

研究论文

卫星互联网中路径重构感知的 TCP 传送优化方法

李宗伦¹, 赖泽祺^{1*}, 金熙明², 吴茜¹, 李贺武¹

摘要 以星链(Starlink)、一网(OneWeb)、中国星网、千帆为代表的低轨卫星星座快速发展,承载着越来越多的互联网流量。然而,低轨卫星高速移动导致链路容量、时延和丢包率剧烈变化,给传输控制协议(transmission control protocol, TCP)端到端传送带来重大挑战。现有卫星网络中的端到端传送优化方案,或者需要修改和升级端系统内核,难以大规模部署;或者面向传统的地球静止轨道卫星场景,难以应用于高度动态时变的低轨卫星环境。为克服现有方法的局限性,围绕低轨卫星网络环境下的 TCP 性能增强方法开展研究,提出一种低轨卫星重构可感知、端系统透明的性能增强代理 StarPEP,通过在代理处调节 TCP 接收窗口以主动控制发送端速率,结合低轨卫星的周期性重构特征、RTT(round-trip time)与瓶颈链路带宽动态变化特征,在不修改端系统的前提下有效提升 TCP 传送性能。本研究依托 Linux 系统实现 StarPEP 原型系统。真实低轨卫星网络数据驱动的仿真试验表明:从吞吐量与时延的二维关系看,StarPEP 可以在不修改端系统的前提下更接近帕累托改进方向,相比使用 Hybla 作为拥塞控制算法的 PEP,将端到端时延降低了 94.8%~95.9%,吞吐量提升了 49.3%~214.0%;相比使用 LeoCC 作为拥塞控制算法的 PEP,在牺牲少量吞吐的同时将端到端时延降低了 89.1%~92.6%。

关键词 卫星互联网; TCP 性能优化; 性能增强代理

得益于星链(Starlink)、一网(OneWeb)、中国星网、千帆等星座的快速部署与发展,低轨道(low earth orbit, LEO)卫星网络发展迅速,为全国各地提供无处不在的宽带覆盖和低时延接入,承载着越来越多的互联网流量。以星链为例,作为当前规模最大的 LEO 卫星网络,截至 2026 年 2 月,为分布在超 100 个国家的约 1000 万用户提供服务^[1],范围覆盖七大洲。不同

于传统地面网络,LEO 卫星网络中部分网络基础设施(即 LEO 卫星)相对地球表面以约 7.5 km/s 的速度快速移动。因此,LEO 卫星网络路径上的部分节点和链路随时间不断发生变化。这种独特的 LEO 卫星动态性不仅导致链路容量快速变化,同时也带来频繁的时延变化(由路径变动引起)及较高的丢包率(由卫星与地面节点之间的连接切换和卫星链路信道质量引起),因此迫切需要一种新型传送层性能优化机制来增强 LEO 卫星网络传送性能。

典型的传送层传输控制协议(transmission control

1. 清华大学网络科学与网络空间研究院,北京 100084

2. 清华大学计算机科学与技术系,北京 100084

收稿日期:2026-04-01;修回日期:2026-06-17

基金项目:国家自然科学基金项目(62372259,62132004);中国科协青年人才托举工程项目(YESS20230686)

作者简介:李宗伦,硕士研究生,研究方向为天地一体化互联网拥塞控制,电子信箱:lzl24@mails.tsinghua.edu.cn;赖泽祺(通信作者),副教授,研究方向为天地一体化互联网、网络空间安全,电子信箱:zeqilai@mails.tsinghua.edu.cn

引用格式:李宗伦,赖泽祺,金熙明,等.卫星互联网中路径重构感知的 TCP 传送优化方法[J].科技导报,2026,44(16):137-146;

doi:10.3981/j.issn.1000-7857.2026.04.00001

protocol, TCP)性能优化机制可以分为端侧的拥塞控制算法(congestion control algorithm, CCA)及网络侧的性能增强代理(performance enhancing proxy, PEP)2类。端侧 CCA 以某种网络假设为基础,其性能高度依赖这些假设在真实网络环境中的准确性,而 LEO 卫星的动态性对网络产生的影响打破了许多 CCA 的底层假设,因此,有研究(如 LeoCC^[2]、SaTCP^[3]、SatPipe^[4])将 LEO 卫星的动态特性纳入算法设计,提升了 CCA 在 LEO 卫星环境下的性能表现。然而由于端侧的 TCP 算法需要对操作系统内核做修改,用户若想体验 CCA 优化带来的性能提升,需要具备较强的专业知识,因此,端侧 CCA 很难真正大规模部署与应用。与之不同的是,网络侧的 PEP 仅需网络运营商修改网络侧设备,对端侧用户透明无感知,可以通过在传送路径中间插入代理节点拆分 TCP 连接,使用缓存、本地重传、内容压缩等技术优化网络性能。先前工作(如 SaTPEP^[5]、PEPsal^[6]、QPEP^[7])聚焦地球静止轨道卫星(geostationary earth orbit, GEO)场景,主要用于解决 GEO 卫星场景下吞吐受限的问题,忽略了 LEO 卫星场景中固有的低时延潜力,并不能对 LEO 卫星网络性能带来较大提升。

因此,本文提出一种适配 LEO 卫星网络环境的 PEP 方案 StarPEP。通过在代理节点动态调节 TCP 接收窗口,主动控制发送端的注入速率,从而降低代理处的队列积压并维持较高链路利用率。其核心设计将稳态下的速率与队列感知窗口控制,与路径重构期间的感知与保护机制结合,使系统既能适应带宽波动,也能在周期性重构时快速恢复到新的稳定工作点。该方案无需对端侧设备做任何修改,便于网络运营商大规模部署与落地应用。

