

# A Distributed Algorithm for Self-adaptive Routing in LEO Satellite Network

Hao Cheng<sup>1</sup>, Meilin Liu<sup>2</sup>, Songjie Wei<sup>1(✉)</sup>, and Bilei Zhou<sup>2</sup>

<sup>1</sup> School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei, Nanjing 210094, China

swei@njjust.edu.cn

<sup>2</sup> Shanghai Institute of Satellite Engineering, Shanghai Academy of Spaceflight Technology, 3666 Yuanjiang Rd, Shanghai 201109, China

**Abstract.** LEO satellite networks, represented by the successful Iridium System, are composed of multiple satellite nodes and inter-satellite links (ISL). Numerous routing algorithms have been designed to determine satisfying routes between data flow sources and destinations, within the constraints including delay, congestion control, and throughput and load balancing. This paper proposes a distributed network-state aware self-adaptive routing algorithm based on neighbor ISL status and node workload. Every satellite node is independently responsible for forwarding datagrams in its queue, with information about network status piggy-backed in the transmitted datagrams. Such information helps understand and predict the network workload status on each direction of the outgoing links, and is used for nearly-optimal selection of datagram outbound links to achieve load balancing and multi-path routing. Experiments are conducted on ns-2 simulation platform with a designed LEO walker, to implement and evaluate the effectiveness and efficiency of the proposed algorithm. The results show a significant improvement of more than 50% on the network workload balancing, with a few more hops in the selected multiple routing paths compared with the traditional Dijkstras shortest path algorithm.

**Keywords:** LEO · Satellite network · Routing algorithm · Self-adaptive

## 1 Introduction

In 1965, Instelsat-1 was launched as the first commercial communication satellite to provide instantaneous cross-continent telecommunication service. After that, communication satellites are extensively applied in all aspects of human civilization advance and daily life, in military, economy, culture and society development in the past 50 years. In terms of quantity, communication satellites make up of the largest category of artificial satellites in space. Constrained by orbit, payload capability, communication coverage, running time and system stability, a single satellite cannot cover the whole earth to supply realtime service continuously. With the development of space technology and application, satellites in

different orbits are deployed and organized as constellation. Satellites in a constellation coordinate for larger ground coverage, operate together under shared control, synchronize for more complicated functionality. Amid the modern computer technology, network protocols and control theory, multiple satellites in different orbits can be coordinated for a full coverage of the Earth for a seamless intelligent information service, i.e., ‘a hybrid of space and terrestrial information network’ for data relaying and information sharing. Every satellite node in the network is independent and capable of sending, receiving and forwarding data. Inter-Satellite Link (ISL) for data communication is maintained between neighbor nodes in constellation. Datagram is transmitted between satellites in specific paths called routes [1]. It is of great value to keep improving and optimizing satellite routing algorithms for promoting the development of space communication and optimizing space information resource usage. Efficient inter-satellite routing strategy improves the data transmission rate, QoS and system reliability in satellite internetworking.

Packet routing is the fundamental functionality on the network layer in TCP/IP protocol suite. A routing algorithm determines how a data packet is transmitted along switches in network. Modern Internet based on TCP/IP employs hierarchical routing by choosing different interior and exterior gateway protocols to implement data switching in network. On the other hand, the topology dynamics and the asymmetric links in satellite network devalue a direct application of the traditional TCP/IP protocols in satellite communication scenario [2]. Other researchers refer and extend the sophisticated MANET routing protocols such as OLSR and AODV, and have proposed several multi-path, dynamic and load-balanced routing algorithms for satellite constellation. Satellite network is fundamentally different from MANET. Mesh nodes usually have isotropic communication capability. Data links in satellite network are influenced by the relative movement and angles between peers. Fast-changing relative distances among satellites result in nonnegligible communication delay fluctuation and link instability, which is a big challenge for those mesh routing protocols to run effectively in satellite constellation [3].

