

一种基于 HTTP 自适应流的混合码率自适应算法

熊丽荣 雷静之 金鑫

(浙江工业大学计算机科学与技术学院 杭州 310023)

摘要 码率自适应算法是 HTTP 自适应流技术的热点和难点。提出一种综合网络带宽和缓存两个因素的终端码率自适应算法(Combined with Bandwidth and Buffer, CBB)。该算法采用“探测”的机制在应用层上估算网络实时带宽,避免视频码率频繁切换;然后构建随缓存状态动态变化的平滑因子模型,并基于指数加权移动平均(EWMA)实现带宽的平滑处理;利用推动缓存趋近均衡级别变化的调度策略,尽可能使缓存区的数据量处于均衡的范围。整个算法经带宽估算、平滑处理、量化及确定调度时间构成一个循环作用的闭环。在使用 MPEG-DASH 标准的参考平台 libdash 上验证该算法的性能,结果表明,在变化的网络状况中所提算法表现良好。

关键词 HTTP 自适应流,码率控制,网络带宽,缓存状态,MPEG-DASH

中图分类号 TN919.85 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.02.019

Hybrid Rate Adaptation Algorithm for Adaptive HTTP Streaming

XIONG Li-rong LEI Jing-zhi JIN Xin

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract The rate-adaptation algorithm is the hotspot and difficulty in adaptive HTTP streaming(AHS). In this paper, we proposed a hybrid rate-adaptation algorithm (Combined with Bandwidth and Buffer, CBB) for adaptive HTTP streaming. It is designed at the application layer using a “probe” principle to estimate the real-time bandwidth and avoid frequent switching of video rate. Then an exponential weighted moving average smoother (EWMA) is applied in bandwidth smoothing procedure. The smoothing factor can vary with the client buffer status to reduce buffer overflows. The schedule strategy is designed to keep buffer level in a balanced range as far as possible. The whole algorithm includes estimating, smoothing, quantizing and scheduling. These four steps proceed to form a circular closed-loop. The algorithm has been verified on the MPEG-DASH standard reference platform-libdash, and the experiments show that our algorithm performs well under varying network conditions.

Keywords AHS, Rate control, Network bandwidth, Buffer state, MPEG-DASH

1 引言

随着网络传输技术的飞速发展,视频业务在整个因特网的数据流量中正占据越来越大的比重。传统的基于 RTP/RTSP 的流媒体传输技术已经不能满足用户日益增长的高质量流媒体需求^[1]。HTTP 自适应流技术已成为视频业务技术的发展趋势^[2-3]。

HTTP 自适应流技术通过感知当前主机的网络、内存、CPU 等状况,自动选取最合适码率的视频资源进行下载^[4]。由于 TCP 数据传输会出现短期波动和长期波动,依据网络吞吐量决定视频片段的码率自适应算法^[5-6]可能会导致缓存区溢出和视频码率切换频繁等问题^[4]。缓存区的状态受到吞吐量的影响,但是基于缓存区状态的码率自适应算法^[7-8]仍不能根据带宽的实时变化切换视频码率,存在切换滞后的问题^[9]。混合控制的自适应算法^[4,10-12]综合考虑多个参数(如网络带

宽、缓存等),以准确反映当前网络状况和客户端情况。文献[4,10,12]构建了简单的客户端缓存模型。Hesse S 等^[11]构造随缓存状态动态变化的带宽平滑因子,能够更好地利用缓存状态和网络带宽之间的关系进行码率选择。但是其将 TCP 下载吞吐量作为估算的可用带宽,不能准确反映出实时的网络带宽。

在动态网络的环境下,TCP 拥塞控制和码率自适应机制的双重作用使得流媒体系统变得很难预测^[1]。设计高性能的自适应算法已成为 HTTP 自适应流技术的热点和难点。

本文提出的混合控制码率自适应算法综合考虑了网络带宽和客户端缓存状态。为了降低视频码率切换的不稳定性,直接在应用层用“探测”的机制估算网络实时带宽。带宽平滑处理时,将缓存状态作为构建平滑因子的关键参数,由网络带宽和缓存状态共同影响带宽平滑,以减少缓存区溢出。设计调度时间策略,推动缓存趋近均衡级别变化。实验结果表明,

收稿日期:2016-01-04 返修日期:2016-04-26 本文受浙江省重大科技专项重大工业项目(2012C11026-2)资助。

熊丽荣(1973—),女,硕士,副教授,CCF 会员,主要研究方向为服务计算和计算机网络,E-mail:lilybear@zjut.edu.cn;雷静之(1992—),女,硕士生,主要研究方向为服务计算、计算机网络,E-mail:leijingzhi_hi@163.com;金鑫(1991—),男,硕士生,主要研究方向为服务计算。

所提算法能够尽可能使缓存区的数据量处于较均衡的范围,在充分利用带宽的前提下保证较佳的视频播放效果,从而保持视频码率切换的稳定性。

本文第2节阐述 HTTP 自适应流技术及码率自适应算法的相关研究工作;第3节详细阐述提出的混合控制码率自适应算法 CBB;第4节模拟实验,并通过结果分析算法性能;最后总结全文并指出下一步的工作。

