An Intent-Based Routing Scheme in Satellite IoT

Guanghui Zhang[1], Kang Liu[1], Xue Chen[2,3(✉)], Xu Liu[1], Zhen Pan[1], and Tianyang Jiang[2]

[1] Power Dispatching Control Center of Guizhou Power Grid Co., Ltd., Guiyang, China
[2] University of Electronic Science and Technology of China, Chengdu, China
chenxue61@163.com
[3] Tianfu Collaborative Innovation Center, University of Electronic Science and Technology of China, Chengdu, China

Abstract. As the satellite IoT integrates the satellite network and the IoT, a large number of IoT services are transmitted through satellite links, which is easy to cause the congestion of satellite links during packet routing. By incorporating the concept of intent, this paper proposes an intent-based routing scheme in satellite IoT to solve the congestion problem of satellite link. The routing scheme consists of two parts: the packet preprocessing and shortest path search. In the stage of packet preprocessing, we construct an intent-based system model and design the *packet preprocessing subalgorithm* based on intent. It is used to analyze the user demand, data priority and network congestion state through the intent. In the stage of shortest path search, we design the *shortest path search subalgorithm* based on the interstellar link and user intent. When packet is routed between satellites, the interstellar links and logical locations between the source and destination satellites jointly determine the next hop of satellite. The simulation results prove that the proposed routing scheme has good performance in average end-to-end delay, packet loss rate and network throughput, which can greatly reduce the congestion of satellite links.

Keywords: Satellite IoT · Intent · Routing · Congestion

1 Introduction

In recent years, the IoT has become an indispensable part of people's daily life [1]. However, the IoT is a kind of terrestrial network. It is greatly affected by terrain conditions. And difficult to cover the remote and untraversed areas, such as the mountains and seas. Especially in the 5G network or 6G in the future, people expect to be able to get whatever service they want regardless of their location (whether in the city center or remote rural areas) [2]. Fortunately, the satellite communications can solve the any time and any place coverage problem to some degree [3, 4]. Thus, the concept of satellite IoT came into being, which is integrated by the satellite network and the IoT [5, 6].

Figure 1 shows the traditional satellite IoT [7]. Satellites are configured on the ground when they fly over a site. If the configurations need to be changed or cancelled, it can only

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2022
Published by Springer Nature Switzerland AG 2022. All Rights Reserved
F. Fang and F. Shu (Eds.): GameNets 2022, LNICST 457, pp. 157–171, 2022.
https://doi.org/10.1007/978-3-031-23141-4_12

be done manually. However, these manual operations are not only inefficient, but also easily cause network congestion during packet routing. And in the traditional satellite IoT, the IoT serves must be transmitted through satellite links. Due to the variety of IoT services, the data volume is large and complex. This easily causes satellite link congestion during packet routing. Therefore, real-time detection of satellite link status is required during packet routing. In this way, the best data packet transmission path can be found to avoid satellite link congestion.

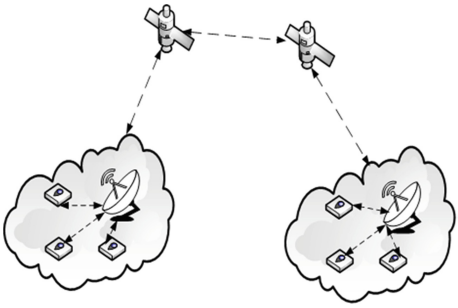

Fig. 1. The traditional satellite IoT.

In this paper, aiming at the congestion problem in satellite IoT, we propose the Intent-Based Routing algorithm (IBR) using the Intent Based Network (IBN) technology. The IBR algorithm consists of two parts: the packet preprocessing and shortest path search. The main contributions of this paper are as follows.

- To preprocess data, an intent-based satellite IoT system model (ISI) is proposed. And we design the packet preprocessing subalgorithm based on intent in IBR. When the packet is received by satellites, the current network state information is detected in real time through the ISI model, and the link state of the packet transmission is analyzed. When a link is congested, IBR performs rollback operations based on the priority of the resolved service type, to reduce the packet loss rate of important packets and improve the network throughput.
- In IBR, the shortest path search subalgorithm based on the interstellar link is designed. After the data packet is preprocessed, a route with minimum transmission delay is found based on the interstellar link model. Then, the logical location of the source/destination satellites and the interplanetary link together determine the next hop of the packet route. It achieves a smaller transmission distance between the source and destination location of the packet and reduces the end-to-end delay of the packet.

