

大规模客户端视频流带宽调度^①

丁佳龙, 谭光

(中山大学 智能工程学院, 广州 510006)

通信作者: 谭光, E-mail: tanguang@mail.sysu.edu.cn



摘要: 视频流服务的迅猛发展, 大规模用户共享带宽链路的场景不断增多. 现存的 DASH 视频流采用的 ABR 算法多用于提高单客户端用户的体验质量 (quality of experience, QoE), 还有一些算法仅针对数个客户端的情况. 本文提出一种应用于大规模客户端场景的带宽调度算法, 通过聚类算法减小调度规模, 再将带宽分配同 ABR 算法结合, 对聚类客户端进行比特率决策, 提高带宽利用率, 保证总体 QoE 最大化. 我们的实验结果表明, 与均分带宽的调度方式相比, 通过对聚类客户端带宽调度并应用到所有客户端的方式总用户 QoE 提升 99.4%, 相比于最先进的 Minerva 方案总 QoE 提高 10.7%.

关键词: 视频流; 体验质量; 多客户端; 聚类; 自适应带宽调度

引用格式: 丁佳龙, 谭光. 大规模客户端视频流带宽调度. 计算机系统应用, 2022, 31(3):294-301. <http://www.c-s-a.org.cn/1003-3254/8339.html>

Video Streaming Bandwidth Scheduling for Large-scale Client

DING Jia-Long, TAN Guang

(School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510006, China)

Abstract: With the rapid development of video streaming services, scenarios in which large-scale users share bandwidth links increase unceasingly. The existing adaptive bitrate (ABR) algorithm used in dynamic adaptive streaming over HTTP (DASH) video streaming is mostly used to improve the quality of experience (QoE) of single-client customers, while some other algorithms are only for multi-client situations. This study proposes a bandwidth scheduling algorithm for large-scale client situations. A clustering algorithm is adopted to reduce the scheduling scale. Then, bandwidth allocation is combined with the ABR algorithm to make bitrate decisions for clustering clients and thereby to improve bandwidth utilization and ensure a maximum overall QoE. The experimental results show that compared with the bandwidth-sharing scheduling method, the method of scheduling the clustering client bandwidth and applying it to all clients achieves a 99.4% increase in overall user QoE. The overall QoE increase is 10.7% on that of the best state-of-the-art scheme Minerva.

Key words: video streaming; quality of experience; multiple clients; clustering; adaptive bandwidth scheduling

1 引言

自视频流服务兴起以来, 视频流量逐渐成为网络上主体流量类型. 预计到 2022 年, 网络流量的 4/5 都将是流媒体服务^[1]. 与此同时, 互联网总用户数量将达到 53 亿, 这意味着多客户端共享带宽成为常态. 例如在家庭、校园或者企业内部, 多个智能设备共享网络带宽

链路请求视频服务. 当多个用户同时在线观看视频时, 不同的视频流之间会相互竞争有限的带宽资源, 如何给多客户端分配合适的带宽以提供最大的用户体验质量非常具有挑战性.

现有的视频流服务采用基于 HTTP 的动态自适应流 (dynamic adaptive streaming over HTTP, DASH)^[2]

① 基金项目: 国家自然科学基金 (61772509); 广东省自然科学基金 (2019A1515011066)

收稿时间: 2021-04-28; 修改时间: 2021-05-28; 采用时间: 2021-06-08; csa 在线出版时间: 2022-01-24

标准传输视频, 客户端可以在播放已缓存的视频内容的同时继续请求下载后续内容. 如图 1 所示, 区别于传统的视频流结构, DASH 服务器上将视频分成多个视频块, 并且编码成多种比特率以供客户端选择, 在客户端上部署的自适应比特率 (adaptive bitrate, ABR) 算法^[3]可以动态调整请求下载的视频块比特率, 从而改善用户的体验质量. ABR 算法会根据不同的网络条件 (预测带宽以及缓冲区占用) 对视频块的比特率作出决策, 保证客户端的最佳 QoE. 鉴于 QoE 的决定因素相互影响, 需要根据不同的用户偏好设定对应的权重以进行权衡, 尽可能为用户提供平滑高清的观看体验服务.

