

## A ROUTING ALGORITHM FOR MPLS TRAFFIC ENGINEERING IN LEO SATELLITE CONSTELLATION NETWORK

CHENGSHENG PAN<sup>1,2,3</sup>, HUANHUAN DU<sup>1,3</sup> AND QINGLI LIU<sup>1,3</sup>

<sup>1</sup>University Key Laboratory of Communication and Signal Processing

<sup>2</sup>Key Laboratory of Communication Networks and Information Processing

<sup>3</sup>College of Information Engineering

Dalian University

No. 10, Xuefu Street, Jinzhou Xinqu, Dalian 116622, P. R. China

{ pcs; liuqingli }@dlu.edu.cn

Received September 2012; revised January 2013

**ABSTRACT.** *With regard to the limited link time of the inter-satellite links in LEO satellite constellation network, this paper proposes a routing algorithm which takes the limited link time into account, and supports traffic engineering. The routing algorithm is based on MPLS satellite network. The initial weight of links is defined to be function of residual bandwidth and link time. And during the weight adjustment period, link weight is dynamically changed by its critical degree. The algorithm can improve short communication time and limited total network traffic that caused by the link handover. Simulation results show that the algorithm has a good performance in terms of request rejection rate, average traffic per node and total network traffic.*

**Keywords:** LEO satellite constellation network, Routing algorithm, MPLS, Minimum interference algorithm

1. **Introduction.** Constituted by several low earth orbit (LEO) satellites, the LEO satellite constellation network can make seamless global coverage, small delay and multi-service delivery, which occupies an important position in personal global communication services [1]. In LEO satellite network, routing technology is one of the key technologies. With the continuous growth of network traffic, the traditional IP routing has been increasingly unable to meet our needs. Multi-protocol label switching (MPLS) technology can effectively solve the traffic balance problem that happens in traditional IP routing. MPLS can combine IP routing and Layer 2 switching, and support quality of service (QoS) and traffic engineering (TE). MPLS network can implement traffic engineering simply and efficiently by establishing label switching path (LSP) between the source node and destination node. In MPLS, the ingress routers calculate routing and the backbone routers transmit the packets. The source node and the destination node of MPLS are knowable [2]. With those benefits MPLS satellite network has become a new research trend [3,4].

In MPLS network, it has generally developed two methods to optimize network resources and performances. One is greedy algorithms [5], such as the constrained shortest path algorithm (CSPF) [6], the widest shortest path algorithm (WSP) [7] and the shortest widest path algorithm (SWP) [8]. The other is the minimum interference algorithm [9]. The minimum interference algorithm considers future LSP selection while choosing the current LSP. It dynamically changes the link weight by the definition of link critical degree, which can lead to selection changes on the LSP. With those measures taken, it can balance network traffic and improve the total throughput of the network. Through the two types of algorithms, the latter tends to be ideal on optimizing the network resources [5,9].

There are some methods that can be used in LEO satellite constellation network to resolve the problems of changing network topology. A kind of method is the snapshot topology of time [10]. With this approach, the changing satellite network topology is divided into multiple time slices and it is not considered to change within one time slice. The problem that topology is always changing can be fixed. However, it ignores the differences of link time between the satellites because all links within a snapshot are treated as the same link time. While in fact, the link time of different links between the satellites which are in high-speed movement is distinctly different. In communication, ignoring the limited time of the satellite links may lead to a bad situation which the selected LSP has just been established soon, and it cannot continue to communicate. The links become invalid because of handover. It makes the LSP selection be of less practical significance.

Therefore, we proposed a new routing algorithm with supporting traffic engineering. It defines independent address for the satellites to avoid the network dividing into snapshot topology. The algorithm uses the idea of minimum interference algorithm and can be concluded by two phases. In initial phase, the link weight is defined as function of residual bandwidth and link time. In weight adjustment phase, the link weight dynamically changes by critical degree. The final communication routing path is calculated from the output of the second phrase. Compared with the existing algorithms, our algorithm takes handover into account. Although it makes a few more average hops, it keeps a low bandwidth rejection, and at the same time, it improves traffic per LSP on average and total network traffic. Simulation results demonstrate that the proposed algorithm performs better than WSP, MHA and LIP in traffic [11].

In addition to the residual bandwidth and link time, the satellite network routing also needs to consider the factor of transmission delay between the satellites. Reference [11] defined the link weight as function of residual bandwidth and transmission delay. For our research, the LEO satellite constellation network, the communication distances between the adjacent satellites are almost the same. Taking Iridium for example, the relative distance of intra-plane between the two satellites maintains 2800km to 4500km, and the difference of maximum and minimum transmission delay between the satellites is not more than 6ms, which is about 1/10 of the link transmission delay. So we ignore the impact of delay. The transmission delay can be reflected by average hops.

