

高铁场景中基于 DASH 协议的流媒体自适应云协同传输方法^①



姜 堃

(北京中科同舟科技有限公司 科创信息研究院, 北京 100020)

通信作者: 姜 堃, E-mail: jiang_12023@163.com

摘要: 高铁逐渐成为流行的出行选择, 高铁途中用户对流媒体服务有着较高的需求. 但是高速移动场景下用户带宽抖动严重、用户的媒体体验难以得到保障. 为此, 本文提出了一种基于 DASH 协议的跨层流媒体自适应云协同传输优化方法. 首先提出了一个基于 DASH 协议的跨层流媒体自适应云协同传输架构, 并提出一个高铁环境下用户的 *QoE* 模型. 在此基础上构建了基于 DASH 的跨层流媒体自适应云协同传输的优化模型, 并提出了基于 DASH 协议的跨层流媒体云协同自适应码率选取算法, 提升用户的媒体体验. 最后仿真实验结果表明, 本文提出的方法能很好地提高高铁用户的媒体体验, 有助于高速移动场景下流媒体优化传输的研究.

关键词: 高铁; DASH 协议; 流媒体; *QoE*

引用格式: 姜堃. 高铁场景中基于 DASH 协议的流媒体自适应云协同传输方法. 计算机系统应用, 2024, 33(1): 263-271. <http://www.c-s-a.org.cn/1003-3254/9382.html>

DASH Protocol-based Adaptive Cloud Collaborative Transmission Method for Streaming Media in High-speed Rail Scenarios

JIANG Kun

(Science and Technology Information Research Institute, Beijing Zhongke Tongzhou Technology Co. Ltd., Beijing 100020, China)

Abstract: High-speed rail (HSR) has gradually become a popular travel option, and passengers have high demand for streaming media services during HSR travel. However, in high-speed mobile scenarios, user bandwidth jitter is severe, and user media experience cannot be guaranteed. To this end, a cross-layer optimization method for adaptive cloud collaborative transmission of streaming media, based on DASH protocol, is proposed in this study. Firstly, a cross-layer architecture for adaptive cloud cooperative transmission of streaming media, based on DASH protocol, is proposed, and a *QoE* model for users in high-speed rail environment is suggested. Next, on this basis, a cross-layer optimization model for adaptive cloud collaborative transmission of streaming media, based on DASH protocol, is constructed, and a cross-layer adaptive bitrate selection algorithm for cloud collaborative transmission of streaming media, based on DASH protocol, is proposed to improve the user's media experience. Finally, the simulation experiment results show that the method proposed in this study can greatly improve the media experience of HSR passengers, and is helpful for the optimization study of the transmission of streaming media in high-speed mobile scenarios.

Key words: high speed rail; DASH (dynamic adaptive streaming over HTTP) protocol; streaming media; *QoE*

随着网络技术的发展, 流媒体流量在全球移动互联网流量中所占比例越来越高, 据思科预测, 2022 年视

频流量占全球移动流量的百分比高达 79%^[1]. 据《中国交通的可持续发展》统计, 至 2020 年我国高速铁路营

① 收稿时间: 2023-07-18; 修改时间: 2023-08-24; 采用时间: 2023-09-15; csa 在线出版时间: 2023-11-28
CNKI 网络首发时间: 2023-11-30

业总里程超过 3.8 万千米。随着高铁里程和速度的飞速发展,乘坐高铁出行的旅客日益剧增,高铁旅客对于旅途中流媒体业务有着强烈的需求。高铁环境下,用户移动速度快导致其网络信道变化频繁,如何为高速移动场景下的用户提供高质量的流媒体是一个巨大的挑战。DASH (dynamic adaptive streaming over HTTP) 是为应对网络信道质量抖动导致用户媒体体验下降提出的视频传输协议。高铁环境下,使用 DASH 协议传输媒体数据是一个提升用户媒体体验的方法。DASH 协议是基于 HTTP 协议的应用层传输协议。而高铁环境下,由于用户基站的频繁切换导致 HTTP 协议的性能有着很大的影响^[2]。

国内外很多学者针对流媒体的优化传输有了大量的研究。Huang 等^[3]认为通过估算网络带宽进行码率的选取可能会因带宽估算不准确而导致用户体验下降,为此提出了基于缓存数据状态的码率选取算法,提升了用户的媒体体验。El Marai 等^[4]利用软件定义的网络的全局视野以及强大的路由规划功能,在网络状态难以估算的情况下,选择更为合适码率的媒体信息。Kan 等^[5]基于强化学习的速率适应和自适应预测,提出了一种感知自适应切片方案来提高带宽效率。Menon 等^[6]提出一种用于实时视频流应用的在线比特率阶梯构建方案,每个目标比特率的优化分辨率由使用离散余弦变换的任何预定义分辨率集确定 (DCT)。Santos 等^[7]提出了一种使用长短期记忆 (LSTM) 神经网络的新 AQM 算法来提高用户感知的视频质量。Huang 等^[8]提出了一种将 MEC 服务器中的视频片段缓存和转码与资源分配相结合的联合优化算法。Yuan 等^[9]提出了基于博弈论的带宽分配算法,同时考虑了用户的缓存长度、接收视频码率、视频长度等,并将带宽分配问题建模为一个合作谈判问题,通过凸优化求得纳什谈判解。最后通过实验证明其提出的方案同时保障了用户的公平性和流媒体的传输性能。Zhang 等^[10]研究了当前 LTE 网络中 eMBMS 传输模式在部署为单频网络 (SFN) 时提供广播服务的能力和局限性。并且通过物理层的研究优化,如非正交复用,分层复用技术,以及更灵活的正交频分复用系统配置,更长的循环前缀和更小的子载波间隔等,实现增强的广播服务传输能力,更高的频谱效率,改善的服务质量,和更高效的 SFN 部署选择。Park 等^[11]研究了 LTE 中基于 DASH 的组播传输方案,并通过向前纠错编码等方案进一步加强传输

