

Received July 12, 2020, accepted August 4, 2020, date of publication August 18, 2020, date of current version September 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017615

A Load Balancing Routing Strategy for LEO Satellite Network

JIANG LIU^{1,2}, RUIZHI LUO¹, TAO HUANG^{1,2}, (Member, IEEE), AND CHUNWEI MENG¹

¹State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

²Purple Mountain Laboratories, Nanjing 211111, China

Corresponding author: Jiang Liu (liujiang@bupt.edu.cn)

This work was supported by the National Key Research and Development Project under Grant 2019YFB1803103 and Grant 2018YFB1800501.

ABSTRACT The LEO network composed of massive satellite nodes is a hot research field at present. In such a network, the problem of load balancing should be reconsidered due to the large number of satellite nodes, the frequently changing topology, and the uneven distribution of global ground users. To solve such problem, a selective iterative Dijkstra algorithm (SIDA) is proposed to optimize the path finding process by reducing the reuse frequency of nodes. On the basis of SIDA, a selective split load balancing strategy (SSLB) is proposed to solve the link congestion problem in low latitude LEO network. SSLB is a distributed strategy, which diverts the traffic flow of congestion nodes to neighboring nodes. SSLB reduces the burden of neighbor nodes, which need the less demand of computing resources and signaling interaction. Simulation results show that SIDA reduces the number of congestion links and balances the load distribution, while SSLB can effectively suppress the number of new congestion links after traffic diversion. These algorithms provide a load balancing solution for LEO satellite network.

INDEX TERMS Satellite communication, routing, load management, network topology.

I. INTRODUCTION

The satellite communication network is a network system formed by organizing satellites as network nodes, which connects the terrestrial network and the space network. With the development of on-satellite processing techniques and inter-satellite link technology, satellites can process information independently and communication directly with each other, which increase the possibility of satellites forming a network. In the current satellite network system, it can be divided into Low Earth Orbit satellite system(LEO), Medium Earth Orbit satellite system(MEO) and High Earth Orbit satellite system(HEO) according to the satellite height. Among them, LEO network has become a new hotspot for large-scale satellite network because of its low transmission delay and low cost. However, there are two major problems in LEO network:

- The off orbit link is disconnected due to antenna angle offset in high latitude area, which leads to the periodic and frequent changes of topological structure [1];
- The proportion of land area and ocean area is uneven, and the distribution and demands of users are different,

which leads to the unbalanced traffic distribution of LEO network [2].

In order to meet the first challenge and reduce the requirements of LEO satellite computing capacity, the snapshot sequence algorithm is usually adopted [3]. The operation period of LEO satellite can be divided into several time slices. In each time slice, the network topology is different and is considered static, which is called topology snapshot. The central node calculates the routing paths of all topology snapshots and uploads them to the satellite node. The nodes execute synchronously and switch periodically. However, considering the second challenge, the unbalanced traffic distribution will lead to link congestion, which cannot be handled by the snapshot sequence algorithm, resulting in the decrease of network balancing. Meanwhile, even if some links are so busy, there are still some links in the network that are idle. To solve the above problems, we propose a load balancing solution, which can avoid link congestion and reduce the number of congestion links. The simulation results show that this solution can effectively balance the network load. The main contributions of the paper are listed as follows.

- We propose a selective iterative shortest path algorithm which improves Dijkstra algorithm and is applied to calculate the route path of topology snapshot. This algorithm can use every link in Leo network evenly,

The associate editor coordinating the review of this manuscript and approving it for publication was Haipeng Yao¹.

so as to realize the load balance in the initialization stage of routing.

- We propose a distributes load balancing strategy for sudden link congestion. The strategy can reasonably shunt the traffic of congestion nodes and reduce the new burden on other regional networks on the basis of reducing the local congestion.

The rest is organized as follows. Related work will be summarized in section II. According to concept and application in section II, load balancing strategy are introduced In section III. From the characteristics of LEO network, the system model is constructed and the problem is defined clearly. Section IV introduces the load balancing strategy in detail, including algorithm process, application scenarios, complexity analysis. Section V provides the simulation results. The results show that the performance of load balancing is improved in terms of the number of link congestion and the distribution of the remaining bandwidth. Conclusion is shown in section VI.

II. RELATED WORK

In order to achieve load balancing in satellite networks, two important procedures need to be completed, namely, information exchange and decision making. The previous efforts on satellite load balancing have been trying to optimize the decision making while minimize the information exchange cost.

In [4], Kim Y S proposed to allocate the remaining bandwidth of lightly-loaded neighbors to congested satellites, so that link congestion can be avoided. Compared to only consider bandwidth of neighboring nodes, the author of proposed to further consider the topology of the satellite network to make better load balancing decision. In [5], each node in the network maintains the information of the whole network, and they would filter out heavy-loaded nodes when making routing decision. However, the scheme requires up-to-date global information at each node, which is not easy to achieve. In [6], the author narrowed the range of information exchange to neighboring satellite nodes. More specifically, congested satellite nodes send signals to inform their neighbors, and their neighbors would look for an alternative path.

Another approaches aimed at solving the global state synchronization directly. The authors of [7] proposed an agent-based load balancing routing scheme, and the scheme introduces the application of agent technology in LEO satellite network. Later, the authors of [7] further improved the solutions, and they proposed an agent-based multi service routing scheme for LEO satellite networks in [8]. The proposed scheme uses distributed method to complete packet forwarding. For [7] and [8], both requires frequent signaling to achieve global state synchronization, which can be heavy burden for the network. Therefore, in [9], the authors proposed a load balanced on demand routing scheme. The proposed scheme limits the range of state synchronization, and each node generates routing paths within the predefined range.