**2. Network Model.** LEO constellation is constituted by satellites which run in accordance with certain rules and shapes and provide certain coverage, and it runs periodically. The satellite constellation network can be defined as graph  $G(V, E, W)$ , where  $V = \{V_1, V_2, \dots, V_n\}$  is the set of vertices,  $E \in VV$  is the collection of edges between vertices, and  $W = \{w_{ij}\}$  is the link weight from node  $i$  to  $j$ .

In LEO constellation network, we consider near polar constellations, with can be expressed as Figure 1. In our study, we focus on space-based routing. Space-based routing has advantages over terrestrial routing since terrestrial infrastructure can be destroyed because of natural disasters or wars. Moreover, space network and the terrestrial network can be operated independently by different service providers. Leading existing/proposed satellite architectures such as Iridium and Teledesic utilize space-based routing [12].

In Figure 1, the index of orbital planes and the index of satellites in orbit start at 0. There are  $N$  transverse orbital planes and  $M$  lengthwise satellites per orbital plane. The intra-plane links (intra-ISL) are the links constituted by the adjacent satellites in the same orbital plane, and the inter-plane links (inter-ISL) are the links constituted by the neighboring satellites in the right-hand and left-hand orbital planes. We define the satellite index as  $(i, j)$  for  $0 \leq i \leq N - 1$  and  $0 \leq j \leq M - 1$ .

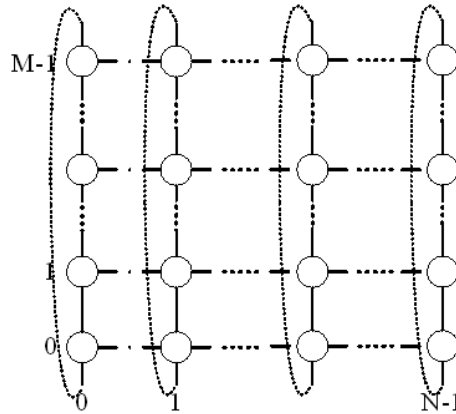


FIGURE 1. The sketch map of near polar constellations

When the algorithm takes into practice, each node can be assigned a network address by the satellite index. In MPLS control plane, there are two main parts. One part is the label distribution protocol and another part is mainly about routing protocol. To apply our algorithm in practice, the label distribution could be either CR-LDP or RSVP-TE. The information that the algorithm needs to calculate LSP could be obtained by routing protocol of the third layer. For example, we could use OSPF-TE as the support of getting residual bandwidth. Moreover, our algorithm needs to know the link time, and the ingress satellite should also store the orbit information of others. References [1,14] drew a solution for distributed routing and reference [3] studied applications for MPLS traffic engineering. The detail of application would go beyond the scope of this paper, and we focus on the routing algorithm here.

**3. Algorithm Description.** Our algorithm consists of two phases. The first phase is to get the initial link weight. In the second phase, we calculate the critical links by all source-destination (SD) node pairs, and dynamically change their initial link weight. After the two phases, we could calculate a final LSP from the critical links.

**3.1. The initial link weight.** The initial link weight is determined by residual bandwidth and current network topology. We define it as  $w_0(u, v)$ , which is the function of residual bandwidth and link time.

$$w_0(u, v) = \begin{cases} T_{link}/t_{link}(u, v) + \lambda_1 B_{left}/b_{left}(u, v), & l(u, v) \in INTER - ISL \\ B_{left}/b_{left}(u, v), & l(u, v) \in INTRA - ISL \end{cases} \quad (1)$$

In (1),  $w_0(u, v)$  is the initial weight from node  $u$  to node  $v$ .  $B_{left}$  is the maximum residual bandwidth of all links in the network, and  $T_{link}$  is the maximum link time of inter-ISL.  $t_{link}(u, v)$  and  $b_{left}(u, v)$  are the link time and residual bandwidth of node  $u$  to  $v$ .  $l(u, v)$  represents the link between node pair  $(u, v)$ ,  $\lambda_1$  is an adjustment factor to adjust the contribution of the residual bandwidth to weight.

**3.2. The critical links and link weight adjustment.**

**3.2.1. The critical links.** We get the critical links by  $K$  shortest paths. The specific method is as follows:

- 1) Calculate the weight of all links according to (1). Initialize the critical links *CriticalCollect* to be empty.
- 2) Find the shortest path of all SD nodes in the network, and add the links to the critical links *CriticalCollect*.

