基于一致性哈希的低轨卫星网络无状态四层负载均衡器架构

刘映江[1]，潘　恬[1,2]，李浩南[1]，阮国皓[1]，郑长刚[3]，沈寓实[4]，黄　韬[1,2]

（1.北京邮电大学网络与交换技术全国重点实验室，北京 100876；
2.紫金山实验室，江苏 南京 211111；
3.北京空间飞行器总体设计部，北京 100094；
4.飞诺门阵（北京）科技有限公司，北京 100081）

摘要：随着低轨卫星由通信转发节点向在轨计算节点演进，多个星上后端实例需要通过四层负载均衡器分担动态业务负载。星上存储受限、接入卫星频繁切换以及后端资源和相对拓扑变化，使依赖连接表的地面方案面临较高的状态维护与迁移开销。针对以上问题，提出一种基于一致性哈希的星上每流无状态四层负载均衡器架构。地面网关为连接绑定哈希环版本，并在上行报文中携带版本信息；接入卫星依据五元组和固定版本的哈希环选择后端。通过历史环预迁移、新旧版本并存及后端计划排空，保持接入切换和计划扩缩容期间的连接亲和性，并采用IP-in-IP/直接服务器返回维持统一的虚拟IP服务入口。128颗卫星场景的模型级仿真结果表明，相较于等价多路径（ECMP）路由，所提方案平均时延、后端CPU负载标准差及丢包率分别降低10.4%、52.5%和72.7%，所评估切换窗口内未记录到长连接重置，且切换仅需迁移KB级哈希环数据。该架构将星上逐连接状态维护转化为服务级映射管理，在降低存储与迁移开销的同时，兼顾资源感知调度和连接保持。

关键词：四层负载均衡器；一致性哈希；星地切换；接入卫星；连接一致性

中图分类号：TP393

文献标志码：A

doi：10.11959/j.issn.2096−8930.tdyt2026033

A stateless L4 load balancer architecture for LEO satellite networks based on consistent hashing

Liu Yingjiang[1], Pan Tian[1,2], Li Haonan[1], Ruan Guohao[1], Zheng Changgang[3], Shen Yushi[4], Huang Tao[1,2]

1. State Key Laboratory of Networking and Switching Technology, BUPT, Beijing 100876, China
2. Purple Mountain Laboratories, Nanjing 211111, China
3. Beijing Institute of Spacecraft System Engineering, Beijing 100094, China
4. NovNet Computing System Technology Co., Ltd., Beijing 100081, China

Abstract: As low earth orbit (LEO) satellites evolve from communication relays into computing nodes, layer-4 load balancers are needed to distribute dynamic workloads across multiple onboard service instances. Limited onboard memory, frequent access-satellite handovers, and changing backend resources and topology make conventional connection-table maintenance and migration costly. This paper presents a per-flow stateless onboard load-balancer architecture based on consistent hashing. A ground gateway assigns each connection a hash ring version and carries it in uplink packets, while access satellites select the backend using the five-tuple and the corresponding immutable ring snapshot. Historical-ring pre-migration, coexisting versions, and planned backend draining preserve connection affinity during handovers and planned scaling. IP-in-IP with direct server return maintains a unified virtual-IP service endpoint. In a model-based simulation of a 128-satellite constellation, the proposed scheme reduces average latency, the standard deviation of backend CPU load, and the packet loss rate by 10.4%, 52.5%, and 72.7%, respectively, compared with ECMP. No resets of long-lived connec-

收稿日期：2026−05−17；**修回日期：**2026−08−17

通信作者：潘恬，pan@bupt.edu.cn

基金项目：国家自然科学基金资助项目（No. 62372053）

Foundation Item：The National Natural Science Foundation of China(No. 62372053)

tions are observed within the evaluated handover windows, and the volume of hash-ring data transferred during handovers remains at the kilobyte scale. By replacing onboard per-connection state maintenance with service-level mapping management, the architecture reduces memory and migration overhead while supporting resource-aware scheduling and per-connection consistency.
Key words: L4 load balancer, consistent hashing, satellite-ground handover, access satellite, per-connection consistency

0　引言

低轨卫星网络的服务能力正从提供广域连接向星地协同计算延伸。星地融合边缘计算和在轨数据处理等研究表明，卫星正在从连接地面用户与地面服务器的转发节点，逐步发展为能够在轨承担数据预处理、内容缓存和业务计算的服务节点[1-2]。当同一服务部署在多颗计算卫星上时，这些分散的实例构成可统一访问的后端资源池。网络因此需要在完成报文传输的基础上，进一步根据业务负载和资源条件，选择合适的后端计算实例处理连接请求。

这一需求与地面云数据中心的四层负载均衡具有相同的出发点：客户端访问稳定的虚拟IP地址（virtual IP, VIP），四层负载均衡器（layer-4 load balancer, L4LB）将连接分配给多个真实服务器（real server, RS），根据后端处理能力分担业务负载，并支持服务的横向扩缩容[3-4]。在星上计算场景中，地理热点迁移、业务到达波动和计算实例增减同样会引起负载变化。仅由接入卫星的本地实例处理请求，容易出现部分卫星过载、其他计算卫星空闲的资源失衡情况；由客户端直接选择RS，则会增加终端发现后端和适应资源变化的复杂度。因此，在星上服务入口引入L4LB，有助于将统一的业务访问与动态的后端资源池连接起来，使流量分配与各RS的处理能力相适应。

将这一能力部署到卫星上，需要应对低轨星座运行环境带来的新挑战。首先，连接表的存储、查找与同步开销随并发连接数的增长而增加，而星上计算和存储资源相比地面节点受到约束；其次，地面分区对应的接入卫星随过顶关系周期性变化，承载L4LB的物理节点也随之迁移[5]，逐连接状态迁移会使控制流量和切换准备时间随业务规模的增大而增加；再次，接入卫星变化会改变其到各RS的相对距离，即使后端成员保持不变，面向新连接的调度权重也需要更新；最后，后端扩缩容进一步要求系统协调两类需求，即让新连接及时利用更新后的资源，同时使存量连接继续到达原RS。

本文面向具备可控地面网关、星间可达路径和星上服务实例的场景，将L4LB部署在地面业务进入星间网络的接入卫星上。接入卫星结合当前入口到各RS的距离和后端处理能力完成服务选择，地面网关负责维护携带流上下文。两者协同，使业务能够从当前接入点进入星上资源池，减少经固定负载均衡锚点转发的绕行。该架构的设计重点，是在接入卫星持续变化的条件下，以较低的星上状态开销兼顾调度适应性与连接保持。

一致性哈希能够通过局部调整映射区间，减少后端成员变化带来的映射扰动[6]。在此基础上，本文引入版本机制，将逐连接的后端绑定转化为“连接绑定版本、版本确定后端映射”：网关为连接选定环版本并逐包携带，接入卫星维护服务级哈希环和相应历史版本，以复现连接建立时的后端选择。本文的主要工作包括构建资源—拓扑联合加权哈希环，设计接入切换时的版本预迁移与历史环查询机制，利用新旧版本并存支持后端实例计划扩缩容，以及形成完整的请求与响应转发流程。仿真实验从调度性能、星上存储、切换迁移和连接扰动等方面，评价方案在负载均衡与连接保持之间的综合表现。

