

基于多元时变图的天地一体化网络组播路由算法

段 洁, 闫子豪, 刘 亮, 孙春霞, 赵国锋

(重庆邮电大学通信与信息工程学院, 重庆 400065)

摘要: 为了保障天地一体化网络中组播业务的时效性, 使能源消耗降低, 提出了基于多元时变图的节能组播路由算法。首先, 针对低轨卫星网络的组播路由能耗问题进行分析, 提出多元时变图模型, 表征低轨卫星网络的时变拓扑、能耗和时延等多元属性。然后, 利用基于K-最短路径候选路径算法得到满足组播业务服务质量需求的路径集, 再通过基于最小路径启发式的组播生成树算法得到能耗近似最优的组播树。最后, 从理论上证明该算法构造的组播树在服务质量约束下能量消耗接近最优。仿真结果表明, 与已有的算法相比, 所提算法的网络能耗更低。

关键词: 天地一体化网络; 组播业务; 服务质量需求; 时变图; 组播路由算法; 能源消耗

基金项目: 国家自然科学基金(No.621701233, No.61701058); 重庆市教委科学技术研究项目(No.KJQN202200615)

中图分类号: TN927.2

文献标识码: A

文章编号: 0372-2112(2025)05-1469-13

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20240695

Multicast Routing Algorithm for Space-Air-Ground Integrated Networks Based on Multi-Dimensional Time-Varying Graphs

DUAN Jie, YAN Zi-hao, LIU Liang, SUN Chun-xia, ZHAO Guo-feng

(School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: To ensure the timeliness of multicast services in space-air-ground integrated networks, for minimizing energy consumption, this paper proposes an energy-efficient multicast routing algorithm based on multi-dimensional time-varying graphs. First, we analyze the energy consumption issues of multicast routing in low earth orbit (LEO) satellite networks and develop a multi-dimensional time-varying graph model to characterize the time-varying topology, energy consumption, and delay of LEO satellite networks. Then, the K-shortest path (KSP) candidate path algorithm is applied to generate a path set that satisfies the quality of service (QoS) requirements of multicast services. A minimum path heuristic (MPH)-based multicast tree construction algorithm is further employed to derive the solution. Finally, it is theoretically proven that the multicast tree constructed by the proposed algorithm achieves near-optimal energy consumption under QoS constraints. Simulation results demonstrate that the proposed algorithm outperforms the existing methods in terms of network energy efficiency.

Key words: space-air-ground integrated network; multicast services; quality of service requirements; time-varying diagram; multicast routing algorithms; energy consumption

Foundation Item(s): National Natural Science Foundation of China (No.621701233, No.61701058); Scientific and Technological Research Program of Chongqing Municipal Education Commission under Grant (No.KJQN202200615)

1 引言

从第一代(1G)无线网络到第五代(5G)无线网络, 地面无线网络在通信速度和服务质量(Quality of Service, QoS)要求方面取得了巨大成功。但是, 由于地理环境、经济成本等限制, 包括5G网络在内的地面网络主要部署在一些人口比较密集的区域, 这导致辽阔的空域、海域以及沙漠等地区无法被地面网络覆盖。低轨

(Low Earth Orbit, LEO)卫星网络以其广覆盖、高带宽、低时延等特点成为未来天地一体化网络的重要组成部分^[1]。目前, 在低轨卫星网络中存在大量具有差异性QoS要求的组播业务, 如视频分发、天气预报等, 组播路由可以极大地提高该类业务的传输效率^[1-3]。随着通信需求的增加和通信技术的进步, 卫星制造和发射成本的降低, 近轨卫星系统和项目陆续开展, 为海量人机通

信终端提供全球无缝接入服务,目前已经建立了各种卫星星座项目来构建全球通信的卫星网络,例如 Starlink、OneWeb、全球星和 Telesat 等. 与中高轨卫星相比,低轨卫星在传播延迟、路径损耗、抗击能力等方面具有优势,并且可以很好地支持偏远或受灾地区的数据采集、信息广播和信令交互,从而满足物联网服务的迫切需求.

然而,由于低轨卫星的飞行高度相对较低、体积和质量具有严格的限制,使得卫星的能源有限,其中大约 20% 的能源用于卫星的正常飞行,约 70% 的能源需要用于处理和传输数据,剩下约 10% 能源用于其他的子系统^[4-6]. 而低轨卫星的能源主要来源于两个方面:普通电池和太阳能电池,普通电池的容量相对较小且无法进行再充电,太阳能电池可以依靠太阳辐射产生电力从而达到循环利用. 然而,即使电力供应充足,无节制地使用能源也会导致卫星快速老化,因为太阳能电池的最大完整充电/放电循环次数是有限的,即循环寿命或循环次数是有限制的. 此外,卫星的建造和发射成本依旧相当昂贵,节省卫星的能耗,延长卫星的寿命有利于降低卫星的成本. 因此,针对高动态的低轨卫星网络中具有差异性 QoS 需求的节能组播路由问题,设计一种满足不同用户 QoS 要求的高效组播路由算法是降低卫星网络能耗的关键问题.

与传统地面网络相比,低轨卫星网络目前存在两个主要挑战:

(1) 源端和目的地的服务卫星动态变化. 由于低轨卫星的高速移动,导致地面的源和目的地的服务卫星在不断地发生变化. 因此,当服务卫星发生变化时,必须重新计算组播树.

(2) 卫星的能源和星载处理能力有限,无限制地使用卫星的能源将缩短卫星的使用寿命,增加卫星的成本. 此外,在传统的地面网络中开发的许多组播路由算法,包括距离矢量组播路由协议,组播开放最短路径优先等基于核心节点组播路由算法中,其核心思想是将组播数据包发送到核心节点,再中继到所有组播成员,该路由方式易造成核心卫星节点负担过大,过度消耗核心卫星节点的能源,这对于高动态且能源有限的低轨卫星网络无法直接应用.

因此,本文针对高动态的低轨网络中具有差异性 QoS 需求的组播业务,充分考虑星间链路和星地链路的变化,提出基于多元时变图(Multivariate Time-Varying Graphs, MTVG)的组播路由算法,在满足组播业务和每组播成员 QoS 需求的条件下,构建跨多时隙的接近最小能耗的组播树.

本文主要的研究工作如下:

(1) 针对高动态的低轨卫星网络中具有差异性 QoS

需求的节能组播问题进行分析.

(2) 基于 MTVG 的组播路由算法. 首先,采用多元时变图模型刻画低轨卫星网络的时变拓扑、能耗和时延的多元属性,并将节能问题转化为路径组合优化问题;然后,利用基于 KSP 的候选路径选择算法得到满足组播业务和每组播成员的 QoS 需求的候选路径;最后,通过基于 MPH 的组播生成树算法将初始组播树优化为能耗接近最小的组播树.

(3) 从理论上证明该算法的正确性,且时间复杂度为 $O(n^2 \log_2^2 n + KZ)$. 仿真结果表明,与已有的算法相比,所提算法的网络能耗更低.

2 相关工作

近年来,国内外学者已提出了不少针对低轨卫星网络的组播路由算法,根据路由决策方式,主要分为两大类:基于软件定义的组播路由算法^[7-9]、分布式组播路由算法^[10-13]. 基于软件定义的算法的主要思想是利用集中式控制器获取卫星网络的全局信息,提出满足 QoS 要求的不同组播路由算法. 文献[7]提出加权直线斯坦纳树算法以平衡节省带宽和 QoS 要求. 文献[8]提出基于分段路由的组播路由算法达到降低通信延时. 文献[9]提出了一种多层直线斯坦纳树构造算法有效地节省了带宽. 然而,对于网络拓扑高动态变化的卫星网络,上述基于软件定义的组播路由算法由于需要实时全局网络拓扑信息,因此控制器节点需要不断获取连接状态信息进行路径规划,这容易导致控制器节点的能耗开销过大,缩短卫星的使用寿命^[10].

分布式组播路由算法的主要思想是利用卫星网络拓扑变化的可预测性和周期性,将一个网络拓扑周期划分为若干个连续时隙,即时隙是卫星网络拓扑保持不变的时间段,再通过时间链路将每个时隙相互关联,然后利用不同的分布式启发算法动态规划满足 QoS 服务的组播路由. 针对卫星网络中延迟约束的节能广播问题,文献[11]提出了一种基于时空图的回溯搜索贪婪算法,求出最佳传输节点和最佳传输时间,实现在受时延约束情况下的最小传输次数,从而达到降低广播传输的能耗. 文献[12,13]提出了基于时序网络网格模型的多功率水平流图来描述时变拓扑,然后分别提出了基于多功率级多传输时空图的算法和基于时序网络网格模型的多功率水平流图的迭代广播路由算法,在满足时延要求的同时,将广播能耗降至最低. 但是,上述路由算法是针对广播传输的,其方法不能直接应用于只有部分目的节点的组播场景. 为此,文献[14]利用时间扩展图(Time-Expanded Graph, TEG)刻画卫星网络的时变特性,提出了一种基于 TEG 的分布式启发式组播算法,在不超过延迟界限的情况下同时减少卫星的

组播传输能耗. 虽然该方法采用的分布式组播路由算法可以有效地避免控制器节点带来的能耗过大问题, 但是未充分考虑卫星高速移动带来的星地链路变化会导致地面节点与不同的目的卫星进行通信. 上述路由算法未同时考虑到卫星的能源有限性和业务 QoS 要求的差异性和多样性, 以及接入卫星的改变造成组播树频繁重构. 因此, 在满足业务差异性 QoS 要求的前提下, 降低卫星的传输能耗是亟须解决的问题.

