

# 一种新型的 VBR 视频传输方法

张源海<sup>1,2</sup>, 钟琳华<sup>1,2</sup>, 皇甫伟<sup>1</sup>, 许长桥<sup>1,2</sup>, 李凯慧<sup>1,2</sup>

(1. 中国科学院软件研究所, 北京 100080; 2. 中国科学院研究生院, 北京 100039)

**摘 要:** 保证变比特率(VBR)视频流在 IP 网络上的平滑传输一直是多媒体通信领域研究的热点. 传统的 PCR 协助的恒定速率传输方法利用了嵌入到视频码流中的程序参考时钟(PCR)来定期校正发送速率, 是以较高的传输速率为代价减少了播放低码率视频流时的缓存需求. 为了同时适应高码率视频流的鲁棒传输, 本文提出了一种改进的方法, 通过对发送速率进行更细粒度的调整和服务器端的自适应缓冲机制, 进一步平滑了传输速率, 而且没有增加终端缓存的需求. 仿真实验的测试结果表明, 改进的方法能更好地服务于各种码率的流媒体传输与回放, 适用范围更加广泛.

**关键词:** MPEG2; 传输流; 变比特率; 程序参考时钟

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2007) 05-0895-05

## A Novel VBR Video Transmission Method

ZHANG Yuan-hai<sup>1,2</sup>, ZHONG Lin-hua<sup>1,2</sup>, HUANGFU Wei<sup>1</sup>, XU Chang-qiao<sup>1,2</sup>, LI Kai-hui<sup>1,2</sup>

(1. Institute of Software, Chinese Academy of Sciences, Beijing 100080, China; 2. Graduate School of CAS, Beijing 100039, China)

**Abstract:** How to guarantee the smooth transmission of video streams over IP network has been a hotspot at the area of multimedia communications. The traditional PCR-assist CBR schemes employ the timing information provided by the Program Clock Reference (PCR) embedded in the video streams to regulate their transmission, which reduce the buffer requirement at the cost of higher transmission rate when playing low-rate video streams. For the robust transmission of video streams with high rates, a novel transmission method is presented in this paper. It uses more detailed time scale and self-adapting buffer mechanism at the server side to smooth the rate variation further. Simulation results show that this method serves best for the transmission and playback by slowing down the transmission rate without higher buffer requirement.

**Key words:** MPEG-2; transport stream; variable bit rate; program clock reference

## 1 引言

随着多媒体技术的发展, 视频图像的质量也越来越高. 通常将垂直分辨率达到 720 线以上的视频称为高清视频节目 HDTV (High Definition Television), 否则称为标清视频节目 SDTV (Standard Definition Television). 目前数字电视广播采用的核心标准是 MPEG2, 因此在 IP 网络上进行 MPEG2 传输流 TS (Transport Stream) 的传输是实现三网融合以及 IPTV 的重要手段. 如何保证 MPEG2 TS 流, 特别是高清视频流, 在 IP 网络上的平滑传输在流媒体业务中占有重要的地位.

MPEG2 标准采用的压缩算法决定 MPEG2 的视频流是变比特率 VBR (Variable Bit Rate) 的. 研究和统计表明<sup>[1,2]</sup>, MPEG2 视频的图片组 GOP (Group of Picture) 内 I 帧的数据量较大, B 帧较小. 为了对 VBR 视频进行平滑传输, Wakamiya 等提出了针对实时压缩视频信号的 TCP-friendly 控制算法<sup>[3]</sup>, 是通过调整量化尺度来改变发送速率, 从而在保证视频质量的前提下平滑了视频数据的突发性, 即尽可能使速率-失真 Rate-Distortion (R-D) 达到最优. 但是, 对于存储类视频的传输, 如 VOD 应用, 视

频节目通常是预先压缩好的, 没有量化尺度可供控制, 所以不能利用类似的算法.

