

视频平滑算法应用研究

李争明¹, 叶德建², 张 佐¹

(1. 清华大学 自动化系, 北京 100084; 2. 复旦大学 软件学院, 上海 200433)

(lzm02@mails.tsinghua.edu.cn)

摘 要:分析和解决了平滑算法应用过程中遇到的两个具体问题:(1)如何应用平滑算法使其不影响音视频在拖动操作后的同步性;(2)如何将平滑算法得到的发送方案同 RTP 协议中关于视频数据打包的规定相融合。

关键词:流媒体; RTP; MPEG 4; 平滑算法; 同步

中图分类号: TP317.4 **文献标识码:** A

Research on application of traffic smoothing algorithm

LI Zheng-ming¹, YE De-jian², ZHANG Zuo¹

(1. Department of Automation, Tsinghua University, Beijing 100084, China;

2. School of Software, Fudan University, Shanghai 200433, China)

Abstract: The application of traffic smoothing algorithms (mainly WTS) in the context of developing a high-performance streaming server was studied. Two practical problems were analyzed and solved: one was to prevent the application of smoothing algorithm from bringing impacts on synchronization of audio and video streams on the client ends, and the other was to modify the sending plans produced by smoothing algorithms in order to make them applicable in RTP transferring. These two problems were studied and analyzed, and corresponding solutions and algorithms were presented and validated by experiments.

Key words: streaming media; RTP; MPEG 4; smoothing algorithm; synchronization

0 引言

流媒体是一种前景广阔的网络应用。目前,流媒体应用中广泛采用 MPEG4 视频标准进行编码,并采用 RTP 协议进行数据传输。

流媒体应用要求视频和音频连续、清晰和实时地播放,需要充足的网络带宽和稳定的网络环境的支持。而现有的网络通常不能满足这种需求,因此需要采用控制算法对流媒体数据的发送过程进行控制,使其适应于波动的网络环境。这些控制算法主要有速率平滑算法和质量自适应算法两类。

的带宽峰值,并减小发送速率的波动性。

平滑算法的效果可以由图 1 直观的表现出来:图中虚线部分为某个电影的视频流码率,实线部分为一种平滑算法的发送方案。

质量自适应算法则在网络带宽不足时按照一定的策略调整视频质量,如采用 Scalable 编码等,以降低视频流的码率,使之能顺利发送到客户端,保证客户端的正常播放。

为提高性能,流媒体服务器应尽量集成平滑算法和质量自适应算法。

1 平滑算法简介

平滑算法本质是对视频流的发送过程进行调度,在缓冲区条件的约束下得到较平滑的发送方案^[1]。缓冲区条件具体描述为:在发送数据的过程中需保证缓冲区不发生上溢和下溢,即任意时刻缓冲区中的数据量既不能为 0 又不能大于缓冲区的容量。上溢会导致视频数据丢失,下溢则会引起播放停顿,两者都会严重影响视频播放质量。

学术界中平滑算法研究成果很多:W. Feng 等在文献[2]中给出的针对预先存储视频流的 MCBA 算法,在做平滑的时候着重使发送方案的带宽变化次数最小;J. M. McManus 等在文献[3]提出了 MVBA 算法,得到的发送方案具有最小的带宽变化方差;J. D. Salehi 等在文献[4]提出了 PRCTT 算法,得到带宽变化次数以及带宽方差都比较小的发送方案;Dejian Ye 等在文献[5, 6]中提出了 WTS (Wavelet-based Traffic Smoothing) 算法,可以得到在带宽变化次数、方差、带宽峰值等多个指标上都比较优化的发送方案,而且算法复杂度较低。