3 问题分析

低轨卫星网络由地面节点和卫星节点以及星地链路和星间链路组成, 地面节点需要借助低轨卫星完成通信. 在低轨卫星网络中卫星的高速移动会导致卫星网络拓扑动态变化, 但是, 在一个时隙内卫星网络拓扑保持不变. 如图 1 展示了三个连续时隙(τ_1, τ_2, τ_3)内的低轨卫星网络拓扑, 其中, 包括地面源节点 s , 地面目的节点 g_1, g_2, g_3 , 低轨卫星节点 R_1, R_2, R_3, R_4 . 在 τ_1 时隙, 地面源节点 s 开始发送组播业务 X_1 , 其有效时间为 120 s. 各组播成员对该组播业务的 QoS 要求具有差异性, 为了方便描述, 本文的 QoS 要求采用时延表示且每个时隙的大小为 60 s. 因此, 目的节点 g_1, g_2, g_3 对业务 X_1 的时延要求分别为 70、90、120 s, 超过接收时延和业务有效时间则表示组播业务传输失败.

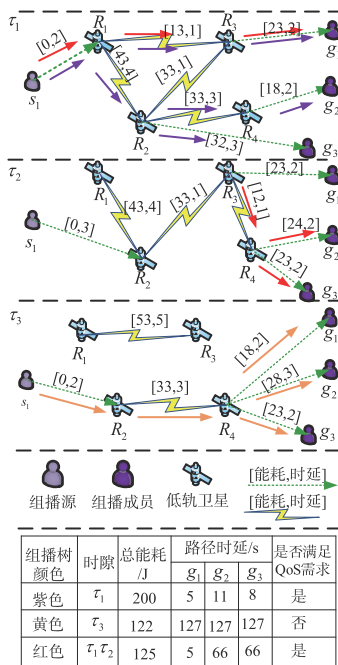


图 1 低轨卫星网络多时隙单组播业务路由场景

然而, 由于低轨卫星的高速运动造成星间链路和星地链路间歇性连接, 使得同一个地面节点在不同时隙与不同的卫星产生连接, 因此, 同一地面目的节点需

要及时更新目的接入卫星. 例如在 τ_1 时隙, 地面目的节点 g_3 与 R_2 连接, 而在 τ_2 时隙, 与中继卫星 R_4 连接, 显然, 中继卫星 R_2 不能一直作为地面目的节点 g_3 接入的卫星, 否则除了 τ_1 时隙, 在其他时隙目的节点 g_3 无法接受组播业务 X_1 .

若在 τ_1 时隙内通过如紫色箭头所示的组播树将组播业务 X_1 发送给目的节点 g_1, g_2, g_3 , 可以满足组播业务 X_1 的有效时间和各地面目的节点的接收时延, 但是该组播树的能耗较大. 若在 τ_3 时隙内通过如黄色箭头所示的组播树将业务 X_1 发送给目的节点 g_1, g_2, g_3 , 则有效地降低卫星能耗, 但是该组播树不满足组播业务的有效时间和各目的节点的接收时延. 若通过联合不同时隙的传输和存储, 建立一颗跨多时隙的组播树, 如红色箭头所示, 在满足组播业务的有效时间和各目的节点的接收时延的前提下, 同时达到减少卫星的能耗.

因此, 针对高动态的低轨卫星网络, 如何充分考虑星地和星间链路的动态性, 在满足组播业务的时效性和各目的节点的接收时延约束的前提下, 以尽可能小的能耗完成组播传输是亟须解决的问题.

4 系统模型

低轨卫星网络是由同一层的低轨卫星和地面节点组成, 其中 R 表示卫星节点集合, $R = \{R_i\}, i = 1, 2, \dots, N, N$ 表示卫星的数目; 地面节点包括地面源节点和地面目的节点, 地面源节点表示发送组播业务的节点, 采用集合 S 表示, 即 $S = \{s_i\}, i = 1, 2, \dots, \mathcal{S}, \mathcal{S}$ 表示源节点的数目; 地面目的节点表示接收组播业务的节点, 采用集合 G 表示, 即 $G = \{g_z\}, z = 1, 2, \dots, Z, Z$ 表示目的节点的数目; 待传输的组播业务采用集合 X 表示, 即 $X = \{X_b\}, b = 1, 2, \dots, Q, Q$ 表示业务的数目. 特别地, 对于任意组播业务 $X_b \in X$, 采用三元组 $(\theta_b, ts_X^b, te_X^b)$ 描述, 其中组播业务大小 θ_b , 组播业务开始时间 ts_X^b , 组播业务截止时间 te_X^b . 各地面目的节点的接收时延为 $\Delta Tg^b = \{\Delta Tg_1^b, \Delta Tg_2^b, \dots, \Delta Tg_z^b\}$, 任意组播业务 X_b 的有效时间为 $\Delta T_b^* = |te_X^b - ts_X^b| \geq 0$.

4.1 能耗模型

网络能源消耗是网络中的设备进行通信和相关动作所消耗的能量, 简称网络能耗. 在传统的无线传感网络中, 网络能耗被视为网络中的节点在接收、处理和传输数据所消耗的能量总和, 而忽略了网络节点在启动相关功能和维护设备正常运行所消耗的能量. 在低轨卫星网络中, 卫星能耗主要来源于如下 5 个部分^[15,16]: (1) 星地链路传输; (2) 星间链路传输; (3) 机载 CPU (Central Processing Unit) 和处理器; (4) 存储数据; (5) 机载平台的自我控制和自我维护. 而机载平台的自我控

制和自我维护所消耗的能量与业务的数据量大小无关,是必须消耗的能量,因此,本文的能耗模型不考虑此能耗.星间链路传输和星地链路传输所消耗的传播能耗与两节点之间的距离相关,因此,星间链路能耗和星地链路能耗被归为链路能耗,而机载平台 CPU 和处理器产生的处理能耗以及存储数据所产生的存储能耗与两节点之间的距离无关,则处理能耗和存储能耗被归为节点能耗.

4.1.1 节点能耗

(1) 处理能耗

卫星的处理能耗分为两个部分:(1)启动机载 CPU 和数据处理器等相关功能所产生的启动能耗;(2)数据处理器处理业务所产生数据处理能耗.启动能耗与业务的大小无关,采用 ε^0 表示,数据处理能耗与节点处理的业务大小相关,采用 φ 表示.因此,对于卫星节点 R_i 的处理能耗的计算式可表示为

$$\Psi_i = \varepsilon_i^0 + \varphi_i = \varepsilon_i^0 + \eta C_i^\alpha \quad (1)$$

式中, ε^0 是常量, η, α 是由处理器本身决定的常量, C 表示节点处理的业务数据量大小.处理的数据量越大所产生的处理能耗越大,若处理的数据量较小,处理器所产生的启动能耗将远远大于数据处理能耗.

(2) 存储能耗

卫星节点的存储能耗与卫星节点是否存储业务数据有关,只有当卫星存储数据时才会产生存储能耗,存储能耗分为三个部分:数据读入能耗 A^w 、数据读出能耗 A^r 、数据储存能耗 A^s .假设卫星节点 R_i 读入、读出和储存单位数据所需功率分别为 p^w, p^r, p^s ,所需时间分别为 t^w, t^r, t^s ,若卫星节点 R_i 存储的数据量为 C ,则卫星节点 R_i 的存储能耗的计算式可表示为

$$A_i = C_i(A^w + A^r + A^s) = C_i(p^w t^w + p^r t^r + p^s t^s) \quad (2)$$

存储能耗与存储的数据量大小有关,数据量越大所产生存储能耗越大.

因此,节点 R_i 的能耗计算式可表示为

$$\text{En}_i = \Psi_i + A_i = \varepsilon_i^0 + \eta C_i^\alpha + C_i A_i^c \quad (3)$$

4.1.2 链路能耗

(1) 星间链路能耗

在低轨卫星网络中,数据传播主要受自由空间损失的影响,基于自由空间传播定理,在任意两颗相邻的卫星 R_i 传播数据给卫星 R_j 所构成的星间链路表示为 (R_i, R_j) ,在链路 (R_i, R_j) 上传播的数据量为 C_{ij} 所消耗的能量表示为 $\text{En}_{ij}^{\text{ISL}}$:

$$\text{En}_{ij}^{\text{ISL}} = C_{ij} \mu_{ij} t_{ij} d_{ij}^2 \quad (4)$$

式中, $\mu_{ij} = \frac{16\pi^2 p_i^r}{G_i^r G_j^t \lambda^2}$ 为固定参数,与传播的数据量大小和传播距离无关, p_j^r 表示卫星的接收功率, p_i^r 表示发射功率, G_i^r

表示接收增益, G_i^t 表示发射增益, λ 为工作波长, t_{ij} 和 d_{ij} 分别表示星间链路 (R_i, R_j) 的传播时间和传播距离.

(2) 星地链路能耗

在星地链路中,星地链路可分为上行链路和下行链路.对于低轨卫星节点来说,由于上行星地链路卫星不需要发送数据,所以上行星地链路不产生链路传输能耗.而对于下行星地链路,卫星需要发送数据给地面用户,则下行星地链路会产生链路传输能耗.因此,在任意下行星地链路 (R_i, g_z) 上,卫星 R_i 传播 C_{iz} bit 数据给地面目的节点 g_z 所消耗的能量可表示为 $\text{En}_{iz}^{\text{down}}$:

$$\text{En}_{iz}^{\text{down}} = C_{iz} \mu_{iz} t_{iz} d_{iz}^2 \quad (5)$$

式中, $\mu_{iz} = \frac{16\pi^2 p_z^r}{G_i^r G_z^t \lambda^2}$ 表示下行星地链路 (R_i, g_z) 的常量, t_{iz} 和 d_{iz} 分别表示下行星地链路 (R_i, g_z) 的传播时间和传播距离.

