A Distributed Congestion Control Routing Protocol Based on Traffic Classification in LEO Satellite Networks

Shiyue Dai
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
158941006@qq.com

LanLan Rui
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
llrui@bupt.edu.cn

Shiyou Chen
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
chenshiyou@bupt.edu.cn

Xuesong Qiu
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
xsqiu@bupt.edu.cn

***Abstract*—With the explosive growth of communication traffic and the non-uniform distribution of users in low earth orbit (LEO) satellite networks, the problem of link congestion becomes more serious, which result in the emergence of congestion control protocols. Load balance, congestion prediction and faster calculation are the basic requirements of the congestion control routing protocol, and lower overhead and higher users' satisfaction are the further goals. We propose a distributed congestion control routing protocol based on traffic classification in LEO satellite networks (DCCR) which uses a low-overhead distributed scheme to compute the routing and uses a traffic classification strategy to optimize the performance of the distributed scheme in congestion control. We divide the traffic into three types and design routing strategies for each type of traffic according to the occupancy level of the satellite, to improve users' satisfaction while balance the load. In the part of the simulation experiment, we compared the performance of DCCR with some existing congestion control algorithms and verified the advantages of the proposed protocol in reducing delay and balancing load.**

Keywords—Low Earth Orbit Satellite, Satellite Routing, Congestion Control, Traffic Classification

I. INTRODUCTION

In the Integrated Terrestrial-Satellite Network (ITSN)[1], LEO satellites[2] have been widely used due to their lower launch cost[3] and lower ground terminal-satellite propagation delay[4]. With the rapid growth of communication traffic, the trend of non-uniform distribution of users has become more obvious, which causes the link load increases rapidly and tends to develop unevenly[5, 6]. The communication quality cannot be guaranteed in the circumstance of link congestion, so the research of congestion control routing with the requirements of load balance, congestion prediction, faster calculation as well as lower overhead and higher users' satisfaction has appeared[7-9].

Although there are many mature routing protocols, there are many differences between satellite networks and the Internet and mobile communication networks. For example, the resource such as storage, computing and bandwidth is lacked in LEO satellites; satellites move fast and network topology changes dynamically; satellites are far from each other with a high transmission latency; the complex environment in space leads to a high probability of interruption[10]. Therefore, it is impractical to use the same routing strategy in LEO satellites as on the ground[11]. Fortunately, the satellite constellation is periodic and predictable[12], and these two main characteristics can be the basis of designing satellite routing protocol.

In this paper, our contributions are as follows.

a) We designed a distributed scheme to compute the routing, considering the geographical conditions such as polar regions and the seam, and routing based on the satellite location and topology. Our scheme is compatible with the limitation of computing resources in LEO satellites, and avoids the excessive computing overhead of some centralized routing protocols.

b) In order to solve the congestion problem in the high-latitude area caused by the distributed scheme, we designed an optimization protocol based on traffic classification. According to the service's sensitivity of latency and throughput, the traffic is divided into three types. We evaluate the occupancy level of a satellite by the queue length of it and determine the transmission path of various traffic types. Simulation results show that the protocol can effectively reduce the delay and balance the load of networks.

The remaining structure of this paper is as follows. In Section II, we discussed the related work of the congestion control protocols in LEO satellite networks. In Section III, we designed the constellation topology, the distributed routing scheme and the optimized congestion control protocol based on traffic classification. In Section IV, we provide the simulation results and performance analysis, which verified the effectiveness of DCCR in reducing latency and balancing load. Finally, we conclude the paper in Section V.

II. RELATED WORK

In the following part, we discussed the work related to congestion control protocols in LEO satellite networks.

A. Congestion control protocols based on service classification

The Traffic Class Dependent Routing (TCD) proposed in [13] designs different routing tables for different types of

978-3-903176-32-4 © 2021 IFIP

traffic, and each type of traffic is forwarded according to its own routing table, which guarantees the demand of sensitive services. However, in the link cost computation of throughput-sensitive traffic, TCD only considers the link utilization but ignores the impact of link latency, so there may be the phenomenon of sacrificing delay due to the excessive pursuit of bandwidth. The Explicit Load Balancing Technique for NGEO (Non-Geostationary Satellite) Satellite IP Networks (ELB) mentioned in [14] classifies the state of satellites by the occupancy of satellites' forwarding queue. Simulation results show that this protocol can predict congestion and notify neighbors before congestion, which can relieve the load pressure of satellites over hot spot regions. ALBR (Agent-based Load Balancing Routing)[15] fully considers the differences between hot spot regions and non-hot spot regions under non-uniform traffic distribution and drives traffic towards non-hot regions successfully through defines the ISL cost modification factor.