With global information properly synchronized, solutions [11] and [12] further improved the load balancing performance. In [11], the author proposed the hybrid load balancing routing, where they designed a long-distance traffic detour method. The method can predict the regional and real-time network states that are prone to cascading congestion, according to which the scheme calculates the multipath route. Considering the congestion state and end-to-end delay of local nodes, neighboring nodes, and other relay nodes, the author of [12] combined offline calculation and online adjustment to recalculate the multipath routing table [12].

The preceding works either requires frequent signaling or intensive computing, which can be challenging to be implemented on the satellite. Therefore, the authors of [10] proposed a low complexity routing algorithm scheme (LCRA), which selects the next hop route by using the mesh structure of LEO satellite networks.

As can be seen, previous efforts require either signaling or computing to achieve satisfying load balancing performance. In this article, our goal is to develop a analyze the characteristics of LEO satellite networks, and propose an light-weight while efficient load balancing scheme. More specifically, we use LCRA for reference to select the next hop route and propose a better strategy. This strategy further reduces the computational complexity and the frequency of signaling interaction, which is better adapted to LEO's low computing ability and high communication cost. By taking account of the link state between local node and adjacent nodes, as well as the information of LEO topology, we can eliminate congestion in time. The proposed load balancing strategy can be easily extended to support different routing algorithms.

III. LEO SATELLITE NETWORK NETWORKING MODEL

Our algorithm is based on the general LEO model. The model is based on the LEO constellation which has multiple orbits and an equal number of satellites distributed evenly on each orbital surface. LEO network, as the transmission network, carries the traffic of terrestrial users. The whole data transmission process can be divided into three parts. Firstly, the ground user or gateway station transmits data to the satellite. The ground is divided into rectangular traffic cells according to location and constellation and each traffic cell should be bind to one satellite anytime which is responsible for the communication within the cell. Then, each satellite communicates with several neighboring satellites in the same orbit or adjacent orbits through Inter-Switch Link(ISL). In this process, the candidate data transmission path will be selected first according to the logical location of the satellite, and then the final path will be selected according to the link failure and link congestion states. Finally, the satellite sends data to the ground user or gateway station to complete the whole process of data transmission.

The LEO constellation scenario is shown in Figure 1. There is a latitudinal deviation between adjacent orbiting satellites. In the low latitude area, the ISL of adjacent orbit

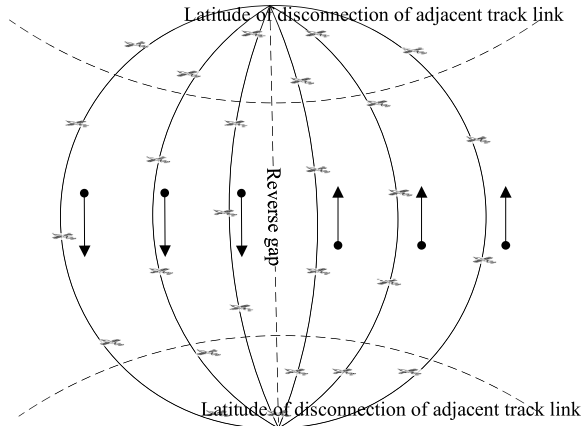


FIGURE 1. Structure diagram of LEO satellite constellation.

will be connected. In the contrary, the ISL will be disconnected in the high latitude area since the left and right orbits will be switched crossing the pole. In addition, due to the opposite movement between the first rising and last declining orbit satellites, there is no ISL between the two orbits, which is called the reverse gap [13]. Supposing the LEO satellite constellation has $N=n \times m$ satellites, where n is the number of orbits, and m is the number of satellites in each orbit. Coordinate (n, m) is defined as the logical location of satellite node routing, and we can obtain the topology matrix I of satellite node. Each satellite node is numbered as i , $i \in [0, N-1]$.

When choosing the routing path of the satellite nodes, link failure and link congestion need to be considered. $F^{(i)}$ stands for the total amount of traffic carried by satellite node i . We can locate the node $F^{(i)}$. k is the order of orbits. r is the order of satellites in orbit.

$$\begin{cases} k = \left\lfloor \frac{i}{m} \right\rfloor \\ r = i \bmod m \end{cases} \quad (1)$$

If node i and node j are adjacent nodes ($i \neq j$, $i, j \in [0, N-1]$) and ISL exists between node i and node j , $E^{(i,j)}$ stands for the remaining bandwidth of the link between the two nodes. Assuming that the communication between node i and node j shares one communication channel, the threshold value of ISL residual bandwidth is set as $Q_{threshold}$. When $E^{(i,j)} < Q_{threshold}$, the ISL is considered to occur congestion. When calculating the routing path, all ISL link weights are considered to be the same.

IV. LOAD-BALANCING ROUTING ALGORITHM

A. SELECTIVE ITERATIVE DIJKSTRA ALGORITHM (SIDA)

As we can see in the model, the topology of LEO network is similar to a symmetrical grid, which induce to the load balancing inefficiency of the traditional Dijkstra algorithm, since multiple shortest paths from the same node may use the same link. Therefore, we propose SIDA algorithm to solve this problem.

