https://crad.ict.ac.cn
62(10): 1−15, 2025

大规模互联网服务的低时延传输技术

吕格瑞[1,2] 赵员康[1,2] 张佳兴[1,2] 潘 恒[2,3] 武庆华[1,2] 李振宇[1,2]

[1](中国科学院计算技术研究所 北京 100190)

[2](中国科学院大学 北京 100049)

[3](中国科学院计算机网络信息中心 北京 100083)

(lvgerui@ict.ac.cn)

Low-Latency Transmission Techniques for Large-Scale Internet Services

Lü Gerui[1,2], Zhao Yuankang[1,2], Zhang Jiaxing[1,2], Pan Heng[2,3], Wu Qinghua[1,2], and Li Zhenyu[1,2]

[1](*Institute of Computing Technology*, *Chinese Academy of Sciences*, *Beijing* 100190)

[2](*University of Chinese Academy of Sciences*, *Beijing* 100049)

[3](*Computer Network Information Center*, *Chinese Academy of Sciences*, *Beijing* 100083)

Abstract The Internet has become one of critical infrastructures in modern society, in which transmission performance directly determines the capability and value of the services it carries. With the rapid development of mobile and interactive Internet applications, low-latency transmission in large-scale Internet faces three technical challenges at the application layer, transport layer, and network layer, that are the mismatch between dynamic sending decisions of application data and available network resources, the incoordination between network transmission decisions and changes in network status, and the inconsistency between differentiated services and the processing logic of network devices. To address the bottlenecks of traditional hierarchical and single-point transmission optimization, we propose and construct a collaborative transmission technology system for large-scale Internet services. It introduces the technological achievements within this system in three aspects: intelligent transmission decision-making for applications, multi-dimensional collaborative transmission, and customized in-network transmission acceleration, including key insights, basic ideas and principles, and results of large-scale deployment. Finally, the next development trends of Internet transmission technology are prospected.

Key words transmission control protocol (TCP); low-latency transmission; content delivery network (CDN); transmission-computation collaboration; software-defined networking (SDN)

摘 要 互联网是现代社会最重要的基础设施之一,其传输性能直接决定了其承载业务的能力和价值.随着移动互联网和交互式应用等快速发展,大规模互联网低时延传输面临应用层、传输层和网络层3方面技术难题,即应用数据动态发送决策与网络可用资源不匹配的难题、网络传输决策与网络状态变化不协同的难题、差异化业务与网络设备处理逻辑不一致的难题.针对传统分层、单点传输优化技术的瓶颈,提出并构建了面向大规模互联网服务的协同传输技术体系.介绍该体系中的应用智能传输决策、多维度协同传输、定制化网络内传输加速3方面技术成果,包括重要洞见、基本思路与原理、大规模部署结果等.最后,展望了互联网传输技术的下一步发展趋势.

收稿日期: 2025-07-15; **修回日期**: 2025-08-06

基金项目: 国家重点研发计划项目(2022YFB2901800); 国家自然科学基金项目(62472404)

This work was supported by the National Key Research and Development Program of China (2022YFB2901800) and the National Natural Science Foundation of China (62472404).

通信作者: 李振宇(zyli@ict.ac.cn)

关键词 传输控制协议;低时延传输;内容分发网络;传算协同;软件定义网络

中图法分类号 TP393

DOI: 10.7544/issn1000-1239.202550536 **CSTR:** 32373.14.issn1000-1239.202550536

互联网是现代社会最重要的基础设施之一,其最本质的功能是数据传输.互联网传输性能决定互联网承载业务的能力,即互联网价值.随着5G/6G以及卫星互联网等新型移动无线网络的发展,底层网络异构性和动态性不断增加.与此同时,视频直播、云渲染和智能体网络等交互式应用对低时延、大并发传输的需求与日俱增.如何在异构动态的网络之上满足互联网业务的大并发、低时延传输需求,是近年来互联网体系结构领域最重要的研究问题之一.

国内出台了一系列政策支持与推进低时延互联网传输的发展.2022年,中央网络安全和信息化委员会印发的《"十四五"国家信息化规划》强调了低时延传输在新型多媒体等5G创新应用,以及空天地海立体化网络中的重要性和紧迫性.2023年,《工业和信息化部等十一部门关于开展"信号升格"专项行动的通知》中明确要求"实现移动用户端到端业务感知明显提升…卡顿、时延等主要业务指标全面优化",表明低时延传输是移动业务性能保障的基础.

从协议层次上看,应用决策、传输控制和网络转发是影响端到端业务传输时延的3个关键环节.应用数据突发产生、网络状态跳变、异构业务巨量并发请求给低时延传输带来3方面技术挑战:

1)交互式应用等业务的数据产生具有突发性,而底层网络可用带宽等资源受限且动态变化,存在应用动态数据发送决策与网络可用资源不匹配的难题;

2)移动无线网络不断升级,4G、5G、Wi-Fi甚至卫星网络等网络环境并存,传输能力差异大,终端移动、网络切换造成网络带宽跳变和信道容量剧烈抖动,而传统传输协议无法及时准确感知网络状态,存在网络传输控制决策与网络状态感知不协同的难题;

3)网络业务具有不同的流量模式和通信模式,而传统网络设备采用固定数据包处理流程,无法根据差异化业务模式进行定制化处理,存在差异化业务与网络设备处理逻辑不一致的难题.

互联网设计遵从严格的层次结构,该结构作为互联网得以蓬勃发展的基石,极大地促进了上层应用与底层接入网络的解耦与独立发展.然而,在面对低时延等严苛传输需求时,其固有的层次间信息隔离特性,使得难以实现端到端传输的最优决策,进而可能导致数据在各层的冗余缓冲,造成时延增加和抖动.比如,传输层无法感知应用发送数据的策略(突发还是连续),也无法感知网络切换等网络层的状态变化,从而难以做出与应用数据和网络资源相匹配的传输决策.另一方面,传统传输技术的主要控制点在发送端,而发送端只能通过接收端的反馈(如ACK)被动推断网络拥塞状态和接收端应用体验质量(quality of experience, QoE),存在感知不及时和推断不准确的问题,同样会影响交互式应用的性能.发展跨层、跨栈、全网协同的新一代互联网传输技术是解决上述挑战的有效途径之一.

本文从应用、传输和网络3个层次开展研究,提出并构建面向大规模互联网服务的协同传输技术体系,显著降低了传输时延,赋能视频直播等低时延互联网应用.具体创新成果包括应用智能传输决策、多维度协同传输、定制化网络内传输加速等3方面,如图1所示.

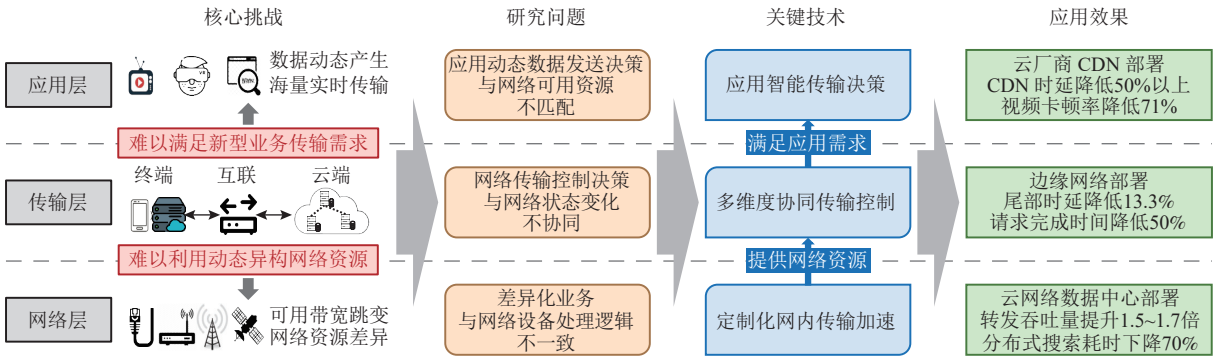


Fig. 1 Research problems and innovative achievements

图1 研究问题与创新成果

1)应用智能传输决策. 针对应用层存在的动态数据发送决策与网络可用资源不匹配的难题，提出了用户行为感知的动态数据发送控制策略，设计了扁平化内容分发网络(content delivery network, CDN)组网架构及其灵活路由转发技术，提出了接收端自适应映射机制，突破了沿用20余年的层次化CDN组网架构，使得千万级并发请求的秒级直播成为可能.

2)多维度协同传输控制. 针对传输层存在的传输控制决策与网络状态感知不协同的难题，提出了多维度协同传输控制的思想，设计了多路径协同传输、跨层协同传输和端网协同传输机制，发展了低时延QUIC(quick UDP Internet connections)传输协议，突破了传统传输控制协议单点被动感知决策的性能瓶颈，实现多路径QUIC和QUIC隧道的大规模部署，为上层应用提供低时延传输通道.

3)定制化网络内传输加速. 针对网络层存在的差异化业务与网络设备处理逻辑不一致的难题，提出了定制化数据包传输加速的思想，设计了多通路数据包转发和传算协同数据包处理技术，提出了网络编程规则的计算理论与正确性验证方法，突破了传统网络设备固定统一处理逻辑的性能瓶颈，推动算网融合新型网络的构建.

最后，本文从应用牵引、协议演进、架构革新与范式变迁等视角展望了低时延传输发展趋势与进一步研究方向.