2 Related Work

In recent years, with the application of various advanced communication technologies in many fields such as data center network [8], wireless access network [9] and optical

network [10], satellite IoT has been greatly developed. For example, in [11], the authors proposed a software defined satellite network. It solved the problem of dynamic change of satellite network topology and limited on-board processing capacity; In [12], the authors proposed a OpenSAN satellite network, which separated the data plane (satellite infrastructure) from the control plane (geosynchronous orbit group). It provided a good idea for the design of multi-layer space network; In [10], the authors discussed the payload of satellite and solved the problems existing in the current construction of space information system.

However, the above studies focused on the space network without considering the idea of the integrated automation of heaven and earth. In order to further improve the problem of network automation, the concept of IBN had been introduced [13] in satellite IoT. In 2015, the ONF organization proposed the concept of "network intent" [14]. And in 2017, Gartner formally defined the IBN [15]. In IBN, intelligent software determines how to translate intents into infrastructure-specific policies that enable the network to operate in the desired manner. However, the traditional satellite IoT requires administrators to enter specific commands to configure manually. In the intent-based network, the administrator can directly input the state to be performed automatically by the network, rather than by the administrator entering program commands step by step. Thus, the network automation of satellite IoT is expected to reduce satellite link congestion by using IBN technology.

In addition, a good routing algorithm plays a decisive role in alleviating link congestion. Nowadays, some existing routing algorithms of satellite IoT have developed into a connection oriented network idea. Its specific idea is that the network control center uniformly calculates the satellite routing. Thus, the satellite routing table can be formulated according to the network link, and the data packets are transmitted according to the routing table. However, the satellite is moving at a high speed, and the network topology changes at any time. Thus, the pre-calculated routing path may not be optimal as satellite nodes move. In order to solve this problem, many scholars have conducted in-depth research on routing algorithm. For example, in [16], the authors proposed the location-based routing (LBR) algorithm to reduce the routing overhead. Although the LBR algorithm generated the minimum propagation delay path, it ignored the existence of reverse rotation slot; In [17], a low complexity routing algorithm based on load balancing was proposed. It obtained the satellite congestion state through mutual communication between satellites. However, it increased the processing burden of satellites and did not consider the priority of users; In [18], a software defined routing algorithm was proposed. And the optimal path could be obtained by collecting the satellite congestion state in real time. However, the QoS and traffic classification were not well guaranteed; In [19], the authors proposed another software defined routing algorithm (SDRA). It could obtain the optimal routing path by centralized routing strategy. However, the SDRA algorithm was based on the fixed network topology. It putted forward higher requirements for the storage capacity of satellite.

Thus, we need to design a more efficient satellite routing algorithm, which can realize automatic configuration through IBN and solve the problem of satellite link congestion at the same time.

3 Model Building

In this section, we describe the ISI system model and interstellar link model, which are used to guide the design of IBR algorithm.

3.1 ISI System Model

In the stage of packet preprocessing in IBR algorithm, the ISI system model is first built based on IBN technology. The ISI model is shown in Fig. 2. According to the function of Satellite IoT, we divide the system model into three planes: the application layer, the control layer, and the data forwarding layer.

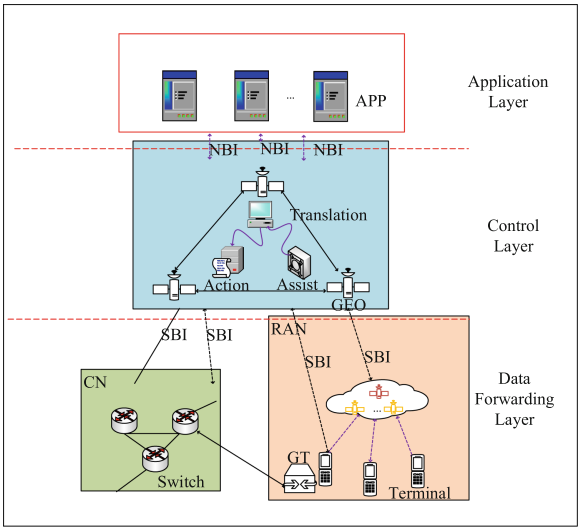

Fig. 2. The ISI system model.

