arXiv:2601.18670v1 [cs.NI] 26 Jan 2026

COMETS：具有网内速率自适应的协调式多目的地视频传输

Yulong Zhang、崔颖、Zili Meng、Abhishek Kumar 和 Dirk Kutscher

摘要——**大规模视频流媒体活动吸引数百万观众同时观看，给现有交付基础设施带来压力。客户端驱动的自适应无法及时应对共享拥塞，而基于服务器的协调则会造成可扩展性瓶颈和单点故障。我们提出 COMETS，这是一种协调式多目的地视频传输框架，利用信息中心网络的请求聚合和网内状态感知等原则，实现可扩展、公平且自适应的速率控制。COMETS 引入了一种新颖的范围兴趣协议和分布式网内决策过程，在尽量减少冗余传输的同时，使接收方组之间的视频质量保持一致。为此，我们开发了一种轻量级分布式优化框架，在无需集中式控制的情况下指导逐跳质量自适应。大量仿真结果表明，与 DASH、MoQ 和信息中心网络基线相比，COMETS 始终能提升带宽利用率、公平性和用户感知体验质量，尤其是在高并发场景下。结果表明，COMETS 是一种实用、可部署的下一代可扩展视频交付方案。**

I. 引言

如今，大型流媒体活动通常会吸引数百万观众，对同时观看视频的需求也在急剧增长。例如，每月观看体育赛事流媒体的观众人数已从2021年的5700万增长到2025 [1], 的9000多万，其中超过17%的用户同时观看多个流 [2]，[3]。这种爆发式增长暴露了现有视频传输架构的根本局限：当数千名用户争夺共享瓶颈资源时，如何维持一致且一般的体验质量（QoE）。

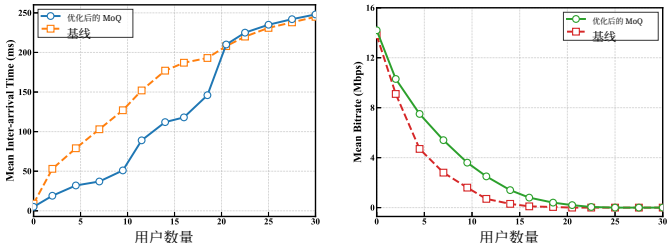
现有基础设施并非为大量同时观看的用户之间进行有效协调与资源共享而设计，因此难以高效管理对相同内容分段的并发请求，也无法充分协调共享基础设施用户之间的网络资源分配 [4]。这些低效问题会导致冗余数据传输和带宽利用率不佳，最终因网络拥塞、比特率不稳定以及缓冲发生率上升而损害用户的体验质量，尤其是在使用高峰期间。为应对这些挑战，理想的视频传输系统必须具备协调性、可扩展性和自适应能力，以最大限度地提高带宽利用率，同时确保所有用户都能获得公平、优质的体验。

Yulong Zhang、崔颖和Dirk Kutscher就职于中国广州的香港科技大学（广州）物联网学域（电子邮箱：Y. Zhang: yzhang893@connect.hkust-gz.edu.cn；Y. Cui：yingcui@ust.hk；D. Kutscher：dku@dkutscher.net）。Zili Meng就职于中国香港的香港科技大学电子与计算机工程学系（电子邮箱：zilim@ieee.org）。Abhishek Kumar就职于芬兰于韦斯屈莱大学信息技术学院（电子邮箱：abhishek.k.kumar@jyu.fi）。

此类系统应聚合同一内容的请求以消除冗余，作出智能的网络内决策，并分配计算负载以避免瓶颈。

当前解决方案在协调性和可扩展性方面存在根本局限。基于客户端自适应的方法，例如基于 HTTP 的动态自适应流媒体（DASH），允许各个客户端独立选择视频表示形式 [5]–[7]。然而，这些决策彼此缺乏协调，且依据的是滞后、局部的网络视图，因此无法及时反映共享网络瓶颈的实际状态，导致带宽竞争和码率振荡。服务器端方法通过集中自适应逻辑 [8], [9],来解决这些局限，并通过全面评估网络状况和用户需求，实现最优资源分配。然而，为大量用户管理状态和控制交互会带来可扩展性挑战，而集中式决策架构则会造成单点故障，可能会——

保证实时性能。我们的实验（图 1）表明，即使是最先进的服务器优化型基于 QUIC 的媒体传输（MoQ），在高并发情况下最终也会遇到与基线方法相同的可扩展性瓶颈。 [1]



（a）延迟与用户负载 （b）平均比特率与用户负载

图 1：随着用户负载增加，基线 MoQ 与服务器优化型 MoQ 的性能对比。

关键洞见。我们发现，有效的多用户视频流媒体需要具备两项特性：I). 聚合感知交付，即合并相同请求以消除冗余传输；以及 II). 分布式协调，即在请求汇聚点而非集中式端点做出自适应决策。因此，我们考虑采用信息中心网络（ICN） [10]。ICN 通过网内缓存和请求聚合，为 CCNx/NDN [11], [12] 等系统中的多用户内容分发带来固有优势。这些特性通过在转发器处合并重复请求来减少冗余传输，

[1]实验设置：使用 Mininet 仿真，链路带宽为 10 Mbps，骨干带宽为 40 Mbps，网络时延为 5 ms；在配备 32 GB 随机存取存储器的 Intel Xeon Platinum 8358P 中央处理器（2.60 GHz，64 核）上进行测试。

©2026年 IEEE。允许将本材料用于个人用途。对于任何其他用途，无论是在当前还是未来的媒体中，包括为广告或宣传目的重印或再版本材料、创作新的合集作品、转售或再分发至服务器或列表，或在其他作品中重复使用本作品中任何受版权保护的组成部分，均须事先获得 IEEE 的许可。

现有基于 ICN 的解决方案 [13]–[15] 侧重于逐跳自适应，而非协调式多用户码率自适应，因此存在决策延迟，且无法确保速率分配高效收敛至稳定、公平的状态（即实现公平的体验质量分布）。为解决这些局限，我们提出 COMETS（网络内速率自适应的协调式多目的地视频传输），这是一个可扩展的、基于 ICN 的多目的地视频流媒体框架，旨在解决大规模视频交付中的挑战：冗余数据传输、缺乏可扩展协调以及系统收敛效率低下。

设计理念。COMETS 基于三项原则，使其有别于以往工作：*I).* 面向群组而非个体的优化。COMETS 不让每个客户端独立选择比特率，而是将能力和网络状况相似的接收端划分为群组，然后在每个群组内统一视频质量。这将个体决策的组合复杂性转化为易于处理的群组级优化。*II).* 主动适应而非被动响应。不同于对拥塞信号作出反应的现有信息中心网络方案，COMETS 采用分布式拉格朗日框架，由转发器交换对偶变量（价格信号），以预判上游约束。这使其能够主动协调，而无需集中式状态收集。*III).* 可部署的覆盖架构。COMETS 无需修改网络基础设施。为确保可部署性，COMETS 采用灵活的架构，可作为应用层覆盖网络部署于现有互联网协议之上（例如，通过 UDP 运行 HTTP/QUIC），类似于 Akamai [16] 或 CloudFlare [17] 等内容分发网络。它无需修改基础设施，并假设同一管理域 [18],内的中间节点可信，从而无需更改网络层即可立即集成到当今的网络中。虽然 COMETS 与 MoQ 一样，致力于将智能移入网络，但它通过信息中心网络原语实现逐跳优化，避免了中心瓶颈，并可作为覆盖网络部署在支持 MoQ 的基础设施之上。

我们的方法。 COMETS 将视频流媒体从孤立的端点控制转变为协调式的网络内协商，主要贡献有四项：

- **用于协调自适应的范围兴趣协议**。我们提出了一种新型协议，客户端表达的是分辨率范围，而不是特定的质量级别。这使转发器能够聚合请求，并跨用户群组优化分辨率分配，将自适应逻辑从端点转移到网络结构中。
- **无中心瓶颈的可扩展架构**。COMETS 将自适应逻辑分布在各个转发器中，将请求聚合与逐跳决策相结合。
- **采用闭式解的分布式优化**。我们将协调的多目的地视频传输形式化为一个统一的整数线性规划（ILP）问题，并开发了一种两阶段分布式算法。不同于依赖启发式方法或被动拥塞信号的既有信息中心网络（ICN）方法，我们的方法推导出逐跳质量决策的解析闭式解，从而能够主动进行面向用户群组的速率分配，并提供经验证的收敛保证-

可靠的收敛保证。

- **实现与评估：** 通过在 Mini-NDN 上进行大规模仿真（最多包含 300 个并发客户端），我们证明 COMETS 在所有测试规模下均能达到高于 0.7 的稳定体验质量评分，而基线在高并发情况下会降至 0.5 以下。COMETS 保持了近乎完美的公平性（Jain 指数≥ 0.93），并在 50ms 内实现优化收敛，速度最高比集中式方法快 3.7×。

表 I 总结了本文中使用的主要缩略语。本文其余部分安排如下：第 II 节考察该领域的相关工作；第 III 节介绍全面的系统设计；第 IV 节和第 V 节详细介绍我们实现的分布式决策算法；第 VI 节给出详尽的实验评估结果；第 VIII 节总结本文的意义并探讨未来研究方向。

表 I：主要缩略语

缩写	全称
ABR	自适应码率
AIMD	加性增大/乘性减小
CDN	内容分发网络
DAG	有向无环图
DASH	基于 HTTP 的动态自适应流媒体
ICN	信息中心网络
ILP	整数线性规划
MoQ	基于 QUIC 的媒体传输
NDN	命名数据网络
PIT	待处理兴趣表
QoE	体验质量
VMAF	视频多方法评估融合

II. 相关工作

基于客户端和基于服务器的自适应

早期的 DASH 研究侧重于利用客户端信息进行单用户自适应。例如，Li 等人 [5] 开发了一种探测并自适应方案，类似于 TCP 的加性增、乘性减（AIMD）机制。Zhou 等人 [6]采用马尔可夫决策过程，针对画质、切换和缓冲区状态进行优化。尽管这些方法对单个用户有效，却无法感知更广泛的网络状况，也无法在用户争用共享资源时进行协调。相比之下，COMETS 利用信息中心网络（ICN）的网内聚合实现协调且能感知群组的自适应，避免了孤立的客户端决策固有的振荡和低效。对于多用户扩展，Tan 等人的博弈论模型 [19] 利用收益函数实现公平性，而 SAND 等网络辅助方法 [20]则提供 QoS 提示。近期的强化学习方法[8], [21] 改进了自适应决策，但会增加开销，且往往优先考虑个体收益而非全局收益。基于 QUIC 的解决方案，例如 Gurel 等人的 I 帧带宽估算[22] 和 Ravuri 等人的混合可靠/不可靠传输[23], ，改善了个体决策，但无法阻止用户争用带宽时出现振荡。Altamimi 等人 [8]通过集中自适应逻辑来解决协调问题，但会面临随规模扩大而加剧的计算瓶颈。COMETS

则不同，它将优化分布到各个转发器，从而消除集中式瓶颈，并在无需管理全局状态的情况下实现可扩展且具备韧性的协调。

基于信息中心网络的流媒体传输与聚合

基于 ICN 的方法借助待定兴趣表（PIT） [24], , 具备固有的内容聚合能力。待定兴趣表会在每个转发器中存储兴趣状态，以执行有状态转发操作，包括互联网聚合。对于单用户自适应，Liu 等人 [11] 开发了 HAVS-CCN，在拥塞期间通过选择性丢层进行调整。Saltarin 等人[14] 修改了 ICN 的数据结构，利用网络编码来使用连接客户端与源的多条路径。对于多用户场景，Wu 等人[13] 提出了 NDN-MMRA，用于无线网络中的多阶段组播码率自适应，通过多阶段传输调整码率，以平衡高速和低速消费者的需求。Alt 等人 [15]提出了上下文质量自适应（CBA）算法，将自适应建模为上下文多臂老虎机问题。尽管这些方法各有优点，但它们主要依赖局部反应（即拥塞信号），缺乏复杂机制来预测上游约束或协调不同用户群组。这导致在面对高用户并发和不断变化的网络状况时出现决策延迟和不稳定。COMETS 更进一步，突破了反应式本地自适应的局限，引入了具有解析闭式解的分布式优化框架，从而实现主动的群组感知速率分配，在充分利用信息中心网络的聚合优势的同时，确保快速收敛和公平性。