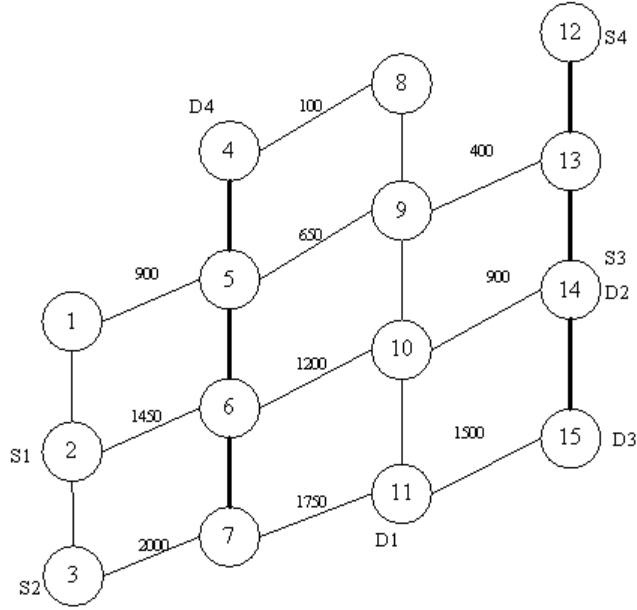


FIGURE 2. Simulation topology

- 3) Remove the links whose weight is maximum, and recalculate the shortest path. Add new links to the critical links *CriticalCollect*.
- 4) Repeat process 3) until the number of shortest paths meets  $K$  or there are no more paths between SD nodes.

According to the four steps above, we can get the critical links *CriticalCollect*. By the collection, we can dynamically adjust the weight of each links. Now we define the critical degree of link  $(u, v)$  as  $\delta(u, v)$ . The dynamic adjustment of link weight is  $w_1(u, v)$ , and

$$w_1(u, v) = \lambda_2 \delta(u, v), \quad (2)$$

where  $\lambda_2$  is the adjustment factor.

3.2.2. *Adjust the link weight.* By the critical links *CriticalCollect* and weight adjustment  $w_1$  calculated in Section 3.2.1, the links of the network are changed as formulation (3) below:

$$w(u, v) = \begin{cases} w_0(u, v) + w_1(u, v), & s'd' \in sd \\ w_0(u, v) - w_1(u, v), & s'd' \notin sd \end{cases}, \quad (3)$$

where  $sd$  is the current SD pair who is requesting LSP, and  $s'd'$  is the SD pair who is being calculated.

After the adjustment, the final link weight  $w(u, v)$  is obtained. The LSP that is established for communication can be calculated from  $w(u, v)$  by shortest path algorithm.

#### 4. Simulation and Evaluation.

TABLE 1. Simulation parameters

<i>Parameter Name</i>	<i>Parameter Value</i>
<i>Satellite altitude</i>	<i>780km</i>
<i>Number of orbits</i>	<i>6</i>
<i>Number of satellites pre plane</i>	<i>11</i>
<i>Inclination</i>	<i>86.4 degree</i>
<i>Inter-plane separation</i>	<i>27 degree</i>

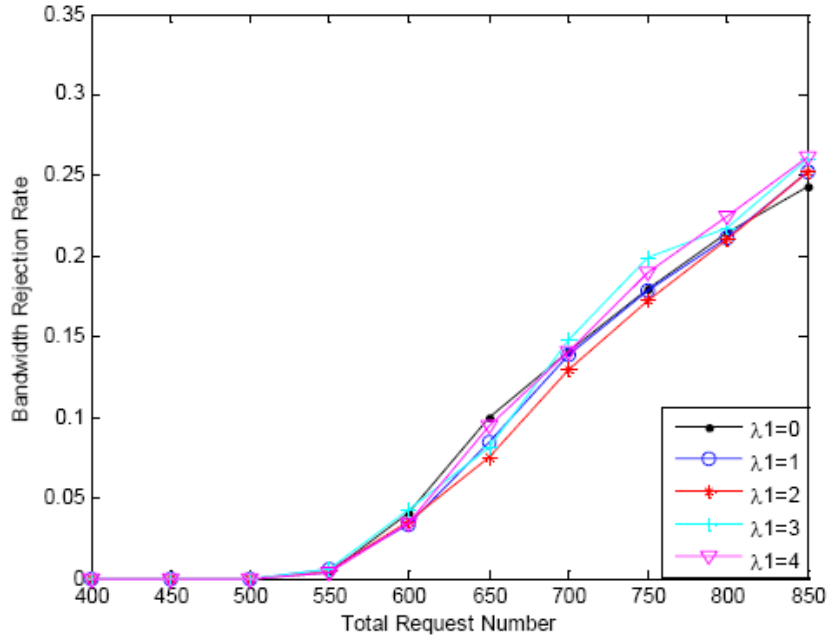


FIGURE 3. Impact of  $\lambda_1$  to bandwidth rejection rate

**4.1. Simulation parameters.** In order to verify our ideas, our simulation network topology is Iridium. The network parameters of Iridium are shown in Table 1. We simulate that users under the satellite call for requests to the destination, and more and more requests gradually add into the network. Moreover, once the LSP has established, the connection will not be destroyed and keeps going on during the whole communication until there is a handover. As the links between the users and the satellites are unique, and the users cannot compute routing, we ignore them and treat the corresponding satellites as the starting nodes of requests. The implication is that there are no exceptions of all nodes and links during the whole communications, the source node well knows the situation of network when calculating LSP.