用传统的恒定比特率 CBR (Constant Bit Rate) 的方法传输 VBR 视频流需要占用播放终端很大的缓存空间. IBM Rochester 提出的<sup>[4]</sup> PCR 协助的 CBR 方法 PCBR (PCR-assist CBR) 利用 MPEG2 编码时嵌入的程序参考时钟 PCR (Program Clock Reference) 值来限制数据包的发送时刻, 减小了终端缓存的需求, 但它以增加传输速率为代价. Du 等<sup>[4]</sup>提出了对 PCBR 方法的改进, 即双重 PCR 协助的 CBR 方法 PDCBR (PCR-assist Dual-Rate CBR), 这种方法对于 MPEG2 标清视频节目的传输效果比较理想, 但传输码率变化范围更大的 MPEG2 高清视频就会造成终端缓冲区的频繁溢出, 而且溢出一般出现在变化剧烈的场景, 这对视频质量的影响更加严重. 本文在 PDCBR 算法的基础上, 针对存储视频的传输给出了一个改进的方法 RPDCBR (Revised PCR-assist Dual-Rate CBR), 通过对相邻 PCR 之间的数据进行更细粒度的速率调整, 使调整后的速率能更及时地反映视频码率的变化, 同时在服务器端增加自适应缓冲区, 通过对缓冲区的控制进一步平滑速率的变化. RPDCBR 方法在不增加

收稿日期: 2006-11-20; 修回日期: 2007-03-02

基金项目: 国家自然科学基金 (No. 60434030, No. 60673178); 国家 973 重点基金研究发展计划 (No. 2007CB307106)

传输速率的前提下,减小了终端缓存的需求,同时提高了视频服务器的处理性能,降低了数据丢弃对视频质量的影响.经过仿真实验证明,这种方法有效地平滑了 MPEG2 TS 码流的变化,对 MPEG2 标清和高清节目都有很好的效果,使视频传输更加鲁棒.

## 2 传统的 PCR 协助的传输方法及其缺陷

文献[5]研究了开始延迟时间、传送速率、终端缓存需求大小的关系,证明对于 MPEG2 TS 流这样的 VBR 编码的视频,用恒定比特率 CBR 方式传输需要很大的缓存和较长的开始延迟时间.为了改善上述问题,IBM Rochester 提出了 PCBR 方法<sup>[4]</sup>,该方法利用编码时嵌入到 MPEG2 TS 流中的 PCR 值来定期校正传送速率. PCR 值用来保证包含相应 PCR 的数据包在正确的时间发送.假设第  $i$  个 PCR 的值是  $|PCR_i|$ ,  $C_i$  是第  $i$  个 PCR 出现的时刻并且  $C_i = |PCR_i|$ .在传输过程中,对于任意  $i > 1$ ,如果  $|PCR_i| > C_i$ ,即传输时间早于确定的时间,说明传输速度过快,需要停止发送数据直到  $C_i = |PCR_i|$ ,目的是减少终端所需的缓存空间;如果  $|PCR_i| < C_i$ ,即传输时间晚于确定的时间,说明传输速度慢了,这时不改变传输的速度.

图 1 描述了 PCBR 方法.下面的曲线是终端缓存曲线(Consumption Curve),表示被解码视频数据的累积值.上面的曲线是传输曲线(Delivery Curve),表示服务器发送数据的总和.从图中可以看出,对于帧率为 30 帧/秒的视频码流,PCBR 方法实际上是每隔 33.33ms 检查一下当前时间状态去决定数据的实际发送情况.

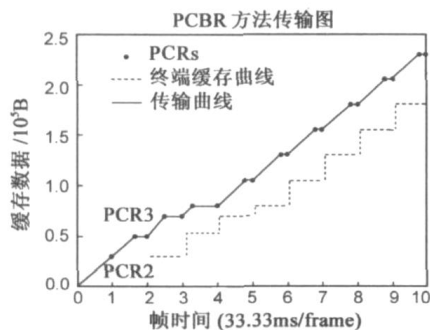


图 1 PCBR 方法的传输曲线和消费曲线

与 CBR 方法相比,PCBR 减少了终端缓存,但由于传输速率是  $R_{PCBR}$  或 0,两者之间过大的差异和频繁的切换会造成图 1 所示的数据脉冲现象,导致其平均传输速率比 CBR 方式有所增加. PCBR 方法的另一个缺点是:如果两个曲线之间没有足够的缓冲空间,而恰好  $|PCR_i| > C_i$ ,即发送进程处于等待状态,这时传输曲线可能会影响到消费曲线,产生“饿死”的现象.为了减小传送速率以及避免“饿死”现象,文献[4]提出了一种更加灵活的 PDCBR 方法.在 PDCBR 方法中,传输速率在两个值  $R_{high}$  和  $R_{low}$  之间切换.采用哪种传输速率取决于