覆盖/组播系统

COMETS 也与传统的分层覆盖组播流媒体系统有着根本区别。这些系统主要致力于构建高效的覆盖拓扑（树、网状结构或混合结构），并在节点动态变化时维护这些拓扑。代表性系统包括带宽自适应分层组播覆盖（ BAHMO） [25],, 它通过将对等节点划分到集群中，并在每个集群中设置服务器来构建分层多播树；以及覆盖组播树协议（OMTP） [26], , 它着重改进覆盖多播树的加入和维护流程。这些系统投入计算资源，根据时延、带宽可用性和节点稳定性等因素优化树结构。相比之下，COMETS 解决的是一个正交问题：在任意分发网络中协调多用户码率自适应。覆盖组播优化的是“数据流向何处”，而 COMETS 优化的是“向每个用户群组传输何种质量版本，以最大化总体体验质量”。从本质上说，COMETS能够补充覆盖组播；它可以运行在任何底层拓扑之上，包括覆盖多播树，为其增添这类系统传统上所缺乏的自适应智能。

III. COMETS 系统设计

鉴于端主机自适应和纯集中式协调都存在不足，COMETS 将自适应逻辑分散到网络本身。我们介绍 COMETS 系统设计，包括架构、协议设计和安全方面。从根本上说，COMETS 将视频流媒体重新构想为网络结构中的分布式协商：转发器（充当智能中间节点）在分层树中递归地汇总、优化和调整内容流。

A. 系统架构

COMETS 采用分布式分层架构，网络转发器会主动参与内容分发优化。总体而言，转发器充当分散式决策节点，汇总下游客户端的请求，监测局部网络状况（例如带宽、拥塞），并为用户群组计算最优视频分辨率，以最大化整体体验质量，同时尽量减少冗余。这一过程会在分发树中递归进行：叶转发器处理直接连接的客户端，并将汇总数据向上游传递；父转发器则针对更大的子树进行优化，从而在无需全局同步的情况下确保可扩展性。

我们假设转发器由视频提供商或内容分发网络运营商（例如边缘PoP节点）部署，并处于同一管理域内。这使其能够控制信任锚点，并支持高效的缓存部署和范围兴趣处理。

图 2 展示了系统架构，图 3 展示了客户端、转发器与服务器之间的通信流程。请求视频内容时，客户端会发送基于范围的兴趣包，其中包含其支持的分辨率能力（例如 [2K、4K、8K]），而不是直接请求特定分辨率。转发器会处理这些兴趣包，计算最优配置——具体而言，根据汇总的下游需求和局部约束，确定最佳分辨率分配，以平衡客户端能力、上游带宽可用性和公平性指标（例如尽量缩小体验质量方差）。我们最初开发了集中式 ILP 方法作为理论基线（问题 1），该方法能够求得全局最优解，但在数据收集和可扩展性方面存在部署限制。为此，我们设计了分布式拉格朗日算法（问题3），使转发器能够求解局部子问题，仅凭邻域信息即可达到接近集中式方法的性能。从根本上说，该方法将分辨率分配这一组合优化难题分解为可通过解析方法求解的部分，从而实现能够随网络规模扩展的实时决策。

B. 协议设计

COMETS 的运行依赖客户端、转发器与协调节点之间有序的消息交换，如图 3 所示。客户端会定期（每隔 $T = 4$ 秒，与标准自适应流媒体实践保持一致）发送 ICN范围兴趣消息，按照命名约定*/ndn/video/TitleID/RangeIn-terest/chunk=seq[/nonce=val]*。这些消息会传达客户端标识符、可接受分辨率级别 L_u 和体验质量权重 w_u （定义见第 V 节）。随后，客户端等待分辨率分配结果，并使用这些结果通过标准ICN兴趣请求视频数据。转发器是主要的计算与交互实体，负责汇总下游客户端能力并监测网络状况。

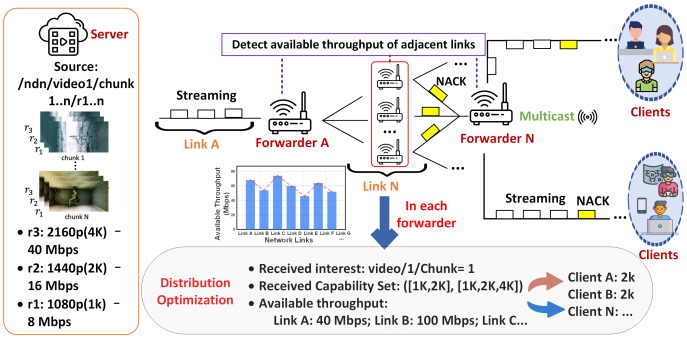


图2：系统架构。转发器根据客户端能力和可用带宽优化质量决策。

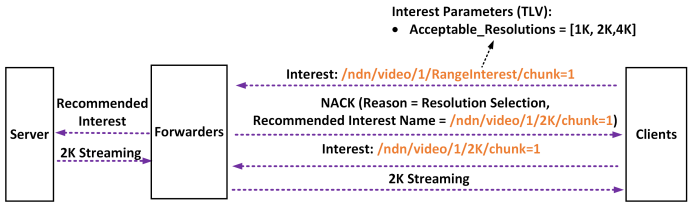


图3：通过兴趣包进行客户端-转发器-服务器通信。

根据部署配置，转发器会实现：

1)**集中式协调模式：**转发器报告汇总信息给指定的优化器，使用信息中心网络兴趣包 I_{report}（命名模式：*/ndn/opt/report/for-转发器=FwdID/...*）。优化器会计算决策并分发签名配置数据，转发器会通过带版本兴趣包获取 I_{config}（例如，*/ndn/opt/config/forwarder=FwdID/v=version*）。过时的版本请求会触发否定确认，指示最新版本，从而促使发送正确的请求。验证通过后，转发器应用这些决策。其可扩展性受限于集中式中心节点。

2) **分布式优化模式：** 转发器通过带版本的签名 ICN 兴趣包/数据包对与对等节点交换状态变量和拉格朗日乘子（例如，I_{state_req}(节点 *ID*、变量、级别、 t) 遵循/*ndn/comets/s-tate/node/NodeID/Var/Level/v=t*）。每个转发器使用第 V 节中的解析解求解局部子问题，根据方程更新其对偶变量，并将结果作为带版本的签名 ICN 数据发布（$D_{\text{state_pub}}$(节点 ID、变量、级别、 $t+1$)）。这一获取—计算—更新—发布周期在 4 秒的优化窗口内迭代进行。

确定客户端 u 的最优分辨率 Res_u^* 后，转发器会主动获取视频片段：通过发送标准的信息中心网络兴趣包来请求数据$N_{video}(TitleID,\ Res_u^*,\ seq)$。随后，转发器会通过带签名的否定确认通知客户端这一最优分辨率，其中的推荐名称字段指定了模式*/ndn/video//Res_u^*/chunk=*，作为提供给客户端的经验证指令。收到这一建议后，客户端会为该片段发送自己的标准信息中心网络兴趣包。这一转发器采用的预取策略旨在确保客户端请求到达时，所需的数据片段可能已缓存在本地或正在传输途中。

范围兴趣包在请求-响应模式上类似于远程过程调用（*RPC*），但语义有所不同，因为它们不假定存在特定的“服务器”作为 RPC 目标。范围兴趣包充当一种分布式协商协议：它们被发送到网络中，而不指向特定服务器，使专用的可信转发器能够根据本地状态和传播来的状态进行拦截、聚合并协作响应。关键区别在于，COMETS 将速率协商语义嵌入兴趣包—数据包交换本身，从而无需依赖反复的反馈循环或逐客户端控制路径，即可实现去中心化的优化决策。

C. 可靠性、拥塞控制与安全性

为确保视频在不同网络状况下都能可靠交付，COMETS 将消费者驱动的传输机制与主动式网络内拥塞控制和严格的信任模型相结合。该设计使系统无需依赖繁重的面向连接协议，即可应对丢包、节点故障和安全威胁。

传输机制与丢包恢复。COMETS 依靠信息中心网络的有状态转发平面来保证可靠性。我们采用接收方驱动的恢复策略，以有效应对丢包。客户端会为每个发出的兴趣包设置一个重传计时器。计时器超时表示可能发生了丢包或链路故障。检测到超时后，客户端会发起有限次数的重传（例如，最多尝试 2 次），以恢复无线干扰等因素导致的暂时性随机丢包。由于数据包会缓存在中间转发器上，重新发出的兴趣包通常可由附近的上游节点通过局部修复来满足，与端到端重传相比，这能显著缩短恢复时延。但是，如果重传失败，或连续多次超时，客户端便会将其判断为网络拥塞或路径故障，而非随机丢包。在这种情况下，客户端会停止重传当前片段，并触发速率自适应逻辑，在后续时间间隔请求较低画质的分辨率。这与优化框架在受限容量下最大化总体体验质量的目标一致。

拥塞控制与动态性。COMETS 采用双环拥塞控制机制，以应对动态带宽波动和用户移动性。在客户端，采用基于加性增、乘性减的窗口控制进行兴趣包流水线传输。窗口大小根据数据到达率进行调整，确保请求速率与瓶颈带宽相匹配。为解决纯客户端方案中常见的“决策延迟”，COMETS 引入了网络内反压。当转发器检测到其下游链路容量已饱和，或其队列超过阈值时，会主动发送携带特定原因代码和建议（例如指向较低分辨率表示）的否定确认包。这样，客户端无需等待超时，即可立即切换到可持续的比特率。该机制还支持无缝移动；当用户切换接入点时，兴趣包的无状态

特性使新的服务转发器能够根据新路径的状态立即处理请求或发送否定确认。

安全性。COMETS 通过单个管理域内类似 CDN 的信任模型，确保内容真实性和协调完整性（例如，类似于 Akamai [16]或 CloudFlare [17] 基础设施）。内容生产者充当信任根，向包括转发器和优化器在内的授权实体颁发证书。关键控制消息（例如状态变量、优化器配置和主动否定确认）均由颁发者签名。转发器根据信任层级验证这些签名，以防止未经授权的状态注入或欺骗攻击。此外，内容块由生产者或经授权的转码器签名。客户端在收到每个数据包时都会验证其签名。这样便将安全性与传输信道解耦，使内容能够从任何缓存或对等节点安全获取，而不必担心数据篡改。在大规模部署中，使用公钥密码学为每个片段签名可能代价高昂 [27]。为提高效率，COMETS 可以采用基于清单的方法，通过哈希链将多个数据对象关联起来，这样每份清单只需签名一次，无需逐个对象签名 [28]。这既能降低密码运算开销，又能保持内容真实性。

后续章节将通过分布式优化框架对这些协调机制进行形式化描述，该框架以数学方式确保所有用户的体验质量分配达到最优。

IV. 系统模型

网络被建模为一个有向无环图（DAG）$\mathcal{G} = (\mathcal{V},\ \mathcal{E})$，其中 $\mathcal{V}$ 是节点集合，$\mathcal{E}$ 是边集合。节点集合被划分为三个互不相交的子集：服务器集合 $\mathcal{S}$（内容源）、转发器集合$\mathcal{F}$（中间节点）和用户集合 $\mathcal{U}$（末端节点）。注意，$\mathcal{V} = \mathcal{S} \cup \mathcal{F} \cup \mathcal{U}$、$\mathcal{S} \cap \mathcal{F} = \varnothing$、$\mathcal{F} \cap \mathcal{U} = \varnothing$ 和 $\mathcal{S} \cap \mathcal{U} = \varnothing$。对于每个节点 $i \in \mathcal{V}$，我们将其上游节点集合和下游节点集合分别定义为 $\mathcal{I}_i = \{j \mid (j,\ i) \in \mathcal{E}\}$ 和 $\mathcal{O}_i = \{j \mid (i,\ j) \in \mathcal{E}\}$。系统提供一组 L 分辨率级别 $\mathcal{L} = \{1,\ 2,\ \ldots\ldots,\ L\}$，其中每个级别 $l \in \mathcal{L}$ 都关联一个带宽需求 B_l 和一个质量评分 Q_l。为了刻画提高分辨率对用户体验的边际收益递减，我们将 Q_l 建模为关于 l的递增凹函数，例如，可以选择 [29]:

$$Q_l = a + b \cdot \ln\left(\frac{R_l}{R_1}\right),\quad l \in \mathcal{L}. \tag{1}$$

其中，R_l 表示与级别 l对应的分辨率，a和 b 为常数系数。每个用户 $u \in \mathcal{U}$ 都有一个权重$w_u > 0$，用于反映其优先级。每条网络链路 $(i,j) \in \mathcal{E}$ 的带宽容量上限为 $C_{i,j} > 0$。

令 $\mathbf{x} \triangleq (x_{i,l})_{i\in\mathcal{V},l\in\mathcal{L}} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{L}|}$ 表示节点的分辨率选择，其中 $x_{i,l} = 1$表示对于所有 $i \in \mathcal{V}$和 $l \in \mathcal{L}$，节点 i选择分辨率为 l的内容。

- 对于所有 $i \in \mathcal{U}$，$x_{i,\ l} = 1$表示用户 i 请求分辨率为 l的内容。
- 对于所有 $i \in \mathcal{F}$，$x_{i,\ l} = 1$ 表示转发器 i 检索并转发分辨率为 l的内容。
- 对于所有 $i \in \mathcal{S}$，$x_{i,\ l} = 1$ 表示服务器 i 提供分辨率为 l的内容。

