Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3759631, 15 pages
https://doi.org/10.1155/2021/3759631

Research Article

Reinforcement Learning-Based Routing Algorithm in Satellite-Terrestrial Integrated Networks

Yabo Yin,[1] **Chuanghe Huang**,[1] **Dong-Fang Wu**,[1] **Shidong Huang**,[1] **M. Wasim Abbas Ashraf**,[1] **and Qianqian Guo**[2]

[1]*School of Computer Science, Wuhan University, Wuhan 430072, China*
[2]*School of Information Engineering, Zhengzhou Institute of Finance and Economics, Zhengzhou 450053, China*

Correspondence should be addressed to Chuanghe Huang; huangch@whu.edu.cn

Received 9 June 2021; Revised 22 September 2021; Accepted 8 October 2021; Published 28 October 2021

Academic Editor: Yanjie Fu

Satellite-terrestrial integrated network (STIN) is an indispensable component of the Next Generation Internet (NGI) due to its wide coverage, high flexibility, and seamless communication services. It uses the part of satellite network to provide communication services to the users who cannot communicate directly in terrestrial network. However, existing satellite routing algorithms ignore the users' request resources and the states of the satellite network. Therefore, these algorithms cannot effectively manage network resources in routing, leading to the congestion of satellite network in advance. To solve this problem, we model the routing problem in satellite network as a finite-state Markov decision process and formulate it as a combinatorial optimization problem. Then, we put forth a Q-learning-based routing algorithm (QLRA). By maximizing users' utility, our proposed QLRA algorithm is able to select the optimal paths according to the dynamic characteristics of satellite network. Considering that the convergence speed of QLRA is slow due to the routing loop or ping-pong effect in the process of routing, we propose a split-based speed-up convergence strategy and also design a speed-up Q-learning-based routing algorithm, termed SQLRA. In addition, we update the Q value of each node from back to front in the learning process, which further accelerate the convergence speed of SQLRA. Experimental results show that our improved routing algorithm SQLRA greatly enhances the performance of satellite network in terms of throughput, delay, and bit error rate compared with other routing algorithms.

1. Introduction

As the 5th generation communication technologies are widely used, terrestrial network can provide high bandwidth and low delay communication services to the users within the coverage of base stations [1]. However, for those remote areas where base stations are not deployed or where base stations are destroyed by natural disasters, terrestrial network usually cannot meet the communication needs of users. Infrastructure of satellite network is rarely damaged by natural disasters, and it has wide coverage [2–4], so it is usually regarded as an essential component of terrestrial network. Satellite-terrestrial integrated network (STIN) has wide coverage and high flexibility and is able to compatible with the existing 5G network. Thus, it is a reliable paradigm to provide the Internet services and is receiving much attention from researchers [5–8]. In particular, when users are unable to communicate through terrestrial network in aviation or navigation, the STINs can provide them with communication services.

Satellite routing algorithm is an important technique in STINs, and there are much researches on it. Considering that the number of satellites is small and the structure of traditional satellite network is simple, existing satellite routing algorithms are developed from the routing algorithms of terrestrial network, such as OSPF [9], RIP [10], and AODV [11]. However, most algorithms are depended on the shortest path or minimum cost. Therefore, satellite network is prone to

congestion in the process of routing. In addition, with the expansion of satellite network (e.g., Starlink), these routing algorithms cannot converge quickly in the limited communication time, which seriously degrades the communication performance of satellite network and wastes the communication resources of satellites at the same time. On the other hand, most of the work on satellite routing ignores the impact of user request resources on the performance of satellite network. Liu et al. [12] proposed a fragment-based load balancing route scheme to control the traffic of LEO satellite network. Qi et al. [13] improved the quality of service (QoS) of users by jointly optimizing the rate and routing in LEO satellite network. Considering the traffic distribution density in different areas, the authors in [14] proposed a distributed routing algorithm based on traffic prediction. However, the above algorithms do not take into account the impact of the current user's routing on the subsequent user's routing, resulting in the network performance is degraded. In addition, considering the power and computing resources of LEO satellite, it is not appropriate to deploy these algorithms on satellites. Therefore, it is very challenging to design an efficient satellite routing algorithm.

Recently, machine learning and deep learning [15, 16] have been used extensively. Some researchers began to use these methods to solve network communication problems [17–19]. The authors in [20] regarded satellite network topology as a series of snapshots and used particle swarm optimization algorithm for routing in each snapshot. However, deep learning is an approximate algorithm, which is not suitable for sequential decision problems. Reinforcement learning is a method based on trial and error. In the process of learning, the agent interacts with the environment and gets a corresponding reward. And this reward guides the agent to find the best strategy. Moreover, reinforcement learning is very suitable for dealing with sequential decision problems and achieves better results than human beings [21], and it has been applied in resource allocation [22], capacity management [23], and combinatorial optimization [24, 25]. Compared with other reinforcement learning algorithms, Q-learning is a simple and efficient reinforcement learning method, which has a fast convergence speed. In addition, it is very suitable for solving discrete problems. Inspired by the above references, we regard the satellite routing problem as a turn-base game and model it as a finite-state Markov decision process. Then, we put forward a Q-learning-based routing algorithm to solve satellite routing problems.

In this paper, we mainly investigate the satellite routing problem in STINs. Choosing a path from source node to destination node can be regarded as a turn-based game. And this is a finite-state Markov decision process. So we model the routing problem as a Markov decision process and define its state space, action space, and reward function. We propose QLRA algorithm to make full use of satellite network resources and improve the quality of service of users. In addition, in order to accelerate the convergence speed of QLRA algorithm, we propose a split-based speed-up convergence strategy and design a speed-up Q-learning-based routing algorithm (SQLRA). Moreover, we update the Q value from back to front to further improve the speed of SQLRA algorithm. The contributions of this paper are summarized as follows:

(1) We model the satellite routing as a Markov decision process and define its action and state spaces and reward function. And we propose a Q-learning-based satellite routing algorithm (QLRA). QLRA algorithm can select the optimal paths according to the current states of satellite network and the users' request resources when routing

(2) Aiming at the slow convergence speed of QLRA algorithm, we analyse the problem and propose a split based speed-up convergence strategy to accelerate the convergence speed of QLRA. Based on QLRA algorithm, we design a speed-up Q-learning-based routing algorithm (SQLRA). In addition, we update the Q-value from back to front to further improve the convergence speed of SQLRA algorithm. Experimental results show that SQLRA algorithm converges faster than QLRA algorithm

(3) Our proposed algorithm SQLRA can make full use of network resources while meeting the requirements of users. Numerical simulation results show that SQLRA algorithm effectively enhances the network performance compared with other algorithms

The remainder of this paper is organized as follows. In Section 2, the related work is reviewed. In Section 3, we introduce network model and problem formulation. In Section 4, the satellite routing algorithm based on Q-learning and the speed-up Q-learning-based routing algorithm are presented. In Section 5, we evaluate the performance of SQLRA algorithm in two different scenarios, analyse, and discuss the experimental results. Section 6 concludes this paper and gives future research issues.

2. Related Work

There are much researches on satellite network routing issues. In order to reduce the link congestion and the imbalance of load distribution, Liu et al. [26] proposed an iterative Dijkstra algorithm to optimize satellite communication path. The traditional LEO satellite network ignores the delay of links in routing, which leads to the incomplete evaluation of satellite network performance. To solve this problem, in [27], a satellite routing algorithm taking delay into account was proposed. The authors in [28] proposed a routing algorithm based on cooperative game theory to solve the problem of propagation delay and traffic load imbalance in LEO satellite network. Jiang et al. [29] designed a routing algorithm based on fuzzy theory to meet the multilevel needs of users. By leveraging orbit prediction information, Pan et al. [30] put forward a dynamic on-demand routing scheme to reduce the routing convergence and the communication overhead. Hao et al. [31] proposed a routing strategy based on energy-aware and load-balancing to meet the different communication services of users.

As a reliable communication paradigm, much work has been done on STINs. In order to improve the power utilization of satellites, the authors in [32] proposed a data offloading scheme in STINs to jointly allocate the power and resources of satellites. Zhang et al. [33] used edge computing techniques to improve the QoS of STINs. In order to reduce the cost of gateway deployment and data routing in STINs, the authors in [34] proposed a joint satellite gateway deployment and routing scheme. Xu et al. [35] proposed a hybrid routing algorithm to realize the seamless integration of STINs. The authors in [36] presented an end-to-end routing method based on heuristic strategy to improve the QoS of STINs.