The satellite technology research communication and industry have not come up with a widely acceptably and universally applicable routing protocol for satellite internetwork. Existing candidates, either from the mature TCP/IP, the fast-paced MANET or the specific CCSDS [4] series, do not fully take account of the asymmetric satellite network traffic scenarios and the uneven data flow distribution. They have rigorous prerequisites for computation power, storage and transmission capabilities on satellite nodes. Many tend to oversimplify the requirements on satellites for stability and robustness.

This paper proposes a self-adaptive datagram relay and routing decision algorithm for LEO satellite constellation. LEO satellites have significant advantages on less time consumption and lower economic cost in design, manufacturing, launching and networking procedures. LEO is suitable for various communication and surveillance missions. It is a trend for LEO network to conduct routing based on inter-satellite link (ISL) independent of ground-based assistance.

A practical LEO inter-satellite routing strategy must implement automatic decision-making, self-adaptive, network autonomous without relying on stationary network topology, satellite orbit, pre-configured routing path, additional assistance from ground.

Although all satellite node in a constellation can be organized into a network to collaborate on missions, ideally every node should independently compute routing decisions and conduct forwarding operations based on only its own perception of network connection and traffic status. Inspired by the cooperative game theory with incomplete information in economics, referring to the Internet Autonomous System (AS) model, we propose an self-adaptive routing framework in which satellite node are autonomous, distributed and independent in sensing network status and making datagram forwarding decisions. We use the network flow strength state as an example to demonstrate how the proposed approach may help improving network-wide workload balancing and flow distribution. The advantages of the proposed routing algorithm can be summarized as follows:

- Distributed routing system. Routing decision is dispersedly made on every node in satellite network. There is no centralized routing coordinator or global routing information. Node routing function is distributional and parallel, avoiding active routing discovery, cooperation and maintenance between nodes.
- Autonomous satellite node. Every satellite node is responsible for datagram forwarding queued locally, calculating routing path independently, and sensing network and link status continuously.
- Self-adaptive routing strategy. Self-adaptive routing is implemented for global convergence through sensing network status and choosing various network performance indicator according to different application scenarios and service requirements.
- Scalable network structure. Proposed network model is scalable when a single node is added, disconnected, overwhelmed or abused.

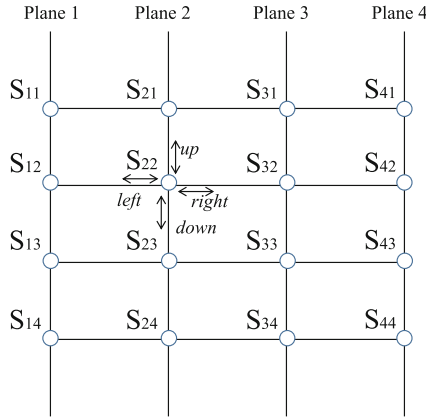
## 2 Self-adaptive Inter-satellite Routing

Satellite constellation is a set of satellites according to specific organizational rules. Design of constellation determines the topology of satellite network. There are two kinds of LEO satellite network structures: polarized and inclined orbit constellations. They differentiate in orbit inclination angle and radian distance distribution in equatorial plane. A satellite network system is naturally divided into two hierarchies: satellite nodes and planes, similarly to hosts and ASes in Internet [5].

### 2.1 Satellite Node and Network Model

We first describe the satellite network model which includes node model, link model, and network flow model. An ISL between neighbor satellites in the same