令 $\mathbf{y} \triangleq (y_{i,\ j,\ l})_{(i,\ j)\in\mathcal{E},\ l\in\mathcal{L}} \in \{0,\ 1\}^{|\mathcal{E}|\times|\mathcal{L}|}$ 表示边缘分辨率传输，其中 $y_{i,\ j,\ l} = 1$ 表示边缘$(i,\ j)$ 传输分辨率为 l 的内容，对于所有 $(i,\ j) \in \mathcal{E}$和$l \in \mathcal{L}$。因此，$\mathbf{x}$ 和 $\mathbf{y}$ 满足:

$$x_{s,l} = 1,\quad s \in \mathcal{S},\ l \in \mathcal{L}, \tag{2}$$
$$x_{u,l} = 0,\quad u \in \mathcal{U},\ l \in \mathcal{L}\backslash\mathcal{L}_u, \tag{3}$$
$$\sum_{l\in\mathcal{L}} x_{u,l} = 1,\quad u \in \mathcal{U}, \tag{4}$$
$$y_{i,j,l} \leqslant x_{i,l},\quad (i,j) \in \mathcal{E},\ l \in \mathcal{L}, \tag{5}$$
$$x_{i,l} \leqslant \sum_{j:(j,i)\in\mathcal{E}} y_{j,i,l},\quad i \in \mathcal{U} \cup \mathcal{F},\ l \in \mathcal{L}, \tag{6}$$
$$\sum_{l\in\mathcal{L}} B_l y_{i,j,l} \leqslant C_{i,j},\quad (i,j) \in \mathcal{E}, \tag{7}$$
$$x_{i,l}, y_{i,j,l} \in \{0,1\},\quad i \in \mathcal{V},\ (i,j) \in \mathcal{E},\ l \in \mathcal{L}. \tag{8}$$

具体而言，(2) 表示每台服务器均提供所有分辨率级别的内容；(3) 表示每个用户只能选择其硬件或网络条件所支持的分辨率级别；(4) 表示每个用户恰好选择一个分辨率级别；(5) 表示每个节点只能传输其已获取的分辨率级别的内容；(6) 表示每个非服务器节点只能获取由上游边传送的分辨率级别的内容；(7) 表示每条边的传输带宽不得超过其带宽容量上限；(8) 表示所有决策变量均为二元变量。网络性能指标设为所有用户的加权和 oE:

$$Z(\mathbf{x},\mathbf{y}) \triangleq \sum_{u\in\mathcal{U}} w_u \sum_{l\in\mathcal{L}} Q_l x_{u,l}.$$

V. 多用户体验质量优化

现在，我们对第 III 节中引入的分辨率选择问题进行形式化。表 II 列出了全文所用符号的完整清单。

A. 问题建模与转换

我们的目标是优化分辨率分配$\mathbf{x}$ 和$\mathbf{y}$，以最大化加权和体验质量 $Z(\mathbf{x},\ \mathbf{y})$，并满足式 (2)–(8) 中的可行性约束。具体而言，我们将其表述为以下优化问题，并假设该问题可行。

问题 *1 (QoE Maximization):*

$$\max_{\mathbf{x},\mathbf{y}} \sum_{u\in\mathcal{U}} w_u \sum_{l\in\mathcal{L}} Q_l x_{u,l}$$
$$\text{s.t.}\quad (2),(3),(4),(5),(6),(7),(8).$$

目标函数以及式 (2)–(7) 中的约束关于$\mathbf{x}$均为线性，而式 (8) 中的约束则要求变量取离散值。因此，问题 1 是一个具有挑战性的整数线性规划问题，通常属于 NP-hard [30]。为高效求解问题 1，我们首先考虑其连续松弛，从而得到线性规划。

问题 *2* 连续松弛 *o* 问题 *1*: f)

$$\max_{\mathbf{x},\mathbf{y}} \sum_{u\in\mathcal{U}} w_u \sum_{l\in\mathcal{L}} Q_l x_{u,l}$$
$$\text{s.t.}\quad (2),(3),(4),(5),(6),(7),$$
$$x_{i,l}, y_{i,j,l} \in [0,1],\quad i \in \mathcal{V},\ (i,j) \in \mathcal{E},\ l \in \mathcal{L}. \tag{9}$$

问题 2（采用式 (9) 中连续约束）可视为问题 1（采用式 (8) 中离散约束）的连续松弛。问题 2 的目标函数和

表 II：符号列表

符号	描述
网络结构	
$\mathcal{G}=(\mathcal{V},\mathcal{E})$	表示该网络的有向无环图
$\mathcal{V}$	所有节点的集合
$\mathcal{S}$	服务器节点集合
$\mathcal{F}$	转发节点集合
$\mathcal{U}$	用户节点集合
$\mathcal{I}_i$	节点 i 的上游节点（入邻居）
$\mathcal{O}_i$	节点 i 的下游节点（出邻居）
视频参数	
$\mathcal{L}=\{1,2,\ldots,L\}$	分辨率级别集合
$\mathcal{L}_u\subseteq\mathcal{L}$	用户 u 支持的分辨率
$B_l>0$	分辨率 l 的带宽需求（随 l 增加）
$Q_l=a+b\cdot\ln(l/l_0)$	分辨率 l 的质量评分（对数模型）
$w_u>0$	用户 u 的优先级权重
$C_{i,j}$	链路 $(i,j)\in\mathcal{E}$ 的容量
决策变量	
$x_{i,l}\in\{0,1\}$	若节点 i 选择分辨率 l，则为 1
$y_{i,j,l}\in\{0,1\}$	若分辨率 l 通过链路 $(i$传输，$j)$，则为 1
$\mathbf{x}=(x_{i,l})_{i\in\mathcal{V},l\in\mathcal{L}}$	所有选择变量的矩阵
$\mathbf{y}=(y_{i,j,l})_{(i,j)\in\mathcal{E},l\in\mathcal{L}}$	所有传输变量的矩阵
对偶变量	
$\boldsymbol{\lambda}_1=\{\lambda_{1,i,l}\geqslant 0\}$	传入传输的乘子约束 (6)
$\boldsymbol{\lambda}_2=\{\lambda_{2,f,k,l}\geqslant 0\}$	传出传输的乘子约束（5）
算法参数	
$\alpha^{(t)},\beta^{(t)}$	迭代 t 时的步长（递减）
α_0,β_0,γ	初始步长参数和衰减率
$T_{\max}$	最大迭代次数
ϵ	收敛容差阈值
σ_i	按密度排列分辨率 λ_1 $,i,l/B_l$
θ_i	容量约束背包填充的截断索引
函数和目标函数	
$Z(\mathbf{x},\mathbf{y})=\sum_{u\in\mathcal{U}}w_u\sum_{l\in\mathcal{L}}Q_l x_{u,l}$	加权平均 QoE 目标
$\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$	松弛问题的拉格朗日函数
$g(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$	对偶函数（对拉格朗日函数进行最大化）

约束函数关于$\mathbf{x}$，$\mathbf{y}$均为线性。因此，问题 2 是一个线性规划 [30, pp. 146]，可使用单纯形法或内点法在多项式时间内求解。然而，对于较大的 L，单纯形法（每次迭代的复杂度为 $\mathcal{O}(L^2)$）和内点法（总体复杂度为 $\mathcal{O}(L^3)$）都非常耗时。此外，问题 2 中线性规划的最优解可能不是整数，因此对问题 1 中原始的整数线性规划而言不可行。

B. 可扩展方法

我们提出一种可扩展方法，以获得问题 1 的近似最优解。[2] 具体而言，我们首先提出一种对偶分解算法，使用对偶分解算法[31]获得问题 2 中线性规划的最优解。然后，基于问题 2 中线性规划的最优解，我们提出一种可行性重构算法，以获得问题 1 的近似最优可行点。

1) 对偶分解算法：鉴于问题 2 是凸的并满足强对偶性，且变量通过约束 (5) 和 (6) 相互耦合，我们通过将约束 (5) 和 (6) 的加权和加入目标函数，构造式 (10) 中的拉格朗日函数 $\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$，如本页顶部所示。这里，$\boldsymbol{\lambda}_1\triangleq(\lambda_{1,i,l})_{i\in\mathcal{U}\cup\mathcal{F},l\in\mathcal{L}}\geq\mathbf{0}$ 以及 $\boldsymbol{\lambda}_2\triangleq(\lambda_{2,f,k,l})_{f\in\mathcal{F},k\in\mathcal{O}_f,l\in\mathcal{L}}\geq\mathbf{0}$分别是与约束 (6) 和 (5) 相关联的拉格朗日乘子。

拉格朗日对偶函数是通过对以下内容求最大值获得的拉格朗日函数 $\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$ 在剩余约束条件下，关于原始变量$(\mathbf{x},\mathbf{y})$ 求最大值：

$$g(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)\triangleq\max_{\mathbf{x},\mathbf{y}}\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)\quad\text{s.t.}\quad(2),(3),(4),(7),(9)\tag{11}$$

对偶问题是求解最优对偶变量（使对偶函数最小化的拉格朗日乘子）：

$$\min_{\boldsymbol{\lambda}_1\geq 0,\boldsymbol{\lambda}_2\geq 0}\quad g(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2).\tag{12}$$

令 $\boldsymbol{\lambda}^*\triangleq(\boldsymbol{\lambda}_1^*,\boldsymbol{\lambda}_2^*)$表示最优拉格朗日乘子。

由于 (10) 中的 $\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$ 关于**可分离**$\mathbf{y}_{s,i}\triangleq(y_{s,i,l})_{l\in\mathcal{L}}$, $s\in\mathcal{S}$, $i\in\mathcal{O}_s$,$\mathbf{x}_u\triangleq(x_{u,l})_{l\in\mathcal{L}}$, $u\in\mathcal{U}$,$\mathbf{x}_f\triangleq(x_{f,l})_{l\in\mathcal{L}}$ 以及 $\mathbf{y}_{f,k}\triangleq(y_{f,k,l})_{l\in\mathcal{L}}$, $f\in\mathcal{F}$, $k\in\mathcal{O}_f$ 可分离，因此 (12) 可拆分为四个由主问题协调的子问题。

Problem 3 (Master Dual Problem):

$$\begin{aligned}\min_{\boldsymbol{\lambda}_1\geq 0,\boldsymbol{\lambda}_2\geq 0}\ &\sum_{s\in\mathcal{S},i\in\mathcal{O}_s}\sum_{l\in\mathcal{L}}\lambda_{1,i,l}y^*_{s,i,l}+\sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}}(w_uQ_l-\lambda_{1,u,l})x^*_{u,l}\\&+\sum_{f\in\mathcal{F}}\sum_{l\in\mathcal{L}}\left(\sum_{k\in\mathcal{O}_f}\lambda_{2,f,k,l}-\lambda_{1,f,l}\right)x^*_{f,l}\\&+\sum_{f\in\mathcal{F},k\in\mathcal{O}_f}\sum_{l\in\mathcal{L}}(\lambda_{1,k,l}-\lambda_{2,f,k,l})y^*_{f,k,l}\end{aligned}\tag{13}$$

其中，$y^*_{s,i,l}$, $x^*_{u,l}$, $x^*_{f,l}$ 和 $y_{f,k,l}$* 分别是在给定对偶变量 i$(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$ 的情况下，问题 4、5、6 和 7 的最优解。

问题*4*（关于 $\mathbf{y}_{s,i}$ 的服务器子问题）：对于所有$\lambda_{1,i,l}\geqslant 0$, $s\in\mathcal{S}$, $i\in\mathcal{O}_s$，我们有：

$$\max_{\mathbf{y}_{s,i}}\quad\sum_{l\in\mathcal{L}}\lambda_{1,i,l}y_{s,i,l}\quad\text{s.t.}\quad(7),(9)$$

问题*5*（关于 $\mathbf{x}_u$ 的用户子问题）：对于所有 $\lambda_{1,u,l}\geqslant 0$, $u\in\mathcal{U}$，我们有：

$$\max_{\mathbf{x}_u}\quad\sum_{l\in\mathcal{L}}(w_uQ_l-\lambda_{1,u,l})x_{u,l}\quad\text{s.t.}\quad(3),(4),(9).$$

问题*6*（关于 $\mathbf{x}_f$ 的转发器子问题）：对于所有$\lambda_{1,f,l}$, $\lambda_{2,k,l}\geqslant 0$, $f\in\mathcal{F}$，我们有：

$$\max_{\mathbf{x}_f}\quad\sum_{l\in\mathcal{L}}\left(\sum_{k\in\mathcal{O}_f}\lambda_{2,f,k,l}-\lambda_{1,f,l}\right)x_{f,l}\quad\text{s.t.}\quad x_{f,l}\in[0,1],\quad l\in\mathcal{L}.$$

