

# 基于邻居卫星负载状态的低轨卫星分布式路由算法

杨明川<sup>1</sup>, 薛冠昌<sup>1</sup>, 李清毅<sup>2</sup>

(1. 哈尔滨工业大学信息与通信工程学院, 黑龙江 哈尔滨 150001; 2. 北京空间飞行器总体设计部, 北京 100086)

**摘 要:** 卫星灵活性和机动性强、不易受地面因素的影响, 使空间卫星技术不断飞速发展。卫星网络随着负载的增加始终存在负载拥塞现象, 使路由算法成为卫星网络研究领域的核心问题。基于卫星网络的拥塞缓解、网络的信令开销以及算法运算复杂度问题, 深入研究低轨卫星网络局部负载均衡的分布式路由算法。构建基于铱星系统的网络模型, 提出一种基于邻居卫星负载状态的分布式路由算法 (DRNL)。DRNL 包含邻居卫星负载状态更新、负载均衡和路由判决等机制, 具有一定的可移植性, 能够适应其他极轨或近极轨星座。OPNET 仿真结果表明, DRNL 在较重的网络负载情况下能较好地适应负载的变化, 缓解卫星拥塞的状况并获得较低的时延和数据分组丢失率。

**关键词:** 负载均衡; 低地球轨道; 分布式路由; DRNL 算法

**中图分类号:** TN927

**文献标识码:** A

**DOI:** 10.11959/j.issn.1000-436x.2021165

## Neighbor satellite load based low orbit satellite distributed routing algorithm

YANG Mingchuan<sup>1</sup>, XUE Guanchang<sup>1</sup>, LI Qingyi<sup>2</sup>

1. College of Information and Communication Engineering, Harbin Institute of Technology, Harbin 150001, China

2. Beijing Institute of Spacecraft Systems Engineering, Beijing 100086, China

**Abstract:** Satellites are highly flexible and maneuverable, and they are not easily affected by ground factors, which has led to the rapid development of space satellite technology. Satellite networks always have load congestion problem as the load increases, making routing algorithms a core issue in the field of satellite network research. Based on the problems of congestion mitigation, signaling overhead, and algorithm complexity, the distributed routing algorithm for local load balancing of LEO (low earth orbit) satellite networks was deeply studied. The network model was constructed based on the Iridium satellite system and a distributed routing algorithm based on the load status of neighbor satellite (DRNL) was proposed, which included such as load status update of neighbor nodes, load balancing and routing decision. It had a certain degree of portability and could adapt to other polar or near polar orbit constellation. The simulation results based on the OPNET show that the DRNL can better adapt to the heavier network loads state, alleviate satellite congestion, obtain lower delay and packet loss.

**Keywords:** load balancing, LEO, distributed routing, DRNL algorithm

## 1 引言

随着通信技术的进步、通信业务的增长和业务种类的增多, 未来的通信网络将向着地面网络和卫

星网络融合的方向发展, 低轨卫星网络在星地互联网络中扮演着重要的角色<sup>[1]</sup>。目前, 移动通信系统已经发展到了 5G 时代, 通信终端数量也发生了爆炸式的增长<sup>[2]</sup>。卫星通信成为地面移动通信系统和

收稿日期: 2021-03-10; 修回日期: 2021-07-23

基金项目: 国家自然科学基金资助项目 (No.62071146)

**Foundation Item:** The National Natural Science Foundation of China (No.62071146)

固定通信网络的延伸,且由于低轨星座能够实现全球通信的无缝覆盖,其作用和战略价值非常重要。

低地球轨道(LEO, low earth orbit)卫星的传输损耗小,能够支持小型移动终端之间的通信,便于实现终端小型化和多样化,所以应用价值很高<sup>[3]</sup>。LEO卫星的运行轨道高度比较低,传播时延比较短,通常在10 ms左右,能够通过星间链路(ISL, inter satellite link)和相邻卫星建立全双工通信,实现对数据进行路由转发。虽然星间链路的建立使星座系统的设计复杂度增加,但不需要在全局范围内建立大量的信关站,也降低了成本。

随着业务量增加,星座网络会出现拥塞现象。为缓解拥塞问题,Liu等<sup>[4]</sup>提出LEO卫星网络基于分段路由的负载均衡算法,动态划分轻负载区和重负载区。轻载区采用预平衡最短路径算法,重载区采用最小权重路径算法。分段路由算法在吞吐量、链路利用率和平均时延方面得到提高,但是采用分段路由算法造成星上存储空间增加且算法复杂度较高。冯玺宝<sup>[5]</sup>提出缓解拥塞状态的最小跳路由算法,根据卫星之间的经纬度信息计算出2条备选路径,根据网络负载情况判决出最优路径。算法的信令开销和运算复杂度较低,并且能在一定程度上缓解网络拥塞现象,但路由表由固定节点计算,导致网络的实时响应性差。

Geng等<sup>[6]</sup>提出LEO卫星通信网络的最优时延路由算法,对卫星网络的时延变化进行表征,并选择有效的候选路径。算法得出路径的时延及其变化的时延评估指标,以网络中信令开销增加为代价,实时响应网络的拥塞情况。Wang等<sup>[7]</sup>提出了基于拥塞预测的负载均衡路由算法,建立了多目标优化模型。采用修正因子来调整路径成本,通过拥塞预测来预测星间链路的拥塞。利用蚁群算法求解该模型,从而为每个连接请求找到最佳路径,运用神经网络以运算复杂度的提高为代价,实现卫星网络的高效负载均衡。