其中 WTS 算法针对 VBR 视频流的码率特性构造一个相

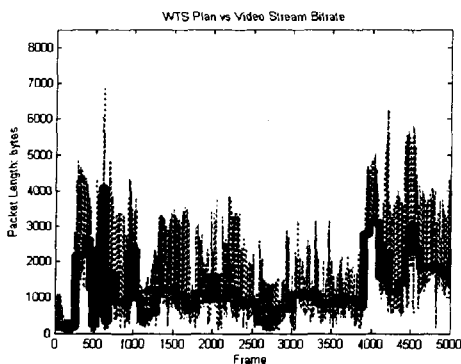


图 1 平滑算法效果图

速率平滑算法 (Rate Smoothing) 简称平滑算法,对 VBR 视频流进行平滑操作。利用对视频流码率的先验知识,平滑算法在视频流码率波峰到来之前预先向客户端缓冲区中发送多余的数据,这样在波峰到来时播放器可以从缓冲区中读取数据进行播放,而不必要求服务器的发送速率马上提高到相应的码率水平。这样,平滑算法可以降低发送视频流所需要

收稿日期:2005-03-24;修订日期:2005-05-23

作者简介:李争明(1981-),男,安徽萧县人,硕士研究生,主要研究方向:流媒体质量控制; 叶德建(1976-),男,浙江椒江人,副教授,主要研究方向:流媒体; 张佐(1967-),女,教授,主要研究方向:智能交通系统、离散事件动态系统。

应的二叉树,树叶取值是各个视频帧大小,各个节点取值是对其树叶 Haar 小波变换的系数。对此二叉树从根到叶进行搜索而得到符合要求的发送方案。

理论分析和仿真实验表明此算法可以得到在上述多个指标上都较优化的发送方案,且可以得到多个解。另外,该算法计算复杂度只有 $O(N \log N)$ [5,6], 较其他算法小。所以 ClearServer 选择应用 WTS 平滑算法对视频流进行速率平滑。

2 WTS 在 Clear Server 中的实现架构和问题

由于流媒体系统多采用 RTP 协议进行数据传输,所以 WTS 或其他平滑算法在视频服务器中的实现必须考虑与 RTP 协议的兼容。RTP 协议族明确规定了在对 MPEG 系列视频流进行 RTP 数据打包时的几种可能情况:

- (1) 一个视频帧作为一个 RTP 数据包发送(1:1);
- (2) 一个视频帧作为多个 RTP 数据包发送(1:n);
- (3) 多个视频帧作为一个 RTP 数据包发送(m:1)。

考虑到实际网络的 MTU 的限制(RTP 数据包的大小不能超过 MTU),大多数视频帧都需要打成多个 RTP 数据包进行发送。

在对视频流数据进行 RTP 打包时可以有两种思路:(1)以视频帧为单位考虑 RTP 打包:首先根据发送策略将视频流的各视频帧组合成逻辑上的 RTP 数据包或称发送单元,然后在发送过程中再考虑 MTU 的约束将已确定的发送单元分成实际的 RTP 数据包发送出去。(2)以 MTU 为基准进行 RTP 数据打包:在 RTP 数据打包时预先考虑 MTU 的约束,然后再考虑实际视频帧的界限对打包过程的约束。

由于平滑算法并没有将具体的网络传输协议约束考虑在内,得到的发送方案不一定能满足 RTP 关于数据打包的规定,所以在服务器中采用以视频帧为单位进行 RTP 打包的策略时,平滑算法的发送方案并不能直接应用。

另外,由于平滑算法的应用修改了媒体数据包的发送时间,WTS 还会影响音视频数据的同步和交织。

一般来说,服务器在响应客户端的拖动操作时会通过两种方式去查找新的音频数据和视频数据起点:按照发送时间查找和按照播放时间查找。第一种方式下,服务器将客户端传来的时间信息转换成为视频流和音频流的发送时间,利用新的发送时间定位数据。第二种方式下,服务器将按照客户端传来的时间信息找到具有对应播放时间的视频和音频数据。

由于速率平滑实质是对视频数据的发送过程进行调度,所以按照发送时间定位数据时,集成 WTS 后视频数据的发送时间发生改变,音频视频数据之间将出现一定的时间错位,可能造成客户端拖动操作后音视频失去同步。

