



基于 SVC 与多网络接口的 DASH 调度算法

韩 煦, 张国强, 高 茜

(南京师范大学 计算机科学与技术学院, 南京 210023)

摘 要: 基于 HTTP 的动态自适应流媒体(DASH)技术是主流的互联网视频传输技术之一,可适应动态网络和异端客户端,然而传统 DASH 使用高级视频编码(AVC),存在编码块复用性低、码率波动性高等问题。为此,在终端设备具有多网络接口的环境下,以可伸缩视频编码(SVC)代替 AVC,提出一种基于视频块优先级的联合调度算法。通过 SVC 编码将整个缓冲区组织为一个包含时间维度和编码层次维度的二维空间,确定视频块在二维缓冲区的位置,将其放入待下载队列,并按照优先级对队列中的视频块进行重新排序。根据每个网络接口的带宽,将队列中的视频块分配给不同接口执行下载操作。实验结果表明,相比于单接口基于带宽和缓冲的调度算法,该算法的平均视频质量提高 46.04%,切换次数降低 36.84%,并且可使缓冲水平下降 44.11%。

关键词: 可伸缩视频编码;多网络接口;流媒体技术;DASH 技术;调度算法

开放科学(资源服务)标志码(OSID):



中文引用格式:韩煦,张国强,高茜. 基于 SVC 与多网络接口的 DASH 调度算法[J]. 计算机工程,2019,45(12):243-248.

英文引用格式:HAN Xu,ZHANG Guoqiang,GAO Qian. DASH scheduling algorithm based on SVC coding and multiple network interfaces[J]. Computer Engineering,2019,45(12):243-248.

DASH Scheduling Algorithm Based on SVC and Multiple Network Interfaces

HAN Xu,ZHANG Guoqiang,GAO Qian

(School of Computer Science and Technology,Nanjing Normal University,Nanjing 210023,China)

[Abstract] The Dynamic Adaptive Streaming over HTTP(DASH) is one of the mainstream Internet video transmission technologies,which can adapt to dynamic networks and heterogeneous clients. However,the traditional Advanced Video Coding(AVC) based DASH has many problems such as low reusing ratio of code block and high rate of oscillation. Under circumstances where the terminal equipments have multiple interfaces,this paper replaces AVC with Scalable Video Coding(SVC) and proposes a joint scheduling algorithm based on video block priority. First,the SVC organizes the whole buffer into a two-dimensional space,containing the time dimension and the coding hierarchy dimension. Then the video blocks in the two-dimensional buffer are located,put into the download queue and rearranged according to their priority. Finally,the video blocks in the queue are allocated to different interfaces to execute download operations according to the bandwidth of each network interface. Experimental results show that compared with the single-interface,bandwidth and buffer-based scheduling algorithm,the video quality of the proposed algorithm improves by 46.04% in average,the switch times reduces by 36.84% and the buffer level decreases by 44.11%.

[Key words] Scalable Video Coding(SVC);multiple network interfaces;streaming media technology;DASH technology;scheduling algorithm

DOI:10.19678/j.issn.1000-3428.0053020

0 概述

近年来,随着互联网视频行业的发展以及手机、平板电脑等移动终端数量的增加,使用移动设备进行视频上传、点播、直播的应用不断涌现,这提高了视频内容

在整个互联网中的流量占比。中国互联网信息中心(China Internet Network Information Center,CINIC)发布的《第 41 次中国互联网发展统计报告》显示,截至 2017 年 12 月,我国手机网民规模达 7.53 亿,其中网络视频使用率不断上升^[1]。据 CISIO VNI 预测,到

基金项目:国家自然科学基金(61772279,41771411);江苏省研究生科研与实践创新计划(KYCX18_1237);赛尔网络下一代互联网技术创新项目(NGII20150102)。

作者简介:韩 煦(1995—),男,硕士,主研方向为视频传输与控制;张国强(通信作者),教授、博士;高 茜,副教授、博士。

收稿日期:2018-10-29 **修回日期:**2018-12-19 **E-mail:**guoqiang@ict.ac.cn

2020 年,全球 IP 流量将增长近 3 倍,其中 IP 视频流量相对于全网流量的占比将从 2015 年的 70% 攀升至 82%,而且 CDN 网络会承载整个互联网的三分之二的流量^[2]。