Reinforcement learning is an effective method to cope with sequential decision problems, and it has been applied in many fields. Liu et al. [37] used Q-learning to implement the content caching problem in dynamic cloud content distribution network. To improve the efficiency of the Internet of Things, Pan et al. [38] used Q-learning to identify blocked links. A Q-learning method is proposed in [39] to improve the network performance and reduce the energy consumption of wireless sensor networks. Qiao et al. [40] proposed a joint optimization scheme of cache content placement and bandwidth resource allocation based on deep reinforcement learning in the Internet of Vehicles. Q-learning technique is widely used, and there are few researches on routing using Q-learning technique in satellite network. In this research, we use reinforcement learning to solve the routing problem in satellite network.

3. System Model and Problem Formulation

3.1. Network Model. The STINs used in this paper are shown in Figure 1. The STINs are composed of a terrestrial network and a LEO satellite network. The terrestrial network consists of base stations, routers, satellite gateways, and user terminals, and the satellite network consists of a large number of LEO satellites. The terrestrial network is able to connect with the satellite network with the aid of satellite gateways. Considering the high-speed mobility of the satellites in satellite constellation, each satellite is only connected to its neighbour satellites or satellites in its adjacent orbits. The communication link between satellites is bidirectional, and the specific structure is shown in Figure 2.

When users communicate with their peers, the system first determines whether to reach their peers through the terrestrial network. Specifically, if their peers can be reached through the terrestrial network, then the data will be transmitted directly to their peers through the terrestrial network. Otherwise, the terrestrial network will transmit the data to the LEO satellites through the satellite gateways and then retransmit the data to their peers through the LEO satellite network. With the increasing number of satellites, existing satellite routing algorithms become unsuitable, which seriously degrade the performance of satellite network. Therefore, we focus on the satellite routing problem in this paper.

Considering that the topology of satellite network changes with time, inspired by reference [20], here we divide the whole operation time T of satellite network into N_T time slices, and the duration time of each time slice is T_t. We assume that the satellite network topology is fixed in each time slice. So the total time can be obtained by

$$\sum_{t=1}^{N_T} T_t = T. \tag{1}$$

The number of snapshots is related to the number of orbits and the number of satellites in each orbit. The time interval of the snapshot T_t is related to the inclination of the orbits. The smaller the time interval, the higher the accuracy of the snapshot. If the time interval is small, a large number of topologies will be generated, which leads to the complexity of the network structure. In practice, the time interval is no more than the minimum visible time of the satellite links. We define T_t as

$$T_t \le \min\{\tau(u, v)\}, u \ne v, u, v \in V, \tag{2}$$

where $\tau(u, v)$ represents the visible time between satellite u and satellite v. Here, we set the time interval T_t to 4 minutes.

Here, we use undirected graph $G = (V, E)$ to represent satellite network topology, where V represents the set of satellites, $V = \{1, 2, \cdots, N\}$, and N is the number of satellites. And E is the set of links between satellites. Here, we assume that the network structure is a connected graph. We define it as

$$E = \{\text{link}(u, v)\}, u \ne v, u, v \in V, \tag{3}$$

where $\text{link}(u, v)$ represents the link between satellite u and satellite v. And satellite v is a neighbour of satellite u. Considering that the state of satellite link consists of many parameters, we redefine link $\text{link}(u, v)$ as

$$\text{link}(u, v) = (\text{bandwidth}, \text{delay}, \text{error}, \text{time}), \tag{4}$$

where variables bandwidth, delay, error, and time represent the available bandwidth, the propagation delay, the bit error rate, and the available time of link $\text{link}(u, v)$, respectively. In addition, considering the duration of each time interval is short, we assume that the communication time of each user is greater than the duration of each time slice. We define it as

$$u^{\text{req\_}t} \ge T_t. \tag{5}$$

When transmitting data, it is necessary to find an optimal path from source satellite to destination satellite according to the current link states of satellite network. We assume that satellite 0 is source satellite and satellite 5 is destination satellite in Figure 3. There are multiple paths from satellite 0 to satellite 5. However, with a large number of users accessing to satellite network, satellite network resources are exhausted due to the load imbalance, which leads to the congestion of satellite links in advance. Therefore, the performance of satellite network is seriously degraded. For example, in the beginning, the optimal path

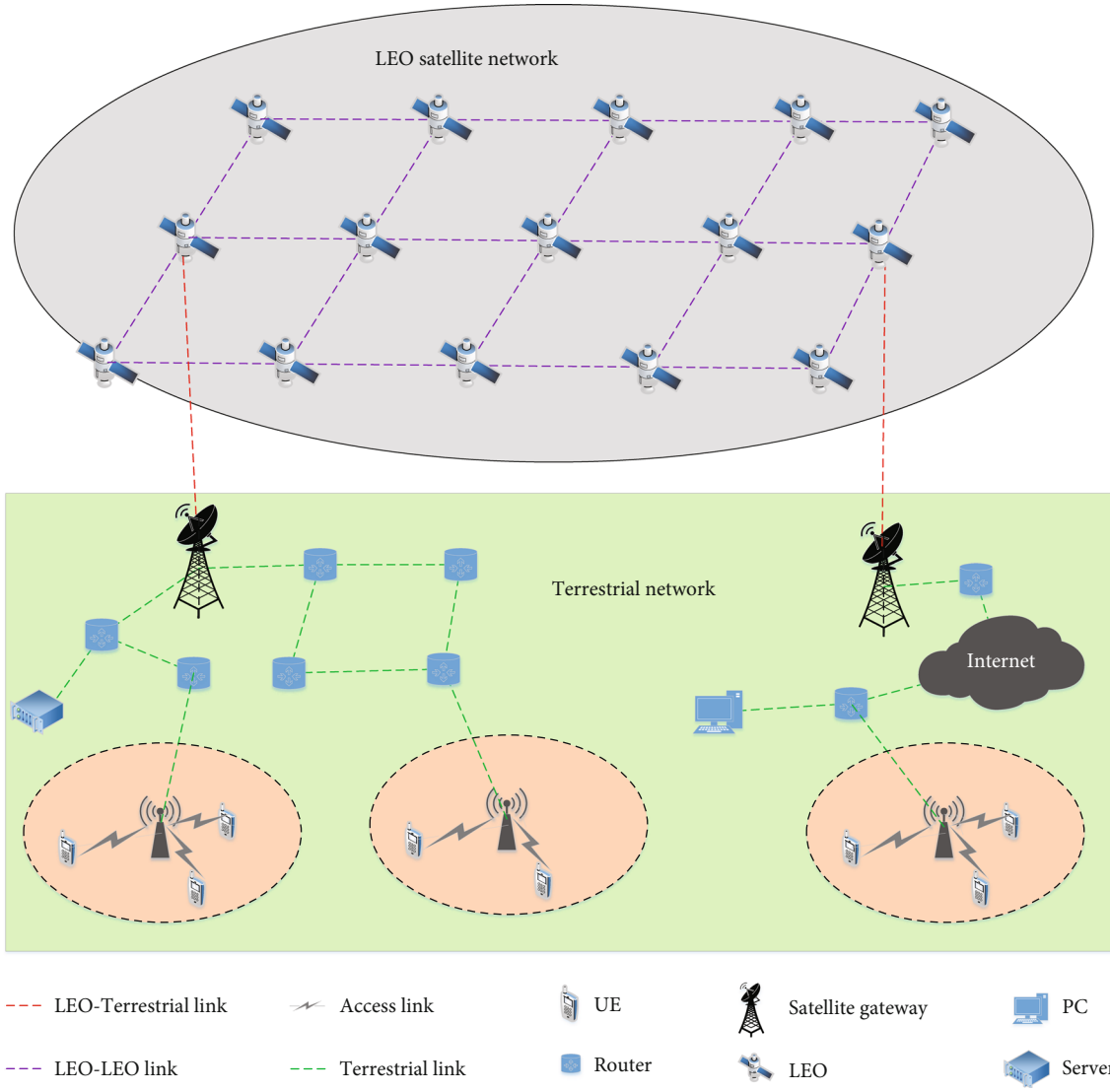

Figure 1: Structure of satellite-terrestrial integrated networks.