4.1.3 网络总能耗

低轨卫星网络的总能耗为网络中各个节点所产生的能耗总和,可表示为

$$\text{En}_{\text{total}} = \sum_{i=0}^N y_i (\gamma_i \text{En}_i^c + \Psi_i) + \sum_{i=0}^N \sum_{j=0}^N \chi_{ij} \text{En}_{ij}^{\text{ISL}} + \sum_{i=0}^N \sum_{z=0}^Z \chi_{iz} \text{En}_{iz}^{\text{down}} \quad (6)$$

式中, En_{total} 表示网络的总能耗,包括星间链路能耗 $\text{En}_{ij}^{\text{ISL}}$ 、下行星地链路能耗 $\text{En}_{iz}^{\text{UDL}}$; $y_i, \gamma_i, \chi_{ij}, \chi_{iz}$ 均为布尔变量,当 $y_i = 1$ 时,表示节点 R_i 处于工作状态,当 $y_i = 0$ 时,表示节点 R_i 处于休眠状态;当 $y_i = 1, \gamma_i = 1$ 时,表示节点 R_i 开启存储功能,否则关闭存储功能;若 $\chi_{ij} = 1$,则 $y_i = 1, y_j = 1$,表示星间链路 (R_i, R_j) 被选择加入组播树,且节点 R_i, R_j 同时开启工作状态,否则不被选择;若 $\chi_{iz} = 1$,则 $y_i = 1, y_z = 1$,表示下行星地链路 (R_i, g_z) 被选择加入组播树,且节点 R_i, g_z 同时开启工作状态,否则不选择.

4.2 路径时延模型

任意低轨卫星节点之间的通信时延包括发送时延、传播时延、处理时延、存储时延和排队时延,由于传播时延和存储时延远远大于发送时延、处理时延和排队时延,因此,通信时延主要考虑传播时延和存储时延.传播时延是组播业务在任意两个节点之间传播所需要的时间可表示为

$$t = \frac{d}{c} \quad (7)$$

式中, d 表示两个节点之间的传播距离, c 表示电磁波在真空中的传播速度.存储时延是将业务存储在节点上以等待发送所消耗的时间.

因此,地面源节点 s 到每个目的节点 g_z 的路径时延可表示为

$$T_{s \rightarrow g_z}^{\text{total}} = \sum_{i=0}^N \sum_{j=0}^N \chi_{ij} t_{ij} + \sum_{i=0}^N (\chi_{si} t_{si} + \chi_{iz} t_{iz} + \gamma_i t_i^c) \quad (8)$$

式中, $T_{s \rightarrow g_z}^{\text{total}}$ 表示地面源节点 s 分别到达每个地面目的节点 g_z 的路径时延, χ_{ij}, χ_{iz} 与第 3 节的意义相同, $\chi_{si} = 1$ 表示上行星地链路 (s, R_i) 加入组播树; t_{ij} 表示星间链路 (R_i, R_j) 的传播时延, t_{si} 表示上行星地链路 (s, R_i) 的传播时延, t_{iz} 表示下行星地链路 (R_i, g_z) 的传播时延, t_i^c 表示节点 R_i 的存储时延; 当 $\gamma_i = 1$ 时, 表示节点 R_i 开启存储功能, 产生存储时延, 否则不产生存储时延。

4.3 问题建模

假设在网络中组播业务的地面源节点为 s , 地面目的节点为 g_z , 对于任意组播业务 $X_b = (\theta_b, ts_X^b, te_X^b)$, 其有效时间为 ΔT_b^* , 不同的目的节点 g_z 对该组播业务 X_b 的接收时延为 $\Delta T_{g_z}^b$, 且 $\Delta T_{g_z}^b$ 并不完全相同, 则组播传输节能问题可建模为

$$\begin{aligned}
 & \text{P1: } \quad \text{Min } \text{En}_{\text{total}} \\
 & \left\{ \begin{array}{l}
 \text{C1: } T_{s \rightarrow g_z}^{\text{total}} \leq \Delta T_b^* = |te_X^b - ts_X^b| \geq 0 \\
 \text{C2: } T_{s \rightarrow g_z}^{\text{total}} \leq \Delta T_{g_z}^b \\
 \text{C3: } \sum_{R_j \in R^+} C_{ji} = C_{ik} = \theta_b \\
 \text{C4: } \sum_{R_i \in R, s \in S} C_{si} = C_{iz} = \theta_b \\
 \text{C5: } \begin{cases} 0 \leq C_{ij} \leq \omega_{ij} \\ 0 \leq C_{si} \leq \omega_{si} \\ 0 \leq C_{iz} \leq \omega_{iz} \end{cases} \\
 \text{C6: } C_i \leq \text{Buf}_i
 \end{array} \right. \\
 & 1 \leq b \leq Q, 1 \leq z \leq Z, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq k \leq N, 1 \leq s \leq S
 \end{aligned} \tag{9}$$

式中, C1 约束表示源节点到不同的目的节点的路径时延小于等于组播业务的有效时间, $T_{s \rightarrow g_z}^{\text{total}}$ 表示源节点 s 发送组播业务 X_b 到目的节点 g_z 的路径时延. C2 约束表示源节点到不同的目的节点的路径时延小于等于该目的节点的接收时延.

C3 约束表示流入节点 R_i 的数据量等于节点 R_i 流出到每个邻居节点 R_k 的数据量等于组播业务的大小; 其中, 对于任意节点 R_i , 若节点 R_j 是节点 R_i 的流入邻居, 则称节点 R_j 为 R_{ji} , 节点 R_i 的流入邻居节点集合可表示为 $R_i^+ = \{R_{ji}\}, j = 1, 2, \dots, N$; 若节点 R_k 是节点 R_i 的流出邻居, 则称节点 R_k 为 R_{ik} , 节点 R_i 的流出邻居节点集合为 $R_i^- = \{R_{ik}\}, k = 1, 2, \dots, N$.

C4 约束表示地面源节点 s 流出的数据量与流入各地面目的节点 g_z 的数据量相等. C_{ji} 表示邻居节点 R_j 流入到节点 R_i 的数据量, C_{ik} 表示节点 R_i 流出到节点 R_k 的数据量.

C5 约束表示链路容量约束. $0 \leq C_{ij} \leq \omega_{ij}$ 表示对于任意星间链路 (R_i, R_j) , 流经该链路的流值不能超过链路的带宽, 式中, C_{ij}, ω_{ij} 分别表示星间链路 (R_i, R_j) 上流

经的数据量和带宽值. 同理可得, $0 \leq C_{ui} \leq \omega_{ui}$ 表示对于任意上行星地链路 (s, R_i) , 式中, C_{si}, ω_{si} 分别表示上行星地链路 (s, R_i) 上流经的数据量和带宽值; $0 \leq C_{iz} \leq \omega_{iz}$ 表示对于任意下行星地链路 (R_i, g_z) , 式中, C_{iz}, ω_{iz} 分别表示下行星地链路 (R_i, g_z) 上流经的数据量和带宽值.

C6 约束表示缓存容量约束, 即任意节点 R_i 缓存的数据量不超过节点 R_i 的存储空间. 式中, C_i 表示节点 R_i 的存储的数据量, Buf_i 表示节点 R_i 的存储空间.

然而, 即使将组播业务 X_b 的有效时间 ΔT_b^* 和各自的节点的接收时延 $\Delta T_{g_z}^b$ 设置为无穷大, P1 问题仍是 NP-hard 问题.

5 基于 MTVG 的节能组播路由算法

由于源节点到每个目的节点之间的传输路径必须满足组播业务的时效性和组播成员的接收时延约束, 因此, 首先, 采用多元时变图对优化问题 P1 进行转化, 然后提出基于 KSP 的候选路径算法, 再利用基于 MPH 的组播生成树算法构建能耗接近最小的组播树. 最后, 证明算法的正确性.

5.1 多元时变图模型

为了更加清楚地表征卫星网络的时变特性, 将时变图 (Time-Varying Graph, TVG) 模型采用四元组 (V, E, ε, D) 描述, 式中 V 表示节点集合, E 表示边集合, ε 表示能耗集合, D 表示时延集合, 并将给定时间范围 $[t_0, t_k]$ 划分为 H 个时隙, 即 $[\tau_1, \tau_2, \dots, \tau_h, \dots, \tau_H], \tau_h = [t_{h-1}, t_h], 1 \leq h \leq H$, 时隙 τ_h 的大小为 $|\tau_h| = |t_h - t_{h-1}|$, 如图 2 所示.