问题*7*（关于 $\mathbf{y}_{f,k}$ 的转发器子问题）：对于所有$\lambda_{1,f,l}$, $\lambda_{2,f,k,l}\geqslant 0$, $f\in\mathcal{F}$, $k\in\mathcal{O}_f$，我们有：

$$\max_{\mathbf{y}_{f,k}}\quad\sum_{l\in\mathcal{L}}(\lambda_{1,k,l}-\lambda_{2,f,k,l})y_{f,k,l}\quad\text{s.t.}\quad(7),(9).$$

引理 *1*（对偶问题分解）：式 (12) 中的对偶问题（其对偶函数由式 (11) 给出）等价于问题3中的主对偶问题与问题4、5、6和7中的子问题。

证明：详细证明请见附录 B。 ■

接下来，我们先求解任意给定 $\boldsymbol{\lambda}_1\geq 0$、$\boldsymbol{\lambda}_2\geq 0$ 下的子问题，然后求解主问题以获得 $(\boldsymbol{\lambda}_1^*$、$\boldsymbol{\lambda}_2^*)$。

问题4是一个连续背包问题 [32]。该问题的最

[2]附录 A 对集中式整数线性规划求解器、基于人工智能的方法以及我们提出的方法进行了详细比较分析。

$$\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)=\sum_{u\in\mathcal{U}}w_u\sum_{l\in\mathcal{L}}Q_l x_{u,l}+\sum_{i\in\mathcal{U}\cup\mathcal{F}}\sum_{l\in\mathcal{L}}\lambda_{1,i,l}\left(x_{i,l}-\sum_{j:(j,i)\in\mathcal{E}}y_{j,i,l}\right)+\sum_{f\in\mathcal{F}}\sum_{k\in\mathcal{O}_f}\sum_{l\in\mathcal{L}}\lambda_{2,f,k,l}(y_{f,k,l}-x_{f,l}) \tag{10}$$

优解可通过按价值密度降序排列分辨率得到 $\lambda_{1,\,i,\,l}/B_l, l\in\mathcal{L}$。对于所有 $s\in\mathcal{S}$和 $i\in\mathcal{O}_s$，令 $\sigma_i:\mathcal{L}\to\mathcal{L}$ 为满足$\lambda_{1,\,i,\,\sigma_i(1)}/B_{\sigma_i(1)}\geqslant\cdots\geqslant\lambda_{1,i,\sigma_i(L)}/B_{\sigma_i(L)}$的排列。将截断索引 θ_i 定义为

$$\theta_i\triangleq\begin{cases}L+1, & \text{if } \sum_{k=1}^{L}B_{\sigma_i(k)}\leqslant C_{s,i},\\ \min\left\{p\in\mathcal{L}\,\middle|\,\sum_{k=1}^{p}B_{\sigma_i(k)}>C_{s,i}\right\}, & \text{otherwise.}\end{cases}$$

Then, the optimal solution of the problem is given by

$$y^*_{s,i,\sigma_i(k)}(\lambda_{1,i,\sigma_i(k)})=\begin{cases}1, & k<\theta_i,\\ \frac{C_{s,i}-\sum_{m=1}^{\theta_i-1}B_{\sigma_i(m)}}{B_{\sigma_i(\theta_i)}}, & k=\theta_i,\\ 0, & \text{otherwise,}\end{cases}\qquad k\in\mathcal{L}. \tag{14}$$

证明：问题4是一个连续背包问题，其目标函数为 $\sum_l\lambda_{1,\,i,\,l}y_{s,\,i,\,l}$，容量为 $C_{s,\,i}$。其中各物品的价值为 $\lambda_{1,\,i,\,l}\geqslant 0$，权重为 $B_l>0$。贪心算法按密度 $\rho_l=\lambda_{1,\,i,\,l}/B_l$ 降序排列物品，将完整单位分配给密度最高的物品，直到容量几乎用满，再为边际物品分配一部分。形式化地，设 σ_i 为排列，使得 $\rho_{\sigma_i(1)}\geqslant\rho_{\sigma_i(2)}\geqslant\cdots\geqslant\rho_{\sigma_i(L)}$。找出累计权重超过容量的最小 θ_i。该解将1分配给前 θ_i-1 个物品，并将准确的比例分配给第 θ_i个物品，从而在满足单调性的条件下最大化目标函数。如果所有密度均为零，解显然成立（全部为零）。这将 $\lambda_{1,\,i,\,l}$ 解释为单位带宽的“交付价值”。

■

问题5是一个具有闭式解的线性规划。选择$l^*\in\arg\max_{l'\in\mathcal{L}_u}(w_uQ_{l'}-\lambda_{1,\,u,\,l'})$。问题5的最优解由下式给出：

$$x^*_{u,l}(\lambda_{1,u,l})=\begin{cases}1, & \text{if } l=l^*,\\ 0, & \text{otherwise.}\end{cases} \tag{15}$$

证明：问题5是在单纯形上的线性规划：最大化 $\sum_{l\in\mathcal{L}_u}(w_uQ_l-\lambda_{1,\,u,\,l})x_{u,\,l}$，满足 $\sum_{l\in\mathcal{L}_u}x_{u,\,l}=1$, $x_{u,\,l}\in[0,\ 1]$, 以及 $x_{u,\,l}=0$，其中 $l\notin\mathcal{L}_u$。最优解位于可行多胞体的一个顶点上，而这些顶点就是标准基向量（独热向量）。因此，选择 $l^*=\arg\max_{l'\in\mathcal{L}_u}(w_uQ_{l'}-\lambda_{1,\,u,\,l'})$，令 $x^*_{u,\,l^*}=1$，其余分量设为 0。

■

问题6是一个最优解由下式给出的线性规划：

$$x^*_{f,l}(\lambda_{1,f,l},\boldsymbol{\lambda}_2)=\begin{cases}1, & \sum_{k\in\mathcal{O}_f}\lambda_{2,f,k,l}-\lambda_{1,f,l}>0,\\ 0, & \text{otherwise,}\end{cases} \tag{16}$$

证明：问题6具有一个简单的解析解。对于每个分辨率 l，如果系数 $\sum_{k\in\mathcal{O}_f}\lambda_{2,\,f,\,k,\,l}-\lambda_{1,\,f,\,l}>0$，则令 $x^*_{f,\,l}=1$；否则，令 $x^*_{f,\,l}=0$。这将在盒约束下最大化线性目标函数。

■

问题7也是一个连续背包问题 [32]，可以用与问题4类似的方法求解。对于所有 $f\in\mathcal{F}$ 和 $k\in\mathcal{O}_f$，令 $\sigma_{f,\,k}:\mathcal{L}\to\mathcal{L}$ 为满足条件的排列，即$(\lambda_{1,\,k,\,\sigma_{f,\,k}(1)}-\lambda_{2,\,f,\,k,\,\sigma_{f,\,k}(1)})/B_{\sigma_{f,k}(1)}\geqslant\cdots\geqslant(\lambda_{1,k,\sigma_{f,k}(L)}-\lambda_{2,\,f,\,k,\,\sigma_{f,\,k}L})/B_{\sigma_{f,\,k}L}$。定义截断索引 $\theta_{f,(\,)}$

$$\theta_{f,k}\triangleq\begin{cases}L+1, & \text{if } \sum_{m=1}^{L}B_{\sigma_{f,k}(m)}\leqslant C_{f,k},\\ \min\left\{p\in\mathcal{L}\,\middle|\,\sum_{m=1}^{p}B_{\sigma_{f,k}(m)}>C_{f,k}\right\}, & \text{otherwise.}\end{cases}$$

那么，问题7的最优解为：

$$y^*_{f,k,\sigma_{f,k}(m)}(\lambda_{1,k,\sigma_{f,k}(m)},\lambda_{2,f,k,\sigma_{f,k}(m)})=\begin{cases}1, & m<\theta_{f,k},\\ \frac{C_{f,k}-\sum_{j=1}^{\theta_{f,k}-1}B_{\sigma_{f,k}(j)}}{B_{\sigma_{f,k}(\theta_{f,k})}}, & m=\theta_{f,k},\\ 0, & \text{otherwise,}\end{cases}\qquad m\in\mathcal{L}. \tag{17}$$

证明：对于每个下游节点 $k\in\mathcal{O}_f$，问题7都是一个连续背包问题。目标函数是最大化$\sum_l(\lambda_{1,k,l}-\lambda_{2,f,k,l})y_{f,k,l}$，满足 $\sum_l B_l y_{f,k,l}\leqslant C_{f,k}$ 且$y_{f,k,l}\in[0,1]$。按密度 $(\lambda_{1,k,l}-\lambda_{2,\,f,k,l})/B_l$降序排列分辨率，并排除系数为负的项。应用贪心算法：优先为高密度分辨率分配全部容量，直到容量用尽；对边际分辨率则进行分数分配。

■

记为

$$\mathbf{x}^*(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)\triangleq\left((x^*_{u,l}(\lambda_{1,u,l}))_{u\in\mathcal{U},l\in\mathcal{L}},(x^*_{f,l}(\lambda_{1,f,l},\boldsymbol{\lambda}_2))_{f\in\mathcal{F},l\in\mathcal{L}}\right),$$

$$\mathbf{y}^*(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)\triangleq\left((y^*_{s,i,l}(\lambda_{1,i,l}))_{s\in\mathcal{S},i\in\mathcal{O}_s,l\in\mathcal{L}},(y^*_{f,k,l}(\lambda_{1,k,l},\boldsymbol{\lambda}_2))_{f\in\mathcal{F},k\in\mathcal{O}_f,l\in\mathcal{L}}\right).$$

Remark on Properties of Solutions: 问题5（问题6）最优解的坐标均为二进制值。对于问题4（问题7）的最优解，至多有一个坐标不是二进制值，其余坐标均为二进制值。问题4（问题7）、问题5和问题6的最优解分别在对应排列唯一、$\arg\max_{l'\in\mathcal{L}_u}(w_uQ_{l'}-\lambda_{1,u,l'})$ 为单元素集合，以及 $\sum_{k\in\mathcal{O}_f}\lambda_{2,f,k,l}-\lambda_{1,f,l}\neq 0$时唯一。实际应用中，这些唯一性条件大多成立。

问题3是一个目标函数非光滑的凸问题。我们使用投影次梯度法求解问题3 [30, 第 8 节2]。首先，我们推导 $g(\boldsymbol{\lambda}_1,\ \boldsymbol{\lambda}_2)$ 关于 $\boldsymbol{\lambda}_1$ 和 $\boldsymbol{\lambda}_2$的偏次梯度。

$$\partial_{\lambda_{1,i,l}}g(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)=\begin{cases}\sum_{j\in\mathcal{I}_i\cap\mathcal{S}}y^*_{j,i,l}(\lambda_{1,i,l})+\sum_{j\in\mathcal{I}_i\cap\mathcal{F}}y^*_{j,i,l}(\lambda_{1,i,l},\boldsymbol{\lambda}_2)-x^*_{i,l}(\lambda_{1,i,l}),\\ \quad i\in\mathcal{U},\\ \sum_{j\in\mathcal{I}_i\cap\mathcal{S}}y^*_{j,i,l}(\lambda_{1,i,l})+\sum_{j\in\mathcal{I}_i\cap\mathcal{F}}y^*_{j,i,l}(\lambda_{1,i,l},\boldsymbol{\lambda}_2)\\ \quad -x^*_{i,l}(\lambda_{1,i,l},\boldsymbol{\lambda}_2),i\in\mathcal{F}.\end{cases} \tag{18}$$

$$\partial_{\lambda_{2,f,k,l}}g(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)=y^*_{f,k,l}(\lambda_{1,k,l},\boldsymbol{\lambda}_2)-x^*_{f,l}(\lambda_{1,f,l},\boldsymbol{\lambda}_2). \tag{19}$$

接下来，我们介绍投影次梯度法。在第 t 次迭代中，我们并行计算式 (14)、(15)、(16) 和 (17) 中给出的最优解，然后按以下方式更新拉格朗日乘子 $\boldsymbol{\lambda}_1^{(t+1)}$ 和 $\boldsymbol{\lambda}_2^{(t+1)}$：

$$\lambda_{1,i,l}^{(t+1)} = \left[\lambda_{1,i,l}^{(t)} - \alpha^{(t)} \left(\partial_{\lambda_{1,i,l}} g(\boldsymbol{\lambda}_1^{(t)}, \boldsymbol{\lambda}_2^{(t)})\right)\right]^+, \quad (20)$$
$$i \in \mathcal{F} \cup \mathcal{U}, l \in \mathcal{L}$$
$$\lambda_{2,f,k,l}^{(t+1)} = \left[\lambda_{2,f,k,l}^{(t)} - \beta^{(t)} \left(\partial_{\lambda_{2,f,k,l}} g(\boldsymbol{\lambda}_1^{(t)}, \boldsymbol{\lambda}_2^{(t)})\right)\right]^+, \quad (21)$$
$$f \in \mathcal{F}, k \in \mathcal{O}_f, l \in \mathcal{L}$$

