



研究与开发

长胖丢包链路下的高效传输：一种基于流编码的UDP传输方案

李业, 孙剑, 张士兵

(南通大学信息科学技术学院, 江苏 南通 226019)

摘要: 在未来的空天地一体化无线网络中, 针对高带宽延迟积 (即长胖网络) 且存在丢包特性的链路, 设计有效的数据传输方案至关重要。然而, 过去数十年作为主流传输层协议的传输控制协议 (transmission control protocol, TCP), 在该场景下具有性能上的局限性。为此, 提出了一种新的传输方案, 该方案结合了用户数据报协议 (user datagram protocol, UDP) 和应用层前向纠错 (forward erasure correction, FEC) 编码技术中的流式编码, 能够在完全不需要重传的情况下提供可靠的数据传输保障。此外, 还引入了一种基于强化学习 (reinforcement learning, RL) 的方法, 以实现拥塞控制与FEC修复传输的联合优化。通过一系列仿真实验, 证明了所提方案在长胖网络环境中, 尤其在面对丢包情况时, 能够保持高效和平稳的吞吐量。这也为TCP性能增强代理 (performance-enhancing proxy, PEP) 的应用提供了新思路。最后, 还展望了一些未来的研究方向。

关键词: 前向纠错; 拥塞控制; 强化学习

中图分类号: TN927.2

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2025040

Efficient packet transmission over long-fat lossy links: research and development

LI Ye, SUN Jian, ZHANG Shibing

School of Information Science and Technology, Nantong University, Nantong 226019, China

Abstract: Designing efficient packet transmission over lossy links with a large bandwidth-delay product (i.e., long-fat) is important in the next-generation wireless networks, which are envisioned to be of a space-air-ground integrated scale. Transmission control protocol (TCP), the dominant transport layer protocol in the past decades, is known to have had performance issues in such links. A new transport approach was proposed using user datagram protocol (UDP) along with an application-layer forward erasure correction (FEC) code called streaming code to provide retransmission-free reliability. A novel approach based on reinforcement learning (RL) was proposed to jointly per-

收稿日期: 2024-11-26; 修回日期: 2025-02-19

通信作者: 张士兵, zhangshb@ntu.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61801248)

Foundation Item: The National Natural Science Foundation of China (No.61801248)

form congestion control and FEC repair transmissions. Proof-of-concept simulations demonstrate that the approach can achieve high and smooth goodput in long-fat lossy links, which can be attractive for performance-enhancing proxy (PEP). It also highlights several future research directions that can be exploited based on the proposed approach.

Key words: forward erasure correction, congestion control, reinforcement learning

0 引言

预计下一代无线网络将具备空天地一体化的规模,其中卫星将在为用户提供连续服务方面发挥关键作用^[1]。由于卫星链路通常伴随着较高的丢包率和较大的带宽延迟积 (bandwidth-delay product, BDP),其通常会呈现出“长胖”特性。针对这些链路实现高效的数据传输变得尤为重要。优化卫星链路的性能不仅直接关系到网络的整体效率,也是确保用户服务质量/体验质量 (quality of service/quality of experience, QoS/QoE) 的基础^[2]。

在长胖丢包链路中,目前多数应用所依赖的传输控制协议 (transmission control protocol, TCP) 在可靠数据包传输方面面临带宽利用率低的问题^[3]。一方面, TCP 是为适应有线链路而设计的技术,其核心关注点在于解决拥塞问题,因此大多数 TCP 变体都将数据包丢失视为拥塞的标志,进而通过降低传输速率来实施拥塞控制 (congestion control, CC)。然而,在无线链路环境中,数据包丢失往往是由随机错误引起的,并非由拥塞造成,这导致了不必要的传输速率下降。特别是在许多新兴的空天地一体化网络应用场景中,这个问题更加突出。由于较长的传播延迟使得利用链路层自动重传请求 (automatic repeat request, ARQ) 变得不切实际,由链路错误引发的数据包丢失难以避免。另一方面,为了防止拥塞, TCP 在传输初期会采取逐步增加发送速率的策略,即“慢启动”,但在带宽和传播延迟均较大的长胖链路中,这一过程可能导致较长时间内无法充分利用链路容量^[4]。特别是当连接持

续时间较短时,这个问题的影响尤为显著。

在过去几十年里,为了提升长胖链路的性能,研究者提出了多种 TCP 拥塞控制变体。例如, TCP Hybla 就是专门为卫星链路设计的^[5]。在 TCP Hybla 中,拥塞窗口 (congestion window, CWND) 的增量不再依赖于实际的往返路程时间 (round trip time, RTT),从而能够更快地使链路达到饱和状态。而谷歌推出的 TCP BBR 则通过交替估算可用带宽 (available bandwidth, AvailBW) 和最小 RTT,使在途数据量接近 BDP,并且 CWND 的变化不会受数据包丢失的影响^[6]。此外,基于机器学习 (machine learning, ML) 的各类 CC 方案也得到了发展,其核心理念是利用训练过的智能体根据吞吐量、RTT 等观测指标来调整 CWND 的增减^[7]。