节点集合 $V = V_R \cup V_G \cup V_S$, 其中 $V_R = \{v_i^h | v_i \in S\}, v_i^h$

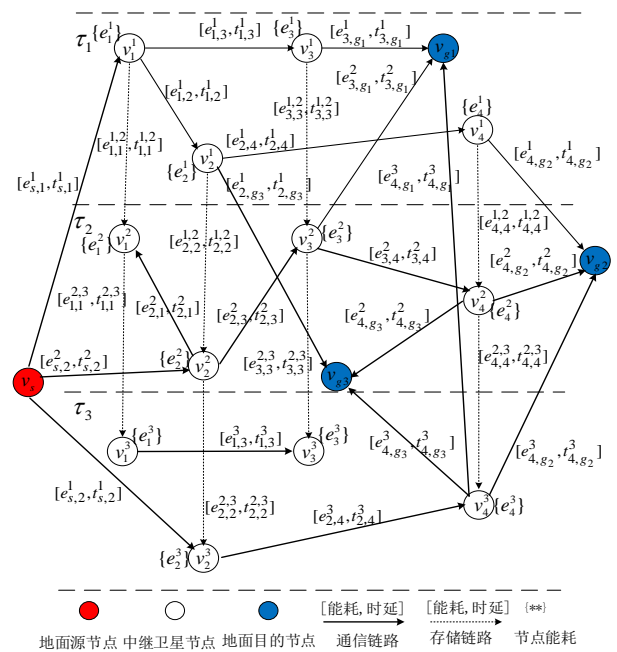


图 2 时变图模型

表示 τ_h 时隙中的卫星节点 v_i , 如图 2 中节点 v_i^1 表示 τ_1 中的节点 v_i ; $V_G = \{g_z | g_z \in G\}$, g_z 表示目的节点; $V_S = \{s | s \in S\}$, s 表示地面源节点. 为了方便对 TVG 中的节点进行统一表征, 采用 v_i^h 表示节点 R_i^h , v_g^z 表示节点 g_z , v_s 表示节点 s .

边集合 $E = E_l \cup E_c$, 其中, E_l 表示由传播链路组成的边集合, 包括星地传播链路和星间传播链路, 即 $E_l = \{l_{i,j}^h\}$, $l_{i,j}^h$ 表示在 τ_h 时隙中节点 v_i 到节点 v_j 的传播链路, 例如在 τ_1 时隙中节点 v_1 到节点 v_3 的传播链路表示为 $l_{1,3}^1$; E_c 表示由存储链路组成的边集合, 即 $E_c = \{l_{i,i}^{h,h+1}\}$, $l_{i,i}^{h,h+1}$ 表示节点 v_i 从 τ_h 时隙到 τ_{h+1} 时隙的存储链路, 例如节点 v_1 从 τ_1 时隙到 τ_2 时隙的存储链路表示为 $l_{1,1}^{1,2}$.

能耗集合 $\varepsilon = \varepsilon_V \cup \varepsilon_E$, 其中, ε_V 表示节点能耗, 即 $\varepsilon_V = \{e_i^h | i \in V_S\}$, e_i^h 表示在 τ_h 时隙中节点 v_i 的节点能耗, 采用式(3)计算; ε_E 表示链路能耗, 即 $\varepsilon_E = \{e_{i,j}^h\}$, $e_{i,j}^h$ 表示在 τ_h 时隙中传播链路 $l_{i,j}^h$ 上的能耗, 星间传输链路能耗采用式(4)计算, 星地传输链路能耗采用式(5)计算.

时延集合 $D = D_l \cup D_c$, 其中, $D_l = \{t_{i,j}^h\}$, $t_{i,j}^h$ 表示在 τ_h 时隙中传播链路 $l_{i,j}^h$ 的传播延迟, 采用式(7)计算; (i,j) 表示传播链路 $l_{i,j}^h$; $D_c = \{t_{i,i}^{h,h+1}\}$, $t_{i,i}^{h,h+1}$ 表示存储链路 $l_{i,i}^{h,h+1}$ 上的存储时间, (i,i) 表示存储链路 $l_{i,i}^{h,h+1}$; 由于网络拓扑在一个时隙内保持不变, 因此, 将 $t_{i,i}^{h,h+1}$ 近似等于一个时隙 τ_h 的大小, 即 $t_{i,i}^{h,h+1} = \lfloor \tau_h \rfloor$.

然而, 在 TVG 中, 由于低轨卫星网络一对多的组播特性以及节点和链路的多元属性, 将导致节点能耗重复计算, 增加组播树的设计复杂度, 因此需要将节点能耗转化为边的能耗, 减少组播树的设计复杂度. 为了克服使用 TVG 构造组播树的困难, 通过在原始的 TVG 模型中增加虚拟节点和虚拟链路, 从而将节点的能耗转化为链路的能耗, 将原始的 TVG 转化为 MTVG, 通过如下步骤进行转换.

定义 1 虚拟中继节点. 对于任意的中继节点 v_i^h 存在 2 条及以上的传播链路, 则节点 v_i^h 增加一个虚拟节点 $v_{i^*}^h$. 例如, 在图 3 中, 中继节点 v_1^1 存在 2 条传播链路 $l_{1,2}^1, l_{1,3}^1$, 则增加一个虚拟中继节点 v_1^1 .

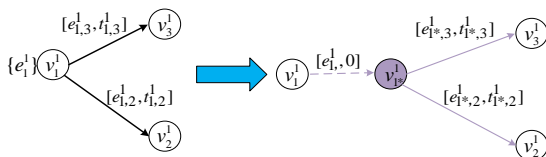


图 3 引入虚拟节点和虚拟链路的示例

定义 2 虚拟传播链路. 若中继节点 v_i^h 增加了虚拟节点 $v_{i^*}^h$, 则节点 v_i^h 和虚拟节点 $v_{i^*}^h$ 之间增加一条虚拟传播链路 l_{i,i^*}^h , 方向由 v_i^h 指向节点 $v_{i^*}^h$. 例如在图 3 中, 中

继节点 v_1^1 增加了一个虚拟中继节点 v_1^1 , 则增加一条虚拟链路 $l_{1,1^*}^1$.

定义 3 虚拟组播链路. 若节点 v_i^h 增加了虚拟节点 $v_{i^*}^h$, 则删除节点 v_i^h 的传输链路 $l_{i,j}^h \in E_l$, 增加虚拟组播链路 l_{i,j^*}^h . 例如在图 3 中, 中继节点 v_1^1 增加了一个虚拟中继节点 v_1^1 , 则需要删除传播链路 $l_{1,3}^1, l_{1,2}^1$, 同时增加虚拟组播链路 $l_{1,3^*}^1, l_{1,2^*}^1$.

MTVG 仍采用四元组 $(V^*, E^*, \varepsilon^*, D^*)$ 描述, 如图 4 所示.

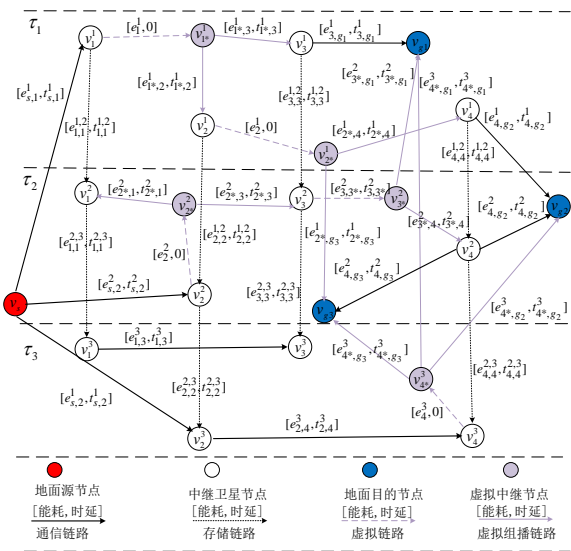


图 4 多元时变图

节点集合 $V^* = V \cup V_{\text{virt}}$, 其中, V 是 TVG 中的节点集合; V_{virt} 表示虚拟中继节点集合, 即 $V_{\text{virt}} = \{v_{i^*}^h\}$, $v_{i^*}^h$ 表示在 τ_h 时隙中的虚拟节点 $v_{i^*}^h$.

边集合 $E^* = E \cup E_{\pi} \cup E_m$, 其中, E 是 TVG 中的边集合; E_{π} 表示虚拟传输链路集合, 即 $E_{\pi} = \{l_{i,i^*}^h\}$, l_{i,i^*}^h 表示在 τ_h 时隙中的节点 v_i 到节点 v_{i^*} 的虚拟传播链路; 集合 E_m 表示虚拟组播链路集合, 即 $E_m = \{l_{i,j^*}^h\}$, l_{i,j^*}^h 表示在 τ_h 时隙中的节点 v_i 到节点 v_{j^*} 的虚拟组播传播链路.

能耗集合 $\varepsilon^* = \varepsilon_E \cup \varepsilon_l \cup \varepsilon_m \cup \varepsilon_c$, 其中, ε_E 与 TVG 中的 ε_E 具有相同意义; $\varepsilon_l = \{e_{i,j}^h\}$, $e_{i,j}^h$ 表示虚拟组播链路 l_{i,j^*}^h 的能耗, 其大小等于 $e_{i,j}^h$; $\varepsilon_m = \{e_{i,i^*}^h\}$, e_{i,i^*}^h 表示虚拟链路 l_{i,i^*}^h 的传播能耗, 其大小等于 $\text{En}_i^0 + \psi_i$; $\varepsilon_c = \{e_{i,i}^{h,h+1}\}$, $e_{i,i}^{h,h+1}$ 表示存储链路 $l_{i,i}^{h,h+1}$ 上的存储能耗, 其大小等于 En_i^c .

时延集合 $D^* = D \cup D_{\pi} \cup D_m$, 其中, D 是 TVG 中的时延集合; $D_m = \{t_{i,j^*}^h\}$, t_{i,j^*}^h 表示虚拟组播链路 l_{i,j^*}^h 传输时延, 其大小等于 $t_{i,j}^h$; $D_{\pi} = \{t_{i,i^*}^h\}$, t_{i,i^*}^h 表示虚拟链路 l_{i,i^*}^h 上的时延, 其大小为 0.