1 动态应用数据智能传输决策技术

互联网交互式应用的时延由发送端发送时延、CDN传输时延和接收端接收时延3部分组成，如图2所示. 在发送端，数据动态、突发产生(如云游戏、视频直播等应用数据产生与画面变化幅度强相关[1-4])，而网络状态也动态变化，导致突发产生数据的发送时延大且不稳定. 在CDN传输方面，传统层次化CDN网络内数据传输路径固定且长，导致CDN时延大. 而接收端侧通常配置了接收缓冲区来应对内容下载速率与消耗(如观看视频)速率不一致的问题，避免传输过程中出现卡顿. 然而，接收端的缓冲区引入了额外的时延. 上述问题产生的根源在于应用动态数据发送决策与底层网络可用资源不匹配. 为此，提出了发送端动态数据控制方法、扁平化CDN架构和接收端缓冲区自适应控制方法，并在大规模实际系统中部署应用，降低端到端传输时延，并提升用户QoE.

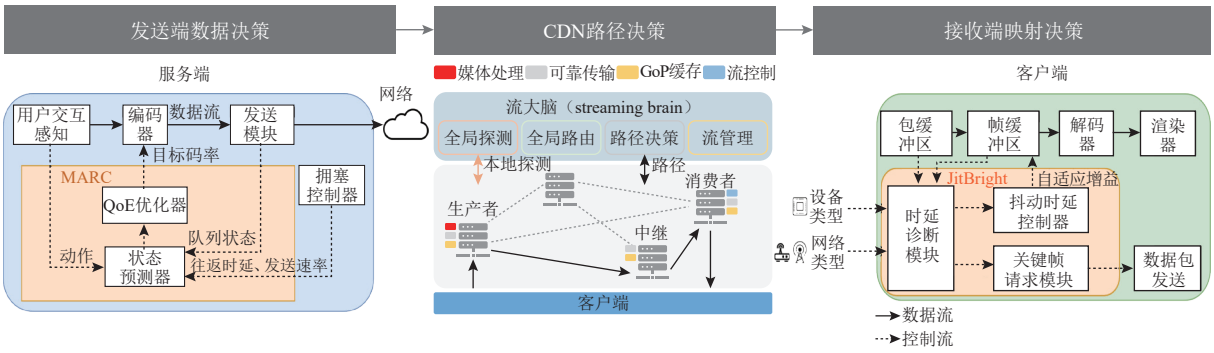


Fig. 2 Intelligent transmission decision-making for dynamic application data

图2 动态应用数据智能传输决策

1.1 动态应用数据控制

交互式应用在发送端进行速率控制，以使得传输数据量动态适应网络可用资源变化. 以视频会议为例，其发送端(服务器)实时决策每个视频帧的编码码率，尽可能使得码率贴近(但不超过)可用带宽上限，从而在降低端到端时延的同时尽可能提高画面质量[5]. 因此，现有研究工作集中在应用层带宽预测以捕获网络资源动态特征[6-9].

然而，在交互式应用中，用户操作(如视频会议中较大的动作等)为发送数据控制引入了额外的不确定性因素. 在此类应用中，服务器根据用户操作实时渲染3D模型，再将其以视频的形式传输至用户设备渲染，利用网络传输能力降低应用的算力需求. 通常来说，用户动作越复杂，对画面的改变会越大，编码所需要消耗的实际码率也就越高[1-4]. 这种用户交互动态性为应用数据控制带来了新的挑战.

我们测量分析了交互式应用的网络传输性能. 基于淘宝的3D云渲染系统，采集了超过100万真实用户的会话数据，分析得到2个重要观察：1)用户的操作呈现出显著的“动—静”交替模式，即在“动作”

阶段(如旋转、缩放画面模型)和“非动作”阶段(如静止观察)之间切换;2)用户的 QoE 偏好会随着其交互行为动态演变,即用户在动作和非动作阶段对于时延和码率的需求存在差异. 动作阶段的时延增加会显著降低用户参与度,即会使得用户提前结束会话.

这里的关键问题在于,当用户处于对时延最敏感的动作状态时,反而是尾时延增长最严重的时候. 这是因为动作阶段的视频帧普遍更大,其数据量平均增加 22%,导致 99 分位的视频帧发送时间突增 1.9 倍. 然而,现有的发送数据控制方法对动作和非动作阶段采用相同的决策逻辑,无法区分这二者迥异的 QoE 需求.

以上述观察为基础,提出了一种感知用户交互的应用数据发送控制框架 MARC(motion-aware rate control)[10]. MARC 将码率决策与用户实时 QoE 偏好对齐,对动作与非动作阶段执行差异化码率分配策略,结合网络与视频信息实现精准地发送控制. 其核心设计包括 2 方面:

1)动态 QoE 建模与量化. 基于真实用户行为数据,量化用户在不同阶段对时延和质量的敏感度差异. 比如,相较于非动作帧,会话时长对动作帧时延的敏感度高出 75.7%. 进一步,构建了动态 QoE 目标函数,基于大规模真实业务数据建模用户在动作与非动作阶段对时延和码率的偏好,从而使优化目标与用户的实际 QoE 需求保持一致.

2)基于预测的细粒度码率控制. 考虑到用户行为的突发性和短暂性,速率控制必须在毫秒级的视频帧粒度上进行. 为此,设计了一种基于随机优化的发送数据控制框架,通过集成状态预测器,实现基于历史数据预测未来短时间窗口内用户行为序列和网络状态. 这种“向前看”机制使得为即将到来的动作帧序列提前规划码率分配成为可能,从而主动降低码率以规避因数据突增而造成的排队时延,而不是在时延发生后被动响应. 上述决策通过随机优化求解器在每个帧的间隔内完成,确保了系统的实时性.

MARC 系统架构如图 2 左侧图所示. 其核心状态预测器模块负责对未来用户行为和网络状况进行预测,以实现对发送数据的动态控制. 该预测模块融合多种信息,包括来自用户交互感知的实时动作数据、来自拥塞控制器[11]的往返时延和发送速率等网络状态,以及发送模块的队列状态等. 这些预测结果随后被送入多步前瞻优化器,基于动态 QoE 目标函数,计算出最优的未来码率序列,并将其中的第 1 步作为当前帧的目标码率输出给编码器.

线上大规模 A/B 测试结果表明,与现有最优方案相比,在服务器端 CPU 额外开销 1.3% 的情况下,MARC 将会话卡顿率降低了 71%,动作帧的尾部时延减少了 30%~55%. 时延改善直接转化为用户参与度的提升:平均会话时长增加了 9%,用户交互时间占比提升了 20%.

MARC 的成功部署说明了系统设计与真实用户 QoE 需求对齐的必要性. 目前,MARC 仅关注用户在动作和非动作状态下的 QoE 偏好差异. 而在以扩展现实(extended reality, XR)为代表的沉浸式应用中,用户的交互动作更加复杂. 因此,基于用户交互模式的发送控制可能成为未来重要的研究方向之一. 例如,可以借鉴计算机体系结构中的分支预测技术,通过预测用户的动作趋势为后续多帧进行预编码. 在网络稳定的环境下,若用户的实际动作与预测相符则立即进行传输,以此降低发送端的编码及发送决策等的时延,并保障连续多帧的最优码率决策.

1.2 扁平化 CDN 架构及其路由转发机制

互联网应用依赖 CDN 把数据分发给大规模在线用户. 传统 CDN 网络为了最大化缓存效用,采用层次化组网架构,应用产生的数据会被传输到集中式系统进行媒体处理,然后通过应用层组播[12]和缓存[13-15]等技术分发到连接接收端的边缘节点. 然而,交互式应用数据动态产生,缓存作用变小. 更为重要的是,层次化架构导致数据 2 次经过层次网络(源→CDN 集中处理中心→目的),数据转发时延长. 通过大规模真实测量结果发现,层次化组网架构不能满足视频直播等交互式应用中端到端时延中值小于 1 秒的要求. 而其中最重要的时延组成是 CDN 内部的数据转发时延,而层次化组网架构是造成 CDN 时延大的根本原因.

随着 CDN 逐渐在公共或者私有云构建,其节点不仅具备存储转发功能,而且有一定的计算能力. 因此,借鉴软件定义网络(software-defined networking, SDN)思想,设计了基于扁平化 CDN 的低时延视频传输网络 LiveNet[16]. LiveNet 把 CDN 分为控制面和扁平化数据面,如图 2 中间图所示. 数据面承担视频数据的所有处理操作,包括转码等计算操作和缓存转发等传输操作. 每个节点既可以是生产节点(接收和处理主播的流),也可以是消费节点(向客户端转发视频流并对流实施精细控制)或中继节点(在传输路径中连接消费者和生产者,提供转发和缓存等服务). 通过将 CDN 节点角色与功能解耦,不仅实现了动态转发路径的灵活性,还能均衡地分配负载,避免了中

心热点问题.

控制面(即流大脑)收集 CDN 网络的实时信息，并据此为每一对生产节点和消费节点计算全局最优路径. 当一个观看请求到来时，路径决策模块为其选择性能最优的转发路径，把数据从生产节点转发到客户端连接的消费节点. 通过逻辑上的集中化管理，LiveNet 可以灵活部署新的选路算法，绕过故障节点或实现应用定制的策略.

尽管扁平化架构具有更好的灵活性，但其面临 2 方面技术挑战. 一是针对每个请求计算最优路径的复杂度高、时间长，影响用户首帧时间. 二是为了实现丢帧快速重传和缩短首帧时间，中继节点需要缓存近期转发的视频帧；同时在逐跳转发下，中继节点需要完成类似 WebRTC 的拥塞控制机制[11]. 这就要求中继节点在应用层处理数据，数据包在中继节点停留时间长，增加了节点处理时延.

