A DTN Routing Scheme for LEO Satellites Topology

YaoWen Qi*, Li Yang1 and ShuangJian Chen1
Communication and Networks Key Laboratory, DaLian University, DaLian, LiaoNing, 116622, China
*Corresponding author’s e-mail: 441826613@qq.com

Abstract. In delay/disruption-tolerant satellite networks, the existing routing algorithms have problems of long transmission delay, high packet loss rate, excessive network overhead and other issues. In the view of these shortcomings, an improved contact graph routing algorithm, ICGR, is proposed in this paper. Firstly, the impact of queuing delay is considered by defining a queue factor and proposing a queue scheduling mechanism in routing calculation; secondly, the effects of contact's remaining capacity are taken into account to avoid the selected path being unable to forward the messages; thirdly, routing update mechanism is optimized by designing a message path update factor to strike a balance between real-time updates of transmission and resource consumption; finally, a LEO satellite network simulation platform is built with OPNET to verify the performance of this algorithm. Simulation results show that the ICGR, compared with the ED and CGR algorithm, has advantage in packet delivery rate, average transmission delay and network overhead.

1. Introduction
Satellite networks have features like limited transmission energy, rapid mobility, frequent switching, sparse node density, frequent equipment failures, time-varying topology and so on [1]. So, satellite network is intermittent contact, which makes it form a typical delay/disruption tolerant network (DTN) [2]. DTN system takes the message exchange mechanism of storage-carry-forward by adding a bundle layer between the transport layer and application layer [3]. The key issue of routing algorithm in DTN network is to deliver the message with lower consumption of network resources as soon as possible. Main DTN routing algorithms can be classified into the following categories:

- algorithms are based on flooding, such as Epidemic algorithm [4], Spray and Wait algorithm [5] and PROPHET algorithm [6]. These algorithms can calculate routes without any information from network, which reduces the complexity of algorithms’ computation. But with more copies of messages adopted, the consumption of network resources will also increase which explains why it cannot be applied to satellite network whose resources are limited [7][8].

- algorithms based on knowledge base, such as ED algorithm and MED algorithm [9]. These algorithms predict occurrence of the edge of link based on calculating transmission delay and propagation delay [10], and a predicted link can identify the first path from source node to destination node. These algorithms contribute to the reduction of network redundancy. However, they will result in large transmission delay and cannot fit high dynamics of satellite network because of adopting source routing mechanism and neglecting status of storage in intermediate nodes [11] and messages in system.

- algorithms based on contact graph, such as CGR routing algorithm [12]. These algorithms make full use of the predictability and periodicity of contact between satellite nodes. Upon a message
arriving each intermediate node, it will recalculate the forwarding path according to access information of entire network [13]. As a result, it can well adapt to the dynamic of network. However, without considering the message’s queuing delay in routing calculation [14], it will lead to message’s missing of currently available contact and the reduction of message delivery rate to some extent.

This paper presents an improved routing algorithm, ICGR, according to the characteristic of LEO satellite network. The main research work has been carried out in the following three aspects: firstly, considering the impact of queuing delay in routing calculation, a queue factor is defined and a queue scheduling mechanism is proposed; secondly, the effects of contact’s remaining capacity are thought about in order to avoid that the selected path cannot forward the messages; thirdly, we optimize routing update mechanism by seeking out the best contact plan update threshold [18] which strikes a balance between real-time updates of transmission and resource consumption.

2. System Model
In this paper, we build a contact graph model based on the contact relationship between the nodes in the network. Let $G = (V, E)$ be a delay/disruption-tolerant satellite network in which $V$ denotes the set of satellites, ground stations and other network nodes; $E$ denotes the set of inter-satellite links, ground-satellite links and other communication links. Each edge $e \in E$ has a label $cp_e(t) = \{(t_{e,\text{start}}, t_{e,\text{end}})\in \mathbb{R}^+\}$ which specifies its contact periods. $t_{e,\text{start}}$ denotes the start time of the $i$th contact of each edge and $t_{e,\text{end}}$ denotes the stop time of the $i$th contact of each edge. We define a parameter $RC_e(t) \in \mathbb{R}^+ \cup \{+\infty\}$ to indicate the remaining capacity of the current contact. In addition, since each node in DTN network has limited resources, each $v \in V$, let $buf(v, t) \in \mathbb{R}^+ \cup \{+\infty\}$ represents the rest buffer size of the node V in time t.