根据图 4 所构造的 MTVG, 可将节能问题 P1 转化为路径组合优化问题 P2.

$$\begin{aligned} \text{P2: Min } & \sum_{i=1}^N \sum_{j=1}^N e_{i,j}, l_{i,j} \subseteq \text{Tr} \\ \text{s.t. } & \begin{cases} \text{C1: } \sum_{i=1}^N \sum_{j=1}^N t_{i,j} \leq \Delta T_b^*, l_{i,j} \subseteq P_{g_z}^b \\ \text{C2: } \sum_{i=1}^N \sum_{j=1}^N t_{i,j} \leq \Delta T_{g_z}^b, l_{i,j} \subseteq P_{g_z}^b \\ \text{C3~C6} \end{cases} \\ & 1 \leq b \leq Q, 1 \leq z \leq Z, 1 \leq k \leq N, 1 \leq s \leq \mathfrak{J} \end{aligned} \quad (10)$$

式中, 目标函数是找到能耗接近最小的组播树 Tr , $e_{i,j}$ 表示 MTVG 中边 $l_{i,j}$ 上的能耗, $t_{i,j}$ 表示 MTVG 中边 $l_{i,j}$ 上的时延, $P_{g_z}^b$ 表示组播树 Tr 中源节点 v_s 发送组播业务 $X_b \in X$ 到目的节点 v_{g_z} 的路径, ΔT_b^* 表示组播业务 $X_b \in X$ 的有效时间, $\Delta T_{g_z}^b$ 表示目的节点 v_{g_z} 接收组播业务 $X_b \in X$ 的接收时延; 约束条件 C1、C2 分别表示满足组播业务的有效时间和各组播成员的接收时延; 约束 C3~C6 与 P1 问题相同.

5.2 组播树构建

为了找出满足组播业务时效性和各目的节点的接收时延的约束, 给出基于 KSP 的候选路径选择算法, 如算法 1 所示.

算法 1 基于 KSP 的候选路径算法

输入: 组播时变图 EC-MTVG = $(V^*, E^*, \varepsilon^*, D^*)$, 目的节点集合 V_G , 源节点 v_s , 业务 X_b , 业务时延约束 ΔT_b^* , 组播成员时延约束 $\Delta T_{g_z}^b = \{\Delta T_{g_1}^b, \Delta T_{g_2}^b, \dots, \Delta T_{g_Z}^b\}$

输出: 路径集合 $P_g = \{P_{g_1}, P_{g_2}, \dots, P_{g_Z}\}$, 路径能耗集合 $P_g^E = \{P_{g_1}^E, P_{g_2}^E, \dots, P_{g_Z}^E\}$, 路径时延集合 $P_g^T = \{P_{g_1}^T, P_{g_2}^T, \dots, P_{g_Z}^T\}$.

1. 利用 KSP 算法得到 $v_s \rightarrow v_{g_z}$ 的所有候选路径 L_z^k
2. 计算出所有候选路径 L_z^k 的能耗 $e_{i,j}^k$, 并按照从小到大的顺序加入到候选路径能耗集合 $P_{g_1}^E = \{e_1^1, e_1^2, \dots, e_1^k\}, \dots, P_{g_Z}^E = \{e_Z^1, e_Z^2, \dots, e_Z^k\}$
3. 根据候选路径能耗集合 $P_{g_z}^E$, 将相应的路径加入到候选路径集合 $P_{g_1} = \{L_1^1, L_1^2, \dots, L_1^k\}, \dots, P_{g_Z} = \{L_Z^1, L_Z^2, \dots, L_Z^k\}$
4. 利用式(11)计算出候选路径集合中所有路径的时延, 得到相应的候选路径时延集合 $P_{g_1}^T = \{t_1^1, t_1^2, \dots, t_1^k\}, \dots, P_{g_Z}^T = \{t_Z^1, t_Z^2, \dots, t_Z^k\}$
5. FOR 对于任意目的节点 v_{g_z} 的所有候选路径 L_z^k DO
6. IF $t_z^m \leq \Delta T_b^*$ 且 $t_z^m \leq \Delta T_b^*$
7. L_z^m 被保留在路径集合 P_{g_z} .
8. ELSE 将 L_z^m 从路径集合 P_{g_z} 中删除且相应的路径能耗和路径时延也从集合 $P_{g_z}^E, P_{g_z}^T$ 中删除 END IF
9. END FOR
10. RETURN P_g, P_g^E, P_g^T

在算法 1 中, 路径时延通过对路径上的时延进行累加可得, 计算式如下:

$$t_z^m = \sum_{i=1}^N \sum_{j=1}^N t_{i,j}, l_{i,j} \in P_{g_z}^m \quad (11)$$

式中, t_z^m 表示目标节点 v_{g_z} 的第 m 条候选路径的时延, $P_{g_z}^m$ 表示目标节点 v_{g_z} 的第 m 条候选路径集合. 由此, 算法 1 所得候选路径均满足组播业务时延 ΔT_b^* 和目的节点接收时延 $\Delta T_{g_z}^b$, 并且候选路径集合中的候选路径按照能耗进行排序. 由于每个卫星节点最多 4 条星间链路, 因此 KSP 算法的复杂度为 $O(n^2 \log_2^n)$, n 是节点总数, 再通过对所有候选路径进行路径时延约束判断, 则复杂度为 $O(KZ)$, K 为候选路径集合的大小, Z 为目的节点数目. 所以算法 1 的复杂度为 $O(n^2 \log_2^n + KZ)$.

通过算法 1 得到所有目的节点的候选路径集合 P_g , 以及相应的路径能耗集合 P_g^E 和路径时延集合 P_g^T . 然而, 由于每个目的节点选择不同的路径进行组合得到的组播树总能耗不同. 因此, 通过基于 MPH 的组播生成树算法进行迭代更新得到能耗接近最小的组播树. 基于 MPH 的组播生成树算法如算法 2 所示.

算法 2 基于 MPH 的组播生成树算法

输入: P_g, P_g^E, P_g^T

输出: 组播树 Tr , 组播树能耗 E_{Tr}

1. 选择每个候选路径集合中的第 1 条路径加入到初始组播树 $\text{Tr}^0 = \{L_1^1, L_2^1, \dots, L_k^1, \dots, L_Z^1\}$
2. 将初始组播树的每条边的能耗相加得到初始组播树的总能耗 E_{Tr^0}
3. FOR 从 v_{g_z} 中选择当前路径能耗最大的目的节点 v_{g_j} DO
4. FOR 从 v_{g_j} 的候选路径集合 P_{g_j} 中选择一条新的路径 L_j^y 替代之前的 L_j^i DO
5. 组播树 Tr^i 变为 Tr^y , 并计算组播树 Tr^y 的总能耗 E_{Tr^y}
6. IF $E_{\text{Tr}^y} < E_{\text{Tr}^i}$
7. $\text{Tr}^i = \text{Tr}^y, E_{\text{Tr}^i} = E_{\text{Tr}^y}$
8. ELSE 保留之前的路径 L_j^i, Tr^i 和 E_{Tr^i} 保持不变
9. END IF
10. FOR 节点集合 V 中的所有节点 v_i DO
11. IF 节点 v_i 的节点入度 $\text{in deg}(v_i) \neq 1$
12. 保留经过节点 v_i 的路径时延最小的目的节点的路径
13. END IF
14. 更新目的节点集合 G_{Tr} , 即 $G_{\text{Tr}} = G_{\text{Tr}} / v_{g_j}$
15. END FOR END FOR
16. IF $G_{\text{Tr}} \neq \emptyset$ END IF
17. END FOR
18. RETURN Tr, E_{Tr}

在算法 2 中, 将算法 1 得到的候选路径集合 P_g 以及相应的路径能耗集合 P_g^E 和路径时延集合 P_g^T 作为算法 2 的输入值. 第 1~2 行是初始化组播树, 将各目的节点的 P_g 中路径能耗最小的路径组合成初始组播树 $\text{Tr}^0 =$

$\{L_1^1, L_2^1, \dots, L_k^1, \dots, L_L^1\}$; 计算出组播树的总能耗 E_{Tr^0} . 第 3~15 行是更新组播树. 从当前路径能耗最大的目的节点 v_{g_i} 的路径集合中选择一条新的路径 L_j^y 加入组播树, 并且满足 $E_{Tr^y} < E_{Tr^0}$, 则组播树被更新为 Tr^y , 组播树能耗被更新为 E_{Tr^y} , 其中, 第 10~13 行是判断组播树的所有节点的入度是否为 1. 若节点 v_i 入度不等于 1, 则只保留包含节点 v_i 的路径时延最小的路径, 重新在被删除的目的节点路径集合中选择使当前组播树能耗最小的路径加入组播树. 第 14 行是更新目的节点集合 G_{Tr} . 若目的节点 v_{g_i} 的路径已经被更新, 则将目的节点 v_{g_i} 从目的节点集合 G_{Tr} 中删除. 第 16~17 行是判断目的节点集合 G_{Tr} 是否为空. 若 $G_{Tr} = \emptyset$, 则输出组播树 Tr , 若 $G_{Tr} \neq \emptyset$, 则重复执行第 3~15 行. 由于每个目的节点更新一次的复杂度为 $O(nK)$, 所有目的节点都需要更新 1 次, 因此算法 2 的复杂度为 $O(nKZ)$.

5.3 理论分析

所有目的节点到源节点满足约束条件的所有候选路径作为算法 2 的输入值, 并且将各目的节点到源节点能耗最小的路径构成初始组播树, 最后在此基础上进一步优化得到最终的组播树, 因此其算法性能应优于单播路由.

引理 1 算法组播路由能耗代价不大于单播能耗开销.