近年来,随着用户数据报文协议 (user datagram protocol, UDP) 再次引起关注,关于数据包传输的新思路不断涌现,这很大程度上得益于 Google 首次提出的快速 UDP 互联网连接协议 (quick UDP internet connections, QUIC)。QUIC 被视为未来互联网中 TCP 的一种替代选择,其完全基于 UDP 构建。QUIC 最初的设计目的是利用 UDP 的无连接性和无序性特点,以减少 TCP 3 次握手带来的延迟,并解决 TCP 在处理多路复用 HTTP 流时出现的队头阻塞 (head-of-line blocking, HoL blocking) 问题。UDP 在部署和功能扩展方面的高度灵活性,使其成为开发新型传输协议的理想平台,尤其是它能够在用户空间而非操作系统内核中运行,这一点在快速变化的未来网络环境中显得尤为重要^[8]。为了使基于 UDP 的协议能够支持广泛的应用,一个关键步骤是在这些协议



中加入拥塞控制和可靠性机制。在QUIC的标准化过程中,采用了NewReno拥塞控制算法,并沿用了TCP的重传机制^[9]。

此外,为进一步优化传输层性能,性能增强代理(performance enhancing proxy, PEP)已成为一种常见方案^[10]。PEP通过接收并终止普通TCP流量,将其转换为适合链路特性的协议或TCP变体,从而避免传统传输协议在长胖链路中面临的性能瓶颈。当流量离开链路时,PEP再将其转回原始形式。此方案不仅提高了协议部署的灵活性和透明度,还有效缓解了传输层的僵化问题。在PEP架构中,其核心指标包括吞吐量的效率、公平性及稳定性,尤其是稳定性,直接影响缓冲区需求和数据包间的延迟。因此,PEP的设计目标是在确保高效传输的同时,维持较低的延迟。

本文提出了一种新的基于UDP的传输框架,与现有方案显著不同的是,本文框架采用了应用层前向纠错(forward erasure correction, FEC)技术来恢复因链路错误或拥塞引发的数据包丢失。该框架结合了强化学习(reinforcement learning, RL)技术,通过动态调整拥塞控制与FEC修复的策略,实现更加灵活和高效的传输。研究表明,融合了FEC与RL技术的UDP方案,相较于传统方案,在吞吐量提升及稳定性方面表现出显著优势,尤其适用于PEP环境。通过网络功能虚拟化(network functions virtualization, NFV)技术,本文提出的框架能够灵活地作为卫星回程链路上的虚拟PEP进行部署^[11],具有广泛的应用前景。

1 结合FEC的UDP:背景介绍

与重传机制相比,FEC通过在正常数据流中插入修复数据包来恢复丢失的数据包,这些修复数据包是依据特定算法对源数据包进行编码后生成的。这里提到的FEC不同于传统的物理层信道

编码,它指的是基于数据包级别的编码,即在传输层或更高层次的数据包间进行编码。基于UDP的一个重要FEC示例是Raptor编码,它能够将一组源数据包编码成理论上无限数量的修复数据包。另一个示例是随机线性网络编码(random linear network coding, RLNC),它允许在中间节点上执行重新编码。

然而,Raptor编码和RLNC都是块编码方式,这意味着一组数据包只有在全部解码完成后才会一并交付给上层应用,导致解码延迟与块大小直接相关。对于需要平滑数据包传输的应用来说,这种延迟可能构成负担,尤其是在编码开始时并非所有数据都已经准备好的情况下,这种方式显得更加不切实际。PEP便是一个典型的应用场景,其中数据包按需到达入口点,而这些数据包应当尽可能迅速地由远程PEP节点发出。虽然系统编码通过先发送源数据包再发送修复数据包的方式可以在一定程度上减轻这一问题,但仍然需要在发送修复数据包前确定一个合理的块长度。

为了克服块编码的延迟问题,流编码(streaming coding, SC)得到了广泛研究,它通过在一个弹性的滑动窗口内,从源数据包生成修复数据包,减少FEC的块解码延迟。相关研究表明,可以通过在数据包流中固定或随机位置插入修复数据包的方式,实现有限的“有序”延迟^[12-13]。然而,这些方案需要提前设置修复数据包的插入频率,以便与基本的数据包丢失率相匹配。由于实际中的链路和拥塞条件可能会发生变化,传输层观察到的数据包丢失率也随之波动,要能够实现目标有序延迟的最佳插入频率是非常困难的。基于数据包确认(ACK)的方案表明自适应修复数据包插入是可行的^[14]。但是,现有方案主要关注丢失数据包的恢复,而没有充分考虑拥塞控制问题。鉴于UDP本身并不包含内置的拥塞控制机制,确定适当的数据包发送速率具有挑战,

特别是在网络可用带宽频繁变化的情况下。

由于在 QUIC 框架中配备了相应的拥塞控制机制，文献[15]和文献[16]分别提出了两种 FEC 方案，通过在常规 QUIC 数据包中嵌入修复数据包来恢复丢失的数据包。具体而言，文献[15]采用了固定速率的块编码 FEC，而文献[16]则实现了基于自适应滑动窗口的异或 FEC。这两种方案在无线环境下的丢包场景中显著缩短了传输完成时间，表现优于原始的 QUIC。然而值得注意的是，这两种 FEC 方案都是在拥塞控制机制之下运行的，意味着当 FEC 未能成功恢复丢失的数据包时，仍然会触发重传机制。由于 QUIC 的拥塞控制直接来源于 TCP，这可能导致在某些情况下出现不必要的传输速率下降。