B. *Congestion control protocols based on load balancing*

The paper in [16] proposed a Compact Explicit Multi-path Routing (CEMR), which evaluates the degree of link congestion through transmission delay and queuing delay. After computing the transmission cost, CEMR drives part of the traffic to the sub-optimal path, but sometimes the alternate path is so far that it causes excessive delay.

ADR (the Agent-based Dynamic Routing)[17] uses roaming agents and fixed agents. The roaming agent is responsible for collecting satellite link state information and reporting it to the fixed agent. The fixed agent computes the traffic rate and uses the HALO (Hop-by-hop Adaptive Link-state Optimal) algorithm to update the routing table. ADR has low packet loss rate, high throughput and low latency at high transmission rates. However, the information transmission between agents increases the overhead.

LCRA (A Low-Complexity Routing Algorithm Based on Load Balancing for LEO Satellite Networks)[18] has a more realistic and complete topology structure as well as a more practical value. However, LCRA reroutes all traffic via alternate paths when one node is busy, which still needs to be optimized.

RBCA mentioned in [19] improved on the basis of LCRA with a traffic classification mechanism, which divides the traffic into three types according to the sensitivity of latency, and used ISL through GEO to route some insensitive data packets when congestion occurs. However, the routing method of traffic with high bandwidth occupancy has not been design.

The comparisons of our protocol and the routing protocols mentioned before are shown in TABLE I. In terms of service classification, we comprehensively consider the impact of delay and throughput on services at the same time, dividing services into three types and planning routes respectively. In terms of load balancing, we predict the occupancy level of satellites based on the queue length, which can prevent more packets from being forwarded to busy satellite. In addition, considering the limited computing resources of LEO satellites, we use distributed routing scheme to reduce computational overhead.

TABLE I. COMPARISONS OF CURRENT CONGESTION CONTROL PROTOCOLS

	Topology Structure	**Satellite constellation**	**Traffic distribution model**	**evaluation parameters**	**Main idea**
TCD [13]	LEO	63 satellites 7 orbit planes	LM-HS (landmasses-hotspot traffic flow dynamics model)	Packet loss rate Latency Throughput	Traffic classification
ELB [14]	LEO	72 satellites 8 orbit planes	LM-HS	Packet loss rate Throughput	Traffic classification Satellites' state classification
ALBR [15]	LEO	72 satellites 8 orbit planes	UP_HS (upper hotspot model)	Packet loss rate Latency	Stationary agents and mobile agents
CEMR [16]	LEO	66 satellites 6 orbit planes	-	Packet loss rate Latency	Path encode
ADR [17]	LEO	66 satellites 6 orbit planes	-	Packet loss rate Latency Throughput Link utilization	Roaming agents and fixed agents HALO algorithm
LCRA [18]	LEO	66 satellites 6 orbit planes	-	Packet loss rate Latency Throughput	Distributed computation
RBCA [19]	GEO/LEO	3 GEO satellites and 288 LEO satellites in 12 orbit planes	-	Packet loss rate Latency	Traffic classification Distributed computation
DCCR	LEO	66 satellites 6 orbit planes	-	Latency Queue length	Traffic classification Distributed computation Congestion control

III. DISTRIBUTED CONGESTION CONTROL ROUTING PROTOCOL BASED ON TRAFFIC CLASSIFICATION

A. *Problem Description*

The inter-satellite routing of the LEO satellite networks aims to compute a suitable path for data packets' transmission by a considerate routing algorithm. Compared with the Internet, limited transmission bandwidth, lower quality storage and computing capabilities, faster movement, and more frequent ISL switching are shown in LEO satellite networks.

Therefore, we propose a distributed congestion control routing protocol based on traffic classification in LEO satellite networks (DCCR). In first part of this Section, we design the satellite constellation topology to which our protocol is applicable. In the next part, we provide a low-cost distributed routing method, and the data packets can be forwarded from source satellite to the destination satellite according to this method. However, this method will cause full occupancy of some nodes in high latitude area and hot spots, so we optimize it with a congestion control routing protocol based on traffic classification. This protocol divides traffic of the service according to the demand of latency and throughput, and adjusts the route for each type of traffic on the basis of distributed routing scheme, which can not only reduce the load of busy nodes but also ensure service's latency and throughput requirements.