1　背景与挑战

1.1　低轨卫星星上四层负载均衡器的需求

本文考虑由地面终端、地面站或网关、低轨接入卫星、星间链路和后端计算卫星构成的服务系统。一个VIP对应同一业务的多个RS，每个RS具有可路由的直接IP地址（direct IP, DIP），并提供兼容的业务接口。客户端通过传输控制协议（transmission control protocol, TCP）或用户数据报协议（user datagram protocol, UDP）访问VIP。服务编排负责部署实例并提供可用RS集合，L4LB在此基础上为进入系统的连接选择后端，网络层完成到目标RS的路径转发。面向低轨边缘计算的虚拟试验平台为这类星地协同服务的软件验证提供了基础[7]。

L4LB以连接为调度单位，通过传输层信息完成后端选择。同一TCP连接的后续报文需要持续到达保存其序列号、接收窗口和会话上下文的RS[8]，这种连接亲和性是维持业务连续性的基础。因此，本文以新连接分配和资源版本更新适应负载变化，并为已建立连接保留原有映射。对于长连接占比较高、单连接承载大量任务的业务，还可结合应用层多连接、任务切分和上层调度，提

高资源池利用的灵活性。

地面区域通过服务分区 pid 标识。每个分区在给定接入时段内由一颗或少数几颗卫星提供入口，控制平面根据星历和可见性预测接替关系。服务、分区和环版本共同标识连接所属的逻辑入口，当前接入卫星则承载这一入口的转发功能。通过将逻辑身份与物理承载位置分离，系统能够在接入卫星切换时延续原有连接上下文，同时为新连接采用适合当前资源与拓扑条件的映射。

1.2 地面云数据中心四层负载均衡器的实现

地面 L4LB 通常先根据 VIP 和端口识别服务，再为连接选择 DIP。常见实现通过连接表记录“五元组—RS”关系，并采用网络地址转换（network address translation, NAT）、IP 隧道或直接服务器返回（direct server return, DSR）组织收发路径[3,9-10]。其中，连接表维持后续报文到同一 RS 的绑定，封装或地址改写负责将请求送达后端，并使响应向客户端呈现统一的 VIP。连接映射与报文转发相互配合，共同构成 L4LB 的完整处理流程。

确定性散列是构建连接映射的常用方法。等价多路径（equal-cost multi-path, ECMP）路由根据报文字段散列选择等价下一跳[11]；L4LB 可借鉴其逐流散列方式，在给定 VIP 对应的 RS 集合中构造均匀分配基线。Rendezvous 哈希计算连接与各后端的联合散列得分，并选择得分最高的节点[12]；Maglev 哈希则通过预构建查找表实现快速、确定性的后端选择[13]。在完整系统中，这些映射方法还可与连接跟踪等机制配合，以适应运行期间的配置变化。

后端资源池会随业务需求进行扩缩容，因此，L4LB 不仅需要分散业务流量，还需要尽量减少后端集合变化对已有连接映射的扰动。一致性哈希将哈希空间组织为首尾相接的环，把连接键和后端标识映射到环上，沿约定方向选择连接位置之后的第一个节点。后端增删时，映射变化集中在部分区间，从而减少直接对节点数取模所带来的大范围重映射[6]。为平滑随机位置分布，一个 RS 可对应多个虚拟节点；调整虚拟节点数，能够近似控制该 RS 获得的流量份额。

1.3 星上四层负载均衡器的挑战

1) 星上大容量连接表存储。高并发连接需要保存五元组和 DIP，还涉及计时器、状态老化、并发访问与副本同步。随着业务规模的扩大，这些持续维护开销会占用更多星上资源，并增加接入切换时的状态迁移负担。因此，需要合理划分地面网关、接入卫星和控制平面的状态职责，将星上状态规模与活跃连接数解耦，使有限资源主要用于服务级映射和数据转发。

2) 星上 L4LB 放置点选择。固定锚点有利于保存连接上下文，但可能增加绕行距离，并集中承受业务流量；接入卫星贴近当前星上入口，却会周期性变化。本文采用接入卫星承载 L4LB，以服务级映射迁移替代逐流结果迁移，兼顾入口位置与切换开销。多个入口同时服务同一分区时，还需要协调版本发布，使各入口共享一致的命名空间和已提交映射，为跨入口处理连接提供统一依据。

3) 连接首包后端 RS 选择。后端 CPU 能力存在差异，接入卫星到各 RS 的路由跳数也随拓扑变化而变化。按连接数量平均分配，容易使低算力节点承受较大负载；只选择最近 RS，则可能造成局部资源集中使用。因此，首包选择需要同时考虑后端计算能力与星间转发代价。本文以 CPU 核数和路由跳数为基础指标，通过联合加权，使连接份额与资源能力及接入位置相匹配。

4) 星地拓扑变化时的连接一致性维护。接入切换后，新入口需要根据当前拓扑为新连接调整后端选择，同时复现存量连接的首包映射。持续时间较长的连接可能经历多个接入周期。因此，入口之间需要传递覆盖这些连接的历史映射。已有低轨切换研究关注链路层与网络层的接入连续性[14]，本文进一步研究服务层后端映射的连续性，使路径和入口变化期间的报文仍能到达原 RS。

5) 后端 RS 变动时的连接一致性维护。扩容要求新增 RS 逐步分担业务，计划缩容要求待退出 RS 有序释放资源。与此同时，已有 TCP 连接仍需由原 RS 处理。系统因此需要协调新连接接纳、存量连接排空与历史版本回收，使资源池调整平稳地反映到后续流量分配中。对于异常退出，则需要将后端集合更新与业务恢复机制协同组织。

2 星上无状态四层负载均衡器

2.1 设计理念与总体架构

面向星地协同服务模型，设计需要同时满足连接亲和性、低星上状态开销、资源—拓扑感知、协议可实现性和低扰动扩缩容 5 个目标。这些目标并非彼此独立：为适应资源变化而更新调度规则，可能改变已有连接的后端；为保留这些选择而维护连接表，又会增加星上存储与切换迁移开销。因此，设计的核心是在允许入口和资源池变化的同时，以较小的共享状态延续连接与 RS 之间的关系。

首先，连接保持需要的是能够再次找到首包选定的 RS，而不一定是在星上逐条保存选择结果。对于同一五

元组，只要使用相同的哈希环和哈希规则，后续报文就能复现原有选择。由此，星上可以从维护“每条连接的结果”转为维护“多条连接共享的映射规则”，把状态组织在服务与分区层面。这也是选择确定性哈希作为基础的原因。

但共享规则不能一直固定。后端CPU能力不同，单纯均匀分配难以匹配各RS的处理能力；仅按算力分配又忽略了当前入口到后端的转发距离。因此，映射规则需要同时反映CPU能力和路由跳数，并随资源配置与接入位置变化更新。这里产生的关键矛盾是：新连接需要采用更新后的规则，而已有连接需要延续建立时的选择。一致性哈希能够减少更新时受影响的区间，但仍需为其中的存量连接保留旧映射。本文据此将连接绑定到环版本：新环接纳新连接，旧环继续服务原有连接。

版本确定后，还需要解决“谁记住连接所属版本”的问题。地面网关是上行流量进入卫星网络的可控入口，可以维护流版本缓存并将版本写入隧道元数据；接入卫星据此选择共享哈希环，而不再保存逐连接绑定。这样，流上下文的维护留在地面，映射计算和业务转发放在当前接入卫星。客户端仍以普通TCP/UDP方式访问VIP，后端通过隧道和VIP配置接收请求，保持原有协议语义。