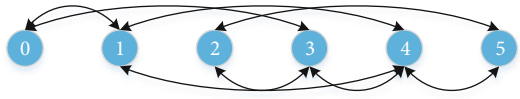

Figure 2: Communication links of LEO satellite network.

from satellite 0 to satellite 5 is 0-3-4-5 in Figure 3. With the increase of users' number, link 0-3 is congested because of the consumption of bandwidth resources, resulting in the next user cannot choose link 0-3 in the optimal path. Therefore, the path from satellite 0 to satellite 5 changes from path 0-3-4-5 to path 0-1-4-5 and finally to path 0-1-4-3-5. The specific process of path changing is shown in Figure 3.

In Figure 3, the dotted lines with different colours represent different selected paths. From Figure 3, we see that the optimal path from satellite 0 to satellite 5 changes gradually with the consumption of communication resources of satellite links.

3.2. Problem Formulation. We assume that the bandwidth capacity of link $\text{link}(u, v)$ is $c(u, v)$, variable u_i^{req} represents the bandwidth resource requested by the ith user. Before transmitting data, we need to find a path from source satellite v_s to destination satellite v_d for the ith user. The path is defined as

$$\text{path}(v_s, v_d) = \{\text{link}(v_s, u), \cdots, \text{link}(v, v_d)\}, u, v \in V, u \neq v. \tag{6}$$

Here, y is an indicator function which indicates whether there is a link in the selected path. If satellite link $\text{link}(u, v)$ is in the selected path, then $y(\text{link}(u, v)) = 1$; otherwise, $y(\text{link}(u, v)) = 0$.

Here, we use functions $B(x)$, $D(x)$, $E(x)$, and $T(x)$ to represent the average bandwidth, the delay, the bit error rate, and the available time of the user in path x, respectively.

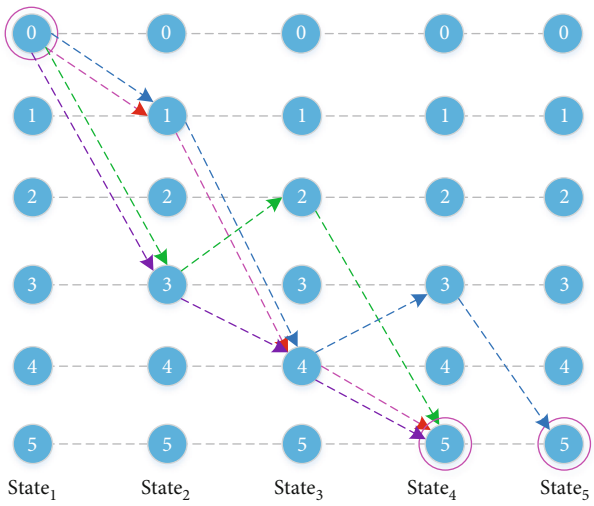

Figure 3: Process of path changing in satellite network.

Variable path^i represents the path of the ith user from source satellite to destination satellite.

$$B\left(\text{path}^i\right) = \frac{\sum \text{band}(\text{link}(u,v))}{\text{length}\left(\text{path}^i\right)}, \text{link}(u,v) \in \text{path}^i, \quad (7)$$

$$D\left(\text{path}^i\right) = \sum \text{delay}(\text{link}(u,v)), \text{link}(u,v) \in \text{path}^i, \quad (8)$$

$$E\left(\text{path}^i\right) = \sum \text{error}(\text{link}(u,v)), \text{link}(u,v) \in \text{path}^i, \quad (9)$$

$$T\left(\text{path}^i\right) = \min\left(\text{time}(\text{link}(u,v))\right), \text{link}(u,v) \in \text{path}^i, \quad (10)$$

where function $\text{length}(\text{path}^i)$ is the length of the path path^i and function $\text{band}(\text{link}(u,v))$ is the bandwidth of link $\text{link}(u,v)$. Similarly, functions $\text{delay}(\text{link}(u,v))$, $\text{error}(\text{link}(u,v))$, and $\text{time}(\text{link}(u,v))$ are the delay, the bit error rate, and the available time of link $\text{link}(u,v)$, respectively.

Our goal is to maximize the utility of all users by considering the bandwidth, the delay, the bit error rate, and the available time of network links in the process of routing.

$$\frac{1}{M} \max\left(\sum_{i=1}^{M} \theta \cdot B\left(\text{path}^i\right) + \beta \cdot D\left(\text{path}^i\right) + \lambda \cdot E\left(\text{path}^i\right) + \omega . T\left(\text{path}^i\right)\right), \quad (11)$$

$$\text{subject to } 0 \le \sum_{i=1}^{M} u_i^{\text{req}} \cdot y(\text{link}(u,v)) \le c(u,v), v \in V, u \ne v, \quad (12)$$

$$y(\text{link}(u,v)) \in \{0,1\}, \quad (13)$$

$$\sum_{u \in V, u \ne v} y(\text{link}(u,v)) \cdot u_i^{\text{req}} = \sum_{u \in V, u \ne v} y(\text{link}(v,u)) \cdot u_i^{\text{req}}, \forall \text{v} \in \text{V}, \text{v} \notin \{\text{v}_s, \text{v}_d\}, \quad (14)$$

$$\theta + \beta + \lambda + \omega = 1, \quad (15)$$

where M is the number of users accessing to satellite network. Equation (12) ensures that the bandwidth resource requested by users is less than the total bandwidth resources of each link. Equation (13) is an indicator function which indicates whether link $\text{link}(u,v)$ is in the selected path. If link $\text{link}(u,v)$ is in the selected path, $y(\text{link}(u,v) = 1$; otherwise, $y(\text{link}(u,v)) = 0$. Equation (14) is used to ensure that for any intermediate link, the incoming traffic and the outgoing traffic are equal. Equation (11) is a combinatorial optimization problem, and we use reinforcement learning to solve it. In the experiment, we use analytic hierarchy process (AHP) to judge the influence of the weight of each parameter on the performance of the satellite network [41].

4. A Satellite Routing Algorithm Based on Reinforcement Learning

Reinforcement learning is mainly composed of the agent and the environment. The agent interacts with the environment and learns the optimal strategy according to the feedback of environment. In particular, the reinforcement learning framework is shown in Figure 4. In the current state s_t, the agent chooses an action a_t according to the policy π and execute it. And the environment returns a corresponding reward to the agent, and the environment moves its state from s_t to the next state s_{t+1}. The agent interacts with the environment continuously until the episode is end or the number of interaction steps reaches the threshold set in advance.

In STINs, the environment is the link states of satellite network, and it is time-varying. And the agent is deployed in ground control center. In route discovery phase, the agent chooses a valid action according to the users' request and jumps to the satellite whose index is the valid action value. The environment gives the agent a corresponding reward, and the link states of satellites are changed simultaneously. The agent interacts with the environment until a path from source satellite to destination satellite is selected. The routing process is modelled as Markov decision processes (MDPs) and represented by $M = (S, A, P, R)$, where S is state space, A is action space, and R is reward value. Furthermore, P is the state transition probability function, $P(s' \mid s, a) = P(s' = s' \mid s = s, a = a)$. The specific details are defined as follows:

(1) *State Space*. The satellite link state considered in this paper includes available bandwidth, propagation delay, bit error rate, and available time. We define the state of link as

$$\text{link}_{i,j} = \left(b_{i,j}, d_{i,j}, e_{i,j}, t_{i,j}\right), 0 \le i, j \le N, \quad (16)$$

where N is the number of satellites. And $\text{link}_{i,j}$ denotes the

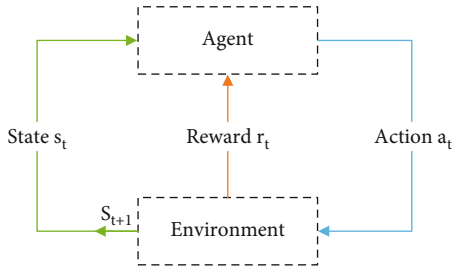

Figure 4: Reinforcement learning framework.

link state between satellite i and satellite j. In addition, variables $b_{i,j}$, $d_{i,j}$, $e_{i,j}$, and $t_{i,j}$ represent the available bandwidth, the propagation delay, the bit error rate, and the available time of link between satellite i and satellite j, respectively. The variable S_t represents the states of all links in satellite network.

$$S_t = \left\{\text{link}_{i,j}\right\}, 0 \le i, j \le N, j \in N(i), \tag{17}$$

where $N(i)$ represents the set of neighbours of satellite i. And variable S_t is the environment of reinforcement learning.