目前,网络视频主要采用流媒体技术来为用户提供服务。流媒体指在互联网中使用流式传输技术传输连续的基媒体,如音频、视频或者多媒体文件^[3]。流式传输不要求客户端具有太大的缓冲容量,而是可以在带宽和存储受限的情况下边下载边播放服务器存储的大视频和音频文件。但是,互联网的高度动态性、移动接入设备的高度异构性等都对流媒体技术提出了巨大的挑战^[4]。随着视频清晰度的提升以及 VR 视频的出现,流媒体视频传输所需要的带宽越来越大,服务器和网络的负担也相应增加。对于网络接入商和内容服务商而言,合理分配带宽,处理海量用户的并发访问并提供令人满意的流媒体服务质量是一件极具挑战性的工作。

基于 HTTP 的动态自适应流媒体 (Dynamic Adaptive Streaming over HTTP, DASH) 传输技术在一定程度上解决了上述问题,是当前主流的流媒体传输技术之一。基于此,苹果、Adobe、微软等公司均实现了各自的方案。为了统一标准,动态图像专家组 (Moving Picture Experts Group, MPEG) 组织提出了 MPEG-DASH^[5-6] 的解决方案。

本文使用可伸缩视频编码 (Scalable Video Coding, SVC) 提升 DASH 的性能,针对具有多网络接口的终端设备,提出一种基于视频块优先级的联合调度算法,在 DASH 中利用 SVC 天然的分层优势,通过多个网络接口来优化用户端的播放质量,同时缓解服务器和网络的存储及带宽压力。

1 相关工作

基于高级视频编码 (Advance Video Coding, AVC) 的 DASH 已经有了商业化的应用,例如 Akamai 的 HD Streaming, Adobe 的 OSMF 以及 Netflix 等。研究结果表明,如果视频播放不能达到用户的要求,用户会很快结束视频的播放,这会为内容提供商带来很大的损失。因此,内容提供商一直致力于给用户提供高质量的视频服务。DASH 中的自适应码率 (Adaptive Bitrate, ABR) 切换算法是内容提供商最基本的优化视频服务质量的手段。

MPEG-DASH 标准框架内的 ABR 切换算法允许终端通过观测网络的状况,并根据自身的播放状况动态地自适应调整获取视频块质量来优化终端用户的体验质量 (Quality of Experience, QoE) 水平。一般来说,终端用户的 QoE 水平由视频播放的码率、视频播放期间的质量切换次数、视频播放期间的暂停次数等一系列因素共同决定。

早期的 ABR 切换算法被分为基于速率和基于缓冲 2 类。基于速率的 ABR 切换算法通过前一个视频块的下载情况来估算可用的网络带宽,然后选

择网络带宽支持的最大码率对视频块进行下载播放^[7-8]。例如,文献[9]通过对之前 5 个视频块的下载状况进行调和,得出网络状况的估测,然后根据这个估测值去选择下一个视频块的码率。然而,这些方法的有效性受制于在 HTTP 协议上进行网络带宽估测存在的偏差^[10]。许多研究致力于优化网络带宽的测量机制,但是在高度动态的网络中精确地预测网络带宽本身就是一个挑战^[11-12]。

基于缓冲的 ABR 切换算法在决定下一个视频块的码率时,仅考虑客户端的播放缓冲长度^[13-14]。这类算法的主要目标是在客户端维持一个相对稳定和安全的缓冲区长度,以此来降低视频暂停的可能性并提高视频的播放质量。例如,文献[14]提出一种新的基于缓冲的 ABR 切换算法 BOLA,使用李雅普诺夫优化针对一个具体的 QoE 指标进行优化选择,同时 BOLA 还支持当客户端怀疑即将会发生缓冲干枯时,丢弃一个正在下载的视频块去选择一个更低质量的视频块来防止缓冲干枯的发生。

上述 2 类算法都能在各自特定的场景下带来很好的 QoE 反馈。基于速率的 ABR 切换算法在链路状态稳定的情况下或在视频启动阶段可以发挥较好的效果,而基于缓冲的 ABR 切换算法在视频稳定播放期间或在网络状况波动较大的情况下可以做出更好的决定。因此,研究人员结合这 2 类技术以提高用户的 QoE 水平。例如, MPC 是一种综合考虑了网络带宽和客户端缓冲水平,然后对后续一组视频块进行一个多项 QoE 指标综合最大化的自适应调度模型^[15]。

