

# 基于邻居节点属性的低轨卫星网络 路由算法研究

王 征, 张大铭, 付继伟

(北京宇航系统工程研究所, 北京 100076)

**摘要:** 在远海目标探测数据传输至地面的过程中, 采用低轨卫星网络转发具有发射成本低、传输时延小的优势; 低轨卫星星座规模较大, 星上资源受限, 负载分布不均, 设计合理的路由算法对提升数据回传的质量具有重要意义; 现有的路由算法对拥塞现象的处理能力不足, 针对此问题, 提出了一种基于邻居节点属性的低轨卫星网络路由算法, 根据各邻居卫星的候选次序属性和综合负载属性, 确定待转发集合, 合理分配集合内不同卫星的转发概率, 为数据转发提供更充分的路径选择; 仿真结果表明, 该算法的平均时延和丢包率等性能指标相比于传统算法得到优化, 能够保证探测数据高效可靠传输。

**关键词:** 低轨卫星网络; 路由算法; 候选次序; 时延; 丢包率; 路由跳数

## Research on Routing Algorithm of LEO Satellite Network Based on Neighbor Node Attributes

WANG Zheng, ZHANG Daming, FU Jiwei

(Beijing Institute of Astronautical Systems Engineering, Beijing 100076, China)

**Abstract:** During the process of transmitting data from offshore target detection to the ground, low earth orbit (LEO) satellite networks have the advantages of low launch costs and transmission delays. Due to the large size of LEO satellite constellations, limited on-board resources, and uneven load distribution, it is of great significance to design a reasonable routing algorithm to improve the quality of data transmission. Existing routing algorithms have insufficient ability to process congestion phenomena. To solve this problem, a routing algorithm for LEO satellites based on neighbor node attributes is proposed. According to the candidate order attributes and comprehensive load attributes of neighbor satellites, it determines the set to be forwarded, reasonably allocates the forwarding probability of different satellites within the set, and provides a more adequate route selection for data forwarding. Simulation results show that compared with traditional algorithms, the proposed algorithm optimizes the average delay and packet loss rate, ensuring the efficient and reliable transmission of detection data.

**Keywords:** LEO satellite network; routing algorithm; candidate order; delay; packet loss rate; route hops

## 0 引言

在远海目标探测中, 通信基础设施缺乏, 难以通过陆地上 4G/5G 等移动通信方式将探测数据传输至地面, 而卫星通信具有覆盖范围广、受地形和天气影响较小的特性, 因此成为远海与地面建立信息传输链路的有效方式<sup>[1]</sup>。在卫星通信中, 低轨卫星 (LEO, low earth orbit) 网络具有发射成本低、链路损耗小、传输时延低等优势<sup>[2-5]</sup>, 具有极高的战略价值, 可以在探测数据回传中发挥重要作用。

低轨卫星星座规模较大, 流量分布不均, 因此设计高效可靠的路由算法尤为关键。然而, 低轨卫星网络具有拓扑变化剧烈、星上资源有限等特点, 使得 AODV、OSPF 等地面路由协议直接应用到低轨卫星系统具有很大困难<sup>[6-7]</sup>。

因此设计适用于低轨卫星网络的路由算法, 对提升网络服务质量 (QoS, quality of service) 具有重要意义<sup>[8]</sup>。

近年来, 许多学者从多个角度对低轨卫星网络路由算法开展研究。文献 [9] 将链路状态感知应用于路由决策中, 可以在降低卫星能耗的同时缓解局部拥塞, 但算法未在星座中进行仿真验证。文献 [10] 使用蚁群算法从全局和局部两个方面优化流量分配, 处理突发拥塞更加及时, 但算法易陷入局部最优解。文献 [11-12] 基于强化学习策略设计路由算法, 在无需先验信息的情况下通过迭代学习获取最佳路径, 算法收敛时间受网络规模的制约较大。文献 [13] 提出一种基于树突神经网络的路由算法, 能够及时感知星间链路状态, 自动调整全局链路权值, 智能地进

收稿日期: 2024-07-11; 修回日期: 2024-08-22。

基金项目: 中国航天科技集团自主研发项目 (YF-ZZYP-2020-004)。

作者简介: 王 征 (2000-), 男, 硕士研究生。

引用格式: 王 征, 张大铭, 付继伟. 基于邻居节点属性的低轨卫星网络路由算法研究[J]. 计算机测量与控制, 2024, 32(11): 294-301.

行路由决策, 但其运算复杂度较高。考虑星上存储空间和处理能力有限的现实因素, 部分学者基于低信令开销和低运算复杂度设计分布式路由算法。文献 [14] 依据最小跳路由策略随机选择待转发卫星, 一定程度上改善了链路拥塞, 但卫星的转发概率无法根据负载情况进行调整, 应对流量变化的能力不足。文献 [15] 在此基础上提出一种基于邻居卫星负载状态的分布式路由算法, 根据邻居卫星的负载占用率动态分配转发概率, 缓解拥塞的效果得到提升, 但其局限性在于, 算法仅考虑最小跳路由方向的卫星, 当它们均存在较严重拥塞时, 不能提供更充分的路径选择, 难以保障较高的网络服务质量。

针对低轨卫星网络负载分布不均、传统算法路径考虑不全面的问题, 兼顾星上资源受限的特性, 本文提出一种基于邻居节点属性的路由算法 (NARA, neighbor node attribute-based routing algorithm), 依据邻居卫星的候选次序属性和综合负载属性, 考虑更多路径选择, 为不同候选卫星合理分配转发概率, 使数据分组尽可能避开拥塞区域, 以较小的信令开销实现网络性能的优化。

## 1 低轨卫星星座建模

### 1.1 星座拓扑

低轨卫星星座主要有倾斜式和近极轨式 2 种, 其中, 近极轨式星座对极地地区有更好的覆盖性, 本文选择近极轨式低轨卫星星座开展分析。星座采用 Walker-Star 构型, 卫星轨道为圆形, 各轨道平面和各轨道内的卫星均匀分布。