(2) *Action Space*. In satellite network, the action is used to describe the process of the agent moving from one satellite to another. For example, taking action a, the agent moves from the current satellite to the satellite whose index is a. The number of satellites is N, so the action set is denoted by $A = \{1, 2, \cdots, N\}$. For the convenience of calculation, the action is coded by one-hot coding in our simulations

(3) *Reward Value*. The rewards are used to motivate the agent to search for the optimal strategy. The agent obtains the rewards by the states of satellite links. Different link states give different rewards. In order to avoid the impact of different rewards on the accuracy of results, we use min-max operation to normalize them. Function max (b) is the maximum of variable b, and function min (b) is the minimum of variable b. We elaborate the specific operation below:

$$r(b) = \frac{b - \min(b)}{\max(b) - \min(b)}, \tag{18}$$

$$r(d) = \frac{\max(d) - d}{\max(d) - \min(d)}, \tag{19}$$

$$r(e) = \frac{\max(e) - e}{\max(e) - \min(e)}, \tag{20}$$

$$r(t) = \frac{t - \min(t)}{\max(t) - \min(t)}, \tag{21}$$

where $r(b)$ is the reward generated by the bandwidth of the selected link. Similarly, $r(d)$, $r(e)$, and $r(t)$ represent the rewards generated by the delay, the bit error rate, and the available time of the selected link, respectively. The link delay and the bit error rate are negative to the link selection. Therefore, we use monotone decreasing function to present the corresponding rewards in the process of normalization. In this way, the total reward generated by the selected links is shown in Equation (22).

$$r = \theta \cdot r(b) + \beta \cdot r(d) + \lambda \cdot r(e) + \omega \cdot r(t), \tag{22}$$

where variables θ, β, λ, and ω are weight coefficients, respectively, which are used to represent the importance of each reward. Here, we use analytic hierarchy process (AHP) to determine the value of these parameters. In addition, variable R_t is the cumulative reward that the agent gets by taking action a_t in current state s_t. We define it as

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}. \tag{23}$$

We use state-action value $Q(s, a)$ to represent the cumulative reward value obtained by the agent taking action a_t in current state s_t. And this value indicates the quality of each action in the current state. We define it as

$$Q_\pi(s_t, a_t) = E_\pi[R_t \mid S_t = s_t, A_t = a_t]. \tag{24}$$

Choosing different strategy functions will get different state-action values. Our goal is to find the best strategy function to make the agent choose the appropriate action in each state.

$$Q_\pi{}^*(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t). \tag{25}$$

According to Equation (25), we maximize the state-action value $Q(s, a)$ to find the optimal strategy. The strategy with the symbol "$*$" is the optimal strategy. At last, the best action in each state can be selected by searching Q-table.

4.1. Path Checking Algorithm. When the communication resources of satellite links are exhausted, the links will be congested. If there is not a path from source node to destination node in satellite network, the routing algorithm cannot find a suitable path to destination node. In order to avoid this situation, we first judge whether there is a reachable path to destination node before looking for a path. If there is no such a path, it means that links are congested or disconnected in satellite network. At this time, the routing algorithm stops looking for paths, reducing the waste of computing resources. We use pseudo code to describe the details of path checking algorithm.

where $flag = 1$ indicates that there is a path from *start_node* to *end_node* and $flag = 0$ indicates that there is no path from *start_node* to *end_node*.

4.2. Q-Learning-Based Routing Algorithm (QLRA). Q-learning is a reinforcement learning method based on value function, and it uses a Q-table to store different state-action

```
Input: start_node, end_node, graph.
Output: the flag which indicates whether there is a valid path from start_node to end_node.
1.Initialize flag = 0, path ={}.
2.Get the neighbours of start_node based on the network structure graph, neighbours_list.
3.Let path = path∪{start_node}.
4.while neighbours_list:
5.   Pop a node from neighbours _list, node.
6.   if node not in path:
7.      if node == end_node:
8.         path = path ∪{node}.
9.         flag =1.
10.     end if
11.     else:
12.           Get the neighbours of node, node_neighbours.
13.           Add the node_neighbours to the list neighbours_list .
14.           path = path ∪{node}.
15.     end else
16.  end if
17.end while
```

Algorithm 1: Path checking algorithm (PCA).

values. Because Q-learning is a mode-free method, it does not need prior knowledge in the process of learning. In addition, Q-learning learns the optimal strategy by trial and error, and it is very suitable for the dynamic satellite network. Therefore, here we try to use Q-learning to select the optimal route. The main idea of Q-learning algorithm is we first initialize the Q-table, then select an action from the current state by ε-greedy strategy, and the environment moves from the current state to the next state and update the state-action value $Q(s, a)$ by using Bellman Equation. At last, this process is repeated until the Q-table converges. The Bellman Equation is defined as

$$Q(s,a) = Q(s,a) + \alpha\left[r + \lambda \max Q\left(s', a'\right) - Q(s,a)\right], \quad (26)$$

where α denotes the learning rate, λ is the discount rate, and s' represents the next state. Furthermore, variable r denotes the immediate reward obtained from the environment.

$$a = \begin{cases} \text{select an action randomly}, & r' < \varepsilon, \\ \arg\ \max_a Q(s,a), & r' \geq \varepsilon. \end{cases} \quad (27)$$

In the process of training, we use ε-greedy strategy, as shown in Equation (27), to avoid the result falling into the local optimal solution. The strategy can achieve the trade-off between exploration and exploitation, where r' represents the random number generated in the process of selecting the action and ε represents the probability of action exploration. Furthermore, in order to speed up the convergence of Q-table, we redefine the reward function, where C is a constant.

$$r(s_{t+1} \mid s_t, a_t) = \begin{cases} r(s_t, a_t) & s_{t+1} \text{ is not terminal state}, \\ r(s_t, a_t) + C & s_{t+1} \text{ is the terminal state}. \end{cases} \quad (28)$$

We first judge whether there is a feasible path. If there is a path, the optimal path is selected by means of Q-learning routing algorithm. Then, the reward matrix and structure of satellite network are updated. The specific Q-learning-based routing algorithm is described as follows:

where the function *PCA* in QLRA represents the path checking algorithm proposed above.

4.3. Speed-Up Q-Learning-Based Routing Algorithm (SQLRA). In the process of selecting the next hop, the agent will jump from the current state to the previous state. And this operation will result in some repeated and invalid sequences in the selected path. When selecting the path from source satellite node B to destination satellite node J, there will be some repeated and invalid sequences. The specific details are shown in Figure 5. For example, in path $B->E->F->C->B->E->H->G->H->G->H->G->J$, the sequences in the magenta dashed box indicate that a routing loop has occurred, and the sequences in the blue dashed box indicate that a ping-pong effect has occurred. From Figure 5, we can see that sequences $E->F->C->B$ and $G->H$ are repeated and invalid. These repeated and invalid sequences will not only waste computing resources but also lead to the slow convergence speed of QLRA algorithm.

Although QLRA algorithm uses ε-greedy strategy to select effective actions in the learning process, it still generates invalid sequences. To avoid the routing loop or ping-pong effect, we must prevent the agent from jumping from the current state to the previous visited state, when the agent selects an effective action. To solve this problem, we propose a split-based speed-up convergence strategy. The specific split process is shown in Figure 6.