B. Constellation Topology

LEO satellite constellations are divided into inclined circular orbit constellations and polar orbit constellations[13] .We choose a polar orbit constellation which is similar to the Iridium satellite constellation[20]. The constellation altitude is 780km and the inclination is 86.4°. Each satellite has four ISLs, including: two inter-plane ISLs in one orbital plane (connecting the satellite to neighbor satellites on the north and south sides of the same orbit) and two intra-plane ISLs between two orbital planes (connecting the satellite to the nearest satellite in the neighbor orbits). When the satellite moves above the polar region, the two satellites in the same intra-plane ISL will exchange positions with each other, and the intra-plane ISL will be disconnected. Besides, there is a seam in the constellation where the satellites' movement directions are opposite in different sides, which makes it difficult for the satellites with the unstable relative position to transmit packets[21]. Therefore, the satellites on both sides of the seam retain only one intra-plane ISL as well as two inter-plane ISLs.

C. Distributed routing scheme

DCCR designs routing methods respectively according to three types of position between the source and the destination as shown in TABLE II.

TABLE II. DISTRIBUTED ROUTING METHOD

Position between the Source(S) and the Destination(D)		**Routing methods**
different positions same plane	-	Route the data packet along the orbital plane with the direction of fewer hops.
relative position different planes	Outside polar region	Route horizontally along the intra-plane ISL.
	In polar region	Choose the node with higher latitude as the next hop as long as it is out of the polar region, and then route horizontally.
different positions different planes	In same polar region	Case 1.
	In different polar regions.	Case 2.
	S: outside polar region D: in polar region	Case 3.
	S: in polar region D: outside polar region	Case 4.
	both outside the polar region on the same side of the seam.	Case 5.
	both outside the polar region on the opposite side of the seam.	Case 6.

Case 1: The source and the destination are in the same polar region. The data packet should be transmitted out of the polar region first, then changing the orbital plane to reach the plane of the destination, and finally be routed to the destination along the plane. When the NodeS (source node) and the NodeD (destination node) are on the same side of the seam as shown in Fig. 1, NodeS first sends the data packet to a node in PlaneS (the plane which NodeS is in) at the same side with NodeD outside of the polar region. Then, after being transmitted horizontally to PlaneD (the plane which the NodeD is in), the packet enters the polar region to reach NodeD.

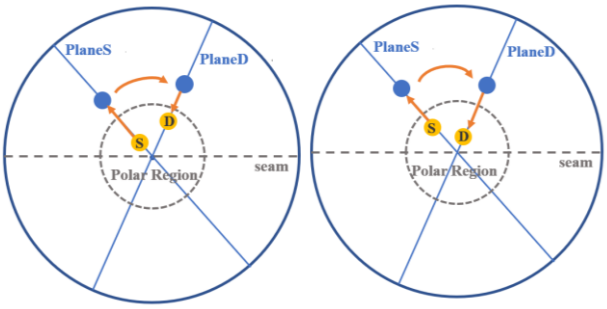

Fig. 1. NodeS and NodeD on the same side of seam.

When NodeS and NodeD are on the opposite sides of the seam, as shown in Fig. 2. In order to find the shortest paths, the latitude relationship between the two nodes needs to be considered. When the latitude of the NodeS is higher than that of the NodeD, the data packet first crosses the pole to reach the other side of the seam, and then is transmitted horizontally to PlaneD; on the contrary, when the latitude of the NodeS is lower than the NodeD, the data packet first leaves the polar region and is routed horizontally to PlaneD, and then crosses the pole vertically.

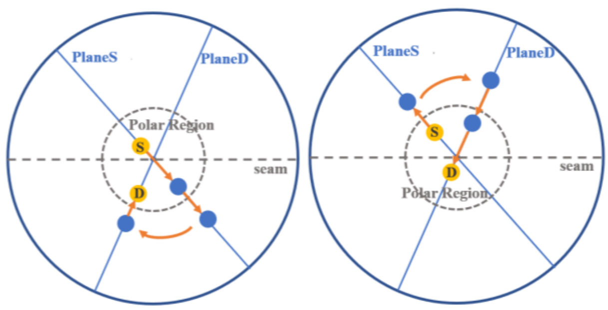

Fig. 2. NodeS and NodeD on the opposite side of seam.