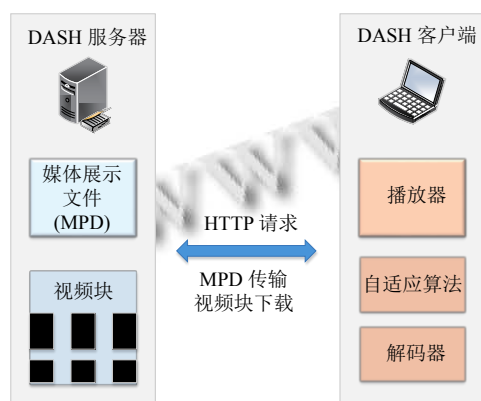


图 1 DASH 视频流传输结构

随着视频流服务的发展, 业内对 ABR 算法的研究也在不断改善. 最初的自适应比特率算法仅从网络预测吞吐量^[4-7]或是当前缓冲区占用^[8-11]角度优化 QoE, 尽管相比于非自适应视频流提升了用户体验质量, 但由于缺乏完整的环境信息造成带宽浪费或者卡顿事件. 在此基础上 Yin 等人综合两种指标提出基于控制理论方式的决策算法^[12], 把预测带宽以及缓冲区当前占用作为输入信号, 通过优化 QoE 进行比特率决策并保证单客户端获得最优体验质量. Mao 等人结合强化学习作为比特率决策算法^[13], 采用 asynchronous advantage actor-critic (A3C) 训练神经网络, 摒弃过去的假设模型, 将客户端的预测带宽和缓冲区占用作为状态空间, 直接从观测结果中学习经验并不断优化控制策略, 最后得到视频流客户端最佳比特率决策.

虽然现存的 ABR 算法可以保证单个用户最优的体验质量, 但是当应用于客户端规模较大的场景时, 每个客户端会基于自身部署的自适应算法请求最大比特率的视频块, 这会占用更多的带宽导致彼此相互制约

而降低观看体验质量. 不同的带宽分配方案会导致 ABR 算法给出截然不同的决策, 尤其是当客户端数目很大时, 缺乏合理的带宽调度方案会使得多客户端相互竞争.

本文提出一种大规模客户端视频流带宽调度方案, 针对较大数量的客户端, 根据其预测带宽以及当前缓冲器占用情况进行聚类并划分成几种不同的客户端类型, 进而将大规模的客户端场景缩小成几类客户端带宽调度情形, 不同类的客户端根据其所在类占比添加权重并计算每一类应分配的带宽份额, 相同类的客户端分配等量的带宽资源, 以此实现大规模客户端实时带宽调度.

对于缩小规模的聚类客户端利用模型预测控制算法 (model predictive control, MPC)^[12] 建立视频流带宽调度模型: (1) 带宽调度: 在客户端之间加入带宽控制器以统计预测带宽, 根据总预测带宽情况判定是否需要带宽调度, 当预测带宽总数超过带宽阈值时为每个聚类客户端预分配带宽; (2) 比特率决策: 结合预测带宽以及缓冲器占用情况计算不同的带宽分配方式下各聚类客户端所能达到的最大 QoE, 最后选取最大 QoE 对应的带宽分配方式以及视频块的比特率决策. 最后, 本文通过在真实带宽轨迹上进行仿真实验, 比较在同一带宽阈值下通过聚类带宽调度、Minerva^[14] 以及均分带宽 3 种方案取得的总用户体验质量, 验证我们的方案能够显著提高总用户的 QoE.

2 研究背景和动机

在多客户端视频流场景中, 多客户端相互竞争可用带宽, 造成带宽资源缺乏. 基于现有的 ABR 算法, 每个客户端都追求自身 QoE 最大化, 请求更高质量的视频块使得带宽占用增加, 而当多个用户受限于同一带宽约束时, 会引发网络拥塞并使得 QoE 受损. 另一方面各客户端上的播放缓冲器的性能不一, 其可存储的视频块上限不同, 此外不同比特率的视频块尺寸大不相同, 每个设备当前剩余可缓存视频容量的差异也会导致不同视频比特率的选择.

对于多客户端视频流, 有研究表明可以通过中心管理控制器统一调度^[15] 来提升客户端 QoE, 其关键指标在于稳定性、公平性以及高利用率. 稳定性要求客户端尽可能请求相近或者同比特率的视频块减少比特率切换次数, 保证平滑播放; 公平性则考虑多客户端设

备的性能以及观看视频内容保证他们平等共享带宽资源;高利用率则是指所有客户端对总带宽的利用率尽可能高,这就要求给每个客户端分配合理的带宽资源避免浪费.基于视频流传输结构特性,统一调度控制器既可以部署在服务器上,也可以部署在DASH客户端之间.

Nathan等人提出多客户端QoE公平调度框架Minerva^[14],避免各客户端长期处于QoE较低或较高的状态.该框架使用MPC作为决策算法,通过计算历史视频块平均QoE以及预测MPC视野块的QoE计算效用函数,在均分带宽的基础上添加权重来实现多客户端的带宽分配,保证各客户端的QoE公平性.