2 SC-UDP 传输方案

流编码 UDP (SC-UDP) 是一种基于 UDP 的无重传的传输方法，通过 FEC 技术恢复由随机链路错误和拥塞导致丢失的数据包。该方案将 CC 与 FEC 修复传输相结合，并由适用于在线连续任务的 RL 技术实现。SC-UDP 的整体架构如图 1 所示。SC-UDP 的两大核心组件包括：

(1) 流编码，旨在实现平滑的丢包恢复同时保持较低的解码延迟；

(2) 强化学习，用于联合优化 FEC 和 CC。

接下来，将对这两个组件进行详细介绍。

2.1 流编码传输

(1) 编码

在发送端，有一个按照数据到达时间存储数据的队列（如 PEP 滤波器）。这些队列的数据被表示为源数据包，每个数据包包含等量的比特位，在添加必要的底层协议头后，可以适配到底层网络的最大传输单元（maximum transmission unit, MTU）。接着，通过 RL 训练的“智能体”，在图 1 所示的 3 个选项中选择操作，操作的时间间隔与链路的 AvailBW 成反比，具体估计方法将在后文讨论。假设智能体选择了操作 1，即发送一个源数据包。此时，根据最近发送的源数据包的记录索引（index, ID），队列中下一个未以源数据包形式发送的连续源数据包将被发送出去。也就是说，队列中的每个源数据包将以未编码的形式准确发送一次。若智能体选择操作 2，则会生成一个修复数据包。这个修复数据包是对之前已发送的源数据包进行随机线性组合的结果：编码过程将每个源数据包的信息位视为来自有限域的元素向量，并从同一有限域中随机选取编码系数来线性组合这些向量。修复数据包将被赋予一个顺序递增的 ID。数据包中包含的源 ID 范围被称为该修复数据包的“编码窗口”（encoding window, EW）。如果智能体选择操作 3，发送端将跳过当前操作，保持空闲状态。

当执行操作 1 或操作 2 时，相应的源数据包或修复数据包在附加了如图 1 所示的头部之后，会被加入 UDP 传输缓冲区（transmission buffer，

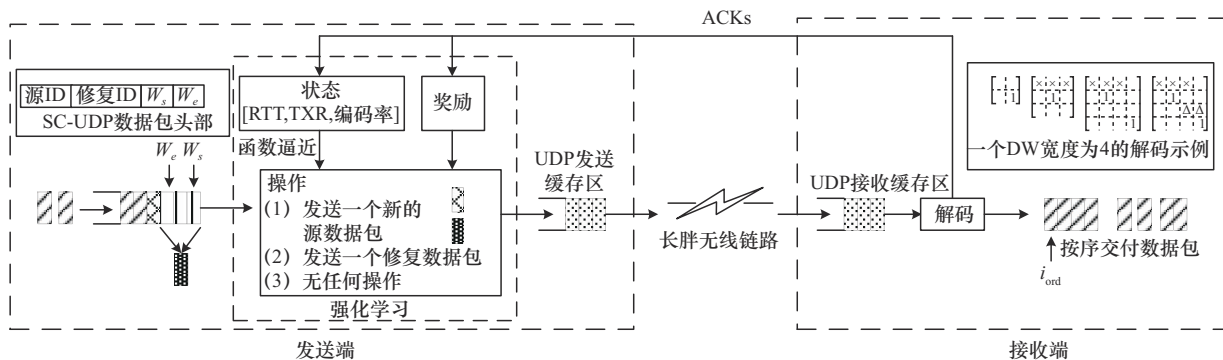


图1 SC-UDP 的整体架构



TxBuffer)。该头部包含4个字段，对于源数据包，第一个字段用于指示源ID；而对于修复数据包，该字段留空，修复ID以及对应的EW的边界信息则存储在剩余3个字段中。值得注意的是，在实际操作中，通常采用伪随机数生成器(pseudo-random number generator, PRNG)来选择编码系数，并在传输前就PRNG的种子达成一致。因此，无须在头部携带这些系数，接收端可以根据修复ID从PRNG中恢复出相应的系数。

(2) 解码

由于UDP的丢包特性和无序传输性质，并不是所有发送出去的数据包都能到达接收端，而且有些数据包可能会以不同于发送时的顺序到达。为了向应用层提供有序的字节流接口，需要一个解码器来解码并按照原始顺序传递源数据包。接收端会追踪一个称为“有序ID”的标识，这是最新按顺序传递给应用层的源数据包的ID。初始状态下，接收端处于有序模式。如果接下来接收到的数据包的源ID能够紧接当前的有序ID，或者一个修复数据包，并且其EW的右边界等于有序ID(即冗余编码数据包)，则保持有序模式；反之，若上述条件不满足，则接收端切换至无序模式。在这种模式下，一个在线高斯消元(on-the-fly Gaussian elimination, OF-GE)解码器将被激活，用于缓存新到达的数据包并尝试解码。解码器通过一系列行操作，构建一个线性方程组，其中接收数据包的编码系数形成“解码矩阵”的行，而接收数据包的(编码)信息符号则构成方程右侧的“信息矩阵”。图1展示了一个随着时间变化的解码矩阵示例(从左至右)。当某个源数据包丢失时(如左侧矩阵中的空白行所示)，解码器将被激活。在此示例中，4个数据包被缓存直至解码成功，其中包括两个无序源数据包，以及两个修复数据包的编码系数(分别用叉号和三角表示，且已经过行操作处理)。解码窗口(decoding window, DW)定义为从有序ID之后的第一个源ID