In ISI model, it not only cares about the upper-layer services and applications, but also pays attention to how the underlying network is set up and run. This greatly increases the inflexibility and controllability of the network. The purpose of ISI model is to convert user intents (or needs) into actual strategies. That is, the intent drives the planning and design of the network, thereby saving network resources. Thus, the intent controller is the key of ISI model. It realizes the real-time detection of the network status. The intent controller is deployed in control layer. It allows all types of applications to express their needs and constraints in their own terms. Then it converts the application requirements to the underlying network requirements. Thus, when intent controller performs a task, it can understand the needs of the application and provide the required services through the management of the underlying network resources.

(1) **Application layer**

The application layer of ISI consists of several applications, such as routing, bandwidth allocation. These applications can submit user intents to the intent controller via the northbound interface (NBI). The NBI is between in the application layer and control layer. By using NBI, we can program the network device, and receive feedback from the control layer or the abstract model of underlying network. Therefore, the main function of application layer is to provide the requirements so that users can be closer to the control network.

(2) **Control layer**

The control layer of ISI model is in charge of collecting the network status information. It is mainly composed of intent controllers. Due to the large coverage of geosynchronous orbit satellite (GEO) [20], three GEOs can cover the entire earth and collect entire network status information. Thus, we employ the intent controller on the GEO. After the intent controller receives the user intents from the application layer, it translates these intents into implementable policies. The intent controller consists of three main modules: translation, action, and assist, as shown in Fig. 3. After the intent is captured, intent controller forms a closed loop to ensure that the intent is not disturbed by the sudden situation.

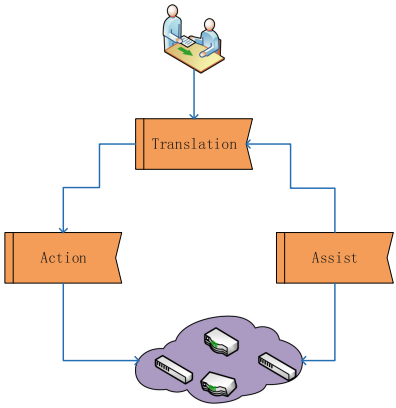

Fig. 3. The intent controller.

(3) **Data layer**

The data layer separates the control decision from the forwarding hardware and forwards the data under the control of the intent controllers. In data layer, as the medium/low orbit satellite has a lower latency and cost compared to the GEO, we use the medium/low orbit satellites as forwarding planes. However, the medium/low orbit satellites in data layer only have the capability of data forwarding. The data is forwarded to the underlying devices via the interstellar links in medium/low orbit satellites. At the same time, when a user wants to communicate with other users or devices with a long distance, the

user can upload the data from the low orbit satellite, and then forward the data using the medium/low orbit satellite links.

3.2 Interstellar Link Model

The interstellar link model is shown in Fig. 4. It consists of N orbital planes. Each orbital plane includes K satellites. Each satellite has a logical address S_{ij}, which denotes its routing address. Here, the i represents the i-th orbital plane and i is not more than N (that is, $0 < i \leq N$). The j represents the j-th satellite in the i-th orbital plane and j is not more than K (that is, $0 < j \leq K$). There exist two kinds of interstellar links, including intra-plane link L_{intra}, and inter-plane link L_{inter}. In Fig. 4, each satellite theoretically involves four interstellar links: two L_{intra} links and two L_{inter} links. However, as the satellite of polar region operates a high speed, the antenna system cannot track the satellite's position in real time. This leads that inter-plane links need to be disconnected and reconnected in polar region. Therefore, the satellites of polar region have two L_{intra} links. In addition, there exists a counter-rotating seam in satellite network (The red dotted line in Fig. 4). It is between in two orbital planes that are moving in opposite directions. The inter-plane between the two planes immediately adjacent to the counter-rotating seam cannot be used for link acquisition. Thus, there are only three satellite links in these two planes, including two L_{intra} links and one L_{inter} link.

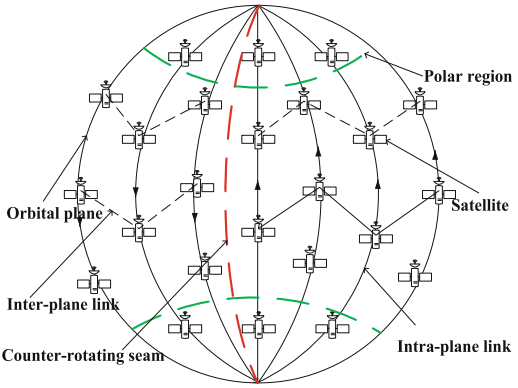

Fig. 4. Interstellar link model. (Color figure online)

From the Fig. 4, we can get that the length of L_{intra} is fixed. It can be calculated according to Eq. (1).

$$L_{intra} = \sqrt{2}(R+h)\sqrt{1 - COS(\frac{360}{K})} \tag{1}$$

where R is the radius of the earth, and h is the height of the orbital plane from the earth.