Spiteri等人通过在服务器上部署来解决可用带宽问题^[16].在服务器端进行比特率整形,通过码率整形器控制各客户端视频块的最终决策.这种方式大大增加了服务器端的开销,而且需要额外修改传输视频文件,可能违反DASH视频流设计标准.Altamimi等人利用服务器和客户端的协同解决多客户端的不公平性问题^[17],让服务器定期更新视频文件清单,结合多代理强化学习^[18]提出Dec-POMDP模型,使用强化学习寻找并发多客户端网络带宽分配的公平性解决方案.

以上所有研究尽管针对多客户端,但大多仅适用于数量不多的客户端场景,当客户端规模较大时无法保持ABR算法的性能.本文考虑对大规模客户端通过聚类缩小规模,然后在聚类客户端和服务器中间加入网络代理,统一对每一类客户端的带宽进行调度以保证所有客户端具有最大的总QoE.

3 视频流决策和多客户端聚类

在本节介绍视频流决策优化函数QoE以及多客户端的聚类.

3.1 QoE公式

QoE是用于评估视频流服务的体验质量的指标,具体而言就是从用户感知因素的角度衡量观看视频的舒适程度,例如当用户观看视频时会重视视频画质、视频卡顿时长以及次数、流畅程度等因素.越高的比特率对应更高质量的视频,卡顿时间越少对应更高的平滑度.考虑到不同用户侧重不同的QoE影响因素,为统一量化用户体验质量,采用视频块质量,缓冲时间以及切换平滑度作为衡量指标.视频块质量可以用比特率度量,缓冲时间则是由视频块下载时间与当前缓

冲器剩余可播放时间差值决定,切换平滑度取决于相邻两个视频块质量变化大小,则QoE的计算公式如下:

$$QoE = \sum_{i=1}^N q(R_i) - \mu \sum_{i=1}^N S_i - \sum_{i=1}^{N-1} |q(R_{i+1}) - q(R_i)| \quad (1)$$

其中, $q(R_i)$ 指第*i*个视频块的视频质量, S_i 是缓冲时间,最后一项则代表切换平滑度.

对于大规模客户端(假定客户端数目为*M*)而言,优化目标变为最大化所有客户端的总QoE:

$$\max \sum_{m=1}^M QoE_m \quad (2)$$

3.2 模型预测控制算法

模型预测控制算法可以通过预测未来几步的环境变量,然后根据预测结果解决一个确切的优化问题.考虑到视频流中的ABR算法需要对预测带宽以及缓冲器占用综合考虑进行决策,故而可以将模型预测控制算法用作比特率决策.对于多客户端场景,MPC控制器可以通过调整QoE影响因素参数以及视野适用于不同类型的客户端.

DASH视频流采用MPC进行决策的过程如下:当前客户端请求下载视频块*i*时,首先通过带宽预测器对前几个视频块的带宽取均值得到预测带宽 $\widehat{T}_{[i,i+L]}$,结合当前缓冲器占用情况 B_i 以及上一个视频块的比特率 R_{i-1} 计算未来*L*个视频块(未来视野)的预期QoE,通过比较不同的比特率选择方式得到的QoE求最大值,并给出预下载视频块的最佳比特率决策:

$$R_i = f_{mpc}(\widehat{T}_{[i,i+L]}, R_{i-1}, B_i) \quad (3)$$

3.3 多客户端聚类

对于大规模客户端视频流,由于客户端上的ABR算法根据下载视频请求时的预测带宽以及缓冲器占用情况进行决策,故而可以采用K均值聚类算法^[19]进行聚类,将上述两个数值作为每一个客户端的坐标,选定*k*个聚类中心并通过计算聚类内平方和(within-cluster sum-of-squares, WCSS)^[20]进行划分,把具有相似分布的客户端聚在一起记为一个簇,将所有客户端分成*k*类,对每一类客户端分配相同的网络带宽和相同的比特率决策.

为了确定K均值聚类算法选取的客户端数目*k*,采用拐点法进行分析.拐点法的衡量指标是数据的误差平方累加和(sum of the squared error, SSE)^[21],可以

代表聚类效果的好坏,通过改变 k 值比较聚类效果,从中选择聚类效果最好的 k 作为客户端分类的代表个数.

$$SSE = \sum_{i=1}^k \sum_{a \in A_i} |a - c_i|^2 \quad (4)$$

其中, A_i 代表第 i 个簇, a 代表簇内的数据, c_i 代表所有数据的均值(质心). k 从 1 开始逐渐增大,客户端划分的类别在增加的同时也变得更加细致,所有客户端依据带宽以及缓冲器占用情况分类得到的簇聚合程度不断上升, SSE 的值逐渐变小,通常随着 k 的增大 SSE 变化幅度越小并逐渐趋于稳定,选取使 SSE 变化幅度从较大过渡到平缓时对应的 k 值即为所需的客户端聚类数.