证明 在算法 2 中得到的初始组播树由各目的节点到源节点能耗最小的路径组成, 所构成的树为 Tr^0 , 其初始组播树能耗为 E_{Tr^0} .

$$E_{Tr^0} = \text{Min} \sum_{i=1}^N \sum_{j=1}^N e_{i,j}, l_{i,j} \in Tr^0 \quad (12)$$

当目的节点 v_{g_i} 选择新的路径 L_j^y 代替原来的路径 L_i^i 加入组播树, 新加入的边的总能耗 $e_{g_i}^+$, 被删除的边的总能耗 $e_{g_i}^-$, 除了目的节点 v_{g_i} 以外其他节点所组成的组播树的能耗为 $E_{G-v_{g_i}}$, 只有当满足 $e_{g_i}^+ - e_{g_i}^- < 0$ 时, 新的路径才会被选择加入组播树, 则

$$E_{G-v_{g_i}} + e_{g_i}^+ < E_{G-v_{g_i}} + e_{g_i}^- \quad (13)$$

因此, 当算法不断运行, 树的能耗也渐渐减少, 被更新后的组播树 Tr^y 与未被更新之前的组播树 Tr^i 之间的能耗总是存在如下关系: $E_{Tr^y} < E_{Tr^i}$, 则最终的组播树和初始组播树存在如下关系:

$$E_{Tr} \leq E_{Tr^0} \quad (14)$$

所以最终被优化后的组播树的能耗最低.

证毕.

引理 2 组播树生成树算法不会破坏其时延约束.

证明 通过算法 1 找到源节点到各目的节点的候选路径都是满足组播业务时延约束和各目的节点

的接受时延的. 将算法 1 得到的所有路径作为算法 2 的输入路径, 算法 2 再进行更新优化时所选的路径都是候选路径集合中的路径, 且候选路径集合中的路径均满足组播业务时延约束和各目的节点接收时延约束, 所以通过算法 2 得到能耗较小的组播树不会破坏其时延特性. 证毕.

定理 1 若满足时延约束的组播树存在, 则基于 MTVG 的组播路由算法就一定能找到一棵满足时延约束且能耗接近最小的组播树.

证明 基于 MTVG 的组播路由算法首先通过基于 KSP 的候选路径选择算法, 以满足组播业务有效时间和各目的节点的接收时延为前提, 寻找到每个目的节点到源节点的所有候选路径, 并且由每个目的节点的最小能耗路径构成初始组播树, 由引理 1 可知, 通过更新后的组播树比未更新之前的组播树能耗更低. 由引理 2 可知, 在进行组播树更新时不会破坏其相关时延约束条件. 因此生成的组播树是一颗满足时延约束的能耗接近最小的组播树. 证毕.

5.4 算法示例

为了清晰地阐述算法的运行过程, 构建如图 5(a) 所示的简单网络模型. 在图 5(a) 中, 将给定的时间范围 T 划分为两个时隙 τ_1, τ_2 , 且 $|\tau_1| = |\tau_2| = 60$ s; 组播业务 (10 bit, 0 s, 120 s), 在 τ_1 开始传输, 组播业务有效时间 $\Delta T^* = 120$ s, 目的节点 $v_{g_1}, v_{g_2}, v_{g_3}$ 的接收时延分别为 $\Delta T_{g_1} = 60$ s, $\Delta T_{g_2} = 70$ s, $\Delta T_{g_3} = 120$ s, 卫星节点的固定能耗和处理能耗为 3 J, 存储能耗 $En^c = 10$ J. 边的权重分别为能耗和时延, 例如边 (v_s, v_1^1) 上的权重 $[20, 1]$ 表示该链路的能耗为 20 J, 时延为 1 s. 通过算法 1 分别得到目的节点 $v_{g_1}, v_{g_2}, v_{g_3}$ 的候选路径, 如表 1 所示, 总候选路径集合 $P_g = \{P_{g_1}, P_{g_2}, P_{g_3}\}$, $P_{g_1} = \{L_1^1, L_2^1\}$, $P_{g_2} = \{L_2^2, L_2^3, L_2^4\}$, $P_{g_3} = \{L_3^1, L_3^2, L_3^3\}$; 总候选路径能耗集合 $P_g^E = \{P_{g_1}^E, P_{g_2}^E, P_{g_3}^E\}$, $P_{g_1}^E = \{e_1^1, e_1^2\}$, $P_{g_2}^E = \{e_2^1, e_2^2, e_2^3\}$, $P_{g_3}^E = \{e_3^1, e_3^2, e_3^3\}$; 总候选路径时延集合 $P_g^T = \{P_{g_1}^T, P_{g_2}^T, P_{g_3}^T\}$,

表 1 算法示例中各目的节点的候选路径

目的节点	候选路径	路径能耗/J	最大路径时延/s
v_{g_1}	$L_1^1 = \{v_u, v_1^1, v_1^1, v_1^1, v_{g_1}\}$	63	6
	$L_1^2 = \{v_u, v_1^1, v_1^1, v_1^1, v_2^1, v_3^1, v_{g_1}\}$	86	11
v_{g_2}	$L_2^1 = \{v_u, v_1^1, v_1^1, v_2^1, v_2^1, v_4^1, v_{g_2}\}$	76	7
	$L_2^2 = \{v_u, v_1^1, v_1^1, v_3^1, v_3^1, v_3^1, v_4^1, v_4^1, v_{g_2}\}$	79	66
	$L_2^3 = \{v_u, v_1^1, v_1^1, v_2^1, v_2^1, v_4^1, v_4^1, v_4^1, v_{g_2}\}$	89	67
v_{g_3}	$L_3^1 = \{v_u, v_1^1, v_1^1, v_3^1, v_3^1, v_3^1, v_4^1, v_4^1, v_{g_3}\}$	76	67
	$L_3^2 = \{v_u, v_1^1, v_1^1, v_2^1, v_2^1, v_{g_3}\}$	79	8
	$L_3^3 = \{v_u, v_1^1, v_1^1, v_2^1, v_2^1, v_4^1, v_4^1, v_4^1, v_{g_3}\}$	89	67

$$P_{g_1}^T = \{t_1^1, t_1^2\}, P_{g_2}^T = \{t_2^1, t_2^2, t_2^3\}, P_{g_3}^T = \{t_3^1, t_3^2, t_3^3\}.$$

以图 5(b) 举例说明, 通过 KSP 算法, 源节点 v_s 到目的节点 v_{g_1} 共有两条候选路径, 分别为 $L_1^1 = \{v_u, v_1^1, v_1^1, v_3^1, v_{g_1}\}$, $L_1^2 = \{v_u, v_1^1, v_1^1, v_2^1, v_2^1, v_3^1, v_{g_1}\}$, 分别计算每个目的节点的候选路径, 如图 5(c) 和图 5(d) 所示, 每条不同颜色的路径代表一条候选路径. 再将每个目的节点的最小路径能耗的路径组合

成一棵初始组播树, 如图 5(e) 所示, 初始组播树的总能耗为 132 J, 目的节点 v_{g_1} 、 v_{g_2} 、 v_{g_3} 的路径时延分别为 6、7、67 s, 然后通过算法 2 对初始组播树进行更新优化可以得到一颗能耗接近最小的组播树, 如图 5(f) 所示, 因此其组播树的能耗为 89 J, 目的节点 v_{g_1} 、 v_{g_2} 、 v_{g_3} 的路径时延 v_{g_1} 分别为 6、66、67 s. 通过算法 2 优化后的组播树比初始组播树节省了 43 J.