3. Algorithm Design

3.1. Algorithm Process
The process of ICGR routing algorithm is shown is figure 2.
The process can be described as:

- Analyze the time data of link connection and disconnection according to the movement of the satellites and create a link state table on each node.
- Calculate the routing table of the whole network according to link state table, wherein the routing cost is considered including queuing delay and remaining capacity of the contact.
- Find the minimum delay path from the source node to the destination node according to the routing table and write the optimal next hop node in the packet.
- If the network topology changes, update the link status table on nodes and return to step 2 to continue the routing discovery; if not, continue step 5.
- When the local queue on the node is changed, judge whether the optimal next hop node needs to be updated.
- After packets successfully reach the next node, consider whether they reach destination node. If the answer is yes, packets have been successfully delivered to destination node; otherwise, return to step 2 to continue transmission until packets reach the destination node.

Wherein, the calculation of queuing delay and remaining capacity of the contact and the optimization strategy for routing update in step ②⑤ have an important influence on improving
message delivery rate, reducing transmission delay and network overhead. This section develops research concentrating on the above aspects in order to improve the performance of DTN network.

3.2. Link Delay Calculation under Influence of Queuing Delay

Minimizing delay can reduce the time taken by the message in the network and the resource requirements. Besides, it will indirectly improve the probability of message delivery. Here are three types of delay in ICGR routing algorithm:

1) Transmission delay: time spent on injecting the whole message into a contact. The Estimated Capacity Consumption (ECC)\(^{[15]}\) of the message is defined as the sum of its payload and its header size. Then the transmission delay can be obtained from the following formula:

\[
t_{tra} = \frac{ECC}{Rate(e)}
\]

Where \(Rate(e)\) denotes the link rate and in this paper we assume it as a constant value.

2) Propagation delay: the time consumed in the link. In the satellite network, the satellite channel is composed of satellite links, and the link length is the sum of the communication links. The distance between satellites is shown in Figure 3.

![Schematic diagram of inter-satellite distance](image)

Figure 3. Schematic diagram of inter-satellite distance

Calculate the distance between the satellites by the instantaneous geocentric angle and the orbit height. Instantaneous geocentric angle is calculated as following:

\[
\alpha = \arccos[\sin \alpha_1 \sin \beta_1 + \cos \alpha_1 \cos \beta_1 \cos(\alpha_1 - \alpha_2)]
\]

\((\alpha_1, \beta_1), (\alpha_2, \beta_2)\) denote latitude and longitude of the two sub-satellite points. The distance between satellites is calculated as following:

\[
L = \sqrt{(R+ H_1)^2 + (R+ H_2)^2 - 2 \cos \alpha (R+ H_1)(R+ H_2)}
\]

The above \(R\) is the radius of the earth. \(H_1\) is the height of the satellite A and \(H_2\) is the height of the satellite B. Putting the instantaneous geocentric angle into the distance formula, we can calculate the communication distance between the two satellites.

The sum of the link length is calculated as follows:

\[
L_{sum} = \sum L_i, i = 1, 2, 3... n
\]

Where \(L_i\) denotes the length of each link, \(n \in N^+\). The propagation delay is calculated as follows:
3) Queuing delay: the time spent in waiting for effective contact on an exit of the node. In this paper, a queuing factor called \( que \) is defined to determine the location of the newly arrived message in the queue of the node. This factor considers the priority of message, ECC, propagation delay, transmission delay and residual time in a comprehensive way.

\[
que = w_1 \ast (t_{tra} + t_{pro}) + w_2 \ast Sur(t) + w_3 \ast ECC + w_4 \ast Prio, (\sum_{i=1}^{4} w_i = 1)
\]

Where \( t_{tra} \) denotes transmission delay, \( t_{pro} \) denotes propagation delay, \( Prio \) denotes the priority of message, which has three types of priority: bulk, standard, expedited. \( Sur(t) \) denotes residual time of the message. Each message in DTN network has a TTL(time-to-live)\[11\], so \( Sur(t) \) is defined as:

\[
Sur(t) = TTL - (t - t_{create})
\]

Where \( t \) is the current time, \( t_{create} \) is the create time of message. In formula (6), \( w_i (i = 1,2,3,4) \) denotes the weight coefficients of the above attributes. In this paper, we use the AHP method to calculate the weights. Decision maker compares the importance of \( n \) attributes in pairs. The relative importance of the \( p \) attribute to the \( q \) attribute is denoted as \( \alpha_{pq} \) and regard it as the approximate value of the ratio of the weight of the attribute \( p \) to the attribute \( q \), \( \alpha_{pq} = \frac{w_p}{w_q} \). The results of pairwise comparison of \( n \) targets form a matrix \( A \).