当前的时间早于还是晚于确定的时间.对于任意  $i > 1$ ,如果  $|PCR_i| > C_i$ ,即传输时间早于确定的时间,PDCBR 方法把速率切换到  $R_{low}$  以减少终端缓冲数据的累积,而不像  $P_{CBR}$  中那样停止发送数据;如果  $|PCR_i| < C_i$ ,即传输时间晚于确定的时间,速率切换到  $R_{high}$  来避免终端解码的“饿死”现象.实际上,PCBR 是 PDCBR 的一个特殊情况.对于 PCBR 而言, $R_{high}$  是  $R_{PCBR}$ ,  $R_{low}$  是 0.

图 2 描述了 PDCBR 方法.其中开始延迟时间设定为与 PCBR 方法中相同, $R_{high}$  比  $R_{CBR}$  稍高,而低于  $R_{PCBR}$ ,  $R_{low}$  比  $R_{CBR}$  稍低.从图 1 和图 2 的对比可以看出,PDCBR 方法克服了 PCBR 方法中的数据脉冲现象,一定程度上可以避免“饿死”现象的出现.实验证明<sup>[4]</sup>,PDCBR 方法的传输速率小于 PCBR,但其对终端缓存的需求要大于 PCBR.

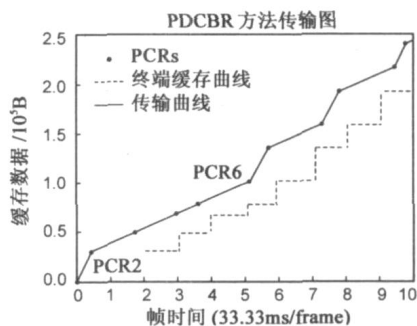


图 2 PDCBR 方法的传输曲线和消费曲线

下面我们对传统 PCR 协助传输方法的缺陷进行理论分析.文献[4]中对于 PCBR 和 PDCBR 的实验是基于码率变化较小的 MPEG2 标清视频节目,对于应用日益广泛的 MPEG2 高清视频则存在着潜在的问题.无论采用 PCBR 还是 PDCBR,其速率的调整都只是在两个值 ( $R_{PCBR}$ 、0 或  $R_{high}$ 、 $R_{low}$ ) 之间进行,由于高清视频的码率范围变化更大,就会造成发送速率与实际码率的差异比标清视频的情况大,从而容易引起终端缓冲区的溢出或“饿死”现象,导致视频质量的下降.

以 PCBR 方法为例,假设在第  $i$  个帧的时间段内,记  $S_i$  为在第  $i$  个帧时间段初所应该传送的数据, $S_i$  是第  $i$  帧的大小, $\bar{S}_i$  为在第  $i$  个帧时间段内已经传送的数据, $F$  为帧的持续时间, $R$  为在第  $i$  个帧时间段内的实际码率.设由 PCBR 方法算出的传送速率为  $R_{PCBR}$ ,则  $\bar{S}_i$  和  $S_i$  可由以下迭代方程式求得:

$$\bar{S}_i = \min(R_{PCBR} \times F, S_i + S_i) \quad (1)$$

$$S_{i+1} = \max(S_i + S_i - \bar{S}_i, 0) \quad (2)$$

$$S_1 = 0 \quad (3)$$

终端最小缓存需求可以通过计算传输曲线和消费曲线的最大差值得出.对一个给定的开始延时  $P$ ,可以得到:

$$f_{PDCBR}(P) = \max_{1 \leq n \leq N} \left[ \sum_{i=1}^{n+P} \bar{S}_i - \sum_{i=1}^{n-1} S_i \right] \quad (4)$$