在实际应用中,考虑带宽受限的情形,按与均分带宽的大小关系划分可分为 3 类(大于、小于、等于);考虑客户端缓冲器占用情况,将缓冲器划分为 3 类(无缓冲、满缓冲和介于二者之间). 综上结合实际数值将聚类客户端 k 值限定在 10 以内. 通过可视化方法计算多客户端不同 k 值下得到的 SSE 曲线斜率变化找寻突变点即为合适的聚类数目,此后随着 k 的增大聚类效果不再有较大变化.

4 多客户端缩放模型和带宽调度

本节介绍聚类客户端缩放模型以及出现带宽竞争现象时的调度策略.

4.1 多客户端缩放模型

在确定多客户端的聚类数后,采用均值聚类算法将大规模客户端缩小为 k 个客户端,并得到 k 个客户端对应的聚类中心点为 (T_k, B_k) . 每一类客户端的预测带宽和缓冲器占用为中心点对应数值,且其权重 α_k 由该类客户端包含的客户端数目在所有客户端总数的占比决定,记总客户端数目为 M ,总约束带宽为 T_{upper} ,第 k 类客户端包含的客户端数为 n_k ,则有:

$$\alpha_k = \frac{n_k}{M} \quad (5)$$

$$T_{overall} = \sum_{i=1}^k n_k T_k \quad (6)$$

如上把大规模客户端的带宽调度转化为缩放的 k 个客户端的带宽分配问题,则总带宽约束条件按缩放模型变为:

$$\sum_{i=1}^k \alpha_k T_k \leq \frac{T}{M} \quad (7)$$

于是可以参照上一节进行比特率决策以及带宽调度,对于第 k 个客户端下载第 i 个视频块时比特率决策 $R_{k,i}$ 和预测带宽为 \hat{T}_k :

$$R_{k,i} = f_{mpc}(\hat{T}_k, R_{k,i-1}, B_i) \quad (8)$$

$$\hat{T}_k = \begin{cases} k\alpha_k T_k, & \sum_{i=1}^k \alpha_k T_k \leq \frac{T}{M} \\ k\alpha_k T'_k, & \sum_{i=1}^k \alpha_k T_k > \frac{T}{M} \end{cases} \quad (9)$$

其中, T'_k 为客户端 k 采用带宽调度算法(第 4.2 节)得到的分配带宽.

4.2 聚类客户端带宽调度

由于真实环境中网络带宽持续波动,通过带宽预测器得到的预测带宽也会随之波动. 在共享带宽链路瓶颈下,当所有聚类客户端的预测带宽总数超过阈值时,需要对各客户端的带宽进行调度. 尽管通过聚类大大降低了客户端的规模,但由于聚类客户端之间互相制约,各 QoE 性能随不同的带宽分配而不同,为了根据网络状况动态进行带宽调度,需要比较多种分配方案下的总 QoE 大小来进行抉择. 本文采用模拟退火算法探索最佳带宽调度方案,性能与迭代次数有关,流程如图 2 所示,具体步骤如下:

(1) 参数设定: 设定多组预分配带宽初始值集合以及迭代次数 l .

(2) 初始化: 从初始值集合选择一组作为各聚类客户端的分配带宽,结合缓冲器占用信息送入 MPC 控制器计算最大总用户 QoE,记为 QoE_{best} ,对应的带宽分配方案记为 T'_k .

(3) 更新: 对分配带宽添加扰动,保证总带宽不超过阈值,计算新的分配方式下最大总用户体验质量 QoE_{new} ,与 QoE_{best} 比较,将最大的数值记为新的 QoE_{best} ,将对应的带宽分配方案更新为 T'_k . 继续添加扰动,重复更新步骤直至达到迭代次数上限.

(4) 返回第 (2) 步,选取新的初始分配带宽,继续执行步骤 (2)–(3) 直至初始值集合所有分配方案计算完毕,此时得到最大总用户体验质量 QoE_{best} 以及相应的最佳带宽分配 T'_k .

4.3 带宽调度算法分析

算法复杂度: 在实际应用中,考虑 M 个客户端,采

用 K 均值聚类算法过程中计算每个点与中心点的距离, 其算法复杂度为 d , 假定聚类数为 k , 迭代次数为 t , 则整个聚类算法的时间复杂度为 $O(t \times M \times k \times d)$, 相比较大规模客户端数目 M 而言, t 、 k 、 d 可视为常数. 在带宽调度过程中, 其迭代次数为 l , 客户端数目为 k , MPC 决策复杂度为 a , 为复杂度为 $O(l \times k \times a)$, 为保证调度性能, 迭代次数相对较大, k 、 a 视为常数. 故本文所用聚类调度算法复杂度为平方复杂度.

