

# 一种基于缓冲区阈值调整的动态自适应码率选择算法<sup>\*</sup>

曾俊凯<sup>1</sup>, 李陶深<sup>1,2\*\*</sup>, 葛志辉<sup>1,3</sup>, 郑东<sup>4</sup>

(1. 广西大学计算机与电子信息学院, 广西南宁 530004; 2. 南宁学院, 广西南宁 530200; 3. 广西高校并行与分布式计算技术重点实验室, 广西南宁 530004; 4. 广西大学机械工程学院, 广西南宁 530004)

**摘要:** 基于 HTTP 的动态自适应流媒体(Dynamic adaptive streaming over HTTP, DASH)是一种可利用 Web 服务器提供在线高质量视频流的自适应码率流技术, 决定了视频播放性能。传统动态自适应码率选择算法存在考虑因素单一、播放缓冲区长度受限、带宽不稳定等问题, 因此, 改进与优化动态自适应码率选择算法十分必要。针对客户端缓冲区长度受限和带宽不稳定造成的视频播放卡顿现象, 综合考虑网络带宽和客户端缓冲区因素, 提出一种基于缓冲区阈值调整的动态自适应码率选择算法(Dynamic adaptive rate selection algorithm based on buffer threshold adjustment, BT-DARA)。该算法首先通过获取不同视频段大小和下载速率, 计算网络吞吐量并对视频下载时间进行预测, 然后结合缓冲区视频片段长度状态进行码率选择。特别地, 在视频播放过程中对不同阶段采取不同的视频码率选择策略, 并在客户端缓冲区达到临界值时动态调整缓冲区阈值参数, 以此增加缓冲区长度, 减少码率切换次数, 确保视频播放的稳定性。实验结果表明, 该算法能提高视频播放的码率, 保证视频播放质量和稳定性, 减少了视频开始播放的启动时间, 为用户提供良好的视频播放体验。

**关键词:** 流媒体 动态自适应 缓冲区阈值 自适应算法

中图分类号: TP391 文献标志码: A 文章编号: 1005-9164(2019)03-0276-07

## 0 引言

近几年来, 基于 HTTP 视频流服务的使用一直在增加, 许多视频网站开始采用 HTTP 协议进行视频流传送。采用 HTTP 传输视频可以使用标准的 HTTP 服务器, 并能通过内容分发网络(Content delivery network, CDN)传送视频, 同时能避免网络地址转换(Network address translation, NAT)和防火墙等问题, 使用灵活、可靠且更加容易扩展部署。基

于 HTTP 的动态自适应流媒体技术(Dynamic adaptive streaming over HTTP, DASH)是苹果、微软等公司推出的一种自适应码率流技术, 可通过传统的 HTTP Web 服务器提供在线的高质量媒体内容流<sup>[1]</sup>。为解决各厂商 DASH 技术方案不兼容的情况, 国际动态图像专家组(Moving Pictures Experts Group, MPEG)于 2011 年发布了动态自适应流媒体技术的国际标准 MPEG-DASH<sup>[2]</sup>。

DASH 技术在视频播放期间可以根据网络状态

<sup>\*</sup> 国家自然科学基金项目(61363067)和广西自然科学基金项目(2012GXNSFAA053226)资助。

### 【作者简介】

曾俊凯(1992—), 男, 硕士研究生, 主要从事流媒体、算法研究。

### 【\*\*通信作者】

李陶深(1957—), 男, 教授, 博士生导师, 主要从事无线网络、分布式数据库、网络计算与安全等领域研究, E-mail: tshli@gxu.edu.cn。

### 【引用本文】

DOI: 10.13656/j.cnki.gxkx.20190618.002

曾俊凯, 李陶深, 葛志辉, 等. 一种基于缓冲区阈值调整的动态自适应码率选择算法[J]. 广西科学, 2019, 26(3): 276-282.

ZENG J K, LI T S, GE Z H, et al. A dynamic adaptive rate selection algorithm based on buffer threshold adjustment [J]. Guangxi Sciences, 2019, 26(3): 276-282.