Similar to the broadcast mechanism, we split the satellite network according to the neighbour information of nodes. As shown in Figure 6, for destination node J, we regard its neighbour nodes as the first layer, and nodes with the same colour belong to the same layer. We update the Q value of all

Input: *start_node*, *end_node*, *graph*, *R*, *users_req_list*.
Output: the optimal paths.
1. Initialize Q-table, α, γ, ε, Episodes=*M*.
2. for user_req in users_req_list:
3. *flag* = *PCA* (*start_node*, *end_node*, *graph*).
4. if *flag* ==1:
5. for *i*=1 to Episodes:
6. *current_state* = *start_node*.
7. while *current_state*!= *end_node*:
8. Select the action *a* based on Eq. (27).
9. Get the corresponding reward value generated by each parameter according to Eq. (18), (19), (20) and (21).
10. Obtain the total reward *r* based on Eq. (22).
11. The agent move to the next state *s'*.
12. Update the Q-table based on Eq. (26).
13. Let *current_state* = *s'*.
14. end while
15. end for
16. Select the optimal path *path* from the converged Q-table based on Eq. (25).
17. Update the reward matrix *R* based on the consumption of link resources.
18. Update the structure of satellite network *graph* based on the consumption of link resources.
19. end if
20. else:
21. There is no path from *start_node* to *end_node*.
22 Break
23. end else
24.end for

Algorithm 2: Q-learning-based routing algorithm (QLRA).

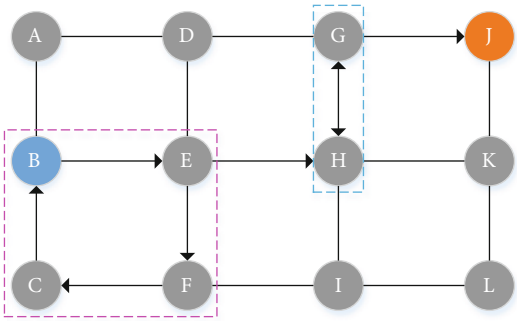

Figure 5: Repeated and invalid sequences during routing.

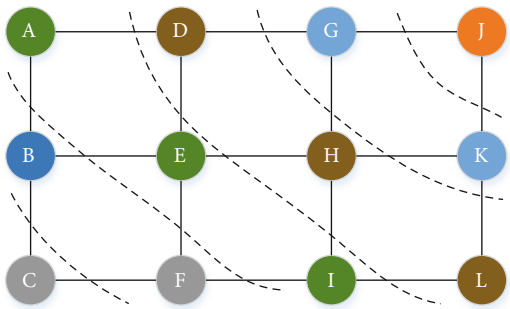

Figure 6: Split process of satellite network.

nodes in the same layer every time until all nodes of satellite network are updated. This scheme ensures that the *Q* value of each node is updated along a horizontal direction, and it destroys the condition of forming a loop between nodes. Therefore, the split operation accelerates the convergence speed of QLRA algorithm.

The traditional reinforcement learning updates the *Q* value of each node from front to back. But in the satellite network, we can obtain all the states of the satellite network in advance. Therefore, no matter what state the agent is in, we can know the next state of agent according to the actions taken by the agent. In addition, according to Equation (26), we know that updating the node's *Q* value from back to front make Q-table converge faster. Figure 7 illustrates how the agent updates the *Q* value of each node. Based on QLRA algorithm, we propose SQLRA algorithm with speed-up convergence strategy. The specific pseudo code of SQLRA is as follows:

where the function *BFS* in SQLRA represents the breadth first search algorithm. We search from the last node *end_node* to get the traversal sequences of the whole network. And we use function *Neighbour* to get the neighbour information of each node from the satellite network structure.

Figure 8 shows the convergence speed of SQLRA and QLRA. We observe Figure 8 that SQLRA converges faster than QLRA. We also observe that SQLRA needs 30 episodes to converge, and QLRA needs 60 episodes to converge. The main reason is that in the process of routing, our split-based speed-up convergence strategy reduces the invalid sequences. In addition, we update the *Q* value of the nodes from back to front, which further accelerate the convergence speed of SQLRA.

Input: *start_node*, *end_node*, *graph*, *R*, *users_req_list*.
Output: the optimal paths.
1. Initialize Q-table, α, γ, Episodes=*M*.
2. for user_req in users_req_list:
3. *flag* = PCA (*start_node*, *end_node*, *graph*).
4. if *flag* ==1:
5. *nodes_list* = ***BFS***(*graph*, *end_node*)
6. for *i*=1 to Episodes:
7. for *node* in *nodes_list*:
8. if *node* != *end_node*:
9. *neighbours* = ***Neighbour***(node).
10. for *neighbour* in *neighbours*:
11. Get the corresponding reward value generated by each parameter according to Eq. (18), (19), (20) and (21).
12. Obtain the total reward *r* based on Eq. (22).
13. Update Q(node, neighbour) based on Eq.(26).
14. end for
15. end if
16. end for
17. end for
18. Select the optimal path *path* from the converged Q-table based on Eq. (25).
19. Update the reward matrix *R* based on the consumption of link resources.
20. Update the structure of satellite network *graph* based on the consumption of link resources.
21. end if
22. else:
23. There is no path from *start_node* to *end_node*.
24 Break
25. end else
26.end for

Algorithm 3: Speed-up Q-learning-based routing algorithm (SQLRA).

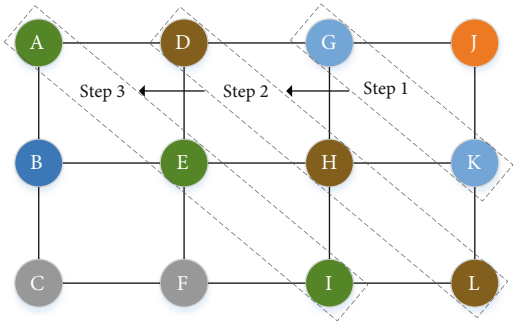

Figure 7: Mechanism for updating *Q* value from back to front.

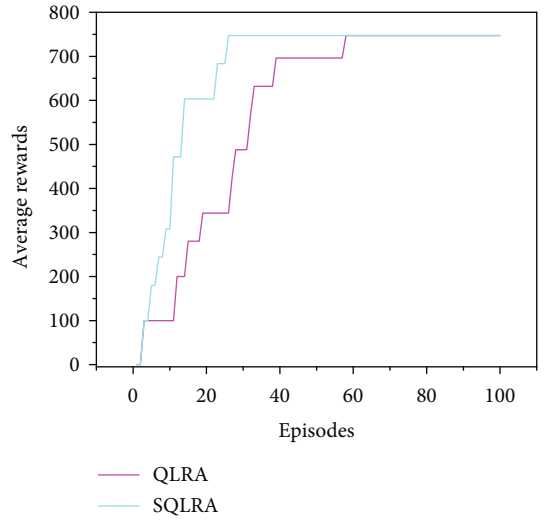

Figure 8: Convergence speed of QLRA and SQLRA with different episodes.

5. Performance Evaluation

In this section, we verify the effectiveness of the proposed algorithm SQLRA. First, the simulation environment and related parameters are introduced. Then, we compare SQLRA with QLAODV [42], QSR [43], OSPF [30], and ACO [44] in the performance of throughput, delay, bit error, and visible time. At last, we analyse and discuss the simulation results.

5.1. Experimental Parameter Settings. We conduct numerical simulations to verify the effectiveness of our proposed routing algorithm SQLRA. As for the satellite network used in this paper, we use satellite tool kit (STK) to simulate it. The satellite constellation adopts the Walker delta model. The satellite network consists of eight orbital planes, and every orbit has six satellites. There are 48 LEO satellites in total. The inclination angle of each satellite orbit is 45 degrees, and the altitude of satellite orbit is 650 km. Each satellite is only connected to its particular neighbour satellites. Please refer to Section 3 for more details. Due to the long

Table 1: Parameters of satellite constellation.

Parameters	Value
Number of satellites	48
Number of orbits	8
Orbit inclination	45°
Satellite number per orbit	6
Altitude of orbit	650 km
Simulation time	1400 s
Right ascension of ascending node	25°
Neighbours of a satellite	4

distance between satellites, the delay of satellite communication is mainly determined by the propagation delay of satellite links. Therefore, we mainly consider the propagation delay of satellite links in this paper. Furthermore, the propagation delay of satellite links in the same orbit is also different. For simplicity, we assume that the bit error rate of each satellite link follows a uniform distribution. And the bandwidth resources requested by each user follows a Poisson distribution. The specific parameters of satellite constellation are shown in Table 1. In our simulations, we use Pycharm as development tool. The environment is Win10 Operating System with 16 G RAM and 3.2 GHz CPU.