设共有  $M$  个轨道平面, 每个轨道平面内有  $N$  颗卫星, 卫星总数为  $L = M \cdot N$ , 每颗卫星编号为  $L_i (i = 1, 2, \dots, L)$ , 具有固定不变的逻辑地址  $(m, n)$ ,  $m$  为卫星的轨道坐标,  $n$  为卫星的轨内坐标, 其中  $1 \leq m \leq M, m \in \mathbb{Z}; 1 \leq n \leq N, n \in \mathbb{Z}$ 。卫星编号  $L_i$  与逻辑地址  $(m, n)$  的关系为:

$$\begin{cases} m = \lfloor (L_i - 1) / N \rfloor + 1 \\ n = \lceil (L_i - 1) \bmod N \rceil + 1 \end{cases} \quad (1)$$

式中, “ $\lfloor \cdot \rfloor$ ” 表示向下取整运算, “ $\bmod$ ” 表示取模运算。逻辑地址利用卫星位置的相对固定性<sup>[16]</sup>, 将星座拓扑虚拟化, 能够简化卫星相对运动的问题。

此外, 近极地轨道星座中各轨道平面升交点赤经平分  $0 \sim 180^\circ$  经度区间, 位于轨道平面 1 和  $M$  的相邻卫星运动方向相反<sup>[17]</sup>, 存在反向缝。

### 1.2 星间链路

卫星间信令和交互数据需通过星间链路<sup>[18]</sup> (ISL, inter satellite link) 实现, 能够建立 ISL 的两颗卫星互为邻居卫星。一般情况下, 每颗卫星可与相邻轨道的“左右” 2 颗卫星和相同轨道的“上下” 2 颗卫星建立星间链路, 前者为轨间 ISL, 后者为轨内 ISL, 如图 1 所示。设某卫星的逻辑地址为  $(m, n)$ , 由于反向缝的存在, 暂定  $m \neq 1$  且  $m \neq M$ 。对于轨间邻居卫星, 其“左” 邻居卫星地址为  $(m-1, n)$ , “右” 邻居

卫星地址为  $(m+1, n)$ 。对于轨内邻居卫星, 其“上” 邻居卫星地址见式 (2), “下” 邻居卫星地址见式 (3), 即卫星路由由上跳转时, 轨内坐标沿  $(1, 2, \dots, N-1, N, 1, 2, \dots)$  变化。

$$\begin{cases} (m, n+1) & 1 \leq n \leq N-1 \\ (m, 1) & n = N \end{cases} \quad (2)$$

$$\begin{cases} (m, N) & n = 1 \\ (m, n-1) & 2 \leq n \leq N \end{cases} \quad (3)$$

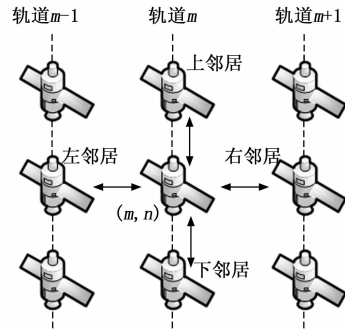
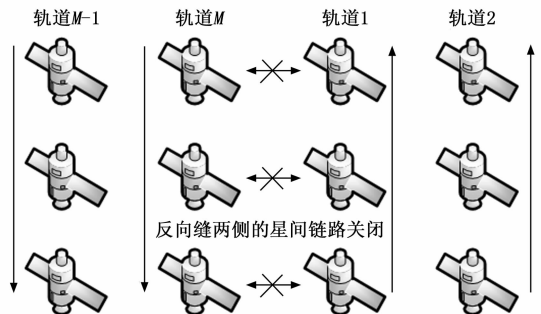
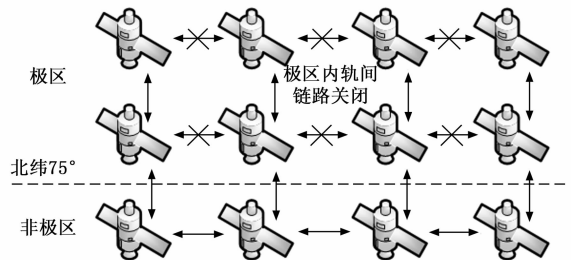


图 1 邻居卫星示意图

在低轨卫星星座中, 存在以下 2 种无法建链的情况: (1) 位于反向缝两侧的卫星运动方向相反, 天线对准困难, 星间链路关闭, 如图 2 (a) 所示; (2) 在极地区域, 不同轨道平面卫星的相对角速度增大, 难以建立稳定星间链路<sup>[19]</sup>, 但同一轨道内相邻卫星的星间链路始终保持联通, 如图 2 (b) 所示。



(a) 反向缝两侧星间链路示意图



(b) 极区内星间链路示意图

图 2 示意图

### 1.3 卫星间方向和逻辑距离

设卫星 A 地址为  $(m_A, n_A)$ , 卫星 B 地址为  $(m_B, n_B)$ , 定义

$dir_{hor}(A, B)$  为  $A$  至  $B$  的轨间方向,  $dis_{hor}(A, B)$  为  $A$  至  $B$  的轨间逻辑距离;  $dir_{ver}(A, B)$  为  $A$  至  $B$  的轨内方向,  $dis_{ver}(A, B)$  为  $A$  至  $B$  的轨内逻辑距离。

$dir_{hor}(A, B)$  和  $dis_{hor}(A, B)$  的计算式如下:

$$dir_{hor}(A, B) = sign(m_B - m_A) \quad (4)$$

$$dis_{hor}(A, B) = |m_B - m_A| \quad (5)$$

$dir_{ver}(A, B)$  和  $dis_{ver}(A, B)$  的计算式如下:

$$dis_{ver}(A, B) = \min[|n_B - n_A|, N - |n_B - n_A|] \quad (6)$$

$$dir_{ver}(A, B) = \begin{cases} 0 & n_A = n_B \\ 1 & n_B - n_A = dis_{ver}(A, B) \text{ 或} \\ & N - (n_A - n_B) = dis_{ver}(A, B) \\ -1 & \text{其他} \end{cases} \quad (7)$$

