

基于 MPEG-4 的流媒体服务器端缓冲区管理

杨格兰¹, 曹理宇², 张建明¹

(1. 湖南城市学院计算机科学系, 益阳 413049; 2. 中国人民解放军 95147 部队, 兴宁 510200)

摘要: 服务器端缓冲技术是流媒体系统中常用的一种应用控制技术, 能够平滑网络抖动, 减少数据丢失, 提高客户端视频播放质量。该文的缓冲区管理增加了码率控制的功能, 控制策略能进一步提高视频质量, 特别适合于系统资源有限、带宽窄或强实时性的视频流式传输。
关键词: 码率控制; MPEG-4; 流媒体; 缓冲区

Management of Server Buffer for Streaming Media System in MPEG-4

YANG Gelan¹, CAO Liyu², ZHANG Jianming¹

(1. Dept. of Computer Science, Hunan City University, Yiyang 413049; 2. 95147 Armies of PLA, Xingning 510200)

【Abstract】 Server buffer is an application-controlled approach that is widely used in streaming media system. It can improve the quality of video playing by smoothing the transmission jitter of network, decrease the loss of data and optimize the quality of video playing in client. The role of rate control is appended in buffer management section. The rate control optimizes video playing quality, especially for such system that resources are limited, bandwidth is narrowed or real-time property is intensively requested.

【Key words】 Rate control; MPEG-4; Streaming media; Buffer

由于当前 IP 网络的 Best-effort 设计思想和底层硬件的局限性还不能够提供充分可靠的服务质量保证(QoS), 因此在 C/S 模式传输时要引入服务器端缓冲技术来弥补当前网络传输的缺陷。MPEG-4 的视频编码流码率波动极大, 它比一般的 C/S 模式传输更需要引入缓冲区来吸收这种码率波动。服务器端缓冲区对于基于 MPEG-4 的流媒体系统是非常重要的。本文结合 MPEG-4 视频流编码的特点, 探讨服务器端缓冲区码率控制策略。

1 服务器端缓冲区描述

服务器端缓冲技术是一种应用级视频质量控制技术, 其思想是在服务器端划分一块内存作为缓冲区, 通过缓冲一定的数据量来平滑网络抖动。如图 1 所示, $filling_rate(t)$ 表示 t 时刻已经编码的视频流填入缓冲区的速度, 如果要求编码视频流无损失地填入缓冲区, 这时, $filling_rate(t)$ 实际上等于当前视频编码码率。 $draining_rate(t)$ 表示 t 时刻视频流流出缓冲区的速度, 因为视频流流出后随即将通过网络发送出去, 所以 $draining_rate(t)$ 实际上等于当前网络发送速率。 $Buffer_filled(t)$ 表示 t 时刻缓冲区已经缓存视频流的总大小。

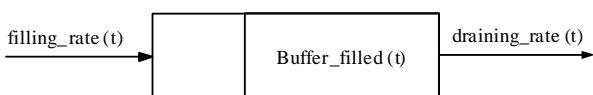


图 1 服务器端缓冲区描述

理想情况下, 视频流数据经过缓冲区不会产生损失, 这就要求填入缓冲区的数据总量必须等于流出缓冲区的数据总量, 即满足式(1)。

$$\int_0^{+\infty} filling_rate(t)dt = \int_0^{+\infty} draining_rate(t)dt \quad (1)$$

然而, 由前面的分析可以知道, $filling_rate(t)$ 取决于当前

的视频编码码率, $draining_rate(t)$ 取决于当前的网络状况, 而视频编码码率和网络状况二者之间并没有必然联系, 如果不在应用级加以控制, 式(1)是难以满足的。即便在应用级加以控制, 也无法保证缓冲区不溢出, 可见式(1)在现有的技术条件下实际上是无法满足的, 除非将缓冲区设置得足够大, 但是如果缓冲区足够大, 一则消耗大量的系统资源, 再则牺牲了流媒体系统的实时性, 故主要考虑在缓冲区有限的情况下如何在应用级加以有效控制来尽量满足式(1)的情况。

由于目前的 Internet 只能提供“尽力而为”服务, 无法控制 IP 分组的吞吐量、丢包率、时延和抖动, 因此网络状况和网络特性都随时间而不断变化。已有许多 QoS 管理机制, 如分配优先级和资源预留。分配优先级对不同的网络流量分配不同的优先级。但怎样为不同的网络流量设置优先级并将应用级别的优先级映射为不同的路由器优先级以及该方法所带来的性能增益仍在研究之中^[1]。资源预留(例如 RSVP 协议^[2])要求沿途所有的路由器支持该协议, 否则服务质量就得不到保证, 而目前大多数路由器没有此功能; 而且为提供 QoS 保证, RSVP 经常造成网络资源的过度分配, 未能充分利用网络资源; 同时由于预留状态信息与业务流个数成正比, 使得路由器的负担会随着网络业务流的增加而加重, 因此网络的可扩展性和鲁棒性不是很好。综上所述, 在现有的网络条件下, 为了满足式(1), 只好将调节的目标集中到 $filling_rate(t)$ 上, 即通过调节 MPEG-4 视频编码器的码率或控制编码流填入缓