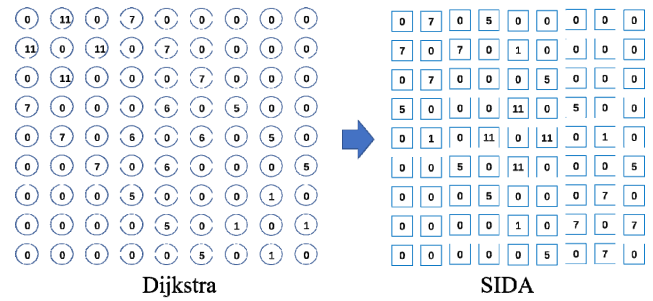


FIGURE 2. Advantage of SIDA to Dijkstra.

SIDA is an optimization algorithm based on the Dijkstra algorithm. Dijkstra algorithm is used to find the shortest path of the remaining nodes by traversing the weight of the nodes. There are two traversals. The first traversal is used to record the weight of the source node to all other nodes. And the second traversal is used to update the minimum weight of the source node to other nodes based on the current shortest path. However, the second traversal is performed in the order of node number from small to large, in which the small numbered nodes are considered first and then the large numbered nodes. For the highly symmetrical regular network structure, when the link weight is the same, new paths will be added on the basis of the original shortest path, resulting in the repeated use of some links.

However, SIDA changes the traversal mode in Dijkstra algorithm, from positive order to reverse order, the larger numbered nodes can be given more priority to increase the utilization rate. In order to ensure the utilization rate of nodes with different numbers is as equal as possible, in SIDA algorithm, when the destination node number is larger than the source node, the second traversal adopts positive order traversal. And when the destination node number is smaller than the source node, the second traversal adopts reverse order traversal. In this way, when calculating the routing path between all nodes, half of the nodes can be traversed in positive order, and half of the nodes can be traversed in reverse order.

For LEO network topology, due to its highly symmetrical shape, there are multiple equivalent shortest paths between the source node and the destination node if the link weight is normalized, which have the same minimum hops. In order to avoid multiple routing paths to select the same node as the next hop, SIDA algorithm deals with the nodes that are repeatedly used. If a node is already used in a path, the weight of the source node to this node should be increased. In this way, the node can be avoided from being selected again in the next routing path calculation, so that other alternative shortest paths can be generated without an increase of hop.

SIDA can be used to calculate the snapshot route in the snapshot sequence algorithm. The algorithm can use every link evenly, which make the load balance. The main advantages of SIDA is shown in Figure 2.

As shown in Figure 2, we use a 9×9 grid as an example. The number on the node identifies the number of times the

node has been used by the path after running the algorithm. In Dijkstra's algorithm, the upper left part of the node is used is obviously greater than the lower right part nodes. In SIDA, the whole network is relatively evenly used.

B. SELECTIVE SHUNT LOAD BALANCING STRATEGY (SSLB)

After using SIDA, the LEO network will have better load balance. However, there is still the possibility of congestion. Therefore, we further proposed SSLB to handle the congestion distributed without much signal interaction, which is suitable for the LOE network.

SSLB is a distributed load balancing strategy that uses the route path calculated by the snapshot sequence algorithm. After the link congestion occurs, the traffic of the congestion node is shunted to the neighbor nodes. SSLB can reduce the requirement for computing resources on the satellite, the signaling overhead and the possibility of congestion in other areas of the network after shunting.

1) APPLICATION SCENARIO

SSLB uses the routing path calculated by SIDA. This strategy aims to distribute traffic to neighbor nodes. If the Dijkstra algorithm is used to calculate the routing path, some links will be included in the routing path repeatedly. Therefore, even if the traffic is diverted to the neighbor node, the next hop of the route from the neighbor node to the destination node may still pass through the congestion node, thus causing the traffic to be transferred back and forth between the neighbor node and the congestion node.

SIDA avoids repeated use of a link. To some extent, it avoids the situation that the routing path between a node and its neighbors to the same destination node is almost identical. Compared with Dijkstra algorithm, the probability that the next hop of its routing path is still a congestion node is far reduced, which greatly increases the enforceability of shunting.

SSLB reduces the link congestion in the low latitude area. In order to reduce packet jitter after shunting, the strategy needs to ensure the minimum hop number of the routing path will not increase too much after shunting. In the low latitude area, and the topology is a highly symmetrical and uniform grid. There are multiple equivalent shortest paths between the source node and the destination node. After the traffic is divided into neighbor nodes, the minimum number of hops increases by two at most. However, ISL is disconnected between satellites in adjacent orbits in high latitude. If there is congestion in the high latitude area, it must be the link in the same orbit. At this time, there is almost no equivalent multipath route. Therefore, SSLB will play a better role at low latitudes. Overall, it will also alleviate the congestion that appears in LEO network.

2) SHUNTING NODE SELECTION

Regardless of the scenario of link disconnection in high latitudes, where SSLB cannot be used, the satellite node have three to four neighbor nodes. $F^{(i)}$ can be located in (k, r) .

Algorithm 1 Calculate the next hop

Require: the location of destination node and congestion node: (kd, rd) , (kc, rc) ; All available neighbor nodes are located in low latitude: A, B; the location of all neighbor nodes of congestion node: (kA, rA) , (kB, rB) ; Remaining bandwidth of all neighbor nodes: EAC, EBC

Ensure: the next hop: n

If congestion link is the same track link:

If there are two neighbors:

If $kd == kc$

$n = \max(EAC, EBC)$

Else if $kd > kc$:

$n = (kc+1, r)$

Else:

$n = (kc-1, r)$

Else: There is only one neighbor A

$n = A$

If congestion link is adjacent track link:

If there are two neighbors:

If $rd == rc$:

$n = \max(EAC, EBC)$

Else if $rd > rc$:

$n = (kc, r+1)$

Else:

$n = (kc, r-1)$

Else: There is only one neighbor A

$n = A$

if the link between node $(k, r+1)$ is congested, the neighbor node $(k, r-1)$ can't accept the shunted traffic flow. Because it will inevitably lead to an increase in the minimum number of hops by two. Therefore, this type of neighbor node should not be used. Similarly, if the link between node $(k+1, r)$ is congested, the neighbor node $(k-1, r)$ can't accept the shunted traffic flow, too.