针对第 1 个挑战，LiveNet 采用“离线算路、在线选路”的可扩展路径选择方法，即离线计算任意〈生产节点，消费节点〉之间的最优路径，并在线获取和更新路径信息. 该方法尽管选择的路径不一定是实时最优的，但可以在传输时延和首帧时间之间取得更好的平衡. 进一步设计路径预取、路径缓存、快速查表等方法，减少每个请求转发路径获取时间，实现大规模并发请求的可扩展实时选路. 针对第 2 个挑战，我们发现 CDN 内部的丢包率非常小(约千分之一)，中继节点没有必要为如此低的丢包率进行复杂操作. 为此，设计了快路径和慢路径结合的数据转发机制，解耦合慢路径(可靠传输)与快路径(数据包快速转发). 快路径收到数据包后立即转发，实现快速转发；而慢路径的数据包副本用来实现拥塞控制和图像组(group of pictures, GoP)缓存等功能.

大规模部署测试结果表明，与经典层次化 CDN 相比，LiveNet 将 CDN 路径平均长度从 4 跳压缩到 2 跳，并将 CDN 传输时延减少了 50% 以上，端到端时延中值降为 0.95 s，满足秒级直播需要. 在用户 QoE 方面，95% 的观看的启动时延在 1 s 以内，98% 的观看零卡顿. LiveNet 突破了沿用 20 余年的层次化 CDN 组网架构，其设计选择和部署经验可以广泛应用于视频会议、在线教育等其他大规模直播场景.

随着流量的激增，CDN 还面临着严重的带宽成本问题. 大型 CDN 提供商逐渐探索如何利用距离用户更近、成本更低的边缘节点，降低成本和时延. 但是，此类节点的带宽、计算存储能力有限，而且不稳定[17]. 如何充分利用不稳定边缘节点提供低时延、低成本和可靠的传输服务，是 CDN 领域未来研究的重点问题之一.

1.3 接收端自适应缓冲区管理

应用数据通常在发送端以固定速率产生，再到接收端以固定速率消耗. 然而，由于传输网络的动态性，数据到达接收端的速率并不稳定. 若数据到达速率低于消耗速率，则会造成卡顿. 为此，接收端设置缓冲区来存储部分应用数据，确保传输的流畅性，其代价是在端到端传输流程中引入了额外的时延. 我们在实际云渲染系统中观察到客户端侧接收到显示过程 R2C(receive-to-composition)在大多数情况下是端到端时延中占比最高的部分(57.2%)，而其中绝大部分(71.5%)源于视频帧在客户端接收缓冲区中的等待时间. 这种等待可归结为两大根本原因：主动等待和被动等待. 主动等待是指为确保平滑播放，已完整接收的视频帧仍需在缓冲区中排队，等待对应的播放时刻到来. 被动等待是指即便一个视频帧的数据已经接收完整，但由于网络丢包等因素，其前序的解码依赖帧仍未完成传输，则该帧无法按照原有时间戳播放，形成“队首(head-of-line, HoL)阻塞”.

导致 R2C 时延增长的根源在于，当前广泛应用的默认接收缓冲区管理策略(如 WebRTC[11] 等)过于保守. 该策略为了应对偶然出现的、尺寸巨大的关键帧(I 帧)，倾向于维持一个过大的缓冲区. 然而，在交互式应用场景下，关键帧非常稀疏(通常每 300 帧 1 个). 现有缓冲区控制策略会引入大量不必要的主动等待时间，从而显著增加整体时延.

提出了一种自适应接收缓冲区管理方法 JitBright[18-19]，在确保播放平滑度的前提下，最大限度降低用户操作到屏幕显示更新 MTP(motion-to-photon)的时延. JitBright 引入“自适应增益”和“主动关键帧请求”2 种创新机制，分别应对主动等待和被动等待的挑战.

JitBright 主要组成部分如图 2 右侧图所示，其核心是时延诊断模块，它接收来自网络和解码器的状态信息，并基于这些信息驱动 2 个关键的决策控制器：抖动时延控制器和关键帧请求模块. 抖动时延控制器通过自适应增益算法，结合网络环境与应用数据信息动态调整接收缓冲区的目标大小；关键帧请求模块则在必要时向服务端主动请求新的关键帧以打破解码依赖，并将决策结果反馈给数据包发送模块，从而形成一个闭环控制系统.

1)自适应增益. 理想的缓冲策略应在极低时延与播放平滑度之间取得动态平衡，因此缓冲区的目

标时长应与视频帧无法被按时播放(发生卡顿)的概率成正比. 为此, JitBright 利用切比雪夫不等式估算帧播放可能发生抖动的概率上界, 并根据已接收视频帧大小的方差与网络带宽估计值等信息, 动态调整缓冲目标大小. 当网络稳定、帧大小均匀时, 抖动概率低, JitBright 会将缓冲区维持在极低水平, 从而减少主动等待; 反之, 当网络波动或帧大小变化剧烈时, 则增加缓冲以保障播放平滑度.

2)主动关键帧请求. 当因丢包导致被动等待时, 传统做法是等待网络重传丢失的参考帧, 但这可能引入较长时延. 另一种做法是丢掉阻塞解码的视频帧并主动请求新的可以独立解码的关键帧, 以此打破解码依赖. 然而, 请求关键帧也有成本, 因为关键帧比非关键帧尺寸更大(4~10 倍)、传输耗时更长, 且丢弃已缓冲的视频帧会影响观看平滑度. 为做出最优决策, JitBright 引入基于成本效益分析的决策机制. 该机制评估等待成本和请求成本, 前者量化继续等待参考帧重传所付出的时间代价, 后者量化立即请求新关键帧所需付出的传输、解码和丢帧平滑度惩罚等综合代价. 每当被动等待发生时, 系统会实时比较等待成本和请求成本, 执行成本更低的方案, 最小化用户体验的损失.

在覆盖超过 12 000 名志愿用户、累计视频时长 314 h 的大规模 A/B 测试结果表明, JitBright 在 Wi-Fi、5G 和 4G 网络下, 将满足 MTP 时延小于 150 ms 的会话比例分别提升了 15%~23%、9%~20% 和 6%~27%. 视频整体卡顿率从 2.4%~2.8% 大幅降低到 0.4%~1.0%.

JitBright 的缓冲区控制以网络动态和视频帧动态作为指导, 可以在其建模中进一步结合媒体内容类别(如动作游戏或剧情电影)和用户偏好(如低时延优先或流畅性优先)信息, 从而实现对更多样化应用类型的适配, 满足真实用户的差异化 QoE 需求. 此外, JitBright 使用了相对较为简单的建模, 未来可以使用更为准确的统计模型进行优化, 或结合数据驱动技术提升控制精度.

2 多维度协同高效传输技术

传输控制协议建立在底层网络之上感知网络状态, 并据此决策数据包发送策略. 在移动互联网和卫星互联网等新型网络环境中, 网络状态跳变. 传统传输控制协议借助 ACK Clocking 实现闭环控制, 其本质是发送端单点、单层被动感知网络状态, 在动态网络中不能实时准确感知网络状态, 导致传输控制决策与网络状态感知的不协同, 无法为上层业务提供低时延传输通道.

提升发送端感知实时性最直接的方法是缩短拥塞感知点(传统为发送端)与接收端之间的距离, 从而减小 ACK 反馈的时延. 这里的挑战是, 如何在保持发送端与接收端之间端到端连接的情况下, 由距离接收端更近的边缘节点执行拥塞感知, 并反馈拥塞信息. 提升传输性能的另一个手段是利用多个物理链路带宽资源, 然而这会引入路径之间紧耦合, 导致快路径被拖慢的问题. 最后, 不同应用的传输需求千差万别, 而现有传输机制通常被设计为“以不变应万变”, 这种僵化的固定分层设计很难为应用提供充分的性能保障. 上述挑战的本质在于发送端单点、单层被动感知网络状态, 缺乏协同感知与控制.

在 UDP 之上实现的 QUIC 使得传输控制的协同成为可能. 提出多维度协同传输控制的思想, 从端网协同、多路径协同和跨层协同 3 个层次提升传输的性能, 如图 3 所示, 多维度协同高级传输技术突破了

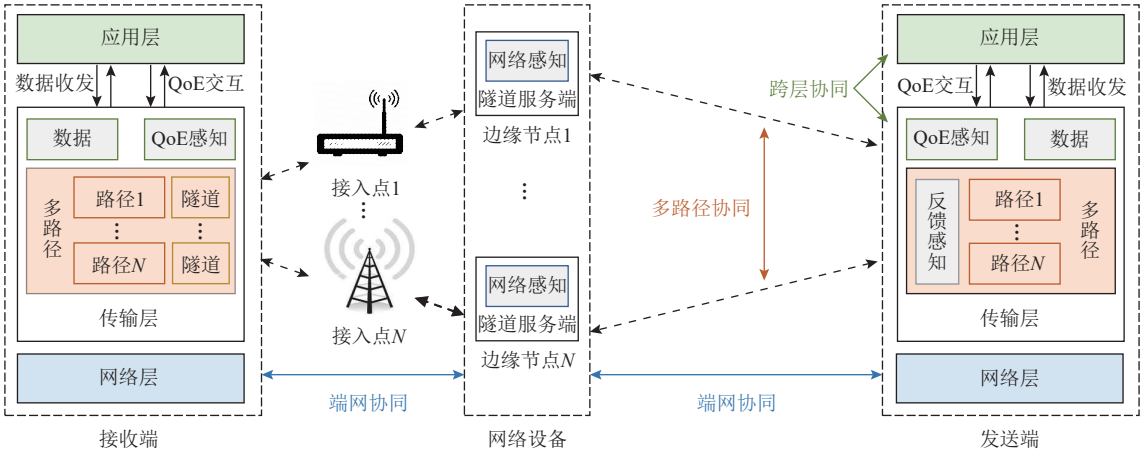


Fig. 3 Multidimensional, cooperative, and efficient transmission technology

图 3 多维度协同高效传输技术

传输控制决策与状态感知不协同的难题，为应用提供低时延、高吞吐传输性能.