plane(orbit) is an intra-plane link, while one connecting cross-plane(cross-orbit) satellites is an inter-plane link. Every satellite maintains inter-satellite links with neighbors in local and adjacent orbits. A single satellite is modeled to have 4 ISLs, illustrated in Fig. 1. For example, satellite  $S_{22}$  has links with  $S_{21}$ ,  $S_{23}$  as *up.link* and *down.link* inside the same orbit plane.  $S_{22}$  also has  $S_{12}$  and  $S_{32}$  to its left and right neighbors in adjacent orbits, respectively as *left.link* and *right.link*. The relative inter-satellite distance and angle is constant inside the same orbit, so that intra-plane ISLs can be built and maintained continuously. The relative distance, speed and angle of satellites are time varying between difference orbits. Cross-plane ISLs need special procedure on link state detection and maintenance. In particular, for polarized satellite constellation, there is a seam when two satellite planes run in opposite directions, such as in Fig. 1 between *orbit1* and *orbit4* when the topology spreads on sphere surface [6]. The cost of building and maintaining trans-seam links is very expensive, and will complicate the 2D structure in Fig. 1 into a 3D structure. Previous research indicates that trans-seam links only have trivial impact on network performance when doing satellite networking and communication [7].



**Fig. 1.** Network topology with nodes and links in satellite constellation

Every satellite node keeps tracking the traffic workload state of its each ISL. Link state as traffic statistics includes inbound/outbound traffic, transmission delay, connection time and stability, etc. Different Network QoS optimization goals can be achieved when balancing routing strategy and network traffic by choosing the corresponding nodes and link performance. The link traffic workload is selected as a performance measure in the following discussion of this paper as an example, and to optimize routing for data traffic sensing and load-balancing between satellite nodes. Every satellite node  $S_i$  records inbound and outbound traffic on link  $j$  as  $L_{i,j}^{in}$  and  $L_{i,j}^{out}$ .

Communication satellite nodes are responsible for data transmission service by routing and forwarding datagrams between data flow sources(senders) and

destinations(receivers). Flow can be initiated and consumed by various end nodes such as spacecrafts, gateways, ground fixed and mobile devices. Every data traffic flow  $F_k$  enters and exits the network via border satellite nodes in the network. This paper only focuses on satellite network routing, and thus separates networking nodes from traffic end nodes. Without losing the generality and faithfulness, we simplify data traffic as each flow  $k$  initiated from the entrance node  $F_k^{src}$ , and sunk at the exit node  $F_k^{dst}$ .

## 2.2 Design Assumptions

Before presenting the proposed link-state sensing self-adaptive routing algorithm on independent satellite nodes, we first discuss and argue a few assumptions about the satellite nodes, inter-satellite links, network structure, and the traffic datagram. While keeping in a realistic and practical scope of satellite networking construction and maintenance, these assumptions simplify the algorithm design complexity and thus makes the presentation clearer with the points.

We first assume that every satellite node has some capability of link-state sensing, data computation and storage, and constellation awareness. Link-state sensing means a satellite is aware of its links to neighbor nodes both intra and inter orbit planes. A satellite also has enough computation and storage resource for routing. As a member of a constellation, each satellite has a global view of the constellation orbit planes and satellite members, including numbers, orbits and movement parameters.

Although the link data transmission rate can be asymmetrical, every ISL can transmit data bidirectional. ISL connection lasts longer than the data transmission delay on it. A satellite node or ISL may fail. A malfunctioning ISL cannot transmit data between the satellite nodes, and a malfunctioning satellite node losses all of ISL connections it has with neighbors.

A data flow in satellite network is transmitted as a sequence of datagrams. Every datagram has a fixed size and a predefined structure. Its content organization is mutually agreed between the data sender and receiver according to a communication protocol. Forwarding nodes in network can repackage datagram by adding envelop head and updating datagram piggyback info. Datagram specifies its targeted receiver node and the receiver's orbit. The transmission process of datagram is similar with Internet UDP datagram transmission which is not connection-oriented, with no guarantee on datagram delivery, integrity, timeliness, and order.

## 2.3 Sensing and Exchanging Network Congestion

Every satellite node needs sense its ISL status, add extra information in passing-by datagram to share node workload status, infer load-balance of neighbors in network based on aforementioned information. All of those information are integrated as an guidance when choosing optimal next hop ISL to forward datagram. As described in last session, datagram piggybacks node traffic statistics. By extending the piggybacked information to cover other measures like queuing

delay, link latency, buffer availability, etc., it is possible to apply the same routing strategy on satellite network to achieve different QoS optimization objectives.