In order to ensure that the hops of the routing path do not increase after the congested node diverts the traffic to the neighbor, we need to consider the relative position of the longitude and latitude of the destination node and the congestion node (Algorithm 1). Congested nodes have one or two neighbors that can accept shunted traffic. We prefer the neighbor node with fewer hops of the routing path to the destination node. For different traffic flow, different neighbor nodes can be selected. If there are two neighbor nodes have the same number of the hops for both routing paths, only one neighbor node is selected to ensure that there is not too much traffic diversion. In order to reduce the probability of link congestion in the neighbor node, the neighbor node with larger residual bandwidth is more advantageous.

3) CALCULATION OF SHUNT FLOW

The satellite nodes at both ends of the congestion link is recorded as X and Y . When the traffic is shunted, in order to ensure that the ISL between the congested node and the neighbor node will not congest again, the value of shunted

traffic needs to be limited. Therefore, the congested node with a small traffic is preferred for diversion. If the congestion cannot be solved, then consider the diversion of large traffic nodes.

The traffic value of node X is F^x , node Y is F^y , $F^x < F^y$, link residual bandwidth is E , link residual bandwidth threshold is $Q_{threshold}$. If $E < Q_{threshold}$, the link is considered to be congested. If the ratio of shunted traffic of small flow nodes is α , the ratio of shunted traffic of large flow nodes is β . In a special state, the congested node has only one traffic flow. The minimum residual bandwidth between the congested node and the neighbor node must be larger than the value of shunted traffic.

Step 1: Priority should be given to node with a small value of traffic: satellite X . The neighbor node is denoted as P , and the minimum remaining bandwidth between X and P is E^{XP}

$$\alpha = \frac{E^{XP} - Q_{threshold}}{F^x} \quad (2)$$

According to the split ratio α , and the traffic transmitted by satellite X according to the original route is (3):

$$(1 - \alpha)F^x \quad (3)$$

If E^{XY} is still less than $Q_{threshold}$, continue with step 2

Step 2: Satellite Y is also involved in the shunting. Set the neighbor node as Q , and the minimum remaining bandwidth between Y and Q as E^{YQ}

$$\beta = \frac{E^{YQ} - Q_{threshold}}{F^y} \quad (4)$$

According to the split ratio β , and the traffic transmitted by satellite Y according to the original route is (5):

$$(1 - \beta)F^y \quad (5)$$

The above analysis takes into account the special case that the congested node has only one traffic flow. The value of residual bandwidth minus $Q_{threshold}$ between the congested node and the neighbor node must be larger than the value of the shunted traffic. Generally, there are multiple traffic flows in satellite nodes, and the traffic will be distributed between two neighbor nodes.

4) STRATEGY COMPLEXITY ANALYSIS

- Computational complexity: SSLB uses the residual bandwidth and the value of traffic flow to calculate the ratio of shunted traffic, and the time complexity is $O(I)$.
- Signaling interaction frequency: the node periodically detects the remaining bandwidth between the neighbor nodes. The congested node only needs one signaling interaction to determine the neighbor node that can accept the shunted traffic. The complexity of signaling interaction is lower.

V. SIMULATION

A. SIMULATION SCENARIO

The proposed algorithm is applied to LEO54 constellation with satellite links, which is designed by combining the characteristics of Walker Constellation and LEO topology. The LEO54 network topology has the following characteristics

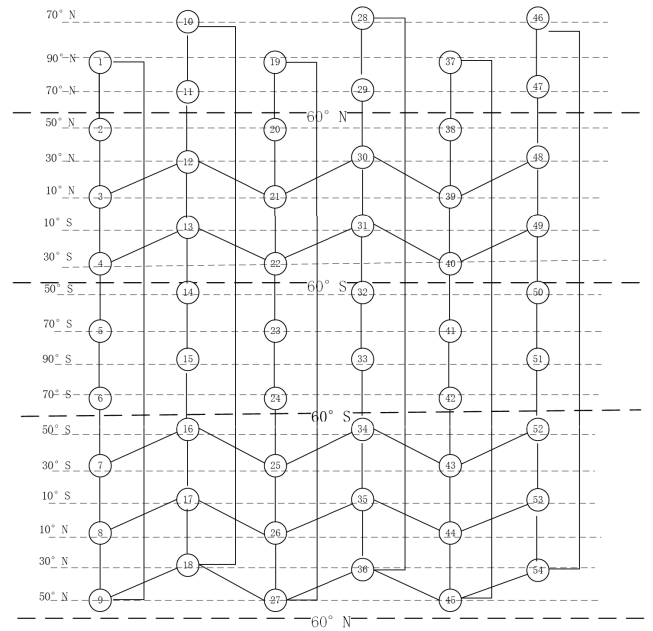


FIGURE 3. The topological structure of a time slice.