The inter-plane link L_{inter} can be calculated by Eq. (2).

$$L_{inter} = \sqrt{2}(R+h)\sqrt{1 - \cos\left(\frac{360}{2 \times N}\right)} \times \cos(D) \tag{2}$$

where D is the latitude of the inter-plane. And the main symbols adopted in this work are illustrated in Table 1 for reference.

Table 1. Summary of key notations

Notation	Description
N	Number of orbital planes
K	Number of satellites in each orbital plane
S_{ij}	The i-th orbital plane and j-th satellite's logical address
L_{intra}	The length of the intra-plane link in interstellar
L_{inter}	The length of the inter-plane link in interstellar
R	The radius of the earth
h	The height of the orbital plane from the earth
D	The latitude of the inter-plane
C_k	The capacity of the k-th satellite link
L_{avg}	The link average load
T_h	The link average load threshold
L_s	The latitude values of the source satellite
L_d	The latitude values of the destination satellite

4 Routing Scheme Design

Due to the large number of services in satellite IoT, when the services are transmitted through the satellite links, it easily causes the problem of satellite link congestion. Thus, the network topology needs to be checked for congestion before routing. Therefore, the IBR algorithm is designed based on this idea. It can detect various status information of the network in real time and view the link congestion status when routing. The IBR algorithm is divided into two stages: the packet preprocessing and shortest path search. In the stage of the packet preprocessing, the ISI model first detects the network status, and obtains the user's type and requirements by parsing intent. Then it determines the actual operations. In the stage of shortest path search, a path with the shortest transmission distance is determined according to the interstellar link and logical location between the source satellite and destination satellite.

Because link resources and capacity are limited, it is necessary to evaluate the link average load at control layer before finding the shortest path. Suppose the capacity of a satellite link is C_k. Thus, we get that the capacity array of n satellite links is $C = \{C_1, C_2 \ldots C_{n-1}, C_n\}$. The link average load L_{avg} is written as Eq. (3).

$$L_{avg} = \frac{\sum_{1}^{n} C_k}{n} \tag{3}$$

We define a link average load threshold T_h, as shown the Eq. (4):

$$T_h = \min\{C_k\}, k \in [1, n] \tag{4}$$

L_{avg} and T_h are two parameters of IBR algorithm. When the data packet comes, the IBR algorithm calculates L_{avg}, and compares L_{avg} with T_h to perform different routing strategies.

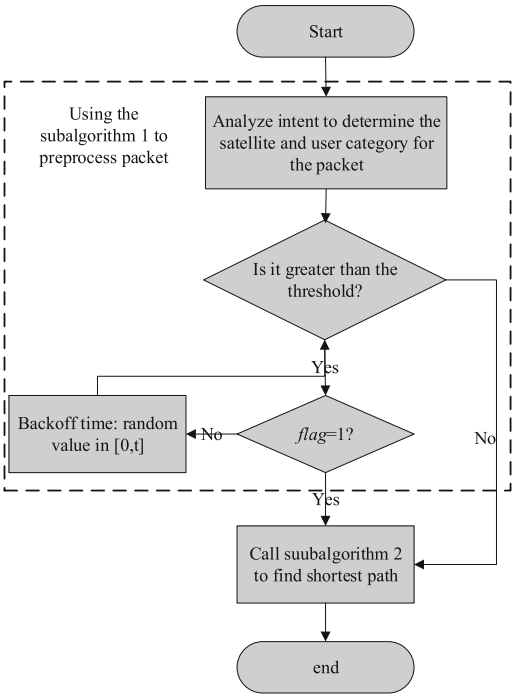

Fig. 5. The flowchart of IBR algorithm.

Figure 5 shows the flow chart for IBR algorithm. To guarantee user's QoS, it preprocesses the upcoming data packets by calling the *subalgorithm* 1. When packet is accessed, the intent control layer calculates the link average load L_{avg} and packet type. In this paper, we divide the data packets into two types using value $flag = \{1, 0\}$. $Flag = 1$ represents high priority data packet and $flag = 0$ means low priority data packet. Each data packet carries an ID, which is the identity for determining the priority of packet. Then we compare the link average load L_{avg} with the threshold value T_h. When L_{avg} is less than T_h, the two types of data packets directly perform operation of shortest routing search. When L_{avg} is greater than or equal to T_h, the high priority data packet performs the operation of shortest routing search, and the low priority data packet performs the back-off. The back-off time is a random value in $[0, t]$. After waiting the back-off time, the control layer decides whether the operation of shortest routing search is performed by calling the *subalgorithm* 2. *Subalgorithm* 2 is to determine the moving direction and next hop of the data packet, and finally reach the destination address.