当  $|n_B - n_A| = N - |n_B - n_A|$  时, 轨内跳转方向  $dir_{ver}(A, B)$  随机取 1 或 -1。

由式 (4) ~ (7) 可知, 对于方向和逻辑距离, 轨内和轨间计算方式存在差异, 这是因为反向缝两侧的相邻卫星无法构建星间链路。

### 1.4 最小跳路由邻居卫星和路径偏离标志

#### 1.4.1 最小跳路由邻居卫星

设源卫星  $A$  地址为  $(m_A, n_A)$ , 目的卫星  $B$  地址为  $(m_B, n_B)$ ;  $A$  的邻居卫星总数为  $P$ , 各邻居卫星编号为  $nbr_i$ , 逻辑地址为  $(m_i, n_i), i = 1, 2, \dots, P$ 。当  $m_A \neq m_B$  且  $n_A \neq n_B$  时, 若  $nbr_i$  满足式 (8), 则称此邻居卫星为最小跳路由邻居卫星, 一般有 2 颗, 如图 3 所示。

$$\begin{cases} dir_{hor}(A, nbr_i) = dir_{hor}(A, B) \text{ 或} \\ dir_{ver}(A, nbr_i) = dir_{ver}(A, B) \end{cases} \quad (8)$$

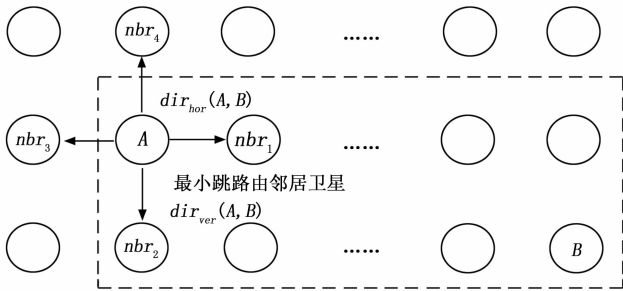


图 3 最小跳路由邻居卫星 ( $m_A \neq m_B$  且  $n_A \neq n_B$ )

当  $m_A = m_B$  或  $n_A = n_B$  时, 源卫星和目的卫星具有相同的轨道编号或轨内编号, 若  $nbr_i$  满足式 (9), 则称此邻居卫星为最小跳路由邻居卫星, 一般仅有 1 颗, 以  $m_B = n_B$  为例, 如图 4 所示。

$$\begin{cases} dir_{ver}(A, nbr_i) = dir_{ver}(A, B), m_A = m_B \\ dir_{hor}(A, nbr_i) = dir_{hor}(A, B), n_A = n_B \end{cases} \quad (9)$$

在数据包转发过程中, 若当前卫星每次都选择最小跳路由邻居卫星作为下一跳卫星, 则可以使得总路由跳数为  $dis_{hor}(src, dest) + dis_{ver}(src, dest)$ , 此时路由跳数最小, 保证路由的收敛性。

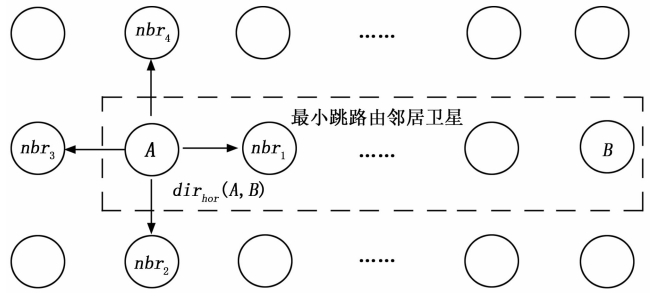


图 4 最小跳路由邻居卫星 ( $m_B = n_B$ )

#### 1.4.2 路径偏离标志

设置轨间路径偏离标志为  $out_{hor}$ , 轨内路径偏离标志为  $out_{ver}$ 。卫星在决定下一跳转发卫星时, 优先选择最小跳路由邻居卫星。若因为规避拥塞而选择非最小跳路由区域内的邻居卫星时, 认为路径已发生偏离, 根据跳转方向将相应的路径偏离标志置 1。当卫星选择最小跳路由区域内的邻居卫星时, 根据跳转方向将相应的路径偏离标志置 0。图 5 展示了路径偏离标志的设置过程。

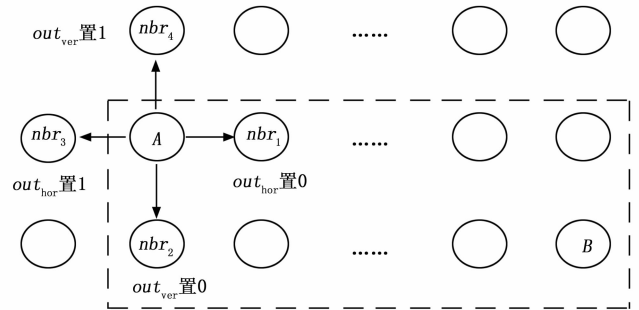


图 5 设置路径偏离标志示意图

## 2 NARA 算法

### 2.1 总体介绍

在本文提出的 NARA 算法中, 首先计算邻居卫星的候选次序属性和综合负载属性, 然后由当前卫星进行判断, 确定待转发集合并分配转发概率, 是一种动态分布式路由算法, 具体过程如下。

NARA 算法: 确定待转发卫星

输入: 当前卫星编号  $cur$ ; 目的卫星编号  $dest$ ; 当前卫星的各邻居卫星  $nbr_i (i = 1, 2, \dots, P, P$  为邻居卫星数目); 数据包中的路径集合  $list$

输出: 下一跳转发卫星  $next$

- 1: for  $i = 1, 2, \dots, P$
- 2: if  $nbr_i = dest$  then  $next = dest$ ; break;
- 3: 运行子算法 1, 确定  $nbr_i$  是否可作为候选卫星;
- 4: if  $nbr_i$  为候选卫星 then
- 5: 运行子算法 2, 计算综合负载占用率;
- 6: else continue;
- 7: end for

- 8: 运行子算法 3, 确定各候选卫星的转发概率
  - 9: 生成随机数与转发概率比较, 确定下一跳卫星 *next*
  - 10: 根据转发方向设置相应路径偏离标志, 算法结束
- 其中: 3 个子算法功能描述如表 1 所示。

表 1 子算法功能描述

子算法	功能
子算法 1	根据邻居卫星的地址, 结合其他信息, 将其划分为不可达、第一候选、第二候选 3 类, 作为此邻居卫星的候选次序属性。
子算法 2	排除不可达邻居卫星后, 为处于第一候选和第二候选的邻居卫星计算综合负载占用率, 作为此邻居卫星的综合负载属性。
子算法 3	根据子算法 1 和子算法 2 的输出结果, 确定待转发卫星集合, 并分配转发概率。