基金项目: 湖南省教育厅科研基金资助项目(05C777)

作者简介: 杨格兰(1975 -), 男, 讲师、硕士, 主研方向: 图像处理, 视频技术; 曹理宇, 工程师、硕士; 张建明, 博士生

收稿日期: 2006-04-29 **E-mail:** yanggelan@126.com

缓冲区来达到目的。

2 缓冲区帧率控制算法

2.1 filling_rate(t)调节的依据

由前面描述可以知道,理想情况下,t时刻缓冲区已经缓存视频流的总大小 Buffer_filled(t)等于 0~t 时间段内缓冲区填入的数据总和减去 0~t 时间段内缓冲区流出的数据总和,可用式(2)表示。

$$Buffer_filled(t) = \int_0^t filling_rate(t)dt - \int_0^t draining_rate(t)dt \quad (2)$$

由式(2)变形成式(3):

$$\int_0^t filling_rate(t)dt = Buffer_filled(t) + \int_0^t draining_rate(t)dt \quad (3)$$

由式(3)可以清楚地看出,filling_rate(t)调节的依据有两种:(1)根据 draining_rate(t)的变化来调节;(2)是根据 Buffer_filled(t)的值来调节。

图2所示是一种典型的MPEG-4视频流网络传输架构^[3]。在发送端,由自适应的MPEG-4编码器对原始视频流进行编码、打包后传递给RTP/UDP/IP模块,通过网络传输到接收端。这些成功接收的包以发送端相反的顺序通过RTP/UDP/IP模块,由MPEG-4解码器进行解码。同时接收端根据数据包到达的情况计算出网络传输状况。通过实时传输控制协议(RTCP)反馈给发送端。发送端根据这些反馈信息调整编码参数,如文献[4]采用率失真模型调节编码参数。

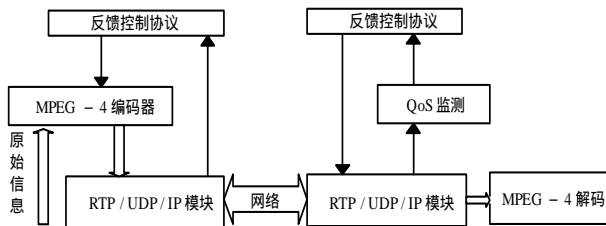


图2 典型的MPEG-4视频流网络传输架构

这种架构实质上是依据接收端计算出来的draining_rate(t)来调节filling_rate(t)。然而这种调节方式需要反馈控制协议的参与而且编码参数调整复杂,会增加系统资源开销和网络带宽的浪费。如果改成根据Buffer_filled(t)来调节filling_rate(t),就无须反馈控制协议的参与,直接用UDP协议传输也可达到码率控制的目的。

2.2 缓冲区码率控制算法

缓冲区码率控制算法分前后两部分:前一部分调整MPEG-4视频编码器编码参数来实现对filling_rate(t)的调节;后一部分根据各类帧重要性来淘汰重要性低的视频帧占据缓冲区,保证当前缓存的视频帧都是最重要的。

MPEG-4视频编码器的码率之所以能够调节主要有赖于它编码的可伸缩性。视频的可伸缩性包括空间可伸缩性、时间可伸缩性、信噪比可伸缩性3种^[3]。空间可伸缩性和信噪比可伸缩性影响视频流的画面质量,时间可伸缩性影响视频流的连续性,选择何种伸缩性实际上取决于画面质量和连续性之间的取舍。本文选取时间可伸缩性(即控制帧率)来调整MPEG-4视频编码码率。

衡量视频传输技术的好坏,不能单纯以如何提高网络利用率为标准。一方面,网络传输的视频流越多并不代表在解码时解出的有效视频帧越多。因为MPEG-4视频编码采用了帧间预测机制,如果参考帧丢了,在接收端由于错误传播将造成相当一段视频无法使用,所以在回放时会出现停顿现象。另一方面,网络传输的视频流越多并不代表它不连续性丢帧,