Subalgorithm 1 describes the operation of packet preprocessing in detail. It includes four main steps: satellite logical area division, intent analysis, network monitoring and route formulation. First, we logically divide the satellite's regional location to determine the control area of each satellite. When a data packet is transmitted through the satellite IoT, the corresponding intent of submission request is determined. Secondly, it analyzes the intent to obtain the information {*flag*, T_h} related to the data packet. Thus, we can determine the specific content and priority of the data packet, as well as the link congestion threshold. Then the ISI model detects the congestion status of the network and obtains the link average load L_{avg}. When $L_{avg} \geq T_h$, it indicates that satellite link congestion has occurred. In this case, data packets may not be transmitted successfully. And it will significantly increase the congestion and packet loss rate. Therefore, in *subalgorithm* 1, we transmit high-priority data packets directly. For the low-priority data packets, we perform the random time backoff. It means to wait for a period of time (0–t s) before transmitting. When $L_{avg} < \mathrm{T_h}$, it means that the satellite link is relatively idle and the number of transmitted data packets is relatively small. Thus, data packets can be directly transmitted after being queued according to priority. By preprocessing the upcoming data packets, the IBR algorithm can effectively reduce the packet loss rate and improve the throughput.

***Subalgorithm* 1: Packet Preprocessing Algorithm**

1: **Begin**
2: Receive a packet;
3: Get L_{avg} , *flag* and T_h of the packet;
4: **if** ($L_{avg} \geq T_h$)
5: **if (***flag* ==1**)**
6: Send packet to the next satellite directly;
7: **else if** (*flag* ==0)
8: Wait for 0~t seconds randomly and send the packet;
9: **end if**
10: **else if** ($L_{avg} < T_h$)
11: Packets are queued for priority;
12: **end if**
13: **end**

For the *subalgorithm* 2, there exists three cases to get the shortest path using the interstellar link model.

Case 1: Both the source satellite and the destination satellite are in the polar region. It means $L_S > 70° \&\& \; L_d > 70°$. Here, L_s and L_d are latitude values of the source and destination satellite respectively.
Case 2: The source satellites and destination satellites are not in the polar region. It means $L_S < 70° \&\& \; L_d < 70°$.
Case 3: One of the source/destination satellites is in the polar region and the other is outside the polar region.

For case 1, we need to determine whether the source and destination satellites are in the same satellite orbital plane. Because only intra-plane link L_{intra} exists in the polar region. There are no inter-plane link L_{inter}. If the source satellite and destination satellite are in the same orbital plane (L_s==$\mathrm{L_d}$), the data packet can reach the destination satellite from the source satellite along the link L_{intra} in the plane by the path $Ps = \{S_{i_s,j_s}, S_{i_s,j_s+1}, S_{i_s,j_s-1}, S_{i_s,j_d}\}$. If the source satellite and destination satellite are not in the same orbital plane, the data packet first needs to move along the link L_{intra} to near-polar earth plane. Then the packet is transmitted to the satellite in the same orbital plane as the destination satellite. Thus, the packet can finally reach to the destination satellite by the path $Ps = \{S_{i_s,j_s}, S_{i_s,1}, S_{i_d,1}, S_{i_d,j_d}\}$, thus completing the entire transmission request.

For case 2, as the source satellite and destination satellite are not in the polar region, both link L_{intra} and L_{inter} can be used. As the higher the latitude of the satellite, the shorter the interstellar link between planes. Thus, the packet first moves vertically to the satellite in the same orbital plane as the destination satellite, and then moves vertically toward the destination satellite as the shortest path. When the latitude of source satellite is higher than destination satellite ($L_s \geq L_d$), the data packet is first transmitted along the shorter link L_{inter}, and then along the link L_{intra} towards to the destination satellite. Thus, the path Ps is $\{S_{i_s,j_s}, S_{i_d,j_s}, S_{i_d,j_d}\}$. When $L_s < L_d$, the data packet is first transmitted along the link L_{intra} in the plane, and then transmitted along the link L_{inter} between the planes, so as to ensure the shortest path length. Thus, the Ps is $\{S_{i_s,j_s}, S_{i_s,j_d}, S_{i_d,j_d}\}$.