接入卫星切换窗口和星间带宽进一步决定了迁移方式。若在切换时传递完整连接表，准备开销仍会受并发规模影响；利用星历预测接替关系，提前传递该服务分区的当前环和仍在使用的历史环。迁移单位因此是“服务—分区—版本”，而不是单条连接。接替卫星读取报文版本后即可复现原有映射，存量连接使用的历史环在后续切换中继续保留。对后端计划扩缩容，同样沿用新旧流分离的处理方式，使入口变化与资源变化能够共用一套连接保持机制。

基于上述思考，本文形成如图1所示的星地协同架构。地面网关负责流版本标记，接入卫星负责查环与转发，后端计算卫星RS负责业务处理；控制平面提供后端资源信息和接入卫星接替关系，并协调哈希环的更新与迁移。表1列出了各类状态的维护位置、规模特征及用途。

1) 流版本标记与SSTH。连接首次经过地面网关时，网关为其记录当前已启用的ring_version和首包接纳时间flow_start_time；该连接的后续上行报文继续携带相同信息。服务标识、地面分区和版本信息统一放入卫星业务隧道头SSTH，字段及写入位置见表2。接入卫星根据服务和分区找到相应环集合，再按版本号选环。

SSTH由网关写入地面至接入卫星的入口隧道，也可采用Geneve扩展选项承载[15]。其中，ring_version直接确定连接使用的哈希环，flow_start_time保留连接建立时的

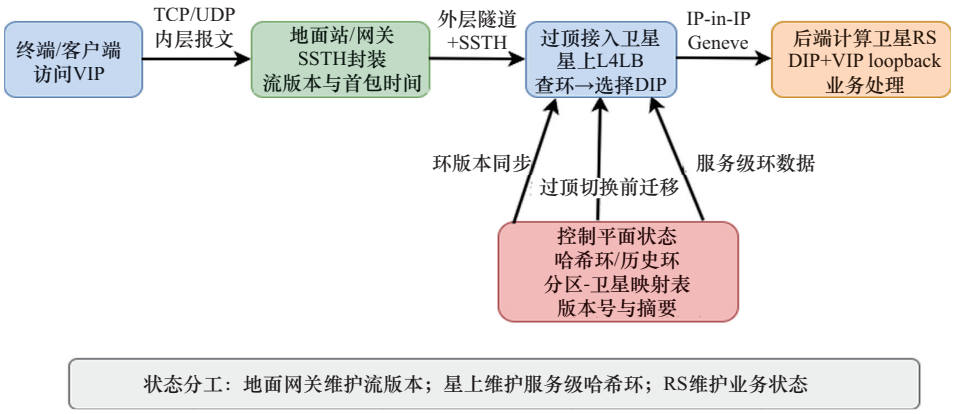

图1　星上每流无状态L4LB总体架构与状态分工

表1　每流无状态架构中的状态分工

状态类型	星上维护	规模特征	主要用途
五元组连接表 / 每连接NAT项	否	不占用星上连接表	通过五元组和环版本计算RS，不维护逐连接的后端绑定及老化计时器
当前哈希环	是	服务、分区数量及每环虚拟节点数	为新建连接选择后端RS
历史哈希环	是	服务、分区数量及历史环总规模	在切换或扩缩容后，复现已有连接的后端选择
分区—卫星映射表	是	随地面分区数的增加而增长	确定接替卫星，支持提前迁移哈希环
环版本、时间戳与摘要	是	随服务和版本数量的增加而增长	用于版本查找、迁移校验和历史环回收
地面侧流版本缓存	否	随地面活跃连接数的增加而增长	使后续上行报文继续携带首包绑定的环版本

表 2　SSTH 元数据头关键字段及写入位置

字段	建议长度	写入位置	用途
service_id / VIP	32 bit 或 128 bit	地面站/网关	标识被访问服务，可由服务 ID 或 VIP 表示
pid	16 / 32 bit	地面站/网关	标识地面服务分区，定位该分区的哈希环
ring_version	32 bit	地面站/网关	记录首包绑定的环版本，作为后续报文选环的依据
flow_start_time	64 bit	地面站/网关	记录首包接纳时间，支持兼容时间查询和历史环回收
flags / next_proto	16 bit	地面站/网关	标识扩展头类型、内层协议和相关事件
header_checksum	16 bit	地面站/网关	检测元数据差错，避免版本号或时间戳误读

时间信息，兼容时间查询方式见第 2.3 节。接入卫星读取这些字段完成后端选择后，再将客户端报文封装至 RS。SSTH 位于外层，不参与内层 TCP/UDP 校验和的计算，其增删不会改变内层传输层校验和。版本标记与隧道处理均由网络侧完成，客户端和 RS 继续使用原有 TCP/IP 协议。

2) 请求与响应转发。默认采用 IP-in-IP/DSR 模式，如图 2 所示。接入卫星读取 SSTH 和内层五元组，查询指定版本的哈希环，得到目标 RS 的 DIP。随后以 DIP 为外层目的地址重新封装请求，内层目的地址仍保持 VIP，源地址、端口及传输层校验和不变。IPv4 的生存时间（TTL）及 IP 头校验和按正常转发规则处理。RS 完成解封装后，通过本地配置的 VIP loopback 接收并处理业务[10]。

响应方向由 RS 以 VIP 为源地址返回地面网关，再送达客户端。客户端始终通过同一 VIP 访问服务，响应报文可沿可用路径返回，无须经过原 L4LB 实例。部署时在 RS 上配置 VIP、隧道终结与返回路由，即可将星上后端选择与常规传输协议衔接起来。

3)静态 NAT 兼容模式。对于响应需要经过指定锚点的部署，可预先配置服务级 VIP 与后端地址或端口的对应关系。请求方向按所选 RS 完成 VIP 到 DIP 的转换，响应方向经返回锚点恢复 VIP，并根据地址或端口的改写，增量更新 IPv4 头校验和及受影响的 TCP/UDP 校验和[16]。各服务采用可区分的后端地址或端口，使双向转换由服务级静态表完成。该模式作为可选配置，不改变默认的 IP-in-IP/DSR 转发方式。

2.2　基于 LB-RS 距离的接入卫星一致性哈希环构建

哈希环的构建同时考虑后端计算能力和星间转发距离。设某服务的可用后端集合为 $S=\{S_1,S_2,\cdots,S_n\}$，节点 S_i 配置的 CPU 核数记为 CPU_i，当前接入卫星到该节点的最短路由跳数记为 Hop_i。接入卫星通过控制平面或路由信息获取这两项指标，分别衡量后端处理能力和转发代价。

为组合不同量纲的指标，先进行归一化。CPU_i 除以候选后端中的最大 CPU 核数 $\mathrm{CPU}_{\max}$，得到计算能力指标 c_i；路由跳数按最大值 $\mathrm{Hop}_{\max}$ 和最小值 $\mathrm{Hop}_{\min}$ 进行反向归一化，得到距离指标 h_i：

$$c_i = \frac{\mathrm{CPU}_i}{\mathrm{CPU}_{\max}} \tag{1}$$

$$h_i = \begin{cases} 1, & \mathrm{Hop}_{\max} = \mathrm{Hop}_{\min} \\ \dfrac{\mathrm{Hop}_{\max} - \mathrm{Hop}_i}{\mathrm{Hop}_{\max} - \mathrm{Hop}_{\min}}, & \text{其他} \end{cases} \tag{2}$$