A satellite node  $S_i$  maintain possible ISLs to neighboring nodes. For each link, the node tracks two statistic records namely  $L_{i,j}^{in}$  and  $L_{i,j}^{out}$  which count inbound and outbound traffic, respectively. Here  $j \in \{left, right, up, down\}$ . The maintained datagram load on the corresponding link is linearly increased as below when time passes:

$$L_{i,j}^{\{in,out\}} = L_{i,j}^{\{in,out\}} \cdot aging\_ratio + c \cdot N \quad (1)$$

where  $N$  is the number of datagram;  $c$  is a non-negative growing rate predefined depending on the type, size and upper-layer protocol of traffic datagram. Since  $L$  accumulates gradually, we apply an aging factor onto  $L$  when time elapses.  $aging\_ratio$  values between 0.0 and 1.0 as a float. It is positively correlated with the size of network and transmission delay between nodes.

A datagram  $P$  piggybacks information about the workload of all passed-by nodes on its route. This information is inserted by the sender node, updated and consumed by the forwarding nodes, and removed by the receiver node. We use  $Load_p^{plane}$  to denote the statistics and the estimate of the datagram load on nodes of current orbit plane. If satellite  $S_i$  is the current node processing this datagram  $P$ , and it chooses ISL  $j$  as an outbound link, then  $Load_p^{plane}$  is calculated as

$$\begin{cases} Load_{S_i} & \text{if } S_i \text{ is a sender} \\ Load_{S_i} + hop_{aging} \cdot Load_p^{plane} & \text{if } S_i \text{ is forwarding to intra-plane ISL} \\ Load_{S_i} & \text{if } S_i \text{ is a forwarding to inter-plane ISL} \end{cases} \quad (2)$$

$hop_{aging}$  is the decay rate as hop count increases when datagram travels along orbit. If there are  $M$  nodes on the same orbit plane without loop, a properly selected decay rate should achieve  $(hop_{aging})^{\frac{M}{2}} \Rightarrow 0$ .

If the other side of the selected ISL  $j$  is the satellite node  $S_k$ , then once the datagram arrives at  $S_k$ , the inbound traffic statistic should be updated as

$$L_{k,j}^{in} = L_{k,j}^{in} \cdot aging\_ratio + c \cdot Load_p^{plane} \quad (3)$$

Here we interpret  $Load_p^{plane}$  as the accumulated traffic load on all the passing-by intra-plane nodes right before the datagram is leaving node  $S_i$  to the next same-plane node. This is also how much traffic the next-hop node may expect maximally from the upstream along this link. When the next hop is on a different orbit plane,  $Load_p^{plane}$  represents an estimate of the traffic load on the current orbit plane.

## 2.4 Packet Forwarding on Satellite Node

When a datagram arrives at satellite  $S_i$ , this node as a network switch starts to compute independently for a forwarding operation. Figure 2 explains the procedure and the decision strategy. If the current node is the dedicated datagram