动态调整视频质量,确保了能够以较高的码率流畅播放视频,从而提高用户的体验质量(Quality of experience, QoE)。DASH 服务器将原始视频切片成固定时长(通常 2~4 s)的一个或多个视频片段,并使用 HTTP 传送到客户端<sup>[3]</sup>。视频媒体表示描述(MPD)文件描述了片段信息(如每段视频的持续时间、URL、视频分辨率、码率等级等),并可根据用例以不同的方式组织。在客户端,DASH 播放器首先下载并解析 MPD 文件以检索媒体信息,然后根据自身性能以及网络状况、缓冲区参数和所处的播放阶段调用动态自适应码率选择算法来选择合适码率的视频片段下载并播放。在视频播放过程中,如果当前的平均下载速率大于视频的比特率,则切换到高级比特率的视频播放;相反,如果平均下载速率小于视频的比特率,则切换到低级比特率的视频播放。DASH 技术兼顾了视频质量和用户观看视频的流畅性。显然,在 DASH 技术中,选择播放何种比特率的视频的决策引擎是位于客户端的,客户端的核心在于自适应码率选择算法。因此,研究有效的自适应码率算法对于提升视频播放的质量和用户观看视频的流畅性是至关重要的<sup>[4]</sup>。

MPEG-DASH 标准仅定义了 MPD 文件和视频段的格式,对于客户端自适应码率选择算法并没有做出规定<sup>[5]</sup>。因此,许多研究人员把 DASH 技术的研究重点放在了自适应码率选择算法上。Liu 等<sup>[6]</sup>提出了一种基于 HTTP 的自适应码率算法,该算法不依靠任何传输层的参数(如 RTT 和丢包率等),只是使用 HTTP 吞吐量平均值来判断当前视频比特率是否匹配网络带宽容量,采用渐进式的方法改变视频码率。但是该算法只是根据网络状况对码率进行切换,忽略了缓冲区等因素对视频播放的影响。Akhshabi 等<sup>[7]</sup>从缩短视频初始化时间、平滑视频码率切换以减少较大的波动、使用平均值来抵消带宽剧烈变化的影响等 3 个方面出发,提出一种新的自适应算法,可以保证播放的稳定性,但是对网络变化的反应不够迅速,影响了视频质量。Huang 等<sup>[8]</sup>提出一种基于客户端缓存的自适应码率选择算法,算法将吞吐量估计限制到初始阶段,并采用视频数据缓冲池来解决网络带宽和视频段大小的变化,同时还根据下载的视频段的大小变化情况来动态更新缓冲池容量。然而,由于视频段的大小变化取决于若干因素,如编码方案、视频内的帧改变数目以及视频的内容,因此难以根据先前数据来比较准确地预测出视频段的大小。Kumar

等<sup>[9]</sup>提出一种自适应视频流缓冲感知率选择算法,通过最大化平均选择比特率来增强观看者的体验质量(Quality of experience, QoE)。张洋洋等<sup>[10]</sup>将基于大偏差原理的下溢概率估计模型引入到客户端缓冲区占有率的分析上,提出一种基于缓冲区下溢概率的码率自适应算法,但是该算法对带宽资源预测方面考虑不够充分。潘鹏程等<sup>[11]</sup>将缓冲区分成上溢区域、下溢区域、重置区域和均衡区域,提出一种基于缓冲区的码率控制算法,通过滑动窗口技术平滑处理不同视频分片的下载时间,根据缓冲区和网络带宽的当前状态采用不同的码率选择策略,以便使缓存尽最大可能地维持在均衡区内,降低码率的切换次数。Venkata 等<sup>[12]</sup>提出一种基于吞吐量驱动的速率自适应算法,以获得平滑的回放连续性为目标,通过将估计的缓冲水平选择表示质量,在短期吞吐量波动的情况下避免了客户端的媒体播放器的缓冲区溢出和下溢。Juluri 等<sup>[13]</sup>提出了一种基于自适应感知缓存框架的码率自适应选择方案,通过改善高速缓存上的预取和使用自适应码率选择方案的知识来估计下一段请求,实现优化缓存命中并最小化 DASH 视频流服务的目的。夏勃等<sup>[14]</sup>对缓冲区进行分级,并采用滑动窗口技术对带宽进行平衡处理,视频播放比较稳定,但视频码率提升较慢。Miller 等<sup>[15]</sup>提出的算法使用平均下载速率作为输入,通过将播放分为两个阶段来避免频繁的码率切换:在快速启动阶段期间积极地增加视频码率,在视频播放阶段限制码率切换事件以提供稳定的播放体验。但是在开始阶段积极增加码率可能导致缓冲时间变长,影响视频观看体验。Juluri 等<sup>[16]</sup>提出一种片长感知的码率选择算法,将视频划分为大小不一的更多分片,通过综合分析分片大小的变化、评估预测网络带宽和缓冲区占有率,预测出下一片下载的时间。