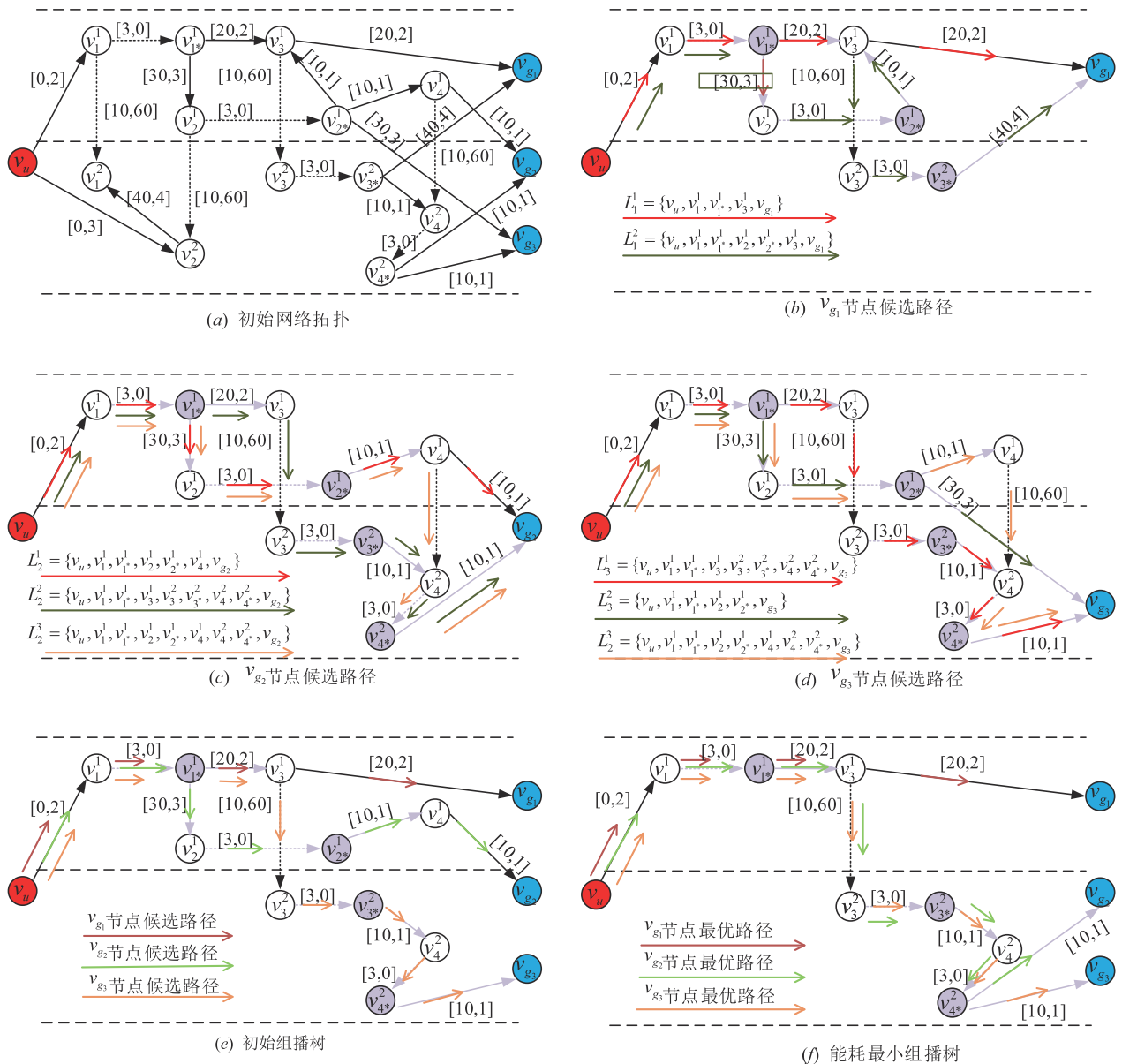


图 5 算法运行示例

6 仿真实验及结果分析

6.1 仿真场景及参数设置

本文的仿真场景由 288 颗低轨卫星和 8 个地面站构成. 其中, 低轨卫星均取之于 Walker 星座, 包含 12 个轨道面 (高度为 700 km, 倾角为 86°) 且每个轨道面均匀分布 24 颗卫星. 地面站分别设置在中国、美国、白令海、加拿大、巴西、澳大利亚、芬兰和尼日利亚. 利用卫星工具包 (STK 10, Satellite Tool Kit 10) 模拟器生成网络拓扑, 网络的拓扑结构高度动态变化, 然后利用 Exata 5.1 网络仿真软件实现所提的组播路由算法, 仿真时间段为 2023 年 6 月 6 日 00:00:00 至 2023 年 6 月 6 日 04:00:00, 并将时间段划分为相同大小的连续时隙, 时隙大小为 60 s, 设置的流量值为 5 MB, 每颗卫星的星间链路的传播速率和功率 15 Mbps 和 8 W, 星地链路的传播速率和功率为 12 Mbps 和 5 W, 每颗卫星的存储速率和功率设置为 8 Mbps 和 2 W. 此外, 星地链路和星间链路的传播时延可以通过 STK 10 获取.

6.2 仿真结果与分析

为了验证基于时变图的高效启发式组播路由算法的有效性与其可行性, 将所提算法与基于 TEG 组播路由算法、基于快照图的组播算法进行比较, 并对仿真结果进行分析讨论. 基于快照图的组播路由算法主要思想是选择满足时延界限的所有时隙, 并将不同时隙上的网络拓扑建模为一系列快照. 然后, 根据时间序列, 使用静态 DST 构建算法在最早的快照中构建接近最小能

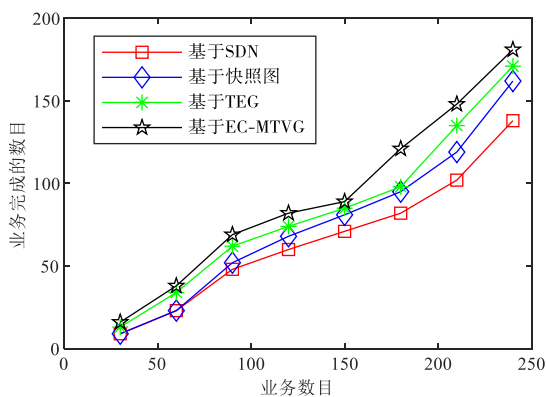
量组播树. 注意, 如果构造的树只能跨越部分目的地, 则有必要在后续快照中生成其他接近最小能耗组播树以跨越尚未覆盖的目的地, 直到到达所有目的节点.

(1) 评估指标

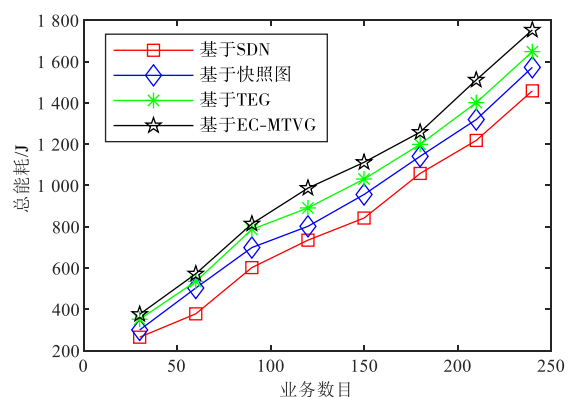
为了分析和评估所提算法的性能, 主要设置如下指标进行评估: ①算法服务能力, 组播业务完成的数目来衡量算法的服务能力; ②算法能量效率, 完成组播业务需要消耗的能量.

(2) 结果分析

图 6 在业务时延约束为 180 s 的情况下, 比较了业务数目对算法的服务能力和网络总能耗的影响. 从图 6 中可以看出, 随着业务数目的增加, 4 种算法所完成的业务数目和网络总能耗都呈现逐渐增加的趋势, 其中, 在相同的业务数目下, 所提算法所完成的业务数目最多, 所消耗的总能耗最大, 原因是所提算法充分考虑星间链路和星地链路的动态变化, 并联合不同时隙的存储和传输资源进行组播传输, 因此, 所完成的业务数目最多, 消耗的总能耗最多. 而基于 SDN 的组播路由算法由于星地和星间链路更新不及时, 所完成的业务数目较少, 网络总能耗较少. 基于快照图的组播路由算法由于只能在单个时隙内完成传输, 所完成的数目较少, 网络总能耗较低. 基于 TEG 的组播路由算法忽略了星地链路的动态变化使得组播成员的接入卫星发生变化, 导致完成的业务数目比所提算法更少, 网络总能耗更低, 网络能源效率更高.



(a) 不同业务数目下的业务完成数目



(b) 不同业务数目下总能耗

图6 业务数目对算法性能的影响

图 7 比较 4 种算法的服务能力. 从图 7 中可以看出, 随着组播业务数目的增加, 4 种算法能够完成的组播业务数目也逐渐增加. 但是无论组播业务数目为 30 个或 60 个, 基于 MTVG 的组播路由算法能够成功传输的组播业务数目大于基于 SDN 的分段路由组播算法、基于快照的组播

算法和基于 TEG 的组播路由算法, 原因是在时变的卫星网络环境中, 由于拓扑的断续连通, 单个快照图难以构建出端到端路径, 业务无法传输到目的节点, 导致组播业务完成数目较少. 而 SDN 控制器部署在地面, 网络拓扑更新不够及时, 此外, 由于基于 MTVG 的算法充分考虑了目的

节点对于组播业务时延的约束和星地链路的动态性,避免了业务传输超时和星地链路的改变导致组播业务传输失败,因此组播业务完成数目更多,算法服务能力更强.

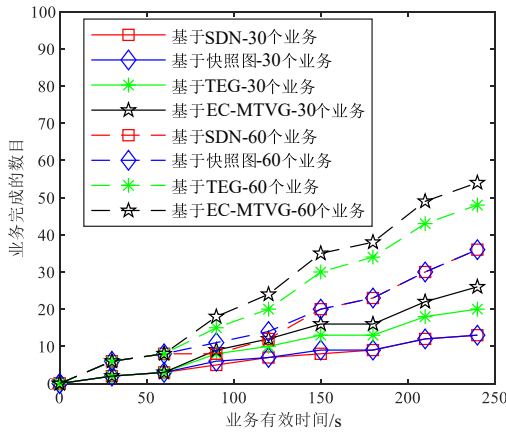
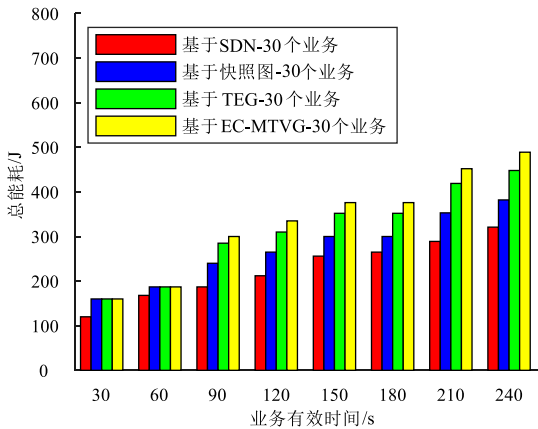
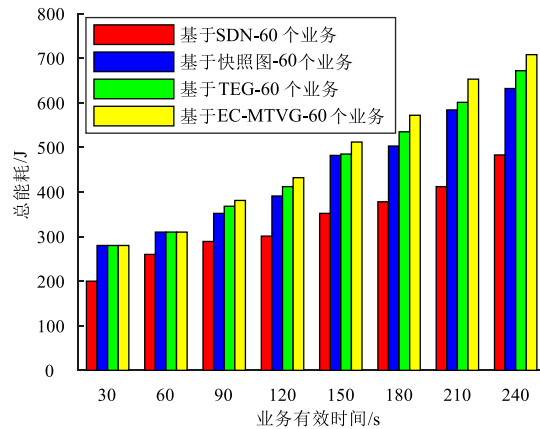


图7 组播算法的服务能力比较



(a) 30个业务的情况下总能耗



(b) 60个业务的情况下总能耗

图8 组播算法的能量效率比较

图9比较了4种组播算法中每个组播业务消耗的平均总能耗,即在相同的业务时延约束下,业务平均总能耗=总能耗/完成的业务数目.从图9可以看出,随着业务时延的增加,每个业务的平均总能耗逐渐减少,其中基于SDN的分段组播算法由于集中控制器不能及时更新网络拓扑导致所构造的组播树能耗较大,并且完成的组播业务较少,使得每个业务的平均能耗较高,基于MTVG的组播算法的业务平均总能耗最小,原因是所提算法充分利用了不同时隙内的存储和传输资源,同时还考虑到了星地链路的动态变化,所以构造的组播树是全局最优组播树,而基于快照的组播树由于只能在单个快照内找到局部最优组播树,导致业务的平均总能耗最大.