2 相关工作

随着全球互联网视频、移动流媒体业务的蓬勃发展,HTTP 自适应流技术被主流的媒体内容提供商用于实现高质量的视频传输^[13-15]。MPEG-DASH^[16]标准综合了主流流媒体协议的基本架构,具有良好的平台兼容性。但 DASH 标准仅定义了信息格式和交互方式,没有规定影响流媒体系统性能的关键——终端自适应算法,这给设计者们留下了极大的竞争空间^[17]。

基于吞吐量的自适应算法依靠估算的吞吐量调整视频码率,分为即时吞吐量和平滑吞吐量两类^[18]。Liu C H 等^[5]采用片段的传输时间计算 HTTP 吞吐量,利用平滑后的 HTTP 吞吐量来判断当前的网络状况,采用 Step-Wise/Aggressive 的码率切换机制。Thang T C 等^[6]设计平滑吞吐量的算法选取历史片段相应的吞吐量的特征,利用 logistic 函数模型对吞吐量偏差和平滑因子之间的关系进行建模。

基于缓存的自适应算法^[7-9]将客户端缓存分成多个阈值区间,依据缓存区的状态选择不同的码率切换策略。文献^[7]基于缓存级别预测网络吞吐量,如果缓存级别大约是最大阈值的 35%~50%,则估算的吞吐量和前一个片段的吞吐量相同;如果缓存级别降低到较低的范围,则估算的吞吐量等于之前的吞吐量乘以一个向下的缩放因子。该算法比即时吞吐量的算法更加积极。Le H T 等^[8]提出了基于未来缓存的算法,即由客户端预测缓存区将来的状态,在视频高质量需求和缓存区稳定性之间寻求平衡。

混合控制的自适应算法综合考虑网络带宽、缓存等因素,以准确反映当前网络的状况和客户端情况。陈立伟等^[4]在视频资源缓存控制机制中,预先设定阈值,调节视频码率并调整单个视频片段进入缓存区所消耗的时间。在带宽控制机制中,根据当前的网络状况,设定不同的码率调整策略:利用保守方法单级别上调码率;利用 logistic 方程决定码率下调幅度。朱晓晨等^[10]综合运用了时间参数、带宽参数、缓存参数和开销参数,以段获取时间^[19]计算吞吐量作为自适应判断的主要依据。在带宽良好、最高级别的码率得到连续采用的情况下考虑缓存对数据传输的影响。Pozueco L 等^[12]对前 5 个片段的 TCP 吞吐量求均值,控制估算的网络带宽比视频码率高出 20%。在缓存的占用百分比达到 35%之前,客户端一直保持下载最低码率的片段。但是这种方法并不能准确地使可选视频码率匹配可用带宽,而且依赖单个阈值的缓存控制机制不能较好地发挥缓存参数的调整功能。Hesse S 等^[11]推荐了一种流会话期间媒体数据选择和恢复机制,设计了一个缓存监控单元,规定缓存上、下限,将缓存状态作为平滑函数中的输入参数,能更好地利用缓存状态和网络带宽之间的关系

进行码率选择。

动态网络环境下, TCP 的瞬时传输速率是时刻变化的。如果上述基于自适应流算法将 TCP 下载吞吐量或者平滑后的 TCP 下载吞吐量作为可用带宽,那么就会出现诸多问题。即时吞吐量的算法遇到 TCP 短期波动会导致频繁的视频质量抖动。平滑吞吐量的算法能较好地应对短期波动,但需要较大的初始缓存空间,且对大型的吞吐量波动反应迟缓。例如当实际吞吐量突然降低时,采用平滑的算法估算得到的吞吐量可能仍然很高,这时请求下载的片段码率比实际的吞吐量大,会造成缓存区下溢^[20]。

用测量到的 TCP 下载吞吐量来衡量网络的传输能力是不合理的^[21]。Li Z 等^[22]基于“探测和自适应”原理去探测实时带宽,从而决定了选择的视频码率并划分出了较细的请求时间间隔。该算法在多个客户端竞争网络带宽时,可显著降低视频码率选择的不稳定性。但带宽良好时,客户端缓存区在视频播放后的短时间内会出现上溢;在带宽骤降时,缓存区容易下溢,造成视频播放停滞。

本文提出的 CBB 算法将网络带宽和客户端缓存作为混合控制的参数,利用缓存状态和网络带宽之间的关系进行码率选择。在 BS 算法^[11]的基础上,改进网络带宽预测方法,采用“探测”机制,结合历史视频片段计算平均 TCP 下载吞吐量,通过试验性地增加数据速率的方式探测出网络实际传输能力,提高网络带宽利用率。参考 PANDA 算法^[22]的量化及调度方法,利用缓存状态和网络带宽之间的关系确定调度时间,保持缓存区数据量处于较均衡的区间。相比于 BS 算法和 PANDA 算法,CBB 算法能够尽量避免缓存发生溢出(上溢或下溢),能够更加充分地利用网络带宽,提高了平均视频质量。

3 基于带宽和缓存的混合控制算法

3.1 基于 AHS 的视频流传输系统

在基于 HTTP 自适应流的视频流传输过程中, HTTP 服务器存储媒体描述文件(Manifest)及不同码率的媒体片段。客户端负责感知网络状态的变化,选择合适码率的片段,支持媒体的持续播放。CBB 算法适用的客户端系统模块及数据传输过程如图 1 所示。