当前卫星网络面对拥塞缓解问题,路由算法往往以网络中信令分组的增加为代价,获取全网卫星的负载状态信息,或者通过神经网络等算法及时进行流量预测来提前缓解拥塞,增加了运算的复杂度以及星上存储空间。针对算法复杂度和信令开销,基于小卫星星座的背景,本文算法设计考虑以较小的运算复杂度和信令开销获得时延和吞吐量性能的提升。

## 2 LEO星座的建模

铱星(Iridium)系统是由66颗LEO卫星交叉连接组成覆盖全球范围的复杂全网状卫星网络,数据在卫星之间进行路由而不经地面。铱星无缝的语音和数据全球覆盖,能实现真正的移动通信,并且系统中卫星数量多,星间链路的互通能够满足未来物联网信息的接入需求<sup>[8]</sup>。本文选用Iridium系统进行网络建模,采用软件STK 11.2进行卫星轨道的参数建模和OPNET 14.5进行卫星网络路由算法的仿真分析。

Iridium系统的卫星运行轨道均采用顺时针方向。在STK11.2中进行卫星轨道建模时<sup>[9]</sup>,设置卫星的编号规则为采用三位数字表示。卫星命名的第10位表示卫星的轨道编号,第12和13位表示卫星的轨内编号<sup>[10]</sup>。卫星进入南北极圈后关闭相邻轨道间的链路(轨间链路),极圈边界纬度为70°。每颗卫星包含一个单波束卫星天线,天线圆锥半角为62°,即可实现全球无缝覆盖。由于Iridium系统轨道1和轨道6相邻,相邻轨道的卫星运动方向相反(此区域称为反向缝),卫星节点之间相对运行速度很快,天线难以及时调整且通信时间比较短,故取消反向缝间卫星的轨间链路。Iridium系统仿真参数如表1所示,3D建模如图1所示。

表 1 Iridium 系统仿真参数

仿真参数	参数设置
轨道面数	6
轨内卫星数量/颗	11
轨道高度/km	780
极区边界纬度	70°
星间链路	4
轨道倾角	87°
反向缝	22°
相邻轨道间经度差	31.6°
相邻轨道间相位差	16.5°

从图1可以看出,Iridium系统实现了全球无缝覆盖。设置地面卫星的通信仰角为5°,随机选取北京市测量任意时刻可接入卫星的数量,仿真结果如图2所示。横坐标为仿真时间,纵坐标为北京市可接入卫星数量,仿真运行24 h,从2020年3月16日04:00到3月17日04:00。

在任意时刻地面终端可选接入卫星数量始终大于或等于1,这意味着地面终端可以采取最长接

入时间覆盖、最大地面仰角、最大接入信号功率等多种星地链路的选择方案。

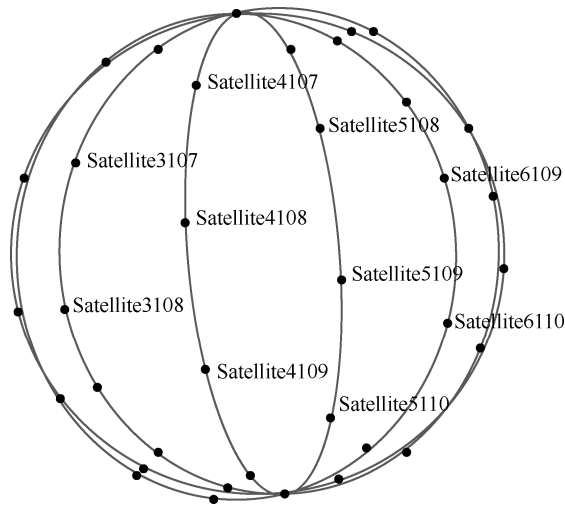


图 1 铱星系统 3D 建模

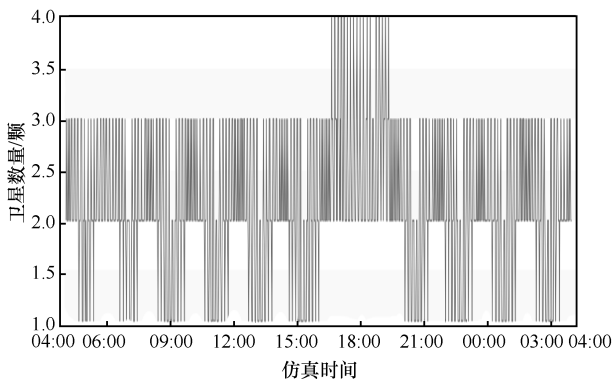


图 2 北京市可接入卫星数量

### 3 DRNL

考虑卫星网络中较低的信令开销以及算法的运算复杂度，基于卫星网络的拥塞缓解问题，本文提出一种基于邻居卫星负载状态的分布式路由算法（DRNL, distributed routing algorithm based on the load status of neighbor satellite），充分考虑了卫星移动的规律性，将星座网络中的每颗卫星看成虚拟节点<sup>[11-12]</sup>。卫星采用双向通信，通过无线收发信机实现数据发送和接收<sup>[13]</sup>。DRNL 采用虚拟节点的思想，将地球表面划分为 66 个逻辑区域，距离逻辑区域中心最近的卫星加载其逻辑地址。每颗卫星节点能够独立进行分布式的路由决策，且卫星的轨道采用近极轨道，轨道倾角为 87°。根据虚拟节点的思想，该策略能够简化卫星之间的相对运动问题，适用于其他近极轨道星座<sup>[14]</sup>。