Case 2: The source and the destination are in different polar regions. The data packet departs from the NodeS and leaves the polar region in the direction closer to the destination, and then is routed according to case 3.

Case 3: The source is outside the polar region as the destination is in the polar region. If NodeS' neighbor located outside the polar region, packets are transmitted to the neighbor node vertically; otherwise, packets are transmitted horizontally to PlaneD, and then enter the polar region vertically.

Case 4: The source is in the polar region as the destination is outside the polar region. NodeS chooses the neighbor on the same side of the seam with NodeD as the next hop. After leaving the polar region, the packet is routed according to case 5.

Case 5: The source and the destination are both outside the polar region and on the same side of the seam. Comparing the latitude of NodeS and NodeD, the packet is routed horizontally near the node with higher latitude, and then carried out according to the "different positions in the same plane" to obtain shortest paths.

Case 6: The source and the destination are both outside the polar region and on the opposite side of the seam. Considering the routing needs to pass the polar region, the packet selects the neighbor node closer to NodeD as the next hop until entrances the polar region, and then is transmitted according to case 4.

D. *Optimized Protocol for Congestion Control Based on Traffic Classification*

To some extent, distributed routing scheme we introduced before increases the traffic on high-latitude satellites since they have a shorter intra-plane ISL. In this part, we propose an optimized protocol for congestion control based on traffic classification, taking into account the differences of service, and rerouting some types of traffic when the satellite being on high occupancy level, which aims to balance the load of each node.

1) *Traffic classification strategy*

We evaluate some services' demand of latency and throughput, and divide traffic into three types as shown in TABLE III.

TABLE III. TRAFFIC CLASSIFICATION

Traffic type	Character	Example service
TrafficA	Latency-sensitive traffic. It has the highest priority. The shortest path should be guaranteed to minimize the delay.	1. Voice over IP 2. Interactive video application
TrafficB	Throughout-sensitive traffic. It should be transmitted with abundant bandwidth	1. Video-on-demand 2. Large file distribution
TrafficC	Ordinary traffic. Do best to transmit it on the basis of ensuring TrafficA and TrafficB.	1. Voice message 2. File transfer

Obviously, TrafficA always follows the shortest paths calculated by distributed routing scheme, so it does not need to consider the occupancy level of the node. In order to allocate as much bandwidth as possible to TrafficB and in the meantime do not increase the delay too much, we select a low-latitude path with the same hop count as the path based on distributed routing scheme. This solution can avoid the bandwidth occupation of TrafficA in the high-latitude area. For TrafficC, when the satellite is in the state of Free, it is transmitted via the same path as the latency-sensitive traffic. When the satellite is in the state of Busy, some traffic of TrafficC is allocated to detour the busy node.

The system checks the queue length of each node every 10ms. Once one node's queue length exceeds the Busy Threshold, it is judged as a busy node and added to Busy array. After shielding the busy node in the satellite constellation, the system calculates the alternate path excluding the busy node by Dijkstra's Shortest Path (DSP)[22]. The next step is Calculating the rerouting proportion of TrafficC, use the alternate path to transmit this part of TrafficC, and continue to check every node's queue length. If the node in Busy array whose queue length is below the Busy Threshold, remove it from the Busy array, and transmit all TrafficC in this node by distributed routing method.

2) *Satellites' state computation and traffic allocation*

We divide traffic into 3 types in previous part. TrafficA follows the shortest paths. TrafficB follows a low-latitude path with the same hop. TrafficC route detouring the busy node with a specific proportion. And in this part, we would introduce how to determine the traffic type and detouring proportion according to the satellite's occupancy level. The notions required are shown in TABLE IV.