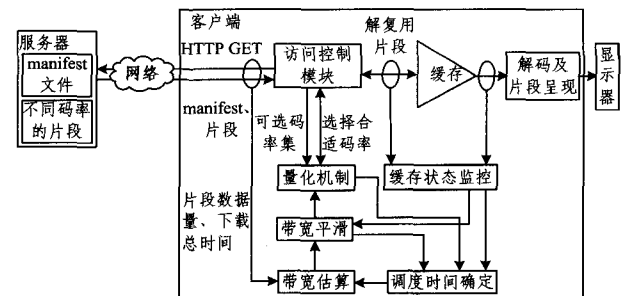


图 1 基于 AHS 的视频流传输系统

客户端首先下载服务器上的媒体描述文件,解析后得到媒体内容的时间长度、分辨率以及 URL 等信息。基于这些信息,客户端向服务器请求合适的多媒体片段。客户端在播放当前片段的同时继续请求后面的片段,并监控网络带宽的

变动,必要时调整片段的码率。调整的机制由自适应算法决定,以保证视频的流畅播放。客户端的访问控制模块负责接收并解析媒体描述文件,接收并解析复用媒体片段,处理一切和 Web 服务器通信的事务。缓存区存储解复用后的片段提供给解码及呈现模块,由显示器播放。

3.2 混合控制算法 CBB 的思路及关键步骤

自适应算法 CBB 基于“探测”机制进行网络实时带宽估计,将缓存状态作为构建平滑因子的关键参数,利用缓存状态和网络带宽之间的关系进行码率选择,确定调度请求时间。

本算法的参数设置和符号定义如表 1 所列。

表 1 参数和符号

参数和符号	含义
k	探测的收敛速率
β	客户端缓存的收敛速率
\tilde{x}	平均 TCP 下载吞吐量
\hat{x}	目标带宽(探测出的带宽)
\hat{y}	平滑后的带宽
$E(\cdot)$	估算带宽
$S(\cdot)$	平滑函数
α	平滑因子
$s(\cdot)$	平滑因子模型
$Q(\cdot)$	视频码率量化函数
$M(\cdot)$	确定下个片段的下载请求时间
$r[n]$	片段 n 的码率
$B[n]$	片段 n 下载结束后的缓存数据量
B_{max}	缓存最大容量
B_r	缓存均衡级别
$T[n]$	片段 n 的请求下载时间
R	视频码率集 $\{r_1, \dots, r_l\}$
ϵ	量化边距
n	当前下载的片段 ID
$B_{filling}$	缓存数据量百分比
B_{rate}	缓存深度变化率
B_t	时刻 t 时缓存数据量
r_t	时刻 t 时下载的码率
B_{up}	缓存上溢阈值
B_{down}	缓存下溢阈值

混合码率自适应算法 CBB 包括带宽估算、平滑、量化及调度 4 个步骤。

(1)带宽估算:直接在应用层用“探测”的机制小幅度地增加数据速率以监测网络带宽的变动,探测合理的网络传输能力,更有效地利用网络带宽。

(2)平滑:在平滑因子的选择上,参考文献[11]设计平滑函数的思想,将缓存数据量百分比和缓存深度变化率作为构建平滑因子的关键参数。基于指数加权移动平均法 EWMA 对上一片段估算出的可用带宽进行平滑处理。

(3)量化:量化处理是将平滑后的可用带宽映射到离散的视频码率上。本文基于 dead-zone 量化器 $r[n] = Q(\hat{y}[n], r[n-1])$, $r[n] \in R^{[22]}$, 以平滑后的带宽和上一片段的码率作为参数选择视频码率。通过设定 $\Delta_{up}, \Delta_{down}$ 的具体值创建出“dead-zone” $[r_{up}, r_{down}]$, 减少在两个相邻级别间的频繁切换。

(4)调度:为了使得缓存区的数据量处于较均衡的范围,尽量避免发生溢出,在调度策略中还考虑了缓存状态。为了推动缓存状态 $B[n]$ 朝均衡级别 B_r 变化,文献[22]最终设计确定调度时间的公式如下:

$$T[n] = \frac{r[n] \cdot \tau}{\hat{y}[n]} + \beta \cdot \tau \cdot (B[n] - B_r) \quad (1)$$

其中, $\beta > 0$ 控制着收敛的速率。

CBB 算法的重点放在自适应算法中两个极其重要的步骤上:带宽估算和平滑。

3.2.1 基于探测的应用层带宽估算方法

传统的基于传输层的带宽探测方法需要 RTT 和丢包率等传输层信息。本文的 CBB 算法引入探测思想在应用层上估算网络带宽,通过不断增加微小的发送速率来探测网络带宽并进行微调,更有效地利用网络带宽。相比于基于即时 TCP 下载吞吐量的算法,考虑历史片段的数据计算出平均 TCP 下载吞吐量的算法能够更好地应对短期的带宽波动,避免频繁的码率切换。

CBB 算法先收集当前需要下载的片段之前的 m 个片段的数据量大小 $sumByte$ 及下载总时间 $duration$, 并将两者相除求得平均 TCP 下载吞吐量 \tilde{x} , 公式如下:

$$\tilde{x} = \frac{sumByte}{duration} \quad (2)$$

若是处于下载初期,还未下载到第 $m+1$ 个片段,则取前面所有片段的数据来计算。客户端通过以每单位时间增加 k 倍的增量速率确定目标带宽 \hat{x} 。估算目标带宽 $\hat{x}[n]$ 的公式如下:

$$\hat{x}[n] = \hat{x}[n-1] + k \cdot (\tilde{x}[n-1] - \hat{x}[n-1]) \cdot T[n-1] \quad (3)$$