从以上技术研究分析可以看出,现有研究主要集中在网络带宽状态和客户端缓冲区占用率大小对视频码率的影响,提出的大多数码率自适应算法都是使用下载速率和当前缓冲器占用率来为下一段选择最合适的码率,未考虑视频段大小对码率选择的影响,在网络状态发生变化时不能较好应对。大多数算法在整个视频播放的不同阶段,缓冲区阈值参数是保持不变的,固定的缓冲区阈值可能会导致视频播放启动偏慢和播放时视频码率切换较多,进而影响视频观看体验,且可能会发生缓冲区溢出等问题<sup>[17]</sup>,进而会影响算法的性能。针对这些问题,本研究提出一种基于

缓冲区阈值调整的动态自适应码率选择算法(Dynamic adaptive rate selection algorithm based on buffer threshold adjustment, BT-DARA)。该算法通过增强视频 MPD 文件来预测视频段下载时间,并结合带宽下载速率和缓冲区状态两个因素来选择视频码率。在视频播放过程中采取分阶段选择视频码率策略,在视频播放启动阶段稳步提高视频码率,减少缓冲时间,保证视频播放快速开始;在启动阶段结束后,采取积极的码率提升策略,提高视频播放质量。然后在缓冲区占有率处于较高水平时动态调整缓冲区阈值参数,以此来增加缓冲区长度,减少码率切换次数,确保视频播放的稳定性。

## 1 基于缓冲区阈值调整的码率选择算法

### 1.1 系统模型

本文研究的 HTTP 动态自适应流媒体系统架构模型如图 1 所示。在该模型中,DASH 服务器使用特定工具将原始视频切片成固定时长的视频片段,并将每个视频片段转换成不同码率的文件,从而形成多个相同内容不同码率的视频片段,同时生成视频媒体表示描述(MPD)文件。在客户端,DASH 播放器首先下载并解析 MPD 文件以检索媒体信息,然后根据网络状况、缓冲区参数和所处的播放阶段调用动态自适应码率选择算法(BT-DARA)来选择合适码率的视频片段下载播放。

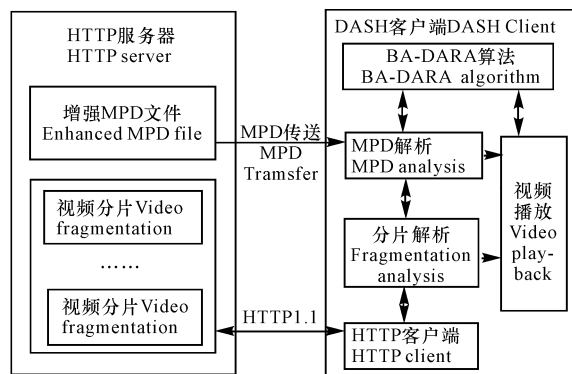


图 1 HTTP 动态自适应流媒体系统架构

Fig.1 HTTP dynamic streaming media system architecture

在本研究改进的动态自适应码率选择算法(BT-DARA)中,对传统的自适应流媒体系统中的 MDP 文件进行了增强,将视频段的大小信息加入到 MPD 文件中。增强的 MPD 文件使得客户端可以获得确切的视频段大小,从而能在网络带宽预测的基础上更准确地估计视频段下载时间。BT-DARA 算法将利用预测的带宽下载速率和当前的客户端缓冲区占用

率进行判断,选择合适的视频码率。

### 1.2 网络带宽估计

为了使码率选择算法能够选择到适当的视频码率,通常需要测量整个视频播放过程的吞吐量,进而估计出网络的当前带宽。整个视频播放过程由下载到的视频片段序列组成。本研究的 BT-DARA 算法采取了计算吞吐量平均值的方式来测量网络的吞吐量,这样做的目的是避免测量时网络带宽瞬时变化带来的对吞吐量计算的影响。被下载的视频片段可能因码率不同而具有不同的大小,因此采用分配与视频片段大小成比例的权重( $W_1, W_2, \dots$ )的方法来预测下一个视频片段的下载速率。这些权重和相应的视频片段下载速率( $d_1, d_2, \dots$ )用于计算加权平均下载速率。 $n$  个下载视频片段的加权平均下载速率由下式给出:

$$H_n = \frac{\sum_{i=1}^n W_i}{\sum_{i=1}^n \frac{W_i}{d_i}}, \quad (1)$$

其中,  $H_n$  表示前  $n$  个视频片段的平均下载速率,由前  $n$  个视频片段的总大小除以下载  $n$  个视频片段所用时间得到,并通过  $W_{n+1}/H_n$  预测下载下一个视频片段的时间。