2.1 基于 QUIC 隧道的端网协同传输

将服务器—客户端端到端控制环路缩短为边缘节点与接收端之间的环路，是增强对网络变化感知能力、降低传输时延的最有效手段之一. 为实现这一目标，IETF MASQUE 工作组提出了在用户与边缘节点之间建立 QUIC 隧道[20-21]，在不改变原有应用连接的前提下，由边缘节点感知拥塞状态并执行部分拥塞控制功能. 然而，我们大规模实际部署后发现，MASQUE 隧道的性能未能达到预期，本质原因在于：

1）独立的丢包恢复. 虽然隧道中的重传对于加速丢包恢复至关重要，但它可能导致隧道服务器和应用服务器的重复重传，造成带宽浪费并加剧网络拥塞；

2）嵌套的拥塞控制. 隧道中的拥塞控制有助于降低重传速率并通过限制数据包速率来缓解网络拥塞，但同时也可能导致内外连接速率的不匹配，从而降低端到端吞吐量.

可以看出，协同隧道连接和应用连接的丢包恢复和拥塞控制至关重要. 具体而言，隧道连接与端到端应用连接的状态应相互配合：隧道重传的数据包不应被应用服务器再次重传，且应用服务器的拥塞控制状态应根据隧道节点感知的更准确状态进行调整.

基于上述观察，提出了一种基于 QUIC 隧道的端网协同传输方法 TECC[22]，如图 4 所示. 其核心思想是将原本由服务器端完成的拥塞控制和重传功能卸载至靠近客户端的边缘隧道节点，从而缩短拥塞控制决策单元与接收端之间的物理距离，提升响应速度. 通过实时监测网络状态，隧道节点能够精确感知丢包和拥塞信息，并将这些反馈通过“隧道服务器→客户端→服务器”的路径迅速传递至服务器端. 借助隧道节点提供的精准反馈，服务器调整其重传和拥塞控制策略，确保服务器端的网络状态感知与隧道节点保持一致. 通过紧密协同，TECC 突破了制约 IETF MASQUE 的 2 个问题.

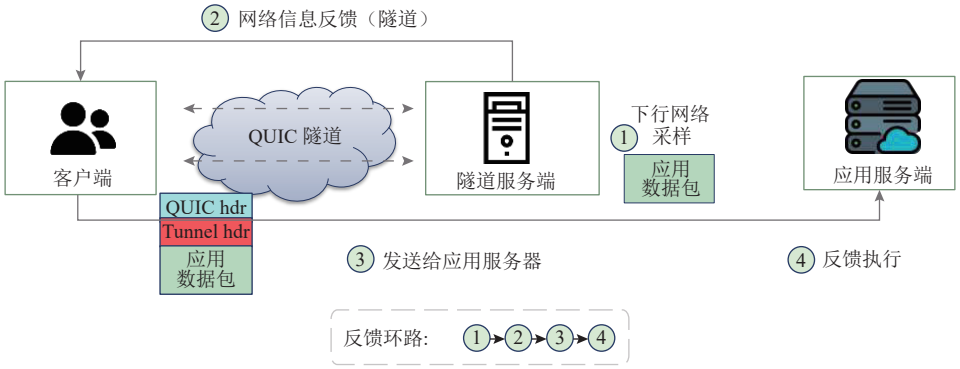


Fig. 4 System architecture of TECC

图 4 TECC 系统架构

在某头部移动应用的大规模测试结果表明，与传统 MASQUE 隧道协议相比，TECC 显著降低了传输时延，例如远程过程调用（remote procedure call, RPC）请求的尾部时延则降低了 13.3%. 部署 TECC 后，即使不使用网络运营商昂贵的服务质量（quality of service, QoS）保障机制，依然能够满足业务传输性能需求，从而大幅减少了运营成本（减少 70%）.

在上述端网协同传输模式中，发送端仍然是被动感知网络状态，其发送策略调整始终滞后于网络状态的变化，是制约端网协同传输机制进一步降低时延的根本原因. IETF 工作组正在推动瓶颈网络设备与应用传输控制点的协同标准制定，以进一步增强应用感知网络的能力，为超低时延高带宽业务的大规模应用提供支持. 实现上述协同的一种形式是网络内设备使用类似显式拥塞通知（explicit congestion notification, ECN）[23] 的手段主动返回拥塞信息，发送端融合多种信息完成更精准和高效的数据发送.

随着 5G/B5G 网络逐步开放，在广域网中网络内设备返回拥塞信号将成为可能，是未来重要的研究方向. 信号的反馈形式可以灵活多样，可通过 ECN 中由接收端通过传输层协议间接反馈，也可通过网络设备直接向发送端返回信息. 反馈信息也可从简单的二进制拥塞标记细化为精确的设备队列深度等[24]. 具体实现而言，可利用如 QUIC 协议的自定义帧机制灵活传输反馈信息，实现网络层与传输层的协同，以更精准地指导发送端数据发送.

2.2 QoE 驱动的多路径协同传输

智能手机等移动设备不断迭代更新，其网络通

信能力也持续演进,使得应用同时利用多个不同网卡在多条不同物理链路上(如 5G 和 Wi-Fi)传输数据成为可能. 加之新兴应用对于带宽的需求与日俱增(几十至上百 Mbps[25-27]),多路径传输由于其更大的聚合带宽和鲁棒性,受到广泛关注. 然而,2013 年完成 RFC 标准化的多路径 TCP(multipath TCP, MPTCP)[28]在互联网中的部署仍然极为有限. 其根本原因在于:数据包调度器严重依赖对路径质量(如带宽、RTT 等)的准确预估来决策数据包发送策略. 但问题在于,当多个路径之间质量差距较大且抖动频繁时,发送端难以及时感知并准确预估路径状态变化,容易在慢路径(如长 RTT 路径)上分配过多的数据,导致快路径(如短 RTT 路径)的可用带宽被浪费,从而在接收端造成队首阻塞[29],使得多路径传输性能甚至可能低于单条路径传输性能. 我们进行的大规模的测试也验证了该结论:MPTCP 视频块下载时间的中位数和 99% 分位数比单路径传输分别慢了 16% 和 28%.

已有工作的思路是引入跨路径重传(又称"重注入"),即把分配给某个路径的数据内容在其他路径上进行冗余发送,以此快速解决队首阻塞问题. 然而,重注入引入了额外的冗余带宽开销,直接增加了应用服务提供商向 CDN 或运营商支付的带宽费用. 我们发现,即使是最先进的数据包调度器,若不对重注入进行限制,应用带宽用量会增加 15% 以上,是不可接受的.

事实上,传输层队首阻塞并不一定导致 QoE 的下降,因为数据只要能在应用层需要该数据之前被提交到应用层即可. 因此,是否开启跨路径重传来解决队首阻塞问题可以利用用户的 QoE 信息来指导. 比如,当应用层视频播放器的缓冲数据较多时,并不需要开启跨路径重传;反之,发送端需要尽快开启跨路径重传. 基于上述思想,提出了一种 QoE 驱动的多路径协同传输方法 XLINK[30]. XLINK 基于多路径 QUIC(multipath QUIC, MPQUIC)实现,利用了 QUIC 协议的用户态特性,相较于操作系统内核之中的 MPTCP 更方便应用程序进行集成.

XLINK 整体架构如图 5 所示,在架构层面上,XLINK 中的 QoE 驱动的多路径调度建立在 QoE 反馈机制之上. XLINK 客户端获取用户感知的 QoE 信号(如视频播放器的缓冲区水平),然后使用 QUIC 协议中的多路径数据确认(ACK_MP)扩展帧将 QoE 信号封装,进而反馈到发送端作为路径管理和数据包调度的依据.

在算法层面,XLINK 利用数据包重注入来解耦多个路径. 与已有重注入策略[29]不同,XLINK 分别在传输(QUIC 流)和应用(视频帧)2 个层级上实现了基于优先级的重注入. 基于流(Stream)优先级的重注入对应应用层请求的多个 QUIC 并发流,确保先发送的流的重注入包不被后续的流的数据包阻塞;基于应用优先级的重注入则会区分(同一个流内)多个不同数据块的传输紧迫性. 比如,将视频首帧视作最高优先级以加速视频启动. XLINK 还引入了基于 QoE 反馈的双阈值重注入控制,根据 QoE 需求平衡传输性能与冗余成本. 例如,对于短视频应用而言,XLINK 服务器会根据客户端的播放缓冲区水平动态调整重注入策略,以最小的带宽流量开销避免卡顿.

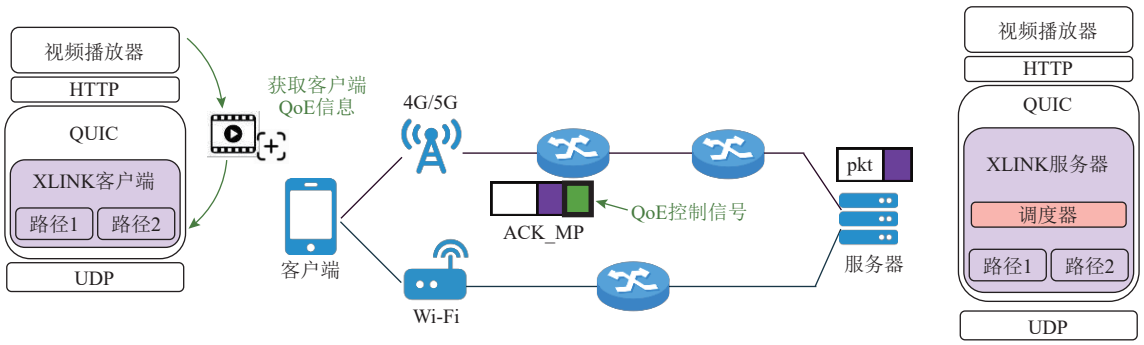


Fig. 5 End-to-end architecture of XLINK

图 5 XLINK 端到端架构

XLINK 在淘宝短视频传输中部署测试,结果表明,与单路径 QUIC 相比,第 99 分位视频块请求完成时间下降了 19%~50%,卡顿率降低了 23%~67%,而冗余流量仅 2.1%.