根据式(2)对 TCP 下载吞吐量 \tilde{x} 保持监测,式(3)将 \tilde{x} 与目标带宽 \hat{x} 进行比较。如果 $\tilde{x}[n-1] > \hat{x}[n-1]$, 则说明上一片段的目标带宽小于上一片段测量的平均 TCP 下载吞吐量,此时需要增加目标带宽使其向平均 TCP 下载吞吐量靠近。如果 $\tilde{x}[n-1] < \hat{x}[n-1]$, 平均 TCP 下载吞吐量不能满足目标带宽,这表明链路上发生了拥塞,这时的增量为负值。

3.2.2 构建与缓存状态相关的平滑函数

CBB 算法的平滑处理过程是基于指数加权移动平均法(EWMA)的。平滑后的可用带宽 $\hat{y}[n]$ 是当前片段的目标带宽和上一片段的可用带宽的折衷,计算公式如下所示:

$$\hat{y}[n] = (1 - \alpha) \cdot \hat{x}[n] + \alpha \cdot \hat{y}[n-1] \quad (4)$$

其中, α 是平滑因子, $0 \leq \alpha \leq 1$ 。

在平滑因子的选择上,参考文献[11]设计了平滑函数的思想,将缓存状态作为构建平滑因子的关键参数,从而影响带宽平滑过程。这样,视频码率的选择不仅与实时的网络带宽有关,还会受到客户端缓存状态的影响。

本算法根据缓存区数据量百分比 $B_{filling}$ ($0 \leq B_{filling} \leq 100\%$) 和缓存深度的变化率 B_{rate} ($0 \leq B_{rate} \leq 1$), 动态调整平滑因子 α 的大小,改变可用带宽 $\hat{y}[n]$ 的倾向性。

$$B_{filling} = \frac{B[n-1]}{B_{max}} \quad (5)$$

$$B_{rate} = \frac{|B[n-1] - B[n-2]|}{B[n-1]} \quad (6)$$

最终平滑因子模型的构建如下所示:

$$\alpha(n) = B_{filling} \cdot (1 - B_{rate}) \quad (7)$$

在视频播放的起始阶段,缓存区还未下载视频片段,缓存区内没有数据量,那么 $\alpha=0$, $\hat{y}[n]$ 全部依赖于当前片段的目标带宽 $\hat{x}[n]$ 。当缓存区内片段数目 $B[n]$ 较少即 $B_{filling}$ 较小时,若缓存深度的变化率 B_{rate} 较大, α 就会变得更加小, $\hat{y}[n]$ 比较依赖于当前片段的目标带宽 $\hat{x}[n]$, 此时自适应调整迅速。随着缓存区内片段数目 $B[n]$ 增加, $B_{filling}$ 的值增加, α 变大, $\hat{y}[n]$ 渐渐依赖于上一片段的可用带宽 $\hat{y}[n-1]$ 。当缓存状态良好时,如缓存内已有片段数目在均衡级别 B_r 左右,由于 $B_{filling}$ 较大,此时 α 相对来说较大。但是若此时链路实时带宽发生较大的波动,无论是骤降还是突升, B_{rate} 都会变大,这会限制 $B_{filling}$ 对 α 的影响,调整可用带宽 $\hat{y}[n]$ 更多依赖于目标带宽 $\hat{x}[n]$, 迅速估算出波动后的实时带宽。

3.3 基于带宽和缓存的混合控制算法 CBB

CBB 算法的整体描述如下。

算法 1 CBB 算法

输入: 当前请求下载片段数目 n , 历史片段数目 m , 探测的收敛系数 k , 缓存的收敛系数 β , 量化安全边际 ϵ , 缓存容量 B_{max} , 片段 $n-1$ 下载结束后的缓存数据量 $B[n-1]$ 和片段 $n-2$ 下载结束后的 $B[n-2]$, 视频码率集 R , 缓存均衡级别 B_r

输出: 当前片段码率 $r[n]$, 当前片段请求时间 $T[n]$

```

1. While( $n \neq$  the last segment) {
2. if  $n=1$ 
3. 请求下载表示级别最低的段;
4. else if  $1 < n \leq m$ 
5. 获取第  $n-1, n-2, \dots, 1$  个片段的数据量 sumByte 及下载总时间 duration;
6. else
7. 获取第  $n-1, \dots, n-m$  个片段的数据量 sumByte 及下载总时间 duration;
8. end if
9. set  $\hat{x}[n-1] = \frac{\text{sumByte}}{\text{duration}}$ ;
10. set  $\hat{x}[n] = E(\hat{x}[n-1], \tilde{x}[n-1], T[n-1])$ ;
11. set  $B_{filling} = \frac{B[n-1]}{B_{max}}$ ;
12. set  $B_{rate} = \frac{|B[n-1] - B[n-2]|}{B[n-1]}$ ;
13. set 平滑因子  $\alpha = s(B_{filling}, B_{rate})$ ;
14. set 平滑后的带宽  $\hat{y}[n] = S(\hat{x}[n], \hat{y}[n-1], \alpha)$ ;
15. set  $\Delta_{up} = \epsilon \cdot \hat{y}[n], \Delta_{down} = 0$ ;
16. 遍历  $R$ , 选出  $r1 = \{r | r \leq \hat{y}[n] - \Delta_{up} \wedge r \in R\}$ ;
17.  $r_{up} = \max_{r1 \in R}$ ;
18. 遍历  $R$ , 选  $r2 = \{r | r \leq \hat{y}[n] - \Delta_{down} \wedge r \in R\}$ ;
19.  $r_{down} = \max_{r2 \in R}$ ;
20. set  $r[n] = \begin{cases} r_{up}, & r[n-1] < r_{up} \\ r[n-1], & r_{up} \leq r[n-1] \leq r_{down} \\ r_{down}, & \text{otherwise} \end{cases}$ ;
21. 确定调度时间  $T[n] = M(r[n], \hat{y}[n], B[n], \beta, B_r)$ ;
22. }