考虑到服务器系统实现的难易及可扩展性等因素,ClearServer 中采用了以帧为单位进行 RTP 打包并按照发送时间定位数据的策略。结合本章的讨论,可知在这种 WTS 实现模式下需要解决两方面的问题:

问题一:在以视频帧为单位考虑 RTP 打包的实际情形下,需要调整 WTS 算法的发送方案使之满足 RTP 数据打包要求。

问题二:由于按照发送时间定位数据时集成 WTS 后视频帧的发送时间发生改变,需要采取措施保证客户端音视频播放的同步性。

3 解决方案

3.1 问题一解决方案及其效果

3.1.1 问题分析

为解决问题一,一方面要根据 RTP 协议对 WTS 发送方

案进行调整,另一方面要保留 WTS 算法的平滑效果。调整算法简称 EWTS 算法。

目前流媒体系统中客户端缓冲区都比较大,一般不会出现上溢。所以本文有以下假设:(1)方案调整时应着重防止缓冲区下溢;(2)将任意时刻的发送数据总量控制在 WTS 发送方案附近就能保证不溢出,因为 WTS 算法的发送方案满足缓冲区条件^[5,6]。

对 WTS 发送方案进行调整的思路描述如下:根据视频流码率分布对 WTS 方案进行调整,得到 EWTS 方案,并使之在满足 RTP 数据打包规定的条件下尽量接近 WTS 方案。为避免缓冲区下溢,采取以下措施:(1)若当前缓冲区占用量较小,则使 EWTS 方案发送单元大于 WTS 方案发送单元;(2)若缓冲区占用量比较大,则无需考虑缓冲区下溢的问题,只需使 EWTS 方案发送单元尽量接近 WTS 方案;(3)为防止缓冲区数据累积,缓冲区占用量达到一定程度时使 EWTS 方案发送单元小于 WTS 方案发送单元。

3.1.2 调整算法具体描述

符号说明:(1) $f_i, i = 1, 2, \dots, N$ 是视频帧的大小, N 为帧数; $S_i, i = 1, 2, \dots, N_i$ 是 WTS 发送方案, N_i 是发送单元数; $M_i, i = 1, 2, \dots, N_m$ 是 EWTS 发送方案, N_m 是其发送单元数;(2) k_i 是 WTS 发送单元序号, $\Delta k_i = k_{i+1} - k_i$; k_m 是 EWTS 方案发送单元序号, $\Delta k_m = k_{m+1} - k_m$;(3) E_{k_i} 为当前确定的 EWTS 方案同 S_{k_i} 之前 WTS 方案的数据总量之差;(4) T 为缓冲区占有量上界,一般取缓冲区的大小; α 为常数系数; P_1 、 P_2 、 P_3 分别为 WTS 方案、EWTS 方案及视频流的峰值带宽。

调整算法具体如下:

STEP0: 令 $k_i = 0, k_m = 0, S_0 = 0, E_0 = 0$ 。

STEP1: 若 $E_{k_i} \leq 0$ 转 STEP2; $0 < E_{k_i} \leq T$ 则转 STEP3; 若 $E_{k_i} > T$ 转 STEP4。

STEP2: 令 $\Delta k_m = 1$, 求解

$$\begin{aligned} & \min (M_{k_m+1} - S_{k_i+1}) \\ \text{s.t. } & M_{k_m+1} > S_{k_i+1} \\ & M_{k_m+1} \leq \alpha P_1 \\ & \Delta k_i \geq 1 \end{aligned}$$

确定 Δk_i ; 转 STEP5。

STEP3: 求解

$$\begin{aligned} & \min |M_{k_m+1} - S_{k_i+1}| \\ \text{s.t. } & \Delta k_m \geq 1, \Delta k_i \geq 1 \\ & (\Delta k_m - 1)(\Delta k_i - 1) = 0 \\ & M_{k_m+1} \leq \alpha P_1, \quad i = 1, \dots, \Delta k_m \end{aligned}$$