有时情愿接收的帧数少一些,只要视频帧丢失得较为均匀就行。本文提出的缓冲区码率控制算法解决了这两方面的问题。算法前一部分选取时间可伸缩性(即控制帧率)来调整MPEG-4视频编码码率除了起预防缓冲区溢出的作用外,实际上还起到了预防连续性丢帧的作用。至于如何提高传输视频帧的有效解码,可先从考察MPEG-4的3类帧入手。MPEG-4视频帧分为3类:I-VOP、P-VOP^[5]和B-VOP^[6]。为了方便表述不妨简称为I帧、P帧和B帧。3类编码帧的重要性依次为I帧、P帧和B帧。同样是P帧,前面的比后面的更重要(因为后面的P帧要参考前面的P帧才能解码)。在缓冲区不得不丢帧的情况下,先将缓冲区中重要性比要进入帧的重要性低的帧释放,直至有足够空间为止,如果缓冲区中该释放的帧都释放了还是没有足够空间,这时才把即将要进入的帧丢弃。本文提出的编码码率控制算法主要分两部分:编码前调整编码帧率和编码后丢帧控制,具体如下:

```
While (MPEG-4 视频流未结束) {
    If (缓冲区占有率<50%) { //编码前调整编码帧率
        按照原始帧率进行编码;
    }
    else {
        If (缓冲区占有率<70%) {隔帧编码}
        else {
            If (缓冲区占有率<90%) {隔两帧编码}
            else {隔三帧编码}
        }
    }
    If (缓冲区的空闲区>帧长) { //编码后丢帧控制直接进入缓冲区 }
    else { while(缓冲区的空闲区<帧长)
        { 从后向前找到一帧重要性低的 VOP 释放 }
        //只要空闲区能缓存要进入的帧,则跳出 while 循环
        If (缓冲区的空闲区<帧长)
            //若缓冲区中重要性低的帧都丢完了,但空闲区仍然不够放入要
            //进入的帧,则不要让该帧进入缓存区
            { 直接丢弃要进入缓冲区的帧 }
        Else { 进入缓冲区 } }
    }
}
```

2.3 算法有效性分析

算法的前一部分能够有效地防止缓冲区的溢出,这是因为在网络带宽一定的情况下传输占有带宽巨大的视频数据,网络拥塞是时有发生,当缓冲区占有率超过50%时,就假定它已经开始出现溢出的苗头,从而开始隔帧编码。这种假定在带宽资源很窄的网络中或在无线视频传输中尤为合理的,在这些情形下,缓冲区的数据还不一定马上就会被调度出去,而编码后的MPEG-4视频数据却要不断地进入缓冲区,如果此时不控制进入缓冲区的MPEG-4速率,缓冲区就会有溢出的危险。也正是这个原因才设定了缓冲区占有率的3个门限:50%,70%和90%。当占有率超过70%时,说明缓冲区更有可能要溢出了,要隔2帧编码。当占有率超过90%时,说明缓冲区溢出已迫在眉睫,要隔3帧编码。当然这3个缓冲区占有率的门限只是一个参考值,它可能只在当前试验的环境下实用,一般来说,当网络带宽确实很宽的情况下,可以适当将这3个门限提高,毕竟门限调得过低对视频质量是有影响的。

算法的后一部分考虑了MPEG-4的编解码原理,能够有效地提高视频质量。在MPEG-4视频序列中,I帧是最重要的,如果一个视频序列没有I帧,则整个视频序列都不能解

码出来。P 帧的重要性居次，因为后续视频帧将要以它作为参考帧进行解码，所以如果 P 帧丢失，后续视频帧将会解码失真或出错。

例如假设一个视频序列为 $IB_1B_2P_1B_3B_4P_2B_5B_6P_3B_7B_8$ 。如果 P_1 丢失，则 B_1 、 B_2 以及 P_1 后面的视频帧都将失去正确的参考帧进行解码，当然解码出来的帧会出现失真或错误。如果 P_2 丢失，则 B_3 、 B_4 以及 P_2 后面的视频帧都将失去正确的参考帧进行解码，当然解码出来的帧也会出现失真或错误。

由这个例子也可看出，同样是 P 类帧，前面的 P 帧要比后面的 P 帧更重要，因为前面的 P 帧丢失后所波及的范围要比后面 P 帧所波及的范围要大。因为 B 帧不对其它帧形成参考，所以它们的重要性都是相同的而且在 3 类帧中重要性是最低的，是要优先考虑丢弃的。综上所述，采用这种丢帧控制能够在缓冲区资源或带宽资源有限的情况下尽可能地提高视频质量。

2.4 仿真试验

采用 DirectShow 技术，开发了一个简单的 MPEG-4 视频流式传输系统，视频采用 CIF 格式，视频采用 XVID 编码器进行编码。在局域网中进行了多次模拟实验，为了让试验具有可重复性，视频源不采用动态采集方式，而是采用已经采集好的视频文件来模拟视频采集。该视频文件一共包含 7 328 个 VOP，帧速采用 PAL 制(24fps)，播放长度为 5min1s。根据缓冲区延迟的不同设计了以下 3 个试验，试验结果用表 1~表 3。