当各 RS 的跳数相同时，距离指标统一取 1；当跳数不同时，距离越近的 RS，其 h_i 越大。综合权重由两项指标线性组合得到：

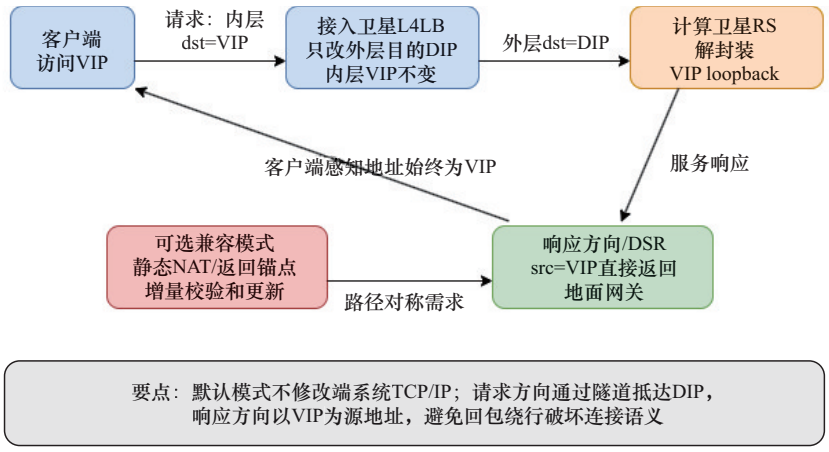

图 2　L4LB 请求转发与 DSR 响应路径

$$W_i = \alpha c_i + \beta h_i,\ \alpha \geqslant 0, \beta \geqslant 0, \alpha + \beta = 1 \tag{3}$$

其中，α和β分别表示对计算能力和较短转发距离的偏好，默认取α=0.8、β=0.2。权重较大的RS在当前接入位置下更适合承担新连接。

设基准虚拟节点数为V_{base}，所有后端权重之和为$W_{\text{total}}=\sum_j W_j$，则节点$S_i$分配的虚拟节点数为：

$$v_i = \left\lceil \frac{V_{\text{base}} W_i}{W_{\text{total}}} \right\rceil \tag{4}$$

式(4)按权重比例分配虚拟节点，并向上取整。权重较大的RS获得更多虚拟节点，从而提高其被新连接选中的概率。实际环大小以各RS分配到的虚拟节点总数为准。

随后，将RS标识与虚拟节点序号一同散列到环上，并按位置排序。报文到达时，把连接五元组映射到同一哈希空间，沿顺时针方向找到第一个虚拟节点，其所属RS即为目标后端。各接入卫星对同一版本使用相同的环内容和哈希规则，因此能够从同一五元组得到相同的选择结果。

与仅按CPU能力分配不同，本文的距离权重会随接入位置变化。同一RS在当前入口附近，在下一颗接入卫星看来可能较远。因此，拓扑或资源变化后，新构建的环用于调度新连接，已有连接仍查询建立时绑定的环。历史环保持原有内容，避免权重调整影响存量连接。

2.3　过顶接入卫星切换时的连接一致性保障

接入卫星切换既可能改变业务进入星间网络的位置，也可能改变后端选择的距离权重。为使已有连接仍到达原RS，本文将每个服务分区的哈希环按版本保存，并在切换前迁移至接替卫星。迁移对象是当前环和仍被连接使用的历史环，而不是逐条连接记录。

每个版本保存服务与分区标识（service_id、pid）、版本号（ring_version）、新连接接纳时间区间（valid_from、valid_to）、环内容（ring_body）和校验摘要（ring_digest）。其中，valid_from与valid_to分别表示开始和停止接纳新连接的时间，区间采用前闭后开形式。该区间用于确定连接首包所属的版本，历史环仍按存量连接的持续时间保留。控制平面协调版本发布，使不同入口对同一版本采用相同映射。旧接入卫星根据预测接替关系提前发送环数据，新接入卫星完成接收和校验后开始承接流量。主要过程见表3。

例如，某连接建立时使用版本v10并被分配到RS2。接入卫星切换后，后续报文仍携带v10，新入口查询迁移过来的v10环，即可再次选出RS2。即使新入口已根据当前位置构建v11环，它也只用于后续新连接。这样，入口位置可以变化，同一连接的后端选择则保持连续。

报文优先按ring_version匹配相应版本的哈希环。对于兼容部署中未携带版本号的报文，可在同一服务和分区的版本记录中，根据valid_from≤flow_start_time<valid_to匹配对应环。时间查询使用与地面网关一致的版本启用记录，只有匹配结果唯一时才采用；否则先由网关补齐版本信息。所需历史环暂不可用时，先补齐该版本再恢复转发，保持原有映射。该处理仍以共享哈希环为单位，不增加星上逐连接表。

历史环的保留时间应覆盖所支持连接的持续时间，并为切换和在途报文留出余量。一个版本停止接纳新连接后，仍用于处理此前绑定的连接，待这些连接结束且相关报文处理完成后，由控制平面依据地面汇总的活动引用回收。长连接经历多次切换时，继续保留其引用的版本即可，无须逐次迁移该连接的记录。

2.4　后端RS横向扩缩容时的连接一致性保障

后端扩缩容及相应的权重调整会改变哈希环上的映射关系。为避免这些变化影响已建立连接，本文采用新旧版本并存的方式：新环反映更新后的资源池，旧环延续已有连接的后端选择。

扩容时，将新增RS加入候选集合，按第2.2节的方法重新计算权重和虚拟节点数，生成新版本环。新RS和新环准备完成后，地面网关为新连接写入新版本号；存量连接仍携带旧版本号并查询旧环。随着新连接到达，新增RS逐步分担业务，而已有连接继续由原RS处理。

计划缩容时，先在新环中移除待退出RS，使新连接分配到其余后端；旧环及其中的后端映射暂时保留，该RS继续处理原有连接。待存量连接结束并完成报文处理后，再关闭对应实例、回收不再使用的历史版本。这一过程通过连接自然结束完成资源退出，不需要在星上迁

表3　接入卫星切换时的哈希环处理

阶段	接入卫星的处理	地面网关的处理
切换前	旧入口向接替卫星迁移当前环和仍在使用的历史环	保留各连接已有的版本标记
切换时	新入口完成接收和校验后，按报文版本查询相应环	将上行报文送往新入口，已有连接的版本不变
切换后	保留旧连接使用的环，并为新连接准备适合当前拓扑的环	新连接绑定已启用的新版本，旧连接继续携带原版本

移逐连接状态。

双版本描述一次更新中新连接与旧连接的分工。若多次更新期间仍有长连接存续，则保留相应的多个历史版本，待连接结束后逐步回收。相比后端纠错转发等连接保持方式[17]，本文利用报文版本和历史环在接入侧复现原有选择，使扩缩容与接入切换沿用同一套处理逻辑。后端故障与会话恢复的协同处理见第4.2节。