### 1.3 算法的设计与描述

BT-DARA 算法的设计思想:1)将视频段大小信息加入到 MPD 文件中,通过增强视频 MPD 文件使客户端能较准确地预测下一个视频片段的下载时间;2)综合考虑视频段下载时间和客户端缓冲区状态,对客户端缓存划分区间并在视频播放的不同阶段采用不同的码率切换策略:在视频播放启动阶段稳步提高视频码率,以此减少缓冲时间,保证视频播放快速开始;启动阶段结束后则采取积极的码率提升策略,以此提高视频播放质量;3)根据网络状态和缓冲区占有率情况通过动态调整缓冲区阈值参数,增加缓冲区长度,以此减少码率切换次数,确保视频播放的稳定性。下面对 BT-DARA 算法进行详细描述。

假设 BT-DARA 算法运行时从视频段集合  $R$  中选出最合适码率的视频段,客户端下载每个段都放在大小为  $B_{\max}$  的缓冲区中。假设参数  $B_{\text{curr}}$  表示当前缓冲区占用率,参数  $I$  表示缓冲区的阈值,参数  $B_a$  表示较小的缓冲区占用率,参数  $B_\beta$  表示较大的缓冲区占用率,其大小关系如图 2 所示。这些阈值参数是根据段的数量来定义的。

客户端是在下载每个视频段后调用 BT-DARA

算法以确定下一个视频段的码率。为了保证视频播放的稳定性, BT-DARA 算法在运行过程中将根据网络状态和缓冲区占用率动态地调整各个缓冲区参数, 并根据以下 5 个阶段进行动态选择视频码率。

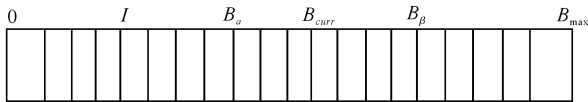


图 2 缓冲区阈值参数示意图

Fig. 2 Buffer threshold parameter diagram

1) 在视频开始播放时(或者在由于缓冲区下溢而中断之后), 缓冲区占用率低于缓冲区阈值  $I$  时, 即  $B_{curr} \leq I$ , 为快速启动阶段。在这一阶段, 算法将选择最低码率, 目的是尽可能减少视频播放开始时间, 保证用户在最短时间内就能观看视频。此外, 延迟因子  $\delta$  被设置为 0, 使得下一段被立即下载。最小化视频播放启动时间对于防止用户放弃视频会话是非常重要的。

2) 一旦缓冲器占用率不断增加且满足  $I < B_{curr} \leq B_\alpha$ , 算法进入加法增加阶段。在这一阶段, 算法下载下一个视频片段所需时间对应于当前比特率由下式给出:

$$W_{n+1}^{curr} / H_n. \quad (2)$$

算法在这一阶段将采用保守策略, 使视频片段码率以单个步长逐渐递增, 以避免缓冲区占用率低于  $I$ 。具体的操作过程中, 当视频播放处于加法增加阶段, 即当前缓冲区占用  $B_{curr}$  低于  $B_\alpha$  时, 如果下载高级码率  $r^{curr}$  的视频片段所需的时间小于  $B_{curr} - I$ , 则选择下一级码率; 否则选择当前码率。在视频播放期间, 如果按当前码率下载下一个段视频所花费的时间大于  $B_{curr} - I$ , 说明选择当前码率下载的效果不好, 应选择码率更小的视频片段。通过使用加法增加策略, 视频质量逐步提升, 从而避免码率的快速提升造成缓冲区占用率处于较低水平的问题。

3) 当  $B_\alpha < B_{curr} < B_\beta$  时, 算法进入主动切换阶段, 此时缓冲区占用率  $B_{curr}$  处于最优的缓冲区占用区间。在该阶段中, BT-DARA 算法将采用更加乐观的策略, 更倾向于选择高码率视频段。具体的做法是: 基于当前网络带宽和缓冲区占用率, 选择大于或等于当前码率的最合适码率。这种方式在确保视频质量稳定性的同时, 也逐步提高视频的播放质量。为此, 算法在决策过程中利用视频段大小信息, 确保即使在视频片段大小波动情况下也能够更准确地预测片段的下载时间。