1 研究背景与相关工作

1.1 低轨卫星网络特征及其对传送层的影响

图1展示了当前主流 LEO 卫星网络普遍采用的架构。用户终端(如笔记本电脑)通过1台专用的卫星终端设备与 LEO 卫星建立连接。数据经由 LEO 卫星被转发到地面站,经过入网点后最终接入地面互联网。如果用户距离地面站较近,通信直接通过“弯管”方式完成,即所连接的 LEO 卫星将数据透明转发

给地面站,不经由星间链路。如果用户距离周围任一地面站较远,则数据包通过星间链路先在不同卫星间传递,最终路由到地面网络^[8]。

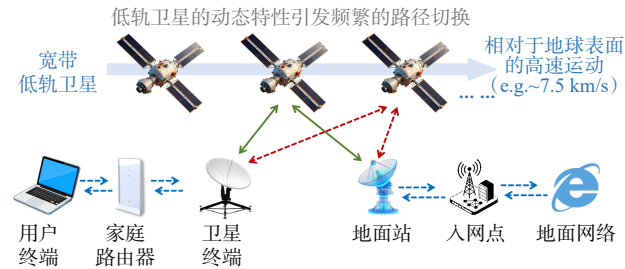


图1 LEO 卫星网络的基本组网架构

与传统地球静止轨道卫星网络相比,LEO 卫星轨道高度更低,传播时延显著下降,但与此同时,单颗卫星对固定地面节点的可见时间较短,导致网络拓扑和可用路径持续变化。为了维持端到端连接,卫星运营商需要调整卫星与地面节点之间的连接关系,触发链路路径重构和波束重分配的过程。Starlink 中该种路径重构行为周期约为 15 s,OneWeb 中观测到约 11 s 的周期性变化现象。除拓扑变化外,经测量发现^[9-12],LEO 接入链路还会受到无线调度、大气衰减、遮挡等因素的影响,使得瓶颈链路带宽与端到端最小往返时延(round-trip time, RTT)在数秒内剧烈变化,且 LEO 卫星链路易受到丢包影响,即便在链路流量未饱和时,仍有 0.5%~6.0% 的随机丢包^[2]。

这类由 LEO 卫星动态性引起的网络变化,对经典互联网拥塞控制机制构成了根本性挑战。传统 TCP 拥塞控制通常隐含 2 个基本假设:其一,瓶颈带宽和带宽时延积(bandwidth delay product, BDP)在短时间内相对稳定;其二,丢包和时延上升主要由网络拥塞引起。然而,在 LEO 卫星网络中,这 2 个假设均难以成立。一方面,卫星重构、无线调度及链路状况波动会持续改变路径上的可用带宽与最小 RTT,使发送端难以围绕稳定目标进行收敛;另一方面,重构期间的瞬时中断、非拥塞随机丢包以及路径切换造成的时延抖动,容易被传统算法误判为拥塞信号,进而触发不必要的降窗降速行为。因此,端到端连接往往既无法持续维持较高吞吐,也难以保持较低排队时延。

1.2 端侧拥塞控制算法(CCA)

针对 LEO 卫星网络的上述动态特性,近年来已有一些研究尝试从端侧拥塞控制算法入手进行优

化。SaTCP^[3]在 Cubic 算法^[13]的基础上进行改动,在 LEO 卫星与地面节点重新建立连接时,冻结拥塞窗口以规避因非拥塞丢包导致的错误降窗行为;SatPipe^[4]在瓶颈带宽和往返传播时间(bottleneck bandwidth and round-trip propagation time, BBR)算法^[14]的基础上进行改动,将 ProbeRTT 这一减少发包的固有阶段移动至周期性网络重构时间之前,以此来尽可能降低网络重构对算法发送行为带来的负面影响(如时延增加);LeoCC^[2]以网络重构作为算法不同阶段切换的标志性事件,旨在感知到重构事件后快速探测新链路的瓶颈带宽与 RTT,其认为链路不存在最小 RTT 定值,将最小 RTT 的固有波动范围融入算法模型以调整发送行为。这些方法说明:若能够将 LEO 网络特有的动态规律纳入算法设计,端侧拥塞控制的性能确实可以得到提升。

然而,端侧方法在实际部署中面临明显障碍。首先,绝大多数 TCP 拥塞控制算法运行于操作系统内核协议栈内部,其部署往往需要修改内核参数、加载额外模块,甚至重新编译和替换协议栈实现。对于普通终端用户而言,这类操作门槛高、可移植性差,也难以在海量异构终端上统一推进。其次,从互联网长期演进的历史经验来看,尽管研究界已经提出了大量面向不同网络场景的 TCP 变体,但真正进入主流操作系统并成为默认实现的算法非常有限,以 RFC 9438^[15]为例,其明确指出 Cubic 已成为 Linux、Windows 及 Apple 系统中的主流默认 TCP 拥塞控制算法。某一新算法在特定环境下即使具有明显性能优势,端侧协议栈的大规模替换与推广仍然是一个缓慢而保守的过程。

1.3 网络侧 TCP 性能增强代理(TCP-PEP)

与端侧方法相比,网络侧性能增强代理具有更强的现实部署可行性。PEP 部署于运营商可控的网络设备之上,无需修改终端操作系统和应用程序,对用户透明,便于集中管理、统一升级和逐步落地。RFC 3135^[16]将这类机制系统归类为用于缓解高时延、高误码和链路不稳定等问题的代理增强方法。

已有工作中,SaTPEP^[5]通过测量接收数据的速率得到链路利用率,将未充分利用的带宽与 RTT 乘积视为可用窗口,该窗口作为增量分配给各个 TCP 连接的接收窗口,同时基于否定确认(negative acknowledgment, NACK)进行丢包恢复;PEPsal^[6]采用连接分段

的方式,将原始端到端 TCP 会话分解为“用户到 PEP”和“PEP 到远端服务器”2 条独立连接,从而将 GEO 卫星高时延对端侧拥塞控制的不利影响限制在部分链路上,进而提升传送性能。