系统的鲁棒性。Hu 等^[12]提出了一个语义感知的自适应传输模型,在传统自适应传输的基础上引入对视频内容的语义分析,如足球比赛可分为 3 种类型,一种正常比赛镜头,一种进球特写,一种回放,根据大众对不同类型视频关注点不同选择适合的码率,用以提升用户的体验。Zhang 等^[13]在研究中提出了一种名为“预测和自适应深度编码 (predictive and adaptive deep coding, PADCC)”框架,以在给定目标传输质量要求的情况下实现灵活的码率优化。PADCC 框架通过引入可变码长,采用 DeepJSCC (deep joint source-channel coding) 模型,从而实现了代码长度的灵活调整。为了预测图像传输任务的峰值信噪比 (peak signal-to-noise ratio, PSNR),他们引入了 Oracle 网络 (OraNet) 模型,该模型结合图像传输任务的内容、信道信噪比 (signal-to-noise ratio, SNR) 和压缩比 (compression ratio, CR) 来进行预测。接着,他们提出了 CR 优化器,旨在寻找满足 PSNR 质量约束的最小数据级或实例级 CR。通过这 3 个模块的协同作用, PADCC 能够以最小的 CR 传输图像数据,极大地提高了带宽的使用效率^[13]。Pu 等提出了一种创新的超高清 (ultra high definition, UHD) 视频流传输框架,其通过将纠错码存储与多源流传输相结合,实现了对视频流的高效传输。在该研究中,他们引入了两种度量标准,一种考虑截止时间,一种考虑延迟敏感性,以评估视频服务器的服务质量。为了实现自适应的服务质量更新,他们倡导采用联邦学习范式,包括基于强化学习的多服务器选择 (也即用户本地训练) 和全局服务质量聚合^[14]。Liu 等提出了一种移动边缘计算 (mobile edge computing, MEC) 辅助的低延迟自适应比特率 (adaptive bitrate, ABR) 视频流传输方法。他们将优化问题建模为一个混合整数非线性规划 (mixed-integer nonlinear programming, MINLP) 问题,并设计了一种低复杂度的两步迭代算法来解决该问题^[15]。Duanmu 等则提出了一种客观 *QoE* 模型,名为贝叶斯流媒体质量指数 (Bayesian streaming quality index, BSQI),该模型通过合理地结合人类视觉系统的先前知识和人工标注数据集,以预测视频流传输的质量^[16]。Yuan 等提出了一种名为 VSiM 的端到端方案,旨在支持异构无线网络中的移动视频流应用。该方案的关键思想是根据多个用户的移动配置文件和用户体验质量 (quality of experience, *QoE*) 相关知识,在多个用户之间分配瓶颈带宽,以实现最大最小 *QoE* 的公平性^[17]。此外, Taraghi 等提出了一种基于

云的开源测试平台,名为LLL-CAdViSE框架,用于评估实时低延迟的流会话,特别是在HTTP自适应流(HTTP adaptive streaming, HAS)格式上运行的情况^[18]. Zhong等将移动场景中的自适应流传输问题建模为线性松弛为凹的多源组播多速率问题(multi-source multicast multi-rate, MMMP).他们提出了分布式交付算法(distributed delivery algorithm, DDA)来解决该问题,并进一步提出了基于问题物理含义的启发式逼近方法(heuristic-based DDA, H-DDA)^[19].对于基于HTTP自适应流的视频流传输,特别是在蜂窝网络边缘(如5G)的情况下,Aguiar-Armijo等提出了自适应视频流传输框架(SPACE),该框架结合了边缘分段预取和缓存策略.他们分析了不同的分段预取策略,并研究了它们在资源利用率、服务质量等方面的表现^[20].

上述文献使用不同的方法提升了用户的媒体体验,但是,高铁场景下流媒体传输的优化研究还是较少.高铁环境下依旧存在用户对高质量流媒体的需求与网络资源时空分布不均匀的矛盾.如何在高铁环境下减少传输时延、提升用户的媒体体验是面临的一个挑战.为了解决上述问题,本文提出了一个高铁环境下基于DASH协议的跨层流媒体云协同传输架构,提出了一个新的适应高铁环境的用户 QoE 模型,并在此基础上对高铁场景基于DASH的跨层云协同传输过程进行了形式化描述,并构建了以最大化用户的 QoE 为目标的优化模型.最后本文提出了一个基于DASH协议的跨层流媒体云协同自适应码率选取算法求解模型,最后进行了仿真实验验证提出算法的性能.

本文的组织结构如下:第1节构建了本文提出的基于DASH的跨层流媒体传输模型,并构建了本文的最优化模型.第2节提出了模型的求解算法.第3节通过实验仿真验证了所提算法的有效性.最后总结全文.

1 高铁场景中基于DASH的跨层流媒体云协同传输架构以及模型

本节提出了基于DASH的跨层流媒体云协同传输架构以及系统模型.首先,本节提出了系统架构并对传输过程进行形式化描述,对优化问题进行建模.之后,本节提出了一个基于DASH的跨层流媒体云协同自适应码率选取算法求解最优化问题.

1.1 基于DASH协议的跨层流媒体云协同传输架构

本文提出的基于DASH协议的跨层流媒体云协同

传输架构如图1所示.媒体数据编码成不同质量的数据存储在服务器中,用户根据自己的网络状态选择合适码率的视频,以最大化网络利用率.基于DASH的跨层流媒体云协同传输架构将传统的客户端与服务器亮点的点对点传输变为如图1所示的媒体服务器与云代理、云代理与客户端的传输.在整个传输流程中本文增加的云代理主要有以下功能,首先,云代理提供算力资源,运行本文提出的算法,并转发服务器与客户端的信息.其次,云代理可以根据算法运行的结果选择将服务器发送来的数据转发到合适的基站.最后,软件定义云代理功能,算法的更新迭代更容易.其运行流程如下所示.