通过卫星网络的拓扑结构可知，卫星不通过极区时，非反向缝两侧卫星存在 4 条星间链路，且每对卫星节点之间包括发送链路和接收链路 2 个方向，如图 3 所示。

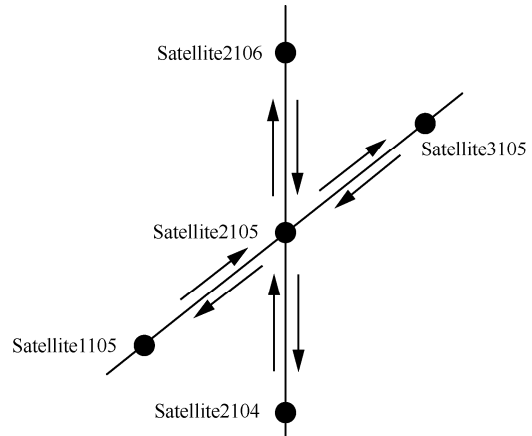


图 3 卫星节点链路拓扑结构

DRNL 中每颗卫星独立地进行路由选择，当前卫星节点在星座中的编号为  $\langle k_s, l_s \rangle$ ，目的卫星节点在星座中的编号为  $\langle k_d, l_d \rangle$ ，其中  $k$  代表轨道编号， $l$  代表轨内编号。通过卫星之间命名的关系（相对位置），判断出可选路由下一跳卫星节点，数据分组的可选节点路由策略如表 2 所示。其中  $\Delta k = k_d - k_s$ ， $\Delta l = l_d - l_s$ ，图中指向箭头表示可选节点路由的方向。

表 2	待选下一跳路由策略		
$\Delta l$	$\Delta k < 0$	$\Delta k = 0$	$\Delta k > 0$
$0 < \Delta l \leq 5$ 或 $-10 < \Delta l \leq -5$			
$\Delta l = 0$		同一卫星节点	
$5 < \Delta l < 10$ 或 $-5 \leq \Delta l < 0$			

数据分组到达卫星节点时，根据分组域中的目的地址和当前节点的编号采用 DRNL 获取可选下一节点。如果当前节点与可选节点的星间链路处于连接状态，则依据负载判决机制选择合适的下一节点。如果与可选下一节点的轨间链路断开，则依据迂回路由算法选择其余的待选路径。待选路径的卫星一般为当前卫星节点轨道面内的上、下 2 颗卫星，依据邻居卫星的纬度和负载状态选择合适的下一跳进行路由转发。

卫星节点间的链路负载情况<sup>[10]</sup>的计算式如式(1)所示。

$$q_{B_1} = \frac{q_{b_1}}{Q_{b_1}}$$

$$q_B = \frac{q_{B_1} + q_{B_2} + q_{B_3} + q_{B_4}}{4}$$

$$q_{AB} = 0.5q_{ab} + 0.5q_B \quad (1)$$

其中,  $q_{b_i}$  为缓存队列长度,  $Q_{b_i}$  为缓存队列容量,  $q_{B_1} \sim q_{B_4}$  为节点 B 的 4 条星间链路的缓存队列利用率,  $q_B$  为邻居节点 B 的综合负载情况,  $q_{ab}$  为当前节点 A 对应相邻节点 B 的缓存队列的利用率,  $q_{AB}$  为综合判决下一跳卫星的链路利用率信息。

链路负载状况采用三级负载判决形式, 代表 3 种等级状态, 如式(2)所示。等级越高说明链路越繁忙, 则减少到拥塞链路的流量, 起到均衡局部负载的作用。

$$L = \begin{cases} 0, & q_{AB} < 0.5 \\ 1, & 0.5 \leq q_{AB} < 0.85 \\ 2, & q_{AB} \geq 0.85 \end{cases} \quad (2)$$

状态信令包含综合判决下一跳卫星的链路利用率信息以及负载等级信息。初始时刻, 卫星每隔 1 s 更新一次本地负载信息, 如果当前时刻和上一时刻的状态相同, 则每隔 2 s 发送当前卫星的状态信令给邻居卫星。如果状态不同, 则每隔 1 s 发送状态信令。相邻卫星接收到状态信令数据后, 更新存储的网络负载信息表, 从而调整选择下一跳卫星的路由策略。

卫星节点定期监测周围卫星的状态信息, 当网络出现负载不均衡或者是链路断开的情况时, 卫星应该及时解决拥塞和链路断开时的路由问题。状态信令含有卫星的纬度和负载信息, 纬度信息用于判断相邻卫星是否进入极区。卫星在经过极区时关闭轨间链路, 卫星拓扑如图 4 所示。

路由转发利用当前和目的卫星的经纬度信息, 根据改进的最小跳路由算法计算下一跳卫星。当前节点 C(loc,lac), 其中 loc 和 lac 分别为当前卫星的经度和纬度, 目的卫星节点记为 D(lod,lad), 其中 lod 和 lad 分别为目的卫星的经度和纬度<sup>[5]</sup>。