然而,以 PEPsal 为代表的传统开源 PEP 最初面向 GEO 场景设计,其核心收益主要来自连接拆分以及在长时延链路上采用更激进的传送策略以提升吞吐,较少关注时延优化。对于 LEO 网络而言,由于其物理特性本身决定了较低的传播时延,PEP 设计不能仅追求吞吐提升,还需同时兼顾低时延目标才能充分发挥 LEO 网络的优势。

实验结果表明,即使将 PEPsal 出口段连接中使用的拥塞控制算法替换为更适应 LEO 动态特性的算法(如 LeoCC),仍然无法同时实现吞吐与时延的良好权衡。图 2 给出了在不同 Starlink 城市对网络流量轨迹(trace)驱动仿真的结果。从图中可以看到,即使 PEPsal 与 LeoCC 的组合相比于原始 PEPsal 带来了一定的时延优化,然而相比环境基准时延,仍带来非常严重的端到端时延膨胀。在 2 组不同城市对(每组包含 30 个 Trace)中,单向时延分别被扩大了 140.89 倍和 102.79 倍。这说明,单纯将 PEP 与为 LEO 卫星网络设计的 CCA 直接组合并不能自然得到适合 LEO 网络的最佳代理方案。

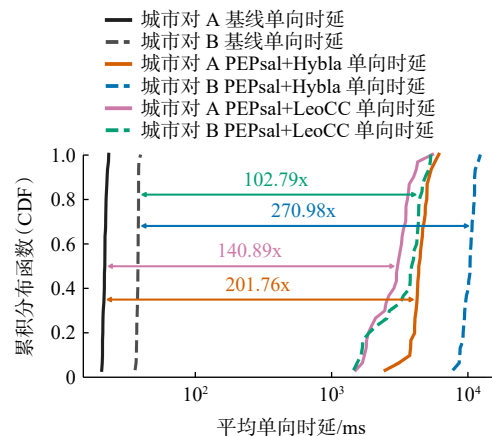


图 2 PEPsal 在不同 Trace 下的时延膨胀现象

为进一步分析图 2 中时延膨胀的成因,图 3 给出了 PEPsal+Hybla^[17]和 PEPsal+LeoCC 在单条 Trace 上的细粒度分析,其中包括吞吐量和端到端单向时延随时间的变化情况。可以看到,LeoCC 替换 Hybla 后,PEP 对链路带宽变化的跟踪能力有所增强,但端到端单向

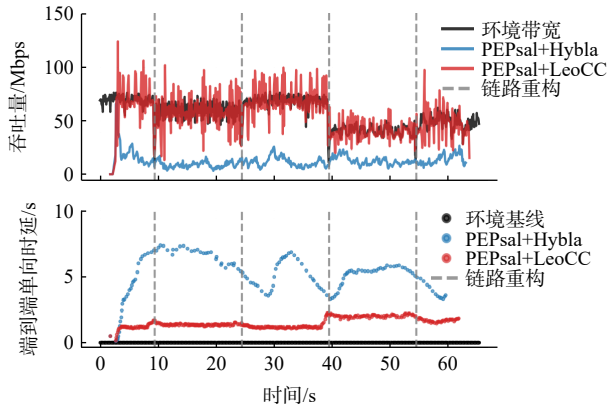


图3 PEPsal 架构下不同 CCA 的细粒度性能表现

时延仍长期显著高于环境基线。特别是在链路重构后,瓶颈链路状态发生改变,代理处可能会持续积累额外排队,导致时延难以及时回落。该结果表明,单纯优化 PEP 下游连接的拥塞控制行为无法从根本上抑制代理内部排队。其根本原因在于,现有 PEP 设计缺乏对原始发送端注入速率的直接约束能力,导致对于入口速率大于出口速率的代理而言,排队数据包长期累积,最终造成严重的时延膨胀。

2 StarPEP 设计

针对 LEO 卫星网络中带宽快速波动、路径周期性重构以及现有 PEP 代理处易发生队列积压的问题,设计了一套面向代理节点的传送优化机制。核心思想是:将发送端注入速率的调节从端到端拥塞控制逻辑中部分剥离出来,由 PEP 在路径中主动调节通告的接收窗口,从而在不改变端系统协议栈行为的前提下,直接约束进入卫星瓶颈前的在途数据量;同时考虑代理处的队列反馈和路径重构事件这 2 个关键因素,使其既能维持较高链路利用率,又能有效控制端到端时延。总体而言,该设计由 2 个核心组成部分构成。第 1 部分是速率与队列感知的稳态接收窗口调控机制,通过持续调整合适的接收窗口,使注入速率与卫星瓶颈链路带宽相匹配;第 2 部分是路径重构感知机制,其目标是在卫星链路发生重构时及时做出调整,通过窗口保护、重构探测与渐进恢复这 3 个过程,使 PEP 系统能够跨越链路切换时刻,快速回到新的稳定工作点,这 2 个部分共同构成 StarPEP。图 4 展示了分布式 StarPEP 架构图,需要注意的是这 2 部分仅