确定 Δk_m 和 Δk_i ; 转 STEP5。

STEP4: 令 $\Delta k_i = 1$, 求解

$$\begin{aligned} & \min (S_{k_i+1} - M_{k_m+1}) \\ \text{s.t. } & M_{k_m+1} \leq S_{k_i+1} \\ & \Delta k_m \geq 1 \end{aligned}$$

确定 Δk_m ; 转 STEP5。

STEP5: 根据 Δk_m 和 Δk_i , 令

$$E_{k_i+\Delta k_i} = E_{k_i} + \sum_{j=1}^{\Delta k_m} M_{k_m+j} - \sum_{i=1}^{\Delta k_i} S_{k_i+i}, \text{ 并令 } k_i = k_i + \Delta k_i, k_m = k_m + \Delta k_m; \text{ 若 } k_i < N \text{ 转 STEP1, 否则结束循环。}$$

算法中, M_{k_m+i} 的计算公式如下:

$$M_{k_m+i} = \begin{cases} \sum_{i=1}^{\Delta k_i} f_{k_i+i}, & \Delta k_m = 1 \\ \frac{f_{k_i}}{\Delta k_m}, & \Delta k_i = 1 \end{cases}$$

3.1.3 算法有效性分析

(1) 算法中 Δk_m 和 Δk_i 不同时大于 1, 可保证新方案满足 RTP 的打包规定。

(2) 算法中有关 $M_{k_{m+i}}$ 的约束可保证 EWTS 方案的发送单元都小于 αP_1 , 因而有 $P_2 \leq \alpha P_1$, α 是松弛系数。一般 $P_3 \gg P_1$, 所以 α 取值适当时 P_2 一般小于 P_3 。

(3) 本算法等价于负反馈系统, 其稳定性是可以保证的, E_{k_i} 不会发散。

(4) STEP2、STEP3 与 STEP4 均在不同条件下使 EWTS 方案的发送单元同 WTS 方案发送单元最接近, 使 EWTS 方案总体上在 WTS 方案附近波动。

3.1.4 算法仿真结果

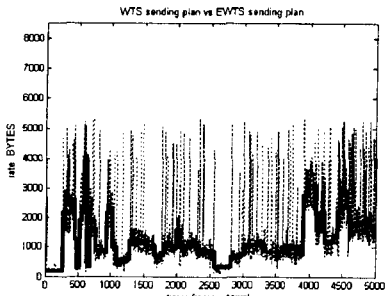
由于考虑平滑算法的效果时一般只考察平滑算法的发送方案相对于视频流码率的平滑效果^[1], 所以本文针对典型的视频流码率制定可行的发送方案来验证调整算法的效果。

对 Adam Wolisz^[7] 提供的《Jurassic Park》MPEG4 (low quality) 视频流码率数据进行仿真实验。利用 WTS 算法得到 WTS 发送方案, 利用 EWTS 算法调整得到 EWTS 发送方案。

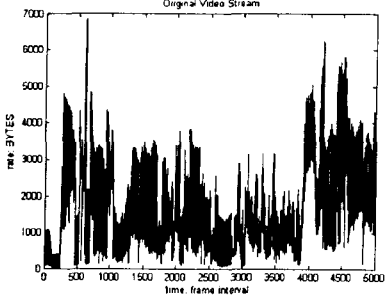
视频流、WTS 方案和 EWTS 方案的峰值带宽、带宽变化方差等指标的比较结果如表 1。

表 1 指标比较

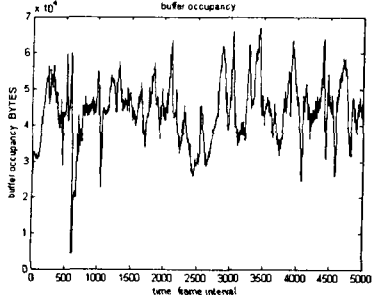
指标/发送方案	视频流	WTS 方案	EWTS 方案
峰值带宽 (bytes/frame)	8 511	4 988	5 345
带宽方差	0.710 6	0.294 0	0.432 2



