellite networks, improving the stability of satellite networks. Global network information is collected to calculate a routing table and make routing decisions based on it.

In recent years, software defined networking(SDN) [3] technology has been developing rapidly. Some scholars believe that this technology can be used to achieve unified control and scheduling of satellite nodes. The prominent advantage of SDN architecture is the separation of data forwarding and control nodes, which reduces the computational pressure on satellite nodes. The control node obtains global network information and makes routing decisions by regularly collecting information from forwarding nodes. Bertaux [4] elaborated on the advantages of satellite networks based on SDN architecture. Papa [5] proposed a mathematical framework to quantify the migration and reconstruction costs of satellite networks based on SDN architecture. Hu [6] optimized the network structure of traditional constellations for multicast routing in large multi-layer satellite networks. The algorithm combines relevant ideas from graph theory to propose a new multicast routing strategy.

In recent years, machine learning has gradually entered people's field of vision. Artificial intelligence provides a new research direction for satellite network routing algorithms due to its outstanding performance in algorithm performance. In research on satellite routing, Bominao [7, 8] advanced an efficient routing strategy using deep learning to quantify network information and enhance network intelligence. Utilizing deep belief networks to extract features from multidimensional network information and achieve routing decisions between source and destination nodes. Zuo [9] proposed an intelligent routing strategy with delay as the optimization objective, and LEO satellites make routing decisions through DQN networks. Liu [10] proposed an energy aware distributed routing strategy to address the node energy issue in satellite networks. Link status and energy information are transmitted between connected nodes. The algorithm optimizes for transmission latency and node energy, and uses deep reinforcement learning for

routing selection. Wang [11] proposed using convolutional neural network (CNN) models to learn historical traffic data in satellite networks, and adjusting its output results based on the QoS of business data, which has better traffic control performance and routing flexibility.

Machine learning brings new opportunities for network routing. Combining SDN technology with machine learning can achieve unified control of satellite networks and adapt to its dynamic characteristics. Using machine learning for network routing optimization can adapt to the dynamics of large-scale networks, capture real-time network state information (such as latency and congestion state), and predict the impact of real-time routing on the network. Therefore, this article considers small low orbit satellite constellations and proposes an intelligent routing strategy for LEO satellite network load balancing based on DQN. It comprehensively considers the dynamics of inter satellite links and link load conditions, optimizes network performance, reduces latency, and improves network throughput, providing new solutions for the future development of low orbit satellite networks. By comparing with traditional algorithms, the method proposed in this paper has significant performance improvements.

2 System Model

2.1 LEO Satellite Network Model

The model scenario currently considers small low orbit satellite constellations and deploys an 8x8 single layer mesh constellation. The high mobility of nodes in low Earth orbit satellite networks is widely present. The network topology will change over time, and the connectivity and distance between satellites will also constantly change. Therefore, using the time slicing method, it is assumed that a satellite network in a time slot can be considered as a static topology, while the mesh structure assumes that each satellite node can have at most two same orbit links and two adjacent orbit links. Based on this, the influence of polar regions and dynamic topology on the connectivity between newer satellites is considered. Consider routing decisions within different time slots here and set a duration to update the network topology. Data packets are selected from the source node through each hop routing to reach the destination satellite node. During the process, each satellite node can obtain the link connection status and distance to each node in that time slot. The system model is displayed in Fig. 1.

2.2 Satellite Communication Model

The communication channel between low Earth orbit satellites can be considered as an additive Gaussian white noise channel, which is a linearly increasing broadband noise with a constant spectral density and amplitude of Gaussian distribution. The main consideration in this channel is free space path loss. When a data packet is transmitted between various nodes, this loss can be expressed as.