\[
A = \begin{bmatrix}
\alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\
\alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn}
\end{bmatrix}
\]

Then \((A - nI)w = 0\), where \( I \) is the unit matrix. If the value of the matrix is accurate, the formula is strictly equal to 0. If the estimate is not accurate enough, then the small perturbation of the element in the \( A \) represents a small perturbation of the eigenvalues. So

\[
Aw = \lambda_{\text{max}}w
\]

\( \lambda_{\text{max}} \) is the maximum eigenvalues of a matrix \( A \). According to the formula, we can get the eigenvector that is weight vector \( w = \begin{bmatrix} w_1, w_2, \ldots, w_n \end{bmatrix}^{T} \).

In order to judge the scientific nature of matrix in this method, the concept of consistency ratio \[16\](CR) is introduced. CR is presented by the ratio of the consistency index (CI) and the random index (RI). It can be used to determine whether the matrix \( A \) is accepted. Among them:

\[
CI = \frac{\lambda_{\text{max}} - n}{n - 1}
\]

The RI value of the matrix corresponding to the order of \( n \) is as follows:

| \( n \) | 2  | 3  | 4  | 5  | 6  | 7  |
|-------|----|----|----|----|----|----|
| RI    | 0.0 | 0.58 | 0.90 | 1.12 | 1.24 | 1.32 |

If the ratio of CR>0.1, it indicates that the elements of the estimated consistency is too poor and should be re-estimated. If CR<0.1, it can be considered that the estimated \( \alpha_{pq} \) is basically the same, and the \( w \) can be obtained from formula.

When a message arrives at the node, we use formula (6) to calculate and update the queuing factor \( que \) of all messages for the purpose of putting the message into the appropriate location of the queue. Then we can calculate the message queuing delay:
t_{\text{que}} = \frac{\sum \text{ECC}}{\text{Rate}(e)} \tag{10}

Where \( \sum \text{ECC} \) is the sum of ECC of all messages that ranked before the message in the queue.

### 3.3. Calculation of Contact Remaining Capacity

In the model defined in Section II, we define \( Q(e, t, s) \) as the length of queue in source side of the edge \( e \) obtained by local node \( s \) in time \( t \). In other words, it is equal to the sum of all messages’ ECC storage in node \( s \) in time \( t \). Define \( t_{\text{residual}}(e) \) as the residual time of contact \( e \); \( m \) as the ECC of the message to transmit.

Assuming there is a node called \( u \) in the current network to send a message to node \( v \) through contact \( e \), whose rate is \( \text{Rate}(e) \). When message arrives at the node \( u \), contact \( e \) is not over yet, which is \( t_{\text{arrival}}(u) < t_{e,\text{end}}^{i} \), then

\[
t_{\text{residual}}(e) = t_{e,\text{end}}^{i} - \max \{ t_{\text{arrival}}(u), t_{e,\text{start}}^{i} \} \tag{11}
\]

\[
\text{RC}_{e}(t_{\text{arrival}}(u)) = \text{Rate}(e) \cdot t_{\text{residual}}(e) - Q(e, t_{\text{arrival}}(u), u) \tag{12}
\]

Where \( t_{e,\text{start}}^{i} \) and \( t_{e,\text{end}}^{i} \) denote the \( i \)th start and stop time of link \( e \). By comparing \( m \) and \( \text{RC}_{e}(t_{\text{arrival}}(u)) \) to determine whether the current capacity of the contact can transmit data.

When a node is sent to transmit a message, our algorithm considers the relationship of the current queue length \( Q(e, t, s) \) and remaining capacity of contact:

1. When a message can be transmitted and the contact has not begun, it needs to calculate the capacity of next contact. If it is sufficient, wait for the start of contact; otherwise, consider the next contact;
2. When a message can be transmitted and the contact has started, there is no need to calculate the remaining capacity of current contact, \( \text{RC}_{e}(t) \). If it is sufficient, send the message; otherwise, consider the next contact.

### 3.4. Routing Calculation

ICGR algorithm improves the Dijkstra algorithm proposed in [17] to calculate the next hop, which can make use of time-vary cost of contact to calculate the minimum delay path.