由等式(1)可以看出,如果  $R_{PCBR}$  与  $R$  相差越多,计算出的  $\bar{S}_i$  与实际的误差就越大,从而引起等式(4)中缓存的误差增大,导致到达终端实际的数据比缓存空间过大而溢出或过小而产生“饿死”现象. 解决问题的一个简单办法是把传输速率划分为更多的等级,根据码率的变化决定采用哪一等级的数据,但速率等级的增多会带来速率选择算法计算复杂度的提高,同时速率等级的设定也将更加复杂,随着并发用户数的增加,会给服务器带来巨大的负担.

### 3 RPDCBR 算法

#### 3.1 整体框架

为此,在 PDCBR 方法的基础上,我们提出了改进的双重 PCR 协助算法 RPDCBR. 图 3 给出了 RPDCBR 算法的整体框架.

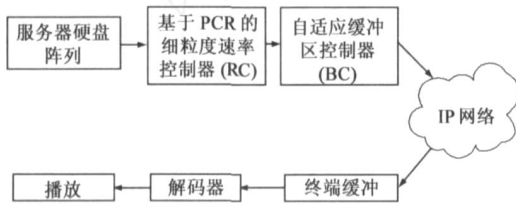


图 3 RPDCBR 算法整体框架

速率控制器 RC(Rate Controller) 根据 PCR 值进行更细粒度的速率计算,从而平滑码率增大带来的影响并决定传输速率  $R_{high}$  或  $R_{low}$ ,同时缓冲区控制器 BC(Buffer Controller) 监测服务器缓冲区的使用情况,对于码率过大导致缓冲区使用超过警戒值的时刻,根据数据包所包含的 PCR 值和帧类型进行有选择的抛弃,这样不仅充分利用了服务器的缓冲区平滑突变的码率,进一步减小终端的缓存需求,而且使终端的视频质量损失达到最小. 实验证明,本文的方法对于标清和高清视频节目的播放都有很好的效果.

#### 3.2 基于 PCR 的细粒度速率控制器

为了进行更细粒度的速率控制,我们考虑了一个帧时间段内包含多个 PCR 的情形. 假设在第  $i$  个帧时间段  $(F_{i-1}, F_i)$  内有  $n_i$  个 PCR. 例如,在图 2 的第 6 个帧时间段内有 2 个 PCR. 这  $n_i$  个 PCR 把第  $i$  个帧时间段分成  $n_i + 1$  段,用  $T = \{T_1, \dots, T_{n_i+1}\}$  表示,每个时间段内对应的传输速率为  $R = \{Rate_1, \dots, Rate_{n_i+1}\}$ . 第  $i$  个帧时间段内已经传输的数据  $\bar{S}_i$  以及第  $i + 1$  个帧时间段初需要传输的数据  $S_{i+1}$  可以用下面的公式来计算得到:

$$\bar{S}_i = \sum_{k=1}^{n_i+1} T_k \times Rate_k \quad (5)$$

$$S_{i+1} = \max(S_{last} - T_{n_i+1} \times Rate_{n_i+1}, 0) \quad (6)$$

在式(6)中,  $S_{last}$  是第  $i$  个帧时间段内最后一个 PCR (出现在  $F_i, T_{n_i+1}$  时刻) 和下一个 PCR 之间的数据,显然

$S_{i+1}$  等于以  $Rate_{n_i+1}$  传输  $T_{n_i+1}$  时间后  $S_{last}$  中剩余的数据.

根据 MPEG2 TS 流的编码规则,可以由相邻的 PCR 值计算出实时的传送码率<sup>[6,7]</sup>来决定发送速率. 设第  $i$  个 PCR 的值是  $|PCR_i|$ , 处于码流中第  $b(PCR_i)$  字节处,第  $i + 1$  个 PCR 的值是  $|PCR_{i+1}|$ , 处于码流中第  $b(PCR_{i+1})$  字节处,由此可以计算出对应 PCR 区间的传输速率  $R_{PCR}$ :

$$R_{PCR} = \frac{[b(PCR_{i+1}) - b(PCR_i)] \times STC}{|PCR_{i+1}| - |PCR_i|} \quad (7)$$

其中  $STC$  是系统时钟频率 27MHz, 这样的计算过程需要对相邻 PCR 在码流中所处的位置以及 PCR 本身进行记录,对于场景变化复杂、码率大的高清视频文件比较消耗系统资源.