其中，$\alpha^{(t)} > 0$ 和 $\beta^{(t)} > 0$ 是满足平方可和但不可和条件的步长：

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \quad \sum_{t=0}^{\infty} \alpha_t^2 < \infty, \quad \sum_{t=0}^{\infty} \beta_t = \infty, \quad \sum_{t=0}^{\infty} \beta_t^2 < \infty.$$

*Th*收敛性分析 *A l i f Al i h 1)*

1) 算法1生成的序列 $\{(\boldsymbol{\lambda}_1^{(t)}, \boldsymbol{\lambda}_2^{(t)})\}_{t\in\mathbb{N}}$ 收敛到问题3的一个最优解 $(\boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*)$。2) 如果$\mathbf{x}^*(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ 和$\mathbf{y}^*(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ 唯一，那么它们也是问题 2的最优解。3) 如果$\mathbf{x}^*(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ 和$\mathbf{y}^*(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ 唯一且取整数值，那么它们也是问题 1 的最优解。

证明：由于$\partial_{\lambda_{1,i,l}} g(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ 和$\partial_{\lambda_{2,f,k,l}} g(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ 有界，因此，根据标准收敛结果[32, 命题 8.26]，步长平方可和但不可和的投影次梯度算法立即可得其收敛性。序列 $\{(\boldsymbol{\lambda}_1^{(t)}, \boldsymbol{\lambda}_2^{(t)})\}_{t\in\mathbb{N}}$ 收敛到问题3的一个最优解。 ■

记$\mathbf{x}^{(t)} \triangleq \mathbf{x}^*(\boldsymbol{\lambda}_1^{(t)}, \boldsymbol{\lambda}_2^{(t)})$ 和$\mathbf{y}^{(t)} \triangleq \mathbf{y}^*(\boldsymbol{\lambda}_1^{(t)}, \boldsymbol{\lambda}_2^{(t)})$。详细内容总结于算法1中。步骤3至步骤5计算所有子问题的最优解。步骤6至步骤7更新拉格朗日乘子。每次迭代的计算复杂度为 $\mathcal{O}(VL\log\ L + EL)$，因为 $L \to \infty$。

算法1 部分对偶分解算法**输入：** $\mathcal{U}$, $\mathcal{S}$, $\mathcal{F}$, B_l, Q_l, w_u 和$C_{i,\ j}$, $(i, j) \in \mathcal{E}$。**输出：** $\mathbf{x}^{(t)}$ 和$\mathbf{y}^{(t)}$。1：初始化 $\boldsymbol{\lambda}_1^{(0)}$, $\boldsymbol{\lambda}_2^{(0)}$ 和 $t \leftarrow 1$。2：**重复** 3：根据(14)、(15)、(16)和(17)计算$\mathbf{x}^{(t)}$ 和$\mathbf{y}^{(t)}$，均并行执行。4：分别根据(18)和(19)更新 $\boldsymbol{\lambda}_1^{(t)}$ 和 $\boldsymbol{\lambda}_2^{(t)}$，均并行执行。5：$t \leftarrow t + 1$。6：**直到**满足停止准则。

如果更新在有限次迭代 t后停止，或虽已收敛但最优解不唯一，则所得解可能不是整数解，也可能不可行。因此，我们提出一种两阶段贪心方法，以获得一个可行点，记为 $\tilde{\mathbf{x}} \triangleq (\tilde{x}_{i,l})_{i\in\mathcal{U}\cup\mathcal{F}, l\in\mathcal{L}}$ 和 $\tilde{\mathbf{y}} \triangleq (\tilde{y}_{i,j,l})_{(i,j)\in\mathcal{E}, l\in\mathcal{L}}$，其中$\mathbf{x}^{(t)}$ 和$\mathbf{y}^{(t)}$ 由算法1得到。具体步骤总结于算法 2。具体而言，我们首先将$\mathbf{y}^{(t)}$的分数值向下取整为整数 $\tilde{\mathbf{y}}$，以确保满足式（7）给出的容量约束（见步骤 2）。然后，我们按照深度非递减的顺序调整 $\tilde{\mathbf{x}}$ 和 $\tilde{\mathbf{y}}$ 的值，以满足式（5）和（6）给出的流量守恒约束（见步骤 3–12）。总体计算复杂度为 $\mathcal{O}((V + E)L)$。算法 2 中基于深度的排序有两个关键作用：

- **依赖解析：**基于深度的排序确保，具有多条路径的节点只有在所有依赖项都解析完毕后才会被处理。例如，考虑一个可以通过以下路径从服务器到达的节点：一条 2 跳路径和一条 3 跳路径。该节点的深度设为 3（即最长路径的长度），确保在处理该节点之前，两条路径上的所有中间节点都已处理完毕。这样可以保证，在确定任意节点 i 的 $\tilde{x}_i$时，来自上游节点的所有入站传输，$i,\ l$ 都已最终确定。
- **确定性执行：**该排序确保算法的执行具有确定性和一致性。深度为 d 的节点仅在深度为$d-1$ 的所有节点都完成处理后才会被处理。这可以避免执行顺序产生任何歧义，并确保不同实现之间的结果可复现。如果没有这种排序，算法可能会根据处理节点时任意采用的顺序产生不同的可行解，从而增加分析和调试的复杂性。

方法属性。所提出的优化框架是**分布式**的——每个转发器仅使用局部状态计算链路价格λ_l，确保系统的可扩展性不受全局网络规模或总流数量的影响。它**轻量**——算法仅在相邻转发器之间交换拉格朗日乘子。每条消息包含：I). 转发器 ID：4 字节；II). 迭代次数：4 字节；III). 乘子值：$L \times 8$ 字节。对于 12 个分辨率级别，每条消息约为 104 字节。这仅占同期传输的视频数据量的 <0.01 %（通常为数十 MB），证实协调开销可以忽略不计。它**本质上具有鲁棒性**——将丢包视为随机梯度噪声，使系统能够利用过时乘子自行纠正，而无需进行激进的控制平面重传（详见第 III-C 节）。它**快速**——无论用户数量多少，都能迅速收敛至全局最优解（$n = 50$，100，200，300），使控制回路能够在单个视频 GOP 的时间范围内有效跟踪观看者和信道波动。

算法 2 可行整数解构造

输入： $\mathbf{x}^{(t)}$ 和$\mathbf{y}^{(t)}$。
输出： $\tilde{\mathbf{x}}$、$\tilde{\mathbf{y}}$。
1：使用最长路径
算法计算有向无环图中的节点的基于深度的排序 [33]。
2: 初始化 $\tilde{\mathbf{y}} \leftarrow \lfloor \mathbf{y}^{(t)} \rfloor$
3: **对**每个深度按非递减顺序排列的节点 i**执行**
4: **如果** $i \in \mathcal{F}$ **则**
5: $\tilde{x}_{i,l} \leftarrow$ 最小值$\{1, \sum_{j\in\mathcal{I}_i} \tilde{y}_{j,i,l}\}$, $l \in \mathcal{L}$ 并行执行。
6: $\tilde{y}_{i,k,l} \leftarrow$ 最小值$\{\tilde{y}_{i,k,l}, \tilde{x}_{i,l}\}$, $k \in \mathcal{O}_i$, $l \in \mathcal{L}$ 并行执行。
7: **否则，如果** $i \in \mathcal{U}$ **则**
8: $\mathcal{L}_i \leftarrow \{l \in \mathcal{L}_i : \sum_{j\in\mathcal{I}_i} \tilde{y}_{j,\ i,\ l} > 0\}$。
9: **if** $\tilde{\mathcal{L}}_i \neq \varnothing$ **然后**
10: $\tilde{x}_{i,\ l} \leftarrow \begin{cases} 1, & \text{如果 } l = \arg\max_{l'\in\tilde{\mathcal{L}}_i} Q_{l'}, \\ 0, & \text{否则} \end{cases} \quad l \in \mathcal{L}_i$。
11: **else**
12: $\tilde{x}_{i,\ l} \leftarrow 0$, $l \in \mathcal{L}_i$。
13: **结束 if**
14: **结束 if**
15: **结束 for**

VI. 实验评估

本节将 COMETS 与最先进的视频流媒体方法进行评估比较。我们的评估聚焦于三个属性：I). 可扩展性——随着并发用户增加，体验质量是否保持稳定；II). 公平性——分布式优化是否能在异构客户端之间实现公平的资源分配；以及 III). 收敛效果——该算法能否比集中式方法更快地做出最优决策。

A. 实验设置

尽管第 V 节为这些性质提供了理论保证——定理 1 证明了收敛性，且每次迭代的复杂度确保了线性扩展——但这些结果假设了理想化条件，即即时消息传递和完美同步。要验证这些性质在实际协议实现、真实视频内容和系统动态下是否仍然成立，需要借助仿真平台。Mini-NDN [3] 是基于 Mininet [4] 构建的命名数据网络仿真器，适用于此目的：它实现了在现实条件下测试范围兴趣聚合所需的完整命名数据网络转发平面（待处理兴趣表、网内缓存、有状态转发），提供了可控环境，以便公平地进行跨系统比较，并记录墙钟时间，反映真实的计算开销和通信开销。Mini-NDN 使我们能够通过逐步增加并发用户和基础设施节点的数量，系统地扩大实验规模，从而直接观察体验质量、公平性和优化时间如何随网络规模变化。

所有仿真均在一台搭载 Intel Xeon Platinum 8358P 中央处理器（2.60GHz，64 核）和 32GB 随机存取存储器、运行 Ubuntu 22.04.5 LTS 的服务器上进行。该配置支持最多 300 个客户端同时运行实验，且不会影响系统的实时行为。虽然我们也考虑过使用更多用户进行测试（例如，$\geqslant 1000$），但我们关于可扩展性的论断有两方面证据支撑。首先，我们的分布式算法在分析复杂度上保证了线性增长和扩展。其次，最多 300 个用户的实证结果显示性能趋势稳定，且没有出现可扩展性瓶颈即将到来的迹象，这表明主要限制因素是仿真服务器的容量，而非 COMETS 架构本身。客户端与初始转发器之间的连接带宽配置为 100 Mbps，骨干链路为 400 Mbps，节点间延迟低于 5 ms（参见 [34], [35]）。

我们将 COMETS 与四种基线系统进行比较，它们代表了引言中讨论的主要方法：I). *DASH-BOLA* [36]：标准的客户端基于缓冲区的自适应算法（客户端自适应方法）；II). 基于 *MoQ* 的方法 [9]：利用全局带宽信息实现以服务器为中心的优化（服务器驱动方法）；III). *GB* *[19]*：一种基于博弈论的 NDN 方法，使用贝叶斯纳什均衡进行带宽分配；IV).*NDN-MMRA* *[13]*：一种基于 NDN 的组播媒体速率自适应算法（

[3]https://minindn.memphis.edu/ [4] https://mininet.org/

算法（这两种均为基于 ICN 的方法）。表 III 展示了与这些方法的比较。为确保公平比较，所有系统使用了相同的源视频内容。对于 DASH 和基于 MoQ 的方法，FFmpeg 流被分段后分别通过 HTTP 和管道传输；对于基于 ICN 的方法，FFmpeg 流被分割为时长为 2 秒的数据块。

表 III：与代表性方法的比较

方法	协调	优化	转换	规模
DASH-BOLA	None	局部式 全局整数线性规划	N/A	线性
MoQ-Server	集中式		Slow	有限
NDN-MMRA	局部式	启发式	N/A	次线性
GB	Game	BNE	Slow	次线性
COMETS	分布式	拉格朗日	Fast	次线性

Coord.：协调机制；Optim.：优化方法；Conv.：收敛速度；BNE：贝叶斯纳什均衡。

B. 评估指标

我们使用评估视频质量、播放连续性、公平性和总体体验质量的指标，将 COMETS 与基线系统进行比较。为同时衡量平均性能和一致性，我们在报告关键指标的均值时，也提供第95百分位数统计数据，重点关注可扩展性以及改善尾部用户体验。

视频质量：我们采用 VMAF（视频多方法评估融合）[37]来量化感知质量。为便于分析，我们按分辨率设定基线 VMAF 范围：480p（50-75）、1K（65-85）、2K（70-85）、4K（80-95）和 8K（95-98）。我们跟踪 VMAF 的均值和第95百分位数，以评估质量一致性。

抖动：我们通过两项指标评估播放平滑度：分块到达间隔时间：连续分块到达之间的时间方差，并根据请求频率进行调整；以及*RFC 3550* 抖动 *[38]*：使用 $J(i) = J(i-1) + (|D(i,\ j)| - J(i-1))/16$, 计算，其中 D 表示实际间隔与预期间隔之间的偏差。