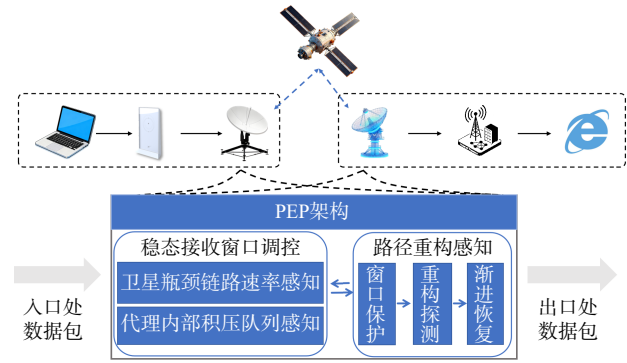


图4 分布式 StarPEP 架构

作用于出口侧是 LEO 卫星链路的 PEP,另一代理仅包含基础的流量转发功能。

2.1 速率与队列感知的稳态接收窗口控制机制

PEP 的入口处注入速率需要与出口处的卫星瓶颈链路带宽相匹配,因此,StarPEP 首先根据当前链路状态确定一个基础接收窗口,用来给发送端提供与卫星瓶颈链路传送能力大致匹配的注入边界。该基础接收窗口由下游卫星瓶颈链路速率和上游链路 RTT 共同决定

$$W_{\text{base}}(t) = R_{\text{down}}(t) \times RTT_{\text{up}} \quad (1)$$

式中, $R_{\text{down}}(t)$ 为卫星链路的实际传送速率, RTT_{up} 为发送端到 PEP 节点的上游地面段链路 RTT。

由于 LEO 链路的带宽和时延具有明显波动性,速率估计不可避免会存在误差和滞后。因此,即使基础接收窗口在平均意义上与链路能力大致匹配,仍可能在短时带宽波动时造成积压。为此,StarPEP 进一步利用代理节点的出口和入口实时队列状态,对基础接收窗口进行在线修正:当代理队列持续高于目标范围时,说明当前注入速率已经超过瓶颈链路的瞬时传送能力,此时 StarPEP 会主动收缩通告给发送端的接收窗口以抑制继续堆积;当队列长期维持在较低水平时,则说明当前窗口偏于保守,可以适度放大接收窗口以提升利用率

$$\hat{Q}(t) = \alpha Q(t) + (1 - \alpha) \hat{Q}(t-1), \quad 0 < \alpha < 1 \quad (2)$$

$$W(t) = \begin{cases} \beta W_{\text{base}}(t), & \hat{Q}(t) > Q_{\text{high}}(t) \\ W_{\text{base}}(t), & Q_{\text{low}}(t) \leq \hat{Q}(t) \leq Q_{\text{high}}(t) \\ W_{\text{base}}(t) + \Delta(t), & \hat{Q}(t) < Q_{\text{low}}(t) \end{cases} \quad (3)$$

式中, $Q(t)$ 为实时队列长度, $\hat{Q}(t)$ 为平滑后的队列估计, $Q_{\text{high}}(t)$ 和 $Q_{\text{low}}(t)$ 分别为代理处设置的排队时延上下界, $\Delta(t)$ 为队列过低时的加性增量, α 和 β 为系数。

值得注意的是,式(2)可以展开为

$$\hat{Q}(t) = \alpha Q(t) + \alpha(1-\alpha)Q(t-1) + \alpha(1-\alpha)^2 Q(t-2) + \dots \quad (4)$$

因此,当前队列估计值 $\hat{Q}(t)$ 实际上综合了当前时刻及多个历史时刻的队列状态,并对不同历史观测值赋予指数衰减权重。较新的队列观测值对估计结果影响更大,较早的观测值影响逐渐减弱,从而在抑制瞬时波动和跟踪队列变化趋势之间取得平衡。

通过先利用下游链路速率和上游链路 RTT 确定基础接收窗口,再利用代理队列反馈对其进行在线修正,StarPEP 把进入代理的稳态数据量控制在卫星瓶颈链路可持续承载的范围附近,在 LEO 链路中同时兼顾链路高利用率与低时延的目标。

2.2 路径重构感知机制

尽管速率与队列感知的接收窗口控制机制能够在链路状态相对连续变化时维持较好的稳态性能,但 LEO 卫星网络中的路径重构具有明显的突变特征。卫星切换可能使瓶颈带宽、传播时延等链路状态在短时间内发生阶跃变化,导致重构前形成的带宽估计、接收窗口和代理队列状态在重构后失效。因此,StarPEP 在稳态窗口控制之外进一步引入路径重构感知机制,将链路重构视为系统状态切换事件,并在重构前后主动调整接收窗口。如图 5 所示,该机制分为具体 3 个模块:窗口保护、重构探测与渐进恢复。

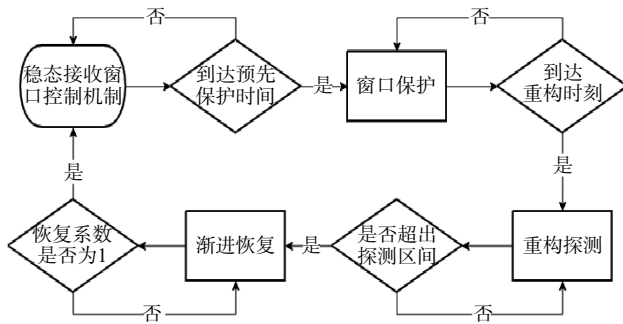


图 5 路径重构感知机制流程图

1) 窗口保护。设路径重构事件序列为

$$T_k = T_0 + kP, k = 0, 1, 2, \dots \quad (5)$$

式中, T_0 为连接建立后首次感知到的路径重构时刻, P 为重构周期,实际运行时均由卫星运营商提供。

对于当前时刻 t ,距离下一次路径重构的剩余时间为

$$\tau_k(t) = T_k - t \quad (6)$$

由于 LEO 卫星网络中的路径重构具有可预测性,StarPEP 不等待队列积压或带宽下降被动触发保护,而是在预计重构时刻到来前提前进入窗口保护阶段。设路径重构前的保护时间窗口为 Δ_{pre} ,则窗口保护阶段的触发条件为

$$0 \leq \tau_k(t) \leq \Delta_{pre} \quad (7)$$

当式(7)成立时,说明当前时刻已经进入路径重构前的保护区间。若此时继续按照稳态窗口向发送端开放较大的注入边界,旧路径上的积压将延续到重构时刻,并在新路径上形成额外排队。因此,StarPEP 将接收窗口限制在保守上限以内,以抑制路径重构前新的突发注入。设稳态速率与队列感知机制计算得到的接收窗口为 $W_s(t)$,重构阶段保守窗口为 W_r ,则保护阶段的窗口控制可表示为