视频数据存储在媒体服务器端,并且按照DASH传输协议的要求分段分层编码,客户端通过数据拉取的形式申请视频数据.在媒体服务器和用户的客户端中间,本文设置一个云代理,用于运行本文提出的算法,转发客户端发送给媒体服务器的视频请求信息.传输流程如下:首先用户第1次申请感兴趣的视频数据,请求信息发送到基站,基站转发给云代理,云代理将用户请求转发给媒体服务器.媒体服务器收到请求之后将视频数据的MPD(media presentation description)文件发送给用户.用户收到MPD文件就可以依据此申请不同时间片段的媒体数据.其次,用户发送媒体请求数据、当前网络质量、当前移动速度、当前位置给基站,基站将这些信息转发给云代理.云代理依据收到的信息运算本文提出的算法计算出最为合适的码率数据以及最优的接收数据基站,基于此修改用户的请求文件转发给媒体服务器,并基于上述信息预测下一时刻用户的信道质量以及位置,申请媒体信息至基站缓存.最后,媒体服务器收到信息后按照请求文件发送合适码率的媒体数据至合适的基站缓存.

本文使用DASH传输协议传输视频,但是本文的传输架构并不局限于此协议,其他媒体的自适应传输协议、方法,有可能编码传输方式不同,或者不会对视频进行预编码,本文提出的传输架构依然适用.

1.2 用户 QoE 模型

本文目的为优化高铁用户的视频观看 QoE ,影响用户 QoE 的参数为视频的客观质量、用户整体观看过程中卡顿次数、卡顿持续时间以及选择视频质量的抖动,影响这些参数的因素有当前用户和不同基站的距离、列车行驶时速、用户信道状态、信道带宽限制、视频码率、视频总时长、以及视频分辨率等.本

文假设高铁用户媒体体验 QoE 为:

$$QoE = \alpha \times \Psi(QoS) - \beta \times \Phi(N, L) - \gamma \times \Omega \quad (1)$$

其中, Ψ 、 Φ 、 Ω 分别表示视频客观质量、视频播放卡顿以及卡顿时间、视频质量抖动对用户 QoE 的影响函数. 参数 α 、 β 、 γ 为影响用户 QoE 因素的权衡系数, 其值可以根据用户经验设置.

$$\Psi(QoS) = \ln(QoS) \quad (2)$$

其中, 函数 Ψ 表示视频客观质量对用户 QoE 影响函数, 客观质量如视频的分辨率、视频码率等. 一般来说, QoE 随着视频质量的增加而增加. 本文中视频质量可用 PSNR 或 SSIM 值等客观评价参数表示.

$$\Phi(N, L) = ae^{-b(L) \cdot N} + c \quad (3)$$

其中, 函数 Φ 表示播放卡顿以及卡顿时间对用户 QoE

影响函数. 其中参数 N 表示视频的卡顿次数, 参数 L 表示卡顿的时长. 参数 a 、 b 、 c 为权衡参数, 其值可以根据用户经验设置.

$$\Omega = 1 - \frac{\sum_{i=1}^M \sum_{j=0}^{S-1} (|l_h - l_{s-j}| \times (S-j))}{\sum_{i=1}^M \sum_{j=0}^{S-1} (l_h \times (S-j))} \quad (4)$$

其中, Ω 表示视频质量抖动指数, 含义是 1 减去之前视频片段切换次数的加权和除以传输时间内接收最高码率等级的加权和. S 表示当前时刻传输的视频片段总数, l_h 表示传输时间内传输视频的最高接收质量等级. $S-j$ 表示最近视频质量等级切换的负增益. Ω 的值越接近 1, 则表示视频传输系统的稳定性越高.