(不包括有序ID本身)到所有缓存数据包EW内最大源ID(包括此ID)之间的源ID范围。因此，图1中的示例展示了一个宽度为4的DW。值得注意的是，随着更多数据包的累积，DW的范围可能会扩大(如图1中从2扩展到4)。通过OF-GE技术，解码器能够保持解码矩阵的上三角形式。一旦解码矩阵的所有对角线元素都非零，则表明解码成功，此时解码器将停止工作。随后，DW内的所有数据包将一同传递给应用层，系统恢复到有序模式，新的有序ID被设定为DW的右边界。有关解码过程的详细说明，请参考文献[12-13]。

流编码的一个关键性能指标是“有序延迟”，它是指源数据包首次以未编码形式发送的时间与该数据包在接收端按顺序传递给应用层的时间之间的差值。有序延迟的波动不仅受修复数据包在数据流中插入比例的影响，同时也受链路拥塞状况的影响。较低的有序延迟意味着用户能够体验到更加平稳的“有效吞吐量”，即单位时间内按序交付的数据包数量。

2.2 基于RL的FEC和拥塞控制

为了实现高效且平稳的有效吞吐量，选择合适的FEC编码率和维持较低的拥塞水平至关重要。采取恰当的动作是实现这一目标的关键。其中，动作1和动作2的相对频率决定了FEC编码率，而动作3的频率则影响着拥塞状况。SC-UDP采用RL技术来学习最优动作。根据标准的RL方法，SC-UDP通过接收端返回的ACK来观察系统的“状态”，并通过学习动作价值函数来决定“动作”，旨在最大化“奖励”。下面将详细介绍这些要素的具体内容。

(1) ACK与状态

接收端的ACK必须包含能够反映拥塞状况和接收端有序传输进度的关键信息。在多种可能的选择中，当前ACK携带的信息包括但不限于以下几项，这些信息既可以在固定的时间间隔发送，也可以在信息发生变化时发送：①最新的有序ID；

②最新接收数据包的ID及其类型；③到目前为止接收到的源数据包和修复数据包的数量。

通过适当的记录，发送方可以确定接收方最新接收数据包的发送时间，以及截至该时间点采取的3种动作的数量。随后，发送方可以通过计算以下指标来表征系统状态：①RTT：当前时间与最新ACK数据包发送时间之间的差值；②传输比率（transmission ratio, TXR）：在最新ACK数据包发送时间之前发送的总数据包数量与该时间段内采取的总动作数量之比；③最新ACK数据包发送时间之前的编码率，定义为已发送的源数据包数量与已发送的总数据包数量之比。

该状态可以通过一个三元组来描述，其中包括：RTT比率，即最新的RTT估计值与迄今为止观察到的最小RTT之比，用于评估当前的拥塞状况、TXR、编码率。其中，TXR和编码率分别反映了连接的利用率和FEC编码的比例。

(2) 动作

在SC-UDP中，由于不涉及重传，动作与ACK是异步执行的，而不是由ACK触发。这意味着需要明确指定一个动作间隔。通常情况下，智能体需要在指定的动作间隔内采取动作，以确保通过发送的数据包能充分利用AvailBW。在SC-UDP中，动作间隔被设定为估计的AvailBW（每秒数据包数）的倒数乘以一个比例因子。这个比例因子可以通过对比最新ACK中报告的接收数据包数量，并采用适当的滑动平均方法来获取（初始阶段使用一个较小的默认间隔来得到第一次的AvailBW估计值）。动作间隔的比例因子范围从0到1，其目的是确保链路不会因为采取动作3（即退避）而出现空闲状态。如果直接使用AvailBW的倒数作为精确的动作间隔，那么后续的AvailBW估计值将持续下降。

(3) 目标与奖励

目标函数用于评估系统状态，并据此确定RL中的奖励值。针对不同的端到端性能目标，可

以选用多种目标函数。在SC-UDP中，采用的目标函数旨在实现比例公平性，具体形式为对吞吐量与指定RTT幂次比值的对数函数，其中幂次反映了RTT的重要性^[7]。这里的吞吐量是从发送方视角定义的，指的是ACK确认的有序传输数据包数与当前时间的比值。其中，采用当前时间而非最新ACK确认数据包的发送时间，是为了确保在没有收到新的ACK时，目标值能够自然下降。

SC-UDP的奖励基于每个发送数据包贡献的平均有效吞吐量（average goodput, avgGP），取值范围在0到1之间。具体来说：

- 如果目标值增加，则对avgGP给予正向奖励；
- 如果目标值减少且链路出现拥塞（RTT比率超过一个可调的超参数阈值），则将avgGP减去1，给予负向奖励；
- 此外，奖励为0。

(4) 学习

为了实现有效的学习，状态元组被映射为离散特征向量（如通过分区编码实现）。在此基础上，智能体可以采用标准的RL流程，使用固定的学习率来估计动作值函数（以持续跟踪非平稳环境），同时应用奖励衰减因子。为了在基于估计的动作值函数做出决策时达到探索与利用的平衡，可以采用 ϵ -贪婪策略。类似于许多基于反馈的RL辅助CC的方案，在SC-UDP中，奖励的发放会因为RTT而有所延迟。在这种情况下，资格迹具有参考历史信息的优势^[17]。例如，采用Sarsa(λ)算法可以获得更优的在线性能。