For case 3, whether the destination satellite or source satellite is within the polar region, the data packets must reach the near-polar plane along the link L_{intra}. After the data packet reaches the near-polar satellite, we check whether it is in the same orbit as the destination satellite. If $L_s \geq L_d$, the data packet can be transmitted directly to the destination satellite along the link L_{intra} by the path $Ps = \{S_{i_s,j_s}, S_{i_d,j_s}, S_{i_d,j_d}\}$; Otherwise, the data packet is routed along the link L_{inter} in the near-polar plane, reaching the satellite in the same orbital plane as the destination satellite, and finally according to the link L_{intra} for transmission. Thus, the Ps is $\{S_{i_s,j_s}, S_{i_s,j_s+1}, S_{i_s,j_s-1}, S_{i_d,j_d}\}$.

Based on the above description, for different packet types and different locations of source/destination satellite, the IBR algorithm will perform different paths strategies. At the same time, the ISI model checks the congestion of the satellite link before routing. This way can greatly avoid the congestion caused by the influx of data packets into satellite links. Therefore, the IBR algorithm has the advantages of high throughput, low delay, and high packet loss rate.

***Subalgorithm* 2: Shortest Path Search Algorithm**

Input: S_{i_s,j_s}, S_{i_d,j_d}, L_s, L_d, set $P_s = \varnothing$ as initial path

Output: the shortest path P_s

1: **if** ($L_s > 70^\circ$ && $L_d > 70^\circ$) // **(case 1)**

2: **if** ($L_s == L_d$)

3: $S_{i_s,j_s} \rightarrow S_{i_s,j_s \pm 1} \rightarrow S_{i_s,j_d}$;

$Ps = \{ S_{i_s,j_s}, S_{i_s,j_s+1}, S_{i_s,j_s-1}, S_{i_s,j_d} \}$;

4: **else**

5: $S_{i_s,j_s} \rightarrow S_{i_s,1} \rightarrow S_{i_d,1} \rightarrow S_{i_d,j_d}$;

$Ps = \{ S_{i_s,j_s}, S_{i_s,1}, S_{i_d,1}, S_{i_d,j_d} \}$;

6: **end if**

7: **else if** ($L_s < 70^\circ$ && $L_d < 70^\circ$) // **(case 2)**

8: **if** ($L_s \geq L_d$)

9: $S_{i_s,j_s} \rightarrow S_{i_d,j_s} \rightarrow S_{i_d,j_d}$;

$Ps = \{ S_{i_s,j_s}, S_{i_d,j_s}, S_{i_d,j_d} \}$;

10: **else**

11: $S_{i_s,j_s} \rightarrow S_{i_s,j_d} \rightarrow S_{i_d,j_d}$;

$Ps = \{ S_{i_s,j_s}, S_{i_s,j_d}, S_{i_d,j_d} \}$;

12: **end if**

13: **end if**

14: **else // (case 3)**

15: **if** ($L_s == L_d$)

16: $S_{i_s,j_s} \rightarrow S_{i_s,j_s-1} \rightarrow S_{i_s,j_d}$;

$Ps = \{ S_{i_s,j_s}, S_{i_s,j_s-1}, S_{i_d,j_d} \}$;

17: **else**

18: $Ps = \{ S_{i_s,j_s}, S_{i_s,j_s+1}, S_{i_s,j_s-1}, S_{i_d,j_d} \}$;

19: **end if**

20: **end if**

5 Simulation and Results

In this section, by OMNeT++ simulation, we compare the performance of the IBR algorithm with the existing algorithms DRA [21] and LCPR [22]. In our simulation, we don't consider the satellites of polar region. Thus, each satellite in our simulation is assigned for two L_{intra} links and one L_{inter} link. The track height and inclination of satellite are set at 780 km and 86.4°, respectively. The capacity of ISL is 20 Mbps. In ISI, the data transmission rate is set to 1–2 Mbps. Each packet size is 1024 bytes. And the data packet obeys the Poisson distribution and is generated with the speed of 300 packets per second. Both high priority packets and low priority packets are subject to Poisson distribution.