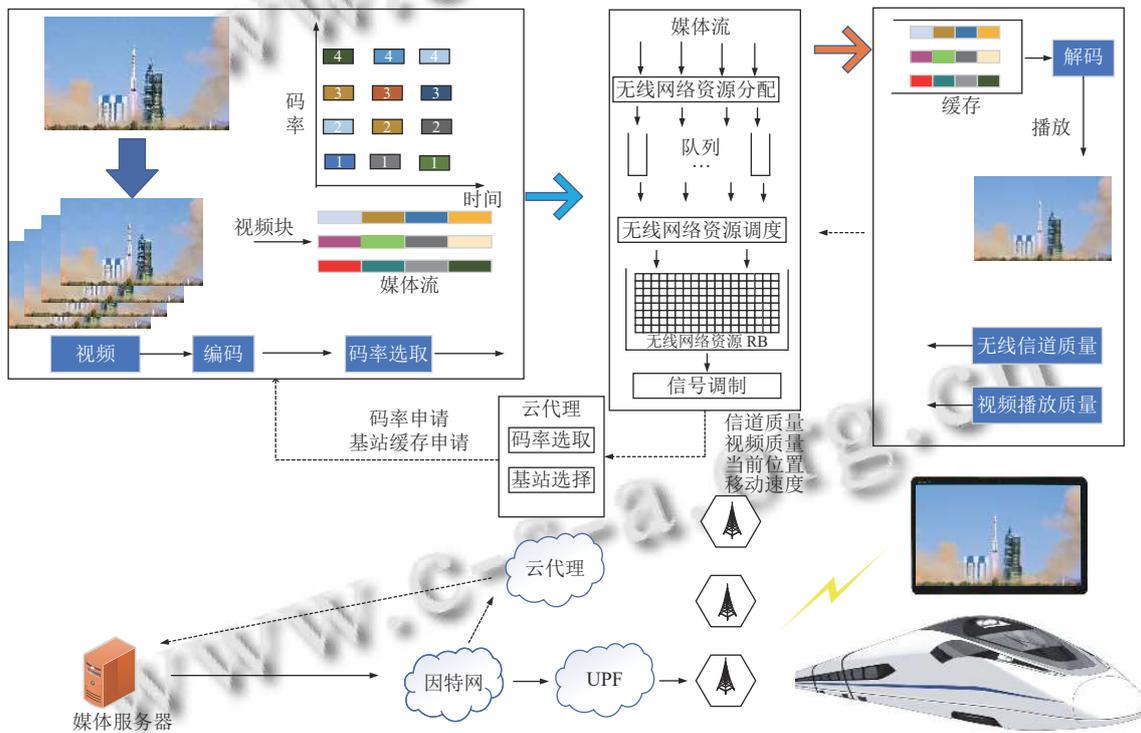


图1 基于 DASH 的跨层流媒体云协同传输架构

1.3 基于 DASH 协议的跨层流媒体云协同传输模型

媒体数据切割成等时长的视频片段, 每个视频片段被编码为不同质量等级数据存储在媒体服务器内. 云代理通过运行提出的算法预取媒体数据至合适的基站缓存. 假设 $\{S_1, S_2, \dots, S_l\}$ 表示不同质量等级的视频片段, $\{1, 2, \dots, l\}$ 表示视频的质量等级. 用户使用字符 UE_k 表示, 其中 $k \in \{1, 2, \dots, K\}$.

$$N_t = RB_t \times (N_{TRB} \times N_{OFDM} \times N_{sub_channel}) \times MCS_t \times R_m \quad (5)$$

其中, N_t 表示预测的 t 时刻网络带宽, RB_t 表示在 t 时刻基站可以分配给用户的无线网络资源个数, N_{TRB} 表示 1 ms 内时隙个数, 值为 2; RB 占 7 个 OFDM 符号, 即 $N_{OFDM} = 7$, 频域上占 12 个连续子载波, 即 $N_{sub_channel} = 12$. RB 个数由基站用户当前信道状态做出决策, MCS

为用户反馈的CQI值对应的数据编码方式。\$R_m\$表示无线蜂窝网络下不同CQI对应的信道编码方法的编码效率。

$$CQI_t = \left(1 + \frac{CQI'_{t-1} - CQI_{t-1}}{CQI_{t-1}}\right) \times CQI'_{t-1} \quad (6)$$

其中, 参数\$CQI_t\$表示\$t\$时刻预测的用户CQI, \$CQI'_t\$表示\$t\$时刻真实的用户CQI。

$$\begin{cases} \text{Bitrate}_t \leq N_t \\ \text{Bitrate}_t = \max(V_1, V_2, \dots, V_l) \end{cases} \quad (7)$$

其中, 参数\$V_l\$表示质量等级为\$l\$的视频片段的码率, \$\text{Bitrate}_t\$表示\$t\$时刻选取视频片段的码率。

$$QoS = f(\text{Bitrate}) \quad (8)$$

视频的客观质量\$QoS\$与视频片段的码率有着直接

$$\begin{cases} d_s = \sqrt{(BS_{tx} - (u_{tx} + v \times t))^2 + (BS_{ty} - (u_{ty} + v \times t))^2} \\ d_{s+1} = \sqrt{(BS_{(t+1)x} - (u_{tx} + v \times t))^2 + (BS_{(t+1)y} - (u_{ty} + v \times t))^2} \end{cases} \quad (10)$$

如果\$d_s > d_{s+1}\$, 则预取视频数据至基站\$s\$的缓存, 如果\$d_s < d_{s+1}\$, 则预取视频数据至基站\$s+1\$的缓存。

本文的优化目标为:

$$\begin{cases} \max(QoE) \\ \text{s.t. Eq.(1) - Eq.(10)} \end{cases} \quad (11)$$

其中, \$QoE\$是本文中(1)提出的\$QoE\$模型, 我们通过用户观看视频时候的视频客观质量、视频播放卡顿以及卡顿时间、视频质量抖动定义了本文中用户的\$QoE\$。因此, 我们可以通过最大化用户的\$QoE\$为本文的优化目标, 如何解决式(11)提出的问题, 即在式(1)–式(10)的约束下, 最大化用户的\$QoE\$。在第2.4节中, 我们提出一个基于DASH的跨层流媒体云协同自适应码率选取算法去求解上述的问题。

1.4 基于DASH协议的跨层流媒体云协同自适应码率选取算法

本节提出一个基于DASH的跨层流媒体云协同自适应码率选取算法(CCABA), 预估用户的信道质量以及位置, 预取合适码率的媒体信息缓存到合适的基站侧, 以减少流媒体传输时延, 最大化用户的\$QoE\$。假设\$N_t\$是时间\$t\$的预测网络带宽。假设在时间\$t\$选择了最高质量级别为\$l_h\$的视频片段。质量等级为\$l_h\$的视频片段的比特率为\$\text{Bitrate}_{l_h}\$。质量等级为\$l_m\$的视频片段的比特率为\$B_m\$。

$$l_m = \text{ceil}\left(\frac{l_h}{2}\right) \quad (12)$$

的映射关系。

$$RB_t \leq RB'_t \quad (9)$$

其中, 参数\$RB'_t\$表示在\$t\$时刻, 用户连接的基站剩余的\$RB\$数量。

假设高铁移动过程中, 用户共需要连接(切换)\$N\$个基站, 高铁旅途中基站的集合为\$\{BS_1, BS_2, \dots, BS_N\}\$, 本文使用\$B_k, k \in \{1, 2, \dots, N\}\$表示第\$k\$个基站。假设时刻\$t\$高铁移动速度为\$V\$, 用户连接第\$s(1 < s < N)\$个基站, 用户的坐标为\$(u_{tx}, u_{ty})\$, 其连接基站坐标为\$(BS_{tx}, BS_{ty})\$, 下一个基站\$s+1\$的坐标为\$(BS_{(t+1)x}, BS_{(t+1)y})\$。预测下一时刻用户的坐标为\$(u_{tx} + v \times t, u_{ty} + v \times t)\$。则此刻用户与两个基站\$s\$以及\$s+1\$的距离分别为:

函数\$ceil\$表示取整上进位。

用户申请视频数据, 并定期上报自己的信道质量\$CQI\$以及当前位置速度。基站将此信息发送给云代理, 云代理通过此信息预估下一时刻用户的\$CQI_{kal}\$, 并且为用户分配合适数量的无线资源\$RB\$。本文提出了一个基于DASH协议的跨层流媒体云协同自适应码率选取算法。云代理端分别计算当码率选取为\$B_h\$以及\$B_m\$的视频片段时, 用户的媒体体验\$QoE(B_h)\$和\$QoE(B_m)\$。并且:

$$d = |QoE(B_h) - QoE(B_m)| \quad (13)$$

通过计算\$d\$的值重置\$B_h\$或者\$B_m\$的值。例如如果\$d > 0\$, 则\$h = h, m = m + (h + m)/3\$。如果\$d < 0\$, 则\$h = m, m = m - (h + m)/3\$。并且对上述步骤进行迭代循环, 直到\$d = 0\$结束循环。最后输出最优的视频码率为\$\text{Bitrate}_n = B_{(h+m)/2}\$作为算法的最终解。并且云代理根据式(10)计算此次数据缓存基站的位置。

云代理将此结果发送给媒体服务器, 媒体服务器按照收到的信息发送要求码率的视频片段至要求的基站缓存。CCABA算法伪代码如算法1。

算法1. 基于DASH的跨层流媒体云协同自适应码率选取算法CCABA

- 1) 初始化;
- 2) 用户发出视频片段申请, 上报当前用户信道状态\$CQI\$以及当前位置、移动速度;
- 3) if \$CQI > 0\$, 预测\$CQI_{kal}\$, 根据用户位置速度获取相应基站可用\$RB\$数量;
- 4) \$N_t = RB_t \times (N_{TRB} \times N_{OFDM}) \times MCS_t \times R_m\$。

```

5)  $d=100$ ;
6)  $B_h \leq N_i$ ;
7) While( $d \neq 0$ ) {
8)   计算并预测  $QoE(B_h)$ ;
9)   计算并预测  $QoE(B_m)$ ;
10)   $d=QoE(B_h)-QoE(B_m)$ ;
11)  if  $d > 0$ 
12)     $h=h, m=m+(h+m)/3$ ;
13)  else  $h=m, m=m-(h+m)/3$ ;
14)  输入  $Bitrate_n=B_{(h+m)/2}$ ;
15) If  $B_h$  不存在, 发送最低质量等级的视频;
16) End

```

CCABA 算法运行前需要用户上传自己的信道状态 CQI 、位置以及移动速度到基站, 详见伪代码第 2 行。之后基站将信息发送给云代理, 云代理运行我们提出的算法, 首先云代理通过用户上传的速度和位置预测用户的路线, 获取用户可用的 RB 数量, 并且根据伪代码第 4 行计算出来当前信道状态下, 可用的 RB 数量可以提供的网络带宽 N_i 。然后定义一个变量初始值为 100, 详见伪代码第 5 行。如果选取的视频码率小于当前网络带宽, 即可以无卡顿的发送当前码率的视频详见伪代码第 6 行。伪代码的第 7–14 行, 主要使用折半查找算法寻找当前网络带宽条件下, 最适合发送视频的码率。伪代码第 15 行表示如果存储在媒体服务器的所有媒体码率需要的带宽都大于当前带宽, 则选择发送最低质量等级码率的视频。

CCABA 算法的复杂度主要集中在步骤 7)–14) 的循环部分, 其余算法代码的复杂度都是常数时间复杂度 ($O(1)$), 可以忽略不计。综上所述, 整个算法的总体复杂度可以表达为: $O(k \times n)$ 到 $O(k \times n^2)$ 。这个范围涵盖了算法中主要的计算成本, 但具体的复杂度取决于算法收敛的迭代次数和输入数据的规模。在实际应用中, 可以根据实验或分析得出更具体的复杂度估计。

2 仿真实验和结果分析

2.1 实验环境

本文选取高铁速度为高铁平均速度 $V = 300$ km/h, 模拟路段基站分布均匀, 无线蜂窝网基站间距离为 500 m。用户通过的基站数量设置为 100 个, 模拟路段长度为 50 km。本文仿真中使用的参数在表 1 中描述。

本文采集了高铁线路用户端的实时的 5G 网络带宽。数据包括高铁从静止到最高速度的不同阶段。高铁

刚启动时, 用户的移动速度为 0–50 km/h。高铁平均时速 280–320 km。由于高铁移动速度快, 当高铁中有大量用户同时接入网络的时候, 用户的网络带宽抖动严重。本文使用免费软件 Wireshark 在高铁运行期间捕获网络数据。本文在高铁上不同速度和不同地形路段, 每 2 s 测量一次网络的用户信道 CQI 。以及基站在高铁上不同速度路段为用户分配的带宽。

表 1 仿真实验参数

参数	数值
基站间距	500 m
最高带宽	100 MHz
高铁速度	300 km/h
视频片段	10 s
仿真路径长度	50 km
基站数量	100
视频片段时长	2 s

为了验证本文提出方法的性能, 选取了 3 种算法进行对比。

(1) BSBC^[21]

该算法可以有效利用缓存资源, 提高带宽利用率, 并通过逐步切换提高码率切换的平滑性和稳定性。

(2) HLS^[22]

此方法由苹果公司提出。采用可变码率视频发送方法, 集成到许多流行的浏览器中。该算法根据用户缓冲率和丢包率自适应地选择比特率。

(3) ADMM^[23]

该算法将速率优化问题数学建模为一个考虑访问控制、资源分配和用户偏好的非凸优化组合问题, 提出了一种基于模拟退火的新算法来解决速率优化问题。该算法随机生成可行解, 设置最大迭代次数, 生成相邻可行解进行比较, 选择最佳迭代计算得到最优解。