```

算法分成 4 个部分, 步骤 1—10 估算网络带宽, 步骤 11—14 平滑处理得到可用带宽, 步骤 15—20 决定下载片段的码率, 步骤 21 确定调度时间。这 4 部分构成一个闭环, 最后一部分确定的调度时间被第一部分的带宽估算使用。

4 实验及结果

4.1 评价指标

从客户端的角度来看, 一个自适应算法的设计应该达到 3 个目标: 尽量避免缓存区溢出; 在带宽充足的情况下, 尽量选择高码率的视频片段; 减少视频码率震荡, 使其尽可能平滑播放。评价一个自适应算法的性能涉及到多个评价标准之间的权衡。本文参考文献[22], 定义了如下 4 个评价指标。

指标 1 缓存上溢性(Overflow)

视频播放期间, 当缓存区存储的片段接近缓存上限值后易造成缓存上溢。码率自适应算法应尽可能使缓存区保持在较稳定的区间, 并且对网络带宽的上下波动有较高的敏感度。可以通过测算缓存区已有片段数目的增加幅度来表明一个算法防止缓存区上溢的能力。若缓存区已有片段数目越多, 则缓存区上溢的可能性越大。将 B_{up} 作为缓存上限阈值(缓存超过 B_{up} 有上溢危险), B_t 是时刻 t 播放器缓存区现有的数据量, t 的时间单位是秒。定义客户端处于时刻 t 的缓存上溢指数 $Overflow$ 如下:

$$Overflow = \frac{\max(0, B_t - B_{up})}{B_{up}} \quad (8)$$

指标 2 缓存下溢性(Underflow)

当缓存区存储的片段接近缓存下限值后易造成缓存下溢, 从而造成视频播放卡顿。同缓存上溢性定义一样, 将 B_{down} 作为缓存下限阈值(缓存超过 B_{down} 有下溢危险), 定义客户端处于时刻 t 的缓存下溢指数 $Underflow$ 如下:

$$Underflow = \frac{\max(0, B_{down} - B_t)}{B_{down}} \quad (9)$$

指标 3 低效率性(Inefficiency)

为了获得较高的平均视频质量, 客户端必须尽可能选择高码率片段。如果网络环境中带宽是固定的, 较高的平均质量就等同于较高的网络利用率。设 C 为网络控制器设置的链路带宽, 在文献[23]测量网络利用率的基础上, 定义客户端在时刻 t 的低效率性 $Inefficiency$ 如下:

$$Inefficiency = \frac{\max(0, C - r_t)}{C} \quad (10)$$

指标 4 不稳定性(Instability)

视频流切换会对 QoE 带来影响, 尽量避免在可用视频码率级别中频繁而显著地改变视频码率, 降低切换代价。用不稳定性表示视频播放的平滑程度, 定义客户端处于时刻 t 时的不稳定性 $Instability$ 如下:

$$Instability = \frac{\sum_{d=0}^{t-1} |r_{t-d} - r_{t-d-1}| \cdot w(d)}{\sum_{d=0}^{t-1} r_{t-d} \cdot w(d)} \quad (11)$$

其中 $w(d) = p - d$ 是权重函数。

¹⁾ http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/bunny_2s_480p_only

4.2 实验设置及结果分析

本文提出的混合控制码率算法(CBB)以 MPEG-DASH 标准的参考平台 libdash^[24]为实验平台,利用网络带宽控制器 Traffic Control(TC)实现带宽控制。实验平台的拓扑结构如图 2 所示。

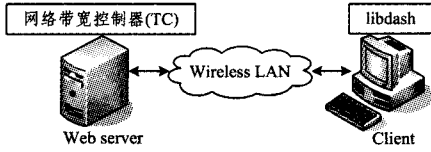


图 2 实验环境拓扑结构

实验数据选用预编码为 8 种不同码率 $R = \{100, 200, 350, 500, 700, 900, 1100, 1300\}$ kbps 的“Big Buck Bunny”^[1], 每个视频片段的持续时间 $\tau = 2s$ 。

4.2.1 收敛速率 k 的评估

探测的收敛速率 k 的设置将对算法性能产生较大的影响。对参数 k 设置不同的取值,通过在限定 1M 的链路带宽下持续 100s 下载视频数据,以验证其对带宽估算的稳定性的影响。

如图 3 所示, k 的取值从 0.04 到 0.63 以 0.07 的倍数变化,且随着 k 值的增加,收敛的速率变大。当 $k = 0.63$ 时,估算的带宽出现了在目标值左右震荡的现象,而之前的取值都能大致收敛到 1M。为了方便实验比较,选择 0.14 作为 k 的默认取值。