图8比较4种组播算法的能量效率.从图8可以看出,在组播业务时延约束相同的情况下,组播业务数目为60个时,4种算法所消耗的总能量更多,原因是随着组播业务的数目增加,能够完成传输的组播业务数目也逐渐增加,导致完成组播传输所消耗的总能量更多.而在组播业务数目相同的情况下,随着业务有效时间的增大,4种组播算法所消耗的总能量也逐渐增加,其中基于SDN的分段组播算法消耗的总能量最小,基于MTVG的组播算法消耗的总能量最多,原因是基于SDN的分段组播算法采用的是集中控制,并且控制器在地面,网络拓扑更新不及时,完成的组播业务数目较少,所消耗的总能量更低,而基于快照的组播路由算法只能在单个快照内构建组播树,所以相比其他两种组播路由算法完成的数目更少,所消耗的总能量较少,而基于MTVG的组播路由算法完成的组播业务数目最多,则消耗的总能量更多.

图10比较了同一组播业务在不同的时延约束情况下4种组播算法的能量效率.从图10可以看出,随着业务时延界限范围的增大,所提的基于MTVG的组播路由算法所消耗的总能量最少,比当前基于TEG的组播算法的网络总能耗平均降低了约10.2%.基于SDN的分段组播算法随着业务时延约束的增大,总能耗逐渐降低,原因是业务时延更大使得SDN控制器具有足够的时间更新网络拓扑,从而得到更优的组播树,而基于快照的组播路由算法只能在单个快照内找到局部最优的能耗树,基于TEG的组播路由算法由于星地链路的变化导致构造的组播树也是局部最优能耗树,而所提算法综合考虑到所有满足时延约束的快照和星地链路的变化,以接近最小能耗构造了全局最优的组播树.

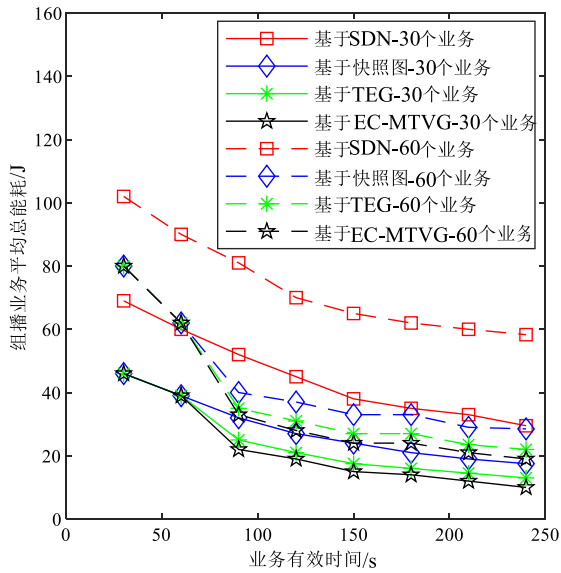


图9 每个业务的平均总能耗

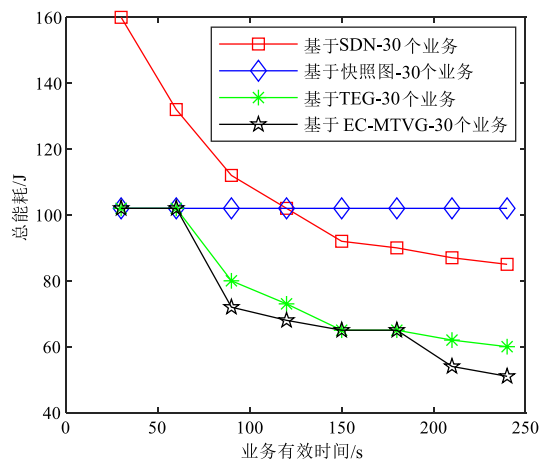


图10 单个业务下组播算法的总能耗的比较

7 结论

针对高动态的低轨卫星网络中时延有界的节能组播路由问题,提出了基于MTVG的组播路由算法,在满足业务时延约束和各组播成员接收时延约束的前提下,最大限度地减少卫星的能源消耗.首先,将卫星的能源消耗问题建模为节能优化问题;其次,利用MTVG模型对高动态卫星网络拓扑、链路和节点的多元属性进行刻画,在此基础上,将节能优化问题转化为路径组合优化问题,并提出基于KSP的候选路径选择算法找到满足QoS约束的所有路径,再通过基于MPH的组播生成树算法构造满足约束的接近最小能耗组播树,并证明了该算法的正确性;最后,仿真结果表明,所提路由算法通过联合利用存储和传输资源有效地降低了卫星网络能源消耗和提高卫星的服务能力.

参考文献

- [1] ZHU X M, JIANG C X. Integrated satellite-terrestrial networks toward 6G: Architectures, applications, and challenges[J]. IEEE Internet of Things Journal, 2022, 9(1): 437-461.
- [2] MA T, QIAN B, QIN X H, et al. Resource scheduling for high-capacity multicast service in ultra-dense LEO satellite networks[J]. IEEE Transactions on Vehicular Technology, 2024, 73(2): 2468-2481.
- [3] NI S, YUE Y. The status quo and prospect of satellite network routing technology[J]. Journal of Electronics & Information Technology, 2023, 45(2): 383-395.
- [4] SHUAI J C, LIU Y, WANG Y M. Energy efficient maximal throughput resource scheduling strategy in satellite networks[J]. IEEE Wireless Communications Letters, 2023, 12(2): 312-316.
- [5] MARCHESE M, PATRONE F. E-CGR: Energy-aware contact graph routing over nanosatellite networks[J]. IEEE Transactions on Green Communications and Networking, 2020, 4(3): 890-902.
- [6] ZHENG K J, SHEU J P. A dynamic multicast tree selection algorithm in LEO satellite networks[C]//2022 IEEE Global Communications Conference. Piscataway: IEEE, 2022: 1546-1551.
- [7] HU M L, YANG R H, HU Y, et al. QoS-aware software-defined multicast in LEO satellite networks[J]. IEEE Transactions on Aerospace and Electronic Systems, 2022, 58(6): 5307-5317.
- [8] HU M L, XIAO M, HU Y, et al. Software defined multicast using segment routing in LEO satellite networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(1): 835-849.
- [9] HU M L, LI J, CAI C, et al. Software defined multicast for large-scale multi-layer LEO satellite networks[J]. IEEE Transactions on Network and Service Management, 2022, 19(3): 2119-2130.
- [10] SHUAI J C, LIU Y, WANG Y M. Energy efficient maximal throughput resource scheduling strategy in satellite networks[J]. IEEE Wireless Communications Letters, 2023, 12(2): 312-316.
- [11] HE F, LIU Q, LV T, et al. Delay-bounded and minimal transmission broadcast in LEO satellite networks[C]//2016 IEEE International Conference on Communications (ICC). Piscataway: IEEE, 2016: 1-7.
- [12] LIU W Q, WANG Z Y, LV T, et al. Delay-bounded and minimal energy broadcast in satellite networks with multi-

- power levels[C]//2017 IEEE International Conference on Communications (ICC). Piscataway: IEEE, 2017: 1-6.
- [13] ZHANG M Q, LIU D H, REN W L, et al. Temporal network model based energy-efficient broadcast in satellite networks[C]//2019 11th International Conference on Wireless Communications and Signal Processing (WCSP). Piscataway: IEEE, 2019: 1-6.
- [14] SHI K Y, ZHANG X S, ZHANG S, et al. Time-expanded graph based energy-efficient delay-bounded multicast over satellite networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(9): 10380-10384.
- [15] WATEL D, WEISSER M A. A practical greedy approximation for the directed Steiner tree problem[J]. Journal of Combinatorial Optimization, 2016, 32(4): 1327-1370.
- [16] YANG Y, XU M W, WANG D, et al. Towards energy-efficient routing in satellite networks[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3869-3886.

作者简介



段 洁 女,1983年12月出生于四川省内江市. 现为重庆邮电大学通信与信息工程学院副教授. 主要研究方向为未来网络体系结构与关键技术、空天地一体化网络、网络安全技术.
E-mail: duanjie@cqupt.edu.cn



闫子豪 男,2001年5月出生于陕西省宝鸡市. 现为重庆邮电大学通信与信息工程学院硕士研究生. 主要研究方向为卫星网络、月面通信网络、路由技术.
E-mail: yanzihao0525@163.com