本文使用 Python 搭建了网络仿真环境。在仿真实验中, 为了验证本文提出的 CCABA 算法, 使用 BLBC、HLS 以及 ADMM 算法进行对比。仿真实验中高铁网络的带宽变化, 本文随机选取之前采集的真实数据。本文下载了 3 个不同分辨率的视频^[24]。每个视频被分成 2 s 的片段, 并分别编码为不同的质量级别。在仿真实验过程中, 视频循环播放。联合可扩展视频模型 (JSVM, 版本 9.19) 用于编码和解码视频^[25]。按照空间可伸缩算法最多将视频片段编码为 6 个质量等级: 4K、1080p、720p、480p、360p 和 240p。根据实验统计, 算法中的参数设置如下, $a = 3.5$, $b = 0.15$, $c = 1.5$ 。为了更直观地对比用户的 QoE , 实验结果将 QoE 的值归一化。

$$QoE_{0,1} = \frac{QoE - QoE_{min}}{QoE_{max} - QoE_{min}} \quad (14)$$

我们通过式 (15) 衡量了客户端视频播放的抖动性:

$$DA = 1 - \frac{\sum_{e=1}^M \sum_{f=0}^{S-1} (|l_h - j'_{S-f}| \times w(d))}{\sum_{e=1}^M \sum_{f=0}^{S-1} (l_h \times w(d))} \quad (15)$$

其中, DA 表示不稳定性指数, 他是前几个片段中所有切换步骤的加权和除以传输时间内最高接收速率水平的加权和. S 表示片段的数量. l_h 代表在传输时间内传输的视频的最高接收速率水平. $w(d) = S - f$ 为最近的比特率切换分配较高的惩罚. DA 的值越接近 1, 稳定性越好.

视频的详细信息见表 2.

表 2 视频信息

视频名字	分辨率	视频长度 (s)
Four_People_1280x720_60	720p	10
Crowd_run_1080p50	1080p	10
Park_joy_2160p50	4K	10

2.2 实验结果

本节比较了本文提出的算法 CCABA 和不同算法 BSBC、HLS 和 ADMM 的结果. 用户 QoE 的结果进行了归一化. 视频的 SSIM 值是通过将用户接收到的视频数据进行上采样与服务器存储的最高质量视频进行计算得到的.

图 2、图 3 和图 4 展示了在不同算法下用户体验 (QoE) 和结构相似性指数 (SSIM) 的对比结果. 与其他算法相比, 这些算法没有考虑到高铁环境下网络的复杂多变特性. 在高铁环境中, 由于用户频繁切换基站, 数据丢失现象频繁发生, 从而导致底层拥塞算法频繁激活. 这进而可能导致对网络带宽的错误估计. 在不同视频源的传输过程中, 本文提出的 CCABA 算法在性能方面表现更为出色. 这是由于 CCABA 算法能够更好地适应高铁环境的变化, 包括频繁切换基站和数据丢失等. 与其他算法相比, CCABA 算法在网络资源的有效利用方面表现更佳, 从而提供了更好的用户体验和更高的 SSIM. 综上所述, 本文提出的 CCABA 算法在高铁环境下表现出色, 通过考虑网络的复杂性和变化性, 克服了其他算法在高铁环境中可能出现的问题. 在不同视频源的传输中, CCABA 算法能够实现更好的性能, 为用户提供更好的视频流媒体体验. 这突显了 CCABA 算法在适应性比特率视频流媒体中的优越性能.

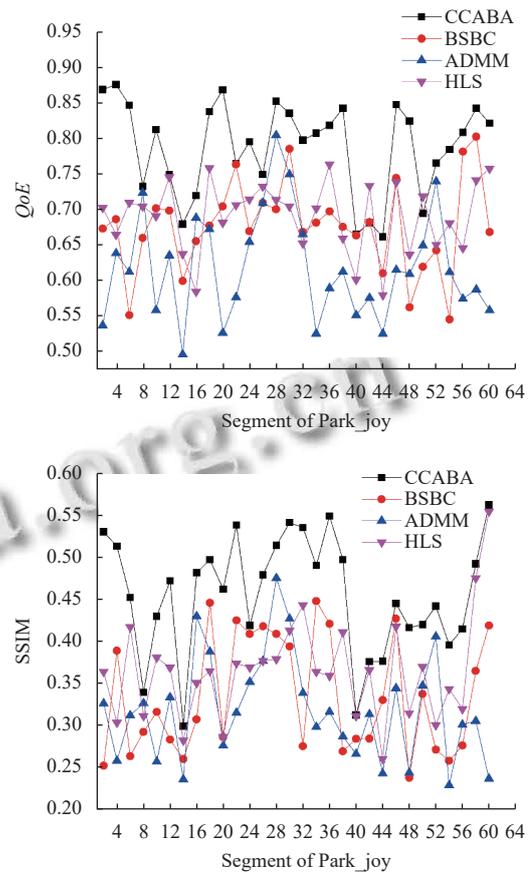


图 2 视频 Park_joy 的 SSIM 以及用户的 QoE

在图 2 中, 我们传输了质量等级为 4K 的视频源. 通过在视频传输过程中选择高分辨率的视频片段, 可以显著提升用户的 QoE . 在这种背景下, 本文提出的 CCABA 算法表现最佳, 用户的 QoE 在某些时刻表现得更为稳定. 图 3 和图 4, 视频源的最高分辨率分别为 720p 和 1080p. 在这种情况下, 用户收到的视频平均 SSIM 较高, 表示用户接收到的视频质量较好. 结合 SSIM 值和 QoE 值的趋势, 可以观察到用户的 QoE 与视频的 SSIM 值呈正相关关系, 说明视频的客观质量对用户体验非常关键. 从图 2-图 4 中可以明显看出, CCABA 算法在高铁环境下表现最佳. 当面临较大的信道状态变化时, CCABA 算法的性能更加稳健, QoE 下降相对平缓. 此外, 在图 2 中, 我们注意到在与其他算法在相同条件下进行对比时, ADMM 算法的 QoE 低于其他算法, 特别是在传输开始阶段. 这可能是因为 ADMM 算法选择随机比特率的视频, 在一开始时可能导致较低的用户体验. 基于以上对比结果, 可以得出结论: 在高速移动环境下, 本文提出的 CCABA 算法更为适用. 通过综合考虑 QoE 和视频质量, CCABA 算法在高铁环境下表现出色, 为用户提供更好的流媒体体验.