3 性能评估

作为概念验证，在ns-3仿真平台上评估了SC-UDP。首先，在一个标准的点对点长胖丢包链路中对该方案进行了评估，该场景中只有一个流，因此不存在拥塞问题。链路参数设置见表1。为了简化模型，哑铃拓扑结构中的叶节点链路被



设置为无丢包。参数均是由在地球静止轨道卫星 ChinaSat-16 上租用链路进行的实际测试结果确定的。此外,针对 RL 的超参数选择(已验证适用于各种链路条件)也在表 1 中进行展示。ns-3 中的“TrafficControl”层的参数和配置(如缓存区大小、队列调度策略等)均使用了默认设置。流编码采用的有限域大小为 256。在 SC-UDP 的实现中,采用了平铺编码和 Sarsa(λ)算法。

表 1 链路参数设置

参数	值
带宽	20 Mbit/s
往返传播延迟	600 ms
前向丢包率	0.3%
反向(ACK)丢包率	0.3%
学习率	0.1
奖励衰减因子	0.95
目标函数中 RTT 权重	0.1
探索率	0.1
轨迹衰减率	0.5
初始动作间隔	2.4 ms
动作间隔缩放因子	2/3
RTT 比率拥塞阈值	1.2
平铺数量	4
每组平铺数量	125

为了验证提出的 SC-UDP 方案的有效性,将其与多种现有传输协议进行了比较,包括 TCP Hybla、TCP BBR^[18]、TCP CUBIC、QUIC^[19]、rQUIC^[16]以及 QUIC-Go。其中, TCP Hybla 是特别为适应卫星链路特性而设计的一种变体,而 TCP BBR 和 TCP CUBIC 则是目前的主流方案。此外, QUIC 及其变体(rQUIC 和 QUIC-Go)均适用于具有较高丢包率和高延迟的卫星网络。SC-UDP 与对比方案的吞吐量对比如图 2 所示,展示了不同方案在发送最多 40 000 个应用层数据包(每个大小为 1 400 byte)时,吞吐量随时间变化的情况,同时整个传输过程大约在 50 s 内完成。由于 ns-3 中缺乏 rQUIC 模块,其结果是通过在 Mininet 中使用已有的开源代码进行仿真得出

的。其中,仿真时采用了与 ns-3 相同的拓扑结构和链路条件。在所有方案的仿真过程中,将默认的套接字发送/接收缓存区大小调整为 60 MB,足够存储 40 000 个数据包,确保流的传输完全由 CC 管理,不受上层流控制且避免目的地溢出的影响。本文吞吐量定义为按序交付到应用层的有效数据包数量与经过时间的比值。需要注意的是,这里的吞吐量统计仅包含源数据包,不包括用于纠错的修复数据包,从而确保该指标能准确反映用户接收到的实际有效负载量。从图 2 中可以看出, SC-UDP 在整个传输过程中实现了较高且平稳增长的吞吐量。在对比方案中, TCP Hybla 表现出了最高的吞吐量,尤其在长胖链路慢启动阶段的改进上,吞吐量迅速上升。实验证明, QUIC-Go 能够提升吞吐量,但由于当前 QUIC 实现中采用的是类似 NewReno 的 CC 算法,提升效果有限。值得注意的是, rQUIC 和 QUIC 也可以使用其他 CC 算法(如 Hybla 和 BBR),这将进一步提高吞吐量,但在本次实验中,保留了文献[16]和文献[19]的源码设置。

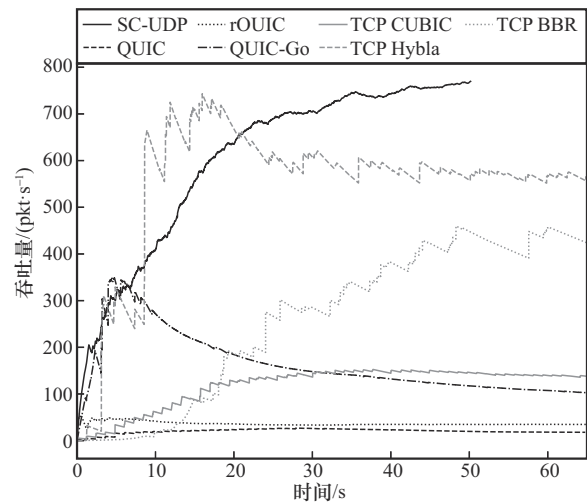


图 2 SC-UDP 与对比方案的吞吐量对比

除了观察到 TCP Hybla 在传输初期吞吐量较高之外,还注意到其吞吐量曲线呈现锯齿状,这表明数据包在按序交付给上层的过程中存在突发

性。不同方案的按序交付数据包数量随时间变化对比如图3所示。通过观察图3中按序交付的数据包数量随时间的变化曲线，可以清晰地看出SC-UDP的曲线更为平滑，相比之下，其他方案的曲线则呈现阶梯状。实际上，如果对比整个传输过程中观察到的最大交付时间间隔，SC-UDP的时间间隔小于200 ms，而TCP Hybla和TCP BBR的时间间隔有时 would 超过2 s。