To illustrate our simulation, we snatch the network topology at a moment that was shown in Figure 2. In Figure 2, S1-S4 are four source nodes, D1-D4 are four destination nodes. The bold lines stand for the links whose bandwidth is 4.8MB while the rest lines are 2.4MB. The values appear on the edge represent the link time, which are in second. Requests are in the form of triple  $(s, d, b_{req})$ , where  $s$  is the source node and  $d$  is the destination node. SD pairs are selected randomly in the SD nodes.  $b_{req}$  is the bandwidth requested, and it is randomly selected between [1k, 4k].

First, we discuss the parameters  $\lambda_1$  in (1) and  $\lambda_2$  in (2). The simulation results are shown in Figures 3 to 6.

In Figure 3, the bandwidth rejection rate is the division of the sum of bandwidth rejected and the sum of bandwidth requested. It is  $\sum b_{rej} / \sum b_{req}$ ,  $b_{rej}$  is the bandwidth rejected.

In Figure 4, the total traffic refers to the sum of communication traffic of all nodes. If we designate  $c_i$  as the multiple of the bandwidth requested and the shortest link time of the link in LSP in the  $i$ -th request, then the total traffic is  $\sum c_i$ . And  $c_i$  can be formulized as  $c_i = b_{req} * \text{Min}(t_{link}(u, v))$ ,  $(u, v) \in LSP$ .

It can be seen that it makes lower total traffic when  $\lambda_1$  equals 0. There are few differences of bandwidth rejection rate when  $\lambda_1$  differs from each other. And it plays a better performance when  $\lambda_1$  is 2.

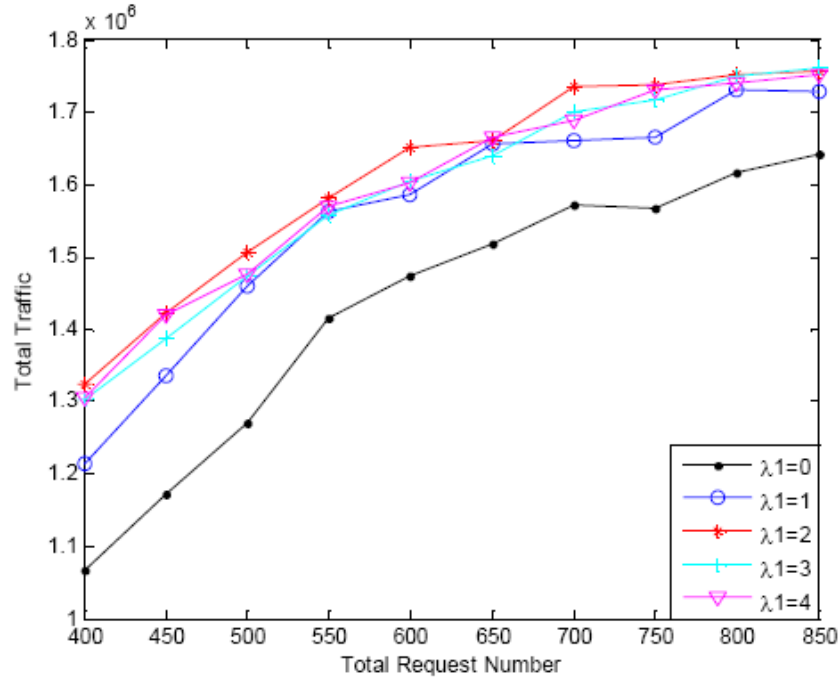


FIGURE 4. Impact of  $\lambda_1$  to total traffic

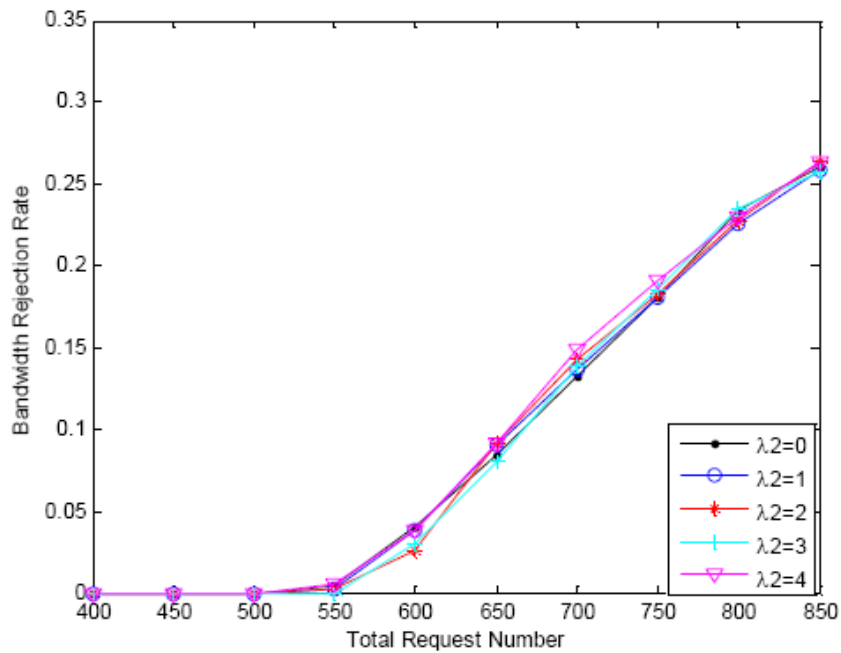


FIGURE 5. Impact of  $\lambda_2$  to bandwidth rejection rate

In Figures 5 and 6,  $\lambda_2$  in different values makes not too much impact to both bandwidth rejection rate and total traffic. Whereas, there are still few differences and  $\lambda_2$  performs better when it is 2.