$$W(t) = \min(W_s(t), W_r), 0 \leq \tau_k(t) \leq \Delta_{pre} \quad (8)$$

2) 重构探测。在路径重构事件发生后,旧路径上的带宽、时延和队列状态不再能够准确反映新路径的传输能力。若控制器在 T_k 时刻后立即恢复稳态接收窗口,则可能使发送端在新路径尚未稳定时注入过量数据。为此,StarPEP 在路径重构发生后的短时间内进入重构探测阶段,使系统以较小的窗口观察新路径的初始状态。设重构探测持续时间为 Δ_{probe} ,在该阶段,StarPEP 将通告接收窗口固定为探测窗口 W_r ,能够限制发送端在路径切换刚完成时继续快速注入数据;同时该机制并非完全关闭传输,而是保留少量数据流动,使控制器能够继续获得新路径上的带宽、RTT 和队列反馈。

3) 渐进恢复。重构探测阶段结束后,控制器已经获得了新路径上的初始带宽、RTT 和队列状态样本。然而,由于新路径状态仍处于收敛初期,若立即将接收窗口恢复到稳态窗口,可能导致发送端注入速率发生突增,从而在新路径上形成新的队列积压。为避免探测阶段与稳态控制阶段之间的窗口跳变,StarPEP 在重构探测后进入渐进恢复阶段。

设渐进恢复阶段持续时间为 $\Delta_{recover}$,在该阶段,StarPEP 首先根据当前带宽估计、上游链路 RTT 以及代理队列反馈计算稳态窗口 $W_s(t)$ 。控制器并不直接采用该窗口,而是利用恢复系数 $\alpha(t)$ 将接收窗口从探测窗口平滑过渡到稳态窗口

$$\alpha(t) = \frac{t - (T_k + \Delta_{\text{probe}})}{\Delta_{\text{recover}}}, \quad (9)$$

$$T_k + \Delta_{\text{probe}} < t \leq T_k + \Delta_{\text{probe}} + \Delta_{\text{recover}}$$

从而有接收窗口

$$W(t) = W_r + \alpha(t)(W_s(t) - W_r) \quad (10)$$

渐进恢复阶段的作用是将重构后的窗口恢复过程由瞬时跳变转化为平滑过渡。由于 $W_s(t)$ 已经包含速率估计和队列反馈信息, 因此恢复目标会随新路径观测结果动态变化。若新路径带宽较高且队列状态保持稳定, $W_s(t)$ 会引导接收窗口在恢复区间内逐步增大, 从而提升链路利用率; 若新路径带宽较低或代理队列已经升高, 稳态队列反馈会降低 $W_s(t)$, 使恢复过程自然受到抑制。

当渐进恢复阶段结束后, StarPEP 重新切换到速率与队列感知的稳态窗口控制机制。此时, 接收窗口完全由当前链路速率估计和代理队列反馈共同决定, 从而使系统在路径重构后重新进入兼顾高利用率和低排队时延的稳定状态。

3 实验与分析

3.1 实验环境设置

本研究基于 Linux 系统搭建仿真环境以模拟 LEO 卫星网络下 StarPEP 的代理转发与拥塞控制行为。实验通过 ip netns 创建 3 个网络命名空间分别作为发送端、接收端和其中一个分布式代理, 物理主机作为另一代理节点。代理节点中, 用户态控制器周期性采集 socket 状态、链路带宽估计和本地队列信息, 动态计算接收窗口; 内核态 eBPF 程序挂载于转发路径, 对确认(acknowledgement, ACK)报文中的通告窗口字段进行实时修改, 以约束发送端的注入速率。与此同时, 实验还在代理节点部署改造后的拥塞控制模块, 用于导出与当前瓶颈链路状态相匹配的速率估计。

为模拟 LEO 卫星网络的时变特性, 实验对 2 个代理节点之间模拟的卫星链路进行网络 Trace 回放。具体而言, 使用 tc-htb 设置带宽, 使用 tc-netem 设置延迟与随机丢包, 按时间序列加载真实测量轨迹中的链路带宽、延迟和丢包变化, 使链路状态能够随时间动态更新。实验所用轨迹更新粒度为 100 ms, 其来源于真实 Starlink 网络测量数据^[2], 该数据集中详细记

录了时变的带宽和时延状况, 能够较为准确地反映 LEO 网络的时变特征。对于丢包率而言, Starlink 实网环境较不稳定, 难以给出某一精确数值, 结合近期工作中给出的丢包范围(如 0.5%~6.0%^[2]以及约 0.64%^[18]), 模拟环境中设置丢包率为 0.5% 以尽可能贴近现实情况。针对 LEO 网络周期性路径重构的特点, 实验结合轨迹数据确定重构发生时间, 并在相应窗口内模拟短时链路中断, 以复现卫星切换带来的瞬态不可用现象。TCP 流量使用 iperf3 工具生成, 使用 -t 参数指定传送持续时间。

具体实验中, 本文设置路径重构前的保护时间窗口 Δ_{pre} 为 100 ms; 参考 BBR 中 ProbeRTT 阶段的参数设置, 将重构探测阶段 Δ_{probe} 设置为 200 ms, 并将探测窗口 W_r 设置为 4 MSS; 设置渐进恢复阶段 Δ_{recover} 为 300 ms。对于稳态接收窗口控制机制中的 α 和 β , 会在第 4.2 节中给出参数敏感性分析与推荐取值范围, $\Delta(t)$ 取 1 MSS。