4) 当网络条件有利并且缓冲区占用率  $B_{curr}$  大于  $B_\beta$  时, 算法处于延迟下载阶段。在这一阶段, BT-DARA 算法选择最适用于当前网络带宽的码率。但不是立即发送对下一个视频片段的请求, 而是等待直到缓冲区占用率下降到  $B_\beta$  时才发送对视频段的请求。延迟下载的目的是限制视频缓冲器中视频片段的总数, 以避免不必要的下载, 防止用户过早地退出观看视频。

5) 当算法运行到延迟下载阶段时, 确保视频稳定播放成为最重要的任务, BT-DARA 算法进入对缓冲区阈值进行调整的阶段。具体的做法是: 算法检测当前缓冲区视频段占有率是否大于  $B_\alpha$  并且当前选择的视频码率是否为最高码率, 如果是, 说明已经度过了视频播放开始阶段, 即用户已经观看了 1~2 min 视频, 可能对该视频比较感兴趣, 用户退出观看视频的可能性较小, 此时可以适当增加缓冲区阈值参数  $B_\alpha$ 、 $B_\beta$ 、 $B_{max}$  的值(例如算法中将 3 个参数值增加 5), 这样使得缓冲区长度增长, 能够缓存更多的视频片段, 有助于视频的稳定播放, 提高抵御恶劣网络环境的能力。当视频播放出现卡顿现象时, 可将缓冲区阈值参数  $B_\alpha$ 、 $B_\beta$ 、 $B_{max}$  恢复到初始状态, 使缓冲区视频段数量尽快恢复正常。

假设可用视频码率列表为  $R = \{r^{\min}, \dots, r^i, \dots, r^{\max}\}$ ,  $n$  为最近下载的段号,  $r^{curr}$  为最近下载的段的码率,  $H_n$  为前  $n$  段加权平均下载率,  $\delta$  为延迟因子, 前权重  $W_{n+1} = \{W_{n+1}^{\min}, \dots, W_{n+1}^i, \dots, W_{n+1}^{\max}\}$  对应码率段的大小为  $\{r^{\min}, \dots, r^m, \dots, r^{\max}\}$ 。缓冲区阈值  $I$  默认为 2,  $B_\alpha$ 、 $B_\beta$ 、 $B_{max}$  的初始值分别为  $B_\alpha = 5$ ,  $B_\beta = 10$ ,  $B_{max} = 12$ 。BT-DARA 算法的伪代码描述如下:

输入:  $n$ ,  $r^{curr}$ ,  $B_{curr}$ ,  $H_n$ ,  $W_{n+1}$ ,  $R$

输出:  $l_{n+1}$ : 算法执行完毕的结果, 即选择的视频码率

基本操作步骤:

**if**  $B_{curr} \leq I$  **then**

$$l_{n+1} = r^{\min}$$

**else**

**if**  $(W_{n+1}^{curr} / H_n) > B_{curr} - I$  **then**

$$l_{n+1} = \max\{r^i \mid r^i \in R, W_{n+1}^{curr} / H_n \leq B_{curr} - I, i \leq curr\}; \delta = 0;$$

**else if**  $B_{curr} \leq B_\alpha$  **then**

**if**  $(W_{n+1}^{curr+1} / H_n) < B_{curr} - I$  **then**

$$l_{n+1} = r^{curr} \uparrow \delta = 0;$$

**else**

$$l_{n+1} = r^{curr} \quad \delta = 0;$$

**else if**  $B_{curr} \leq B_{\beta}$  **then**  
 $l_{n+1} = \max\{r^i \mid r^i \in R, W_{n+1}^i / H_n \leq B_{curr} - I, i \geq curr\}; \delta = 0;$   
**else if**  $B_{curr} > B_{\beta}$  **then**  
 $l_{n+1} = \max\{r^i \mid r^i \in R, W_{n+1}^i / H_n \leq B_{curr} - B_{\alpha}, i \geq curr\}; \delta = B_{curr} - B_{\alpha};$   
**else**  
 $l_{n+1} = r^{curr}; \delta = 0;$   
**if**  $l_{n+1} = r^{max} \& \& B_{curr} > B_{\alpha} \& \& B_{\alpha} = 5 \& \& B_{\beta} = 10 \& \& B_{max} = 12$  **then**  
 $B_{\alpha} = B_{\alpha} + 5; B_{\beta} = B_{\beta} + 5; B_{max} = B_{max} + 5;$   
**else if**  $B_{curr} = 0 \& \& B_{\alpha} = 10 \& \& B_{\beta} = 15 \& \& B_{max} = 17$   
 $B_{\alpha} = B_{\alpha} - 5; B_{\beta} = B_{\beta} - 5; B_{max} = B_{max} - 5.$