$$L = \left(\frac{c}{4\pi D_{ef}} \right)^2 \quad (1)$$

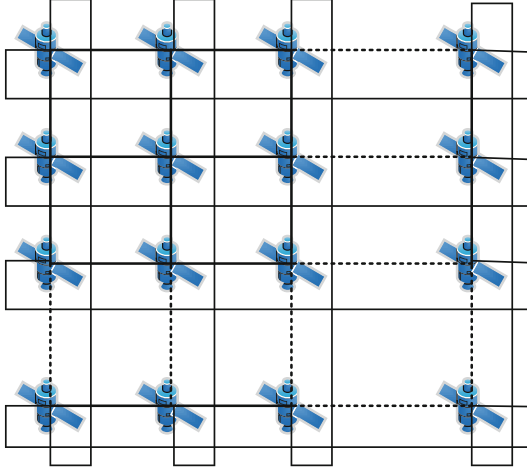


Fig. 1. Model diagram of LEO satellite network

Among them, f is the signal frequency, D_e represents the distance between satellite nodes, and c represents light speed. Signal to noise ratio is an effective parameter for measuring the channel state. In this communication model, this parameter is usually calculated using the following formula.

$$SNR = \frac{P_t G_t G_r}{k_B T_e B} \quad (2)$$

where k_B is a constant value representing the Boltzmann constant. T_e means the equivalent noise temperature. And B represents the channel bandwidth. P_t represents the transmitting power. G_t and G_r represent the gain of the transmitting antenna and the receiving antenna. The channel capacity of links between satellite nodes is usually calculated according to Shannon formula.

$$C = B \log_2(1 + SNR) \quad (3)$$

In this communication model, the latency is considered from three parts, mainly including data transmission delay, signal propagation delay, and queuing delay. In this communication model, time delay is considered from three parts, mainly including data transmission delay, signal propagation delay, and queue waiting delay. The calculation method is as follows.

$$t_{total} = \sum_{e \in E} t_{prop} + \sum_{n \in N} t_{tran} + t_{queue} \quad (4)$$

Among them, t_{prop} represents the propagation delay which is only related to the distance between nodes. t_{tran} represents data transmission latency, which is related to the amount of data and transmission rate. t_{queue} Indicates queue waiting delay, which is related to queue size and transmission rate. The busier the network, the longer the waiting time in the queue. The packet loss rate of the data stream during the entire transmission process is obtained using the following formula.

$$loss = 1 - pac/n_{total} \quad (5)$$

Among them, pac indicates successfully received data packets. And n_{total} represents the total forwarding volume of data packets during transmission.

2.3 Constraints of Satellite Networks

Bandwidth constraint. To ensure the normal transmission of data in the link, it is required that the required bandwidth for the task does not exceed the minimum value of the available bandwidth for all links on the path:

$$B_P \leq \min(B_{ij}, B_{jk}, \dots, B_{pq}) \quad (6)$$

Among them, B_P represents the required bandwidth for the task, and the bandwidth of each sub link is represented by B_{ij} .

Link Connectivity Constraints. The satellite network in a time slot can be considered as a static topology, and the connection status of links can be determined based on the duration of the links. To ensure that the links remain connected during data transmission, it is required that the minimum duration of all links on the path is not less than the set time slot length:

$$\min(c_{ij}, c_{jk}, \dots, c_{pq}) \geq T \quad (7)$$

Among them, T is the time slot, and the bandwidth of each sub link is represented by c_{ij} .

Transmission Rate Constraint. The task transfer rate must not exceed the set maximum transfer rate:

$$R_p \leq R_m \quad (8)$$

Among them, R_p represents the task transmission rate, and R_m represents the set maximum transmission rate.

2.4 DQN Satellite Routing Model

Deep Q network is currently a popular deep reinforcement learning algorithm. And it is usually described as a Markov model. It can be represented as a quadruple(s, a, s', r), where s means the state space, a means the action space, r indicates the reward obtained after performing action a , and s' represents the new state reached after performing action

a . The Q function maps actions and states to Q values, and chooses the appropriate action corresponding to the maximum probability in the Q function. Due to their expertise in modeling high-dimensional complex functions, neural networks can be used as function approximators to estimate this Q function. Combining DQN with routing algorithms can provide new ideas for solving high-dimensional multi objective optimization problems. In the routing model of the LEO satellite network in this paper, the controller collects link states and node information of the network at regular intervals as the state space for training routing strategies. The controller uses this routing strategy to output the optimal next hop node, and ultimately completes the routing process throughout the entire transmission process.