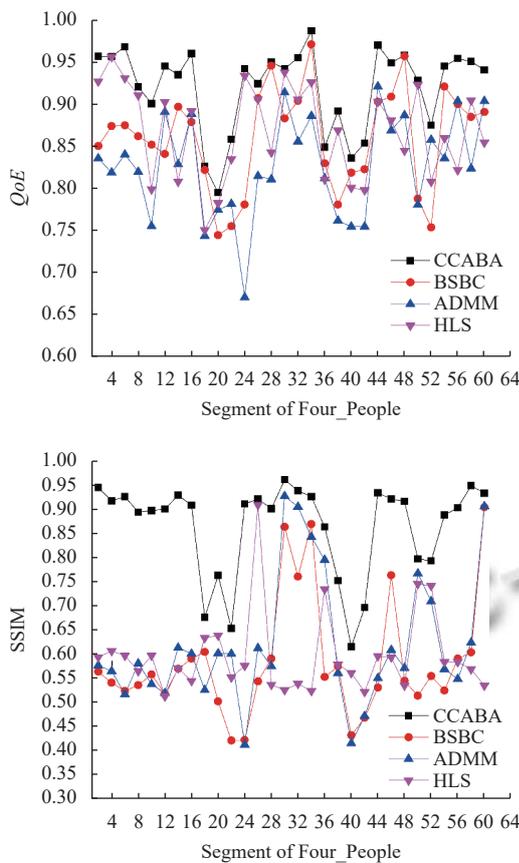


图3 视频 Four_People 的 SSIM 以及用户的 QoE

根据表3,我们可以观察到在不同算法下,视频码率分配对客户视频播放的抖动系数(DA)产生的影响.其中,DA越接近1表示视频播放越稳定,抖动越小.从表3的数据可以清楚地看出,本文提出的CCABA算法在效果上表现最佳.这种优越性主要源于CCABA算法的特殊设计,以解决高速移动场景下可能出现的问题.因此,CCABA算法更适合高速移动环境,其设计充分考虑了用户观看视频的环境参数,例如用户的速度、位置等.这使得CCABA算法能够更好地适应高速移动场景的挑战.此外,在其他算法中,BSBC算法也展现出相对良好的性能.其成功之处在于BSBC算法考虑了用户的缓存情况,并在算法中充分考虑了保障用户视频播放平滑性的重要性.然而,ADMM算法和HLS却未充分考虑高速移动环境中可能存在的问题,因此在这种场景下的性能相对有限.这进一步凸显了CCABA算法和BSBC算法相对于其他算法在高速移动环境下的优势.综上所述,表3的结果清晰地显示了CCABA算法在高速移动场景下的最佳性能,而BSBC算法在平滑用户视频播放方面也取得了不错的效果.在相对缺乏适应性的ADMM算法和HLS算法面前,

CCABA算法和BSBC算法在提高视频播放稳定性方面具备显著优势.

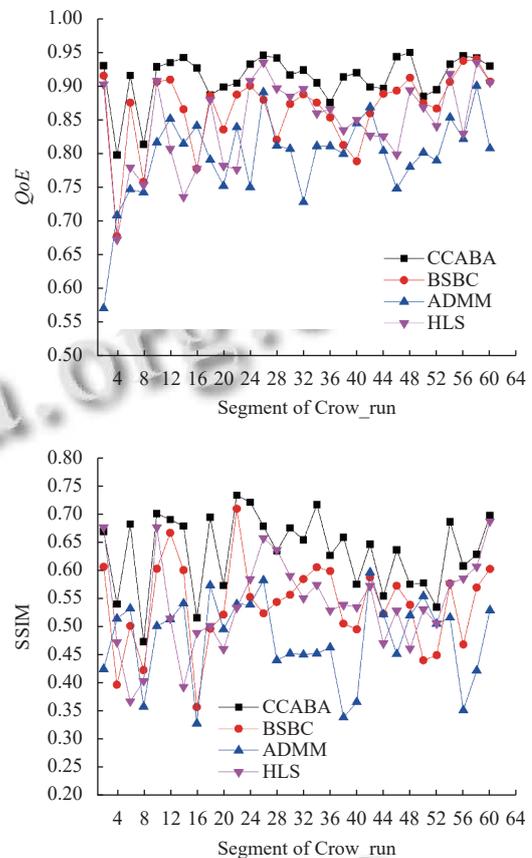


图4 视频 Crow_run 的 SSIM 以及用户的 QoE

表3 把不同码率选择算法下的视频播放的抖动

方法	Four_People	Crow_run	Park_joy
CCABA	0.8653	0.8578	0.8945
BSBC	0.8641	0.8557	0.8647
ADMM	0.8166	0.7156	0.7874
HLS	0.7561	0.6987	0.7258

3 总结

高铁环境下,网络环境变化频繁,严重影响用户的媒体体验.本文首先提出了一个基于DASH协议的跨层流媒体云协同传输架构,利用云代理的计算能力,为用户选择合适码率的媒体数据预测到合适位置的基站.之后定义了高铁环境下用户的QoE模型,基于此对高铁环境下流媒体传输进行了建模,并提出了最优化问题.为了解决最优化问题,本文提出了一个基于DASH协议的跨层流媒体云协同自适应码率选取算法.最后,通过仿真实验,本文提出的算法对比了其他3种不同的算法,结果表明,本文提出的算法在高铁环境下性能最高.