The Fig. 6 has displayed the relationship between packet loss rate and data transmission rate for IBR, DRA and LCPR algorithms. The average packet loss rate of IBR is about 4% lower than LCPR algorithm, and 10% lower than that of DRA algorithm. When the data transmission rate is lower than 1.4 Mbps, the packet loss rate of IBR is lower than 3.5% and the growth rate is slow. This is because we queue the data packets and introduce random back-off time to effectively reduce the packet loss rate. When the data transmission rate exceeds 1.5 Mbps, more data packets enter the satellite link. Thus, the impact of link congestion on the packet loss rate increases gradually, resulting in a positive correlation between the packet loss rate and the data transmission rate.

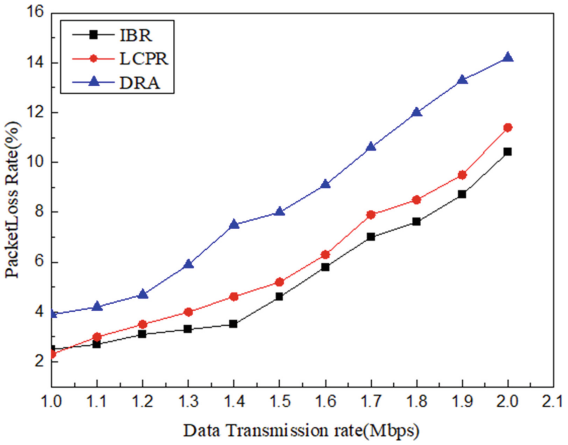

Fig. 6. The relationship between packet loss rate and data transmission rate.

Figure 7 shows the performance of end-to-end delay. IBR algorithm optimizes the average end-to-end delay by 7% and 11% respectively compared with LCPR algorithm and DRA algorithm. Because ISI can monitor the network status and adjust the routing path in real time. When the link is idle, data packets will be queued for transmission.

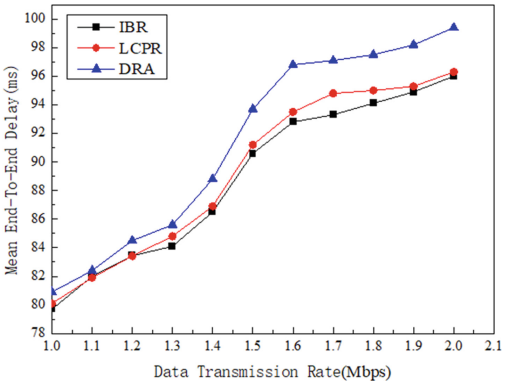

Fig. 7. The relationship between average end-to-end delay and data transmission rate.

When the link congestion occurs, high priority packets have no waiting time and can be transmitted first, which effectively improves the average end-to-end delay.

In Fig. 8, our experiments compare throughput performance. The throughput is also positively correlated with the data transmission rate. When the data transmission rate increases, the data packet increases and the throughput also improves. IBR algorithm increases the average throughput by 5.8 Kbps and 11.6 Kbps compared with LCPR and DRA algorithms, respectively. Thus, the IBR algorithm has a significant improvement in terms of throughput.

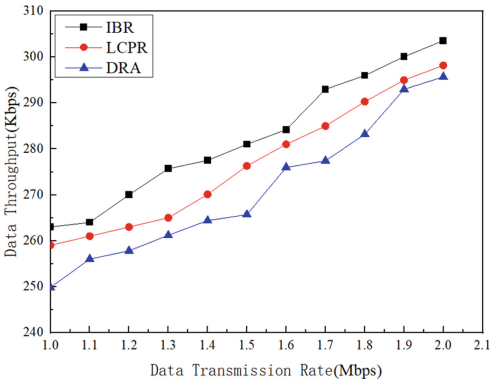

Fig. 8. The relationship between throughput and data transmission rate.

6 Conclusion

Aiming at the problem of satellite link congestion caused by the large number of service types in satellite IoT, we have proposed the ISI system model and designed an effective routing algorithm TBR by using INB technology. The IBR routing algorithm is mainly designed based on the ISI system model. The ISI model realizes the automatic configuration and security monitoring of the underlying network by parsing the business intent. In TBR algorithm, after network status detection through ISI model, the operation of data packet access or waiting is decided according to the congestion state of satellite link. Then, a shortest path is found according to the logical position of satellites and the interstellar link between satellites. However, the proposed routing algorithm only considers the single-layer satellite network, but the actual satellite IoT is complicated. Therefore, we will consider more complex satellite routing models, and design flexible and universal routing strategies in the future.

Acknowledgement. This research was partially supported by National Natural Science Foundation of China (61971105), Key Science and Technology Project of China Southern Power Grid Co., Ltd. (GZKJXM20210011), Sichuan science and technology program (2021YFG0150, 2021YFG0037, 2022YFG0173).