### 2.2 子算法 1-确定邻居卫星候选次序

设当前卫星编号为 *cur*, 逻辑地址为  $(m_{cur}, n_{cur})$ , 目的卫星编号为 *dest*, 逻辑地址为  $(m_{dest}, n_{dest})$ , 邻居卫星总数为 *P*, 各邻居卫星编号为 *nbr<sub>i</sub>*, 逻辑地址为  $(m_i, n_i), i = 1, 2, \dots, P$ 。

邻居卫星为不可达状态的条件如下: (1) 当前卫星和邻居卫星不在同一轨道 (即  $m_{cur} \neq m_i$ ), 且邻居卫星处于极区; (2) 邻居卫星已存在于路径集合中, 为避免路由回环, 将其设置为不可达; (3) 若路径偏离标志已置 1, 表明此前为规避拥塞将数据包路由至非最小跳路由区域, 为避免路由发散现象, 在当前选择中, 将非最小跳路由邻居卫星设置为不可达, 如图 6 所示。

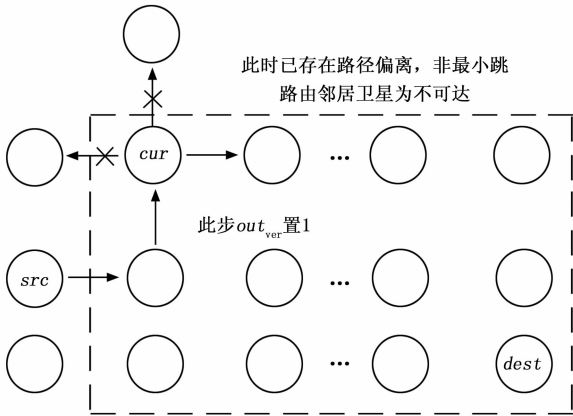


图 6 存在路径偏离时不可达卫星示意图

在筛选不可达卫星结束后, 将最小跳路由邻居卫星划分到第一候选卫星集合中, 其余卫星划分到第二候选卫星集合中。图 7 展示了不存在路径偏离的条件下的邻居卫星候选次序。

在前述分析的基础上, 将子算法 1 总结如下。

子算法 1: 确定邻居卫星候选次序

输入: 当前卫星编号 *cur* 及地址  $(m_{cur}, n_{cur})$ ; 目的卫星编号 *dest* 及地址  $(m_{dest}, n_{dest})$ ; 各邻居卫星编号 *nbr<sub>i</sub>*、纬度 *lat<sub>i</sub>* 及地址  $(m_i, n_i)(i$

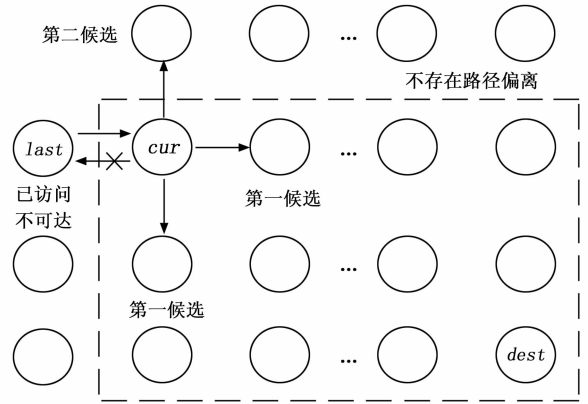


图 7 卫星候选次序示意图

$= 1, 2, \dots, P, P$  为邻居卫星总数); 当前路由表中路径集合 *list*

输出: 各邻居卫星 *nbr<sub>i</sub>* 的候选次序

1: for  $i = 1, 2, \dots, P$

2: if *nbr<sub>i</sub>* 满足不可达条件 then

3: 此邻居卫星 *nbr<sub>i</sub>* 为不可达;

4: else 计算  $dir_{hor}(cur, dest), dir_{hor}(cur, nbr_i), dir_{ver}(cur, dest), dir_{ver}(cur, nbr_i)$

5: if *nbr<sub>i</sub>* 为最小跳路由邻居卫星 then

6: 此邻居卫星 *nbr<sub>i</sub>* 为第一候选;

7: else *nbr<sub>i</sub>* 为非最小跳路由邻居卫星

8: 此邻居卫星 *nbr<sub>i</sub>* 为第二候选;

9: end for

### 2.3 子算法 2-计算候选卫星综合负载占用率

在地面路由算法中, 通常采取泛洪方式收集网络状态信息, 计算最优转发路径。然而, 低轨卫星星座规模较大且处理能力有限, 泛洪将产生大量开销, 实际意义较小<sup>[20]</sup>。因此, 本文根据候选卫星和其邻居卫星的负载情况计算综合负载占用率, 以较小的信令分组开销反映候选卫星的拥塞程度。

单颗卫星的负载占用率为  $\frac{q_i}{Q_i}$ , 其中  $q_i$  为卫星缓存队列中待处理的数据包数量,  $Q_i$  为缓存队列数据包容量。每颗卫星周期性地将其自身负载占用率和已掌握的邻居卫星负载占用率写入星间 Hello 数据包, 通过星间链路发送给邻居卫星, 实现负载状态信息的局部交互。因此在进行路由决策前, 每颗卫星可获知候选卫星及其邻居卫星的负载占用率。

设当前卫星编号为 *cur*; 候选卫星编号为 *cand*, 负载占用率为  $q_c$ ; 候选卫星的邻居卫星总数为 *P*, 编号为 *nbr<sub>i</sub>*, 负载占用率为  $q_m, i = 1, 2, \dots, P$ ; 待计算的候选卫星综合负载占用率为  $q$ , 用来反映此候选卫星的拥塞程度。

下面确定能够参与综合负载占用率计算的卫星。在当前卫星下, 若选择候选卫星作为转发卫星, 则下一跳转发卫星为候选卫星的某颗邻居卫星。然而, 根据子算法 1, 候选卫星的部分邻居卫星为不可达状态, 因此这些邻居卫星的负载占用率没有参考意义, 不应参与到  $q$  的计算中。以