**4.2. Simulation results.** We compare with three algorithms to testify the performance of our algorithm. Two of them are greedy algorithms, including MHA and WSP, and the rest one is minimum interference algorithm, LIP. It should be noticed that we take both

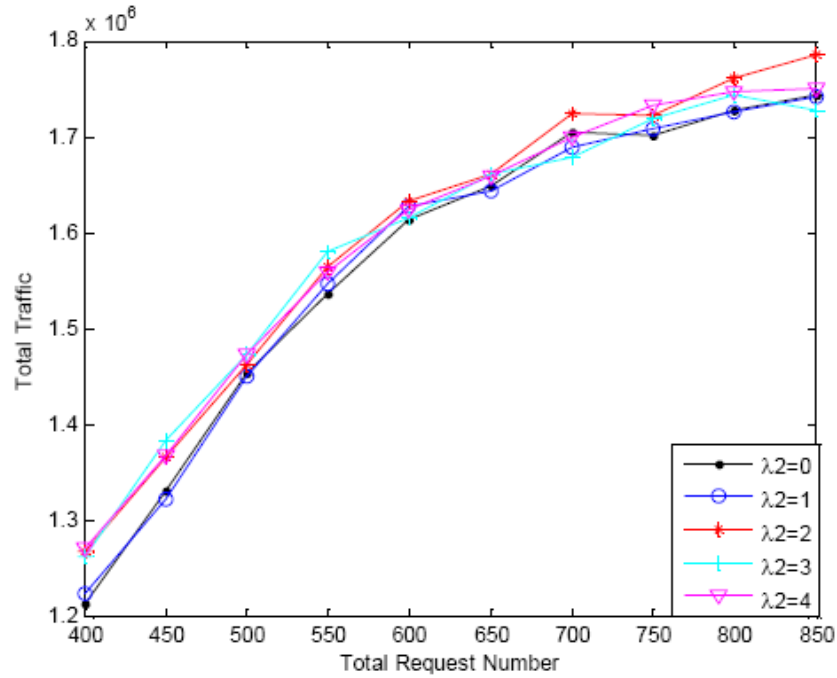


FIGURE 6. Impact of  $\lambda_2$  to total traffic

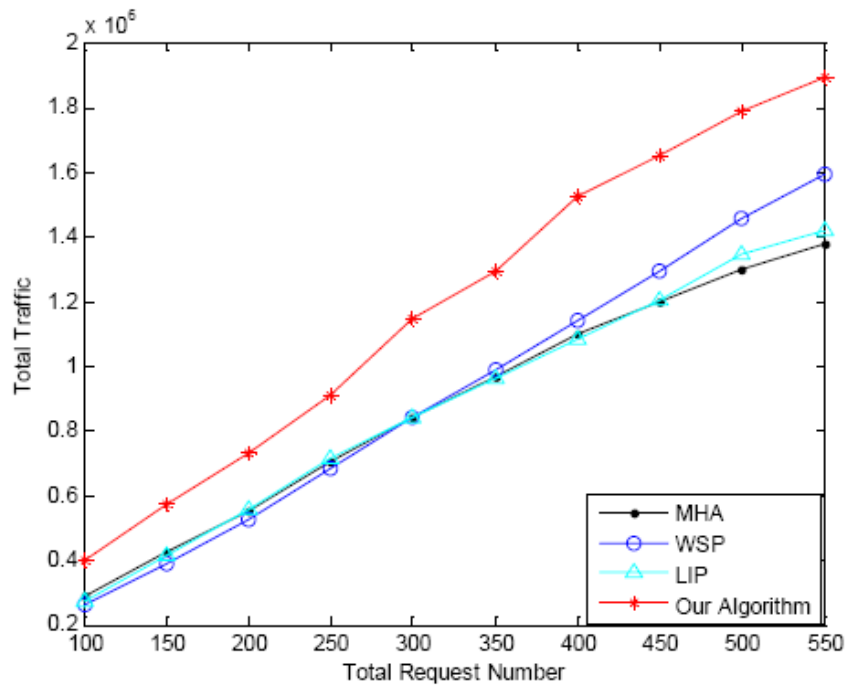


FIGURE 7. Total traffic under the low-bandwidth rejection rate

$\lambda_1$  and  $\lambda_2$  to be 2 in our simulation and run the simulation for 20 times for the results below.

When under the condition of low-bandwidth rejection rate, the performances of four algorithms in total traffic and average traffic are in Figures 7 and 8. In Figure 8, the average traffic means the average traffic of a single LSP. It can be expressed as formulation  $\sum c_i / \sum i$ , which means the division of total traffic and total times requested.