3　实验评估

本文采用Python实现模型级仿真，比较不同调度方式下的连接映射、后端负载与接入切换开销。实验使用10个随机种子，报告平均值和95%置信区间。星座采用8条极地轨道、每轨16颗卫星的LEO128构型，仿真星座与业务参数见表4。各时间片依据接入卫星到RS的最短跳数更新权重；业务随机到达，单流需求服从对数正态分布，用于表示不同连接的负载差异。

表4　仿真星座与业务参数

参数	取值
星座规模	128颗；8条轨道，每轨16颗卫星
轨道倾角 / 高度	90° / 1 450 km
轨道周期	约6 894 s
后端RS	S115、S206、S214、S308、S502、S608
CPU核数(依上列顺序)/核	8、16、12、8、24、16
权重参数	α=0.8，β=0.2
随机种子数 / 时间粒度	10个 / 100 s
业务模型	随机到达；单流需求服从对数正态分布

比较对象包括ECMP式均匀散列、普通一致性哈希（CH）、Rendezvous哈希、Maglev哈希、有状态L4LB和Beamer类基线。各方案以相应调度和状态维护机制的模型参与比较，图中“Maglev”和“Beamer类”沿用这一模型含义。评估围绕第2.1节的设计目标展开：通过星上存储和切换迁移量观察共享映射的成本，通过时延、CPU负载、吞吐量与丢包观察联合权重的调度表现，通过切换重置率和扩缩容重映射比例观察连接保持效果。协议开销则按字段及转发链路分段核算。

3.1　星上存储开销

图3比较连接表和哈希环的存储开销。有状态L4LB若维护1 000万条IPv4连接状态，按每条约17 B的压缩估计，仍需约170 MB；如果考虑计时器、老化队列和同步元数据，实际开销通常更高。相比之下，本文方案在1 024个虚拟节点和5跳历史环设置下，哈希环的存储量为KB级，版本与时间戳元数据为字节级。该结果说明，本文方案将星上状态规模与连接数解耦，适合高并发但星上内存受限的场景。

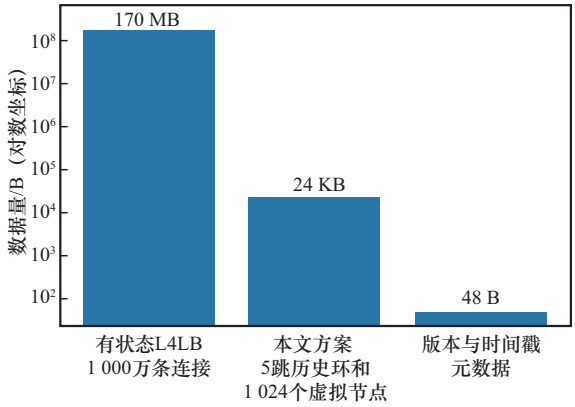

图3　星上存储开销对比

3.2　协议封装与元数据开销

本文方案引入的额外开销包括隧道封装带来的每包传输开销，以及地面网关维护版本绑定所需的每流存储开销。图4展示隧道头和地面流版本缓存两类开销，蓝色部分对应B/包的隧道头传输开销，橙色部分对应B/流的流版本状态存储开销，两者按各自量纲分项读取。为进一步呈现真实转发路径中的成本，表5按地面网关至接入卫星、接入卫星至RS等链路段分别核算新增头部。报文线上开销据该表计算，地面版本缓存则计入逐流存储。

以表2中的32 bit服务ID和32 bit分区ID为例，6项SSTH逻辑字段合计24 B，具体承载隧道、选项描述、对齐和认证字段按部署方式计入。接入卫星消费元数据并完成后端选择后，默认IPv4 IP-in-IP转发增加20 B外层头。采用Geneve时，IPv4、UDP和Geneve基础头合计

表5　分链路段的头部开销核算

链路段 / 模式	新增线上字节	核算说明
地面网关→接入卫星	SSTH逻辑字段24 B起，另加承载头	按32 bit服务ID和pid计算；承载、对齐及认证另计
接入卫星→RS：IPv4 IP-in-IP	20 B / 包	按无选项IPv4头计算；入口完成SSTH处理
IPv4 / UDP / Geneve基础封装	36 B / 包起	基础头20+8+8 B；扩展选项与其他外层开销另计
静态NAT地址改写	地址改写本身增加0	按实际返回隧道或锚点配置核算新增头部
地面每流版本缓存	单独计入存储	按B / 流核算地面版本缓存

36 B，扩展选项及可能使用的内层以太网头另行核算[10,15]。

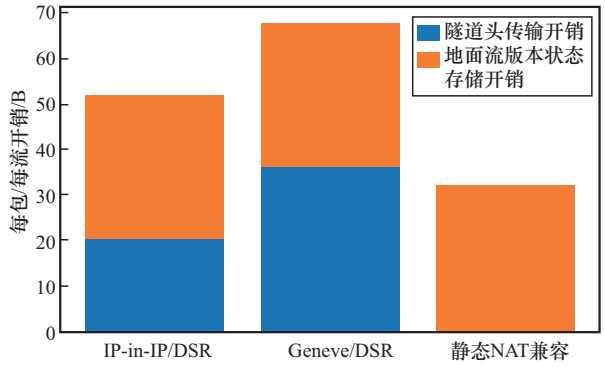

图4　隧道头与地面每流元数据开销示意

隧道部署还需为新增头部预留路径最大传输单元（MTU）空间，并正确处理互联网控制报文协议（ICMP）差错与分段[18]。为使接入卫星获得完整的五元组，可结合隧道MTU配置和路径MTU发现，减少入口处需要额外分类的IP分片；对于已有分片，由地面侧按统一规则处理后送入入口隧道。这样可将报文分类与封装配置相协调，保持星上数据面以元数据读取和哈希环查询为主的处理流程。

3.3　转发时延与负载均衡

随机流量下平均时延随时间变化情况如图5所示。随着星座运动，接入卫星到各RS的相对距离周期性变化，各方案的时延也随之波动。本文方案在图示时段保持较低时延。相比均匀散列，联合权重同时考虑后端CPU能力和入口到后端的距离，使新连接更倾向于分配到处理能力较强、转发代价较低的RS。曲线表现与这一调度目标一致，反映了在当前模型下联合考虑资源和拓扑的收益。

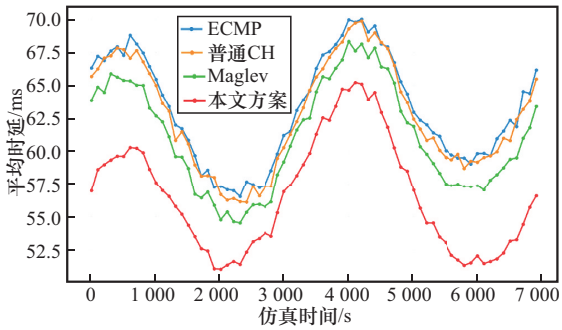

图5　随机流量下平均时延随时间变化情况

后端CPU负载标准差随时间变化情况如图6所示，用于观察负载在各RS之间的分布。对于异构资源池，需要结合节点处理能力理解其负载水平。评估结果显示，本文方案相较ECMP的负载标准差降低52.5%，时间序列也显示各后端之间的负载差异相对较小。联合权重通过调整虚拟节点数量，使高算力RS承担更大流量份额；有限哈希粒度、连接大小差异和存量映射的持续保留，则共同影响实际分配的细粒度波动。