The action value function (Q function) represents the expected reward calculated by taking action a based on state s under the guidance of strategy π . Strategy π represents the mapping from state to behavior, equivalent to the decision-making strategy of an intelligent agent, and selects different behaviors based on different states, i.e. $a = \pi(s)$. Under the guidance of strategy π , the Q function of the robot is expressed as follows:

$$Q(s, a) = E_{\pi}[G_t | S_t = s, A_t = a] \quad (9)$$

where G_t represents discount reward, defined as follows:

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=t}^T \gamma^{k-t} R(s_k, a_k) \quad (10)$$

Among them, γ is the discount factor, which refers to the value ratio of future expected rewards at the current moment. γ^t decreases iteratively during the training process, and a smaller γ^t indicates a smaller impact of future rewards on the current reward. The calculation formula of the Q function is represented as follows:

$$Q_{\pi}(s_t, a_t) = E[R(s_t, a_t) + \gamma E_{a_{t+1} \sim \pi} Q_{\pi}(s_{t+1}, a_{t+1})] \quad (11)$$

Among them, $R(s_t, a_t)$ indicates the immediate reward corresponding to the action taken. The formula for solving the optimal behavior strategy by selecting the action corresponding to maximum Q value is as follows:

$$a = \arg \max_{a \in A} Q(s, a) \quad (12)$$

The DQN algorithm constructs labels for algorithm training through reward values of behavior, and the experience replay and target network effectively reduce the connections between datasets and improve the sparsity of data. The schematic diagram of the DQN algorithm is as follows (Fig. 2).

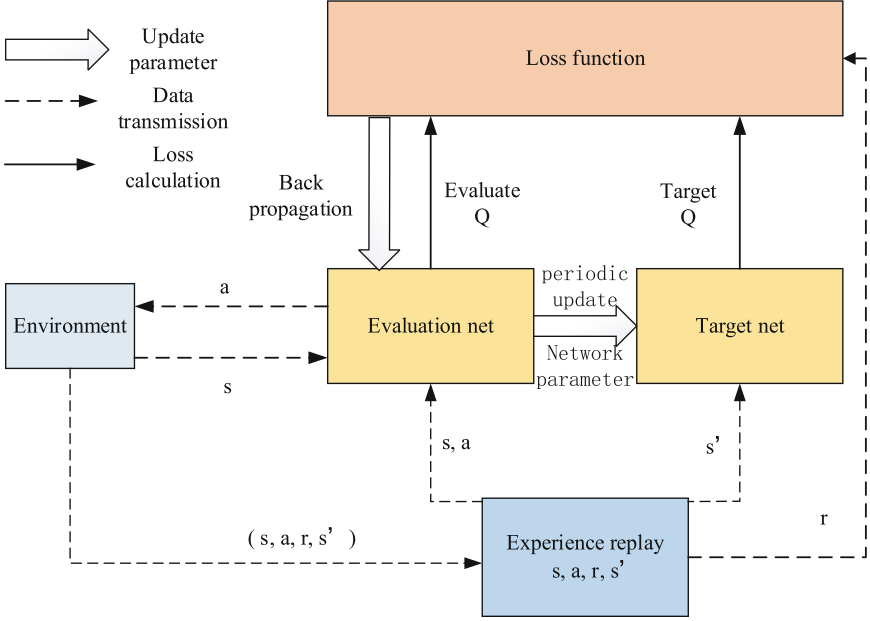


Fig. 2. Schematic diagram of DQN structure

3 Load Balanced Routing Algorithm Based on DQN

Figure 3 is the flowchart of the LEO satellite load balancing routing strategy based on DQN proposed in this article. To transform the satellite routing problem into a Markov decision process, it's important to define the state space, action, and reward in DQN reasonably, and optimize and compress the spatial dimension to improve network training efficiency. (1) State space: In this network model, the state space consists of link states and node information, mainly including the satellite node where the data packet is currently located, the load situation of adjacent satellite links, and the connectivity and distance between each satellite node. It is divided into three state matrices: position relationship matrix, load matrix, and distance matrix, which are merged into one state. (2) Action space: Actions represent next hop node selection. Considering the satellite network in this paper, each node is connected to four adjacent nodes, representing four output directions and four actions. Meanwhile, real-time connectivity of the four links is considered for judgment, and different output values are given. (3) Reward function: The reward function contains the objective we want to optimize. In this algorithm, the reward is defined as a function of distance, congestion level, and remaining available bandwidth, as shown in the following equation:

$$reward = -\omega_1 * d_n - \omega_2 * C_n + \omega_3 * B_n \quad (13)$$

Among them, $reward$ represents the reward. And d_n represents the distance to the destination node. C_n represents congestion levels of node n , $C_n = q_n / Q_m$. q_n represents the current queue length, and Q_m indicates the maximum queue length. B_n represents

the remaining available bandwidth of the link, $\omega_1, \omega_2, \omega_3$ represents the weight coefficient, $\omega_1 + \omega_2 + \omega_3 = 1$. The weight coefficient can be adjusted according to the network situation.

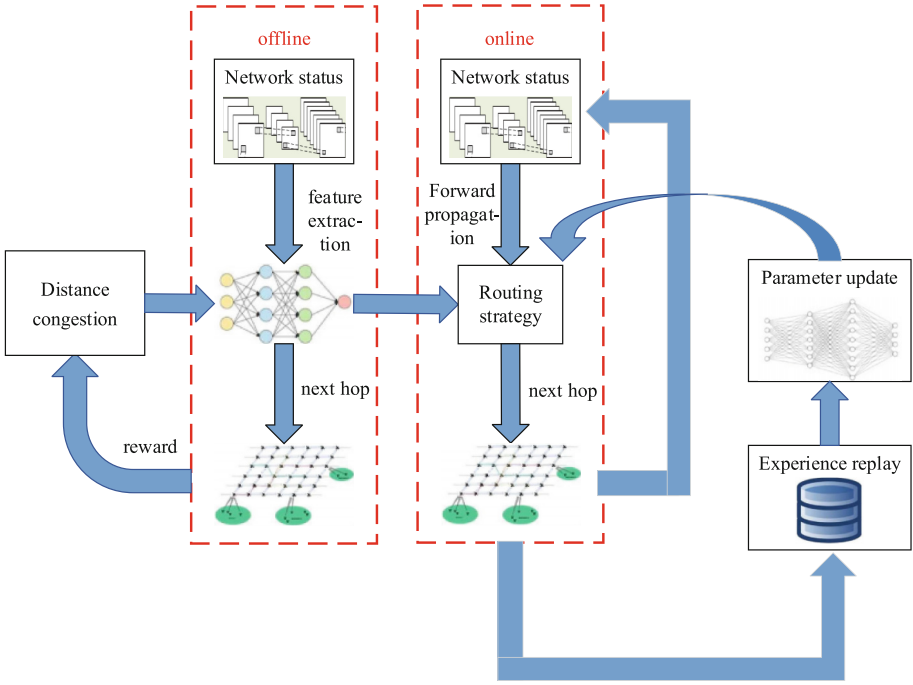


Fig. 3. DQN load balanced routing algorithm model

The algorithm mainly include two processes: offline learning and online decision making. The offline stage of the DQN routing model mainly utilizes historical traffic data and simulated data for learning. Firstly, preprocess the data, then select each hop action based on the reward function, store each hop action and its pre and post state, which includes multidimensional matrices such as connectivity, node congestion, link load, etc. Finally, use multi-channel convolutional neural network(CNN) for learning. In the online phase, the connection and distance between nodes in the satellite network are first calculated within the time slot, and network topology information is transmitted to each satellite node. The number and size of data packets are set, and the state of each node is initialized and initial loads following a Poisson distribution are allocated. Then, each node sends requests to each other and establishes connections, then shares network status and node information. Based on the link load and distance, the objective function value is calculated, and the optimal value is chosen as the next hop. Continuously repeat the above operation until the data transmission is completed, and update the network topology information, state matrix, store current state actions and other information when entering the next time slot to continue learning. Experience replay and network parameter

updates will be conducted, and routing decisions will continue to be optimized to adapt to the new network state. The specific algorithm process can be found in Algorithm 1.