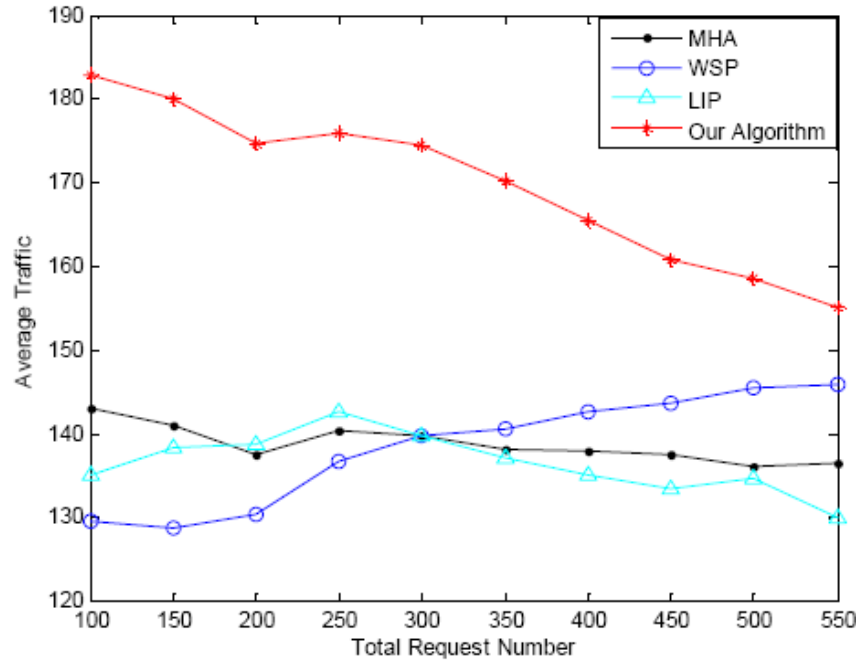


FIGURE 8. Average traffic under the low-bandwidth rejection rate

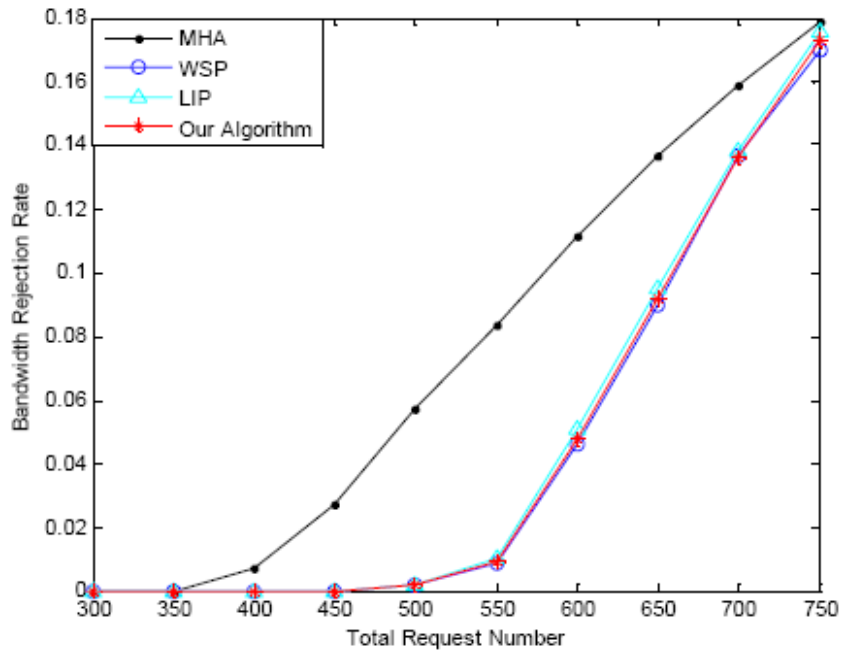


FIGURE 9. Bandwidth rejection rate

Figures 7 and 8 show when the bandwidth rejection rate is low (less than 0.05), the total traffic and average traffic of four algorithms. It is obvious that our algorithm is better than the rest.

Figures 9, 10 and 11 are the simulation results of the bandwidth rejection rate, total traffic and average traffic when the request number increases.