- Each satellite node has four ISLs, including two same orbit links and two adjacent orbit links.
- The constellation has six orbital planes and 54 satellites. Nine satellites are evenly distributed on each orbital plane.
- When the latitude position of the satellite is higher than 60° , the adjacent orbit link will be disconnected.
- This constellation is polar orbit. The orbit has no angular deflection at the pole.

LEO constellation operates regularly, which has the characteristics of high dynamic but periodic change. We can calculate each time slice from the changing state of satellite topology. In this article, SIDA is proposed and applied to the algorithm of the snapshot sequence. Therefore, this section can use the topological snapshot of one time slice to verify the feasibility of the algorithm. Shown in Figure 3:

- There are 54 satellites, numbered from 1 to 54. Set node 1 at the North Pole of the eastern hemisphere at 90°N ;
- The same orbit satellite is connected end to end. The latitude difference between adjacent satellites in the same orbit is 40° ;
- The latitude phase difference of adjacent orbital links is $\pm 20^\circ$;
- The area above 60°N and 60°S is high latitude area, and the adjacent track link is disconnected.

Some traffic flows have higher requirements for link bandwidth but are less sensitive to delay, such as large file transmission. Generally, this type of traffic flow accounts for 10% of the total flow but accounts for 80% of the bandwidth. However, some traffic flows are sensitive to the transmission delay, but occupy a smaller link bandwidth, such as voice calls and so on. In order to simulate different traffic flows in LEO network, we generation the traffic distribution in Gamma distribution.

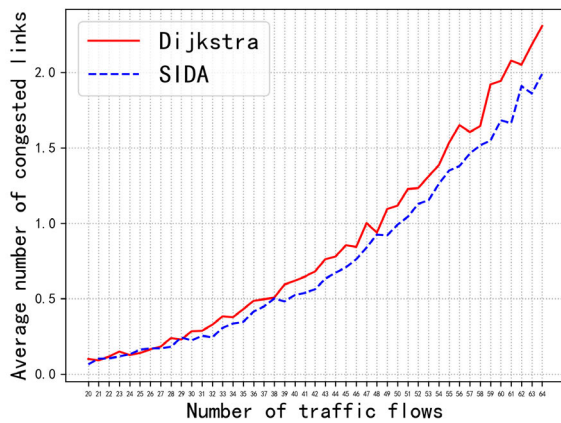


FIGURE 4. The average number of link congestion under the same number of traffic flows (Dijkstra algorithm and SIDA).

Based on this gamma distribution, N-group traffic flow is generated. The simulation assumes that several users send different traffic flows at the same time. Each group of experiments assumes that N-users send traffic flows. The parameters of each traffic flow are as follows:

- Source node(S): source satellite node mapped by user IP;
- Destination node(D): the final hop satellite node of the destination IP mapping of the traffic flow;
- Value of traffic flow(F): traffic value of each traffic flow;
- The traffic flow paths of different users are different.

B. SIMULATION RESULT

In this section, we evaluate the performance of the proposed algorithm applied to satellite topological networks with different traffic distributions. The programming simulation is used to simulate network scenario and implement the algorithm. We take the residual bandwidth of link and the average number of congested links as the indexes to evaluate the load balancing of satellite networks, and compare them with other algorithms to prove the superiority of the proposed algorithm in load balancing.

1) RESULTS AND ANALYSIS OF AVERAGE NUMBER OF CONGESTED LINKS

a: SIDA

When the utilization rate of links in the network is more uniform, the traffic will be more evenly distributed in all parts of the network. Thus, the probability of congestion will also be reduced. This experiment uses Dijkstra algorithm as a comparison (the links weight in SIDA and Dijkstra are both normalized). Under the same network topology and the same traffic flow condition, the average congested link number of the network using the SIDA is tested to see whether it is optimized.

Under [20], [64] group traffic flow, the average number of congested links of the two algorithms is shown in Figure 4. Even if the number of traffic flow in the network is small, link congestion may occur. This is because the source node and the destination node are randomly generated, and there are

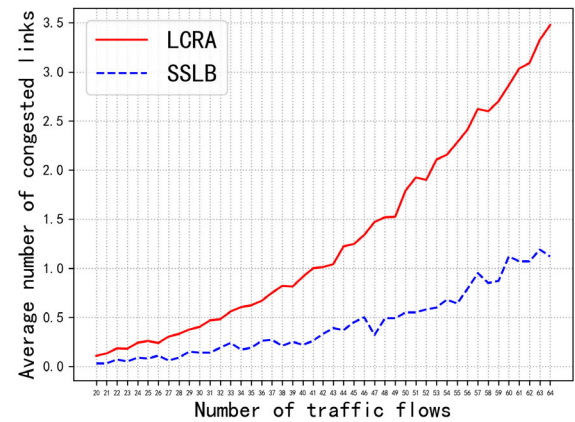


FIGURE 5. The average number of link congestion under the same number of traffic flows (SSLB and LCRA).

several traffic flows on one link. In the traffic flow of [20], [30] groups, SIDA and Dijkstra algorithm both have oscillation in the average number of congested links. The amplitude is similar, and there is no significant difference in the value. In the traffic flow of [31], [64] groups, the average number of congested links of the Dijkstra algorithm increases by oscillation, while SIDA shows a stable and smooth increase. At the same time, the average number of congested links under Dijkstra algorithm is gradually more than that under SIDA.