Algorithm 1: DQN based routing algorithm

Offline stage: initialize the environment, set source node S and destination node D ; Train neural network, find and set appropriate hyperparameters, and send routing strategies to satellite nodes.

Online stage: Set the topology update duration T , timer t , and number of packets n ;
 Calculate network topology information;
 for packet $i = 1$ to n do:
 While the current node is D :
 Send requests to neighboring nodes to obtain link load information and distance information;
 Calculate the objective function value *reward* based on the trained routing strategy, and select the node that maximizes it as the next hop;
 Update node and link status, set the next hop as the current node;
 if $t > T$:
 Update link connection status and distance relationship;
 Reset timer;
 end if
 Store the action a , state s , s' , reward r and perform experience replay and network parameter updates;
 end while
end for

4 Simulation and Analysis

Generate a single-layer LEO constellation of 64 satellites using STK, with a total of 8 orbits, each with 8 satellites. The orbit altitude is 895.5km and the inclination angle is 86.4°. Use STK's corresponding functions to calculate the connection status and distance between satellite nodes, and generate a link matrix and a distance matrix. Due to the need to consider the multidimensional information of the network, multiple channels can be used to express the network state as a multidimensional matrix trained in neural networks. The neural network in the DQN model uses CNN to better utilize its matrix computing power for processing multidimensional resources and extracting features. The CNN model contains two convolutional layers with 16 and 32 convolutional kernels respectively. Related simulation parameters are listed in Table 1. The link channel capacity is set to 100 Mb/s, and the maximum data transmission rate for each task is 2 Mb/s. The initial load of each node follows a Poisson distribution, and the packet size is set to 512bit. The source and destination nodes are set for the task.

We use two traditional routing algorithms as comparative algorithms to highlight the performance advantages of the algorithm proposed in this paper. Dijkstra's shortest path is currently one of the most common routing algorithms, which provides the optimal routing decision for distance. Maximum flow algorithm considers maximizing traffic flows, which has good performance in throughput. Figure 4 shows the performance

Table 1. Simulation parameters

Parameters	Values
Height of LEO track	895.5 km
Number of LEO satellites	8×8
Orbit inclination angle	86.4°
Channel capacity	100 Mb/s
Maximum data rate	2 Mb/s
Size of packets	512 bit
Maximum queue length	100
Patch size	200
Learning rate	0.01
Soft update network weight	0.3
Action exploration attenuation	0.998
Minimum exploration rate	0.1
Reward attenuation	0.9

differences of the three algorithms in total end to end latency. From the simulation graph, it can be concluded that DQN based algorithm has a lower end to end latency. When the network is idle, the delay performance of three algorithms is not significantly different because the congestion situation in the network is not severe at this time, mainly focusing on propagation and transmission delay. As the number of packets increases, the congestion situation of the shortest path algorithm becomes more apparent, the queuing delay in nodes gradually increases, leading to a significant increase in total latency. Although the maximum flow algorithm can choose alternative paths to appropriately alleviate congestion, some routes are selected at the cost of sacrificing distance and latency, which increases propagation delay and overall delay slightly better than the shortest path algorithm. DQN based load balancing routing algorithm performs better in terms of end to end latency.

The simulation comparison of network throughput performance is shown in Fig. 5. From the simulation graph, it's not difficult to see the proposed algorithm performs the highest network throughput. When the network is idle, the difference in throughput performance between algorithms is not significant because the network load is low and there is no congestion when transmitting packets. When the traffic becomes busy, the congestion situation of the shortest path algorithm becomes more obvious due to link capacity limitations, reaching and maintaining the maximum value first, and the network throughput is the lowest. The maximum flow algorithm has a higher network throughput due to its ability to find an optimal traffic allocation scheme in the network, which maximizes the total traffic in the network. However, due to its high latency, its peak network throughput is slightly lower than the proposed algorithm. This algorithm continuously searches for nodes with lower congestion levels based on changes in network load, expands alternative paths, averages network load to more nodes, reduces single node

reuse times, and alleviates network pressure. It follows that the load balancing routing algorithm based on DQN also has good performance in terms of network throughput.