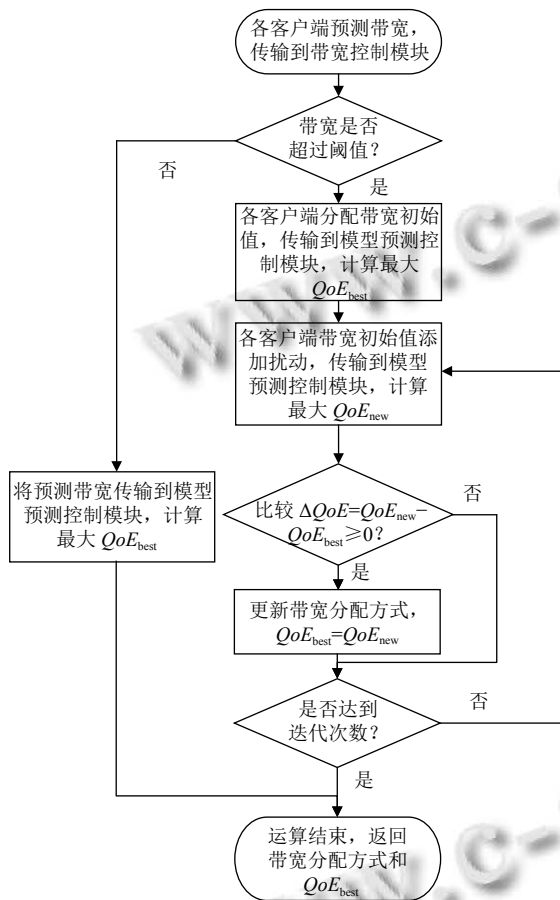


图2 多客户端带宽调度流程图

时间开销: k 值的不同选择会相应地增加执行时间, k 值越小执行时间越短. 为验证算法调度的实时性能, 统计 1-10 个客户端的单个视频块在是否产生调度情况下的运行时间, 在未发生调度时决策时间平均为 4.5 ms, 产生带宽调度的决策时间在 11-110 ms 范围变化, 而单个视频块的下载时间在 600-7 000 ms 范围变化.

本文提出的多客户端聚类带宽调度算法复杂度从原始的立方复杂度降为平方, 在执行时间上远小于视频块的下载时间, 故而在实际应用中可以满足实时性要求.

5 实验评估

在本节描述多客户端聚类调度算法的实施.

实验设置: 仿真实验所需的 100 个客户端分别采用来自 HSDPA 数据集的 100 条不同网络带宽轨迹模拟视频播放环境, 总带宽瓶颈设置为 100 Mb/s. 实验预先准备 10 个客户端和视频流服务器, 并从视频数据集 DASHDataset2014^[22] 中选取 10 种具有不同比特率编码的视频文件, 每一个视频文件包含 48 个视频块 (相同时长). 本文使用 Mahimahi^[23] 仿真视频流请求播放网络环境, MPC 算法的预测视野设为 3, 带宽调度算法迭代次数设为 100 次.

对照组设置: 选取目前多客户端带宽调度领域最新技术 Minerva 以及均分带宽方案进行对比. Minerva 方案通过计算 QoE 占比决定未来视频块的带宽分配, 而均分带宽方案则对所有客户端设置相同的带宽阈值.

客户端聚类数 k 值以及预分配带宽: 通过统计 100 个客户端当前请求视频块时的预测带宽以及缓冲器占用情况计算 SSE 变化幅度不再突变时对应的 K 值.

如图 3 所示, 从图中可以得知最为合适的聚类数 k 在 4 附近, 故我们实验选取 4 个客户端作为聚类客户端, 选取 4 个提供不同视频的服务器, 并通过预实验选定 3 组预设分配带宽, 如表 1 所示.