为此,提出了一个计算第  $i$  个帧时间段内的集合  $T$  和  $R$  的算法. 假设  $U_i$  和  $V_i$  是  $(F_{i-1}, F_i)$  之间的两个值.  $S_{last}$  是检测到下一个 PCR 之前需要传输的数据量. 开始,  $S_{last}$  置为第一帧的大小  $S_b$ , 其中  $b = 1$ .  $Rate$  是当前速率,可能是  $R_{high}$  或  $R_{low}$ .  $Rate$  的初始值设置为  $R_{high}$ . 终端所需要最小缓冲可以由公式(4)得到.

表 1 细粒度的速率调整算法

$T = \phi; R = \phi;$
$U_i = F_{i-1}; V_i = F_{i-1};$
while $V_i < F_i$
$V_i = \min\left(\frac{S_{last}}{Rate} + U_i, F_i\right);$
$S_{last} = \max(S_{last} - (V_i - U_i) \times Rate, 0);$
把 $Rate$ 添加到集合 $R$ 中
把 $(U_i, V_i)$ 添加到集合 $T$ 中
if $S_{last} = 0$
if $\left\lfloor \frac{V_i}{PCR \text{ Interval}} \right\rfloor < b$
$Rate = R_{low};$
else
$Rate = R_{high};$
$b = b + 1;$
$S_{last} = S_b;$
if $Rate = 0$
$Rate = R_{high};$
$V_i = F_i;$
$U_i = V_i;$

通过表 1 中的算法,我们可以把速率的调整精细到帧内每个 PCR 之间的时间段内,从而使发送速率的调整能更加及时地应对码率的变化.

#### 3.3 自适应缓冲区控制器

进行了精细的速率调整后,如果直接发送视频数据还有可能对播放终端缓存有一定影响,这就需要同时对视频码流进行缓冲处理后再传输,从而进一步平滑突

变的码率. 由于磁盘访问速度和 IP 网络丢包的突发性和随机性, 为降低 CPU 和内存资源占用率, 设置一个公用的循环队列用于存储从 RC 中读出的视频数据, 启动一个读线程来完成数据的读取功能, 启动一个发送线程来完成数据的发送功能. 这些线程以怎样的策略与算法操作这个数据缓冲区, 在很大程度上决定了系统的性能和终端缓存状态及播放视频的质量<sup>[8]</sup>.

数据缓冲策略的基础实际上是一个特定环境下的生产者-消费者问题. 传统的有限缓冲条件下的生产者-消费者问题的基本要求是数据传输的正确性, 所以只有当消费者线程消费掉某个“产品”时, 才推进出队列指针, 而只有队列里面有空槽时, 才允许生产者将其“产品”置入其中. 常用的解决方法是引入互斥变量和整型信号量. 如图 4 所示.

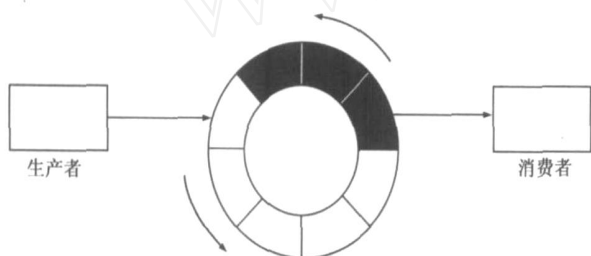


图 4 传统的缓冲控制模型

在实时视频传输的特定环境下, 传统的方法因为生产者和消费者之间的耦合度太大而降低了系统的灵活性. 尤其当码率增大时, 如果服务器不进行缓冲, 就会造成客户端缓存需求的增加; 如果用传统的方法缓冲, 因为码率增大带来生产者生产速度 ( $R_{high}$ ) 加快或网络状况不好带来消费者消费速度减慢, 必然会产生缓冲区充满生产者等待的情况, 从而引起客户端播放出现停顿.