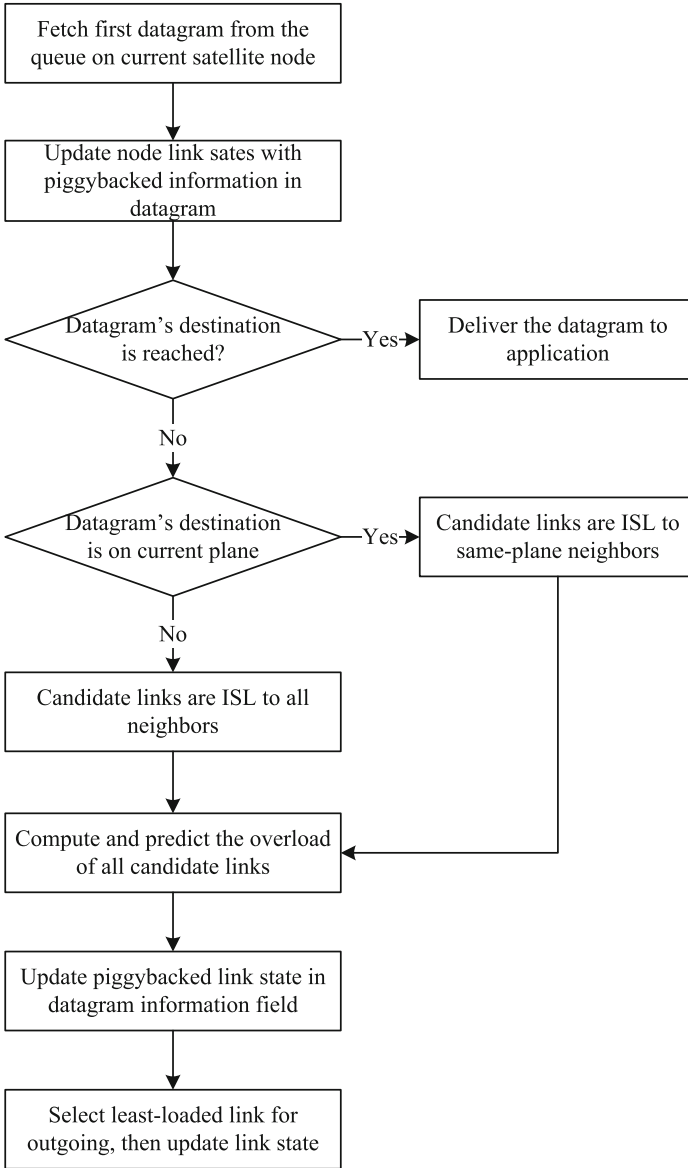
receiver, then it delivers the datagram to the above-layer application and finishes. If the destination is on the same orbit plane as  $S_i$ , it picks an intra-plane ISL as an outbound link. For datagram heading to a destination on a different plane, the next hop could be either of the four neighbor nodes connected with  $S_i$ . If there is more than one candidate appropriate as outbound link for the datagram,  $S_i$  chooses by comparing the sum of  $L^{in} + L^{out}$  value as the link workload. ISL with the minimum cumulative workload wins to carry on the datagram.

## 2.5 Avoiding Loops in Routing

With the layout and organization of satellite nodes in a constellation, ISLs within and between orbit planes may form various routing loops. Any effective routing algorithm should avoid an intermediate node to receive any datagram more than once, i.e., a loopback in routing. Figure 3 shows all the possible inter-satellite links in a 120/10/1 LEO constellation. Close loops are formed in every orbit plane and across all the planes. Since we are working on forwarding datagram in a distributed routing system without centralized coordinator or monitor node, no satellite itself can either detect or avoid loop in routing paths alone. Special regulation and treatment is obligatory to avoid routing loops when conducting the proposed self-adaptive distributed routing algorithm in LEO satellite network.

Setting and checking Time-To-Live (TTL) value is a light-weighted but effective mechanism in avoiding infinite forwarding loop. In each datagram as supplement information, send node initializes two positive TTL values namely  $TTL_{overall}$  and  $TTL_{plane}$ . The former is the maximum number of satellite hops the datagram may traverse before it either reaching the destination or getting dropped. The initial value is determined based on the network size, the node resource richness, and the application service type of the satellite constellation. The latter defines the most number of intra-plane forwarding that a datagram may experience before it proceed to the next orbit plane. Every forwarding node should deduct these two TTLs by 1 before sending the datagram to an outbound link.  $TTL_{plane}$  is restored to the initial value when a datagram is forwarded across orbit plane. During the forwarding procedure in Fig. 2, if both inter- and intra-plane links are available, the forwarding satellite will choose intra-plane link with a priority proportional to the remaining value of  $TTL_{plane}$ . With a zero value of  $TTL_{plane}$ , unless there is no inter-plane satellite link available,  $S_i$  is forced to forward this datagram to a neighbor satellite on an adjacent orbit.