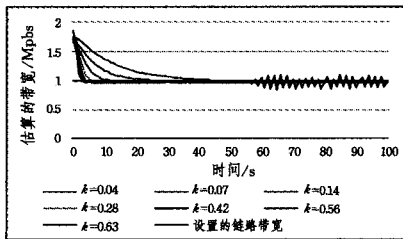


图 3 收敛系数 k 对带宽估算稳定性的影响

4.2.2 参数设置和实验情况分类

算法性能比较的实验中,参数设置取 $k = 0.14, \beta = 0.2, m = 5$,客户端缓存容量 $B_{max} = 15$,缓存均衡级别 $B_r = 10$,量化安全边距 $\epsilon = 0.15$,缓存上溢阈值 $B_{up} = 80\% \cdot B_{max}$,缓存下溢阈值 $B_{down} = 20\% \cdot B_{max}, p = 20$ 。量化函数为 dead-zone 量化器。

实验过程为分别由 CBB 算法、BS 算法和 PANDA 算法驱动的客户在以下 4 种实验情况中持续 200s 下载视频片段,得到对应 4.1 节中的评价指标的相关数据。

设计 4 种实验情况:1)链路维持固定带宽 1000kbps; 2)试验带宽骤降,0~100s 设置链路带宽为 1000kbps,100~200s 链路带宽下降为 500kbps;3)带宽突升,0~100s 设置链路带宽为 500kbps,100~200s 链路带宽上升为 1000kbps; 4)0~160s 带宽短期内多次波动,0~100s 设置链路带宽为 1000kbps,100~150s 后每隔 10s 降低或升高带宽,直至 150s 后维持在 1000kbps。运行 libdash,在以上 4 种实验环境中进行多次实验,发现 3 种算法在不同实验情况下得到的性能分析都较稳定,于是能够任选一组实验数据做后文的各类结果分析。

4.2.3 CBB 算法效果分析

图 4 描绘的是第 4 种实验情况,即带宽短期内多次波动,采用 CBB 算法的客户端获得码率和缓存数据量百分比的变化。根据图 4 中设置的链路带宽和选择的视频码率的比较,观察到 CBB 算法选取的视频码率波形较平整,没有过多的因码率切换引起的毛刺现象,说明码率能够较平地切换;右纵坐标是缓存数据量百分比,在 200s 的下载期间缓存数据量百分比基本处于 20%~80%之间。当链路带宽出现大幅度下降时,CBB 算法也能迅速切换低级别的码率,防止缓存区下溢。

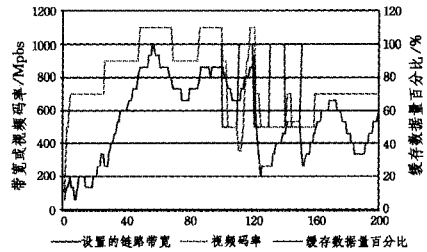


图 4 带宽短期内多次波动时 CBB 算法的码率和缓存数据量百分比的变化

4.2.4 性能对比

BS 算法在带宽估计中直接将测量 TCP 下载吞吐量作为实时带宽。PANDA 算法虽然也是基于探测的原理估算带宽,但是其平滑因子设定为固定值。每种实验环境中,任意选取一组实验数据,对其做均值处理,将本文提出的 CBB 算法分别从带宽估计和平滑函数设计两个角度与 BS 算法、PANDA 算法的结果进行比较。

缓存的上、下溢性越低,说明算法避免缓存发生上、下溢的性能越佳。如图 5 所示,4 种实验情况下,CBB 算法的缓存上溢性都明显低于 BS 算法和 PANDA 算法的,说明 CBB 算法在阻止缓存上溢方面性能最佳,能最大程度阻止缓存长时间爆满。观察图 6 中 3 种算法的缓存下溢指数变化,CBB 算法应对稳定带宽、带宽骤降和带宽短期内多次波动 3 种情况时,其缓存下溢性明显低于其他两种算法,遇到带宽突升的情况时,CBB 算法的缓存下溢性低于 PANDA 算法,但是高于 BS 算法。考虑带宽突升说明网络状况变好,缓存一般也不会有下溢的危险,所以下溢性指标评价整体算法性能的力度不大。