(a) WTS方案和EWTS方案码率



(b) 视频流码率曲线



(c) EWTS方案下客户端缓冲区占用情况

图 2

可见 EWTS 方案在峰值带宽和方差上相对于视频流有很大的改善, 基本保留了 WTS 方案的效果。

图 2 为码率变化曲线和缓冲区占用量曲线, 为方便画图只取其中的前 5000 个帧。图 2(a) 中粗实线为 WTS 方案的曲线, 虚线为 EWTS 方案的码率曲线; 可见 EWTS 方案基本上跟踪了 WTS 方案的变化趋势。图 2(b) 是视频流码率变化曲线。图 2(c) 则是 EWTS 方案的缓冲区占用情况, 可见缓冲区没有下溢。

对比之下可以得出: EWTS 方案相比视频流码率要平滑, 并跟踪了 WTS 方案的变化趋势。可见调整算法可以有效保留 WTS 算法的效果, 并得到同 RTP 协议兼容的发送方案。

3.2 问题二解决对策

问题二的关键是 WTS 改变了视频数据的发送时间, 使得拖动后有可能出现音频或视频数据的漏传, 造成音频视频的不同步。所以解决思路应该是保持播放时间相同的音频视频数据的发送时间相近。本文提出一种简单易行的方法: 根据新发送方案改变视频数据发送时间的同时相应修改音频数据的发送时间, 以保持音频视频数据的相对位置不变。

具体方法 (针对 MPEG4 数据打包的三种可能情况): (1) 某视频帧要分成多个 RTP 数据包时, 将其后各音频帧的发送时间相应增加; (2) 多个视频帧打成一个 RTP 数据包时, 将其后的各音频帧的发送时间相应减小; (3) 某视频帧打成一个 RTP 数据包时, 其后音频帧发送时间不变。这样客户端拖动后新的发送时间点后的音频视频的播放时间比较接近, 防止了音频数据或者视频数据漏传, 从而消除了 WTS 算法带来的同步隐患。

应用此对策后, 消除了同步隐患。点播实验证明无论正常播放还是进行拖动操作, 服务器系统都可以保证客户端享受到同步的音频和视频。

4 结语

本文提出的两个问题及其解决方案虽然是针对 WTS 的实际应用而提出的, 但同时也适用于其他平滑算法在 RTP 协议发送 MPEG 系列视频流中的应用, 具有一定的实际意义和参考价值。

参考文献:

[1] FENG WC, REXFORD J. A Comparison of Bandwidth Smoothing Techniques for the Transmission of Preencoded Compressed Video [A]. Proceedings of INFOCOM '97, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies [C], 1997. 58 - 66.

[2] FENG W, JAHANIAN F, SECHREST S. Smoothing and Buffering for delivery of preencoded compressed video [J]. ACM Multimedia Syst. J, 1997, (9): 297 - 309.

[3] MCMANUS JM, ROSS KW. Video on demand over ATM: constant-rate transmission and transport [J]. IEEE J. Select. Areas Comm., 1996, (14): 1087 - 1098.

[4] SALEHI JD, ZHANG ZL, KUROSE JF, et al. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing [J]. IEEE/ACM Trans. Networks, 1998, 6(4): 397 - 410.

[5] YE DJ, XIONG ZX, et al. Wavelet-based Smoothing and Multiplexing for VBR Video Traffic [A]. Proceeding of IEEE. Globecom 2001 [C], 2001. 25 - 29.

[6] YE D, BARKER C, et al. Wavelet-based VBR video traffic smoothing [J]. IEEE Trans. Multimedia, 2004, (6): 611 - 623.

[7] WOLISZ A. Terse_Jurassic_10. dat [DB/OL]. <http://www-tnk.ee.tu-berlin.de/research/trace/ltvt.html>, 2004-1-10/2005-03-29.