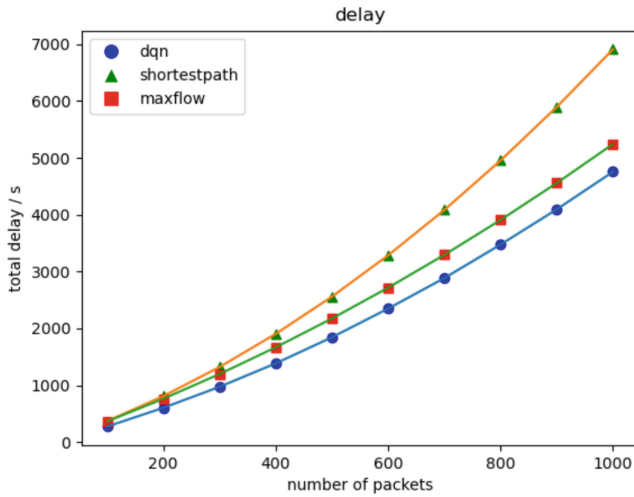


Fig. 4. Comparison of end to end latency performance

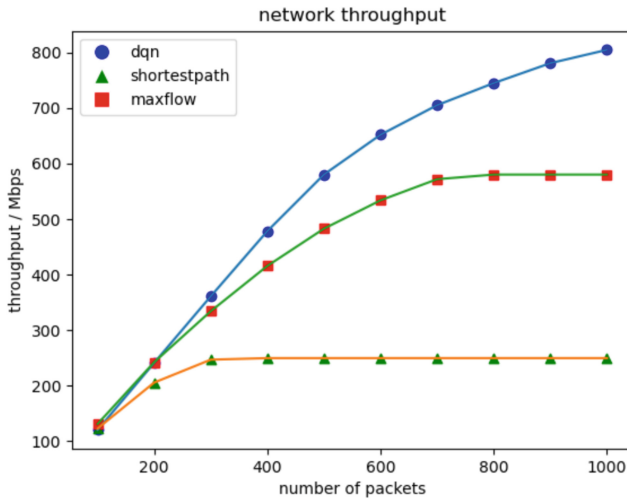


Fig. 5. Comparison of network throughput performance

Figure 6 displays the difference in the number of participating satellites among the three algorithms, which mainly reflects the performance of the algorithms in load balancing. The more satellites participate in forwarding, the fewer times each satellite node is reused, and more satellite nodes and routing paths are discovered to share network pressure and achieve load balancing. It is not difficult to see in the simulation diagram

that DQN algorithm involves a larger number of satellites in forwarding. And as the number of data packets increases, new nodes will continue to expand, reducing the reuse frequency of old nodes and alleviating network congestion. The shortest path algorithm only considers the shortest path. So the satellites participating in forwarding only include the nodes on that path, resulting in link congestion and plenty of data packet backlog, which results in high transmission and queuing delays. The maximum flow algorithm is limited by network traffic. When the current flow in the link reaches its maximum capacity, the path containing that edge cannot be used as an augmenting path to expand new satellite nodes. As the amount of data continues to increase, the traffic near the destination node in the network approaches saturation, and the maximum flow algorithm no longer has new nodes participating in forwarding. From this, it is obvious to find that DQN based load balancing routing algorithm proposed in this paper involves a larger number of satellites in forwarding, and has a stronger ability for load balancing.