3.2 实验结果分析

为对比 StarPEP 与其他传送优化机制的性能表现, 本文选择了以下 5 种机制进行对比。

(1) PEPsal+TCP Hybla: 采用经典开源 PEP 实现 PEPsal, 并在其下游连接中使用面向高时延卫星链路设计的 TCP Hybla。(2) PEPsal+SaTCP: SaTCP 以 Cubic 算法为基础, 在发生路径重构时冻结拥塞窗口, 从而排除此时非拥塞丢包造成的错误降速影响。(3) PEPsal+SatPipe: SatPipe 将 BBR 算法中的 PROBE_RTT 阶段用于路径重构发生前后, 以 4 个 MSS 的窗口大小限制数据包的发送量, 从而达到降速目的。(4) PEPsal+LeoCC: LeoCC 在路径重构后清空旧链路状态信息并快速上探新的瓶颈链路带宽, 在稳态时对带宽样本采用最大滤波与卡尔曼滤波, 根据 RTT 样本与最小 RTT 的差值来选择实际发送速率。(5) StarPEP: 本文提出的面向 LEO 卫星网络的性能增强代理, 通过调节发送端注入与代理下游发送过程, 在兼顾链路利用率的同时抑制代理处的持续排队。

图 6 比较了不同方案在多个城市对、多组 Trace 下的平均吞吐量与平均单向时延分布。横轴表示平均吞吐量, 纵轴表示平均单向时延, 因此越靠近右下角, 说明方案越能同时实现高吞吐和低时延。箱形区域刻画了结果的主要分布范围: 左右边缘分别对应吞

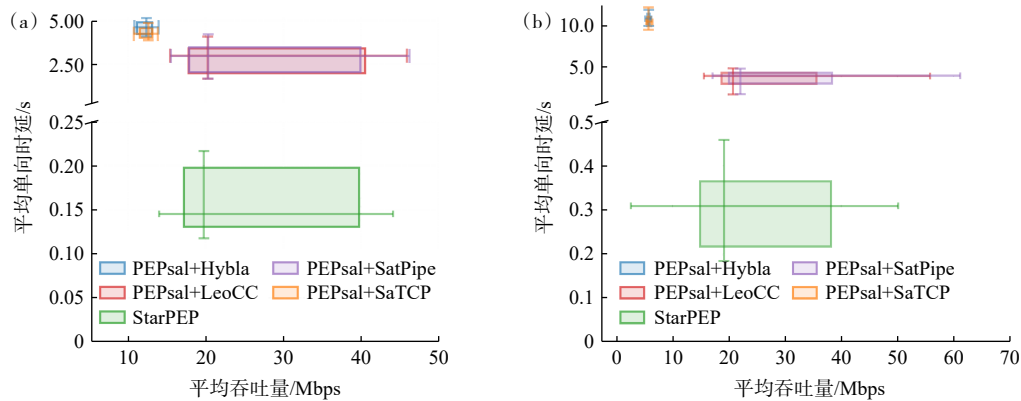


图6 StarPEP在不同城市对多组Trace上的实验性能分析

吐量的 25% 和 75% 分位点,上下边缘分别对应时延的 75% 和 25% 分位点;误差线则给出 10%~90% 分位点范围,水平与垂直误差线的交点表示 2 项指标的中位数。

从结果可以看出,PEPsal+Hybla 和 PEPsal+SaTCP 整体分布在左上区域,说明它们在吞吐量和时延上都表现不佳。主要原因是,这类基于丢包反馈的拥塞控制算法容易将 LEO 网络中的非拥塞丢包误判为拥塞信号,从而频繁降低发送窗口,导致 PEP 节点向卫星链路的发送速率偏低。较低的发送速率不仅限制了吞吐量,也使代理内部已积压的数据难以及时排空,最终导致较高的时延。PEPsal+LeoCC 和 PEPsal+SatPipe 虽然显著提高了吞吐量,但由于缺乏对代理缓冲区占用的有效控制,其平均单向时延仍处于秒级。相比之下,StarPEP 的结果整体更接近右下区域,说明它在保持较高吞吐量的同时显著降低了时延。以中位数为例,StarPEP 相比 PEPsal+Hybla 将平均单向时延分别降低了 94.8% 和 95.9%,相比 PEPsal+LeoCC 分别降低了 92.6% 和 89.1%;吞吐方面相比 PEPsal+Hybla 分别提升了 49.3% 和 214.0%,与 PEPsal+LeoCC 接近,体现出更好的吞吐量与时延权衡。

为了细粒度分析 StarPEP 机制在吞吐量和单向时延 2 个维度的优化效果,图 7 展示了 StarPEP 和 PEPsal+LeoCC 在同一 Trace 下随时间变化的吞吐量和单向时延细节,该 Trace 在约 9 s 和 24 s 处分别发生一次带宽突降与带宽突增的链路路径重构。可以发现,StarPEP 在吞吐量相似的情况下,大幅降低了端到端的单向时延,在路径重构事件发生后,时延在短暂上升后恢复至稳态时的较低水平。

为分析 StarPEP 中不同参数对性能的影响,图 8

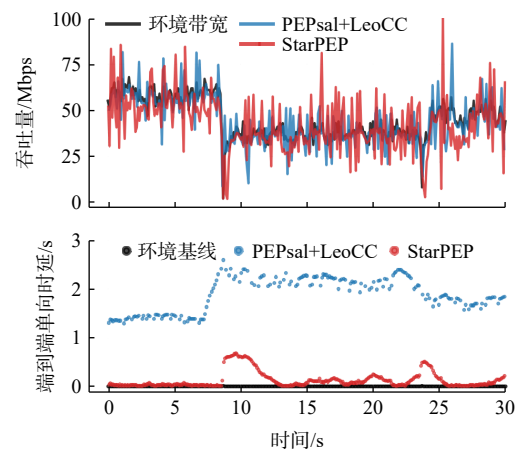


图7 StarPEP 吞吐与时延细节分析

展示了 2 个重要参数,即 $Q_{low}(t)$ 和 $Q_{high}(t)$ 的不同组合对吞吐量与单向时延的影响(单位: ms)。在 2 组 Trace 上可以看到,较小的 $Q_{high}(t)$ 能够更早抑制代理处的队列增长,因此平均单向时延整体更低;而较大的 $Q_{high}(t)$ 虽然在部分情况下带来略高的平均吞吐量,但同时也引入更高的时延。相比之下, $Q_{low}(t)$ 主要决定窗口扩张的积极程度:该值较小使控制器更快进入保持接收窗口的状态;该值较大则使控制器对接收窗口的扩张更为积极,有助于充分利用链路带宽资源。综合 2 组 Trace 的结果,将目标队列区间控制在适中