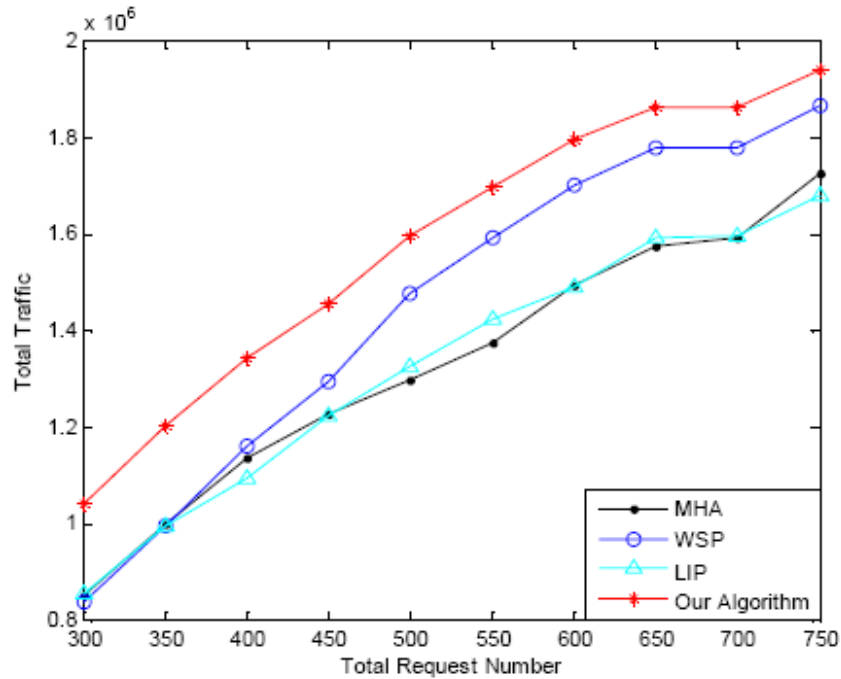


FIGURE 10. Total traffic

It can be seen from Figure 9 that the MHA algorithm bandwidth rejection rate increases fastest. When the requests reach 450, all algorithms make bandwidth rejection. Except MHA, the rest three algorithms are almost the same.

We can conclude from Figures 10 and 11 that, our algorithm is better than the other three algorithms on total traffic and average traffic. Figure 10 shows that the total traffic increases slower as the LSP requests are more. In Figure 11, the average traffic of our algorithm goes down first and trends to smooth later. It is mainly because that the network goes saturated with more and more requests. The bandwidth of the rejection rate starts to rise, and the acceptance of LSP requests is less then. MHA, WSP and LIP do not consider link time, so the average traffic has less effect.

Figure 12 shows the average hops of four algorithms. Our algorithm makes more average hops compared with the other three algorithms, and LIP is higher than the WSP and MHA. MHA and WSP are roughly the same. The main reason is that the minimum interference algorithm which tries to reduce the path interference tends to increase the number of hops. Link time of links closing to the bottom of topology is longer, which further promotes our algorithm trends to more hops.

**5. Conclusion.** Due to the change of LEO satellite constellation network, the snapshot topology of time could resolve the problems. Whereas, it may lead to a bad situation. We proposed a new method with support of traffic engineering in this paper that avoids cutting the topology into time slices by defining a separate index for satellites. In practice, satellites can be designated a network address by the index and compute routing by existing routing component. The algorithm takes handover into account and takes the link time as a metric when routing. The simulation results show that the proposed algorithm has a good performance in rejection rate and traffic, though it is higher in average hops. Another topic for future study is rerouting when there are node fails or link fails. We will study more on rerouting based on the proposed routing algorithm.