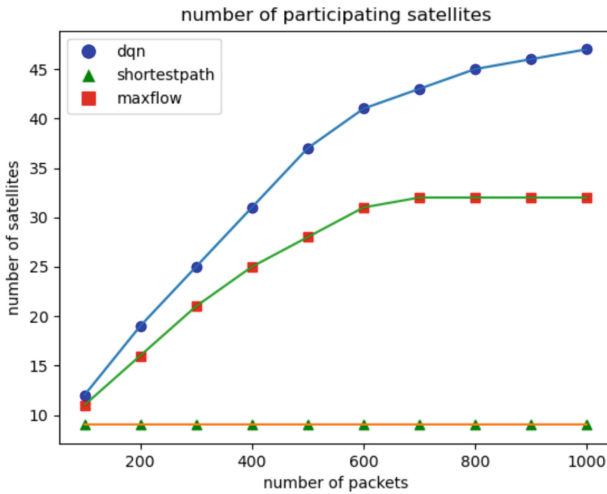


Fig. 6. Comparison of satellites participating in forwarding

5 Conclusion

This paper considers the high dynamism, large latency, and uneven network load characteristics of low orbit satellite networks. It analyzes the limitations of traditional satellite routing algorithms, which lack real-time adaptability and intelligence to network mobility, leading to a decrease performance and significant overhead. The idea of deep reinforcement learning is introduced in the routing scheduling strategy in the paper. By combining the features of LEO satellite networks and the advantages of DQN, an intelligent routing algorithm based on DQN which can effectively balance network load is proposed for low orbit satellite networks. This algorithm adopts the method of time slicing to learn the node connection relationship and load situation in the static topology

of each time slot. It jointly optimizes multiple objectives such as the distance between target satellites, node congestion, and available link bandwidth, and provides an optimized next hop selection strategy for each node. Update the link connectivity matrix for different time slots and continue training to update network parameters. By comparing with traditional satellite network routing algorithms, it is obvious to find that the proposed algorithm has greater advantages in latency and network throughput, reduces the reuse of satellite nodes, alleviates the problem of uneven load in the network, and improves network performance.

References

1. Liu, Y., Liu, C.: Distributed dynamic routing algorithm for satellite constellation. In: Conference 2018 10th International Conference on Communication Software and Networks (ICCSN), pp. 300–304. IEEE (2018)
2. Liu, Z., Li, J., Chen, S., et al.: HGL: a hybrid global-local load balancing routing scheme for the Internet of Things through satellite networks. *J. Int. J. Distrib. Sens. Netw.* **13**(3), 161–169 (2017)
3. Nunes, B., Mendonca, M., Nguyen, X., et al.: A survey of software-defined networking: past, present, and future of programmable network. *J. IEEE Commun. Surv. Tutorials* **16**(3), 1617–1634 (2014)
4. Bertaux, L., Medjah, S., Berthou, P., et al.: Software defined networking and virtualization for broadband satellite networks. *J. IEEE Commun. Mag.* **53**(3), 54–60 (2015)
5. Papa, A., De, C., Vizarreta, P., et al.: Design and evaluation of reconfigurable SDN LEO constellations. *J. IEEE Trans. Network Serv. Manage.* **17**(3), 1432–1445 (2020)
6. Hu, M., Li, J., Cai, C., et al.: Software defined multicast for large-scale multi-layer LEO satellite networks. *J. IEEE Trans. Network Serv. Manage.* **19**(3), 2119–2130 (2022)
7. Mao, B., Fadlullah, Z., Tang, F., et al.: A tensor based deep learning technique for intelligent packet routing. In: Conference Globecom 2017–2017 IEEE Global. Communications Conference, pp. 1–6. IEEE (2017)
8. Mao, B., Fadlullah, Z., Tang, F., et al.: Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning. *J. IEEE Trans. Comput.* **66**(11), 1946–1960 (2017)
9. Zuo, P., Wang, C., Yao, Z., Hou, S., Jiang, H.: An intelligent routing algorithm for LEO satellites based on deep reinforcement learning. In: Conference in IEEE Vehicular Technology Conference (VTC-Fall), pp. 1–5 (2021)
10. Liu, J., Zhao, B., Xin, Q., Su, J., Ou, W.: DRL-ER: an intelligent energy-aware routing protocol with guaranteed delay bounds in satellite mega-constellations. *J. IEEE Trans. Netw. Sci. Eng.* **8**(4), 2872–2884 (2021)
11. Wang, F., Jiang, D., Wang, Z., Lv, Z., Mumtaz, S.: Fuzzy-CNN Based Multi-Task Routing for Integrated Satellite-Terrestrial Networks. *J. IEEE Trans. Vehicular Technol.* **71**(2), 1913–1926 (2022)