图 8 为例进行说明, 在此前的路由中, 因轨内路径偏离而使得  $out_{ver}$  置 1, 若沿轨间方向选择  $cand$  作为转发卫星, 仍不能使  $out_{ver}$  置 0, 因此在下一跳路由决策中, 非最小跳路由由邻居卫星为不可达, 只有 2 颗最小跳路由由邻居卫星可以参与到综合负载占用率计算中。

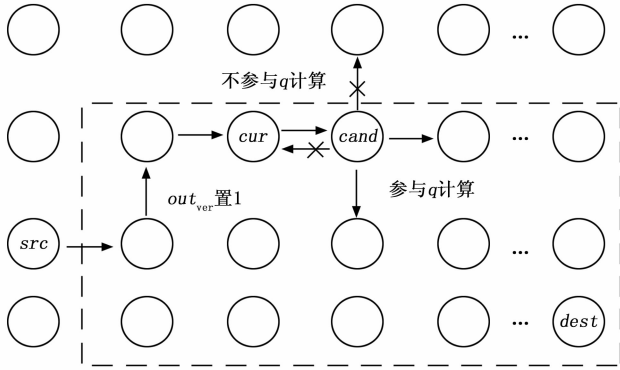


图 8 参与综合负载占用率计算的邻居卫星

遍历候选卫星的邻居卫星  $nbr_i$ , 运行子算法 1, 设能够参与综合负载占用率计算的邻居卫星总数为  $R$ , 对应的负载占用率为  $q_n, n = 1, 2, \dots, R$ 。则综合负载占用率的计算式如下:

$$q = 0.5 \cdot q_c + 0.5 \cdot \frac{\sum_{n=1}^R q_n}{R} \quad (10)$$

式中, 候选卫星及其邻居卫星的平均负载占用率的权重均为 0.5, 此权重选取的合理性和有效性已在文献 [15] 中体现, 候选卫星自身负载占用率反映选择未来第 1 跳卫星的代价, 其邻居卫星的平均负载占用率反映选择未来第 2 跳卫星的代价, 在进行路由决策时两者应具有同等地位, 这样可以使得  $q$  能够更全面地反映当前候选卫星的拥塞程度。

此外, 当候选卫星的邻居卫星包含目的卫星时, 如果选择此候选卫星, 则下一跳卫星可直接选择目的卫星, 因此将候选卫星自身的负载占用率作为其综合负载占用率, 即  $q = q_c$ 。

在前述分析的基础上, 将子算法 2 总结如下。

子算法 2: 计算候选卫星综合负载占用率

输入: 当前卫星编号  $cur$ ; 候选卫星编号  $cand$  及其负载占用率  $q_c$ ; 候选卫星的邻居卫星  $nbr_i (i = 1, 2, \dots, P, P$  为邻居卫星总数) 及其负载占用率  $q_n$ ; 当前路由表卫星集合  $list$

输出: 候选卫星的综合负载占用率  $q$

- 1: 初始化  $R = 0$ , 记录候选卫星的下一跳候选卫星数目
- 2: for  $i = 1, 2, \dots, P$
- 3: if  $nbr_i$  为目的卫星 then
- 4:  $q = q_c$ , 程序结束;
- 5: else
- 6: 将  $cand$  作为本次转发卫星, 运行子算法 1, 判断  $nbr_i$  是否为下一跳候选卫星;
- 7: if  $nbr_i$  可作为候选卫星 then

8: 记录其负载占用率  $q_n$ ;  $R = R + 1$ ;

9: else continue;

10: end for

11: 按照式(10)计算综合负载占用率  $q$ , 程序结束

### 2.4 子算法 3-转发概率分配

设经过子算法 1 和子算法 2 计算后, 第一候选卫星总数为  $S_1$ , 各卫星编号为  $S_{1i} (i = 1, 2, \dots, S_1)$ , 综合负载占用率为  $q_{1i} (i = 1, 2, \dots, S_1)$ ; 第二候选卫星总数为  $S_2$ , 各卫星编号为  $S_{2i} (i = 1, 2, \dots, S_2)$ , 综合负载占用率为  $q_{2i} (i = 1, 2, \dots, S_2)$ 。下面确定待转发的卫星。在本算法中, 第一候选卫星总是处于待转发集合内, 根据负载情况判断第二候选卫星是否应当加入待转发集合。

对于每一颗第二候选卫星  $S_{2i}$ , 若满足如下条件

$$\min_{1 \leq j \leq S_1} q_{1j} > q_{2i} + thre \quad (11)$$

则将  $S_{2i}$  加入待转发集合, 否则不予加入,  $thre$  为阈值。由上式可知, 相对于第一候选卫星, 若第二候选卫星的负载较轻, 应当选择第二候选卫星进行迂回路由, 从而降低排队时延开销。

下面分析  $thre$  的选取。当  $q_{1i}$  均高于 0.9 时, 第一候选卫星拥塞严重, 丢包风险较高, 此时阈值取 0, 保证及时将数据包路由至第二候选卫星。在其他情况下, 第一候选卫星的拥塞情况相对不极端, 若阈值过小, 大量数据包将进行迂回路由, 严重增加路由跳数, 更多卫星参与路由处理也会增加计算资源的消耗; 若阈值过大, 数据包无法及时路由至负载更轻的第二候选卫星, 无法充分发挥其缓解拥塞的作用; 综合考虑上述因素, 将阈值选取为 0.3。

遍历第二候选卫星, 得到待转发集合。设集合内卫星总数为  $S$ , 各卫星编号为  $S_i$ , 综合负载占用率为  $q_i, i = 1, 2, \dots, S$ , 则卫星  $S_i$  的转发概率  $p_i$  为:

$$p_i = \frac{1 - q_i}{\sum_{j=1}^S (1 - q_j)} \quad (12)$$

式中,  $1 - q_i$  表示该卫星空闲程度, 转发概率按照各卫星的空闲程度进行比例分配。若卫星空闲程度越低, 相应的转发概率也越低, 在  $q_i$  接近 1 的极端拥塞情况下, 转发概率将趋近于 0。转发概率的分配可以实现局部负载均衡, 提升网络服务性能。