目前,大规模部署多路径传输的收益主要得益于提升弱网环境下的传输鲁棒性和性能[31-33],因此在卫星网络中有广阔的应用前景. 随着增强现实/混合现实(augmented reality/virtual reality, AR/VR)的发展,多

路径传输的带宽聚合收益有望逐步放大. 然而, AR/VR 对传输时延也有苛刻的要求, 这对多路之间的协同提出更高要求, 也是可进一步研究的重点问题之一.

2.3 基于双向反馈控制环路的跨层协同传输

互联网分层设计使得各层可以独立快速演进, 从而确保了良好的可扩展性. 在此前提下, 传输层关注通用传输策略, 旨在最大程度上优化 QoS 指标, 例如提升吞吐量或缩短下载时间. 然而, 这种分层设计也逐渐暴露出了其固有缺陷: 传输层完全不感知应用需求, 很难为差异化应用提供恰到好处的传输性能. 由于应用通常有独立于传输层的发送控制逻辑, 哪怕是传输层提供了其自认为最优的性能, 也可能会对应用控制逻辑造成干扰, 从而导致 QoE 下降.

以多路径传输为例, 现有工作集中于优化数据包调度器以最大化传输性能[34-36]. 然而, 我们在大规模自适应码率(adaptive bitrate, ABR)视频的多路径传输评估表明, 相对于单路径视频流而言, 即使多路径视频流具有更高的吞吐量, 仍会取得更差的 QoE(码率下降 11.7%, 卡顿时间增加 6.2%), 尤其是在高动态移动网络环境中. 这种 QoE 下降是因为多路径视频流会更频繁地触发卡顿事件, 而这种卡顿往往是由错误(过高)的码率选择导致, 并非传输性能不足. 通过进一步分析发现, 多路径调度机制会为应用的吞吐量预测带来额外的不确定性, 使得应用更容易误判网络的实际可用资源. 在高波动网络中, 多路径调度决策的频繁改变使应用倾向于发送过多的数据. 尽管传输层可以通过数据包或 RTT 级别的细粒度反馈信息对其决策做出迅速调整, 但应用的决策粒度相对更粗(例如视频块码率为秒级决策), 很难及时改变. 当应用数据量远超网络带宽时, 就会产生严重的卡顿.

本质上, 应用层无法感知多路径调度对吞吐量的影响, 难以准确预测多路径环境下的可用网络资源. 因此, 即便是传输层提供了更高的传输性能, 应用也无法充分利用. 针对上述问题, 我们的解决思路是将应用与传输进行跨层协同, 并提出了一种多路径调度与应用数据控制的协同框架 Chorus[37], 满足优化 QoE 的 2 个必要条件: 1)确保最优应用数据控制决策; 2)提供满足 QoE 需求的传输性能.

Chorus 建立了如图 6 所示的双向反馈控制回路. 在 MPQUIC 之上, Chorus 允许服务器和客户端之间进行跨层信息传递, 使得双端跨层可以共同优化 QoE. 在第 1 个控制环路中, 客户端的应用数据控制算法根据服务器预先确定的调度决策(图 6 中的“路径分配比例”)预测吞吐量, 并进行数据请求与发送. 在第 2 个控制环路中, 服务器的数据包调度器在应用数据传输过程中及时调整调度决策, 以满足应用的预期传输时间(图 6 中的预期传输时间). 这 2 个控制环路的决策粒度不同, 前者与应用决策周期(如视频块级别)保持一致, 后者则作用于数据包或 RTT 级别.

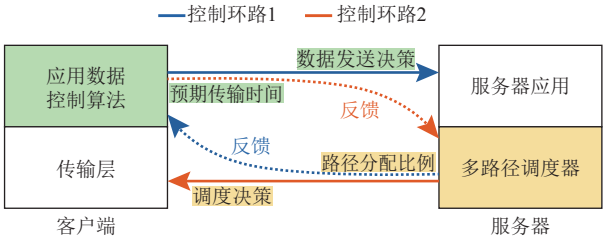


Fig. 6 Bidirectional feedback control loops in Chorus

图 6 Chorus 中的双向反馈控制环路

为了实现双端跨层协同, Chorus 引入了融合粗粒度决策与细粒度矫正的策略. 粗粒度决策阶段发生在传输应用数据之前, 而细粒度矫正阶段则作用于数据传输期间. 在粗粒度决策阶段, 首先预先确定所请求数据块中所有数据包的调度决策, 然后基于该决策预测吞吐量并确定发送的数据量. 一旦应用数据传输开始, 就进入细粒度矫正阶段, 并通过少量重调度和重注入来矫正先前的一次性调度(如果该调度决策非最优的话). 值得一提的是, Chorus 的传输层并非追求最优传输性能, 而是以应用预期的传输时间为指导, 一来减少了不必要的冗余重传, 二来也可以限制传输机制对应用发送控制的干扰.

Chorus 已集成至真实移动视频流系统中, 包括网页服务器和移动客户端上的视频播放器应用. 实验结果表明, 与 XLINK[30] 和单路径 QUIC 相比, Chorus 的平均 QoE 提高了 65.7%~114.4%. Chorus 在取得上述 QoE 性能改进的同时, 并没有带来传输成本的提升, 反而降低了 10.7% 的重注入比例. 在设计与评估 Chorus 的过程中, 我们发现客户端的预测器在不牺牲准确性的前提下最好适当保守(即预测值略低于真实值), 这有助于在高波动网络下避免严重卡顿. 据此, 我们设计了实时通信业务的带宽预测方法, 在微软的 Teams 会议应用部署中取得了稳定的 QoE 提升[38].

目前, Chorus 使用简单客户端吞吐量预测器, 其基于多路径的接收速率平滑值与路径分配比例直接计算未来的吞吐量估计值, 该预测器可以通过既有数据驱动技术(例如监督学习方法)进一步优化准确性. 此外, 随着应用的丰富以及低层网络的异构性逐渐加强, 跨层设计优化应用 QoE(而非优化传输性能)

的优势将更加明显.

3 定制化网络内传输加速技术

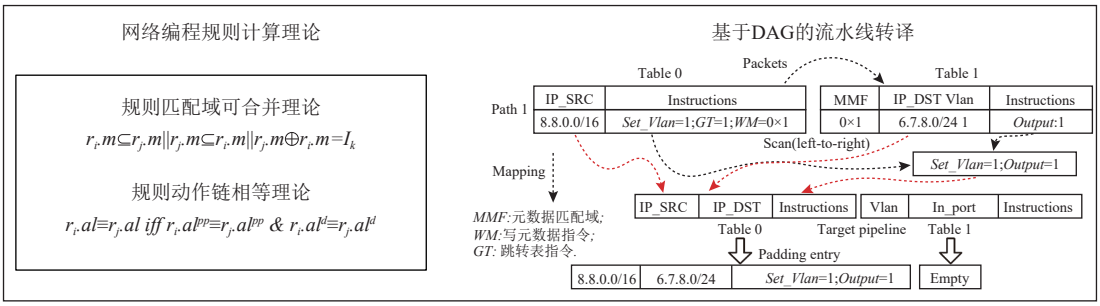
网络内传输加速通过加快网络设备处理数据的速度,或者减少数据包传输数据量,减少端到端的传输时延. 这种思路是在网络转发处理层面提升数据传输效率和性能的最直接、最有效的手段之一. 智能网卡(smart network interface card, SmartNIC)、可编程交换芯片和可编程数据平面等新型硬件技术的不断发展与普及,为实现高效、灵活的网络内传输加速技术提供了可能. 然而,传统网络设备通常采用固定统一的数据包处理流程,功能单一且缺乏灵活性,主要面向基本的转发任务,如基于IP地址的查找与匹配. 因此,绝大部分研究聚焦IP查找等基础功能的性能提升,不能有效支持多样化、差异化的业务处理功能,制约了网络设备在复杂场景下的应用潜力. 传统横向通用的数据面优化思路发展遇到瓶颈,从"通用"到定制化的"专用"已成为支撑差异化网络业务性能的必要手段.

我们分别从控制面与数据面创新,实现网络业务定制化的传输加速,突破了多通路数据包转发、传算协同的传输加速以及网络规则计算等关键技术,主要工作如图7所示.