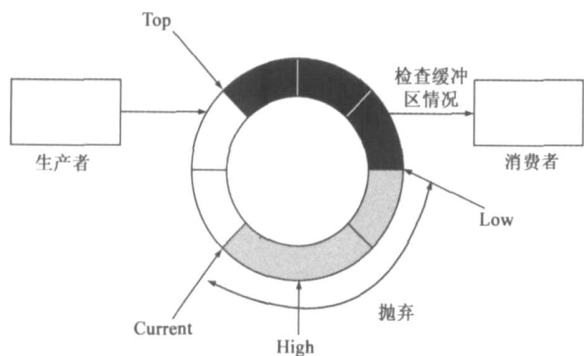


图 5 BC 中的缓冲控制模型

由上面的分析可以看出, 缓冲队列中堆积的数据量大小不仅取决于 RC 选择的传输速率, 而且能间接地、动态地反映网络状况的变化, 即当网络带宽不充足时, 队列相对较满. 因此, 如图 5 所示, 在 BC 中, 利用获得数据包的 PCR 信息, 同时结合人眼视觉系统 HVS (Human Vi-

sual System) 的特性, 我们提出了一种改进的缓冲策略: 消费者消费“产品”之前首先要检查循环队列的使用情况, 如果满足丢弃条件, 那么根据数据包中包含的 PCR 信息, 抛弃一部分循环队列中陈旧的音视频数据. 抛弃条件的选择是 BC 的关键, 其中一个简单策略是: 一旦发现队列元素超过某一警戒值, 就丢弃某一阈值的数据, 将队列元素降低到合理水平. 设队列为容量为  $Top$ , 队列中当前的数据量为  $Current$ , 该警戒值为  $High$ , 阈值为  $Pace$ , 降低到的合理值为  $Low$ . 我们给出以下分析: (1) 显然,  $Top > High$ , 这个约束条件可以保证生产者与消费者之间不发生读写互斥问题, 即在队列容量达到最大值之前, 丢弃就产生, 从而将队列容量降低到  $Low$  的位置; (2) 当  $Current > High$  时, 才发生丢弃, 并且  $Low = Current - Pace$ . 其中阈值  $Pace$  的设置, 是一个多方面因素综合考虑的结果. 如果设置太小, 会造成丢弃处理事件频繁发生, 导致系统性能的降低; 如果设置太大, 则可能会丢弃没必要丢弃的数据, 最终导致终端音视频质量的大幅度下降; (3) 为了使抛弃对人眼视觉的影响降到最低, 需要判断抛弃队列中数据帧类型, 按着先抛弃不作为参考帧的数据进行, 即首先抛弃  $B$  帧数据, 然后是离  $I$  帧较远的  $P$  帧, 再次是离  $I$  帧较近的  $P$  帧, 最后才是  $I$  帧数据. 这时  $Pace$  的取值根据一帧或多帧图像的长度动态的进行设置. 这种自适应的丢弃策略可以有效的减少马赛克的数量, 提高系统传输的性能.

这里给出的数据缓冲机制建立在一种新型的生产者-消费者模型的基础之上. 它最大限度的利用了和满足了多媒体网络传输的需求特点, 不仅进一步平滑了 RC 输出的码率, 同时也考虑到了网络状况的变化, 将系统对资源的要求降低到了较低的水平, 具有很好的系统稳定性和网络适应性.