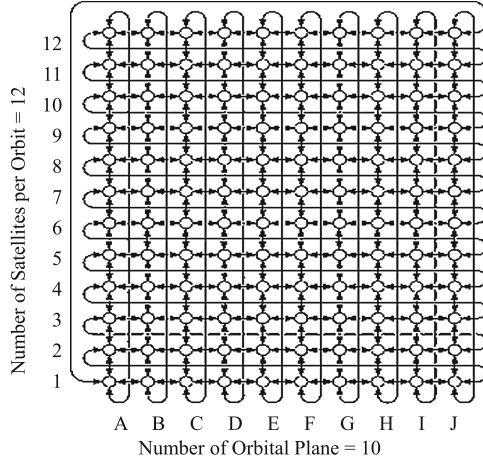
As a special case of loops in routing, another unexpected scenario is when a datagram is forwarded back-and-forth between two adjacent nodes. In Fig. 1, it is possible that after computing the forwarding link priority based on the procedure in Fig. 2, satellite  $S_{22}$  chooses the downlink to  $S_{23}$  as local routing optimal, which happens to repeat the procedure and determines the uplink to  $S_{22}$  is the best choice locally. Such locally back-and-forth shifting will of course be terminated once the TTL value is exhausted, or when the workload on  $S_{22}$  or  $S_{23}$  changes after some iterations, but still it brings in extra routing delay, unnecessary resource consumption, and meaningless TTL deduction. To deal with this case, we insert one more complement parameter  $D_p^{plane} \in \{0, 1, -1\}$  in



**Fig. 2.** Datagram forwarding procedure on satellite node

a datagram. This parameter flags the direction in constellation that this datagram is traveling. A value of 1(or  $-1$ ) indicates the datagram is traveling in the up(or down) direction along the current orbit. On the first hop in current plane,  $D_p^{plane} = 0$  means the intra-plane direction is not determined yet. The satellite may freely forward the datagram to either direction and set the  $D_p^{plane} = 0$  value.





**Fig. 3.** Logical connections in a constellation with inter-satellite links [6]

This value is reset to 0 when cross-orbit forwarding happens. Also only the next orbit plane closer to the targeted orbit is selected for inter-plane forwarding.

### 3 Experiments and Results

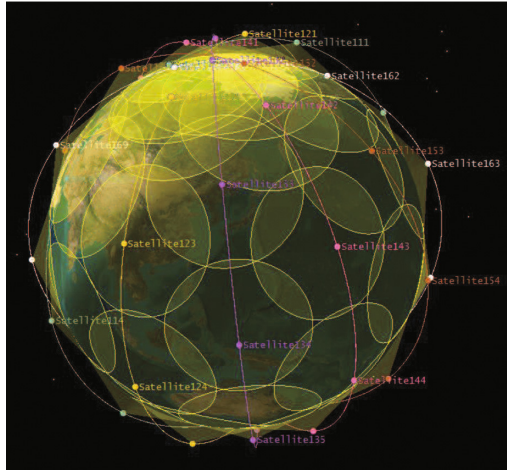
To validate the effectiveness and reliability of the proposed routing strategy, we first use the Satellite Toolkit software (STK) to design a polarized orbit constellation similar the Iridium system. STK exports the orbit, connection status and continuous time parameters into ns-2 simulator. We extend the routing decision module in ns-2 to implement the proposed link-state sensing, information piggyback in datagram, and routing strategy. Two different application scenarios are configured for simulation experiments.

#### 3.1 Simulation Configuration

A walker star constellation is designed using STK which consists of 6 circular orbits of height 680 km and inclination angle  $84^\circ$ , 54 satellites, to covers the whole Earth surface. A 3D demonstration of the constructed constellation is shown in Fig. 4. The  $TTL_{plane}$  and  $TTL_{overall}$  are set as 8 and 32, respectively.  $aging\_ratio$  and  $hop_{aging}$  are 0.98 and 0.5 each.