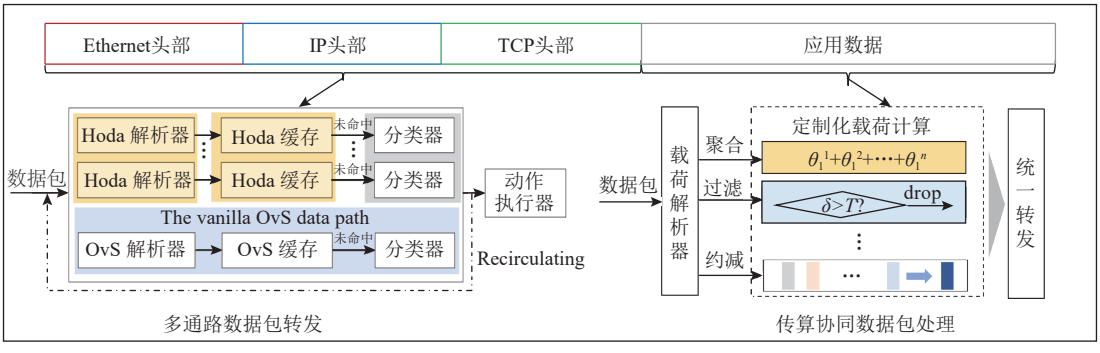
3.1 多通路数据包转发

数据包转发性能决定网络转发时延,是端到端时延的重要组成. 软件定义网络分离数据面和控制面,使得网络内数据包处理灵活、可定制,但同时也使得转发规则呈现多表、多域以及高动态的特征,查找转发耗时大. 采用多级缓存是提升查找效率的有效手段. 比如,Open vSwitch(OvS)[39]基于OpenFlow[40]提供通用编程模型,在数据面采用包括签名匹配缓存(signature match cache, SMC)和巨流缓存(Megaflow cache, MFC)等多级缓存. 数据包到达后,经过可编程解析器和多级缓存,确定所需执行的操作(如转发、丢弃等).

传统OvS数据平面基于"匹配—动作"抽象构建,为了保证灵活性,数据面采用统一数据路径,即任何一个数据包都需要经过解析器的完整解析以及缓存所有网络功能规则的多级缓存结构. 由于MFC匹配的域更多、耗时更长,大量的研究关注更高效的数据包分类算法[41-42]用于提升MFC查找效率. 然而,我们在实际云网络观察到,由于SMC的缓存容量扩大至100万,大多数情况(>99%)下数据包会命中SMC缓存,很少进入MFC匹配阶段. 更为重要的是,当OvS加载复杂网络功能(如隧道、状态防火墙)时,其性能显著下降. 例如,与简单的L2转发相比,同时运行隧道和防火墙功能会导致吞吐量下降高达50%.



(a) 控制平面



(b) 数据平面

Fig. 7 Customized in-network transmission acceleration technology

图7 定制化网络内传输加速技术

分析每个阶段的耗时发现，数据包解析和 SMC 缓存查找这 2 个长期被性能优化忽略的模块，在处理复杂网络功能数据包时耗时明显增加. 根本原因在于 OvS 中统一数据路径与异构网络功能之间的矛盾，导致数据包解析时存在不必要的解析操作，缓存查找时冗余匹配域也会产生无效匹配开销. 受此启发，我们设想为不同网络功能生成专用数据通路. 具体而言，鉴于同一网络功能（即一个 OpenFlow 程序）对应的处理规则具有相似结构（如相似的匹配谓词）并服务于同一目的，可以为每个网络功能构建专门的解析器和 SMC 缓存，仅关注该功能所需的匹配字段，消除不必要操作和冗余操作.

基于上述思想，我们设计并实现了定制化多通路数据包转发框架 Hoda[43]（图 7（b）左下图），可针对不同网络功能构建定制化的数据包解析器与缓存系统，生成定制化的数据包转发通路，以实现轻量级的数据包解析与缓存查找，从而提升数据包处理性能. 此外，为了降低定制化开销，进一步提出了可配置版本 C-Hoda，通过在数据路径中引入可编程接口，以轻微性能下降为代价实现快速数据面定制化（即几乎零代码）. 在实际云网络的验证结果表明，相比于最新 OvS，数据包转发吞吐提升 1.5~1.7 倍，Nginx 服务端到端请求处理时间降低 20%.

随着智能网卡的发展，硬件卸载成为进一步提升数据包转发性能的重要手段. 如何保证多通路数据包处理逻辑中的软硬件状态一致性是硬件卸载的关键. 一方面，频繁更新处理规则，分析规则的分布和更新特征，从而在保证规则一致性的前提下，尽量减少数据面规则的更新. 另一方面，网络功能的某些状态需要在软硬件共享（如数据包计数），如何保持这些状态的一致是另一难题.

3.2 传算协同的网络内数据包处理

近年来，网络设备向传输与计算融合发展，传输和计算融合成为可能. 传算融合在分布式深度学习训练、分布式最近邻搜索等多对少传输场景中应用潜力巨大. 然而，传统数据包处理逻辑只根据数据包头部信息转发和处理数据包，并不对数据包载荷进行处理，无法发挥新型网络设备的计算潜力. 传算协同的网络内数据包处理借助网络设备所处的中间位置，卸载服务器侧的部分或全部处理功能，对数据包载荷进行高效处理，达到减少数据包传输量和系统时延的目的.

数据包载荷计算需要理解载荷的格式和语义，因此也是针对应用定制化的. 为了减少定制化的实现复杂度，我们首先总结并实现了常用的网络内计算原语，包括聚合（aggregate）、过滤（filter）、约减（reduce）和排序等（sort）等. 在此基础上，针对 2 种典型分布式应用（分布式搜索和分布式训练），设计了传算协同的数据包处理机制.

针对分布式搜索，设计并实现了基于在线答案约减的搜索系统[44-45]，通过在可编程交换机上自适应设置阈值，过滤携带较差搜索答案的数据包，同时进行多答案数据包网内聚合，以提升整体搜索效率（如图 7（b）右下图所示）. 实验结果显示，与同期主流分布式搜索系统相比，文本、视频等多类型数据的搜索耗时减少 70%，数据传输量减小至 1/4~1/10. 针对分布式 AI 训练，设计并实现了冷热参数感知的在网梯度聚合机制[46]，在线识别稀疏模型的“热”参数（更新频繁的参数），并在存储受限的数据面完成对“热”参数的网络内梯度聚合，而“冷”参数（更新不频繁的参数）则直接转发给服务器聚合. 由于“热”参数占据了大部分的流量和更新操作，冷热参数分离处理可在有限的存算资源约束下，降低网络传输负载，提升分布式训练效率. 基于实际稀疏模型的实验结果表明，与同期主流网内聚合机制相比，冷热参数感知机制可把分布式稀疏深度模型训练效率提升 1.4~2.6 倍.

随着网络设备逐步开放，网络内数据包处理在实际网络中的大规模部署将成为可能，但仍然面临巨大挑战. 首先，网络内设备存算能力有限，需要扩展存算能力和在有限资源限制下实现数据包处理. 其次，网络内数据包处理需要感知应用语义，这就要求端网定制化处理以及需要发展统一的协议或者标准避免定制化导致的碎片化问题. 上述挑战的有效解决也将促进算网融合的发展.

3.3 网络规则计算理论与方法

可编程数据包处理机制依赖于控制面的组合编程能力，通过对处理规则的灵活组合，以满足多样化业务场景对网络功能的差异化需求. 然而，组合编程极易引入规则配置错误，尤其是在处理规则存在重叠（如一条规则覆盖另一条规则）、冲突（如同一数据包命中 2 条处理动作不同的规则）或冗余的情况，影响系统稳定性和正确性. 同时，规则的语义和结构在异构网络芯片之间存在不一致性，导致部署过程中的适配难题，进一步制约了定制化网络功能的应用.

判断规则配置错误需要把规则映射到同一表示空间. 首先引入代数运算方法[47]，将原本采用 0/1/*形式表示的网络策略规则转化为一组在规则空间中的线段集合，构建从逻辑规则到几何空间的映射模型.

在此基础上,提出了规则匹配域可合并理论和规则动作链等价理论,并设计了规则可用性原则与最简化表达原则,用于识别和剔除冲突、冗余或无效的规则组合,确保生成规则集的功能完整性与表达最优性.进一步提出了基于同步位移的快速检测算法,通过对规则空间中的位置与变化趋势进行并行检测,提升规则组合计算的处理效率与配置准确性,从根本上保障了组合编程在逻辑层面的语义一致性与在运行时的安全可靠性.

为了支撑异构硬件平台上的规则表示与执行机制[48],建立了规则等价转换的基础理论,将多级流水线抽象成有向无环图,并基于图的顺序遍历算法,实现规则在不同平台间的等价映射.为了支持动态更新,提出了面向等价类的有向边计算方法和基于代数式的优先级计算方法.通过等价映射机制,解决了定制化规则在多样硬件平台间迁移部署困难的问题.

可编程成为网络的要素之一,多用户编程与配置易出错的问题将日益突出.为此,可以在用户意图层面引入大语言模型等检测配置错误,也可以在数据面规则级别设计错误检测机制.这些问题的有效解决有助于提升网络空间的安全性和可靠性,是重要的研究方向之一.

4 展 望

聚焦大规模互联网服务的低时延传输技术,结合当前技术演进与产业实践,从以下4个方面对互联网传输体系的未来趋势进行展望:

1)应用牵引,即大模型低时延推理.以大语言模型为代表的AI技术正以前所未有的深度和广度赋能各类应用,其推理阶段的实时性与交互性对网络时延提出了极致要求.为确保AI应用的低时延需求,网络协议与底层架构也需要进行针对性演进,核心在于构建为AI应用深度定制的网络架构.这不仅包括采用RDMA等低时延网络协议,更关键的是实现计算与网络的协同,模型优化(量化、键值对缓存、推测解码等)与网络优化(智能路由、解耦服务、专用互联等)的协同[49-53],确保“毫秒级”的低时延响应,为未来涌现的各类AI原生应用提供坚实的网络基础.