尽管基于 AVC 的 DASH 调度算法得到了研究人员持续不断的优化,但是 AVC 本身固有的一些缺点仍然会造成一些问题:

- 1) 额外的存储开销和带宽需求。由于不同码率的文件都需要进行独立编码和存储,因此服务器不能对其相同的部分进行重用,带来了额外的存储开销。

- 2) 客户端需要维持一个较长的播放缓冲,以保证播放的稳定性,防止在网络剧烈抖动时导致客户端出现缓冲的干枯。

- 3) 在客户端处于稳定播放的阶段,调度算法会维持一个特定水平的缓冲,这时网络带宽的抖动会导致非最优的视频质量选择^[16]。

针对这些不足,本文将采用 SVC 代替 DASH 中传统的 AVC,利用 SVC 自身的优势来弥补 AVC 的固有缺点。

2 基于 SVC 与多网络接口的调度算法

2.1 问题描述

SVC 是 H. 264 标准中的一个扩展协议,可以对视频在时间、空间、质量上进行划分,输出多层的码流,包括基础层和增强层。其中,基础层的数据可以由解码器完全解码输出基本的视频内容,但是仅包含基础层的视频图像可能导致帧率、分辨率或视频质量较低。在网络状况较差的情况下,接收到的基础层码流可以保证解码端观看到流畅的视频。当网

络状况较好时,可以请求增强层的数据来提高帧率、分辨率和视频质量。

AVC 视频编码标准对于不同的终端要求,需要对一个内容进行多次编码,而 SVC 视频编码只需对内容进行一次编码便可提供一个单独的多层嵌套码流,从这个码流中可以抽取出不同帧率、分辨率或者比特率的码流来满足不同终端的要求^[17]。因此,在 SVC 的视频服务器端,只需要存储最高层次的视频码流即可满足不同层次视频文件的请求,而 AVC 的视频服务器则需要对所有层次的视频码流进行存储,因此,即使 SVC 编码在每一层都会比 AVC 编码多出 10% 的开销,但是其总的存储开销会大幅降低,具体对比如表 1 所示。

表 1 AVC 和 SVC 服务器开销对比

编码	Q1	Q2	Q3	总存储
AVC	800	1 150	1 500	3 450
SVC	800	1 265	1 800	1 800

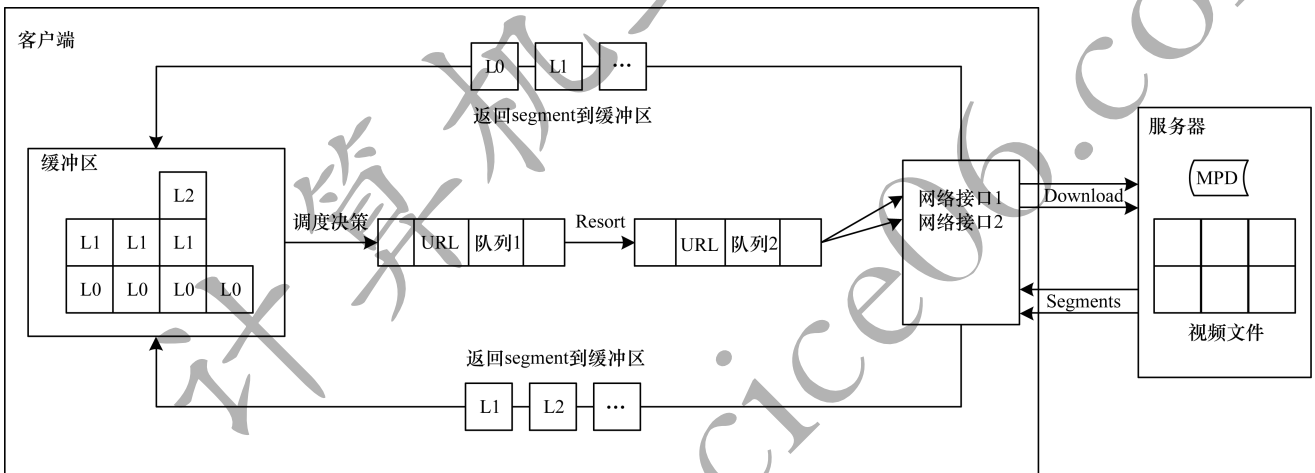


图 1 基于视频块优先级的联合调度算法

整个算法在逻辑上分为两步：

- 1) 确定下一次待下载的视频块(即确定视频块在二维缓冲区的位置,如第 5 块 L1 层),将其放入待下载队列,并根据优先级对队列中的视频块进行重新排序。
- 2) 根据每个网络接口的带宽,将队列中的视频块分配给不同的接口进行下载操作。