**Input:** \( G=(V, E), s, \ \text{current time } T, \ \text{cost function } w(e, t) \);

**Output:** \( L \) (denotes the cost of packets reach other node through the shortest path)

\[
Q \leftarrow \{V\};
\]

for each node \( v \in V \)

if \( v \neq s \)

\( L[s] \leftarrow 0, \ L[v] \leftarrow \infty \);

end if

end for

while \( Q \neq \{\} \) do

let \( u \in Q \) that satisfy \( L[u] = \min_{x \in Q} L[x] \);

\( Q = Q - \{u\}; \)

for each edge \( e \in E \) that start from \( u \), do

if \( L[v] > (L[u]+w(e, L[u]+T)) \)

\( L[v] \leftarrow (L[u]+w(e, L[u]+T)) \);

end if

end for

end while

Figure 4. The Dijkstra algorithm used to calculate the minimum delay path
Where \( w(e, t) \) denotes the cost of sending message through contact \( e \) in time \( t \). It is not only related to contact and time, but also related to the size of current message and local node. So the cost function is written as \( w(e, t) = w'(e, t, m, s) \), where \( m \) is the size of message and \( s \) is the node to start dijkstra algorithm. Which is calculated as

\[
\begin{align*}
  w'(e, t, m, s) &= t' (e, t, m, s) - t + d(e, t) \\
  t' (e, t, m, s) &= \min\{t' \mid \int_{x=t}^{t'} R_C x \geq m + Q(e, t, s)\}
\end{align*}
\]

In which, \( t' \) represents the earliest time for injecting the message into network through contact \( e \); \( d(e, t) \) represents the function of transmission delay. The algorithm selects the remaining capacity which is big enough for transmitting message and the minimum delay path as a reference path. The source node stores reference path in extension block of the message and the message is passed in accordance with it. But when message arrives at each node, it will recalculate the optimal next hop in the current state, which can have some adaptability to unexpected situation of the link. This kind of routing algorithm, which calculates the path while sending, will result in the phenomenon of message shocking between nodes; therefore, each message will record the node that it passes in the algorithm to avoid the emergence of shock.

3.5. Optimization of Routing Updates

When the satellite network topology changes, it is necessary to recalculate the transmission path of the message. And when a message in the queue of the sending node is sent out which leads to the queuing delay of current message changes, the previously calculated transmission path may be not the best at this time. So it also needs to recalculate the optimal next hop. But in the satellite network, calculating the cost for each message in any precise time means the great occupancy of the satellites’ storage, which will affect the effective use of resource on satellites and be difficult to realize. In this paper, we import the contact plan update threshold, which is defined as the ratio of the queuing delay of the current message in the queue to the current contact time. That is:

\[
\beta = \frac{t_{que}}{t_{e\_end} - t_{e\_start}}
\]

For the same contact, higher message path update factor means that the message wait longer in the queue and the worse adaptability of the high dynamic network. It will lead to the previously calculated next hop failing to be optimal or a sending failure. This indirectly reduces the message delivery rate and increases message transmission delay. Theoretically the optimal next hop of message should be updated in real time, but this will increase the computational complexity and result in significant overhead.

A routing update threshold \( \lambda \) is set in this paper to optimize the routing update strategy. According to different values of message path update factor \( \beta \), we take different route update strategies. Specific programs are as follows: ① When \( \beta > \lambda \), it needs to update the queuing delay of the message and recalculate the next hop to transmit the message; ② When \( \beta \leq \lambda \), forwarding the message according to previously calculated next hop. The routing update threshold \( \lambda \) is directly related to the average transmission delay of the message, therefore, the value of \( \lambda \) will vary from one topology of network to another. We will calculate the value of \( \lambda \) for the topology of 9 LEO satellites which obey the walker distribution in section IV and complete the optimization of routing update.