2)协议演进,即基于QUIC的内容传输.作为HTTP/3的基础,QUIC协议凭借其低连接开销、无队头阻塞、端到端加密等优势,正重新定义互联网传输层.当前,业界正积极探索在QUIC之上承载更多应用语义.其中,基于QUIC的媒体(media over QUIC, MoQ)传输[54]便是典型代表,它将媒体信令和数据块直接映射至QUIC流,致力于为点播、直播、实时、交互式等多样化视频应用提升传输效率.这一趋势可能会从媒体领域拓展至更广泛的内容交付,形成“Content over QUIC(CoQ)”的演进范式.无论何种形式的交互应用,其内在的内容结构与优先级信息都可映射为QUIC的原生语义,实现更加精细化的内容传输,最终构建一个面向通用内容的高效交付体系.

3)架构革新,即AI驱动的互联网设计.“AI for Network”正推动网络设计从人工经验驱动向智能数据驱动的深刻变革[55-56].未来,这一演进将与领域定制网络(domain specific networks, DSN)[57-58]理念深度融合.DSN主张针对特定应用领域(如工业互联网、智算中心)的需求,定制化设计专用的网络架构与协议,而AI技术将成为实现DSN的核心使能工具.一方面,AI可以用于深度理解和建模特定领域的流量特征与性能需求;另一方面,AI强大的推理与生成能力可辅助甚至自主完成网络协议、调度方法和控制策略等的优化与演进,实现网络的“自配置、自优化、自修复”,从而构建出高度契合领域需求的智能、高效、敏捷的定制化网络.

4)范式变迁,即端网的深度融合.终端与网络之间的信息鸿沟是制约传输性能的关键瓶颈.未来的演进方向将打破这一隔阂,实现端网深层次的融合演进.以无线网络(如Wi-Fi、5G)为例,其向终端适度开放其底层的物理层与MAC层信息,例如信道质量、时频资源分配、CSMA/CA的竞争窗口与退避状态等.终端应用可以基于这些以往不可见的网络设备及链路信息,精准感知链路的瞬时容量与抖动,从而在应用层主动调整数据发送策略,例如自适应调整视频码率、优化游戏指令的发送时机等.这种跨越协议栈的融合设计与演进,将无线链路从一个不可靠的“黑盒”转变为一个可预测的“灰盒”甚至“白盒”,实现端到端传输的精细化、主动式控制.

5 总 结

互联网系统的演进正处在一个关键的十字路口,一边是5G/6G、卫星互联网等构成的高度异构和巨量动态的网络基座,另一边则是视频交互、智能体等要求极致低时延与海量并发的上层应用.如何弥合二者之间巨大的性能鸿沟,是当前互联网传输体系研究的核心议题,也是决定未来数字业务价值的关键.

本文从应用、传输和网络3个层次，系统性梳理了当前互联网传输体系面临的挑战. 传统分层解耦、单点被动的优化范式，在应对流量突发、网络跳变与业务多样化的三重挑战时已凸显出力所不及. 我们从实践中思考认为，破局的关键在于突破现有体系的局限，构建一套“跨层、跨栈、全网协同”的新一代传输体系，不仅是针对传统互联网传输理论的一次重要演进，也为解决万物互联时代的性能瓶颈提供了关键技术支撑，为构建高效、智能、自主的新一代网络基础设施奠定了基础.

作者贡献声明：吕格瑞提出了论文整体框架；赵员康、张佳兴、潘恒撰写了第1~3节内容；武庆华撰写了第4~5节内容；李振宇完善了论文思路并修改论文.

参 考 文 献

[1] Fouladi S, Emmons J, Orbay E, et al. Salsify: Low-latency network video through tighter integration between a video codec and a transport protocol[C]//Proc of the 15th USENIX Symp on Networked Systems Design and Implementation (NSDI’18). Berkeley, CA: USENIX Association, 2018: 267−282

[2] Qin Yanyuan, Hao Shuai, Pattipati K R, et al. ABR streaming of VBR-encoded videos: Characterization, challenges, and solutions[C]//Proc of the 14th Int Conf on Emerging Networking Experiments and Technologies (CoNEXT’18). New York: ACM, 2018: 366−378

[3] Zhou Anfu, Zhang Huanhuan, Su Guangyuan, et al. Learning to coordinate video codec with transport protocol for mobile video telephony[C]//Proc of the 25th Annual Int Conf on Mobile Computing and Networking (MobiCom’19). New York: ACM, 2019: 1−16

[4] Jia Zhidong, Zhang Yihang, Li Qingyang, et al. Tackling bit-rate variation of RTC through frame-bursting congestion control[C]//Proc of the 32nd Int Conf on Network Protocols (ICNP’24). Piscataway, NJ: IEEE, 2024: 1−11

[5] Dobrian F, Sekar V, Awan A, et al. Understanding the impact of video quality on user engagement[C]//Proc of the ACM SIGCOMM 2011 Conf. New York: ACM, 2011: 362−373

[6] Sun Yi, Yin Xiaoqi, Jiang Junchen, et al. CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction[C]//Proc of the 2016 ACM SIGCOMM Conf. New York: ACM, 2016: 272−285

[7] Yan F Y, Ayers H, Zhu Chenzhi, et al. Learning in situ: A randomized experiment in video streaming[C]//Proc of the 17th USENIX Symp on Networked Systems Design and Implementation (NSDI’20). Berkeley, CA: USENIX Association, 2020: 495−511

[8] Lv Gerui, Wu Qinghua, Tan Qingyue, et al. Accurate throughput prediction for improving QoE in mobile adaptive streaming[J]. IEEE Transactions on Mobile Computing, 2023, 23(5): 5799−5817

[9] Agarwal N, Pan R, Yan F Y, et al. Mowgli: Passively learned rate control for real-time video[C]//Proc of the 22nd USENIX Symp on Networked Systems Design and Implementation (NSDI’25). Berkeley, CA: USENIX Association, 2025: 579−594

[10] Zhao Yuankang, Yang Furong, Lv Gerui, et al. MARC: Motion-aware rate control for mobile e-commerce cloud rendering[C]//Proc of 2025 USENIX Annual Technical Conf (ATC’25). Berkeley, CA: USENIX Association, 2025: 217−232

[11] Carlucci G, De Cicco L, Holmer S, et al. Analysis and design of the google congestion control for web real-time communication (WebRTC)[C]//Proc of the 7th Int Conf on Multimedia Systems. New York: ACM, 2016: 1−12

[12] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast[C]//Proc of the ACM SIGCOMM 2002 Conf. New York: ACM, 2002: 205−217

[13] Song Zhenyu, Berger D S, Li Kai, et al. Learning relaxed belady for content distribution network caching[C]//Proc of the 17th USENIX Symp on Networked Systems Design and Implementation (NSDI’ 20). Berkeley, CA: USENIX Association, 2020: 529−544

[14] Sundarrajan A, Kasbekar M, Sitaraman R K, et al. Midgress-aware traffic provisioning for content delivery[C]//Proc of 2020 USENIX Annual Technical Conf (ATC’20). Berkeley, CA: USENIX Association, 2020: 543−557

[15] Berger D S, Sitaraman R K, Harchol-Balter M. AdaptSize: Orchestrating the hot object memory cache in a content delivery network[C]//Proc of the 14th USENIX Symp on Networked Systems Design and Implementation (NSDI’17). Berkeley, CA: USENIX Association, 2017: 483−498

[16] Li Jinyang, Li Zhenyu, Lu Ri, et al. LiveNet: A low-latency video transport network for large-scale live streaming[C]//Proc of the ACM SIGCOMM 2022 Conf. New York: ACM, 2022: 812−825

[17] Tian Yu, Li Zhenyu, Liu M Y, et al. Cost-saving streaming: Unlocking the potential of alternative edge node resources[C]//Proc of the 2024 ACM on Internet Measurement Conf. New York: ACM, 2024: 580−587

[18] Zhao Yuankang, Wu Qinghua, Lv Gerui, et al. JitBright: Towards low-latency mobile cloud rendering through jitter buffer optimization[C]//Proc of the 34th Edition of the Workshop on Network and Operating System Support for Digital Audio and Video. New York: ACM, 2024: 36−42

[19] Zhao Yuankang, Wu Qinghua, Lv Gerui, et al. Understanding and taming the inflated latency in mobile cloud rendering[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2025. https://dc.acm.arg/doi/abs/10.1145/3746283

[20] Schinazi D, Pardue L. HTTP datagrams and the capsule protocol[EB/OL]. (2022-08-24)[2025-06-30]. https://datatracker.ietf.org/doc/rfc9297/

[21] Schinazi D. Proxying UDP in HTTP[EB/OL]. (2023-02-21)[2025-06-30]. https://datatracker.ietf.org/doc/rfc9298/

[22] Zhang Jiaxing, Yang Furong, Liu Ting, et al. TECC: Towards efficient QUIC tunneling via collaborative transmission control[C]//Proc of the 21st USENIX Symp on Networked Systems Design and Implementation (NSDI’24). Berkeley, CA: USENIX Association, 2024: 253−266

[23] Floyd S, Ramakrishnan Dr K K, Black D L. The addition of explicit congestion notification (ECN) to IP[EB/OL]. (2020-01-21)[2025-06-30]. https://datatracker.ietf.org/doc/rfc3168/

[24] Zhang Jiaxing, Wu Qinghua, Lv Gerui, et al. Predictable real-time video latency control with frame-level collaboration[C]//Proc of 2025 IEEE Real-Time Systems Symp (RTSS). Piscataway, NJ: IEEE, 2025: 1−14

[25] Guan Yu, Zheng Chengyuan, Zhang Xinggong, et al. Pano: Optimizing 360 video streaming with a better understanding of quality perception[C]//Proc of the ACM SIGCOMM 2019 Conf. New York: ACM, 2019: 394−407