#### 1.4 算法的性能验证

为了验证 BT-DARA 算法性能,我们在 Apache 上搭建了 DASH 服务器,并使用 Python 实现了一个模拟 DASH 客户端。通过这个客户端,我们能够模拟视频播放的整个过程,从而验证 BT-DARA 算法的性能。DASH 服务器运行 Ubuntu 14.04 操作系统上,主要实现视频编码转码功能、视频分片功能以及视频分发功能。模拟 DASH 客户端运行在 Window 7 系统上,由 MPD 解析器模块、视频段下载模块、视频缓冲器模块、码率选择模块等 4 个模块组成。其中,MPD 解析器模块用来下载并解析 MPD 文件以获得段信息(可用码率集,视频段 URL 和段大小信息);视频段下载模块负责从 DASH 服务器下载视频段;视频缓冲器模块则模拟视频播放器缓冲器;码率选择模块用于确定将要播放的视频码率。BT-DARA 算法在码率选择模块中实现。为了对比算法性能,我们也同时实现了 Liu 等<sup>[6]</sup>提出的码率选择算法(以下称为传统码率选择算法)。该算法主要从网络吞吐量的角度来选择视频码率,具有普遍性。

实验中,DASH 服务器和 DASH 客户端之间的带宽控制使用的是 Linux 中的 traffic control 工具。为了评估,视频源使用流行的开源电影 Big Buck Bunny,该视频的开源版本提供 20 种不同的表示形式。我们使用 DASHEncoder 对视频源进行编码、切片生成 20 种具有不同码率的 MPD 文件,视频的总长度大约为 596 s,并且每个表示被分割成 150 个段,每个段具有 4 s 的播放持续时间。仿真实验中使用的传输协议是 TCP,并且使用现有的 HTTP Web 服务器来传送视频信息。实验参数的设置为  $I=2, B_{\alpha}, B_{\beta}, B_{max}$  初始配置值为  $B_{\alpha}=5, B_{\beta}=10, B_{max}=12$ 。

实验的目的是对比分析传统码率选择算法和本研究的 BT-DARA 算法的性能。实验过程如下:首先由播放器使用视频 MPD 文件的 URL 进行初始化,再由 MPD 解析器模块下载并解析 MPD 文件以检索所有表示和视频段 URL 的列表。接下来,调用速率码率模块以确定合适码率的视频段。然后,所选择的片段由片段下载模块下载并放入视频缓冲器中。一旦视频缓冲器中的段的数量增加超过初始阈值,则模拟视频播放器从缓冲器中开始读取一个视频段的数据。

## 2 结果与分析

图 3 给出了服务器和客户端之间带宽为 4 Mbps 时传统码率选择算法和本研究算法的  $B_{\alpha}$  实验结果。传统码率选择算法在视频播放开始阶段(视频段 1~10)选择的码率就比本研究算法要大,并且增加较快,这是因为该算法主要考虑吞吐量因素来选择视频码率,在视频开始播放阶段没有对码率增加进行限制。视频播放到达稳定阶段后(视频段 10 之后),本研究算法也经历了加法增长和主动切换两个阶段,视频码率较为理想且客户端缓冲区达到了较高的水平。实验中,传统码率选择算法在 25,49,73 视频段时码率下降幅度较大,这是受到网络带宽变化的影响所致。尽管此时本研究算法的码率也有所下降,但幅度较小,总体上还是比较稳定的。在视频播放后半段,网络带宽持续下降,传统码率选择算法选择的码率也随之下降。由于本研究算法可动态调整缓冲区阈值参数,扩大了缓冲区,使得视频播放较为稳定。总体来

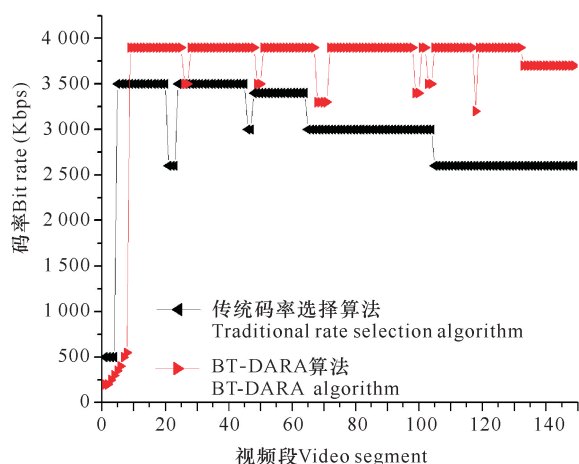


图 3 4 Mbps 网络带宽下传统码率选择算法与 BT-DARA 算法实验对比

Fig. 3 Comparison of traditional rate selection algorithm and BT-DARA algorithm under 4 Mbps network bandwidth