References

1. Kodheli, O., Maturo, N., Shankar, B., et al.: Satellite communications in the new space era: a survey and future challenges. IEEE Commun. Surv. Tutor. **23**, 70–109 (2021)
2. Zhu, X., Jiang, C.: Integrated satellite-terrestrial networks toward 6G: architectures, applications, and challenges. IEEE Internet of Things J. **9**, 437–461 (2022)
3. Yan, Y., Guang, H., Hui, X.: A survey on secure routing protocols for satellite network. J. Netw. Comput. Appl. **145**, 1–17 (2019)
4. 2017 SIA State of Satellite Industry Report. http://www.sia.org/annual-state-of-the-satellite-industry-reports/2017-sia-state-of-satellite-industry-report/. Accessed on 2017
5. Hassan, N.U.L., Huang, C., Yuen, C., Ahmad, A., Zhang, Y.: Dense small satellite networks for modern terrestrial communication systems: benefits, infrastructure, and technologies. IEEE Wirel. Commun. **27**, 96–103 (2020)
6. Gao, Y., Ao, H., Zhou, Q., et al.: Modeling of satellite communication systems design with physical layer security. In: International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1680–1683 (2018)
7. Lu, Y., Min, G., Zuo, Z., Liang, R., Duan, Z.: Structural performance of satellite networks: a complex network perspective. IEEE Syst. J. **15**, 3848–3859 (2021)
8. Jain, S., Kumar, A., Mandal, S., et al.: B4: experience with a globally-deployed software defined WAN. In: ACM SIGCOMM Conference, Hongkong, pp.3–14 (2013)
9. Patel, A.N., Ji, P.N., Wang, T.: QoS-aware optical burst switching in OpenFlow based software-defined optical networks. In: 17th International Conference on Optical Networking Design and Modeling (ONDM), Brest, pp. 275–280 (2013)
10. Yang, X.N., Xu, J.L., Lou, C.Y.: Software-defined satellite: a new concept for space information system. In: 2nd International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Harbin, pp. 586–589 (2012)
11. Gudipati, A., Perry, D., Li, L.E., et al.: SoftRAN: software defined radio access network. In: 2nd ACM SIGCOMM Workshop Hot Topics in Software Defined Networking, Hongkong, pp. 25–30 (2013)
12. Bao, J.Z., Zhao, B.K., Yu, W.R.: OpenSAN: a software-defined satellite network architecture. In: SIGCOMM Conference, Chicago, pp. 347–348 (2014)
13. Intent-Based Networking-Building the bridge between business and IT. https://www.cisco.com/c/en/us/solutions/intent-based-networking.html. Accessed on 2017
14. Intent NBI. https://www.opennet-working.org/. Accessed on 2016
15. Mohammed, A., Gharbaoui, M., Martini, B., et al.: SDN controller for network-aware adaptive orchestration in dynamic service chaining. In: IEEE NetSoft Conference and Workshops, pp. 126–130 (2016)
16. Ding, J.D., Zhang, Y., Li, R.N., et al.: A distributed routing algorithm for LEO satellite networks. In: International Conference on Human Centered Computing, Kazan, pp. 258–264 (2017)
17. Liu, X.M., Yan, X.M., Jiang, Z.Q., et al.: A low-complexity routing algorithm based on load balancing for LEO satellite networks. In: Vehicular Technology Conference, Boston, pp. 1–5 (2015)
18. Zhang, L.L., Wang, X.W., Huang, M.: A routing scheme for software-defined satellite network. In: 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, pp. 24–31 (2017)
19. Guo, A.D., Zhao, C.L., Xu, F.M., et al.: LEO satellite routing algorithm in software defined space terrestrial integrated network. In: 17th International Symposium on Communications and Information Technologies (ISCIT), Cairns, pp. 1–6 (2017)

20. Cai, Y., Zhang, S., Hou, J., Jiao, H.: Internet of Things over GEO satellite: a novel space information network solution. In: 2021 International Wireless Communications and Mobile Computing (IWCMC), pp. 177–181 (2021)
21. Ekici, E., Akyildiz, I., Bender, M.: Datagram routing algorithm for LEO satellite networks. In: Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 500–508 (2000)
22. Liu, H.Y., Sun, F.C., Yang, Z.A., et al.: A novel distributed routing algorithm for LEO satellite network. In: International Conference on Industrial Control and Electronics Engineering (ICICEE), Xi'an, pp. 37–40 (2012)