4. Simulation And Performance Evaluation

4.1. Simulation Setting

To verify the performance of the ICGR algorithm, we use OPNET Modeler 14.5 to model the LEO network and carry out the simulation evaluation in this paper. The LEO network consists of 9 satellites
which obey walker distribution and 3 ground stations. The network model parameters are shown in Table 2. In order to reflect the purpose, the satellites are the source nodes to send data and the ground stations are the destination node to receive data. In this paper, the size of generated messages obeys Poisson distribution with the expected value of 128KB, and the interval of the messages is set to 0.5 seconds.

| Parameters                  | Values     |
|-----------------------------|------------|
| Altitude of satellite orbit | 1680km     |
| Number of orbital plane     | 3          |
| Number of satellites in each orbit | 3     |
| Inclination of satellite orbit | 60       |
| Link rate                   | 5Mbps      |
| Ground station              | Kashi, Beijing, Sanya |

As the different value of the routing update threshold $\lambda$ will affect the routing update strategy of messages in the node queue, it will then affect the performance of the routing algorithm. So firstly, we will discuss the relationship between the value of $\lambda$ and the average transmission delay. The average transmission delay is the average value of the transmission delay of packets received by all destination nodes, the smaller the average transmission delay is, the higher the communication efficiency of the network will be.

![The relationship between average transmission delay and routing update threshold $\lambda$](image)

Figure 5. The relationship between average transmission delay and routing update threshold $\lambda$
In the simulation environment of this paper, we set the node cache to 30MB, and the relationship between the average transmission delay and the routing update threshold $\lambda$ is shown in Figure 5. It can be seen that, when $\lambda < 0.3$, the average transmission delay decreases with the increase of $\lambda$; when $\lambda > 0.3$, the average transmission delay increases with the increase of $\lambda$. This is because when $\lambda$ is small, the computation load of routing updates is high, which reduces the performance of the algorithm; when $\lambda$ is large, routing update frequency is too low, which also reduces the performance of the algorithm. So we set routing update threshold $\lambda$ to 0.3.

4.2. Simulation Analysis
In the same network scenarios, we respectively carry out simulations for ED routing algorithm, CGR routing algorithm and ICGR routing algorithm. By changing the cache size of nodes, we inspect the
performance of each algorithm from the message delivery rate, average transmission delay and network overhead ratio. The message delivery rate = number of delivery messages / number of messages generated on source node; network overhead ratio = (number of messages relayed in the network - number of delivery messages) / number of delivery messages, smaller network overhead means better performance of the network.

Figure 6. Message delivery rate

Figure 6 shows the message delivery rate of three routing algorithms. As shown in the picture, the message delivery rate of three routing algorithms all increases with the increase of node cache. This is because with the increase of node cache, more messages can be stored in the nodes, thereby reducing the rate of loss of messages. The performance of ICGR routing algorithm is significantly better than CGR routing algorithm and ED routing algorithm. As node cache varies from 10MB to 60MB, message delivery rate of ICGR increases of 5.8%~58.3% than the ED routing algorithm and increases of 4.3%~11.7% than the CGR routing algorithm. When the node cache is small, the advantages of ICGR routing algorithm is more obvious. This is because in the completely predictable space environment, ICGR forwards messages with the most reliable link connection and optimize the queue scheduling.

Figure 7. Average transmission delay
Figure 7 shows the average transmission delay of three routing algorithms. As shown in the picture, the average transmission delay of three routing algorithms all decrease with the increase of node cache. This is because with the increase of node cache, more messages can be stored on the next hop node without waiting for the next contact and can be sent as soon as possible. Since the ICGR routing algorithm adds queue scheduling, as well as the calculation of queuing delay, which increases the complexity of algorithm. So on the indicator of average transmission delay, ICGR increases fewer than CGR routing algorithm. When node cache is 10MB, average transmission delay of ICGR is a little higher than the CGR routing algorithm and when node cache varies from 20MB to 60MB, it decreases of 1.9%~4.4% than the CGR routing algorithm. As node cache varies from 10MB to 60MB, average transmission delay of ICGR decreases of 5.1%~29.7% than the ED routing algorithm.

Figure 8 shows the network overhead ratio of three routing algorithms. As shown in the picture, the network overhead ratio of three routing algorithms all decrease with the increase of node cache. This is because with the increase of node cache, the counts of forwarding redundant message in the network decrease. The performance of ICGR routing algorithm proposed in this paper is significantly better than ED routing algorithm and slightly better than CGR routing algorithm. As node cache varies from 10MB to 60MB, network overhead ratio of ICGR decreases of 4%~33% than ED routing algorithm and decreases of 1.6%~5.8% than the CGR routing algorithm.