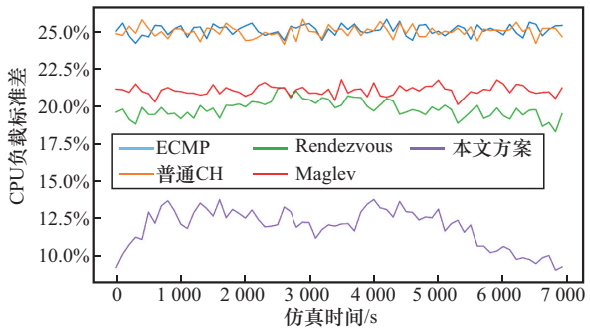

图6　后端CPU负载标准差随时间变化情况

图7和图8分别给出平均时延与99分位时延。本文方案的平均时延为56.61 ms，低于ECMP的63.20 ms，降幅约10.4%；普通CH、Rendezvous和Maglev基线分别为62.69 ms、60.78 ms和61.04 ms。与有状态L4LB及Beamer类基线相比，本文方案的平均时延接近。结合其服务级状态维护方式，这一结果表明，在所评估场景下，方案能够以较低的星上状态维护开销，获得与有状态L4LB及Beamer类基线接近的平均时延表现。

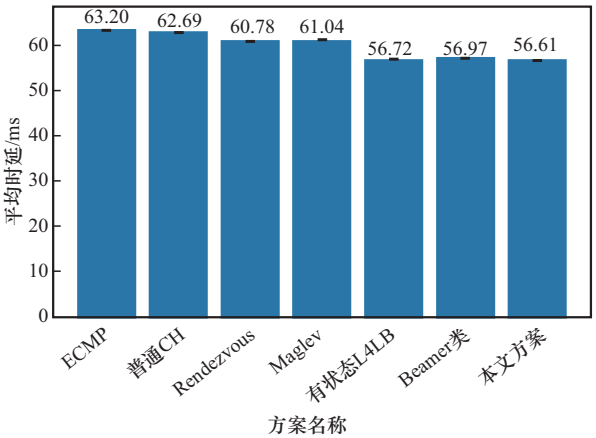

图7　平均时延及95%置信区间

在99分位时延方面，本文方案低于ECMP、普通CH、Rendezvous和Maglev基线，略高于有状态L4LB与Beamer类基线。综合平均时延和尾时延结果可见，资源—拓扑联合加权有助于改善普通散列基线的调度表现，

同时保留了进一步优化尾部性能的空间。本文方案的主要特点，是通过共享映射与版本管理，在连接保持、调度性能和星上状态开销之间取得平衡。

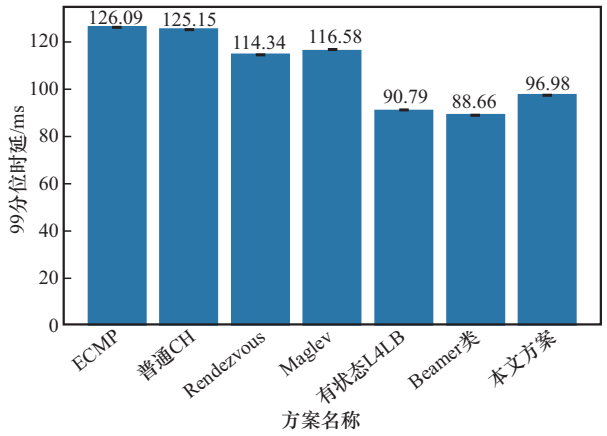

图 8　99 分位时延及 95% 置信区间

3.4　有效吞吐量与丢包率

图 9 给出了各方案的有效吞吐量及 95% 置信区间。本文方案为 4.49 Gbit/s，高于 ECMP 的 4.27 Gbit/s，并与有状态 L4LB 和 Beamer 类基线接近。图 10 展示了丢包率及 95% 置信区间，本文方案和 ECMP 的图示均值分别为 0.24% 和 0.89%。吞吐量与丢包结果共同表明，本文方案在该负载条件下具有较好的调度表现。

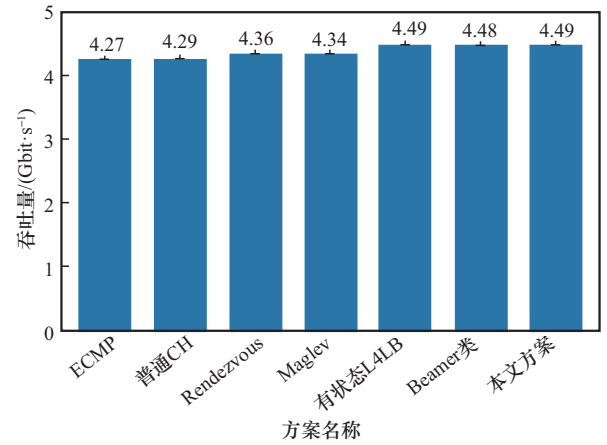

图 9　有效吞吐量及 95% 置信区间

从调度机制看，按 CPU 能力分配连接能够减少低算力节点过度接纳，距离权重则将星间转发代价纳入选择过程。当前吞吐量、丢包率与 CPU 负载结果整体上与这一设计目标一致。进一步结合 RS 过载持续时间、链路队列和 CPU 饱和分布，可分析计算侧与网络侧各自对性能的影响，并为引入实时拥塞反馈提供依据。

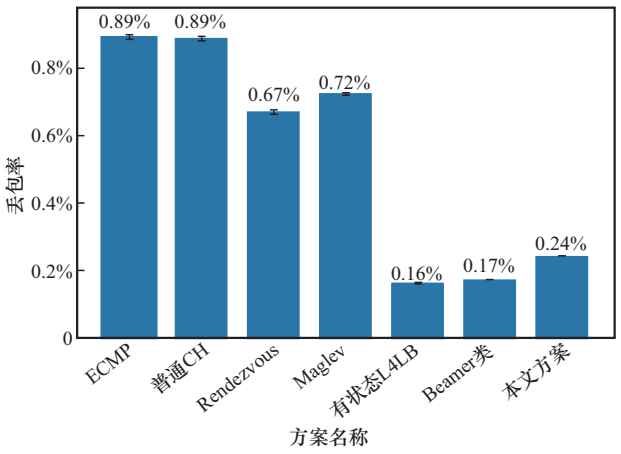

图 10　丢包率及 95% 置信区间

3.5　接入卫星切换下的连接保持与迁移开销

图 11 给出了接入卫星切换窗口内的长连接重置率。在当前模型的切换设置下，ECMP 基线最高，普通 CH、Rendezvous 和 Maglev 模型仍有非零重置，本文方案为 0。该结果与第 2.3 节描述的连接保持方式相对应：在版本可用、地面流上下文连续且后端正常运行时，入口切换后的报文沿用原版本查询历史环，可复现首包选择的 RS。不同基线的结果需结合各自入口配置与版本处理方式理解，图 11 用于观察模型切换过程中的连接保持表现。