Therefore, when traffic flow increases, the average number of congestion links under SIDA is gradually lower than that under Dijkstra algorithm. Moreover, the increase in the average number of congested links under SIDA is relatively gentle, which shows the stability of the algorithm.

b: SSLB

When link congestion occurs, the traffic of the congestion node is partially diverted to the neighbor node, which may aggravate the load of the neighbor node and cause new link congestion. This experiment is divided into two parts. In the first part, LCRA is used as a reference to calculate the average the average number of congestion links in the network under the two strategies. Under the condition of the same traffic flow, a more effective shunting strategy can make the overall traffic distribution of the network more balanced and the number of the link congestion less.

In the second part, SSLB is used to solve the problem of link congestion in low latitude areas, and the average number of link congestion under different conditions is calculated. Then the effectiveness of SSLB is analyzed.

We use LCRA for reference to select the next hop routing node. Considering the residual bandwidth of neighbor nodes, an optimization strategy SSLB is proposed. It can be seen from Figure 5 that the average link congestion of SSLB is much smaller than that of LCRA. SSLB shows better load balancing performance.

SSLB is used to solve the link congestion in the low latitude area. When the link is congested, the total number of congested links in the network is recorded as C, and the

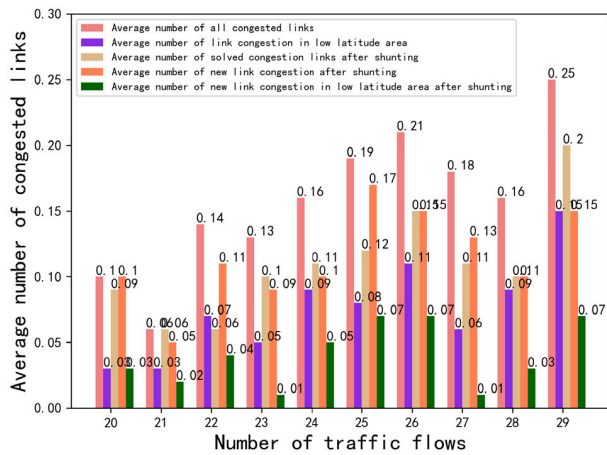


FIGURE 6. Link congestion state before and after shunting (under [20]–[29] traffic flows).

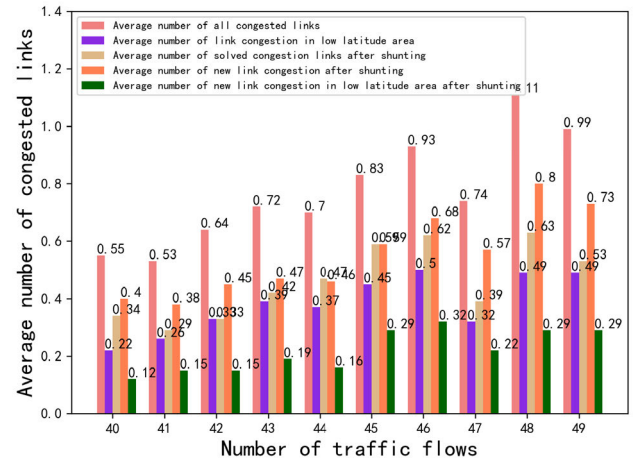


FIGURE 8. Link congestion state before and after shunting (under [40]–[49] traffic flows).

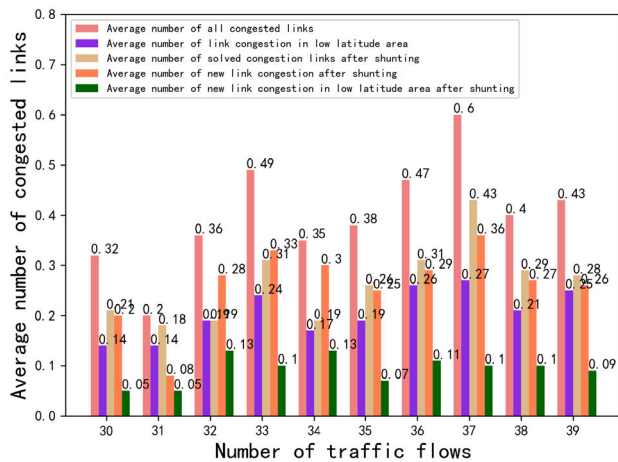


FIGURE 7. Link congestion state before and after shunting (under [30]–[39] traffic flows).

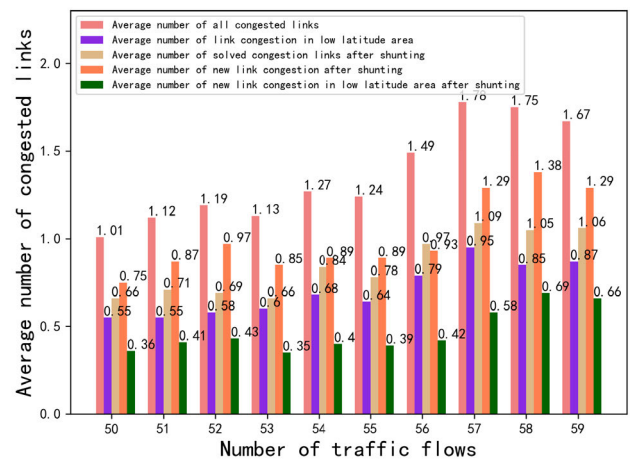


FIGURE 9. Link congestion state before and after shunting (under [50]–[59] traffic flows).

number of congested links in the low latitude area is D. The partial congestion of the low latitude area is solved by using the SSLB. Furthermore, with the flow diversion, the congestion in the high latitude area may also be solved. The congestion link solved after shunting is recorded as E. After shunting, the number of new link congestion is recorded as F after shunting, and the number of link congestion in the low latitude area is G.