为了保证播放质量,一方面应尽可能地下下载增强层的编码块,另一方面还应降低播放码率的切换次数。为此,本文采用基于缓冲区的算法,并辅以平滑操作^[19]来实现上述目标。

2.3 算法描述

本文设计一种基于视频块优先级的联合调度算法。

算法 1 基于视频块优先级的联合调度算法

1. 输入 $bufferlength, bw_{eth}, bw_{wifi}$
2. $x \leftarrow 0$ // x 表示时间维度
3. WHILE $x \leq N$ DO
4. IF $bufferlength < a$ THEN
5. $s \leftarrow L0$ //如果缓冲区长度很短,则仅下载 L0 层
6. ELSEIF $bufferlength < c$ THEN

目前,一般的终端设备都具有多个网络接口,例如笔记本电脑可以同时接入有线网络和 WIFI,用户的手机一般都有 4G 和 WIFI 2 种网络接入方式。在国内,4G 网络的下载和上传速率分别可以达到 100 Mb/s 和 20 Mb/s^[18]。在现有的设置下,用户设备对链路的使用在同一时刻只能是二选一的方式。当用户下载数据时,如果接入 WIFI 就会优先使用 WIFI,但在此期间,如果 WIFI 链路中断,那么只能提示下载失败,而不会切换到 4G 网络继续下载。同时,对于已经存在的多条链路,并没有并行使用以提高用户体验。

另一方面,在视频质量越来越高以及 VR 视频传输的情况下,流媒体传输对带宽的要求也逐渐增加。随着资费策略的改变,设备并行利用多个网络接口的带宽是提高用户体验的有效途径之一。

2.2 算法框架

SVC 编码将整个缓冲区组织成一个包含时间维度和编码层次维度的二维空间。基于 SVC 和多网络接口的调度算法框架如图 1 所示。

7. $s \leftarrow L0, L1$ // 如果缓冲区比较安全,则下载 L0、//L1 层
8. ELSE
9. $s \leftarrow L0, L1, L2$ // 如果缓冲区有足够的缓冲块,则下载 //L0、L1、L2 层
10. IF $bufferlength > b$ THEN
11. $smooth()$ // 如果缓冲区的块大于某个阈值,则可 //以选择一个平滑块下载
12. $resort(s)$ // 对队列中的待下载视频块的 URL 进行 //重新排序
13. $download(s, bw_{eth}, bw_{wifi})$ // 下载队列中的视频块
14. $x \leftarrow x + 1$ // 推进到下一个时间块
15. END WHILE

在算法 1 中, $bufferlength$ 为缓冲区长度, bw_{eth} 为以太网链路的加权带宽, bw_{wifi} 为 WIFI 链路的加权带宽, N 为整个视频的切片数量, s 为待下载的切片的 URL 队列。

每个接口的加权带宽计算公式如下：

$$bw = \alpha \times bw + (1 - \alpha) \times bw_{down}$$

其中, bw 为计算出的加权带宽, bw_{down} 为实际下载带宽, α 为带宽权重因子。

缓冲区的阈值设置为 $[0, a, b, c, max]$ 。其中, a, b, c 3 个参数代表缓冲区的不同等级划分, 用来控制第一步下载层数, 其计算公式为:

$$a = \left(\frac{R_0}{R_2} \right) Th_{safe}$$

$$b = \left(\frac{R_1}{R_2} \right) Th_{safe}$$

$$c = \left(\frac{R_2}{R_2} \right) Th_{safe} = Th_{safe}$$

$$Th_{safe} = \beta max$$

其中, β 是一个缓冲区安全因子, R_0, R_1, R_2 表示 MPD 文件中对应的 L0、L1、L2 的带宽要求。根据不同的 MPD 文件, 可以产生不同的缓冲区阈值。