播放缓冲时间： 模拟客户端缓冲区在分块到达时增加一个分块时长（2.0 秒），并在播放期间逐渐耗尽，以此表明系统能够保持连续播放。

启动延迟： 播放开始前的初始等待时间（缓冲区达到阈值时开始播放），反映了受初始分辨率分配、网络延迟和优化计算时间影响的用户体验。

Jain公平性指数 [39]： 我们根据各客户端的平均交付分辨率计算Jain公平性指数，以评估资源分配公平性。公式 $f(x) = (\sum x_i)^2/(n \cdot \sum x_i^2)$ 的取值范围为 1/n 至 1，其中 x_i 表示客户端 i的平均交付分辨率， n表示客户端总数。取值为 1 表示完全公平。

总体QoE评分： 我们通过加权归一化构建综合评分：

$$QoE = \text{最大}(0, \alpha \cdot normalized_VMAF + \beta \cdot normalized_fairness + \gamma \cdot normalized_buffer + \delta \cdot (1 - normalized_jitter) + \epsilon \cdot (1 - normalized_startup_delay))$$

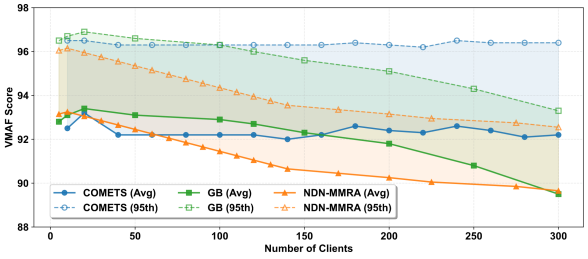
权重 $[\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon] = [0.4,\ 0.2,\ 0.15,\ 0.15,\ 0.1]$ 优先考虑

视频质量和公平性，同时惩罚重缓冲、抖动和启动延迟。[5]

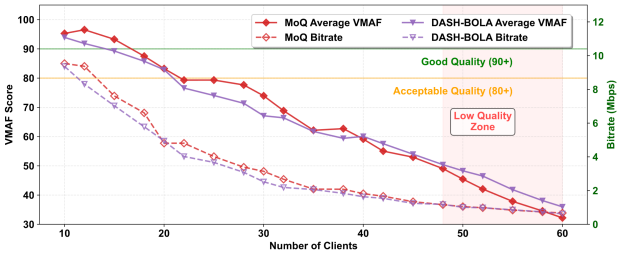
C. 比较性能评估

本小节比较 COMETS、DASH-BOLA、MoQ、GB 和 NDN-MMRA 在并发用户数不断增加时的表现，重点关注可扩展性和体验质量指标。我们预计 COMETS 在高并发场景下表现出色，维持稳定的体验质量，而其他方案则会逐渐退化。

1) 视频质量（*VMAF*）分析：图 4 展示了各系统截然不同的性能表现。在 10 至 300 个客户端的整个范围内，COMETS 始终保持卓越的稳定性，平均 VMAF 分数为 92.2，第95百分位数为 96.4，即使负载达到最大值，性能也几乎没有下降。GB算法的表现则有所退化，分数从 93.4 降至 89.5，这归因于其贝叶斯纳什均衡（BNE）计算中的信息不对称——不完整的先验信念导致其在高并发情况下做出次优码率决策。随着规模扩大，NDN-MMRA 的 PIT 聚合效率下降，表现出类似的退化（93.25 降至 89.65）。DASH-BOLA 和 MoQ 在并发用户超过 60 后均出现灾难性的质量崩溃，VMAF 跌至 40 以下，远低于 80 的可接受质量阈值；因此，鉴于它们无法维持可接受的质量，没必要继续测试。COMETS 通过分布式分辨率选择和兴趣聚合，以分层方式优化带宽，从而确保即使规模扩大也能保持稳定的质量。



(a) 基于 NDN 的系统的 VMAF 分数



(b) MoQ 和 DASH-BOLA 的 VMAF 分数

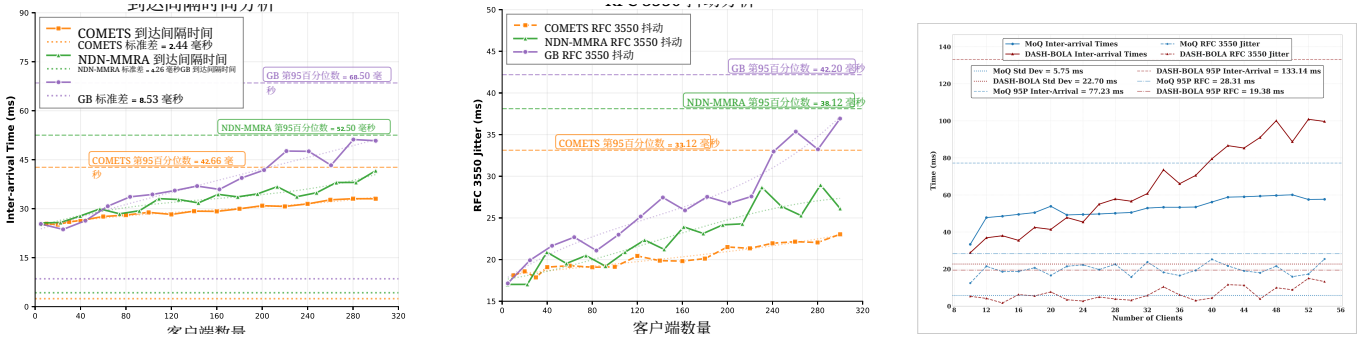
图 4：VMAF 分数比较

2) 抖动与播放稳定性：图 5a 和 5b 展示了 COMETS 的低抖动：即使客户端数量达到 300，到达间隔时间也稳定在 30ms，RFC 3550 抖动则低于 23ms。相比之下，GB 的抖动方差不断增大（第95百分位数：68.50ms），这是由于异构客户端先验在非最优均衡下破坏了 PIT聚合，导致周期性兴趣泛洪；而 NDN-MMRA 通过多阶段速率自适应机制将抖动维持在中等水平（第95百分位数：52.50ms）。随着客户端数量从 5 增至 56，DASH-BOLA 和 MoQ 的抖动均有所上升（图 5c）。随着客户端竞争加剧，DASH-BOLA 的到达间隔时间从 30ms 大幅升至 100ms 以上，而其 RFC 3550 抖动仍保持较低水平（第95百分位数：19.38ms），这得益于其基于缓冲区的算法固有的平滑作用。与 DASH 相比，MoQ 的到达间隔时间更稳定（50–60ms），但 RFC 3550 抖动更高（第95百分位数：28.31ms），反映了服务器端优化带来的计算开销。缓冲区动态（图 6a）表明，所有基于 NDN 的系统都能将播放缓冲区维持在 8 秒阈值以上。COMETS 在整个测试过程中保持 8.5–10.5 秒，GB 和 NDN-MMRA 也呈现出相近的模式。NDN变体之间的一致性证明了网内缓存在保障缓冲区稳定性方面的有效性。

3) 资源公平性与启动延迟：图6b显示，COMETS 的 Jain公平性指数稳定在 0.98–0.93，表明全网协调实现了公平的解析分配。NDN-MMRA 通过待处理兴趣表驱动的子集优化优先照顾弱势客户端，公平性指数保持在 0.98–0.94；而 GB 则从 0.98 降至 0.89，这是因为信息不对称下其收益函数会惩罚公平性。MoQ 和 DASH-BOLA 的指数从 0.98 降至 0.87–0.90，凸显了以服务器为中心和由客户端驱动的资源分配效率低下。所有系统的指数均高于 0.86，但 COMETS 的网内聚合确保了高负载下更优的公平性。就启动延迟而言（图6(b)），COMETS 的启动延迟保持在 2.3 秒以下，仅缓慢上升，这得益于高效的初始分配。NDN-MMRA 和 GB 的启动延迟从 1.5–1.7 秒增至 2.6–2.8 秒，受到初始探测和均衡求解开销的影响。MoQ 和 DASH-BOLA 则由于客户端信息和带宽探测，在规模扩大时从 0.9–1.3 秒增至 1.8 秒。所有系统的启动延迟均可保持在 3秒卓越阈值以内。

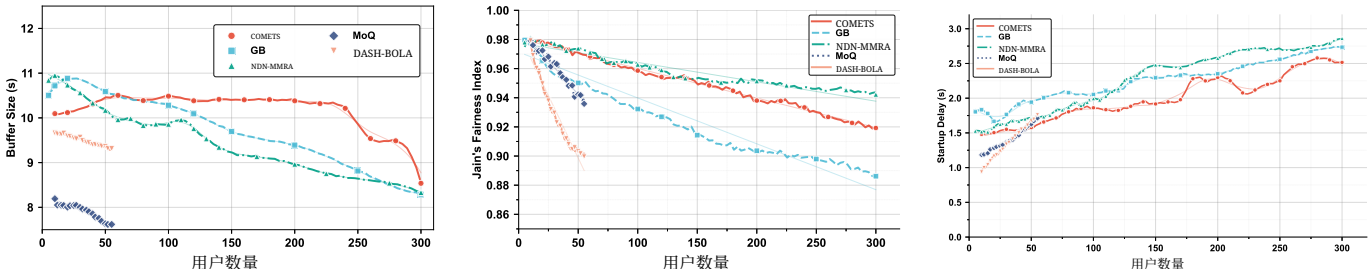
4) 算法比较：图7展示了分布式与集中式优化架构在计算方面的根本优势。随着并发用户数从 10 增至 300，COMETS 的优化时间近乎线性增长，从 5 毫秒增至 47.58 毫秒，即使在最大规模下也能将延迟维持在 50 毫秒以内。相比之下，基于 MoQ 的方法呈现出更陡的线性增长，从 10 毫秒增至 178 毫秒，反映了集中式全局优化固有的开销。性能比（下方面板）量化了这一效率差距：客户端数量较少时，两者性能接近，但随着规模增大，差距显著拉开。在整个测试过程中，COMETS 始终保持稳定优势，在 300 名用户时性能提升达到 3.7×。这一差距不断扩大，源于两者的架构差异：MoQ 必须求解规模随决策变量数量呈二次增长、日益复杂的全局优化问题；而 COMETS 将解析决策分配到各网络节点并行处理，从而将计算复杂度限制在局部子问题范围内。其实际影响不止于原始计算效率。在 300 名用户时，MoQ 的 178 毫秒优化延迟已接近关键阈值——进一步扩

[5]这些权重并非任意设定，而是基于既有 QoE 敏感性分析研究的见解 [40],，这些研究已验证它们能够代表大多数常见流媒体场景中的用户偏好。



(a) 基于 NDN 的系统：到达间隔时间　(b) 基于 NDN 的系统：RFC 3550 抖动　(c) MoQ 与 DASH-BOLA 的抖动对比

图5：抖动性能比较：（a）和（b）展示了基于 NDN 的系统的两项抖动指标，而（c）比较了 MoQ 与 DASH-BOLA 的抖动性能。



（a）缓冲性能比较　（b）资源公平性比较　（c）启动延迟比较

图6：缓冲区、启动延迟和公平性比较

展规模可能会损害对网络动态的响应能力。相比之下，COMETS 的 47.58 毫秒延迟为扩展规模留出了充足空间，同时能够维持自适应流媒体所必需的低于 100 毫秒的响应时间。这种计算效率直接促成了第 VI-C 节中观察到的卓越 QoE 稳定性，尤其体现在稳定的启动延迟和极小的抖动上，这些都是 COMETS 在大规模场景下性能的特点。

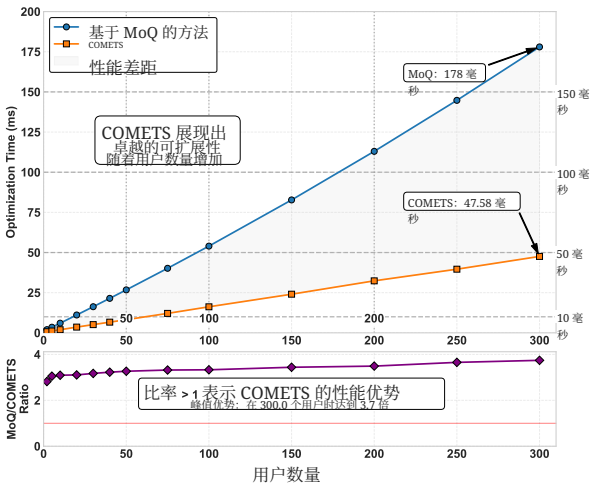
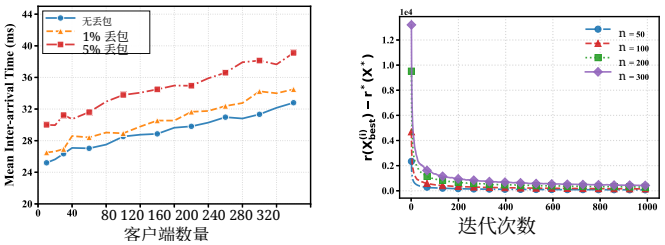