When there is congestion in the low latitude, $d > 0$. If SSLB is effective, some congested links should be solved, then $E > 0$. If $E > D$, it means that part of congestion in some high latitude areas can be solved after shunting. SSLB should reduce the number of all link congestion in LEO network. So it should meet the requirements of $F < C$, $G < D$. In this simulation, [20], [59] traffic flows are selected and divided into four groups on average. The simulation results are shown in Figure 6 to Figure 9 below:

When link congestion occurs, the number of link congestion in the low latitude area is almost the same as that in the high latitude area. After using SSLB, not only

the congestion in low latitude areas can be solved, but also part of the congestion in some high latitude areas. After traffic diversion, the number of link congestion in the network decreases as a whole, and the number of link congestion in low latitude areas also decreases.

2) RESULT AND ANALYSIS OF LINK LOAD DISTRIBUTION

a: SIDA

If the traffic is evenly distributed in the network, the number of idle links is small, and the utilization rate of network links is high. In the case of the same traffic flow, the distribution of residual bandwidth is more uniform, and the variance of residual bandwidth is smaller. This experiment uses Dijkstra algorithm as a comparison. The mean and standard deviation of the residual bandwidth of the two algorithms are calculated. If the standard deviation of the remaining bandwidth of the link under SIDA is smaller than Dijkstra algorithm, it shows better load balancing ability.

Shown in Figure 10. The standard deviation of residual link bandwidth under Dijkstra algorithm is always higher

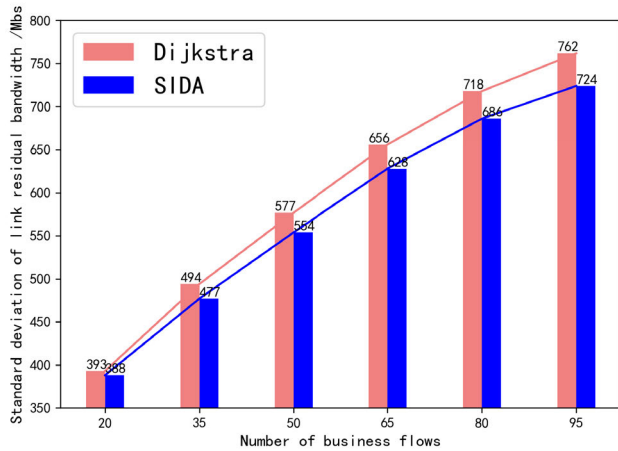


FIGURE 10. Histogram of standard deviation of link remaining bandwidth(Dijkstra algorithm and SIDA).

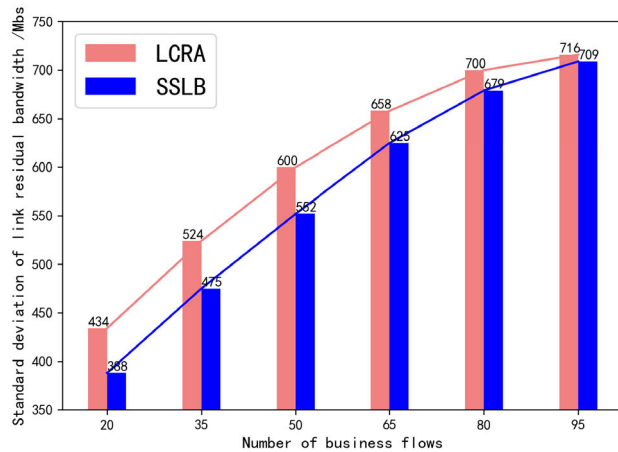


FIGURE 11. Histogram of standard deviation of link remaining bandwidth(SSLB and LCRA).

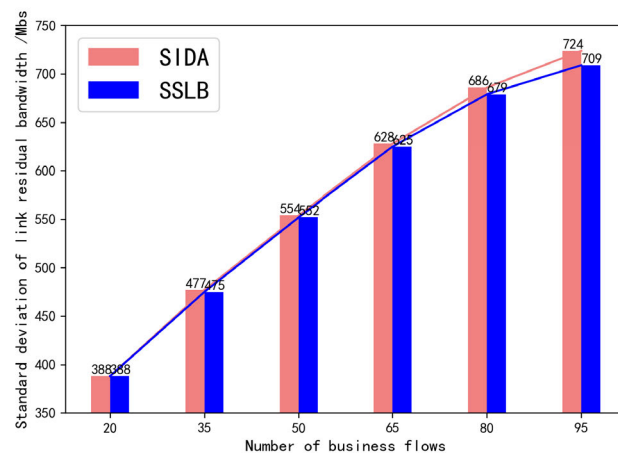


FIGURE 12. Histogram of standard deviation of link remaining bandwidth(SIDA and SSLB).

than SIDA. With the increase in the value of traffic flows, the standard deviation of the remaining bandwidth under SIDA increases relatively more slowly. Thus, SIDA has significantly higher link-utilization.

b: SSLB

As shown in Figure 11 and Figure 12. Under six traffic flows, the standard deviation of LCRA residual link bandwidth is much larger than SSLB when the traffic value is small. And the standard deviation of the remaining link bandwidth of SSLB is almost always greater than SIDA. With the increase of traffic flow, the difference between the two algorithms increases.