参考文献

- Forecast and methodology. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. [2023-03-01].
- Wang J, Zheng YF, Ni YZ, *et al.* An active-passive measurement study of TCP performance over LTE on high-speed rails. Proceedings of the 25th Annual International Conference on Mobile Computing and Networking. Los Cabos: ACM, 2019. 18.
- Huang WW, Zhou YP, Xie XY, *et al.* Buffer state is enough: Simplifying the design of QoE-aware HTTP adaptive video streaming. IEEE Transactions on Broadcasting, 2018, 64(2): 590–601. [doi: 10.1109/TBC.2018.2789580]
- El Marai O, Prados-Garzon J, Bagaa M, *et al.* Ensuring high QoE for DASH-based clients using deterministic network calculus in SDN networks. Proceedings of the 2019 IEEE Global Communications Conference. Waikoloa: IEEE, 2019. 1–6.
- Kan NW, Zou JN, Li CL, *et al.* RAPT360: Reinforcement learning-based rate adaptation for 360-degree video streaming with adaptive prediction and tiling. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(3): 1607–1623. [doi: 10.1109/TCSVT.2021.3076585]
- Menon VV, Amirpour H, Ghanbari M, *et al.* Efficient bitrate ladder construction for live video streaming. Proceedings of the 1st Mile-high Video Conference. Denver: IEEE, 2022. 99–100.
- Santos CEM, da Silva CAG, Pedroso CM. Improving perceived quality of live adaptative video streaming. Entropy, 2021, 23(8): 948. [doi: 10.3390/e23080948]
- Huang XY, He LJ, Wang LJ, *et al.* Towards 5G: Joint optimization of video segment caching, transcoding and resource allocation for adaptive video streaming in a multi-access edge computing network. IEEE Transactions on Vehicular Technology, 2021, 70(10): 10909–10924. [doi: 10.1109/TVT.2021.3108152]
- Yuan H, Wei XK, Yang FZ, *et al.* Cooperative bargaining game-based multiuser bandwidth allocation for dynamic adaptive streaming over HTTP. IEEE Transactions on Multimedia, 2018, 20(1): 183–197. [doi: 10.1109/TMM.2017.2724850]
- Zhang L, Wu YY, Walker GK, *et al.* Improving LTE eMBMS with extended OFDM parameters and layered-division-multiplexing. IEEE Transactions on Broadcasting, 2017, 63(1): 32–47. [doi: 10.1109/TBC.2016.2630269]
- Park J, Hwang JN, Li QY, *et al.* Optimal DASH-multicasting over LTE. IEEE Transactions on Vehicular Technology, 2018, 67(5): 4487–4500. [doi: 10.1109/TVT.2018.2789899]
- Hu SH, Sun LF, Xiao CX, *et al.* Semantic-aware adaptation scheme for soccer video over MPEG-DASH. Proceedings of the 2017 IEEE International Conference on Multimedia and Expo. Hong Kong: IEEE, 2017. 493–498.
- Zhang WY, Zhang HJ, Ma H, *et al.* Predictive and adaptive deep coding for wireless image transmission in semantic communication. IEEE Transactions on Wireless Communications, 2023, 22(8): 5486–5501. [doi: 10.1109/TWC.2023.3234408]
- Pu LJ, Shi JX, Yuan XJ, *et al.* EMS: Erasure-coded multi-source streaming for UHD videos within cloud native 5G networks. IEEE Transactions on Mobile Computing, 2023. [doi: 10.1109/TMC.2023.3238356]
- Liu WJ, Ding H, Zhang HX, *et al.* Low-latency oriented resource allocation for MEC-assisted adaptive bitrate video streaming. IEEE Transactions on Vehicular Technology, 2023. [doi: 10.1109/TVT.2023.3282962]
- Duanmu ZF, Liu WT, Chen DQ, *et al.* A Bayesian quality-of-experience model for adaptive streaming videos. ACM Transactions on Multimedia Computing, Communications, and Applications, 2023, 18(3S): 141.
- Yuan YL, Wang WJ, Wang YH, *et al.* Joint optimization of QoE and fairness for adaptive video streaming in heterogeneous mobile environments. IEEE/ACM Transactions on Networking, 2023. [doi: 10.1109/TNET.2023.3277729]
- Taraghi B, Hellwagner H, Timmerer C. LLL-CAdViSE: Live low-latency cloud-based adaptive video streaming evaluation framework. IEEE Access, 2023, 11: 25723–25734. [doi: 10.1109/ACCESS.2023.3257099]
- Zhong LJ, Wang M, Xu CQ, *et al.* Decentralized optimization for multicast adaptive video streaming in edge cache-assisted networks. IEEE Transactions on Broadcasting, 2023, 69(3): 812–822. [doi: 10.1109/TBC.2023.3254165]
- Aguilar-Armijo J, Timmerer C, Hellwagner H. SPACE: Segment prefetching and caching at the edge for adaptive video streaming. IEEE Access, 2023, 11: 21783–21798. [doi: 10.1109/ACCESS.2023.3252365]
- 吉爱国, 栾云哲. 基于缓存补偿的视频码率自适应算法. 计算机应用, 2022, 42(9): 2816–2822.
- McLean P. Apple launches HTTP live streaming standard in iPhone 3.0. https://appleinsider.com/articles/09/07/09/apple-launches_http_live_streaming_standard_in_iphone_3_0. (2009-07-09).
- Liu CY, Zhang HL, Ji H, *et al.* MEC-assisted flexible transcoding strategy for adaptive bitrate video streaming in small cell networks. China Communications, 2021, 18(1): 200–214. [doi: 10.23919/JCC.2021.02.013]
- Xiph.org video test media. <https://media.xiph.org/video/derf/>. [2023-03-01].
- Joint scalable video model11 (JSVM 11). <https://github.com/floriandejonckheere/jsvm>. [2023-03-01].

(校对责编: 牛欣悦)