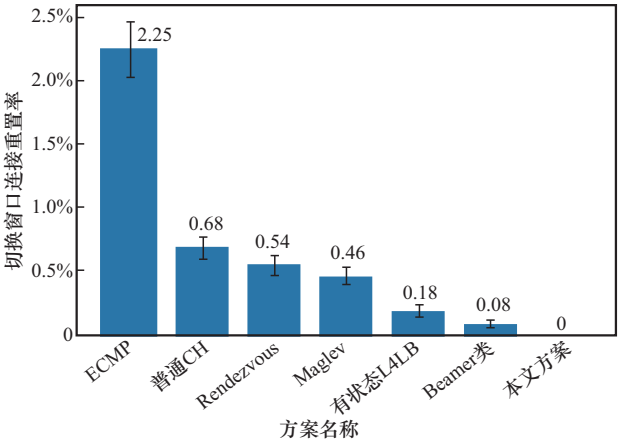

图 11　接入卫星切换窗口内长连接重置率

图 12 比较切换期间各方案需要传递的连接状态或共享映射信息。本文方案以哈希环为迁移单位，迁移量主要取决于环大小和历史保留深度；有状态 L4LB 则需要同步逐连接记录，其迁移量随活动连接数增长。两者的差异体现了共享映射在降低切换状态传输开销方面的优势。

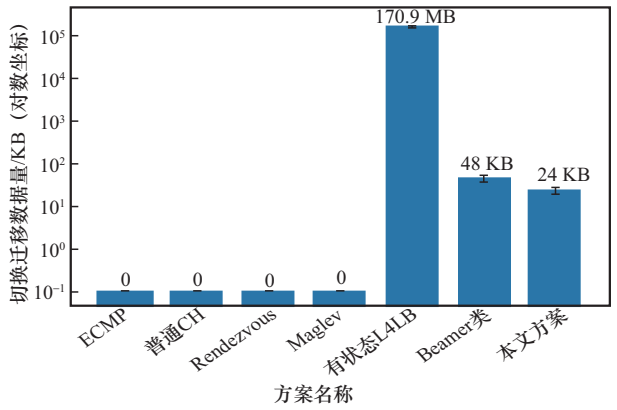

图 12　切换期间连接状态或哈希环迁移开销

综合连接重置率与迁移量可见，本文方案通过在接替入口复用原有映射，在模型级评估中兼顾了切换期间的连接保持和低迁移开销。后续可结合真实协议栈，进一步测量切换时的报文重传、乱序和应用停顿，将映射层结果与实际业务体验联系起来。

3.6　扩缩容扰动与双版本机制

图 13 给出了增加或移除一个 RS 时的存量连接重映射比例，用于比较是否保留旧环对连接映射的影响。未启用双版本机制时，本文方案在成员变化后产生约 13%~16% 的存量映射扰动；启用双版本机制后，新连接使用更新后的环，旧连接继续引用原环，图示存量连接扰动为 0。结果体现了“新连接适应更新、旧连接保持映射”的设计思路。对于第 2.4 节的计划缩容，待退出 RS 继续处理已有连接，使这种映射保持与后端有序退出相衔接。

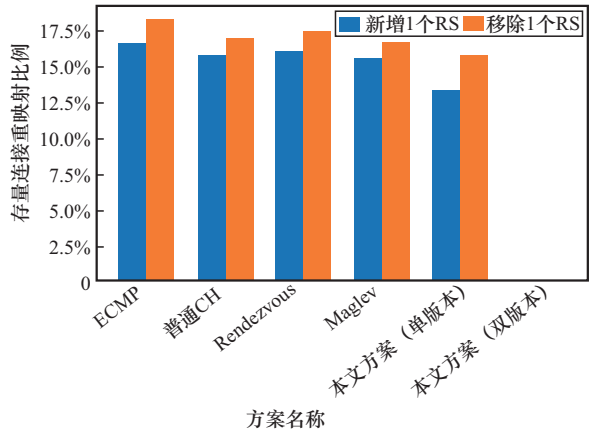

图 13　增加或移除一个 RS 时的存量连接重映射比例

图 13 评价的是存量连接的映射变化，资源调整速度则取决于连接自然周转：扩容后，新 RS 随新连接到达逐步承担业务；缩容时，原 RS 随存量连接结束完成排空。因此，连接扰动与资源调整速度构成两项互补的评价维度，后者可进一步结合新流到达过程和长连接持续时间分析。

4　讨论

4.1　可部署性与兼容性

方案采用端系统协议透明、地面网关增强和星上入口每流无状态的部署方式。客户端继续访问普通 VIP，RS 配置标准隧道终结、VIP 和返回路由，传输连接沿用原有 TCP 处理流程。地面网关在连接生命周期内维护版本缓存，并在切换或重启后恢复相关上下文。多网关部署可采用流亲和、缓存复制或稳定锚点，使同一连接在不同入口下保持一致的版本标记，支撑星地两侧协同运行。

多租户场景可按租户划分 service_id/VIP 命名空间，对版本目录和隧道元数据实施授权，并通过异常版本查询限速控制历史拉取及缓存压力。可信隧道、缓存复制和租户隔离共同构成部署侧保障，其处理与存储开销可结合具体平台进一步评估。

4.2　故障场景与可靠性

服务可靠性由接入连续性、版本可用性和后端恢复协同保障。入口故障时，已预置共享环的备份入口可按原版本恢复后端选择；历史环尚未到达时，通过补发、按需拉取或受控转交补齐映射。向多个候选接替卫星预分发环，可降低预测误差对版本可用性的影响。部署时需要结合星地链路条件，在预分发范围、复制开销和恢复速度之间选择合适配置，并联合观察路由恢复与报文交付表现。

后端故障后的会话恢复，可根据业务要求采用应用重试、后端状态复制或传输连接迁移，与 L4LB 的后端集合更新配合。哈希环负责维持和发布连接映射，会话恢复机制负责接续后端处理状态，二者共同支撑服务连续性。地面网关的版本缓存也应具备恢复能力，并在接入锚点或客户端地址变化时维持可识别的流上下文。后续可通过故障注入评估各环节的恢复时间及资源开销。

4.3　系统参数选择

虚拟节点数影响权重表达的平滑性，也决定环构建、查询和迁移成本。部署评估可从 1 024 个至 4 096 个虚拟节点的候选范围中展开，结合后端规模和流量分布选择合适粒度。历史深度可根据所支持连接的寿命与版本发布速率配置；历史环也可依据地面汇总的活动引用动态回收。权重更新越频繁，同时保留的历史版本可能越多。因此，虚拟节点数、历史深度和更新周期需要联合考虑，并通过适度平滑遥测控制版本增长。

CPU 与距离权重可围绕业务目标调整：增加β提高邻近 RS 的选择份额，增加α提高高算力节点的选择份额，具体效果结合时延、吞吐量和后端负载测量判断。在此基础上，可进一步引入队列长度、链路拥塞、内存、能耗和任务优先级等指标，形成更丰富的资源视图。扩展时仍通过新版本发布权重，并保持历史环固定。对于负载变化快于连接周转的业务，还可与应用层调度或状态迁移协同，提高资源调整速度。

5 相关工作

5.1 记录连接表的 L4LB

Linux Virtual Server 通过集群化 IP 负载均衡提供统一服务入口[3]。Ananta 结合可扩展软件数据平面与控制平面，支持云规模负载均衡[4]；Duet 利用网络交换设备和软件协同转发[19]；SilkRoad 面向可编程交换芯片实现低成本有状态 L4LB[20]；Concury 通过紧凑查找结构降低连接映射的存储与更新开销[21]。这些工作从软硬件协同和数据结构等角度提高逐连接状态维护的效率。本文关注接入卫星周期性迁移带来的持续同步需求，以服务级映射迁移替代与连接规模相关的状态传输，适应星上资源和切换条件。