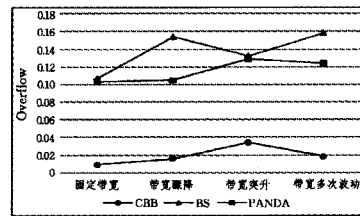


图 5 缓存上溢性

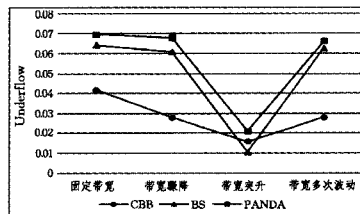


图 6 缓存下溢性

由图5和图6可知,相比于另外两种算法,CBB算法整体的设计即在带宽估计和平滑函数构建方面的改进及确定调度时间的策略,能够降低缓存区数据量溢出的概率。

图7和图8中的低效率性和不稳定性都是越低其算法的性能越好。

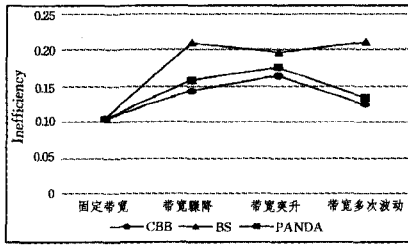


图7 低效率性

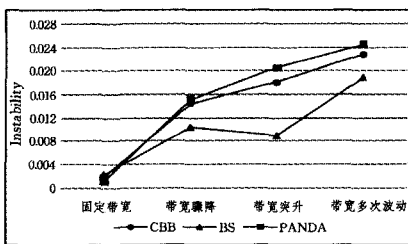


图8 不稳定性

如图7所示,4种实验情况下,CBB算法和PANDA算法有类似的低效率性变化趋势,主要因为两者的带宽估计都是基于探测的原理。CBB算法的低效率性明显低于BS算法,可见CBB算法的基于探测的带宽估计方法的优势,即不论在固定带宽还是网络带宽波动的环境中都能维持较高的网络利用率。如图8所示,BS算法的不稳定性最低,这主要是因为其码率的切换主要依赖于缓存的变化,当带宽发生变化时不能及时做出反应,码率的切换频率较小。在带宽发生波动时,CBB算法的不稳定性低于PANDA算法,说明CBB算法设计的随缓存状态动态变化的平滑因子对带宽平滑处理起到较好的作用,在避免频繁的码率切换方面性能优于PANDA算法。综合这两图可知,运用CBB算法能够充分利用网络带宽,保证较高的平均视频质量,使码率较平滑地切换。

结束语 本文提出一种综合考虑网络带宽和缓存状态的混合码率自适应算法(CBB);采用自适应探测方法预测网络带宽,尽量接近实时链路带宽;设计随缓存状态变化的平滑因子模型,动态地改变带宽平滑程度。实验结果表明,CBB算法在使缓存区数据量处于均衡范围和提高网络带宽利用率方面具有显著优势。但是文中只验证了CBB在单客户环境下的效果,其在减少码率切换方面的性能还有待提高。未来将在多客户竞争带宽的环境下实验,以验证采用CBB算法的客户端共享带宽的公平性,从而提高视频码率切换的稳定性。

参考文献

[1] ZHU X C, SHEN S B. 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.
[2] BEGEN A C, AKGUL T, BAUGHER M. Watching video over the web; Part 1: Streaming Protocols[J]. IEEE Internet Computing, 2011, 15(2): 54-63.
[3] BEGEN A C, AKGUL T, BAUGHER M. Watching video over the web; Part 2: Applications, standardization, and open issues [J]. IEEE Internet Computing, 2011, 15(3): 59-63.
[4] CHEN L W, LI G P, TENG G W, et al. Adaptive hybrid rate control algorithm based on HTTP streaming [J]. Journal of Shanghai University (Natural Science), 2014, 20(3): 313-320. (in Chinese)
陈立伟,李国平,滕国伟,等.基于HTTP流化的自适应码率混合控制算法[J].上海大学学报(自然科学版), 2014, 20(3): 313-320.
[5] LIU C, BOUAZIZI I, GABBOUJ M. Rate adaptation for adaptive HTTP streaming [C]// Proceedings of the Second Annual ACM Conference on Multimedia Systems. New York: ACM, 2011: 169-174.
[6] THANG T C, HO Q D, KANG J W, et al. Adaptive streaming of audiovisual content using MPEG DASH [J]. IEEE Transactions on Consumer Electronics, 2012, 58(1): 78-85.
[7] MUELLER 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: ACM, 2012: 37-42.
[8] LE H T, NGUYEN D V, NGOC N P, et al. Buffer-based bitrate adaptation for adaptive HTTP streaming [C]// 2013 International Conference on Advanced Technologies for Communications. Washington: IEEE, 2013: 33-38.
[9] YOU X Q. DASH scheduling algorithm based on a penalty factor [J]. Modern Electronics Technique, 2014, 37(16): 45-48. (in Chinese)
尤小泉.一种基于罚因子的DASH调度算法[J].现代电子技术, 2014, 37(16): 45-48.
[10] ZHU X C. Research on technology of network video transport over HTTP and its applications [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2013. (in Chinese)
朱晓晨.基于HTTP的网络视频传送技术及其应用研究[D].南京:南京邮电大学, 2013.
[11] HESSES S. Design of scheduling and rate-adaptation algorithms for adaptive HTTP streaming [C]// Applications of Digital Image Processing XXXVI. San Diego: SPIE, 2013: 88560M-88560M-8.
[12] POZUECO L, PANEDA X G, GARCÍA R, et al. Adaptation engine for a streaming service based on MPEG-DASH [J]. Multimedia Tools and Applications, 2014, 74(18): 1-20.
[13] Apple Corporation. HTTP Live Streaming Overview [EB/OL]. [2014-02-11]. <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html>.
[14] Microsoft Corporation. Smooth Streaming protocol [EB/OL]. [2015-06-30]. <http://msdn.microsoft.com/en-us/library/ff469518.aspx>.