在前述分析的基础上, 将子算法 3 总结如下。

子算法 3: 转发概率分配

输入: 第一候选卫星总数  $S_1$ , 编号  $S_{1i}$  及综合负载占用率  $q_{1i} (i = 1, 2, \dots, S_1)$ ; 第二候选卫星总数  $S_2$ , 编号  $S_{2i}$  及综合负载占用率  $q_{2i} (i = 1, 2, \dots, S_2)$ ;

输出: 待转发卫星集合内每颗卫星的转发概率

- 1: 将全部第一候选卫星加入待转发集合
- 2: for  $i = 1, 2, \dots, S_2$
- 3: 根据式(12)计算阈值  $thre$ ;
- 4: if  $q_{2i}$  满足式(11) then
- 5: 将  $S_{2i}$  加入待转发集合, 并记录  $q_{2i}$ ;

6: else continue;  
 7: end for  
 8: 获取待发转发集合, 卫星总数为  $S$ , 各卫星编号为  $S_i$  ( $i = 1, 2, \dots, S$ ), 综合负载占用率为  $q_i$   
 9: 对每一颗待发转发卫星  $S_i$ , 根据式(13)计算转发概率  $p_i$ , 程序结束

### 3 仿真建模及结果分析

#### 3.1 仿真建模

本文采用 OPNET 作为仿真工具, 通过网络域、节点域、进程域建模实现拓扑构建和协议设计<sup>[21]</sup>。选用的 Walker-Star 星座共有 8 个轨道面, 每个轨道面内有 13 颗卫星, 相位因子为 0, 轨道倾角为  $85^\circ$ , 高度为 800 km。在 STK 中建立此星座, 如图 9 所示。

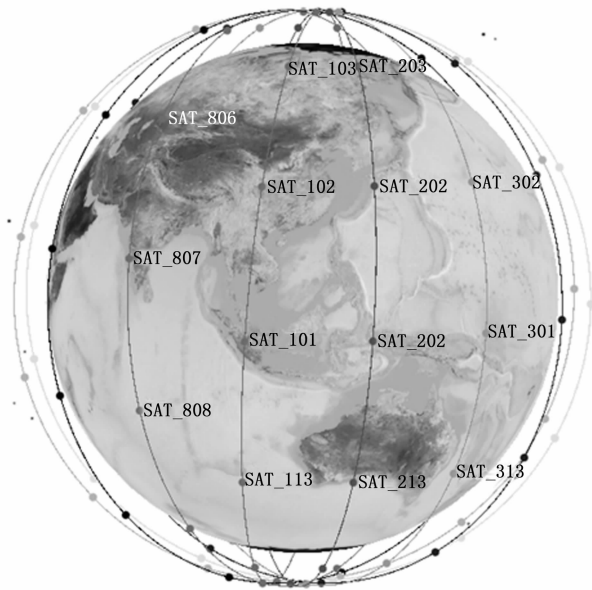


图 9 STK 中的低轨卫星星座

本文以远海目标探测为背景, 海面节点将目标探测数据以固定速率传输至地面。海上发射节点和地面接收节点分别接入距离最近的卫星作为源卫星和目的卫星, 如图 10 所示。数据包发送速率为 2 Mbps, 单个数据包大小为 5 kB, 卫星处理速率为 5 Mbit/s, 在处理过程中根据路由算法得到下一跳转发卫星。

为了模拟流量分布, 为每颗卫星配置数据源, 发包间隔服从参数为  $\lambda$  的指数分布, 单个数据包为 5 kB。当卫星接收到此数据源的数据包时, 同样以 5 Mbit/s 的速率进行处理, 处理结束后将其销毁。通过改变  $\lambda$  的值, 可以模拟卫星网络出现不同程度的拥塞, 数据源平均流量生成速率可由  $\lambda$  推算得到。

此外, 基于卫星存储空间受限的实际情况, 本文为每颗卫星配置容量为 100 的缓存队列<sup>[22]</sup>。在相同流量分布下, 当缓存队列容量增加时, 因队列溢出而产生丢包的风险降低, 抵御严重拥塞的能力更为突出, 但同时提高了对星上

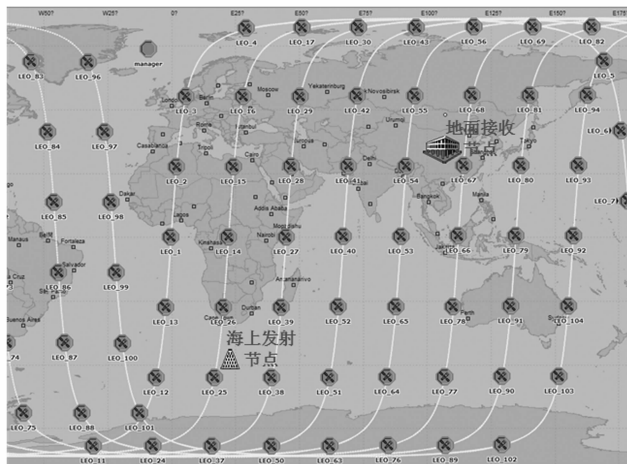


图 10 OPNET 网络拓扑图

存储设备的要求。

本文仿真参数设置总结如下。

表 2 仿真参数设置

仿真参数	参数值
星座参数	(104/8/0, 800 km, $85^\circ$ )
单个数据包大小	5 kB
单星处理速率	5 Mbit/s
卫星缓存队列容量	100
仿真时长	2 min

#### 3.2 结果分析

本文选择 DRNL 算法<sup>[15]</sup>进行对比, NARA 算法的改进如下: 在 DRNL 算法中, 在当前卫星具有 2 颗最小跳路由邻居卫星且均较为拥塞时, 无法提供更多路由选择; NARA 算法额外考虑第二候选卫星, 当第一候选卫星均出现拥塞且第二候选卫星负载较轻时, 及时将部分数据包分流至第二候选卫星。

如图 11 所示, 在仿真时间内, 海面发射节点接入卫星 25, 地面接收节点接入卫星 67。为了产生不同卫星拥塞程度的差异, 体现算法性能, 随机选择卫星 27、40、41、52、65, 设置其平均流量生成速率在 2 Mbps~6 Mbps 间变化, 其他卫星的平均流量生成速率固定为 2 Mbps。随着以上 5 颗卫星流量生成速率的增加, 相应的综合负载占用率逐渐升高。