For QLRA and SQLRA, the learning rate α affects the convergence speed of algorithms. And the discount factor represents the impact of future rewards on the current result, which can prevent the agent from falling into the local optimum. The discount factor is between 0 and 1. The higher the value is, the more critical the future reward is. Through the analysis of experimental results, when the learning rate α and discount factor γ are set to 0.001 and 0.9, respectively, the convergence effect of the algorithms is the best. At this time, the convergence results of QLRA and SQLRA are the same. The weight of each parameter in the reward value can be obtained by analytic hierarchy process. Here, we set the values of θ, β, λ, and ω in Equation (11) as 0.30, 0.15, 0.18, and 0.37, respectively.

5.2. Results Analysis and Discussion. In this section, we evaluate the performance of SQLRA algorithm in two different scenarios: one is that all users communicate with each other by the same source satellite node and destination satellite node, and the other is that all users communicate with each other through different source satellite nodes and destination satellite nodes.

(1) Performance in communication scenario with same node pair

The users' requests in this scenario have the same source satellite node and destination satellite node. Because the source satellite node and destination satellite node of the selected paths are fixed, here we use average throughput, average delay, average bit error rate, and average visible time to measure the performance of routing algorithms. We compare the performance of algorithms with varying numbers of users.

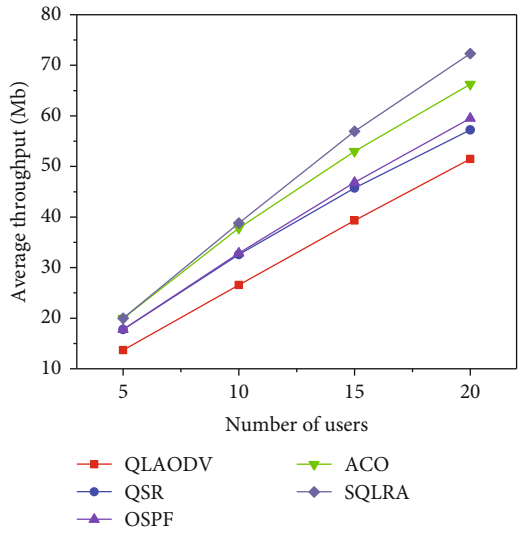

Figure 9: Average throughput with varying number of users.

(A) Average throughput analysis

Figure 9 shows the average throughput of different algorithms with varying number of users. From Figure 9, we can see that as the number of users increases, the average throughput obtained by all algorithms is increasing. At the same time, we can draw the following conclusions from Figure 9. First, QLAODV algorithm has the worst performance. Compared with AODV, QLAODV considers not only the number of hops and the delay but also the bit error rate and the available bandwidth. However, QLAODV still prefers to select the path with fewer hops when selecting the next hop. In satellite network, the distance between satellites in the same orbit is different from that between satellites in different orbits; the path with the minimal hops is not always the optimal path. Second, although both ACO and OSPF consider the same characteristics of satellite links in the process of routing, the performance of ACO is better than that of OSPF. The main reason is that OSPF is based on greedy strategy and is easy to fall into the local optimum, while ACO algorithm tries to find the global optimum as much as possible by using the positive feedback mechanism. Lastly, compared with other routing algorithms, our proposed SQLRA has the best performance. The reason is that in the process of path selection, SQLRA not only considers the states of current satellite links but also considers the impact of future rewards on the current selected links. In addition, the Q-table of SQLRA can converge after a certain number of iterations. Therefore, SQLRA mostly finds the optimal solution.

(B) Average delay analysis

We show in Figure 10 the average delay with different number of users. We observe that with the number of users

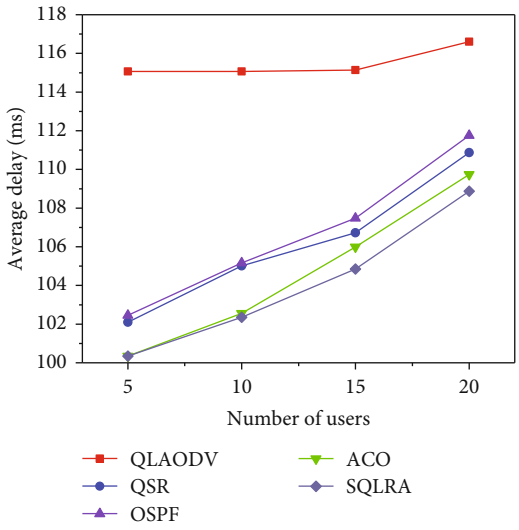

Figure 10: Average delay with varying number of users.

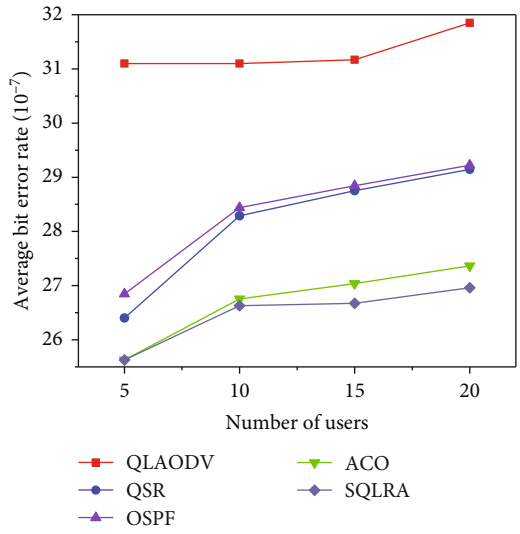

Figure 11: Average bit error rate with varying number of users.

increasing, the average delay obtained by all algorithms becomes larger. Moreover, we note that the curve of QLAODV grows slowly compared with other algorithms. For QLAODV, it prefers to select path with fewer hops. In addition, only when the current path is disconnected or saturated due to the consumption of users, QLAODV starts to select new path. Therefore, QLAODV does not choose new path frequently. This explains why the curve of QLAODV is mostly unchanged at the beginning. For QSR, OSPF, ACO, and SQLRA, they all consider the delay of links when calculating the cost function. Therefore, they prefer to choose the path with lower delay. As the number of users increases, the current paths cannot meet the needs of users, and these algorithms begin to select new paths. At this time, the delay of the selected paths is greater than that of the previously selected paths. This is the reason why the average delay of paths by QSR, OSPF, ACO, and SQLRA increases. Furthermore, we also observe that SQLRA performs better than ACO. The main reason is that ACO always achieves the suboptimal solution, and SQLRA mostly attains the global optimum.

(C) Average bit error rate analysis

Figure 11 plots the average bit error rate changes with varying number of users. As the number of users increases, the average bit error rate obtained by all algorithms presents an upward trend. Because these algorithms consider the bit error rate characteristic of the satellite links when selecting the path, the paths with the lower bit error rate are selected at first. As the resources of links are consumed by the increasing users, these algorithms begin to choose new paths which have larger bit error rate. Compared with ACO, SQLRA has the best performance. Even if the number of users is largest, the average bit error rate of SQLRA algorithm is lowest.

(D) Average visible time analysis

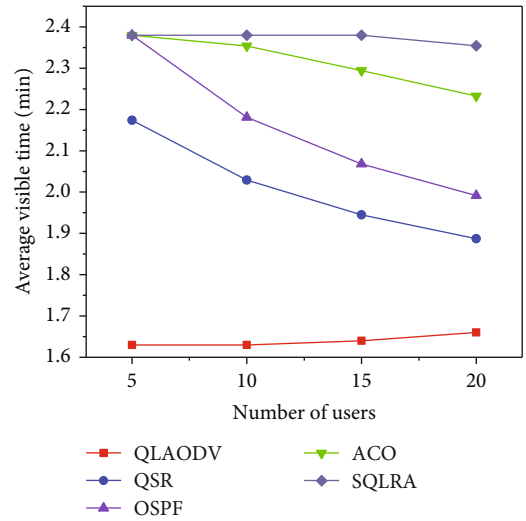

Figure 12: Average available time with varying number of users.