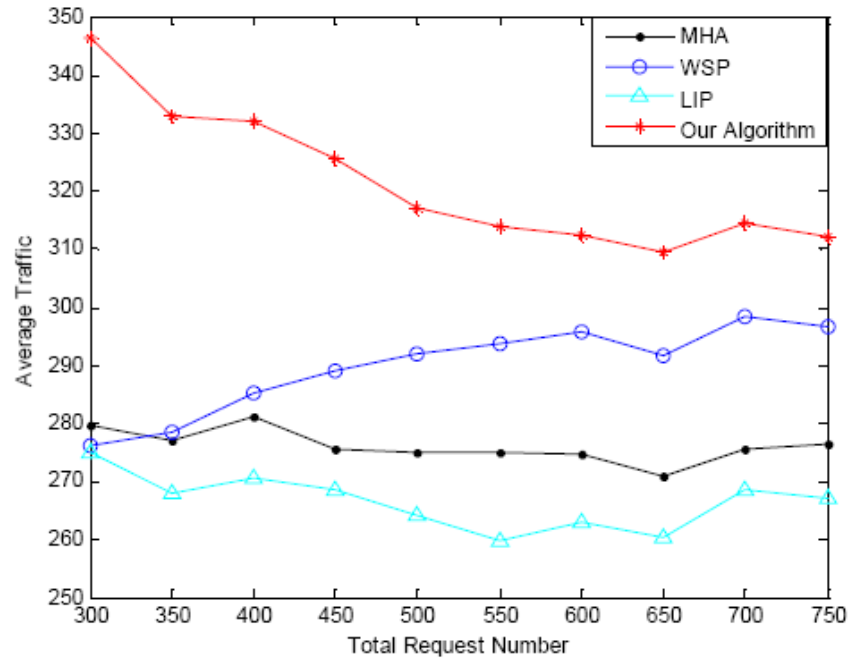


FIGURE 11. Average traffic

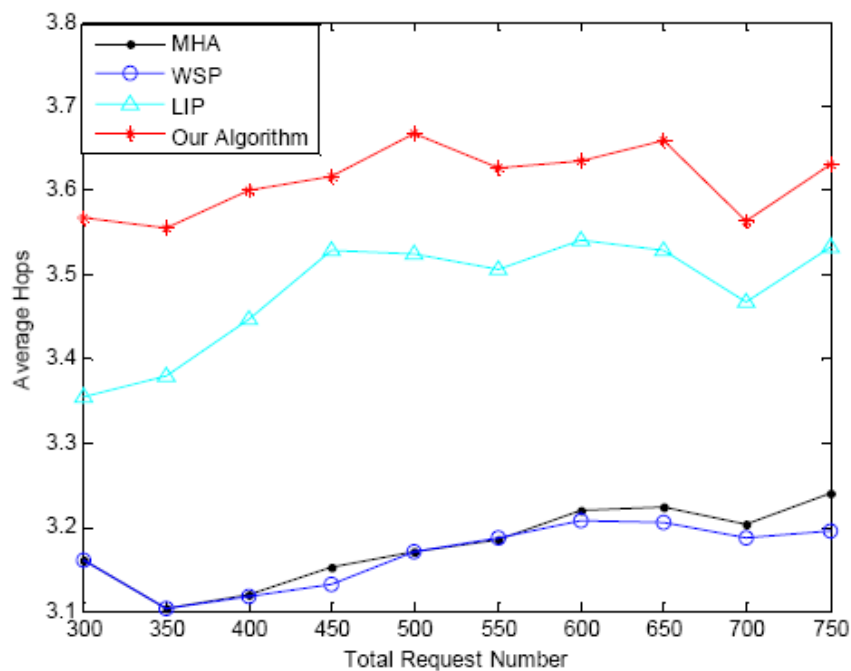


FIGURE 12. Average number of hops

**Acknowledgment.** This work was supported by the Scientific Research Project of Liaoning Province Educational Committee (No. L2011217). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

#### REFERENCES

- [1] T. Zhang, J. Zhan and Z. Liu, Framework of MPLS based mobile satellite communication networks, *Computer Engineering*, vol.32, no.13, pp.130-132, 2006.

- [2] M. Kodialam, T. V. Lakshman and S. Sengupta, Online multicast routing with bandwidth guarantees: A new approach using multicast network flow, *IEEE/ACM Transactions on Networking*, vol.11, no.4, pp.676-686, 2003.
- [3] A. Donner, M. Berioli and M. Werner, MPLS-based satellite constellation networks, *IEEE Journal on Selected Areas in Communications*, vol.22, no.3, pp.438-448, 2004.
- [4] T. Taleb, N. Kato and Y. Nemoto, Recent trends in IP/NGEO satellite communication systems: Transport, routing, and mobility management concerns, *IEEE Wireless Communications*, vol.12, no.5, pp.63-69, 2005.
- [5] Z. Zheng and Y. Cui, Study of minimum interference routing algorithm for MPLS traffic engineering, *Journal of Software*, vol.17, no.4, pp.814-821, 2006.
- [6] E. Crawley, R. Nair, B. Jajagopalan and H. Sandick, A framework for QoS-based routing in the Internet, *RFC2386*, 1998.
- [7] R. Guerin, D. Williams and A. Orda, QoS routing mechanisms and OSPF extensions, *Proc. of Globecom'97*, pp.1903-1908, 1997.
- [8] S. Kamei and T. Kimura, Evaluation of routing algorithms and network topologies for MPLS traffic engineering, *Proc. of IEEE Global Communications Conference*, pp.25-29, 2001.
- [9] M. Kodialam and T. Lakshman, Minimum interference routing with applications to MPLS traffic engineering, *Proc. of the 19th Annual Joint Conf. of the IEEE Computer and Communications Societies*, pp.884-893, 2000.
- [10] M. Werner, A dynamic routing concept for ATM-based satellite personal communication network, *IEEE Journal on Selected Areas in Communications*, vol.15, no.8, pp.1636-1648, 1997.
- [11] F. Xiao, L. Sun, X. Ye and R. Wang, Routing algorithm for MPLS traffic engineering in satellite network, *Journal on Communications*, vol.32, no.5, pp.104-111, 2011.
- [12] I. F. Akyildiz, H. Uzunalioglu and M. D. Bender, Handover management in low earth orbit (LEO) satellite networks, *Mobile Networks and Applications*, vol.4, pp.301-310, 1999.
- [13] S. Karapantazi, E. Papetrou and F. N. Pavlidou, Multiservice on-demand routing in LEO satellite networks, *IEEE Transactions on Wireless Communications*, vol.8, no.1, pp.107-112, 2009.
- [14] E. Papetrou, S. Karapantazi and F. N. Pavlidou, Distributed on-demand routing for LEO satellite systems, *Computer Networks*, vol.51, pp.4356-4376, 2007.
- [15] Z. Zheng and Y. Cui, A least interference path algorithm for MPLS traffic engineering, *Chinese Journal of Computers*, vol.30, no.6, pp.934-944, 2007.