在此背景下, 以远海目标探测数据传输至地面为研究对象, 选取平均时延、丢包率和平均路由跳数作为评价算法性能的指标, 改变 5 颗卫星数据源流量生成速率, 探讨网络性能的变化。

##### 3.2.1 平均时延

平均时延随 5 颗卫星平均流量生成速率的变化如图 12 所示。当流量生成速率不超过 4 Mbps 时, 网络较为空闲, 各卫星间综合负载占用率相差较小, 两种算法的平均时延接近。从 4.5 Mbps 开始, 随着流量生成速率的增加, NA-

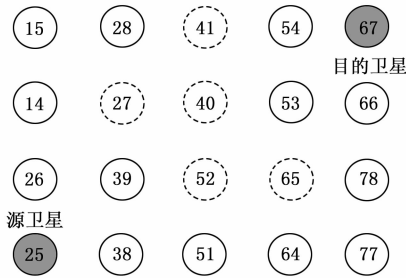


图 11 仿真时间内卫星拓扑图

RA 算法的时延始终低于 DRNL 算法，这是因为 NARA 算法根据邻居卫星的拥塞程度，考虑更多的路径选择，避开拥塞区域。当流量生成速率大于 5 Mbps 时，时延增幅逐渐减小，此时卫星的缓存队列溢出概率显著增大，DRNL 算法仅在第一候选卫星内进行选择，缓存队列溢出会产生大量丢包现象，这些丢包不计入时延统计中，从而使得时延增加缓慢，而 NARA 算法通过合理分配转发概率，将部分数据迂回路由至第二候选卫星，且第一候选卫星的负载越重，相应的转发概率越低，实现局部负载均衡，从而保证时延的相对稳定。

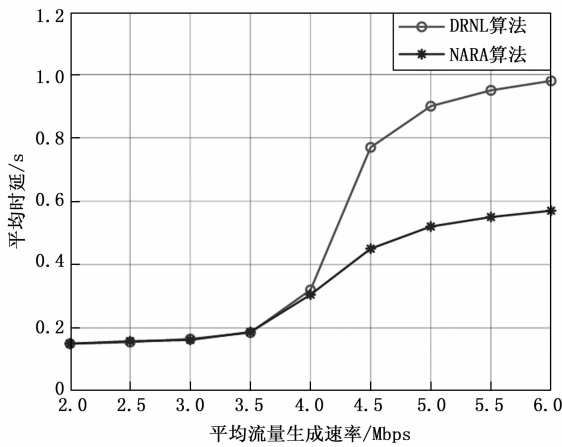


图 12 平均时延随平均流量生成速率的变化

例如，当数据包沿“25→38→39”路由至卫星 39 时，第一候选卫星 27 和 52 的综合负载占用率均较高，第二候选卫星 26 的负载较轻，NARA 算法会将部分数据包路由至卫星 26，如图 13 所示。

### 3.2.2 丢包率

丢包率随 5 颗卫星平均流量生成速率的变化如图 14 所示，主要由卫星缓存队列溢出、数据包碰撞等引起。当流量生成速率不超过 4 Mbps 时，两种算法的丢包率均接近于 0。随着流量生成速率的增加，NARA 算法的丢包率显著低于 DRNL 算法，在 6 Mbps 下，DRNL 算法的丢包率高达 9.6%，而 NARA 算法仅为 2.3%。NARA 算法能够在第一候选卫星拥塞较为严重时，将部分数据转发至第二候选卫星，从而降低选择第一候选卫星带来的丢包风险。

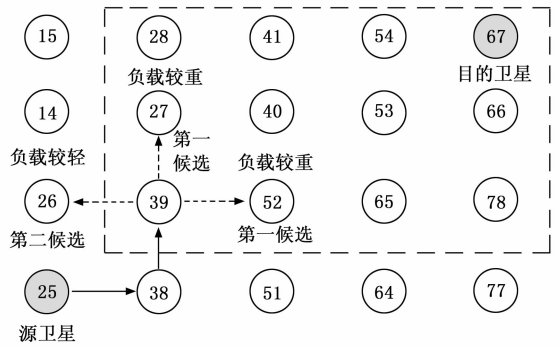


图 13 NARA 算法迂回路由示意图

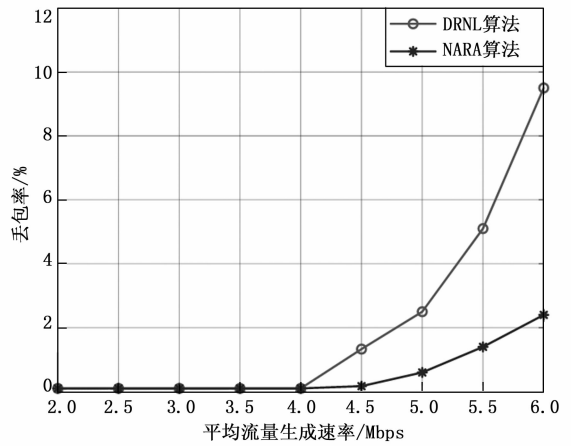


图 14 丢包率随平均流量生成速率的变化

### 3.2.3 平均路由跳数

平均路由跳数随 5 颗卫星平均流量生成速率的变化如图 15 所示。当流量生成速率较低时，两种算法的平均路由跳数接近，NARA 算法未进行额外的迂回路由。随着流量生成速率的增加，NARA 算法为规避拥塞会有一定概率选择第二候选卫星，因此平均路由跳数会高于 DRNL 算法。例如，当数据包路由至卫星 27 时，若选择第二候选卫星 26

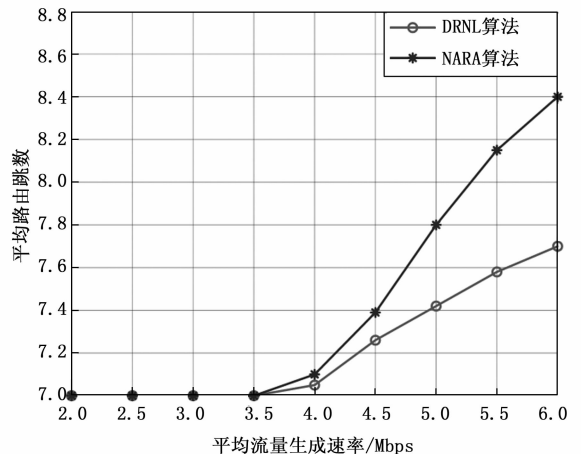


图 15 平均路由跳数随平均流量生成速率的变化

作为下一跳节点, 则至少增加 2 跳路由。

由于子算法 1 的限制, NARA 算法能够在避开拥塞的同时, 有效保证路由收敛性, 使得平均路由跳数增加有限。相比于平均时延和丢包率的大幅降低, 路由跳数增加的代价在可接受范围内。