We show in Figure 12 the variation trend of the average visible time with varying number of users. We from Figure 12 observe that the average visible time of all algorithms shows a descend trend to varying degrees for algorithms QSR, OSPF, ACO, and SQLRA when the number of users is increasing. Considering that the visible time has an important impact on the performance of satellite network, these algorithms prefer to select paths with large visible time at the beginning. With more users accessing the network, the current paths cannot meet the requests of users. These algorithms start to select new paths with lower visible time. This explains why the average visible time of these algorithm gradually decreases. For QLAODV, the visible time of the links is not considered when selecting path. Therefore, when selecting the path, QLAODV algorithm does not prefer to choose the link with long visible time.

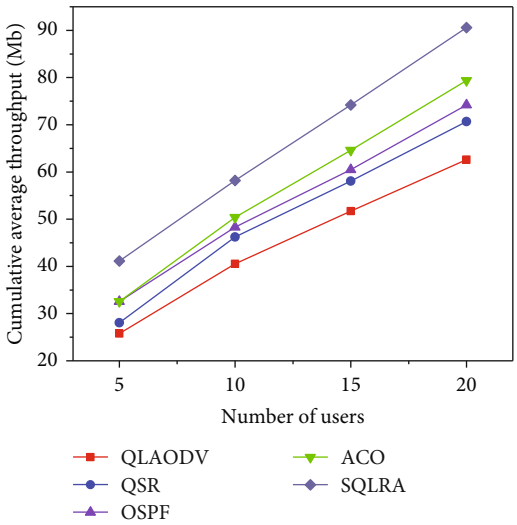

Figure 13: Cumulative average throughput with varying number of users.

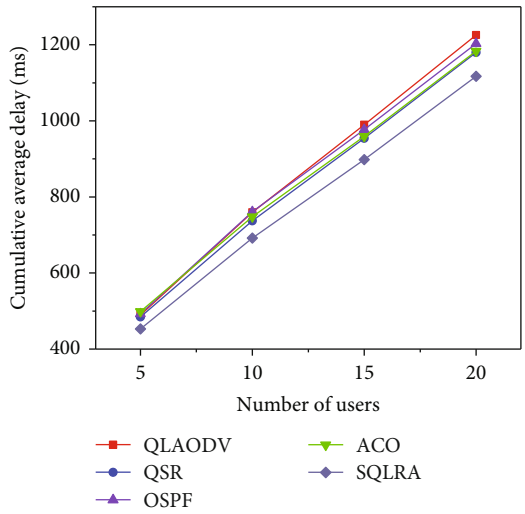

Figure 14: Cumulative average delay with varying number of users.

And this explains why the average visible time curve of QLAODV algorithm does not show a downward trend.

(2) Performance in communication scenario with different node pairs

The users' requests in this scenario have different source satellite nodes and destination satellite nodes. Considering the destination nodes of users are different, we use cumulative average throughput, cumulative average delay, cumulative average bit error rate, and cumulative average visible time as metrics, which can more clearly and reasonably evaluate the overall performance of the algorithms.

(A) Cumulative average throughput analysis

Figure 13 shows the relationship between the cumulative average throughput and the number of users. We note from Figure 13 that as the number of users increases, the cumulative average throughput of all algorithms presents an upward trend to varying degrees. Moreover, we also find that SQLRA has the best performance and QLAODV has the worst performance. The main reason is that when selecting the next hop, SQLRA considers the impact of the next link on the current link, and it attains the global optimum, while QLAODV does not consider the visible time of links when selecting the next hop. Furthermore, we also know that the visible time of the satellite link has a great impact on the network performance. When selecting the next hop, OSPF considers the available bandwidth, bit error rate, delay, and visible time. However, it adopts greedy strategy to select the next hop, which is easy to fall into local optimum. Compared with OSPF, ACO algorithm is initialized by random strategy, and it achieves global optimum as much as possible in the process of routing.

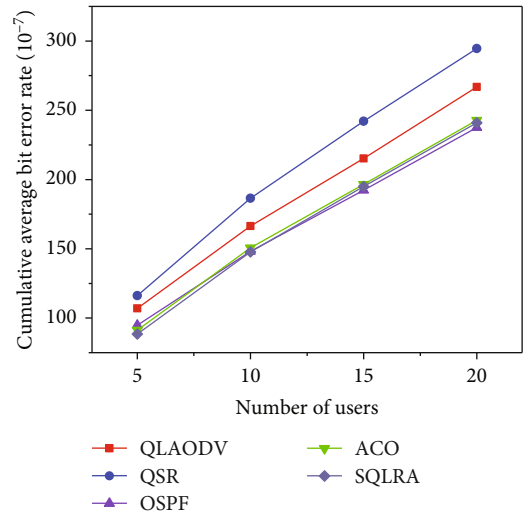

Figure 15: Cumulative average bit error rate with varying number of users.

(B) Cumulative average delay analysis

Figure 14 presents the change of cumulative average delay with varying number of users. As the number of users increases, the cumulative average delay obtained by all algorithms is increasing. Because users have different destinations, the paths selected by all algorithms are different. With the number of paths increasing, the total delay of the selected paths in the network increases. Therefore, the cumulative delay of the paths also increases. We note that SQLRA performs better than other algorithms. The main reason is that when the user's request is satisfied, SQLRA tends to choose the path with less delay, and it almost get the global optimum. We also observe that the performance of algorithms QSR, OSPF, and ACO is almost the same.

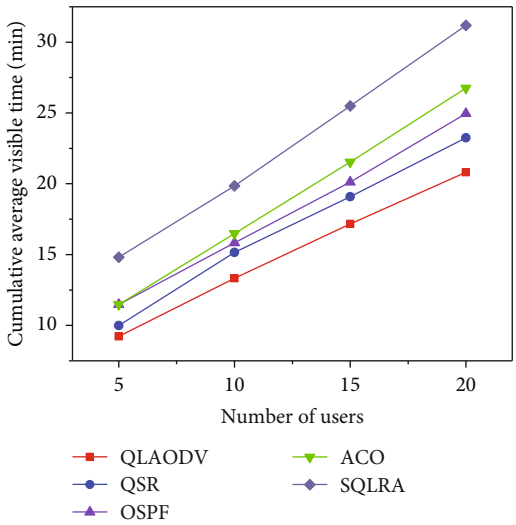

FIGURE 16: Cumulative average visible time with varying number of users.

The main reason is that although these algorithms all consider the delay of satellite links when computing the cost function, they use heuristic strategy to select the optimal path.

(C) Cumulative average bit error rate analysis

Figure 15 demonstrates the cumulative average bit error rate with varying number of users. We find from Figure 15 that the performance of QSR is worst and that of OSPF is best. We observe that although the performance of SQLRA is not the best, it is close to that of OSPF. Moreover, we also note that the performance of QLAODV is not worst compared with the curve in Figure 11. The main reason is that due to the paths of users are different, QLAODV algorithm begins to choose a new path when a new user arrives. This operation reduces the probability of selecting the path with high bit error rate. In addition, QLAODV algorithm considers the bit error rate when selecting new path.

(D) Cumulative average visible time analysis

Figure 16 shows cumulative average visible time with different number of users. From Figure 16, we see that as the number of users increases, the cumulative average visible time of all algorithms presents an upward trend to varying degrees. We also find that SQLRA performs better than other algorithms. When the number of users is 5, algorithms ACO and OSPF have the same performance. However, as the number of users increases, ACO performs better than OSPF. The main reason is that compared with ACO, OSPF is easier to fall into the local optimum. In the second scenario, the source nodes and destination nodes requested by users are different. As the number of paths increases, the total visible time of paths also increases. Therefore, the cumulative average visible time of the selected paths increases gradually.

By testing our proposed SQLRA algorithm in two different cases, we find that SQLRA performs better than other algorithms. In addition, we also find that SQLRA not only has good performance but also has strong robustness. In practice, we train the SQLRA algorithm at the ground control center, which reduces the consumption of computing resources and storage resources of satellites. Therefore, SQLRA algorithm is very suitable for dynamic satellite networks.

6. Conclusions

In this paper, we investigated the routing problem in STINs. We considered that selecting different satellite routes has an essential impact on the QoS of users and satellite network performance. We modelled the routing problem as a finite-state Markov decision process and proposed a routing algorithm based on Q-learning (QLRA). In addition, to solve the problem of slow convergence speed of QLRA algorithm, we proposed a split-based speed-up convergence strategy and designed a speed-up Q-learning based routing algorithm (SQLRA) and adopted a back-to-front update scheme to further improve the convergence speed of SQLRA algorithm. Moreover, we evaluated the performance of SQLRA in two different scenarios. Experimental results show that SQLRA algorithm has the best communication performance compared with other routing algorithms.