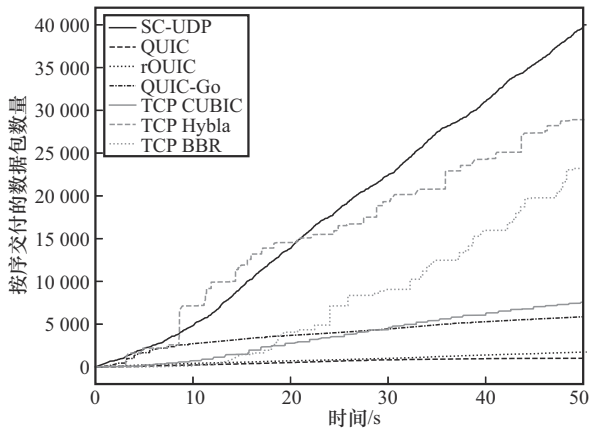


图3 不同方案的按序交付数据包数量随时间变化对比

SC-UDP随时间变化的FEC编码率和TXR如图4所示。FEC编码率是从接收端的角度进行绘制的，可计算为接收到的源数据包数量除以接收到的总数据包数量。从图4中可以看出，编码率趋向于0.5，这表明实际吞吐量几乎是理想吞吐量的两倍。在给定20 Mbit/s的带宽和1 400 byte的MTU条件下，这个数值接近链路的AvailBW，约为每秒1 600个数据包。因此，可以证明SC-UDP能够充分利用有损链路。此外，学到的FEC编码率为0.5实际上是对链路随机丢包率（仅为0.3%）的一种过高估计。这一点是因为选择了特定的RL奖励函数，该函数旨在整个传输过程中维持平稳的高吞吐量，因此经常插入修复数据包以降低按序交付的延迟^[13]。

TXR表示用于发送数据包的动作比例（包括发送方为避免拥塞而采取的暂停动作）。从图4中可以看出，传输初期TXR值较高，表明智能体向

链路注入了大量数据包，以充分利用 AvailBW。从FEC编码率曲线可以看出，这些数据包大多为源数据包。随着训练的进行，TXR值逐渐趋于稳定，并最终收敛至约0.66。需要注意的是，这一变化体现了智能体在与网络环境的实时交互中，持续根据链路状态调整发送策略。随着训练的推进，TXR值逐步收敛至稳定值，显示了智能体在不断的在线学习和优化中学会了如何平衡网络资源的利用与延迟控制。此外，由于动作间隔缩放因子设置为2/3，TXR值的收敛进一步表明智能体成功地优化了其传输策略，避免了过度拥塞，同时保持了高吞吐量。

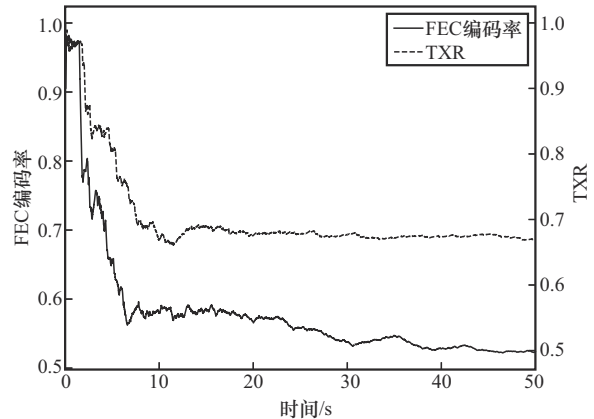


图4 SC-UDP随时间变化的FEC编码率和TXR

不同方案的吞吐量和Jain公平性指数对比如图5所示，比较了当不同数量的数据流共享同一链路时，SC-UDP与TCP Hybla和TCP BBR的表现。实验采用哑铃拓扑结构，瓶颈链路与图2中的设置相同。每对叶节点代表一个发送-接收对，即一个流。为了简化模型，叶节点链路被设置为无损，带宽为40 Mbit/s，延迟为5 ms。叶节点的套接字发送/接收缓冲区大小设为4 MB，以此来模拟流在PEP入口处占用缓冲区的情况。为了使仿真更加贴近实际情况，发送方在0~20 s随机开始发送数据，选择50~100 s的时间段来对比它们的吞吐量。图5使用堆叠柱状图展示了该时间段内SC-UDP、TCP Hybla和TCP BBR的吞吐量，