当前卫星和目的卫星之间的纬度关系满足式(3)时, 可选路径的方向向上。

$$0^\circ \leq lad - lac < 90^\circ$$

$$-180^\circ \leq lad - lac < -90^\circ \quad (3)$$

当前卫星和目的卫星之间的纬度关系满足式(4)时, 可选路径的方向向下。

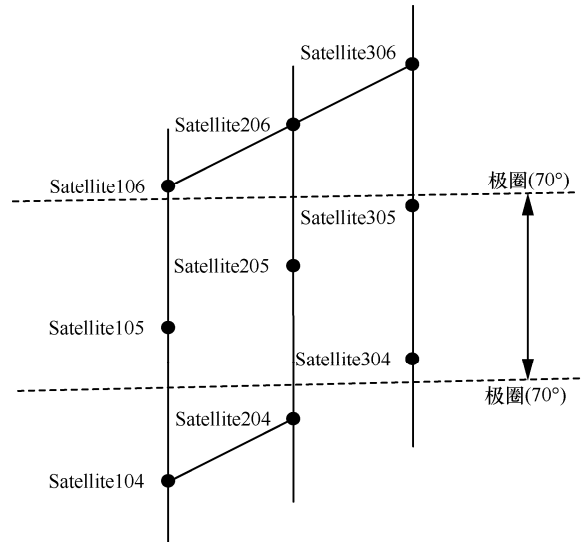


图 4 极圈内卫星拓扑

$$-90^\circ \leq lad - lac < 0^\circ$$

$$90^\circ \leq lad - lac < 180^\circ \quad (4)$$

当前卫星和目的卫星之间的经度关系满足式(5)时, 可选路径的方向向左。

$$-180^\circ \leq lod - loc < 0^\circ$$

$$180^\circ \leq lod - loc < 360^\circ \quad (5)$$

当前卫星和目的卫星之间的经度关系满足式(6)时, 可选路径的方向向右。

$$0^\circ \leq lod - loc < 180^\circ$$

$$-360^\circ \leq lod - loc < -180^\circ \quad (6)$$

当前卫星与目的卫星之间的经纬度关系总满足上述 4 个条件中的 2 个, 即该算法存在 2 个方向的可选下一跳卫星。在数据分组域中设置过极区标识, 如果经过极区, 则将其置 1, 反之置 0。此时由于极区上空的卫星关闭轨间链路, 如果数据分组路由到达极区, 但目的卫星不在极区, 则数据需保持进入极区的路由方向直至离开极区, 然后通过星间链路到达目的卫星节点, 防止根据最小跳算法在极区发送到轨间链路上造成数据分组丢失。

卫星发生拥塞现象时, 队列中缓存的数据分组将会增加, 从而造成排队时延的增加。此时, 根据卫星负载状态更新策略检测出拥塞等级。相邻卫星接收到状态信令后, 将减少发往拥塞节点的数据分组。当卫星节点 A 具有 2 个可选下一跳卫星 B 和 C 时, 此时采用负载判决机制, 根据可选卫星的拥塞等级进行判断。当  $L_B = L_C$  时, 下一跳卫星按比例 0.5 随机选择。如果  $L_B \neq L_C$ , 当  $|L_B - L_C| = 1$  时, 下一跳

卫星按比例 0.7 选择负载等级低的卫星，按比例 0.3 选择负载等级高的卫星，当  $|L_B - L_C| = 2$  时，下一跳卫星按比例 0.85 选择负载等级低的卫星，按比例 0.15 选择负载等级高的卫星。如果当前卫星与目的卫星在同一轨道面或具有相同轨内编号时，采取改进的最小跳算法为缓解拥塞，可能将数据分组路由到非最小跳数的路径方向，造成路由的跳数增加。

当源卫星节点为 104、目的卫星节点为 306 时，如图 5 所示，源节点首先判断与节点 204 之间的链路是否连接。如果链路断开，选择节点 105 作为下一跳卫星。反之，如果链路连接，则根据节点 105 和节点 204 的负载情况选择下一跳卫星。如果选择节点 105，则给数据分组设置极区标志，表示其经过极区的方向。然后当前卫星再根据极区标志判断下一跳卫星是 106，此时和目的节点在同一水平环内，途经卫星节点 206 转发数据分组到目的节点。如果选择节点 204，由于源节点与节点 304 的轨间链路关闭，只能选择到下一跳卫星节点 205，同理途经节点 206 到达目的节点。

## 4 OPNET 建模

### 4.1 星座负载建模

考虑到极地区域业务负载比较少，卫星在极区关闭轨间链路的情况下，对极地地区和非极地地区的业务分布进行单独建模。星座中共包含 66 颗卫星，对应将地球表面划分为 66 个服务区域，每个区域对应一颗 LEO 卫星承担区域中的通信业务。在卫星节点建模的过程中使用模拟数据源，用以模拟卫星在实际过程中接收到的地面终端发送的数据分组<sup>[15]</sup>。考虑极地地区和非极地地区业务负载不均衡，将星座中信元生成负载情况按照卫星在运行中所处的纬度进行更新。

卫星网络中模拟业务源的数据产生速率为  $10.24 \text{ kbit/s} \times W \times W_0$ 。其中， $W$  为不同卫星产生模拟通信业务的权重参数， $W_0$  为全网卫星设置的初始权重参数。卫星业务权重参数  $W$  如表 3 所示。

表 3 卫星业务权重参数  $W$

位置	目的节点在极区	目的节点不在极区
源节点在极区	0.05	0.14
源节点不在极区	0.14	0.67

### 4.2 网络层建模

OPNET 是商业化仿真软件，其中 Modeler 能够

对多种业务进行模拟，绘制仿真图和输出仿真报告。在仿真的过程中，Modeler 能够收集统计量数据、查看网络运行的动画和调试程序。Modeler 在仿真时维护一个全局事件列表，在仿真运行的时刻执行排在当前列表最前面的事件，并将基于数据分组的统计方法结合基于数学统计的建模方法<sup>[16]</sup>。