TABLE IV. THE MEANING OF NOTATIONS IN OUR PAPER

Notions	Meaning of notions
$QLength$	Maximum forwarding queue length. (pkt)
q_t	Forwarding queue length now. (pkt)
$PLength$	Average packet size. (Kb)
$Input$	Input traffic rate. (Kb/s)
$Output$	Output traffic rate. (Kb/s)
ΔT	Time before dropping packets. (s)
$delayISL$	Average ISL link delay. (s)
$pDrop$	Packet drop rate.
BT	Busy Threshold. The ratio of q_t to $QueueLength$ exceeding this value means the node is in Busy state.
Q_{BTA}	Forwarding queue length of the busy node when BTA (Busy Threshold Advertisement) reaching the neighbor satellite. (pkts)
$T_{recovery}$	Time of the node keeping in Free state. (s)
TRR	Threshold rate of new input traffic.
C_{remove}	TrafficC in neighbor node which should be transmitted via alternate path. (pkts)

The congestion control protocol calculates BT based on the time before dropping packets. When one node's input traffic rate is more than its output traffic rate and remain constant for ΔT time, the forwarding queue will up to $QLength$ and start to drop packets. (1) shows the calculation of ΔT.

$$\Delta T = \frac{(QLength - q_t) \times PLength}{\max(0, Input - Output)} \quad (1)$$

The calculation of the packet drop rate $pDrop$ is shown in (2).

$$pDrop = \min\left(1, \frac{delayISL}{\Delta T}\right) \quad (2)$$

After the satellite gets q_t, it needs at least $delayISL$ to send message to the next hop to inform the packet drop will occur after ΔT. When $\Delta T < delayISL$, packet drop will occur before the message reaches the next hop, and the packet drop rate is 1 at this time. When $\Delta T > delayISL$, The probability of packet drop can be described by the ratio of $delayISL$ to ΔT.

BT of the queue is set as (3).

$$BT = (1 - pDrop) \times 100\% \quad (3)$$

The system checks the queue length of each node every 10ms. Once the occupation of queue exceeds BT as (4), the system adds the node to Busy array and sends BTA to each neighbor node.

$$\frac{QLength}{q_t} > BT \quad (4)$$

The neighbor nodes receive BTA after $delayISL$, and now the queue length of the busy satellite is calculated by (5).

$$Q_{BTA} = \min\left(QLength, q_t + delayISL\frac{Input - Output}{PLength}\right) \quad (5)$$

The input traffic of a busy node is divided into direct input traffic ($Input_{direct}$) and indirect input traffic ($Input_{indirect}$). $Input_{direct}$ is the traffic whose destination is the busy node, and $Input_{indirect}$ is the traffic who will pass by the busy node and the destination is another node. In the process of load balancing, direct input traffic cannot be reduced, but some indirect input traffic can detour to other nodes. Assuming that $QLength$ of the Free state is no more than half of the Busy state, and the node will not reach Q_{BTA} until it remains Free state for $T_{recovery}$. The new input indirect flow rate acceptable to the node during this period can be calculated by (6).

$$Input_{indirectNew} = Output - Input_{direct} + \frac{PLength*\left(Q_{BTA} - QLength*\frac{BT}{2}\right)}{T_{recovery}} \quad (6)$$

Set the threshold rate of new input traffic TRR as (7).

$$TRR = min\left(max\left(0, \frac{Input_{indirectNew}}{Input_{indirect}}\right), 1\right) \quad (7)$$

After the node enters the Busy state, $changeC$ which represents the detouring proportion of TrafficC's $Input_{indirect}$ in each neighbor node can be computed by TRR. When $1 - TRR > 0$, we calculate $changeC$ according to (8); when $1 - TRR < 0$, all TrafficC's $Input_{indirect}$ is routed via alternate path which means $changeC = 1$. Therefore, C_{remove} can be described in (9).

$$changeC = (1 - TRR) \times \frac{Input_{indirect}}{Input_{indirectC}} \quad (8)$$

$$C_{remove} = C_{indirect} \times changeC \quad (9)$$

Where $C_{indirect}$ is all indirect traffic stored by the neighbor node, and $Input_{indirectC}$ represents the input traffic rate of TrafficC.

The process of the congestion control optimized mechanism is as follows. Set i as the number of current node, and set $BusyNode$ as the name of the Busy array.

Algorithm 1 Pseudo Code of the Congestion Control Optimized Mechanism

1. **Do 2-13 lines until all packet has been sent over**
2. $BusyNode = \emptyset$
3. **for** i=1; i<66; i++ **do**
4. compute $pDrop_i$ and BT_i
5. **if** $QLength/q_{ti} > BT$
6. $BusyNode \leftarrow BusyNode \cup \{i\}$
7. compute alternate path for TrafficC
8. compute $changeC$
9. **else if** i ∈ $BusyNode$
10. delete i from $BusyNode$
11. **else** continue
12. **end if**
13. **end for**

IV. Performance Evaluation

A. Simulation Environment

We use Matlab to build our constellation topology and transmission model. The simulation parameters are shown in TABLE V.