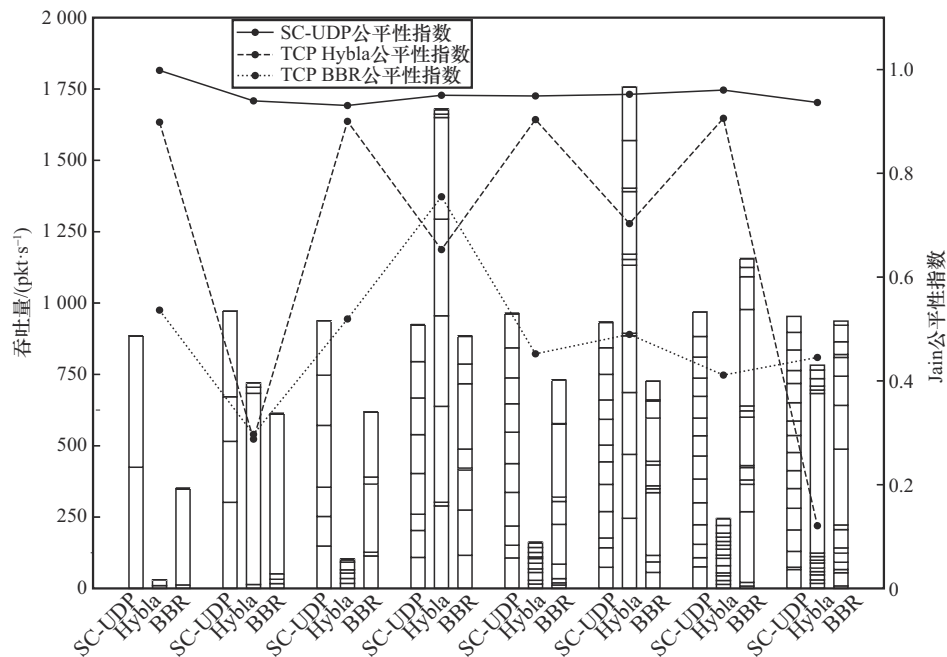


图5 不同方案的吞吐量和Jain公平性指数对比

并计算了各方案的Jain公平性指数。SC-UDP在大多数情况下实现了比TCP Hybla和TCP BBR更高的总吞吐量。尽管TCP Hybla在某些场景下(如8对和12对)能够达到较高的吞吐量,但其表现不够稳定,在其他场景下(如2对、6对、10对和14对)吞吐量较低。相比之下,SC-UDP和TCP BBR的表现更为稳定。特别是,SC-UDP流能够更公平地共享长胖丢包链路,这显示出其作为传输层方案的巨大潜力。

4 开放问题与研究方向

尽管SC-UDP在长胖丢包链路中展现出了有效前景,但为了进一步改进并实现实际的部署,仍有许多问题需要解决。本文将重点关注其中存在的一些关键问题。

(1) FEC的代价

仅依赖FEC来恢复由链路随机丢包和拥塞导致的数据包丢失,确实需要承担一定的成本。首先是编码和解码的计算开销,这一开销与EW和DW的宽度成正比,并受丢包率和修复数据包包括

入频率的影响^[12-13]。对于较大的EW和DW,该方案可能会产生过重的CPU负载。一个潜在的解决方案是从EW中选择“子集”来生成稀疏的修复数据包,但这需要精心设计,否则可能会导致修复数据包的有效性降低。

第二个代价是接收端的缓存需求。其中一部分缓存用于解码器,其大小与DW宽度成正比。通过稀疏编码和对OF-GE的精心实现,结合适当的数据结构,是保持低内存消耗的关键。另一部分缓存则用于暂时存储已按序交付的数据包。由于传播延迟,发送端可能会对已经按序到达但尚未收到ACK的数据包生成修复数据包。如果在解码后立即删除已按序交付的数据包,这些修复数据包可能无法正确解码,从而不得而被丢弃,造成传输资源的浪费。因此,这部分缓存需求与链路的RTT成正比。

(2) 学习优化

尽管RL已被证明能够支持联合自适应的CC和FEC修复传输,但目前SC-UDP的RL设计仅作为一个概念验证示例,未来还有大量的研究空

间以进一步优化学习性能。其中,可能的优化方向包括探索更有效的状态表示方法、目标函数和奖励函数设计以及动作值函数的近似方法。

改进状态表示的一个例子是在ACK中加入显式的解码器状态以及按序交付延迟的选定统计信息。目标函数和奖励函数的选择对公平性、FEC编码率以及良好吞吐量有着重要影响。然而,类似于大多数基于RL的方案,寻找更优的目标函数和奖励函数通常依赖于启发式和试错的方法,这需要大量的调试工作。同时,参数调试的可解释性也是一个挑战。在当前对SC-UDP的评估中,采用了平铺编码来进行动作值函数的近似,这是一种线性近似方法。此外,使用非线性函数近似,特别是神经网络(通常被称为深度RL),可能是一个值得探索的方向,有助于提升SC-UDP的学习性能。

(3) TCP友好性

SC-UDP的设计初衷是在长胖丢包链路上提高PEP的性能。在这些场景中,链路上的TCP连接通常会被终止,并转换为SC-UDP流。因此,SC-UDP流之间的公平性成为主要关注点。

然而,SC-UDP与常规TCP连接的共存也是一个值得研究的课题。在这种情况下,SC-UDP可能对TCP流不公平。原因在于SC-UDP对丢包非常不敏感,这影响了其探测AvailBW的能力。此外,SC-UDP的发送速率只有在智能体通过高RTT比率检测到拥塞时才会下调。然而,这时数据包可能已经开始丢失,促使TCP连接降低其传输速率。因此,SC-UDP流可能会观察到AvailBW的增加并发送更多的数据包,从而进一步挤压TCP连接的带宽。

解决这一问题的关键在于使SC-UDP能够准确估计AvailBW,同时避免拥塞导致的数据包丢失。尽管端到端的解决方案看起来极具挑战性(类似于TCPBBR与普通TCP共存时遇到的问题),一种可能的解决办法是引入特定的主动队

列管理(active queue management, AQM)机制。这种机制可以在中间节点检测到拥塞时,优先丢弃SC-UDP数据包而不是TCP数据包。然而,这需要对中间节点进行相应的修改。

(4) 结合QUIC

SC-UDP的核心组件,即流编码/解码和基于RL的联合FEC与CC,可按模块化的方式实现。一个有趣的研究问题是探讨能否以及如何将其应用于增强QUIC,例如作为QUIC当前拥塞控制和重传机制的替代方案。在QUIC现有的框架中,包括其数据加密、标准化的数据包格式和连接维护过程,已经相当成熟^[9]。这些特性将使SC-UDP的部署变得更加便捷。此外,基于开源的QUIC项目(如PicoQUIC)进行QUIC与SC-UDP结合的实验正在进行中。