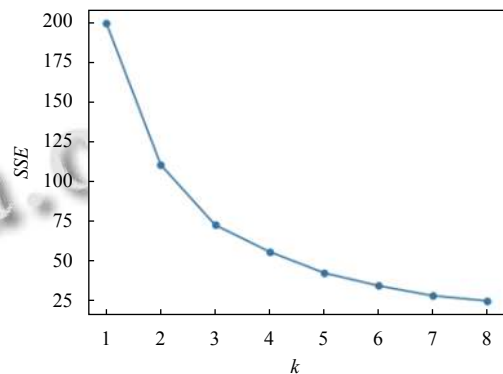


图3 SSE 与聚类数 k 的对应关系图

表1 各客户端预设带宽初始值 (Mb/s)

分组	客户端1	客户端2	客户端3	客户端4
1	1	1	1	1
2	0.8	0.8	1.2	1.2
3	0.5	0.5	1.5	1.5

多客户端的聚类: 为保证聚类算法的精准性, 对 100 个客户端当前视频块的预测带宽和缓冲器占用数据值进行标准化处理. 再根据上述所选 k 值设定聚类器数目为 4, 对多客户端采用 K 均值聚类, 各客户端聚

类结果如图4所示. 得到4个聚类客户端对应的聚类中心(包括预测带宽与缓冲器占用两个信息), 并统计每一类客户端所包含的客户端数量.

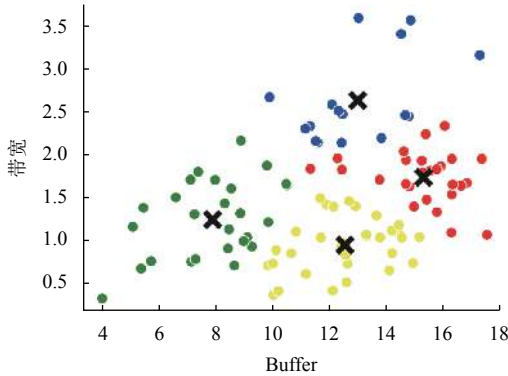


图4 各客户端聚类结果

对缩放的聚类客户端进行带宽动态调度, 计算各聚类客户端的最大 QoE, 再将每一类客户端得到的 QoE 乘以该类客户端包含的客户端数目, 然后累加得到最大总用户体验质量. 为消除客户端视频对实验的影响对照组同样采用4个客户端以及与聚类带宽调度方案相同的服务器, 每一种客户端从聚类带宽调度所用的100条网络带宽轨迹中各选取25条进行实验, 最后将所有客户端上的最大 QoE 累加得到 Minerva 和均分带宽方案下的总用户体验质量.

图5中3个子图分别为聚类调度、Minerva 以及均分带宽3种方案下各视频块对应的具体比特率选择、相应的带宽分配数值以及缓冲器占用情况. 相比于均分带宽, 聚类调度和 Minerva 方案可以更好地进行比特率决策, 各客户端的实际带宽在 1 Mb/s 左右持续上下波动, 两种方案的区别在于聚类调度保证总带宽的利用率而 Minerva 旨在维持各客户端 QoE 的公平. 而对于均分带宽方式, 由于客户端 1 Mb/s 的带宽限制, 当部分客户端未完全使用带宽而其他客户端带宽完全占用时产生带宽浪费, 较低的带宽利用率使得总用户 QoE 显著下降.

图6为以上3种方式下对各类客户端计算总 QoE 的数值, 如图所示通过聚类后进行带宽调度 (cluster1-4) 和 Minerva 方案 (Minerva1-4) 相比于均分带宽方式 (mean1-4), 每一类客户端上的 QoE 都有显著提升. 通过带宽动态调度提高了带宽资源的利用效率, 将部分客户端多余的带宽分配给其他客户端, 进而提高用户体验质量. 将各客户端的 QoE 累加可以得到两种方式下所有客户端的总用户体验质量, 具体数值见表2. 通过聚类调度方案获得的总 QoE 为 5 654.8, 目前最新技术 Minerva

方案获得的总 QoE 为 5 109.1, 均分带宽的总 QoE 值 2836.4, 就总体 QoE 而言, 聚类调度的 QoE 相比于均分方案提升 99.4%, 相比于 Minerva 质量提升 10.7%.