#### 3.2 Network Traffic Scenario

To evaluate the effectiveness and stability of proposed adaptive routing algorithm, multiple network traffic flows are configured with source and destination nodes distributed across the whole network. The simulation runs for 100s with 1000 randomly-generated network traffic flows. The traffic sender/receiver nodes



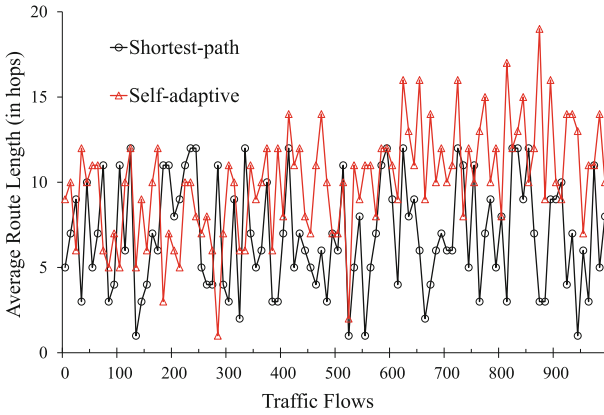
**Fig. 4.** 3D overview of the simulated satellite walker

are choosing randomly in 54 satellite nodes to make the total traffic flows follow lognormal distribution [8]. This means a few nodes in network initiate and sink most datagrams, and most others nodes only involve by relaying datagrams. Flow sending rate is defined as the number of datagrams sent from traffic source node per second, and it follows normal distribution with a mean of 100. Every data flow lasts 10–20 s.

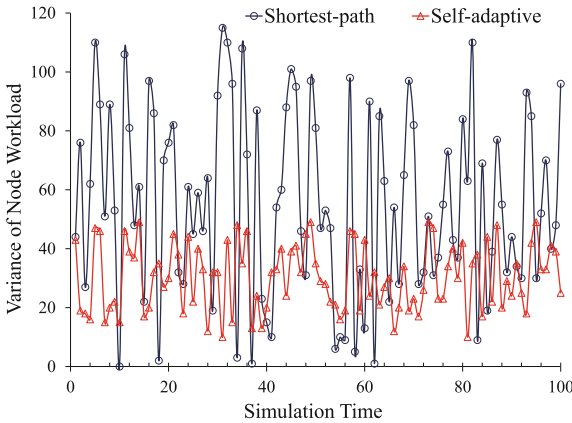
### 3.3 Results and Discussion

Two performance indicators are measured in the experiments. First, the length of inter-satellite routing path is calculated as the hop count in satellite network. Routing hops indicate that the routing overhead of a datagram during its path from source to destination. Fewer hops implies less transmission delay and thus possibly less time delay between the two ends which is composed of queue and routing delay, propagation and transmission delay. The proposed adaptive routing algorithm in this paper utilizes link-state awareness to dynamically distribute traffic flows. Datagrams from the same flow may take multiple paths besides the shortest path. So the hop count increases as the extra routing cost in system, i.e., extra number of nodes passed by. Second, the distribution of traffic datagrams in network is measured as the number of datagrams queued, processed and forwarded on different node at the same time. It is quantified as the variance of this number. Only nodes with inbound traffic in the past second are counted.

In order to benchmark the performance of the proposed routing algorithm, we also test the above two measures with the shortest-path routing simulated. Figures 5 and 6 presents the simulation results and comparisons. Flows using the short-path routing traverse 7 relay nodes on average. While with the proposed self-adaptive routing, Fig. 5 shows similar hop counts at the beginning,



**Fig. 5.** Length distribution of inter-satellite routing paths



**Fig. 6.** Workload distribution of inter-satellite flow traffic

but dynamically self-adjust during the simulation when more flows and traffic join in. With more information about link-state is collected and exchanged gradually, satellite nodes in the proposed routing tend to choose outbound link to the less overwhelmed neighbors and planes. This increases the average flow hop count by 2–3 to 7–10 hops. In a realistic LEO satellite network, ISL transmission delay is in the magnitude of tens of milliseconds, but the node queuing and processing delay may be as large as hundreds of milliseconds or more, due to less powerful computation and onboard storage capability. So the increased number of hops do not necessarily lead to a growing end-to-end (source-to-destination) flow transmission delay, as long as idle and faster satellite hops are preferred.