5 结束语

为了满足即将到来的空间信息网络中的严格要求,长胖丢包链路(如卫星链路)上的数据包传输协议需要进行改进。近年来工业界对UDP的重新关注,以及FEC和ML技术的发展,为长期以来依赖的TCP框架革新提供了新的机遇。本文提出了一种基于UDP和FEC的全新传输层设计方案,并证明了其在长胖丢包链路中能够实现高效且平稳的良好吞吐量,同时也适用于PEP。此外,本文也探讨了该框架在实际部署中仍需解决的一些开放性问题。值得相信的是,UDP、FEC和ML技术将在未来的空天地一体化空间网络传输层创新中发挥关键作用。

参考文献:

- [1] 张更新, 廖磊瑶, 何元智. 面向空天地海一体化的卫星通信关键技术研究[J]. 电信科学, 2024, 40(6): 11-24.
ZHANG G X, LIAO L Y, HE Y Z. Study on the key technologies of satellite communication for space-air-ground-sea integration[J]. Telecommunications Science, 2024, 40(6): 11-24.
- [2] JIANG D D, WANG F, LV Z H, et al. QoE-aware efficient con-

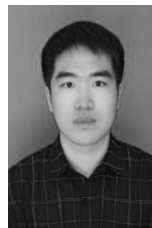


- tent distribution scheme for satellite-terrestrial networks[J]. IEEE Transactions on Mobile Computing, 2023, 22(1): 443-458.
- [3] LOW S H, PAGANINI F, DOYLE J C. Internet congestion control[J]. IEEE Control Systems Magazine, 2002, 22(1): 28-43.
- [4] KACHOOEI M A, CHUNG J, LI F, et al. SEARCH: robust TCP slow start performance over satellite networks[C]//Proceedings of the 2023 IEEE 48th Conference on Local Computer Networks (LCN). Piscataway: IEEE Press, 2023: 1-4.
- [5] 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.
- [6] CARDWELL N, CHENG Y C, GUNN C, et al. BBR: congestion-based congestion control: measuring bottleneck bandwidth and round-trip propagation time[J]. Communication of the ACM, 2017, 60(2): 58-66.
- [7] WINSTEIN K, BALAKRISHNAN H. TCP ex machina: computer-generated congestion control[C]//Proceedings of the ACM SIGCOMM 2013 conference on SIGCOM. New York: ACM Press, 2013: 123-134.
- [8] POLESE M, CHIARIOTTI F, BONETTO E, et al. A survey on recent advances in transport layer protocols[J]. IEEE Communications Surveys & Tutorials, 2019, 21(4): 3584-3608.
- [9] LYENGAR J, THOMSON M. OUIIC: a UDP-based multiplexed and secure transport[M]. Fremont: Internet Engineering Task Force, 2020: 1-151.
- [10] 梁向斌, 赵宝康, 彭伟. 卫星网络传输优化新机制研究进展[J]. 计算机工程与科学, 2023, 45(11): 1949-1959.
- LIANG X B, ZHAO B K, PENG W. Recent progress of new technologies for satellite network transport protocol optimization[J]. Computer Engineering & Science, 2023, 45(11): 1949-1959.
- [11] BUJARI A, LUGLIO M, PALAZZI C E, et al. A virtual PEP for web optimization over a satellite-terrestrial backhaul[J]. IEEE Communications Magazine, 2020, 58(10): 42-48.
- [12] KARZAND M, LEITH D J, CLOUD J, et al. Design of FEC for low delay in 5G[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(8): 1783-1793.
- [13] LI Y, ZHANG F F, WANG J, et al. On streaming coding for low-latency packet transmissions over highly lossy links[J]. IEEE Communications Letters, 2020, 24(9): 1885-1889.
- [14] COHEN A, MALAK D, BRACHA V B, et al. Adaptive causal network coding with feedback[J]. IEEE Transactions on Communications, 2020, 68(7): 4325-4341.
- [15] MICHEL F, CONINCK Q D, BONAVENTURE O. OUIIC-FEC: bringing the benefits of forward erasure correction to QUIC[C]//Proceedings of the 2019 IFIP Networking Conference (IFIP Networking). Piscataway: IEEE Press, 2019: 1-9.
- [16] GARRIDO P, SANCHEZ L, FERLIN S, et al. rOUIIC: integrating FEC with OUIIC for robust wireless communications[C]//Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE Press, 2019: 1-7.
- [17] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. Cambridge MA: MIT Press, 1998.
- [18] CLAYPOOL M, CHUNG J W. BBR': an implementation of bottleneck bandwidth and round-trip time congestion control for ns-3[C]//Proceedings of the 2018 Workshop on Ns-3. New York: ACM Press, 2018: 1-8.
- [19] BIASIO A D, CHIARIOTTI F, POLESE M, et al. A QUIC implementation for ns-3[C]//Proceedings of the 2019 Workshop on Ns-3. New York: ACM Press, 2019: 1-8.

[作者简介]



李业 (1986-), 男, 博士, 南通大学信息科学技术学院副教授、博士生导师, 主要研究方向为无线通信、前向纠错编码、网络编码等。



孙剑 (1999-), 男, 南通大学信息科学技术学院硕士生, 主要研究方向为无线通信、网络编码等。



张士兵 (1962-), 男, 博士, 南通大学信息科学技术学院教授、博士生导师, 主要研究方向为宽带数字通信、智能信号处理、认知无线电等。