- Puerto Vallarta, Mexico, 2012; 305-307.
- [2] MA Z C, SUN Y N, MEI T. Survey on wireless sensors network[J]. Journal of China Institute of Communications, 2004, 25(4): 114-124. (in Chinese)
马祖长, 孙怡宁, 梅涛. 无线传感器网络综述[J]. 通信学报, 2004, 25(4): 114-124.
- [3] SAKTHIDHARAN G R, CHITRA S. A survey on wireless sensor network; an application perspective[C]//Proceedings of the International Conference on Computer Communication and Informatics. Coimbatore, India, 2012; 1-5.
- [4] HEINZELMAN W, CHANDRAKASAN A, BALAKRISHNAN H. Energy-efficient communication protocol for wireless microsensor networks[C]//Proceedings of the International Conference on System Sciences. Hawaii, USA, 2000; 3005-3014.
- [5] LIU Y H, ZHAO Y F, XU K H, et al. Improvement of LEACH in wireless sensor networks[J]. Computer Engineering and Applications, 2010, 46(17): 117-120. (in Chinese)
刘玉华, 赵永峰, 许凯华, 等. 无线传感器网络 LEACH 协议的改进[J]. 计算机工程与应用, 2010, 46(17): 117-120.
- [6] HEINZELMANWR, ANANTHAP, CHANDRAKASANHB. An application specific protocol architecture for wireless microsensor networks [J]. IEEE Transactions on Wireless Communication, 2002, 1(4): 660-670.
- [7] SHI W R, BAI D, GAO P, et al. Cluster-head range adaptive adjustment clustering routing protocol for wireless sensor networks [J]. Chinese Journal of Scientific Instrument, 2012, 33(8): 1779-1785. (in Chinese)
石为人, 柏荡, 高鹏, 等. 无线传感器网络簇头半径自适应调节路由算法[J]. 仪器仪表学报, 2012, 33(8): 1779-1785.
- [8] JIANG C J, SHI W R, TANG X L, et al. Energy-Balanced unequal clustering routing protocol for wireless sensor networks [J]. Journal of Software, 2012, 23(5): 1222-1232. (in Chinese)
蒋畅江, 石为人, 唐贤伦, 等. 能量均衡的无线传感器网络非均匀分簇路由协议[J]. 软件学报, 2012, 23(5): 1222-1232.
- [9] MHATRE V, ROSENBERG C. Design guidelines for wireless sensor networks; Communication, clustering and aggregation [J]. Ad Hoc Networks, 2004, 2(1): 45-63.
- [10] BANDYOPADHYAY S, COYLE E J. An energy efficient hierarchical clustering algorithm for wireless sensor networks[C]//IEEE Societies INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE, 2003, 3: 1713-1723.
- [11] MHATRE V, ROSENBERG C. Homogeneous vs heterogeneous clustered sensor networks; a comparative study[C]//2004 IEEE International Conference on Communications. IEEE, 2004, 6: 3646-3651.
- [12] LEE S, CHOE H, PAR B, et al. An energy-efficient unequal clustering algorithm using location information for wireless sensor networks[J]. Wireless Personal Communications, 2011, 56(4): 715-731.
- [13] LI C F, CHEN G H, YE M, et al. An Uneven Cluster-Based Routing Protocol for Wireless Sensor Networks [J]. Chinese Journal of Computers, 2007, 30(1): 27-36. (in Chinese)
李成法, 陈贵海, 叶懋, 等. 一种基于非均匀分簇的无线传感器网络路由协议[J]. 计算机学报, 2007, 30(1): 27-36.
- [14] LIU Z, QIU Z D. Ring based multi-hop clustering routing on algorithm for wireless sensor networks[J]. Journal of China Institute of Communications, 2008, 29(3): 104-113. (in Chinese)
刘志, 裘正定. 基于分环多跳的无线传感器网络分簇路由算法[J]. 通信学报, 2008, 29(3): 104-113.
- [15] LU S, XU W C, YANG Y. A ring based multi-hop clustering routing weighting algorithm in wireless sensor networks [J]. Journal of Shandong University (Engineering Science), 2012, 42(4): 24-28. (in Chinese)
鲁松, 徐文春, 杨云. 一种分环多跳的无线传感器网络分簇路由加权算法[J]. 山东大学学报(工学版), 2012, 42(4): 24-28.

(上接第 134 页)

- [15] Adobe. HTTP Dynamic Streaming on the Adobe Flash Platform [EB/OL]. https://bugbase.adobe.com/index.cfm?event=file.view&id=2943064&seqNum=6&name=httpdynamicstreaming_wp_uc.pdf.
- [16] Dynamic Adaptive Streaming over HTTP [EB/OL]. https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP.
- [17] STOCKHAMMER T. Dynamic adaptive streaming over HTTP: standards and design principles[C]//Proceedings of the Second Annual ACM Conference on Multimedia Systems. New York: ACM, 2011; 133-144.
- [18] THANG T C, LE H T, PHAM A T, et al. An evaluation of bitrate adaptation methods for HTTP live streaming[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(32): 693-705.
- [19] LIU C, BOUAZIZI I, HANNUKSELA M M, et al. Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network[J]. Signal Processing; Image Communication, 2012, 27(4): 288-311.
- [20] ROMERO L R. A dynamic adaptive HTTP streaming video service for Google Android[C]//Communication Systems Cos. 2011; 33-51.
- [21] LIN Rui. The design and implementation of multi-rate adaptive streaming transmission system based on server-driven[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2013. (in Chinese)
林锐. 基于服务器驱动的多码率自适应流媒体传输系统的设计与实现[D]. 南京: 南京邮电大学, 2012.
- [22] LI Z, ZHU X, 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.
- [23] JIANG J, SEKAR V, ZHANG H. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive[C]//Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies. New York: ACM, 2012; 97-108.
- [24] MUELLER C, LEDERER S, POECHER J, et al. Demo paper: Libdash-an open source software library for the mpeg-dash standard[C]//2013 IEEE International Conference on Multimedia and Expo Workshops. Washington: IEEE, 2013; 1-2.