TABLE V. Simulation Parameters

Parameters	Value
Maximum forwarding queue length (QLength)	200pkts
Average ISL link delay (delayISL)	20ms
Average packet size (PLength)	1KB
Time of the node keeping in Free State ($T_{recovery}$)	200ms
Type of the forwarding queue	FIFO

B. Simulation Results

In this section, we first compare the end-to-end delay of our DCCR and LCRA to vertify the performance of distributed routing scheme. Thwe conduct a comprehensive comparison from the perspective of the delay and the length of node queue.

1) The Performance of distributed routing scheme

We select 6 different source-destination pairs with the same number of hops in the shortest path, and measure their average end-to-end delay. Then gradually increase the source-destination pairs located surrounding them, making the path pass through the same node as much as possible. The simulation result is shown in Fig. 3. The larger number of source-destination pairs, the greater possibility that each node forwarding queue is busy. As the number of source-destination pairs increases, DCCR's mean latency increases slowly, while LCRA rises significantly until tends to be relatively stable. The result proves that these two algorithms both have a certain ability to adjust the load of busy nodes. Meanwhile, both DCCR and LCRA use the distribution routing scheme, but traffic classification is only added to DCCR, so the lower latency is due to the traffic classification mechanism.

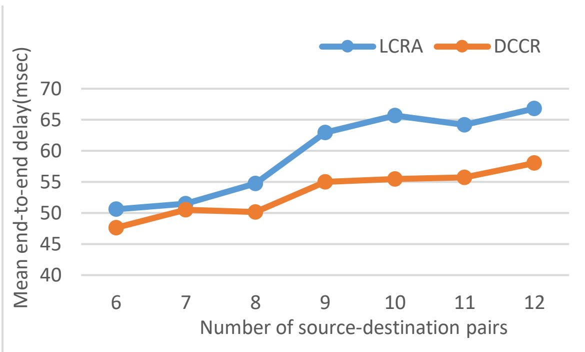

Fig. 3. Delay versus number of source-destination pairs.

2) Comprehensive performance

In this part, we evaluate the comprehensive comparison of our protocol and some other congestion control routing protocols including ELB, DSP, TCD and LCRA in the aspect of average queue length and latency.

Fig. 4 shows the average queue length for different sending rates, examining the load balancing capabilities of each algorithm under high network load. The stronger ability of load balancing, the less likely it is for data packets to stack in busy nodes, so the average queue length gets shorter. It is proved that the average queue length of various algorithms are similar in low sending rates. As the sending rate increases, the queue length of ELB and DCCR increases more smoothly. These two algorithms use the queue length to measure the occupancy level of the node. When the queue length is too large, the traffic will be redistributed to maintain the queue length at a relatively stable level. It shows that ELB and DCCR have a positive effect on balancing load and avoiding node congestion.

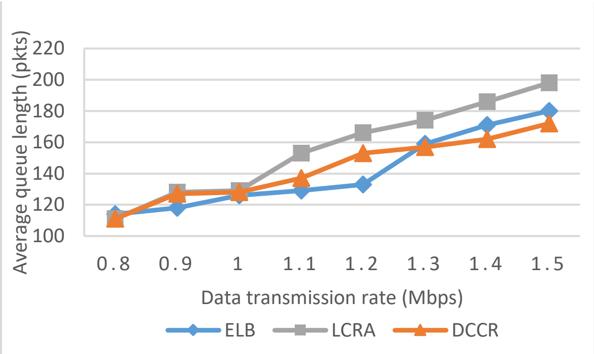

Fig. 4. Queue length in different transmission rate.

Fig. 5 shows the delay of different types of traffic in TCD, ELB and DCCR we proposed. The three algorithms all use traffic classification strategy to distinguish delay-sensitive traffic (TrafficA) and ordinary traffic (TrafficC). The link cost of TrafficB (throughput-sensitive) in TCD is only related to link utilization rate but no relationship to link latency at all. As a result, it causes the maximum extra delay. In fact, in order to ensure the effectiveness of information, end-to-end latency is an important consideration for every type of traffic. ELB regards TrafficB as slightly sensitive to delay without considering the sensitivity of the throughput, so the delay of TrafficB is between the value of TrafficA and TrafficC. Compared with the ELB and TCD, DCCR considers the throughput-sensitive traffic like TCD, but selects a higher bandwidth path with the same number of hops as the distributed routing. Although it brings slightly increasing in distance, hop count guarantees the latency to rising indistinctively. Therefore, the end-to-end delay of three types of traffic all keeps at a stable and low level. Furthermore, TrafficA in DCCR has the shortest delay while TrafficC has a slightly larger delay, which meets the demand of urgent service and ensures user's satisfaction.