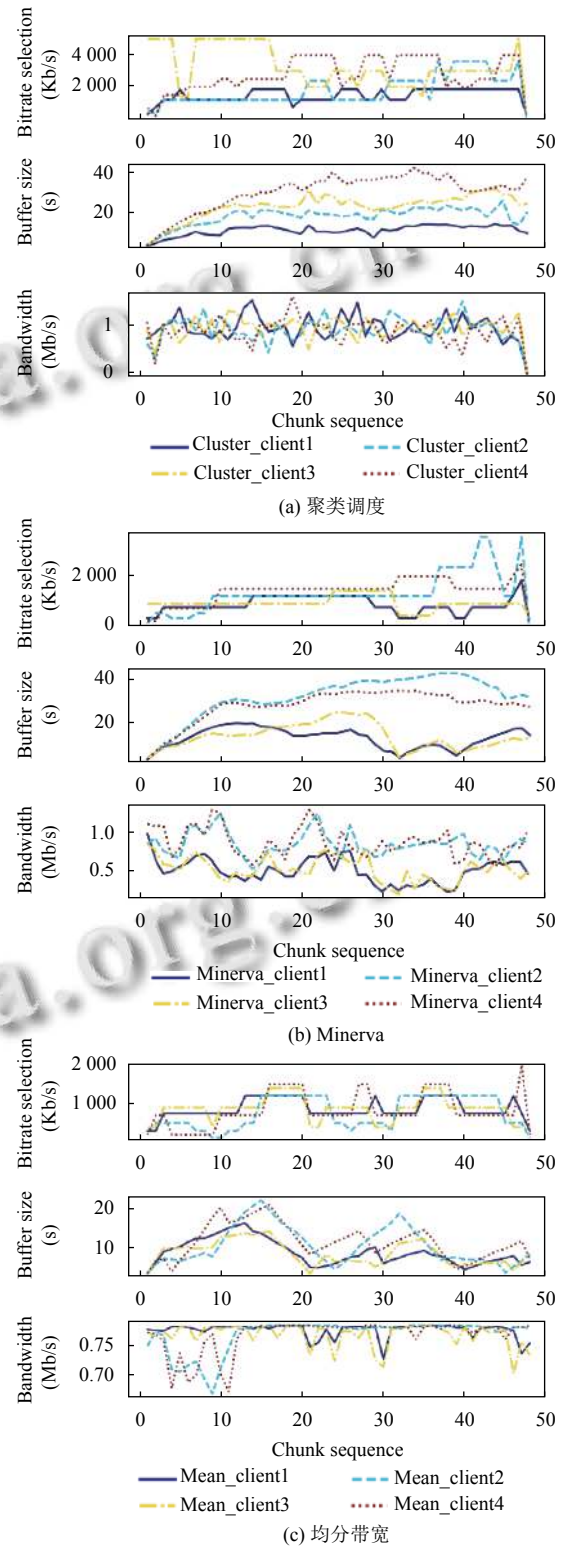


图5 不同方案下各客户端决策信息

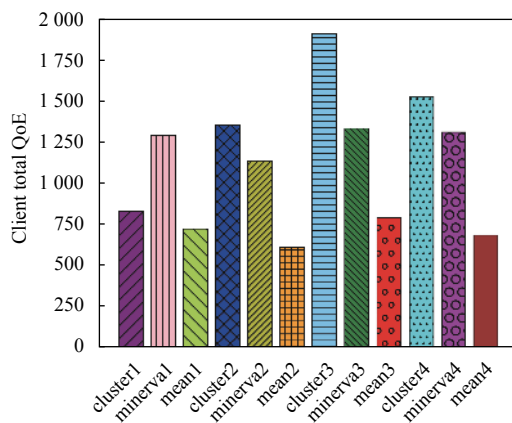


图6 不同方案各客户端 QoE 分布情况

表2 各类客户端上 QoE 总和

方案	客户端1	客户端2	客户端3	客户端4	总计
聚类调度	836.7	1361.5	1919.8	1536.8	5654.8
Minerva	1301.9	1145.2	1342.7	1319.3	5109.1
均分带宽	729.3	618.5	796.3	692.3	2836.4

尽管 Minerva 方案相比于均分带宽性能大大改善,但从总体用户体验质量角度出发,不能实现总 QoE 的最大化,其考虑的 QoE 公平性在实际应用中具有很大的局限性,例如选用不同设备观看视频其观看体验有一定差异,本文所提方案从总 QoE 出发,通过聚类和带宽调度保证资源的最大化利用,提供极致的体验质量,相比于传统的均分带宽, QoE 获得显著提高。

6 结论与展望

采用 K 均值聚类算法降低规模,把问题缩小为几个客户端之间的带宽调度和比特率决策问题,保证带宽调度及时性。本文选取 100 个客户端进行仿真实验并通过与 Minerva 和均分带宽的方案对比证明该方案在 QoE 上的显著提升。然而,多客户端调度受预测带宽的影响很大,错误的预测会影响到各客户端的具体调度以及 QoE 决策。此后的研究工作是改进带宽预测方式,利用 LSTM 确保带宽预测的精准度。此外,可以根据不同视频播放类型定义不同权重 QoE 指标。