每当待下载队列为空时, 算法 1 首先依据缓冲区的状态确定待下载的视频块。其中, *smooth* 操作是对当前的缓冲区进行一次质量提升。对当前缓冲区中所有质量未达到最高层次的块, 选取一个提升质量之后与左右播放质量相差最小的块作为最后 *smooth* 操作确定的下载块。如图 2(a) 所示的缓冲区, 选择待提高的块为图 2(b) 中的 a、b、c、d 4 个块, 最终计算得出的 *smooth* 块为 c。

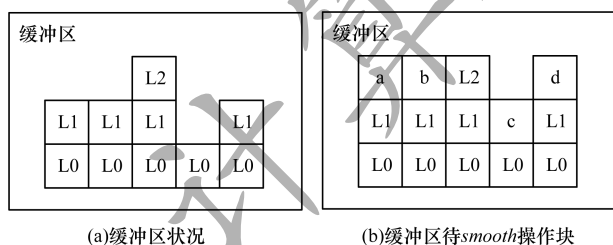


图 2 *smooth* 操作示意图

Resort 操作是对一次调度算法确定插入待下载队列中的视频块按优先级进行排序, 优先级最高的为基础层, 用以保证视频播放的连续性, 其次为 *smooth* 操作确定的块, 用来有效地提高视频的播放质量, 然后依次是增强层 1 和增强层 2。

在 *download* 的过程中, 输入参数为一个待下载的视频块 URL 队列 s 、以太网链路的加权带宽 bw_{eth} 以及 WIFI 链路的加权带宽 bw_{wifi} 。加权带宽同时考虑了对当前链路状况的预测和之前链路的稳定性。对客户端的 2 个接口的加权带宽进行比较, 选择其中加权带宽较大的网络端口拉取待下载队列 s 中的第 1 个 URL 并进行下载, 另一个网络端口下载第 2 个 URL。当每个接口完成自身正在进行下载任务后, 继续从队列 s 中拉取 URL 进行下载, 直至其为空, 并将下载好的视频块输入到视频缓冲区中。*download* 过程的伪代码如算法 2 所示。

算法 2 *download*

```
1. 输入  $s, bw_{eth}, bw_{wifi}$ 
2. IF  $bw_{eth} < bw_{wifi}$  THEN // 选择接口加权带宽大的下载
   // 队列中优先级高的视频块, 另一个端口下载次优的视频块
```

```
3. wifidownload ←  $s.pop()$ 
4. ethdownload ←  $s.pop()$ 
5. ELSE
6. ethdownload ←  $s.pop()$ 
7. wifidownload ←  $s.pop()$ 
8. WHILE  $s.size() > 0$  DO // 当队列未空时, 按接口下
   // 载任务的完成情况, 依次从队列中拉取 URL 进行下载
9. IF ethdownload.finish() THEN ethdownload ←  $s.pop()$ 
10. ELSEIF wifidownload.finish() THEN wifidownload ←  $s.pop()$ 
11. END WHILE
```

3 实验与性能评价

3.1 实验环境

本文实验的环境如图 3 所示, 主要分为 3 个部分: 服务器, 网络带宽流量控制器和客户端。在 Linux 系统上使用 Apache2.2.4 搭建服务器, 然后将编码好的 SVC 视频切片和相应的 MPD 文件部署在服务器上。网络带宽流量控制器主要使用 Linux 下的 Traffic Control (TC) 和 Netem 工具来实现, 负责对实验中的网络链路带宽进行控制, 模拟网络中的网络波动、丢包等实际情况, 使得实验环境贴合实际的应用场景。客户端是一个使用 C++ 编写的播放程序, 主要分为下载模块、MPD 解析模块、码率自适应调度模块和播放模块。客户端通过 2 块网卡分别连接以太网和无线网。

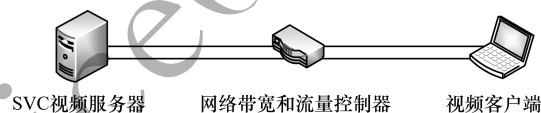


图 3 实验环境

实验视频总长度为 200 s, 视频块的长度为 2 s, 共分为 3 个质量层次。视频播放至每个层次的带宽总要求分别为 800 Kb/s、1 265 Kb/s、1 800 Kb/s。以太网链路和 WIFI 链路的带宽分别为 $(1\ 100 \pm 150)$ Kb/s 和 (800 ± 100) Kb/s。实验选取基于带宽的调度算法、基于带宽和缓冲的调度算法和本文提出的基于视频块优先级的联合调度算法进行对比。