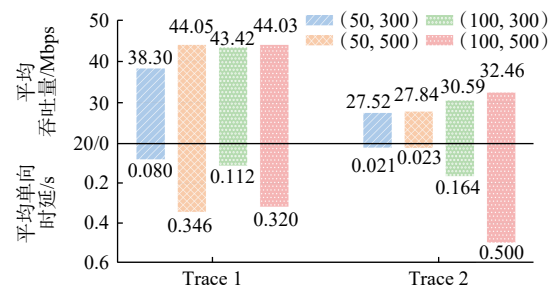


图8 StarPEP 不同参数对性能的影响

的范围内,能够在避免过深排队的同时维持较高的链路利用率。

为进一步说明参数设置依据,并验证 StarPEP 对控制参数的敏感性,在一组典型 Trace 上对 2 个控制参数进行敏感性分析。 α 用于控制 PEP 内部排队队列的平滑程度, β 用于控制高队列状态下接收窗口的收缩削减幅度,对应结果分别如表 1 和表 2 所示。

表 1 参数 α 的敏感性分析

α	平均吞吐量/Mbps	平均单向时延/ms
0.1	37.856	0.129
0.2	44.484	0.143
0.3	45.464	0.134
0.4	46.812	0.142
0.5	47.167	0.143

表 2 参数 β 的敏感性分析

β	平均吞吐量/Mbps	平均单向时延/ms
0.5	38.902	0.100
0.6	39.416	0.101
0.7	41.615	0.106
0.8	44.174	0.122
0.9	45.741	0.134

可以发现,较小的 α 更偏向于平滑估计,而较大的取值则提高了队列的响应速度,平均吞吐量较高,本文推荐 α 取值范围为 0.3~0.5。

可以发现,较小的 β 更偏向于积极的队列抑制,而较大的取值则能保持较高吞吐,本文推荐 β 取值范围为 0.7~0.9。

4 未来工作

尽管 StarPEP 能够有效缓解 LEO 卫星网络中 PEP 因切分连接导致的队列积压和时延膨胀问题,但仍存在一些局限性,有待未来进一步研究。

首先,本文主要关注传送层机制,并通过可观测的带宽、时延和路径重构信息刻画下游卫星链路状态,而未显式考虑指向误差、大气衰落等物理层因素对链路质量的影响^[19-20]。这些因素可能进一步影响链路可用容量和服务能力,尤其是在未来复杂空间通

信场景中更为明显。因此,未来可进一步探索融合物理层状态信息与传送层队列状态的跨层优化机制,以提升系统在复杂空间环境下的鲁棒性。

其次,本文采用真实 Trace 的 tc 重放方式构建实验环境,能够较好地复现 LEO 卫星网络中的带宽波动、时延变化以及路径重构等关键动态特征。然而,该评估方式主要关注网络环境的动态重现,而业务流量模型仍以传统传输测试流量为主,尚未充分反映真实业务场景下复杂且具有时序相关性的流量特征。由于近些年来,面向通信网络的数字孪生技术正受到越来越多关注^[21-22],为了在虚拟环境中同时复现真实网络动态和真实业务行为,数字孪生系统通常需要兼顾网络环境建模与业务流量建模 2 个方面。因此未来工作中,计划将基于 Trace 重放的网络环境与智能流量生成技术相结合,开展更完整的 LEO 卫星网络仿真实验。

5 结束语

本文研究并实现了一种适配 LEO 卫星网络的 TCP 性能增强代理 StarPEP。该系统通过在代理节点调节接收窗口控制发送端速率,降低排队时延的同时维持较高链路利用率。StarPEP 结合稳态下的速率与队列感知窗口控制机制以及路径重构期间的感知与保护机制,使系统既能适应带宽波动,也能在周期性重构时快速恢复到新的稳定工作点,为网络运营商提供一种易部署的传送层性能优化方案。

参考文献(References)

[1] Starlink. Starlink is connecting more than 10M active customers with reliable high-speed internet across 160 countries, territories and many other markets [EB/OL]. (2026-02-14) [2026-03-01]. <https://x.com/Starlink/status/2022446814591615013>.

[2] Lai Z Q, Li Z L, Wu Q, et al. LeoCC: Making Internet congestion control robust to LEO satellite dynamics[C]// Proceedings of the ACM SIGCOMM 2025 Conference. New York: ACM, 2025: 129-146.