表 1 在服务器端缓存 1min 视频后才开始由网络发送的情形

	缓冲区溢出次数	网络传输帧数	有效解码帧数	主观感受
缓冲区码率控制	0	7 322	7 322	流畅
无缓冲区码率控制	0	7 328	7 328	流畅

从 3 个表可以看出：在缓冲区足够大时(如表 1)，缓冲区码率控制的作用并不明显，甚至还不如不用无缓冲区码率控制。但是当缓冲区变小时(如表 2、表 3)，虽说二者在网络传输帧数方面相差不大，但有效解码帧数却相差极大。如果将有效解码帧数除以网络传输帧数作为有效解码率，则可以很清楚地看出，随着缓冲区的变小，采用缓冲区码率控制的系统有效解码率和缓冲区溢出的次数没有很大波动，而没有采

用缓冲区码率控制的系统有效解码率急剧下降，缓冲区溢出次数急剧增加，视频质量也越来越差。

表 2 在服务器端缓存 40s 视频后才开始由网络发送的情形

	缓冲区溢出次数	网络传输帧数	有效解码帧数	主观感受
缓冲区码率控制	0	7 319	7 319	流畅
无缓冲区码率控制	2	7 318	7 297	比较流畅

表 3 在服务器端缓存 20s 视频后才开始由网络发送的情形

	缓冲区溢出次数	网络传输帧数	有效解码帧数	主观感受
缓冲区码率控制	0	7 303	7 303	比较流畅
无缓冲区码率控制	7	7 305	7 265	不流畅

3 结论

本文提出的服务器端缓冲区码率控制算法充分考虑了 MPEG-4 视频编码流的特点，能够很好地预防缓冲区溢出和提高有效解码率，在不得不丢帧的情况下能够比较均匀地丢帧，进一步提高了视频质量。本算法非常适合系统资源有限、带宽窄或强实时性的视频流式传输。

参考文献

- Shin J, Kim J W, Kuo C-C J. Content-based Packet Video Forwarding Mechanism in Differentiated Service Networks[C]//Proc. of 10th Int. Workshop Packet Video, Sardinia, Italy. 2000-05.
- Braden R, Zhang L, Berson S, et al. Resource Reservation Protocol(RSVP)——Ver.1 Functional Specification[S]. RFC 2205, 1997.
- Peng D. On End-to-end Transport Architecture for MPEG-4 Video Streaming over the Internet[J]. IEEE Trans. on Circuits Syst. Video Technol., 2000, 10(6).
- Chiang T, Zhang Y Q. A New Rate Control Scheme Using Quadratic Rate Distortion Model[J]. IEEE Trans. on Circuits Syst. Video Technol., 1997, 7(1).
- Li W. Overview of Fine Granularity Scalability in MPEG-4 Video Standard[J]. IEEE Trans. on Circuits Syst. Video Technol., 2001, 11(3).
- ISO/IEC JTCE1/SC29/WG11, N3342. Overview of the MPEG-4 Standard[S]. 2000-03.

(上接第 207 页)

4 实验结果

选择 9/7 小波，采用第(4)种边界处理办法，对图像实施 3 级小波变换，然后对小波系数进行量化^[5]和算术编码。所开发的应用系统，对 8bpp86 × 109 图像的压缩结果与 JPEG 的比较：图 5(b)是 JPEG 算法压缩的图像，图 5(c)是使用 JPEG2000 算法压缩 35 倍的彩色图像，可以看出，JPEG 算法压缩产生了明显的方块效应；而本文算法有很好的效果。

5 结束语

为了实现图像的高效压缩，即 JPEG2000，我们使用了 mallat 算法，并且在开发过程中，逐个探讨并解决了滤波器的选取、边界效应等实际问题。对该算法的深入研究，这些问题都是非常关键的。

参考文献

- Antonini M, Barlaud M, Mathieu P, et al. Image Coding Using Wavelet Transform[J]. IEEE Trans. on Image Processing, 1992, 1(2): 205-220.
- 李建平. 小波分析与信号处理——理论、应用及软件实现[M]. 重庆: 重庆出版社, 1997.
- Christopoulos C, Skodras A, Ebrahimi T. The JPEG2000 Still Image Coding System: An Overview[J]. IEEE Trans. on Consumer Electronics, 2000, 46(4): 1103-1127.
- 王祥林. 基于小波变换的静态图像压缩编码算法研究[D]. 北京: 清华大学, 1997.
- Shapiro J. Embedded Image Coding Using Zerotrees of Wavelet Coefficients[J]. IEEE Transactions on Singal Processing, 1993, 41(12): 3445-3462.