We comprehensively compare the end-to-end delay under different numbers of routes through the busy node in Fig. 6. We set a busy node and select several routes passing through this node at the same time, and measure the end-to-end delay of one route when the number of routes increase. The result shows that latency increases with the number of routes, and each algorithm has a certain ability of controlling the congestion. Compared with LCRA, ELB and DCA with traffic classification strategy have better load balancing performance, and the occupancy level of nodes has less impact on the delay, which is because traffic classification strategy makes traffic be routed in different paths, reducing the probability of TrafficB and TrafficC passing busy nodes. When we analysis the time complexity, ELB mainly computes a routing table based on the shortest path algorithm, and then balances the load according to a congestion control strategy. When the number of nodes is large, DCCR with the time complexity of $O(1)$ is greatly simplified than the shortest path algorithm with the time complexity of $O(n^2)$. The low-cost distributed routing scheme of DCCR relies on the characteristics of the constellation topology and computes the forwarding direction according to the position of the satellite, so it has better delay performance than ELB.

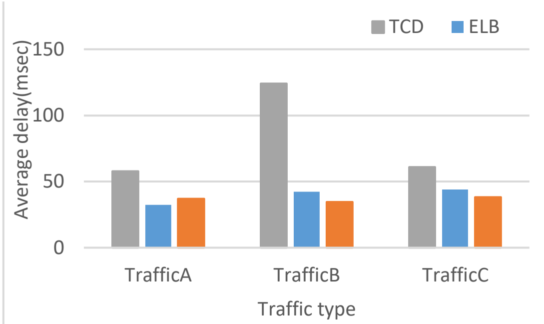

Fig. 5. Delay of different types of traffic.

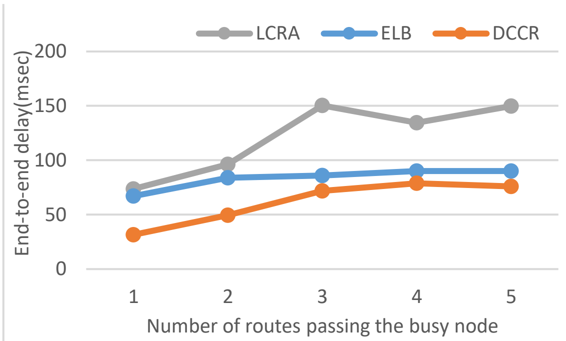

Fig. 6. Delay for different number of routes.

V. Conclusion

In this paper, we propose a distributed congestion control protocol based on traffic classification. The distributed routing scheme we designed decreases the computing overhead, being compatible with the limited resource in LEO satellites. Traffic classification strategy ensures that different types of services meet the most necessary transmission resource. The optimized protocol for congestion control divides satellites' state and allocates the traffic into various paths, achieving the goal of load balance and congestion control. Accordingly, the protocol we proposed not only can better meet the demand of service to enhance users' satisfaction, but also can effectively reduce the queue length of busy nodes and maintain a stable low latency under high network load.

Acknowledgment

This work is supported by National Key R&D Program of China (2020YFB1807802).

References

[1] Z. Yang, H. Li, Q. Wu and J. Wu, "Topology discovery sub-layer for integrated terrestrial-satellite network routing schemes," China Communications. vol. 15, no. 6, pp. 42-57, June 2018.

[2] J. Li, H. Lu, K. Xue and Y. Zhang, "Temporal netgrid model-based dynamic routing in large-scale small satellite networks," IEEE Transactions on Vehicular Technology. vol. 68, no. 6, pp. 6009-6021, June 2019.

[3] Y. Liu and L. Zhu, "A suboptimal routing algorithm for massive LEO satellite networks," 2018 International Symposium on Networks, Computers and Communications (ISNCC). Rome, pp. 1-5, 2018.