3.2 实验评价指标

本文实验使用的性能评价指标如下:

- 1) 平均的视频播放质量 (Average Play Quality, APQ): 整个播放期间, 每个块的视频质量 (平均比特率) 的总和除以块的数目。
- 2) 视频切换次数: 整个视频播放期间, 视频质量的切换次数, 包括向上切换次数和向下切换次数。
- 3) 平均缓冲水平: 整个视频播放期间, 播放器缓冲区的平均缓冲的视频长度。

3.3 实验结果

3.3.1 基于带宽的调度算法

客户端使用基于带宽的调度算法 RB^[10] 进行实验, 结果如图 4 所示。

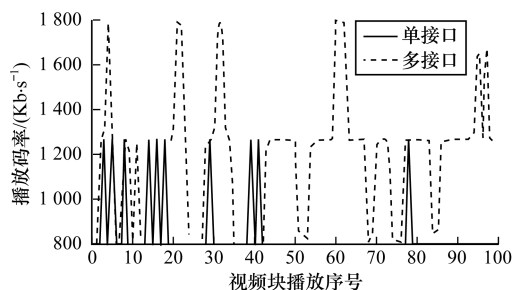


图4 基于带宽的调度算法播放质量

从图4可以看出:

1) 在单接口的情况下, 大部分播放质量为 L0 层, 当链路带宽波动至 1 200 Kb/s 以上时, 播放质量为 L1 层。整个播放期间的平均播放码率为 846.03 Kb/s, 切换次数为 20 次, 平均缓冲水平为 7 s。