说,传统码率选择算法对于网络带宽的变化较为敏感,视频播放不太稳定且选择的码率相对较低,而本研究算法虽然在开始阶段码率增长相对较慢,但在播放阶段能提供较高的码率及稳定的播放,提高了视频播放质量。

图4是8 Mbps的网络带宽下传统码率选择算法与本研究算法实验对比结果。在网络带宽变大的情况下两个算法的表现均好于4 Mbps网络带宽。传统码率选择算法直接从较高码率的视频段开始播放视频,而本研究算法在开始阶段有一个明显的增长过程。在稳定播放阶段,本研究算法码率选择有一些波动,但还是要优于对比算法。在视频播放后段,两个算法表现都很稳定,但因为由较大的缓冲区支撑,本研究算法的平均码率明显高于传统码率选择算法。

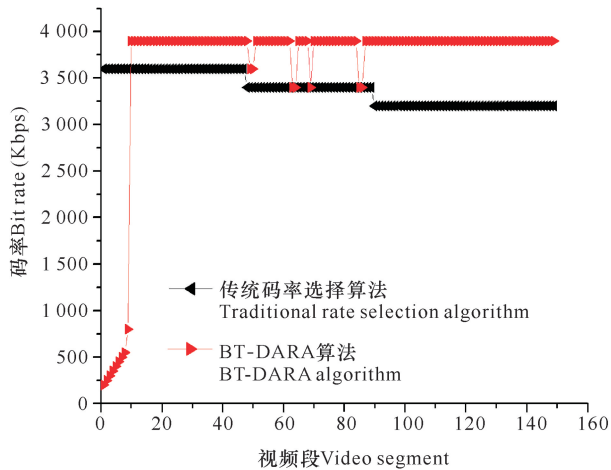


图4 8 Mbps网络带宽下传统码率选择算法与BT-DARA算法实验对比

Fig. 4 Comparison of traditional rate selection algorithm and BT-DARA algorithm under 8 Mbps network broadband

### 3 结论

本研究提出了一种基于缓冲区阈值调整的动态自适应码率选择(BT-DARA)算法。该算法通过增强视频MPD文件的表示,使视频段下载时间的预测更加准确;通过将带宽下载速率预测与缓冲区状态结合起来考虑,在视频播放过程中采取分阶段选择视频码率策略,动态地选择视频码率,稳定和提升了视频播放质量;通过动态调整缓冲区阈值参数,扩大了缓冲区的容量,减少了码率切换次数,确保了视频播放的稳定性。仿真实验结果表明,该算法能提高视频播放质量和稳定性,可以为用户提供良好的视频播放体验。下一步的研究工作将从两个方面去考虑:一是建立合适的视频播放的QoE评价体系,以便更加科学