5.2 无连接状态的 L4LB

普通一致性哈希、Rendezvous 和 Maglev 哈希为后端选择提供可复现的映射方法[6,12-13]。Maglev 系统将查找表与连接跟踪结合。Beamer 类则通过后端纠错分担连接一致性处理，使其负载均衡器保持逐连接无状态[17]。本文沿用确定性映射的基本思想，针对低轨入口持续移动的特点，将地面逐流版本标记、星上固定环快照、历史环预迁移和计划排空组织为统一机制。地面网关保存流上下文，星上入口保留活动版本，二者协同复现连接建立时的后端选择。

5.3 卫星网络流量负载均衡

卫星网络相关研究关注接入切换、低时延路由和计算卸载等问题[1-2,5,14]。Handley[22]分析低轨星间网络的时延与多路径特性，为路径选择提供依据；Celestial 面向包含卫星运动和地面节点的 LEO 边缘软件提供试验环境[7]。本文从服务入口出发，研究连接应由哪个 RS 处理，以及入口移动后如何延续该选择。服务编排负责实例放置，L4LB 负责连接到后端的映射，网络层负责当前入口到目标 RS 的可达路径，三者共同支撑星上资源池的动态服务能力。

6 结束语

本文面向低轨卫星由通信转发节点向在轨计算服务节点演进的场景，研究多后端资源池的星上四层负载均衡。采用 CPU 能力与 LB-RS 距离联合加权的一致性哈希，通过地面逐流版本标记、星上历史环预迁移和新旧版本并存，将频繁切换入口上的逐连接维护转化为服务级映射管理。IP-in-IP/DSR 默认转发与静态 NAT 兼容模式进一步连接后端选择和请求响应处理，形成星地协同的 L4LB 架构。

128 颗卫星场景的模型级仿真结果表明，相较于 ECMP，本文方案改善了平均时延、CPU 负载标准差和丢包表现；在所评估的切换窗口内未记录到长连接重置，新旧版本并存时的存量连接重映射比例为零。这些结果分别对应联合权重、历史环迁移和新旧流隔离的设计目标，为兼顾调度性能、连接保持与低星上状态开销提供了模型级支持。后续将结合真实协议栈和硬件平台，进一步验证长连接、多入口及故障场景下的运行表现，并研究融合链路拥塞、能耗和任务优先级的多维调度机制。

参考文献

[1] Xie R C, Tang Q Q, Wang Q N, et al. Satellite-terrestrial integrated edge computing networks: architecture, challenges, and open issues[J]. IEEE Network, 2020, 34(3): 224-231.

[2] Denby B, Lucia B. Orbital edge computing: nanosatellite constellations as a new class of computer system[C]//Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2020: 939-954.

[3] Zhang W. Linux virtual server for scalable network services[C]//Proceedings of the Ottawa Linux Symposium. Ottawa: Ottawa Linux Symposium, 2000.

[4] Patel P, Bansal D, Yuan L H, et al. Ananta: cloud scale load balancing[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 207-218.

[5] Hu J, Pan T, Chen Y J, et al. LISP-LEO: location/identity separation-based mobility management for LEO satellite networks[C]//Proceedings of the GLOBECOM 2022 - 2022 IEEE Global Communications Conference. Piscataway: IEEE Press, 2023: 1558-1563.

[6] Karger D, Lehman E, Leighton T, et al. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web[C]//Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing - STOC '97. New York: ACM, 1997: 654-663.

[7] Pfandzelter T, Bermbach D. Celestial: virtual software system testbeds for the LEO edge[C]//Proceedings of the 23rd ACM/IFIP International Middleware Conference. New York: ACM, 2022: 69-81.

[8] Eddy W. Transmission control protocol (TCP): RFC 9293[S].

IETF, 2022.
[9] Srisuresh P, Egevang K. Traditional IP network address translator (traditional NAT): RFC 3022[S]. IETF, 2001.
[10] Perkins C. IP encapsulation within IP: RFC 2003[S]. IETF, 1996.
[11] Hopps C. Analysis of an equal-cost multi-path algorithm: RFC 2992[S]. IETF, 2000.
[12] Thaler D G, Ravishankar C V. Using name-based mappings to increase hit rates[J]. IEEE/ACM Transactions on Networking, 1998, 6(1): 1-14.
[13] Eisenbud D E, Yi C, Contavalli C, et al. Maglev: a fast and reliable software network load balancer[C]//Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2016: 523-535.
[14] Chowdhury P K, Atiquzzaman M, Ivancic W. Handover schemes in satellite networks: state-of-the-art and future research directions[J]. IEEE Communications Surveys & Tutorials, 2006, 8(4): 2-14.
[15] Gross J, Ganga I, Sridhar T, et al. Geneve: generic network virtualization encapsulation: RFC 8926[S]. IETF, 2020.
[16] Rijsinghani A. Computation of the internet checksum via incremental update: RFC 1624[S]. IETF, 1994.
[17] Olteanu V, Agache A, Voinescu A, et al. Stateless datacenter load-balancing with beamer[C]//Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation. New York: ACM, 2018: 125-139.
[18] Mogul J, Deering S. Path MTU discovery: RFC 1191[S]. IETF, 1990.
[19] Gandhi R, Liu H H, Hu Y C, et al. Duet: cloud scale load balancing with hardware and software[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(4): 27-38.
[20] Miao R, Zeng H Y, Kim C, et al. SilkRoad: making stateful layer-4 load balancing fast and cheap using switching ASICs[C]//Proceedings of the Conference of the ACM Special Interest Group on Data Communication. New York: ACM, 2017.
[21] Shi S Q, Yu Y, Xie M H, et al. Concury: a fast and light-weight software cloud load balancer[C]//Proceedings of the 11th ACM Symposium on Cloud Computing. New York: ACM, 2020: 179-192.
[22] Handley M. Delay is not an option: low latency routing in space [C]//Proceedings of the 17th ACM Workshop on Hot Topics in Networks. New York: ACM, 2018: 85-91.

[作者简介]



刘映江（2000– ），男，北京邮电大学硕士生，主要研究方向为低轨卫星网络、网络中间设备、网络测量等。



潘恬（1987– ），男，博士，北京邮电大学教授、博士生导师，主要研究方向为云数据中心网络、智算中心网络、低轨卫星网络、高速可编程网络设备等。



李浩南（2000– ），男，北京邮电大学博士生，主要研究方向为云数据中心网络、低轨卫星网络、高速可编程网络设备等。



阮国皓（1999– ），男，北京邮电大学博士生，主要研究方向为低轨卫星网络、网络路由协议等。



郑长刚（1998– ），男，博士，北京空间飞行器总体设计部工程师，主要研究方向为网内计算、云数据中心网络、空间算力网络等。



沈寓实（1978– ），男，博士，飞诺门阵（北京）科技有限公司创始人、首席科学家，主要研究方向为云计算、人工智能、网络安全等。



黄韬（1980– ），男，博士，北京邮电大学教授、博士生导师，主要研究方向为未来网络体系架构、软件定义网络、确定性网络等。