Iridium 星座网络层模型如图 5 所示，主要建模星座的拓扑结构，是实现仿真分析 LEO 路由算法的基础。在 OPNET 的网络层导入修改后的 STK 生成的轨道文件，每条轨道对应生成一个卫星节点，图 5 中包含 66 颗卫星节点和一个接收主询配置。接收主询模块通过基于卫星的经纬度、链路可视性以及衰落现象判断数据分组是否可达、星间链路是否连通，控制卫星之间数据分组的路由转发经过链路的连接情况。

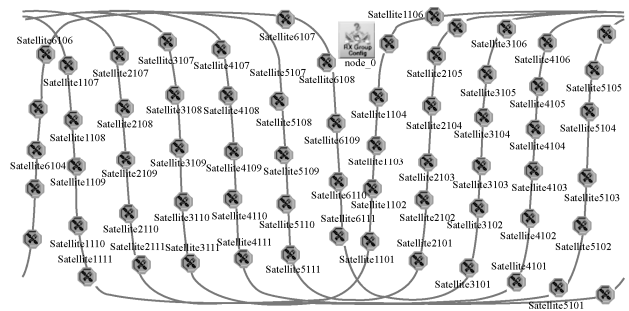


图 5 Iridium 星座网络层模型

### 4.3 节点层建模

Iridium 星座节点层模型如图 6 所示，用于模拟卫星节点实现的通信功能，是实现路由算法的硬件条件。天线的指向通过 ant\_point 模块进行周期调整，数据分组的接收和发送都是通过无线收发信机模块来实现的，每个无线收发信机对应一条 ISL。其中，虚线为状态连接线，传输队列模块的存储信息；实线为数据分组发送、接收的路径。

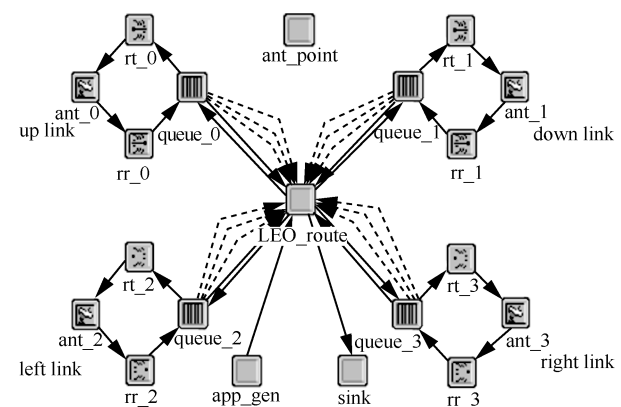


图 6 Iridium 星座节点层模型

图 6 中的 `app_gen` 模块用于模拟业务源的产生；`sink` 模块用于统计数据分组的传输时延，销毁数据分组释放仿真运行占用的内存空间；`LEO_route` 模块用于数据分组下一节点的选择。由于卫星路由考虑星间链路，每颗卫星节点需要有 4 条星间链路，因此节点层需要 4 对无线链路收发信机。

无线信号在仿真过程中需要配置 14 个管道阶段，包括调制方式、噪声系数、接收增益、背景噪声、信噪比、比特误码率等模拟真实无线信号传输的环境。模拟信号产生后通过分组流连接传输到路由模块，选择合适的下一跳卫星节点，传输到对应链路的缓存队列 `queue` 模块，然后经收发信机进行发送。

#### 4.4 进程层建模

节点域模块中每个模块都有不同的进程层模型，进程层是通过有限状态机模型进行建模的，每个状态机对应进程模型的一个状态，通过状态连接线实现状态的转移。在进程层中，状态分为强制性状态和非强制性状态 2 种，强制性状态是绿色标识，非强制性状态是红色标识<sup>[10]</sup>。当进程执行强制性状态时，直接执行入口代码和出口代码，然后跳入其他状态或终止。当进程执行非强制性状态时，先执行状态的入口代码，然后将控制器交给仿真核心，等待其他进程或者事件的触发返回再执行出口代码。

##### 1) `ant_point` 模块

由于卫星处在不停的运行过程中，卫星之间的位置会发生改变，因此需要天线校准模块及时地跟踪邻居卫星节点的位置，其状态转移如图 7 所示。其中，实线表示状态转移的起始路径，虚线表示状态返回路径。

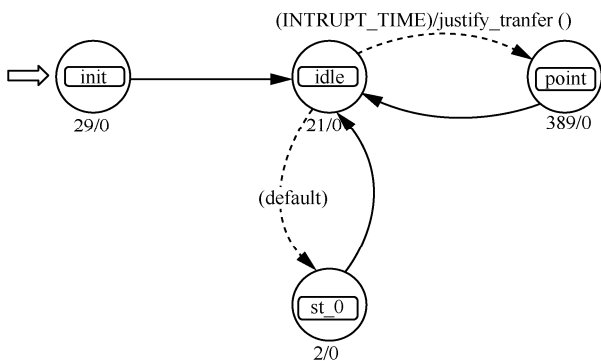


图 7 天线校准模块状态转移

图 7 中 `point` 模块是获取相邻卫星的位置参数进行天线校准的状态。首先获取相邻卫星节点的

ID，依次获取其经度、纬度和高度信息，然后赋值给天线的指向参数。天线指向跟踪的调整时间间隔为 1 s，通过设置天线校准模块提高发射增益，减少数据路由的误码率和数据分组丢失率。