2) 在使用了多接口之后, 整个视频播放期间的平均码率为 1 198.26 Kb/s, 切换次数为 33 次, 平均缓冲水平为 6.8 s。

3.3.2 基于带宽和缓冲的调度算法

客户端使用基于带宽和缓冲的调度算法 Combination^[20] 进行实验, 结果如图5所示。

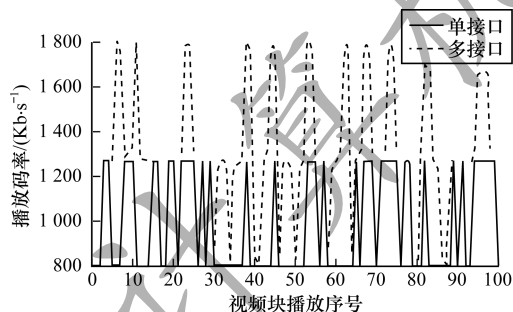


图5 基于带宽和缓冲的调度算法播放质量

从图5可以看出, 当缓冲水平较高时, 可以适当牺牲缓冲长度来提高视频播放的质量。

1) 在单接口的情况下, 整个视频播放期间的平均码率为 988.76 Kb/s, 切换次数为 38 次, 平均缓冲水平为 5 s。

2) 在使用了多接口之后, 整个视频播放期间的平均码率为 1 271.04 Kb/s, 切换次数为 47 次, 平均缓冲水平为 5.2 s。

3.3.3 基于视频块优先级的联合调度算法

客户端同时使用以太网卡和无线网卡进行实验, 结果如图6所示。

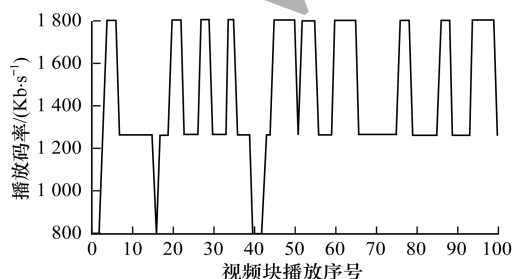


图6 基于视频块优先级的调度算法播放质量

从图6可以看出, 整个播放期间的平均播放码率为 1 443.96 Kb/s, 切换次数为 24 次, 平均缓冲水平为 3.8 s。

3.4 结果分析

上述3个实验的平均播放码率、切换次数以及缓冲水平的对比如图7~图9所示。

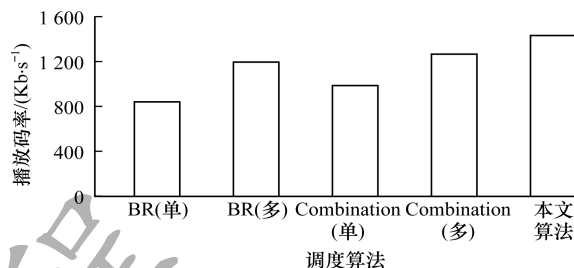


图7 平均播放码率对比

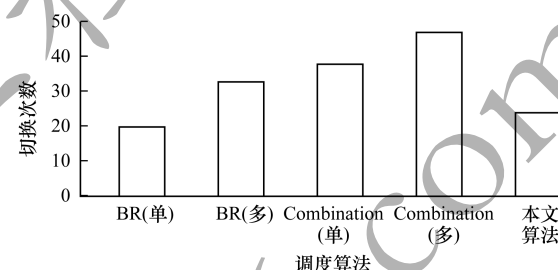


图8 切换次数对比

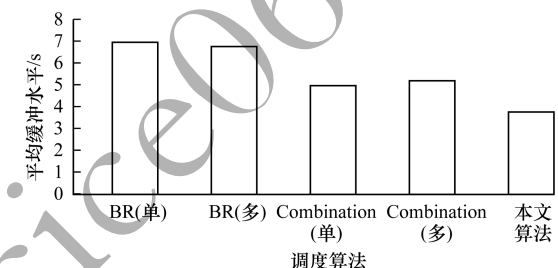


图9 平均缓冲水平对比

从图7~图9可以得出以下结论:

1) 多接口 RB 算法较单接口 RB 算法, 其缓冲水平相差不大, 平均播放码率提高了 41.63%, 但切换次数增加了 65%。

2) 多接口 Combination 算法较单接口 Combination 算法, 其缓冲水平相差不大, 平均播放码率提高了 28.54%, 但切换次数增加了 23.68%。

3) 本文算法较单接口 RB 算法, 其平均视频质量提高了 70.67%, 缓冲水平下降了 45.71%, 但切换次数增加了 20%。

4) 本文算法较多接口 RB 算法, 其平均视频质量提高 20.50% 的同时, 切换次数降低了 27.27%, 并且缓冲水平下降了 24%。

5) 本文算法较单接口 Combination 算法, 其平均视频质量提高 46.04% 的同时, 切换次数降低了 36.84%, 并且缓冲水平下降了 44.11%。

6) 本文算法较多接口 Combination 算法, 其平均视频质量提高 13.60% 的同时, 切换次数降低了 48.93%, 并且缓冲水平下降了 26.92%。

4 结束语

本文提出一种基于视频块优先级的联合调度算法,采用 SVC 编码将视频码流分为基础层和增强层,根据缓冲队列中视频块时间和编码层次的优先级以及各个网络接口的带宽,利用多接口同时执行下载任务。实验结果表明,相较于基于带宽的调度算法和基于带宽和缓冲的调度算法,该算法在有效提高平均视频播放码率的同时,能够减少切换次数和服务端存储开销,同时降低缓冲水平。下一步将根据接口带宽、待下载的视频块大小和下载总时间对接口进行协同调度,优化调度算法。此外,本文算法以 PC 作为设备终端,而多网络接口更适用于移动设备,因此,后续将考虑在安卓和 IOS 平台下,针对 WIFI 和蜂窝网络进行调度算法研究。