Figure 6 shows how the traffic workload is distributed and balanced among multiple satellites in network. The shortest-path based routing, when handling non-uniformly distributed traffic flows which are typical in practice, is more likely

to cause congestion in some pivotal satellites nodes and links, especially those inter-plane links [9]. In this case, workloads on satellites are dramatically differentiate without being balanced, with the variance measure fluctuating between 0 and 120. Using the proposed self-adaptive routing, workload in number of datagrams processed on each satellite is balanced with the variance more stable in 0–50, nearly 50% improvement in load balancing.

Measures of load balancing from other prospects such as network flow delay, congestion spread and traffic loss ratio are also simulated, with similar results observed supporting the superiority of the proposed algorithm over shortest-path routing. We skip the result illustrations due to page limit.

## 4 Conclusion

LEO communication satellite systems, such as Iridium, Globalstar and Teledesic, provide global connection and seamless coverage. They represent the future direction of satellite communication technology and application. A single LEO satellite has limited communication service capability because of its insufficient orbit height. Multiple LEO satellites can form constellation to break such limit. Networking and routing is one of the critical problems to be solved before LEO satellites can coordinate to provide reliable, economical and efficient communication service.

We propose a distributed self-adaptive routing strategy for LEO satellite in network. Each satellite node by tracking and estimating the traffic workload in neighbor nodes and orbit planes, independently prioritize nodes and planes with more resources and capabilities. By transmitting flows on multiple routes, the proposed algorithm can gradually self-adjust routing on each satellite to achieve network-wide node workload balancing. Experiments with ns-2 simulation demonstrate the effectiveness of the new routing algorithm, and provide a framework to be extended for optimizing network transmission performance measured in various aspects and expectations.

**Acknowledgments.** This material is based upon work supported by the CASC Innovation Fund No. F2016020013, and the CERNET Next-Generation Internet Innovation Project under contract No. NGII20160601. Opinions and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

## References

1. Lu, Y., Zhao, Y., Sun, F., et al.: Routing techniques on satellite networks. *J. Softw.* **5**, 1085–1100 (2014)
2. Papapetrou, E., Pavlidou, F.: Distributed load-aware routing in LEO satellite networks. In: *IEEE Globecom*. IEEE Press (2008)
3. Li, J., Li, D., Wang, G.: New dynamic routing algorithm based on MANET technology in the LEO/MEO satellite network. *J. Commun.* **26**(5), 50–56 (2005)

4. Consultative Committee for Space Data Systems (CCSDS): Space Internetworking Services Area Publications <https://public.ccsds.org/Publications/SIS.aspx>
5. Liu, G., Li, H.: Satellite Communication Network Technology. Posts & Telecommunications Press, Beijing (2015)
6. Suzuki, R., Yasuda, Y.: Study on ISL network structure in LEO satellite communication systems. *Acta Astronaut.* **61**(7–8), 648–658 (2007)
7. Gavish, B., Kalvenes, J.: Impact of intersatellite communication links on LEOS performance. *Telecommun. Syst.* **8**(2), 159–190 (1997)
8. Camps, F., Thibault, E., Harasse, S.: Statistical distribution of traffic sources in network simulation tools. In: World Congress on Engineering and Computer Science (2008)
9. Li, C., Liu, C., Jiang, Z., et al.: A novel routing strategy based on fuzzy theory for N GEO satellite networks. In: Vehicular Technology Conference (2015)