##### 2) `LEO_route` 模块

数据分组的路由需要 `LEO_route` 模块选择合适的下一跳，路由模块内的有限状态机模型如图 8 所示。其中，实线表示状态转移的起始路径，虚线表示状态返回路径。

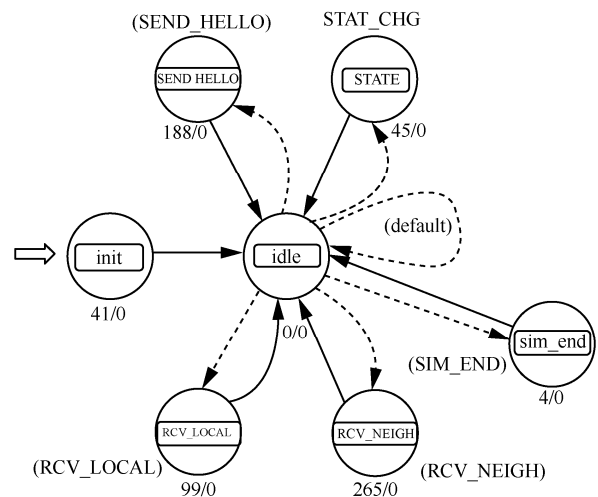


图 8 路由模块内的有限状态机模型

`STATE` 状态为当前卫星节点进行检测自身的 4 条链路的缓存情况，并划分相应的拥塞等级。`SEND_HELLO` 状态为当前节点发送的信令分组，将 `STATE` 状态统计的卫星节点的缓存数据和纬度信息存入信令分组。

`RCV_LOCAL` 状态为接收到本节点产生的模拟数据分组后执行的操作。`RCV_NEIGH` 状态首先判断接收到的数据是信令分组还是数据分组，如果是数据分组则判断其目的地址，再判断是执行销毁还是路由转发。

## 5 仿真结果与性能分析

在研究过程中，改变网络中模拟数据分组的产生速率，与典型的最小跳路由算法进行比较，通过分析数据分组的端到端时延、数据分组丢失率、平均路由跳数指标，判断星座网络的拥塞处理能力<sup>[17]</sup>。

### 5.1 路由性能指标

#### 1) 平均端到端时延

平均端到端时延为星座网络中数据分组从模拟信源中产生的时刻开始，到通过路由转发到达目

的卫星节点时刻的间隔的平均值。

$$T_{\text{avg}} = \frac{\sum_{i=1}^N T_i}{N} \quad (7)$$

其中,  $T_{\text{avg}}$  为数据发送到接收的平均时延,  $N$  为接收到的数据分组的数量,  $T_i$  为在星座中传输的单个数据分组的端到端时延。

### 2) 数据分组丢失率

数据分组在星座网络传输的过程中会遇到缓存溢出、误判决和路径损耗等数据分组丢失的现象, 丢失的数据分组数量与网络中传输的数据分组的总数的比值即数据分组丢失率。

$$R_{\text{loss}} = \frac{N_{\text{loss}}}{N_{\text{send}}} \quad (8)$$

其中,  $N_{\text{loss}}$  为在传输过程中丢失的数据分组的总数,  $N_{\text{send}}$  为网络中发送的数据分组的总数,  $R_{\text{loss}}$  为数据分组丢失率。

### 3) 平均路由跳数

在路由过程中, 平均路由跳数是所有数据分组到达目的节点经过的跳数的平均值。

$$H_{\text{avg}} = \frac{\sum_{i=1}^N H_i}{N} \quad (9)$$

其中,  $H_{\text{avg}}$  为平均路由跳数,  $N$  为接收到的数据分组的数量,  $H_i$  为网络中单个数据分组的路由跳数。

## 5.2 OPNET 仿真参数设置

本文采用 BPSK 进行数据编码, 并将卫星天线的校准周期设置为 1 s, 本文仿真参数设置如表 4 所示。

表 4 仿真参数设置

仿真参数	参数设置
天线校准周期/s	1
数据分组大小/bit	1 024
缓存队列处理速度/(bit·s <sup>-1</sup> )	56 320
收发信机处理速率/(Mbit·s <sup>-1</sup> )	2
信号中心频率/GHz	3
信号带宽/kHz	10
数据调制方式	BPSK
仿真运行时间/min	5

## 5.3 DRNL 算法仿真与分析

本文采用的比较路由性能的 2 种算法为文献[5]

提出的最小跳路由算法及其改进型, 均为分布式路由算法。

### 1) 固定路径的路由算法 (FPRA, fixed path routing algorithm)

卫星节点在路由决策时不考虑负载状况, 当数据分组到达时, 按照当前卫星和目的卫星的经纬度位置, 依据最小跳算法采用固定方向的策略选择下一跳节点进行路由。

### 2) 随机选择路径的路由算法 (RSRA, random select routing algorithm)

卫星节点不考虑负载状况, 当数据分组到达时, 卫星按照最小跳路由算法计算的 2 条备选方向随机选择一条作为下一跳卫星节点转发数据分组, 能够避免卫星网络在一些链路上的传输引起拥塞, 从而增强网络的路由能力。