[4] Y. Lv, C. Xing, N. Xu, X. Han and F. Wang, "Research of adaptive routing scheme for LEO network," 2019 IEEE 5th International Conference on Computer and Communications (ICCC). Chengdu, China, pp. 987-992, 2019.

[5] C. Dong, X. Xu, A. Liu, F. Cheng, B. Cai and J. Chang, "Research on new methods of LEO satellite networks routing," 2019 6th International Conference on Information Science and Control Engineering (ICISCE). Shanghai, China, pp. 495-499, 2019.

[6] Y. Zhu, L. Qian, L. Ding, F. Yang, C. Zhi and T. Song, "Software defined routing algorithm in LEO satellite networks," 2017 International Conference on Electrical Engineering and Informatics (ICELTICs). Banda Aceh, pp. 257-262, 2017.

[7] X. Li, F. Tang, L. Chen and J. Li, "A state-aware and load-balanced routing model for LEO satellite networks," 2017 IEEE Global Communications Conference (GLOBECOM). Singapore, pp. 1-6, 2017.

[8] H. Wang, G. Wen, N. Liu, J. Zhang and Y. Tao, "A load balanced routing algorithm based on congestion prediction for LEO satellite networks," Cluster Computing. vol. 22, pp. 8025–8033, 2019.

[9] C. Dai, G. Liao, P. T. Mathiopoulos and Q. Chen, "Multi-service routing with guaranteed load balancing for LEO satellite networks," Communications and Networking, pp. 283-298, 2020.

[10] W. Shi, D. Gao, H. Zhou, Q. Xu and C. Foh, "Traffic aware inter-layer contact selection for multi-layer satellite terrestrial network," 2017 IEEE Global Communications Conference (GLOBECOM). Singapore, pp. 1-7, 2017.

[11] H. Han, Y. Zhao, J. Wei and S. Cao, "GHRP: an efficient routing protocol for satellite networks," 11th International Conference on Wireless Communications and Signal Processing (WCSP). Xi'an, China, pp. 1-5, 2019.

[12] Y. Liu and C. Liu, "Distributed dynamic routing algorithm for satellite constellation," 10th International Conference on Communication Software and Networks (ICCSN). Chengdu, China, pp. 300-304, 2018.

[13] Svigelj, M. Mohorcic, G. Kandus, A. Kos, M. Pustisek and J. Bester, "Routing in ISL networks considering empirical IP traffic," IEEE Journal on Selected Areas in Communications. vol. 22, no. 2, pp. 261-272, 2004.

[14] T. Taleb, D. Mashimo, A. Jamalipour, N. Kato and Y. Nemoto, "Explicit load balancing technique for NGEO satellite IP networks with on-board processing capabilities," IEEE/ACM Transactions on Networking. vol. 17, no. 1, pp. 281-293, 2009.

[15] Y. Rao and R. Wang. "Agent-based load balancing routing for LEO satellite networks," Computer Networks. vol 54, pp: 3187-3195, 2010.

[16] J. Bai, X. Lu, Z. Lu and W. Peng, "Compact explicit multi-path routing for LEO satellite networks," Workshop on High Performance Switching and Routing (HPSR). Hong Kong, China, pp. 386-390, 2005.

[17] N. Michael, A. Tang and D. Xu, "Optimal link-state hop-by-hop routing," 21st IEEE International Conference on Network Protocols (ICNP). Goettingen, pp. 1-10, 2013.

[18] X. Liu, X. Yan, Z. Jiang, C. Li and Y. Yang, "A low-complexity routing algorithm based on load balancing for LEO satellite networks," IEEE 82nd Vehicular Technology Conference (VTC2015-Fall). Boston, pp. 1-5, 2015.

[19] J. Ma, X. Qi and C. Chen, "A routing algorithm based on congestion avoiding in satellite networks," Journal of Jilin University. vol. 57, pp.357-362, 2019.

[20] M. Jia, S. Zhu, L. Wang, Q. Guo, H. Wang and Z. Liu, "Routing algorithm with virtual topology toward to huge numbers of LEO mobile satellite network based on SDN," Mobile Netw Appl. vol. 23, pp. 285–300, 2018.

[21] 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.

[22] P. Xie, Z. Zhang and J. Zhang, "Inter-satellite routing algorithm by searching the global neighborhood for dynamic inter-satellite networks," Tenth International Conference on Advanced Computational Intelligence (ICACI). Xiamen, China, pp. 673-678, 2018.