参考文献

- [1] China Internet Network Information Center. The 41st statistical report on the development of China's Internet [EB/OL]. [2018-10-08]. http://www.cac.gov.cn/2018-01/31/c_1122347026.htm. (in Chinese)
中国互联网信息中心. 第 41 次中国互联网发展统计报告[EB/OL]. [2018-10-08]. http://www.cac.gov.cn/2018-01/31/c_1122347026.htm.
- [2] Cisco visual networking index 2015-2020 [EB/OL]. [2018-10-08]. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>.
- [3] CHEN Jian, HUANG Fenghui, ZHOU Wenan, et al. Overview of QoE Management Based on HTTP Mobile Streaming[J]. Computer Systems & Applications, 2014, 23(7): 1-6. (in Chinese)
陈坚, 黄凤辉, 周文安, 等. HTTP 移动流媒体 QoE 管理综述[J]. 计算机系统应用, 2014, 23(7): 1-6.
- [4] ZHU Xiaochen, SHEN Subin. Survey of dynamic adaptive streaming over HTTP (DASH)[J]. Journal of System Simulation, 2013, 25(11): 2522-2527, 2533. (in Chinese)
朱晓晨, 沈苏彬. 基于 HTTP 的动态自适应流技术综述[J]. 系统仿真学报, 2013, 25(11): 2522-2527, 2533.
- [5] SODAGAR I. The MPEG-DASH standard for multimedia streaming over the Internet[J]. IEEE Multimedia, 2011, 18(4): 62-67.
- [6] DE CICCIO L, MASCOLO S. An adaptive video streaming control system: modeling, validation, and performance evaluation[J]. IEEE/ACM Transactions on Networking, 2014, 22(2): 526-539.
- [7] MÜLLER C, LEDERER S, TIMMERER C. An evaluation of dynamic adaptive streaming over HTTP in vehicular environments [C]//Proceedings of the 4th Workshop on Mobile Video. New York, USA: ACM Press, 2012: 37-42.
- [8] JIANG Junchen, SEKAR V, ZHANG Hui. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive [J]. IEEE/ACM Transactions on Networking, 2014, 22(1): 326-340.
- [9] SUN Yi, YIN Xiaoqi, JIANG Junchen, et al. CS2P: improving video bitrate selection and adaptation with data-driven throughput prediction [C]//Proceedings of 2016 ACM SIGCOMM Conference. New York, USA: ACM Press, 2016: 272-285.
- [10] LI Zhi, ZHU Xiaoqing, GAHM J, et al. Probe and adapt: rate adaptation for HTTP video streaming at scale[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(4): 719-733.
- [11] ZOU Xuan, ERMAN J, GOPALAKRISHNAN V, et al. Can accurate predictions improve video streaming in cellular networks? [C]//Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. New York, USA: ACM Press, 2015: 57-62.
- [12] JEONG J M, KIM J D. Effective bandwidth measurement for dynamic adaptive streaming over HTTP [C]//Proceedings of 2015 International Conference on Information Networking. Washington D. C., USA: IEEE Press, 2015: 375-378.
- [13] HUANG Teyuan, JOHARI R, MCKEOWN N, et al. A buffer-based approach to rate adaptation: evidence from a large video streaming service [C]//Proceedings of ACM SIGCOMM Computer Communication Review. New York, USA: ACM Press, 2014: 187-198.
- [14] SPITERI K, URGAKONKAR R, SITARAMAN R K. BOLA: near-optimal bitrate adaptation for online videos [C]//Proceedings of the 35th Annual IEEE International Conference on Computer Communications. Washington D. C., USA: IEEE Press, 2016: 1-9.
- [15] YIN Xiaoqi, JINDAL A, SEKAR V, et al. A control-theoretic approach for dynamic adaptive video streaming over HTTP [C]//Proceedings of ACM SIGCOMM Computer Communication Review. New York, USA: ACM Press, 2015: 325-338.
- [16] SIEBER C, HÖBFFELD T, ZINNER T, et al. Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC [C]//Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management. Washington D. C., USA: IEEE Press, 2013: 1318-1323.
- [17] KREUZBERGER C, POSCH D, HELLWAGNER H. A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP [C]//Proceedings of the 6th ACM Multimedia Systems Conference. New York, USA: ACM Press, 2015: 213-218.
- [18] DENG Yonghong. 4G communication technology overview[J]. Digital Space, 2005(2): 58-63. (in Chinese)
邓永红. 4G 通信技术综述[J]. 数字通信世界, 2005(2): 58-63.
- [19] LEKHARU A, KUMAR S, SUR A, et al. A QoE aware SVC based client-side video adaptation algorithm for cellular networks [C]//Proceedings of the 19th International Conference on Distributed Computing and Networking. New York, USA: ACM Press, 2018: 1-27.
- [20] OZCAN S G, KIVILCIM T, CETINKAYA C, et al. Rate adaptation algorithm with backward quality increasing property for SVC-DASH [C]//Proceedings of the 7th International Conference on Consumer Electronics-Berlin. Washington D. C., USA: IEEE Press, 2017: 24-28.