## 5.4 仿真结果

### 1) 平均端到端时延

分组产生间隔越小, 卫星节点的负载越重, 所以端到端时延呈向上增长的趋势。仿真结果如图 9 所示。

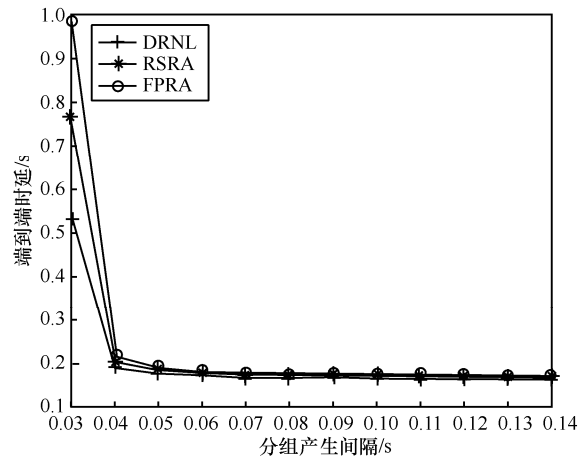


图 9 平均端到端时延

分组产生间隔在 0.08~0.14 s 时, 3 种路由算法为相对空闲状态, 端到端时延在 0.167 s 左右。分组产生间隔为 0.04~0.08 s 时, 3 种路由算法为相对繁忙状态, DRNL 路由算法依然保持稳定的路由能力, 端到端时延保持在 0.2 s 以下。分组产生间隔为 0.03 s 时, 网络达到拥塞状态, 3 种算法端到端时延大幅度上升, 但 DRNL 时延维持在 0.538 s。仿真结果表明, DRNL 具有缓解网络拥塞、降低时延的能力。

### 2) 数据分组丢失率

仿真结果表明, 随着网络负载加重, 数据分组

丢失率增加,如图 10 所示。当分组产生间隔为 0.03 s 时,3 种路由算法的数据分组丢失率均显著上升,这是由于网络的拥塞现象加重,部分节点的缓存队列出现溢出、数据分组同时到达收发信机引起冲突造成数据分组丢失,但是当分组产生间隔低于 0.03 s 时,DRNL 的数据分组丢失率维持在 0.005% 左右,保持较好的性能。

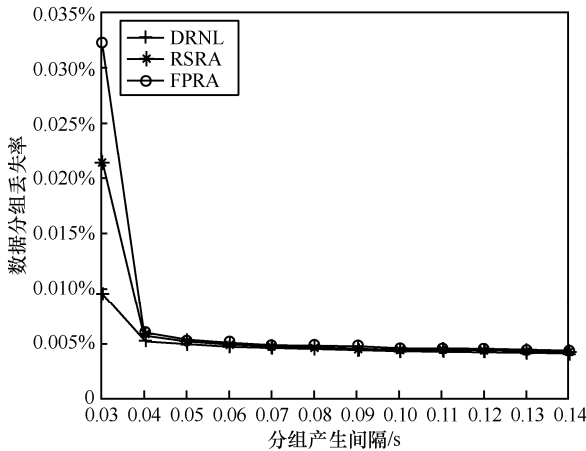


图 10 数据分组丢失率

### 3) 平均路由跳数

数据分组路由的跳数增加会造成时延增大,需要考虑卫星在自由空间的传播时延。同轨道面内卫星均匀分布,星间距离保持固定,即轨道内相邻卫星之间通信具有固定的时延,如式(10)所示。

$$t_{内} = \frac{\beta}{c}$$

$$\beta = \sqrt{2}R \sqrt{1 - \cos\left(\frac{360^\circ}{M}\right)} \quad (10)$$

其中,  $R$  是轨道面的半径,  $M$  是轨道面内卫星数,  $c$  是电磁波传播速度。

轨道间卫星之间的距离随着卫星的相对运动不断发生变化。在赤道附近纬度较低,卫星的分布比较稀疏,传输时延最大,计算式为

$$t_{间} = \frac{\alpha \cos(\text{lat})}{c}$$

$$\alpha = \sqrt{2}R \sqrt{1 - \cos\left(\frac{360^\circ}{2N}\right)} \quad (11)$$

其中,  $N$  是轨道面的数量,  $\text{lat}$  是卫星的纬度。

3 种算法平均路由跳数的仿真结果如图 11 所示。当分组产生间隔大于 0.07 s 时,网络相对空闲,FPRA 和 RSRA 的路由跳数基本保持不变;DRNL

的平均路由跳数最小,使平均端到端时延最小。当分组产生间隔小于 0.04 s 时,FPRA 和 RSRA 平均路由跳数下降,但从图 9 可知,端到端时延仍呈上升趋势。

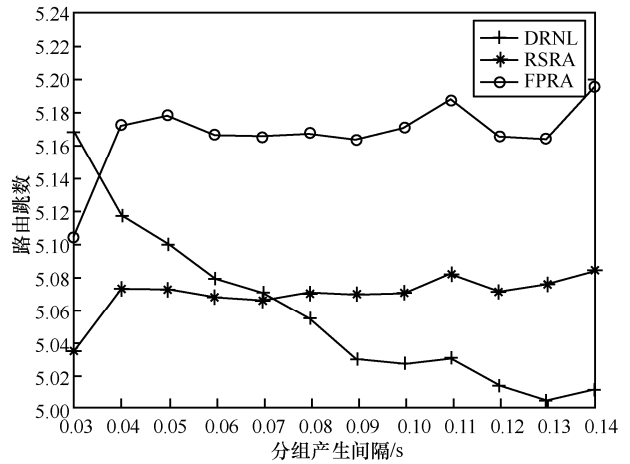


图 11 平均路由跳数

端到端时延需考虑排队时延和传播时延,若平均路由跳数减少,则传播时延降低,但是排队时延增加。这是因为数据分组丢失率的增加导致数据分组不能到达目的地,在路由过程中丢失。由于 DRNL 中数据分组采用迂回路由转发,具有局部拥塞缓解的功能,此时数据分组丢失率依旧很低,但是随着网络负载增加,DRNL 平均路由跳数逐渐增加。