[3] Cao X Y, Zhang X Y. SaTCP: Link-layer informed TCP adaptation for highly dynamic LEO satellite networks[C]// Proceedings of IEEE INFOCOM 2023 - IEEE Conference on

- Computer Communications. Piscataway, NJ: IEEE, 2023: 1–10.
- [4] Zhao D, Zhang X Y, Lee M. SatPipe: Deterministic TCP adaptation for highly dynamic LEO satellite networks[C]//Proceedings of IEEE INFOCOM 2025 – IEEE Conference on Computer Communications. Piscataway, NJ: IEEE, 2025: 1–10.
- [5] Velenis D, Kalogeras D, Maglaris B. SaTPEP: A TCP performance enhancing proxy for satellite links[M]//NETWORKING 2002: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications. Berlin, Heidelberg: Springer, 2002: 1233–1238.
- [6] Caini C, Firrincieli R, Lacamera D. PEPsal: A performance enhancing proxy for TCP satellite connections[J]. *IEEE Aerospace and Electronic Systems Magazine*, 2007, 22: B–9–B–16.
- [7] Pavur J, Strohmeier M, Lenders V, et al. QPEP: An actionable approach to secure and performant broadband from geostationary orbit[C]//Proceedings 2021 Network and Distributed System Security Symposium (NDSS). Reston, VA: Internet Society, 2021.
- [8] Musk E. Inter-satellite lasers are currently only used if the satellite cannot see the user terminal and ground station simultaneously[EB/OL]. (2024–04–07) [2026–03–01]. <https://x.com/elonmusk/status/1776985584596287622>.
- [9] Garcia J, Sundberg S, Caso G, et al. Multi-timescale evaluation of starlink throughput[C]//Proceedings of the 1st ACM Workshop on LEO Networking and Communication (LEONET). New York: ACM, 2023: 31–36.
- [10] Garcia J, Sundberg S, Brunstrom A. Inferring starlink physical layer transmission rates through receiver packet timestamps[C]//Proceedings of IEEE Wireless Communications and Networking Conference (WCNC). Piscataway, NJ: IEEE, 2024: 1–6.
- [11] Garcia J, Sundberg S, Brunstrom A. A detailed characterization of starlink one-way delay[C]//Proceedings of the 2025 3rd Workshop on LEO Networking and Communication (LEONET). New York: ACM, 2025: 43–49.
- [12] Chen J Y, Li Y, Chai H Y, et al. Minimum round-trip time prediction for low earth orbit satellite networks[J]. *International Journal of Satellite Communications and Networking*, 2025, 43(4): 309–317.
- [13] Ha S, Rhee I, Xu L S. CUBIC: A new TCP-friendly high-speed TCP variant[J]. *ACM SIGOPS Operating Systems Review*, 2008, 42(5): 64–74.
- [14] Cardwell N, Cheng Y, Gunn C S, et al. BBR: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time[J]. *Queue*, 2016, 14(5): 20–53.
- [15] Xu L, Ha S, Rhee I, et al. RFC 9438: CUBIC for fast and long-distance networks[R]. Fremont: RFC Editor. 2023.
- [16] Border J, Kojo M, Griner J, et al. Performance enhancing proxies intended to mitigate link-related degradations[R]. Fremont: RFC Editor. 2001.
- [17] Caini C, Firrincieli R. TCP *Hybla*: A TCP enhancement for heterogeneous networks[J]. *International Journal of Satellite Communications and Networking*, 2004, 22(5): 547–566.
- [18] Deng Z Y, Wang Z, Wang L, et al. SatProxy: Accelerating web browsing in LEO satellite networks *via* loss-hiding[J]. *IEEE Transactions on Mobile Computing*, 2026. DOI: [10.1109/TMC.2026.3693335](https://doi.org/10.1109/TMC.2026.3693335).
- [19] 马玉娟, 李侠宇. 低轨卫星通信星地间无线信道传播特性研究[J]. *信息通信技术*, 2024, 18(5): 12–18.
- [20] Gao M, Xu G J, Song Z H, et al. Performance analysis of LEO satellite-assisted deep space communication systems[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2025, 61(5): 12628–12648.
- [21] Lai Z Q, Deng Y T, Li H W, et al. Space digital twin for secure satellite Internet: Vulnerabilities, methodologies, and future directions[J]. *IEEE Network*, 2024, 38(1): 30–37.
- [22] 赵晓红, 陈浩, 施冠炬, 等. 基于扩散模型改进的数字孪生网络流量生成技术[J]. *数据采集与处理*, 2026(4): 1103–1117.

Reconfiguration-aware TCP optimization for LEO satellite Internet

LI Zonglun¹, LAI Zeqi^{1*}, JIN Ximing², WU Qian¹, LI Hewu¹

1. Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China

2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract In recent years, low earth orbit (LEO) satellite constellations, represented by Starlink, OneWeb, China Satellite Network Group, and SpaceSail (Qianfan), have developed rapidly and are carrying an increasing volume of Internet traffic. However, the high mobility of LEO satellites causes substantial variations in link capacity, latency, and packet loss, posing significant challenges to end-to-end TCP transport. Existing end-to-end transport optimization approaches for satellite networks either require modifications and upgrades to end host kernels, making large-scale deployment difficult, or are designed primarily for conventional geostationary earth orbit (GEO) satellites and therefore cannot readily adapt to highly dynamic and time-varying LEO environments. To overcome these limitations, this work investigates TCP performance enhancement in LEO satellite networks and proposes StarPEP, a reconfiguration-aware and transparent performance enhancing proxy (PEP). StarPEP proactively regulates the sender's transmission by adjusting the TCP receive window at the proxy. By jointly exploiting the periodic reconfiguration characteristics of LEO satellite networks and dynamic variations in round-trip time (RTT) and bottleneck bandwidth, StarPEP effectively improves TCP transport performance without requiring any modification to end hosts. We implement a prototype of StarPEP on Linux. Trace-driven emulation experiments based on real-world LEO satellite network measurements show that, in terms of the trade-off between throughput and delay, StarPEP achieves closer to the Pareto frontier without modifying end hosts. Compared with the classic PEP using Hybla as its congestion control algorithm, StarPEP reduces end-to-end latency by 94.8%~95.9% while improving throughput by 49.3%~214.0%. Compared with the classic PEP using LeoCC, StarPEP reduces end-to-end latency by 89.1%~92.6% at the cost of only a small decrease in throughput.

Keywords satellite Internet; TCP performance optimization; performance enhancing proxy ●



(责任编辑 王微)