#### 4 结束语

低轨卫星网络具有发射成本低、传输时延小的优势, 在远海目标探测数据回传中发挥重要作用。传统的低轨卫星网络路由算法应对拥塞的能力不足, 路径选择存在局限性。在本文提出的 NARA 算法中, 根据各邻居卫星的候选次序属性和综合负载属性, 在第一候选卫星拥塞较为严重时, 通过合理分配转发概率, 将部分数据分流至负载较轻的第二候选卫星, 从而提升网络服务质量。此外, NARA 算法将星座拓扑虚拟化, 为每颗卫星分配逻辑地址, 适用于其他近极轨星座, 具备较高的拓展性。仿真结果表明, NARA 算法降低了数据传输的平均时延和丢包率, 能够保障探测数据的高效可靠传输。

#### 参考文献:

[1] WANG D, SUN C, LIU L, et al. 5G integrated radio transmission scheme for low earth orbit satellite access network [C] // 2022 IEEE 22nd International Conference on Communication Technology (ICCT). Nanjing, China; IEEE, 2022: 1417-1420.

[2] 杨华果, 陈全, 杨自鹏, 等. 低轨巨型星座网络容量评估与分析 [J]. 宇航总体技术, 2023, 7 (6): 24-30.

[3] CAO J, ZHANG S, CHEN Q, et al. Computing-aware routing for LEO satellite networks: a transmission and computation integration approach [J]. IEEE Transactions on Vehicular Technology, 2023, 72 (12): 16607-16623.

[4] ZHU J, SUN X, CHEN D. Deep reinforcement learning for congestion control and routing optimization in LEO satellite networks [C] // 2023 5th International Conference on Artificial Intelligence and Computer Applications (ICAICA). Dalian, China; IEEE, 2023: 739-742.

[5] ZHANG N, NA Z, TAO J, et al. Time-varying graph and binary tree search based routing algorithm for LEO satellite networks [J]. IEEE Transactions on Vehicular Technology, 2023, 72 (10): 13683-13688.

[6] WANG F, JIANG D, WANG Z, et al. Fuzzy-CNN based multi-task routing for integrated satellite-terrestrial networks [J]. IEEE Transactions on Vehicular Technology, 2022, 71 (2): 1913-1926.

[7] LI S, WANG F, YAO H, et al. A multi-region division routing algorithm based on Fuzzy-Shortest-Path-First for LEO satellite networks [C] // 2023 International Wireless Communica-

tions and Mobile Computing (IWCMC). Marrakesh, Morocco; IEEE, 2023: 800-805.

[8] 朱立东, 张勇, 贾高一. 卫星互联网路由技术现状及展望 [J]. 通信学报, 2021, 42 (8): 33-42.

[9] 刘洵, 谢金森, 陈双武. 链路状态感知的低轨卫星网络路由机制 [J]. 宇航总体技术, 2020, 4 (2): 33-40.

[10] 顿聪颖, 金凤林, 谭诗翰, 等. 基于负载均衡的低轨卫星网络动态路由算法 [J]. 无线电工程, 2022, 52 (5): 790-798.

[11] FU L, WEI W, LU X, et al. Reinforcement learning based intelligent routing for software defined LEO satellite networks [C] // GLOBECOM 2023-2023 IEEE Global Communications Conference. Kuala Lumpur, Malaysia; IEEE, 2023: 2323-2328.

[12] WANG X, DAI Z, XU Z. LEO satellite network routing algorithm based on reinforcement learning [C] // 2021 IEEE 4th International Conference on Electronics Technology (ICET). Chengdu, China; IEEE, 2021: 1105-1109.

[13] 刘洋, 王丽娜. 基于树突神经网络的低轨卫星智能感知路由算法 [J]. 工程科学学报, 2023, 45 (3): 465-474.

[14] 冯玺宝. 多层卫星网络 QoS 路由协议研究 [D]. 哈尔滨: 哈尔滨工业大学, 2016.

[15] 杨明川, 薛冠昌, 李清毅. 基于邻居卫星负载状态的低轨卫星分布式路由算法 [J]. 通信学报, 2021, 42 (8): 43-51.

[16] 童文婷, 黄圣春, 王玲. 基于虚拟拓扑的低轨卫星路由切换算法仿真 [J]. 计算机仿真, 2021, 38 (9): 31-34.

[17] LIU J, LUO R, HUANG T, et al. A load balancing routing strategy for LEO satellite network [J]. IEEE Access, 2020, 8: 155136-155144.

[18] TANG F, ZHANG H, YANG L T. Multipath cooperative routing with efficient acknowledgement for LEO satellite networks [J]. IEEE Transactions on Mobile Computing, 2019, 18 (1): 179-192.

[19] 张路, 燕锋, 章跃跃, 等. 基于星间链路状态的低轨卫星网络路由算法 [J]. 上海航天 (中英文), 2021, 38 (4): 92-100.

[20] 汪伊婕, 赵伟, 成飞, 等. 基于负载均衡的大规模低轨卫星互联网路由算法 [J]. 天地一体化信息网络, 2022, 3 (1): 27-34.

[21] ADDAY G H, SUBRAMANIAM S K, ZUKARNAIN Z A, et al. Investigating and analyzing simulation tools of wireless sensor networks: a comprehensive survey [J]. IEEE Access, 2024, 12: 22938-22977.

[22] 韩驰, 熊伟, 于荣欢, 等. 基于跳数背压策略的低轨卫星网络负载均衡路由 [J]. 系统工程与电子技术, 2023: 1-13. [2024-08-18]. <https://link.cnki.net/urlid/11.2422.TN.20231218.1557.002>.