DRNL 考虑相邻卫星的负载状态,卫星采用分布式思想,使每颗卫星能够独立进行负载均衡路由判决,实时响应卫星网络中负载的变化以及链路的通断问题。DRNL 以相对较低的运算复杂度换取端到端时延和数据分组丢失率的降低,但是以网络中相对较低的信令开销为代价,使星座具有一定的抗毁性。

## 6 结束语

当前,世界掀起新一轮卫星网络建设风潮,建设低轨卫星网络成为关系国计民生的重大战略性工程。路由算法是卫星网络的核心技术,本文提出的 DRNL 能够通过较低的信令开销获取缓解网络拥塞的能力,满足一定的实时性要求。DRNL 通过信令分组及时获取邻居卫星节点的负载状态和链路的连接情况,提升了网络的稳健性,能够满足我国对低轨卫星网络通信系统的发展需要。

### 参考文献:

[1] 何辞, 张亚生, 孙晨华, 等. 低轨星座组网及地面 IP 路由技术适应性分析[J]. 天地一体化信息网络, 2020, 1(1): 36-41.

- HE C, ZHANG Y S, SUN C H, et al. Analysis on low-earth-orbit constellation networking and adaptability of ground IP routing technology[J]. *Space-Integrated-Ground Information Networks*, 2020, 1(1): 36-41.
- [2] 林圣晔. 卫星通信系统[J]. *数字化用户*, 2019, 25(6):31-32.  
LIN S Y. Satellite communication system[J]. *Digital User*, 2019, 25(6): 31-32.
- [3] 甘仲民, 张更新. 卫星通信技术的新发展[J]. *通信学报*, 2006, 27(8): 2-9.  
GAN Z M, ZHANG G X. Current development of satellite communications technology[J]. *Journal on Communications*, 2006, 27(8): 2-9.
- [4] LIU W, TAO Y, LIU L. Load-balancing routing algorithm based on segment routing for traffic return in LEO satellite networks[J]. *IEEE Access*, 2019, 7: 112044-112053.
- [5] 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.  
FENG X B. Research on QoS routing protocol of multi-layer satellite network[D]. Harbin: Harbin Institute of Technology, 2016.
- [6] GENG S Y, LIU S F, FANG Z G, et al. An optimal delay routing algorithm considering delay variation in the LEO satellite communication network[J]. *Computer Networks*, 2020, 173: 107166.
- [7] WANG H T, WEN G L, LIU N J, et al. A load balanced routing algorithm based on congestion prediction for LEO satellite networks[J]. *Cluster Computing*, 2019, 22(4): 8025-8033.
- [8] 牟焕森, 郝玲玲. 铱星系统的创新过程及其经验分析[J]. *工程研究-跨学科视野中的工程*, 2007, 3: 173-182.  
MOU H S, HAO L L. The innovation process and experience analysis of Iridium system[J]. *Journal of Engineering Studies*, 2007, 3: 173-182.
- [9] 王欣蕊. 基于 STK 的北斗卫星导航系统仿真与分析[J]. *企业科技与发展*, 2018(3): 79-80.  
WANG X R. Simulation and analysis of Beidou satellite navigation system based on STK[J]. *Enterprise Science and Technology & Development*, 2018(3): 79-80.
- [10] DAELE G V, HINES K W, SANDERS B T, et al. Inter-satellite method for routing packets[R]. (1997-02-04)[2021-03-10].
- [11] LIU Z G, WU J H. Mobile satellite network virtual mapping algorithm based on node risk[C]//2016 International Conference on Network and Information Systems for Computers. Piscataway: IEEE Press, 2016: 76-79.
- [12] CHEN Q, GUO J M, YANG L, et al. Topology virtualization and dynamics shielding method for LEO satellite networks[J]. *IEEE Communications Letters*, 2020, 24(2): 433-437.
- [13] DONNER A, ZAMORANO R, COSTACHE M. Maximum persistence routing in satellite constellation networks[C]//22nd AIAA International Communications Satellite Systems Conference & Exhibit. Reston: AIAA, 2004. doi.org/10.2514/6.2004-3167.
- [14] 张悦. 低轨卫星网络路由算法研究[D]. 西安: 西安电子科技大学, 2019.  
ZHANG Y. Research on routing algorithm of LEO satellite network[D]. Xi'an: Xidian University, 2019.
- [15] WAHLBERG P. Communication device and a link system for satellite communication: CN, CN1522505 A[P]. 2004.
- [16] CHANG X J. Network simulations with OPNET[C]//Proceedings of the 1999 Winter Simulation Conference. Piscataway: IEEE Press, 1999: 307-314.
- [17] ICHIYOSHI O. Satellite communication conference system and conference method capable of real-time conference[R]. (2001-11-15)[2021-03-10].

## [作者简介]



杨明川 (1977- ), 男, 甘肃靖远人, 博士, 哈尔滨工业大学副研究员、博士生导师, 主要研究方向为卫星移动通信系统、高空平台站通信系统、深空通信及空、天、地一体化信息网络。



薛冠昌 (1998- ), 男, 安徽砀山人, 哈尔滨工业大学博士生, 主要研究方向为卫星通信、星地一体化网络技术。



李清毅 (1976- ), 女, 河北承德人, 北京空间飞行器总体设计部工程师, 主要研究方向为航天器总体仿真和体系仿真。