和直观地评价视频播放质量;二是针对现有的缓冲区参数调整策略较为简单的问题,研究更为合理和有效的缓存参数调整策略,以进一步提高视频观看体验。

### 参考文献

- [1] MUELLER C, LEDERER S, TIMMERER C, et al. Dynamic adaptive streaming over HTTP/2.0 [C]// 2013 IEEE International Conference on Multimedia and Expo (ICME 2013). San Jose, CA, USA: IEEE, 2013: 1-6.
- [2] SODAGAR I. The MPEG-DASH standard for multimedia streaming over the internet [J]. IEEE MultiMedia, 2011, 18(4): 62-67.
- [3] KUA J, ARMITAGE G, BRANCH P. A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP [J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1842-1866.
- [4] AYAD I, IM Y, KELLER E, et al. A practical evaluation of rate adaptation algorithms in HTTP-BASED adaptive streaming [J]. Computer Networks, 2018, 133: 90-103.
- [5] BENTALEB A, TAANI B, BEGEN A, et al. A survey on bitrate adaptation schemes for streaming media over HTTP [J]. IEEE Communications Surveys & Tutorials, 2019, 21(1): 562-585.
- [6] LIU C, BOUAZIZI I, GABBOUJ M. Rate adaptation for adaptive HTTP streaming [C]// MMSys '11 Proceedings of the Second Annual ACM Conference on Multimedia Systems. San Jose, CA, USA: ACM Press, 2011: 169-174.
- [7] AKHSHABI S, NARAYANASWAMY S, BEGEN A C, et al. An experimental evaluation of rate-adaptive video players over HTTP [J]. Signal Processing: Image Communication, 2012, 27(4): 271-287.
- [8] HUANG T Y, JOHARI R, MCKEOWN N, et al. A buffer-based approach to rate adaptation: Evidence from a large video streaming service [J]. ACM SIGCOMM Computer Communication Review, 2015, 44(4): 187-198.
- [9] KUMAR M V P, RAVI K C, MAHAPATRA S. Design and implementation of a dynamic adaptive video streaming system with a buffer aware rate selection algorithm [C]// Image Analysis and Processing-ICIAP 2015. Lecture Notes in Computer Science, 2015, 9280(1): 227-238.
- [10] 张洋洋, 陈双武, 完颜许哲, 等. 基于HTTP的自适应视频传输算法[J]. 小型微型计算机系统, 2016, 37(7): 1572-1576.
- [11] 潘鹏程, 李国平, 王国中. 基于HTTP的流媒体自适应码率切换算法[J]. 电子测量技术, 2017, 40(9): 141-145.
- [12] VENKATA P K M, SUDIPTA M. A modulated

- throughput driven rate adaptation algorithm for dynamic HTTP streaming [C]// 2016 International Symposium on Wireless Communication Systems (ISWCS). Poznan, Poland; IEEE, 2016;143-147.
- [13] JULURI P, MEDHI D. Cache'n dash: Efficient caching for dash [C]// ACM SIGCOMM Computer Communication Review. ACM, London, UK, 2015, 45 (4): 599-600.
- [14] 夏勃,林浒,李鸿彬,等. 一种动态自适应 HTTP 流媒体码率切换算法[J]. 小型微型计算机系统, 2015, 36 (2):275-279.
- [15] MILLER K, QUACCHIO E, GENNARI G, et al. Adaptation algorithm for adaptive streaming over HTTP [C]// 2012 19th International Packet Video Workshop (PV). Munich-Garching, Germany; IEEE, 2012: 173-178.
- [16] JULURI P, TAMARAPALLI V, MEDHI D. SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP [C]// 2015 I. E. International Conference on Communication Workshop (IC-CW). London, UK: IEEE, 2015;1765-1770.
- [17] RAHMAN W U, CHUNG K. A novel adaptive logic for dynamic adaptive streaming over HTTP [J]. Journal of Visual Communication and Image Representation, 2017, 49:433-446.

## A Dynamic Adaptive Rate Selection Algorithm based on Buffer Threshold Adjustment

ZENG Junkai<sup>1</sup>, LI Taoshen<sup>1,2</sup>, GE Zhihui<sup>1,3</sup>, ZHENG Dong<sup>4</sup>

(1. School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China; 2. Nanning University, Nanning, Guangxi, 530200, China; 3. Guangxi Colleges and Universities Key Laboratory of Parallel and Distributed Computing Technology, Nanning, Guangxi, 530004, China; 4. School of Mechanical Engineering, Guangxi University, Nanning, Guangxi, 530004, China)

**Abstract:** Dynamic Adaptive Streaming over HTTP(DASH) is an adaptive rate stream technology that can provide online high-quality video streams using web servers, which determines video playback performance. There are some problems in the traditional dynamic adaptive rate selection algorithm, such as considering only a single factor, limited play buffer length and bandwidth instability. It is necessary to improve and optimize the dynamic adaptive rate selection algorithm. Aiming at the video playback carton caused by the limited buffer length and bandwidth instability of the client, and comprehensively considering the network bandwidth and client buffer factors, a dynamic adaptive rate selection algorithm based on buffer threshold adjustment (BT-DARA) is proposed. The algorithm first calculates the network throughput to predict the video download time by acquiring different video segment sizes and downloading rates, and then combines the buffer video segment length state to select the bit rate. In particular, different video bit rate selection strategies are adopted for different stages during video playback, and the buffer threshold parameter is dynamically adjusted when the client buffer reaches a critical value. Therefore, the algorithm can increase the buffer length, reduce the number of code rate switching, and ensure the stability of video playback. The experimental results show that the algorithm can improve the bit rate of video playback, ensure the quality and stability of video playback, reduce the startup time of video playback, and provide users with a good video playback experience.

**Key words:** streaming media, dynamic adaptive, buffer threshold, adaptive algorithm