VI. CONCLUSION

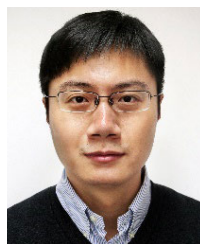
For LEO satellite networks, the SIDA and the SSLB are proposed. The SIDA is optimized based on Dijkstra algorithm, which is used to calculate the route snapshot in the snapshot sequence algorithm. The SIDA changes the way of node traversal and limits the frequency of each node as the last hop. Simulation results show that SIDA can improve the link utilization, reduce the average number of congested links, and the network can undertake more traffic flows.

The SSLB is applied in the scenario of SIDA to solve the sudden link congestion in low latitude area. This strategy which solves link congestion from the perspective of traffic diversion considers the selection of shunted nodes and the calculation of shunted traffic. The SSLB has low computational complexity and low signaling complexity, which is suitable for LEO satellite networks. The simulation results show that the SSLB not only reduces the congestion in low latitude area, but also reduces the congestion in high latitude area, and improves the link utilization.

REFERENCES

- [1] P. Chowdhury, M. Atiquzzaman, and W. Ivancic, "Handover schemes in satellite networks: State-of-the-art and future research directions," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 4, pp. 2–14, 4th Quart., 2006.
- [2] X. Pan, R. Liu, and X. Lv, "Low-complexity compression method for hyperspectral images based on distributed source coding," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 2, pp. 224–227, Mar. 2012.
- [3] V. V. Gounder, R. Prakash, and H. Abu-Amara, "Routing in LEO-based satellite networks," in *Proc. IEEE Emerg. Technol. Symp. Wireless Commun. Syst.*, Apr. 1999, pp. 22.1–22.6.
- [4] Y. S. Kim, Y.-H. Bae, Y. Kim, and C. H. Park, "Traffic load balancing in low Earth orbit satellite networks," in *Proc. 7th Int. Conf. Comput. Commun. Netw.*, Oct. 1998, pp. 191–195.
- [5] L. Franck and G. Maral, "Static and adaptive routing in ISL networks from a constellation perspective," *Int. J. Satell. Commun.*, vol. 20, no. 6, pp. 455–475, Nov. 2002.
- [6] T. Tarik, M. Daisuke, and J. Abbas, "ELB: An explicit load balancing routing protocol for multi-hop NGE0 satellite network," in *Proc. IEEE Globecom*, 2006, pp. 1–5.
- [7] Y. Rao and R.-C. Wang, "Agent-based load balancing routing for LEO satellite networks," *Comput. Netw.*, vol. 54, no. 17, pp. 3187–3195, Dec. 2010.
- [8] Y. Rao, J. Zhu, C.-A. Yuan, Z.-H. Jiang, L.-Y. Fu, X. Shao, and R.-C. Wang, "Agent-based multi-service routing for polar-orbit LEO broadband satellite networks," *Ad Hoc Netw.*, vol. 13, pp. 575–597, Feb. 2014.
- [9] J. Yuan, P. Chen, Q. Liu, and H. Li, "A load-balanced on-demand routing for LEO satellite networks," *J. Netw.*, vol. 9, no. 12, p. 3305, Dec. 2014.
- [10] X. Liu, X. Yan, Z. Jiang, C. Li, and Y. Yang, "A low-complexity routing algorithm based on load balancing for LEO satellite networks," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Sep. 2015, pp. 1–5.
- [11] P. Liu, H. Chen, S. Wei, L. Li, and Z. Zhu, "Hybrid-traffic-detour based load balancing for onboard routing in LEO satellite networks," *China Commun.*, vol. 15, no. 6, pp. 28–41, Jun. 2018.

- [12] J. Huang, W. Liu, Y. Su, and F. Wang, "Load balancing strategy and its lookup-table enhancement in deterministic space delay/disruption tolerant networks," *Adv. Space Res.*, vol. 61, no. 3, pp. 811–822, Feb. 2018.
- [13] T.-H. Chan, B. S. Yeo, and L. Turner, "A localized routing scheme for LEO satellite networks," in *Proc. 21st Int. Commun. Satell. Syst. Conf. Exhib.*, Apr. 2003, p. 2357.



includes network architecture, network virtualization, software defined networking (SDN), information centric networking (ICN), space-terrestrial integrated networks, and network innovation environment.

JIANG LIU received the B.S. degree in electronics engineering from the Beijing Institute of Technology, Beijing, China, in 2005, the M.S. degree in communication and information system from Zhengzhou University, Zhengzhou, China, in 2009, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, in 2012. He is currently an Associate Professor with the Beijing University of Posts and Telecommunications. His current research inter-

ests include network architecture, network virtualization, software defined networking (SDN), information centric networking (ICN), space-terrestrial integrated networks, and network innovation environment.



forwarding, and network virtualization.

TAO HUANG (Member, IEEE) received the B.S. degree in communication engineering from Nankai University, Tianjin, China, in 2002, and the M.S. and Ph.D. degrees in communication and information system from the Beijing University of Posts and Telecommunications, Beijing, China, in 2004 and 2007, respectively. He is currently a Professor with the Beijing University of Posts and Telecommunications. His current research interests include network architecture, routing and



RUIZHI LUO received the bachelor's and master's degrees from the Beijing University of Posts and Telecommunications, in 2017 and 2020, respectively. She has published a patent. Her research interests include low Earth orbit satellite on-board routing communication, and network load balancing strategy research.



CHUNWEI MENG was born in China, in 1999. She is currently pursuing the bachelor's degree majoring in communication engineering with the Beijing University of Posts and Telecommunications.

...