图7：MoQ 与 COMETS 的优化时间

5) 鲁棒性分析： 为验证 COMETS 在现实网络损伤下的韧性，我们在瓶颈链路上引入了 1% 和 5% 的随机丢包率。这些丢包率分别代表有线网络中的典型拥塞场景，以及无线/移动环境中的严苛条件 [41]。如图8a所示，COMETS 在这两种丢包场景下均能保持稳定的播放性能。在 1% 的丢包率下，与基线相比，平均到达间隔时间增加约 5%（在 300 个客户端时从 32.8ms 增至 34.5ms）。在 5% 的丢包率下，到达间隔时间增加约 19%（达到 39.1 ms），仍远低于可接受的流媒体阈值。该系统表现出优雅降级，而非性能发散，展现了其在所有测试用户规模下固有的鲁棒性。这种韧性源于 COMETS 的丢包恢复机制（第 III-C 节）。与 TCP/QUIC 中的端到端重传不同，后者的恢复时延随到源服务器的往返时间增加；COMETS 利用分层网络内缓存。当客户端通过超时检测到丢包时，会重新表达兴趣包，并进行有限次数的重传（最多 2 次）。由于热门流具有较高的冗余度，这些重新表达的兴趣包通常由附近的上游转发器满足，而非远处的源服务器。

为量化这一效应，我们分析了整个拓扑中的缓存命中分布（50 个转发器、300 个客户端）。第三级节点（最靠近用户的边缘转发器）的有效命中率约为 75–80%，在局部吸收了大部分重传请求。第一级节点（更靠近源的核心转发器）的命中率维持在 50–60% 左右，可过滤剩余的跨区域流量。这种多级聚合能够有效地将物理层丢包屏蔽在应用层之外，把单播请求洪泛转化为高效的共享传输树。

图 8b 进一步证实了算法在不同问题规模下的稳定性。无论用户数量如何变化（$n = 50$，100、200、300），分布式优化都能快速收敛至全局最优解。这种快速收敛确保控制回路能够在单个视频 GOP 的时间范围内跟踪动态网络波动，即使底层条件发生变化，也能维持系统稳定性。



(a) 丢包行为　　(b) 收敛行为

图 8：优化算法性能分析。

6）总体体验质量：综合QoE指标（图9）证明了 COMETS 的可扩展性优势。在整个测试范围内，COMETS 将体验质量评分维持在 0.7–0.9 之间，通过分布式优化平衡高 VMAF、低抖动和稳定的缓冲区。NDN-MMRA 的评分从 0.9 降至 300 个客户端时的 0.61，受限于基于PIT的自适应延迟，以及高客户端负载和动态网络状况下不断累积的多播阶段开销。GB 则呈现出更陡的下降趋势，从 0.9 降至 0.5。博弈论方法的性能下降反映了信息不对称的累积影响——随着客户端异质性增加，次优 BNE解的影响会不断叠加。传统方法（MoQ 和 DASH-BOLA）在客户端数量超过 50 时基本无法使用，体验质量因集中式扩展问题而骤降至 0.3 以下。这些结果验证了 COMETS 的架构设计：通过将信息中心网络的固有优势（兴趣聚合、网内缓存）与分布式智能相结合，该系统在规模扩大的情况下仍能实现稳健的体验质量，避免性能急剧下降。

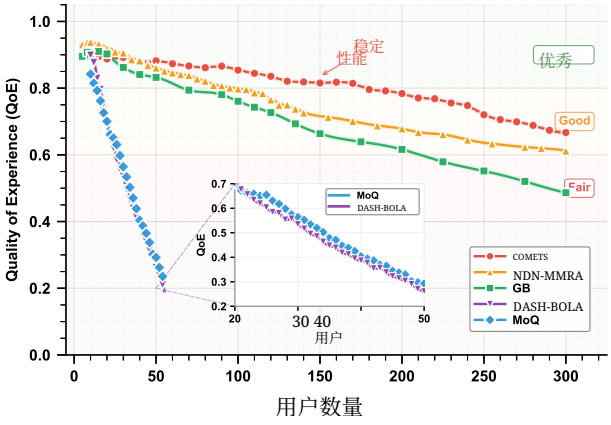


图 9：体验质量比较

VII. 讨论

除了分析和实验性能评估外，我们还讨论了系统的假设、部署方面以及 COMETS 的技术新颖性。

信任模型与部署范围。COMETS 假设转发器在共同信任锚下的单一管理域内运行。这一模型与内容分发网络部署（例如 Akamai、CloudFlare）相符，在这类部署中，边缘服务器由内容提供商或可信合作伙伴运营。然而，COMETS 并不直接适用于对等（P2P）或联邦场景，因为其中的转发器可能不可信或存在利益冲突。将信任模型扩展至多管理域环境，并采用密码学验证或信誉系统，是未来值得探索的方向。

体验质量评估范围。我们的评估采用了先前研究中经过验证、可用于评估主观质量的客观指标。开展一项涵盖不同人口统计特征的全面主观用户研究，可以提供额外验证；这将留待未来工作完成。

内容源假设。当前设计面向单源视频点播（*VoD*）和近实时直播，其内容来自已知服务器，并预先编码为多个质量级别。对于采用实时编码的真正直播流，还会面临其他挑战，包括：I）编码延迟。源必须实时生成多个质量版本，这可能导致编码器成为瓶颈。II）多源场景。用户生成内容（例如 Twitch）涉及许多并发源，需要承担逐流优化的开销。

技术新颖性。I). 新问题建模。虽然以往研究已将优化应用于视频流媒体，但 COMETS 首次将请求聚合支持的内容组播与逐跳质量自适应统一建模为一个整数线性规划问题，并通过两阶段分布式优化应对其 NP 难性：首先采用基于对偶分解的分布式算法，根据局部信息并行执行闭式更新，精确求解连续松弛；随后采用低复杂度贪心算法，得出近乎最优的整数可行解。以往的算法大多使用求解器或启发式算法，存在复杂度高、运行时间长等诸多问题。**II). 范围兴趣协议**。该协议将客户端与转发器之间的交互从“请求-响应”转变为“能力通告”，使转发器能够基于充分信息做出组级决策。这与以往的信息中心网络兴趣语义有着根本区别。**III). 架构集成**。虽然信息中心网络缓存和请求聚合是已知技术，但将它们与用于协同码率自适应的分布式优化相结合则具有新颖性。以往的信息中心网络系统利用聚合来提高效率，却并非用于协同——它们仍然在每一跳独立、被动地做出决策。

八、结论

在这项工作中，我们介绍了**COMETS**，这是首个将网内聚合与分布式优化

相结合的框架，用于解决现有流媒体架构在可扩展性和公平性方面的局限。与客户端驱动的 DASH 或服务器驱动的 MoQ 不同，COMETS 将自适应转变为协作式的逐跳协商，从根本上将流媒体从孤立的端点控制转变为协调式网内智能。

研究成果。我们的结果表明，在数百名并发用户的情况下，COMETS 始终能提供高体验质量；与最先进的基线方案相比，它实现了更高的带宽效率、稳定的播放表现和近乎完美的公平性。这得益于其架构创新：(i) 对接收端进行分组并统一分辨率，从而消除冗余流；(ii) 将适配逻辑交由转发器处理，以消除中心瓶颈；(iii) 提供可部署的覆盖网络，可与当前的内容分发网络和 MoQ 基础设施无缝集成。

优化亮点。 COMETS 的一项关键创新，在于将协调式适配形式化为统一的整数线性规划问题。通过采用对偶分解、闭式更新以及轻量级的贪心可行性重构，COMETS 能够以低于 50 毫秒的收敛时间，在大规模场景下实现近乎最优的分配决策——速度最高可达集中式求解器的 3.7× 倍。这种理论严谨性 与实际可扩展性 的结合，使 COMETS 有别于以往仅依赖局部启发式方法或响应式拥塞信号的基于 ICN 的方法。

特性。 这些贡献共同赋予 COMETS 三项鲜明特性：(1) 可扩展协调，通过按网络规模成比例地分布计算任务实现；(2) 公平且高效的资源使用，通过协调各组的质量实现；(3) 便于实际部署，无需更改底层互联网架构。

更广泛的影响。 通过与 MoQ 等新兴传输计划以及内容分发网络覆盖层的发展保持一致，COMETS 为迈向下一代媒体系统提供了切实可行的迁移路径。我们认为，将受信息中心网络启发的原语与可扩展优化相结合，为未来大规模、低延迟流媒体和实时应用的发展指明了一个有前景的方向。

未来的工作将把 COMETS 扩展到异构边缘环境，开展网络层与传输层之间的跨层优化，并将其集成到真实部署中，以进一步验证其稳健性和适用性。

致谢

本研究部分得到了广东省项目（资助编号：2023QN10X048）、广州市未来网络系统重点实验室（资助编号：2024A03J0623）、广东省泛在物联网通信、感知与计算融合重点实验室（资助编号：2023B1212010007）、广州市科技计划项目（资助编号：2023A03J0011）、广东省项目（资助编号：2023ZT10X009）、国家自然科学基金（资助编号：U23A20339）、芬兰商业促进局6GSoft项目（资助编号：8541/31/2022）以及中国国家重点研发计划（资助编号：2024YFE0200603）的部分资助，中国国家移动信息网络科技重大专项（资助编号：2024ZD1300400）、国家自然科学基金委员会（资助编号：62371412）以及广东省基础与应用基础研究基金自然科学基金项目（资助编号：2024A1515011184）也为本研究提供了资助。作者还要感谢香港科技大学（广州）的Yiqing Zhai博士、Yu Zhang博士和Bowen Tan博士，他们在核验和完善优化设计方面给予了宝贵帮助。

参考文献

[1] 普华永道，《流媒体赛事：数字平台的兴起如何改变体育消费》，https://www.pwc.com/us/en/industries/tmt/library/sports-streaming-platforms.html，2024年，访问时间：2025年4月。[2] Ross Video，《体育观众统计数据：球迷行为变化背后的数据》，https://www.rossvideo.com/blog/sports-audience-statistics-the-data-behind-changing-fan-behavior/，2025年2月，访问时间：2025年4月。[3] Y. Wang、Y. Zhu、X. Li、Y. Zhang、R. Wu 和 D. Kutscher，《Vifusion：用于可扩展视频特征索引的网内张量融合》，载于*2025*年国际多媒体检索会议论文集，2025年，第1452–1460页。[4] Y. Zhang、D. Kutscher 和 Y. Cui，《网络化元宇宙系统：基础、差距与研究方向》，*IEEE*《通信学会开放期刊》，2024年。[5] Z. Li 和 X. Zhu，《探测并自适应：大规模HTTP视频流的码率自适应》，*IEEE*《通信领域精选期刊》，第32卷，第4期，第719–733页，2014年。[6] C. Zhou、C.-W. Lin 和 Z. Guo，《MDASH：一种用于动态HTTP流媒体的基于马尔可夫决策的码率自适应方法》，*IEEE*多媒体汇刊，第18卷，第4期，第738–751页，2016年。[7] K. Spiteri、R. Sitaraman 和 D. Sparacio，《从理论到实践：改进DASH参考播放器中的比特率自适应》，《*ACM*多媒体计算、通信与应用汇刊》（*TOMM*），第15卷，第2s期，第1–29页，2019年。[8] S. Altamimi 和 S. Shirmohammadi，《利用服务器端强化学习实现QoE公平的DASH视频流》，《*ACM*多媒体计算、通信与应用汇刊》（*TOMM*），第16卷，第2s期，第1–21页，2020年。[9] M. H. Dwijaksara 和 M. H. Hilman，《基于QUIC的媒体流有限回程链路码率自适应技术》，*IEEE*《*Access*》，第12卷，第139 028–139 041页，2024年。[10] R. Chai、Y. Zhu、X. Li、J. Li、Z. Meng 和 D. Kutscher，《INDS：面向实时点云视频的增量式命名数据流》，载于第*33*届*ACM*国际多媒体会议论文集，2025年，第12 102–12 110页。[11] Z. Liu 和 Y. Wei，《内容中心网络中的逐跳自适应视频流》，载于*2016*年*IEEE*国际通信会议（*ICC*）。IEEE，2016年，第1–7页。[12] J. Samain 和 G. Carofiglio，《动态自适应视频流：迈向信息中心网络与TCP/IP的系统性比较》，*IEEE*多媒体汇刊，第19卷，第10期，第2166–2181页，2017年。[13] F. Wu、W. Yang、J. Ren、F. Lyu、P. Yang、Y. Zhang 和 X. Shen，《NDN-MMRA：命名数据网络无线局域网中的多阶段组播码率自适应》，*IEEE*多媒体汇刊，第23卷，第3250–3263页，2021年。[14] J. Saltarin、E. Bourtsoulatze、N. Thomos 和 T. Braun，《采用网络编码的命名数据网络自适应视频流》，*IEEE*多媒体汇刊，第19卷，第10期，第2182–2196页，2017年。[15] B. Alt 和 T. Ballard，《CBA：自适应比特率视频流的上下文感知质量自适应》，载于*IEEE INFOCOM 2019—IEEE*计算机通信会议。IEEE，2019年，第1000–1008页。[16] A.-J. Su 和 Choffnes，《借助Akamai顺势而为：根据CDN重定向推断网络状况》，*IEEE/ACM*网络汇刊，第17卷，第6期，第1752–1765页，2009年。[17] B. Shobiri、M. Mannan 和 A. Youssef，《CDN的阴暗面：CDN到源站连接中的安全问题》，《数字威胁：研究与实践》，第4卷，第1期，第1–22页，2023年。[18] A.-M. K. Pathan、R. Buyya等，《内容分发网络的分类与综述》，墨尔本大学网格计算与分布式系统实验室技术报告，第4卷，第2007期，第70页，2007年。