Although SQLRA algorithm performs better than other routing algorithms in two different scenarios, it does not have the ability of online learning. When the number of users in satellite network changes or some satellites do not work well, SQLRA needs to retrain model. In the future research, we are going to use deep neural network to design a routing algorithm with online learning ability. When the number of users changes or the link state of satellite network changes, SQLRA algorithm is able to update the existing model and reduce the training time as much as possible. In addition, traffic scheduling and satellite handoff management are also our future research issues.

Data Availability

The simulation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science Foundation of China (No. 61772385).

References

[1] F. Boccardi, R. W. Heath Jr., A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[2] H. Yao, L. Wang, X. Wang, Z. Lu, and Y. Liu, "The space-terrestrial integrated network: an overview," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 178–185, 2018.

[3] N. U. L. Hassan, C. Huang, C. Yuen, A. Ahmad, and Y. Zhang, "Dense small satellite networks for modern terrestrial communication systems: benefits, infrastructure, and technologies," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 96–103, 2020.

[4] J. P. Choi and C. Joo, "Challenges for efficient and seamless space-terrestrial heterogeneous networks," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 156–162, 2015.

[5] A. Guidotti, B. Evans, and M. Di Renzo, "Integrated satellite-terrestrial networks in future wireless systems," *International Journal of Satellite Communications and Networking*, vol. 37, no. 2, pp. 73–75, 2019.

[6] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: integrating terrestrial-satellite networks into 5G and beyond for data offloading," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 47–62, 2019.

[7] K. An and T. Liang, "Hybrid satellite-terrestrial relay networks with adaptive transmission," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 12448–12452, 2019.

[8] A. A. Dowhuszko, J. Fraire, M. Shaat, and A. Pérez-Neira, "LEO satellite constellations to offload optical terrestrial networks in placement of popular content in 5G edge nodes," in *2020 22nd International Conference on Transparent Optical Networks, International Conference on Transparent Optical Networks-ICTON*, pp. 1–6, Bari, Italy, 2020.

[9] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 756–767, 2002.

[10] C. Hedrick, "Routing information protocol," *Request for Comments*, vol. 1058, 1988.

[11] S. Anamalamudi, A. R. Sangi, M. Alkatheiri, and A. M. Ahmed, "AODV routing protocol for cognitive radio access based Internet of Things (IoT)," *Future Generation Computer Systems-the International Journal of Escience*, vol. 83, pp. 228–238, 2018.

[12] W. Liu, Y. Tao, and L. Liu, "Load-balancing routing algorithm based on segment routing for traffic return in LEO satellite networks," *IEEE Access*, vol. 7, pp. 112044–112053, 2019.

[13] X. Qi, B. Zhang, and Z. Qiu, "Joint rate control and load-balancing routing with QoS guarantee in LEO satellite networks," *IEICE Transactions on Communications*, vol. E103B, no. 12, pp. 1477–1489, 2020.

[14] Z. Na, Z. Pan, X. Liu, Z. Deng, Z. Gao, and Q. Guo, "Distributed routing strategy based on machine learning for LEO satellite network," *Wireless Communications & Mobile Computing*, vol. 2018, pp. 1–10, 2018.

[15] M. Majd and R. Safabakhsh, "Correlational convolutional LSTM for human action recognition," *Neurocomputing*, vol. 396, pp. 224–229, 2020.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[17] C. X. Jiang, H. J. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[18] N. Kato, Z. M. Fadlullah, B. Mao et al., "The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 146–153, 2017.

[19] F. Tang, B. Mao, Z. M. Fadlullah et al., "On removing routing protocol from future wireless networks: a real-time deep learning approach for intelligent traffic control," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 154–160, 2018.

[20] R. F. Dhila, T. M. Hamdani, and A. M. Alimi, "A multi objective particles swarm optimization algorithm for solving the routing pico-satellites problem," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1402–1407, Seoul, Korea, 2012.

[21] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2019.

[22] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.

[23] C. Jiang and X. Zhu, "Reinforcement learning based capacity management in multi-layer satellite networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4685–4699, 2020.

[24] V. Kirilin, A. Sundarrajan, S. Gorinsky, and R. K. Sitaraman, "RL-cache: learning-based cache admission for content delivery," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2372–2385, 2020.

[25] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco, and F. Liberal, "Virtual network function placement optimization with deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 292–303, 2020.

[26] J. Liu, R. Z. Luo, T. Huang, and C. W. Meng, "A load balancing routing strategy for LEO satellite network," *IEEE Access*, vol. 8, pp. 155136–155144, 2020.

[27] S. Geng, S. Liu, Z. Fang, and S. Gao, "An optimal delay routing algorithm considering delay variation in the LEO satellite communication network," *Computer Networks*, vol. 173, article 107166, 2020.

[28] S. Wei, H. Cheng, M. Liu, and M. Ren, "Optimal strategy routing in LEO satellite network based on cooperative game theory," in *International Conference on Space Information Network*, pp. 159–172, Singapore, 2017.

[29] Z. Jiang, C. Liu, S. He, C. Li, and Q. Lu, "A QoS routing strategy using fuzzy logic for NGEO satellite IP networks," *Wireless Networks*, vol. 24, no. 1, pp. 295–307, 2018.

[30] T. Pan, T. Huang, X. Li, Y. Chen, W. Xue, and Y. Liu, "OPSPF: orbit prediction shortest path first routing for resilient LEO satellite networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, 2019.

[31] L. Hao, P. Ren, and Q. Du, "Satellite QoS routing algorithm based on energy aware and load balancing," in *2020 12th International Conference on Wireless Communications and Signal Processing*, pp. 685–690, Wuhan, China, 2020.

[32] Z. Ji, S. Wu, C. Jiang, D. Hu, and W. Wang, "Energy-efficient data offloading for multi-cell satellite-terrestrial networks," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2265–2269, 2020.

[33] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2019.

[34] N. Torkzaban, A. Gholami, J. S. Baras, and C. Papagianni, "Joint satellite gateway placement and routing for integrated satellite-terrestrial networks," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, 2020.

[35] H. Xu, D. Li, M. Liu, G. Han, and C. Xu, "A hybrid routing algorithm in terrestrial-satellite integrated network," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 90–95, Chongqing, China, 2020.

[36] Q. Guo, R. Gu, T. Dong et al., "SDN-based end-to-end fragment-aware routing for elastic data flows in LEO satellite-terrestrial network," *IEEE Access*, vol. 7, pp. 396–410, 2019.

[37] Y. Liu, D. Lu, G. Zhang, J. Tian, and W. Xu, "Q-learning based content placement method for dynamic cloud content delivery networks," *IEEE Access*, vol. 7, pp. 66384–66394, 2019.

[38] S. Pan, P. Li, D. Zeng, S. Guo, and G. Hu, "A Q-learning based framework for congested link identification," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9668–9678, 2019.

[39] Z. Wei, F. Liu, Y. Zhang, J. Xu, J. Ji, and Z. Lyu, "A Q-learning algorithm for task scheduling based on improved SVM in wireless sensor networks," *Computer Networks*, vol. 161, pp. 138–149, 2019.

[40] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 247–257, 2020.

[41] M. Alhabo, L. Zhang, and N. Nawaz, "GRA-based handover for dense small cells heterogeneous networks," *IET Communications*, vol. 13, no. 13, pp. 1928–1935, 2019.

[42] C. Wu, K. Kumekawa, and T. Kato, "Distributed reinforcement learning approach for vehicular ad hoc networks," *IEICE Transactions on Communications*, vol. 93-B, no. 6, pp. 1431–1442, 2010.

[43] T. Li, H. Zhou, H. Luo, and S. Yu, "SERvICE: a software defined framework for integrated space-terrestrial satellite communication," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 703–716, 2018.

[44] H. Alayed, F. Dahan, T. Alfakih, H. Mathkour, and M. Arafah, "Enhancement of ant colony optimization for QoS-aware web service selection," *IEEE Access*, vol. 7, pp. 97041–97051, 2019.