[26] Baig G, He Jian, Qureshi M A, et al. Jigsaw: Robust live 4K video streaming[C]//Proc of the 25th Annual Int Conf on Mobile Computing and Networking (MobiCom'19). New York: ACM, 2019: 1−16

[27] Wang Shibo, Yang Shusen, Li Hailiang, et al. SalientVR: Saliency-driven mobile 360-degree video streaming with gaze information[C]// Proc of the 28th Annual Int Conf on Mobile Computing And Networking (MobiCom'22). New York: ACM, 2022: 542−555

[28] Ford A, Raiciu C, Handley M J, et al. TCP Extensions for multipath operation with multiple addresses[EB/OL]. (2024-01-25)[2025-06-30]. https://datatracker.ietf.org/doc/rfc8684/

[29] Raiciu C, Paasch C, Barre S, et al. How hard can it be? designing and implementing a deployable multipath TCP[C]//Proc of the 9th USENIX Symp on Networked Systems Design and Implementation (NSDI'12). Berkeley, CA: USENIX Association, 2012: 399−412

[30] Zheng Zhilong, Ma Yunfei, Liu Yanmei, et al. XLINK: QoE-driven multi-path QUIC transport in large-scale video services[C]//Proc of the 2021 ACM SIGCOMM 2021 Conf. New York: ACM, 2021: 418−432

[31] Dhawaskar S S, Lee K, Grunwald D, et al. Converge: QoE-driven multipath video conferencing over WebRTC[C]//Proc of the ACM SIGCOMM 2023 Conf. New York: ACM, 2023: 637−653

[32] Ni Yunzhe, Zheng Zhilong, Lin Xianshang, et al. Cellfusion: Multipath vehicle-to-cloud video streaming with network coding in the wild[C]//Proc of the ACM SIGCOMM 2023 Conf. New York: ACM, 2023: 668−683

[33] Zhou Yuhan, Wang Tingfeng, Wang Tingfeng, et al. AUGUR: Practical mobile multipath transport service for low tail latency in real-time streaming[C]//Proc of the 21st USENIX Symp on Networked Systems Design and Implementation (NSDI'24). Berkeley, CA: USENIX Association, 2024: 1901−1916

[34] Guo Y E, Nikravesh A, Mao Z M, et al. Accelerating multipath transport through balanced subflow completion[C]//Proc of the 23rd Annual Int Conf on Mobile Computing and Networking (MobiCom 17). New York: ACM, 2017: 141−153

[35] Shi Hang, Cui Yong, Wang Xin, et al. STMS: Improving MPTCP throughput under heterogeneous networks[C]//Proc of 2018 USENIX Annual Technical Conf (ATC'18). Berkeley, CA: USENIX Association, 2018: 719−730

[36] Saha S K, Aggarwal S, Pathak R, et al. MuSher: An agile multipath-TCP scheduler for dual-band 802.11 ad/ac wireless LANs[C]//Proc of the 25th Annual Int Conf on Mobile Computing and Networking (MobiCom'19). New York: ACM, 2019: 1−16

[37] Lv Gerui, Wu Qinghua, Liu Yanmei, et al. Chorus: Coordinating mobile multipath scheduling and adaptive video streaming[C]//Proc of the 30th Annual Int Conf on Mobile Computing and Networking (MobiCom'24). New York: ACM, 2024: 246−262

[38] Tan Qingyue, Lv Gerui, Fang Xing, et al. Accurate bandwidth prediction for real-time media streaming with offline reinforcement learning[C]//Proc of the 15th ACM Multimedia Systems Conf. New York: ACM, 2024: 381−387

[39] Pfaff B, Pettit J, Koponen T, et al. The design and implementation of open vSwitch[C]//Proc of the 12th USENIX Symp on Networked Systems Design and Implementation (NSDI'15). Berkeley, CA: USENIX Association, 2015: 117−130

[40] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69−74

[41] Rashelbach A, Rottenstreich O, Silberstein M. Scaling open {vSwitch} with a computational cache[C]//Proc of the19th USENIX Symp on Networked Systems Design and Implementation (NSDI'22). Berkeley, CA: USENIX Association, 2022: 1359−1374

[42] Rizzo L. Netmap: A novel framework for fast packet I/O[C]//Proc of the 21st USENIX Security Symp (Security'12). Berkeley, CA: USENIX Association, 2012: 101−112

[43] Pan Heng, He Peng, Li Zhenyu, et al. Hoda: A high-performance Open vSwitch dataplane with multiple specialized data paths[C]//Proc of the 19th European Conf on Computer Systems. New York: ACM, 2024: 82−98

[44] Zhang Penghao, Pan Heng, Li Zhenyu, et al. Accelerating LSH-based distributed search with in-network computation[C]//Proc of IEEE Conf on Computer Communications (INFOCOM'21). Piscataway, NJ: IEEE, 2021: 1−10

[45] Zhang Penghao, Pan Heng, Li Zhenyu, et al. NetSHa: In-network acceleration of LSH-based distributed search[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(9): 2213−2229

[46] Pan Heng, Cui Penglai, Li Zhenyu, et al. Zebra: Accelerating distributed sparse deep training with in-network gradient aggregation for hot parameters[C]//Proc of the 32nd Int Conf on Network Protocols (ICNP'24). Piscataway, NJ: IEEE, 2024: 1−11

[47] Pan Heng, Li Zhenyu, Zhang Penghao, et al. Misconfiguration-free compositional SDN for cloud networks[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 20(3): 2484−2499

[48] Cui Penglai, Pan Heng, Li Zhenyu, et al. Enabling in-network floating-point arithmetic for efficient computation offloading[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 4918−4934

[49] Liu Yuhan, Li Hanchen, Cheng Yihua, et al. CacheGen: KV cache compression and streaming for fast large language model serving[C]//Proc of the ACM SIGCOMM 2024 Conf. New York: ACM, 2024: 38−56

[50] Qian Kun, Xi Yongqing, Cao Jiamin, et al. Alibaba HPN: A data center network for large language model training[C]//Proc of the ACM SIGCOMM 2024 Conf. New York: ACM, 2024: 691−706

[51] Zhong Yinmin, Liu Shengyu, Chen Junda, et al. DistServe: Disaggregating prefill and decoding for goodput-optimized large

language model serving[C]//Proc of the 18th USENIX Symp on Operating Systems Design and Implementation (OSDI'24). Berkeley, CA: USENIX Association, 2024: 193−210

[52] Zhao Liangyu, Pal S, Chugh T, et al. Efficient direct-connect topologies for collective communications[C]//Proc of the 22nd USENIX Symp on Networked Systems Design and Implementation (NSDI'25). Berkeley, CA: USENIX Association, 2025: 705−737

[53] Warraich E, Shabtai O, Manaa K, et al. OptiReduce: Resilient and tail-optimal AllReduce for distributed deep learning in the Cloud[C]//Proc of the 22nd USENIX Symp on Networked Systems Design and Implementation (NSDI'25). Berkeley, CA: VSENTX Association, 2025: 685−703

[54] Internet Engineering Task Force. Media over QUIC (MoQ)[EB/OL]. (2025-06-23)[2025-06-30]. https://datatracker.ietf.org/wg/moq/

[55] Wu Duo, Wang Xinda, Qiao Yaqi, et al. NetLLM: Adapting large language models for networking[C]//Proc of the ACM SIGCOMM 2024 Conf. New York: ACM, 2024: 661−678

[56] He Zhiyuan, Gottipati A, Qiu Lili, et al. Designing network algorithms via large language models[C]//Proc of the 23rd ACM Workshop on Hot Topics in Networks. New York: ACM, 2024: 205−212

[57] Song Congxi, Han Biao, Li Ruidong, et al. Aquilas: Adaptive QoS-oriented multipath packet scheduler with hierarchical intelligence for QUIC[C]//Proc of the 44th Int Conf on Distributed Computing Systems (ICDCS). Piscataway, NJ: IEEE, 2024: 485−495

[58] Chen Xin, Han Biao, Xu Cao, et al. MPP: A paradigm to reconstruct multipath transmission in user-space[C]//Proc of 2024 IEEE/CIC Int Conf on Communications in China (ICCC). Piscataway, NJ: IEEE, 2024: 1081−1086

Lü Gerui, born in 1994. PhD, assistant professor. His main research interests include network transport protocols and Internet measurements.

吕格瑞, 1994 年生. 博士. 助理研究员. 主要研究方向为网络传输协议、互联网测量.

Zhao Yuankang, born in 1997. PhD candidate. His main research interests include low-latency video transmission and intelligent congestion control algorithms.

赵员康, 1997 年生. 博士研究生. 主要研究方向为低时延视频传输、智能拥塞控制算法.

Zhang Jiaxing, born in 1998. PhD candidate. His main research interests include network transport protocols and real-time video systems.

张佳兴, 1998 年生. 博士研究生. 主要研究方向为传输协议、实时视频系统.

Pan Heng, born in 1990. PhD, associate professor. His main research interests include programmable network and intelligent computing network.

潘　恒, 1990 年生. 博士, 副研究员. 主要研究方向为可编程网络、智算网络.

Wu Qinghua, born in 1987. PhD, associate professor, PhD supervisor. His main research interests include network transport protocols and Internet measurements.

武庆华, 1987 年生. 博士, 副研究员, 博士生导师. 主要研究方向为网络传输协议、互联网测量.

Li Zhenyu, born in 1980. PhD, professor, PhD supervisor. His main research interests include Internet measurements and network systems.

李振宇, 1980 年生. 博士, 研究员, 博士生导师. 主要研究方向为互联网测量、网络系统.