[19] X. Tan 和 L. Xu，《基于博弈论的命名数据网络多客户端动态自适应视频流》，IEEE多媒体汇刊，第24卷，第3491–3505页，2021年。[20] E. Thomas、M. van Deventer、T. Stockhammer、M.-L. Begen 和 O. Oyman，《服务器与网络辅助的DASH（SAND）的应用与部署》，载于IBC 2016会议。IET，2016年，第22页。[21] A. Bentaleb 和 Akcay，《Bob：利用启发式方法和强化学习预测实时通信带宽》，IEEE多媒体汇刊，第25卷，第6930–6945页，2022年。[22] Z. Gurel、T. Erkilic Civelek、A. Bodur、S. Bilgin、D. Yeniceri 和 A. C. Begen，《基于QUIC的媒体传输：初步测试、发现与结果》，载于第14届ACM多媒体系统会议论文集，2023年，第301–306页。[23] H. K. Ravuri、M. T. Vega、J. Der Van Hooft、T. Wauters 和 F. De Turck，《通过QUIC-HTTP/3对沉浸式媒体进行自适应部分可靠传输》，IEEE《Access》，第11卷，第38 094–38 111页，2023年。[24] W. Geng、Y. Zhang、D. Kutscher、A. Kumar、S. Tarkoma 和 P. Hui，《ICN中的分布式计算：现状综述》，载于第10届ACM信息中心网络会议论文集，2023年，第88–100页。[25] M. Bagheri 和 A. Movaghar，《带宽自适应分层多播覆盖网络》，载于2010年第五届全球信息技术计算国际多会议。IEEE，2010年，第262–267页。[26] T.-M. Kwan 和 Yeung，《覆盖多播树的构建与维护》，载于2005年协作计算：网络、应用与协作国际会议。IEEE，2005年，第6页。[27] T. Yu、X. Ma、V. Patil、Y. Kocaogullar、Y. Zhang、J. Burke、D. Kutscher 和 L. Zhang，《安全网络对象：构建元宇宙互操作性与去中心化的基石》，载于2024年IEEE元宇宙计算、网络与应用国际会议（MetaCom）。IEEE，2024年，第25–33页。[28] C. Tschudin、C. A. Wood、M. Mosko 和 D. R. Oran，《类文件信息中心网络集合（FLIC）》，互联网工程任务组，互联网草案 draft-irtf-icnrg-flic-06，进行中。 [在线]。可访问：https://datatracker.ietf.org/doc/draft-irtf-icnrg-flic/06/ [29] A. Raake、M.-N. Garcia、W. Robitza、P. List、S. Göring 和 B. Feiten，《用于HTTP自适应流媒体的基于比特流、可扩展的视频质量模型：ITU-T P. 1203.1》，载于2017年第九届国际多媒体体验质量会议（QoMEX）。IEEE，2017年，第1–6页。[30] S. P. Boyd 和 L. Vandenberghe，《凸优化》。剑桥大学出版社，2004年。[31] S. Boyd 和 J. Xiao，《分解方法笔记》，斯坦福大学EE364B课程笔记，第635卷，第1–36页，2007年。[32] D. Bertsekas、A. Nedic 和 A. Ozdaglar，《凸分析与优化》。Athena Scientific，2003年，第1卷。[33] T. H. Cormen、C. E. Leiserson、R. L. Rivest 和 C. Stein，《算法导论》。MIT出版社，2022年。[34] S. Awiphan 和 K. Poobai，《基于物联网辅助内容分发的命名数据网络自适应视频流》，载于第22届国际计算机科学与工程会议，2018年，第1–4页。[35] W. Yang、F. Wu 和 K. Tian，《利用命名数据网络无线局域网多播实现高性能自适应视频流》，载于第8届ACM信息中心网络会议论文集，2021年，第42–51页。[36] K. Spiteri、R. Urgaonkar 和 R. K. Sitaraman，《BOLA：在线高清视频的近最优比特率自适应》，IEEE/ACM网络汇刊，第28卷，第4期，第1698–1711页，2020年。[37] B. García、L. López-Fernández、F. Gortázar 和 M. Gallego，《VMAF感知视频质量在WebRTC应用中的实际评估》，Electronics，第8卷，第8期，第854页，2019年。[38] H. Schulzrinne、S. Casner、R. Frederick 和 V. Jacobson，《RFC3550：RTP：实时应用传输协议》，2003年。[39] R. Ware 和 Mukerjee，《超越Jain公平性指数：为部署拥塞控制算法设定标准》，载于第18届ACM网络热点主题研讨会论文集，2019年，第17–24页。[40] X. Yin、A. Jindal、V. Sekar 和 B. Sinopoli，《基于控制理论的方法：通过HTTP实现动态自适应视频流》，载于2015年ACM数据通信特别兴趣组会议论文集，2015年，第325–338页。[41] 思科系统，《服务质量设计概述》，https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_ 和_MAN/QoS_SRND_40/QoSIntro_40.html，2024年。



Yulong Zhang 于2018年获得中国武汉理工大学工学学士学位。目前，他正在香港科技大学（广州）攻读博士学位。他的主要研究兴趣包括新兴网络系统的设计与优化、分布式计算与网络（计算优先网络、网络内计算），以及互联网架构与去中心化通信。



Ying Cui 于2007年获得中国西安交通大学电子与信息工程专业工学学士学位，并于2012年获得香港科技大学博士学位。她曾于2011年在美国耶鲁大学、2012年在澳大利亚麦考瑞大学担任访问职务。2012年6月至2013年6月，她在美国东北大学担任博士后研究助理。2013年7月至2014年12月，她在马萨诸塞州理工学院担任博士后研究助理，美国。从2015年1月至2022年7月，她在中国上海交通大学担任副教授。自2022年8月起，她担任香港科技大学（广州）物联网学域副教授。她目前的研究兴趣包括优化、学习、物联网通信、感知与通信一体化以及边缘智能。她于2014年入选国家青年人才计划。她曾获 IEEE ICC 2015、IEEE GLOBECOM 2021 和 IEEE GLOBECOM 2025 最佳论文奖。她担任《IEEE通信汇刊》编委，并担任《IEEE无线通信汇刊》执行编委。她曾于2018年至2024年担任《IEEE无线通信汇刊》编委。



Zili Meng （S’18-M’23）获清华大学工学学士学位和博士学位。他目前是香港科技大学电子与计算机工程学系助理教授。其研究兴趣包括多媒体流传输和网络。



Abhishek Kumar （IEEE会员及IEEE信号处理学会会员）现任芬兰于韦斯屈莱大学信息技术学院高级讲师。他分别于2014年、2016年和2023年从巴特那国家理工学院、韩国科学技术院和赫尔辛基大学获得工学学士、理学硕士和博士学位。他还担任IEEE芬兰信号处理与电路和系统分会秘书（2025年1月至2026年12月）。目前的研究领域包括边缘智能，意图驱动的神经符号人工智能，以及信息物理系统，尤其关注隐私和安全方面。



Dirk Kutscher 获得了德国不来梅大学的博士学位。他是香港科技大学（广州）的教授，并担任未来网络系统实验室主任。此前，他曾担任华为德国研究院虚拟网络首席技术官、NEC欧洲首席网络研究员，以及日本KDDI研发中心访问研究员。他目前担任互联网研究任务组（IRTF）主席。他撰写了多项国际专利、IETF RFC文档，书籍，以及广受引用并获奖的研究论文。他的主要兴趣在于分布式计算与网络在计算优先网络中的交叉领域，以及互联网架构。

附录 A 分布式优化方法的理论依据

公式（1）中构建的问题是一个整数线性规划，属于 NP 难问题。为应对其固有挑战，尤其是在大规模、动态网络环境中，我们在阐述选择方案的依据之前，先分析了几种潜在的解决范式。

A. 集中式整数线性规划求解器

商业求解器（例如 Gurobi、CPLEX）可以找到全局最优解。然而，它们面临两个关键限制：

1) 指数级复杂度：随着整数变量数量的增加，求解时间呈指数增长。对于具有 V 个节点和 L 个分辨率级别的网络，我们有 $\mathcal{O}(VL)$个节点选择变量和 $\mathcal{O}(EL)$ 个传输变量，这使得该问题对于拥有数百个节点的网络而言难以处理。

2) 全局信息需求：它们需要一个集中式控制器，完整且即时地掌握整个网络拓扑、链路容量以及所有用户状态的信息，这违背了 NDN 的分布式特性，并带来了显著的信令开销。

B. 基于人工智能的方法

近期的进展已将机器学习，尤其是深度神经网络（DNN），应用于求解组合优化问题。然而，由于以下原因，这些方法并不适用于我们的问题：

1)泛化与鲁棒性问题：训练好的模型仅在统计分布（例如网络拓扑、用户密度）与训练数据相似的问题实例上表现良好。网络环境发生重大变化的情况在现实场景中十分常见，而这将需要耗费大量成本进行重新训练。

2) 缺乏可解释性和最优性保证：DNN 如同“黑箱”，无法保证所得解的质量；对于性能界限至关重要的网络资源分配而言，这是一个重大缺陷。

C. 拉格朗日对偶分解

相比之下，我们采用拉格朗日对偶分解。这一选择源于其对我们的问题结构所具有的内在优势：

1)可分解性：通过对耦合约束进行对偶化，原本难以处理的问题自然分解为针对每个节点的相互独立、规模更小的子问题。这与命名数据网络的分布式架构完美契合。

2) 理论可靠性：该框架提供了一种系统化的迭代方法，具有收敛性质，并给出一个对偶间隙，用于界定任何可行解的次优性。

3)效率：由此产生的子问题可以通过闭式解或高效的算法解来解决，从而使每个节点的计算复杂度都较低。

因此，为了在分布式系统中平衡最优性、计算复杂度与实际部署能力，拉格朗日对偶分解是最有前景的路径。

附录 B 引理 1 的证明

证明：式（10）中的拉格朗日 $\mathcal{L}$ 可通过对项进行分组改写为：

$$\begin{aligned}\mathcal{L} = &\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{O}_s}\sum_{l\in\mathcal{L}}\lambda_{1,i,l}y_{s,i,l} + \sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}}(w_uQ_l - \lambda_{1,u,l})x_{u,l}\\ &+ \sum_{f\in\mathcal{F}}\sum_{l\in\mathcal{L}}\left(\sum_{k\in\mathcal{O}_f}\lambda_{2,f,k,l} - \lambda_{1,f,l}\right)x_{f,l}\\ &+ \sum_{f\in\mathcal{F}}\sum_{k\in\mathcal{O}_f}\sum_{l\in\mathcal{L}}(\lambda_{1,k,l} - \lambda_{2,f,k,l})y_{f,k,l}. \end{aligned} \tag{22}$$

E按组分开：i i d d

- 第一项仅取决于 $\mathbf{y}_{s,i}$，约束为（7）、（9）
- 第二项仅取决于 $\mathbf{x}_u$，约束为（3）、（4）、（9）
- 第三项仅取决于 $\mathbf{x}_f$，约束为 $x_{f,l}\in[0,1]$
- 第四项仅取决于 $\mathbf{y}_{f,k}$，约束为（7）、（9）

对偶化后，不再存在交叉项或共享约束 (5) and (6). Thus,

$$\begin{aligned}g(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2) = &\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{O}_s} g_{s,i}(\boldsymbol{\lambda}_1) + \sum_{u\in\mathcal{U}} g_u(\boldsymbol{\lambda}_1)\\ &+ \sum_{f\in\mathcal{F}} g_{f,x}(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2) + \sum_{f\in\mathcal{F}}\sum_{k\in\mathcal{O}_f} g_{f,k}(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)\end{aligned} \tag{23}$$

其中，每个 g_* 都是相应子问题的最优值。主问题对该总和进行最小化，根据定义，这等价于对偶问题。

■