参考文献

- Forecast G. Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022. Update, 2019. 1–36.
- Stockhammer T. Dynamic adaptive streaming over HTTP: Standards and design principles. Proceedings of the 2nd Annual ACM Conference on Multimedia Systems. New York: ACM, 2011. 133–144.
- Sani Y, Mauthe A, Edwards C. Adaptive bitrate selection: A survey. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2985–3014.
- Li Z, Zhu XQ, Gahm J, et al. Probe and adapt: Rate adaptation for HTTP video streaming at scale. IEEE Journal on Selected Areas in Communications, 2014, 32(4): 719–733. [doi: 10.1109/JSAC.2014.140405]
- Xie XF, Zhang XY, Kumar S, et al. piStream: Physical layer informed adaptive video streaming over LTE. Proceedings of the 21st Annual International Conference on Mobile Computing and Networking. New York: ACM, 2015. 413–425.
- Xiao MB, Swaminathan V, Wei S, et al. Dash2M: Exploring http/2 for internet streaming to mobile devices. Proceedings of the 24th ACM international conference on Multimedia. New York: ACM, 2016. 22–31.
- Miller K, Al-Tamimi AK, Wolisz A. QoE-based low-delay live streaming using throughput predictions. ACM Transactions on Multimedia Computing, Communications, and Applications, 2017, 13(1): 1–24.
- Huang TY, Johari R, McKeown N, et al. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. Proceedings of the 2014 ACM Conference on SIGCOMM. New York: ACM, 2014. 187–198.
- Mueller C, Lederer S, Grandl R, et al. Oscillation compensating dynamic adaptive streaming over http. 2015 IEEE International Conference on Multimedia and Expo (ICME). Turin: IEEE, 2015. 1–6.
- Yadav PK, Shafiei A, Ooi WT. Quetra: A queuing theory approach to dash rate adaptation. Proceedings of the 25th ACM International Conference on Multimedia. New York: ACM, 2017. 1130–1138.
- Spiteri K, Uргаonkar R, Sitaraman RK. BOLA: Near-optimal bitrate adaptation for online videos. IEEE/ACM Transactions on Networking, 2020, 28(4): 1698–1711.
- Yin XQ, Jindal A, Sekar V, et al. A control-theoretic approach for dynamic adaptive video streaming over HTTP. Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. New York: ACM, 2015. 325–338.
- Mao HZ, Netravali R, Alizadeh M. Neural adaptive video streaming with pensieve. Proceedings of the Conference of the ACM Special Interest Group on Data Communication. New York: ACM, 2017. 197–210.
- Nathan V, Sivaraman V, Addanki R, et al. End-to-end

- transport for video QoE fairness. Proceedings of the ACM Special Interest Group on Data Communication. New York: ACM, 2019. 408–423.
- 15 Seufert M, Egger S, Slanina M, *et al.* A survey on quality of experience of HTTP adaptive streaming. IEEE Communications Surveys & Tutorials, 2015, 17(1): 469–492.
- 16 Spiteri K, Sitaraman R, Sparacio D. From theory to practice: Improving bitrate adaptation in the DASH reference player. ACM Transactions on Multimedia Computing, Communications, and Applications, 2019, 15(2S): 67.
- 17 Altamimi S, Shirmohammadi S. Client-server cooperative and fair DASH video streaming. Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video. New York: ACM, 2019. 1–6.
- 18 Foerster J, Nardelli N, Farquhar G, *et al.* Stabilising experience replay for deep multi-agent reinforcement learning. Proceedings of the 34th International Conference on Machine Learning. Sydney: PMLR, 2017. 1146–1155.
- 19 Yang W, Long H, Ma LH, *et al.* Research on clustering method based on weighted distance density and K-means. Procedia Computer Science, 2020, 166: 507–511. [doi: [10.1016/j.procs.2020.02.056](https://doi.org/10.1016/j.procs.2020.02.056)]
- 20 Li XJ, Liang W, Zhang XP, *et al.* A cluster validity evaluation method for dynamically determining the near-optimal number of clusters. Soft Computing, 2020, 24(12): 9227–9241. [doi: [10.1007/s00500-019-04449-7](https://doi.org/10.1007/s00500-019-04449-7)]
- 21 Nainggolan R, Perangin-angin R, Simarmata E, *et al.* Improved the performance of the K-means cluster using the Sum of Squared Error (SSE) optimized by using the elbow method. Journal of Physics: Conference Series, 2019, 1361: 012015. [doi: [10.1088/1742-6596/1361/1/012015](https://doi.org/10.1088/1742-6596/1361/1/012015)]
- 22 Lederer S, Müller C, Timmerer C. Dynamic adaptive streaming over HTTP dataset. Proceedings of the 3rd Multimedia Systems Conference. New York: ACM, 2012. 89–94.
- 23 Netravali R, Sivaraman A, Das S, *et al.* Mahimahi: Accurate record-and-replay for HTTP. 2015 USENIX Annual Technical Conference. Santa Clara: USENIX Association, 2015. 417–429.