5. Conclusion
The CGR routing algorithm relies on the correctness of the contact plan, so it cannot cope with the change of temporary topology. For this reason, we propose an improved routing algorithm, ICGR, in this paper. In this algorithm, we firstly take into account the impact of queuing delay in routing calculation and propose a new queue scheduling algorithm; secondly consider the effects of link status, compare the remaining capacity of the contact when a message arrives in the subsequent nodes with the desired size of message to transmit, to avoid the selected path is not fit to forward messages; thirdly, optimize routing update mechanism, to avoid tremendous computational overhead brought by the message transmission path’s updating in real time. By building the LEO satellite network simulation platform, we analyze the performance of ICGR routing algorithm. The simulation results show that ICGR routing algorithm can effectively improve the message delivery rate, reduce the average transmission delay and network overhead. So it can better satisfy the requirements of LEO satellite network.
Acknowledgments
Thank the teacher for her help in the writing process, thank the laboratory for providing me with a quiet and serious research atmosphere, thank the students for their cooperation in coding simulation, and thank the review experts of the organizing committee for their recognition.

References
[1] Yuan Rao, Ru-chuan Wang. Agent-based Load Balancing Routing for LEO satellite networks.[J]. Computer networks, 2010, 54(17): 3187-3195.
[2] Yue Cao, Zhili Sun. Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges.[J]. IEEE Communications Surveys & Tutorials, 2013, 15(2) : 654-677.
[3] M. Grossglauser, D. N. C. Tse. Mobility Increases the Capacity of Ad-Hoc Wireless Networks.[J]. IEEE/ACM Transactions on Networking, 2002, 10(4): 477-486.
[4] Amin Vahdat, David Becker. Epidemic Routing for Partially-Connected Ad-Hoc Network.[R]. Duke University Technical Report Cs-2000-06 Tech, 2000: 1-14.
[5] Spyropoulos T, Psounis K, Raghavendra C S. Spray and Wait: Efficient Touting in Intermittently Connected Mobile Networks.[C]. ACM SIGCOMM WDTN, 2005: 252-259.
[6] Lindgren A, Doria A, Schelen O. Probabilistic Routing in Intermittently Connected Networks.[J]. Mobile Computing and Communications Review, 2003, 7(3): 19-26.
[7] Remi Diana, Emmanuel Lochin and Laurent Franck. “A DTN routing scheme for quasi-deterministic networks with application to LEO satellites topology”, IEEE 2012
[8] Li Yang, Fuli Li, Dong Zhang. Research on Routing Algorithm in Disruption-Tolerant Satellite Networks,[J]. Fire Control & Command Control, 2016, 41(3) : 57-61.
[9] Jain S, Fall K, Patra R. Routing in A Delay Tolerant Networks.[C]. The 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2004, 34(4): 145-158.
[10] Bing Lin, Hongyan Li, Yan Long. A Buffer-limited Maximum Throughput Routing Algorithm for Satellite Network.[C]. 2015 22nd International Conference on Telecommunications, 2015: 378-383.
[11] Lu Tan, Qing Guo, Minchuan Yang. A Routing Strategy for Two-Layered Satellite DTN Network Based on Earliest Genetic Algorithm.[C]. The 12th Academic Annual Conference on Satellite Communications, 2016: 325-334.
[12] S. Burleigh. Contact Graph Routing. Internet-Draft, July 2010. work-in-process http://tools.ietf.org/html/draft-burleigh-dtnrg-cgr.
[13] Hongchen Yan, Qingjun Zhang, Yong Zhang. Research on Routing Performance of Navigation Satellite Network Based on Contact Graph Routing Algorithm.[C]. The 5th China Satellite Navigation Conference, 2014: 111-115.
[14] Juan A. Fraire and Jorge M. Finochietto. Design Challenges in Contact Plans for Disruption-Tolerant Satellite Networks.[J]. Satellite Communications and Networking, 2015, 53(5): 163-169.
[15] Giuseppe Araniti, Nikolaos Bezirgiannidis, Edward Birrane. Contact Graph Routing in DTN Space Networks: Overview, Enhancements and Performance.[J]. IEEE Communications Magazine, 2015, 53(3): 38-46.
[16] Yue CY. Decision theory and method[M]. Beijing, Science Press, 2003.
[17] J. Segui, et al. Enhancing contact graph routing for delay tolerant space networking[C]. IEEE GLOBECOM 2011, Houston, USA, Dec. 2011.
[18] N. Bezirgiannidis, et al. Contact Graph Routing Enhancements for Delay Tolerant Space Communications[C]. ASMS 2014, Livorno, Italy, Sept. 2014, pp.17-23.