#### 4 仿真实验

为了评价以上机制的性能, 本文使用仿真软件 NS-2 对 RPDCBR 算法进行了仿真, 同时与 CBR, PCBR, PD-CBR 这三个算法进行了比较. 图 6 是试验的网络拓扑, 其中节点 0 和 1 是 MPEG-2 TS 流的源, 节点 2 和 3 是中间网络设备, 节点 4 和 5 是客户端, 整个链路的带宽为

表 2 视频数据源参数

数据源名称	湖南卫视数字卫星节目*	Under Water*
编码类型	MPEG-2 TS	MPEG-2 TS
视频文件长度	6.392min	4.266min
分辨率	720 x576	1920 x1080
平均码率	5.8Mbps	19.6Mbps
档次	标清	高清
帧模式	IBB (PBB) <sup>4</sup>	IBB (PBB) <sup>4</sup>

\* MPEG-2 pre-recorded transport stream data traces. ftp://159.226.5.89/pub/Mpeg2TSSStream/

100Mbps, 实验中分别采用码率较小的标清视频文件和码率较大的高清视频文件进行验证, 数据源具体参数如表 2 所示。



图 6 网络拓扑

在具体的仿真实验中, 节点 0 作为视频服务器提供数据源, 采用本文中讨论的几种方法发送 MPEG2 TS 流, 节点 4 作为客户端接收数据, 由于在 NS-2 下无法播放视频流, 所以采用根据解码要求实时地丢弃接收到的数据来模拟正常播放。图 7 和图 8 分别表示几种传输方法对于标清和高清视频的实验结果, 图 7(a) 和图 8(a) 是终端缓存需求曲线, 图 7(b) 和图 8(b) 是传输速率曲线。从图中可以看出, 对于标清视频, PCBR、PDCBR 和 RPDCBR 三种方法的缓存需求相似, 都小于传统的 CBR 方法, RPDCBR 和 PDCBR 方法由于避免了在每个 PCR 值时的脉冲现象, 其传送速率比 PCBR 方法有所降低; 对于高清视频, PDCBR 算法对缓存的需求明显增加, 同时发送速率也提高了很多, PCBR 算法的现象则更为严重, 而改进的 RPDCBR 方法由于采用了更细粒度的速率调整策略和合理的服务器缓冲区, 在不增加终端缓存需求的情况下, 降低了传送速率, 使得在相同条件下, RPDCBR 算法能传送更多的信息, 保证了码率更大的数据的平稳传送。

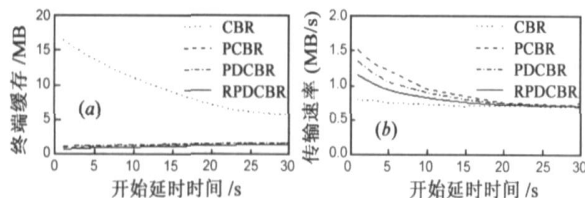


图 7 标清视频输入时, CBR、PCBR、PDCBR 和 RPDCBR 方法的比较。(a) 终端缓存; (b) 传输速率

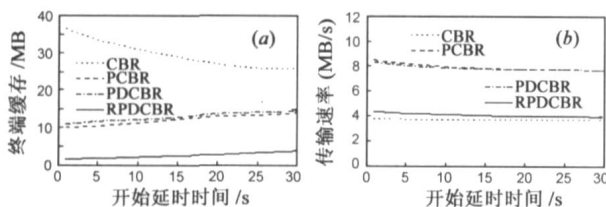


图 8 高清视频输入时, CBR、PCBR、PDCBR 和 RPDCBR 方法的比较。(a) 终端缓存; (b) 传输速率

## 5 结论

本文研究了用 PCR 协助 MPEG2 TS 流传输的几种方法, 并在 PDCBR 算法的基础上加以改进。对各种方法的实验结果表明, PCBR 和 PDCBR 方法在播放标清节目时较传统的 CBR 方法降低了缓存需求, 但对于高清视频的播放, 上述两种方法则需要更大的传送速率和更多的缓存需求。本文改进的方法对于标清和高清视频

都有很好的效果, 尤其在高清视频播放时, 相对于 PCBR 和 PDCBR 方法降低了速率, 节省了带宽, 同时没有很大的缓存需求, 适用范围更加广泛。

## 参考文献:

- [1] Zhang Z L, Kurose J, Salehi J D, et al. Smoothing, statistical multiplexing, and call admission control for stored video[J]. IEEE JSAC, 1997, 15(6): 1148 - 1166.
- [2] O ROSE Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems[A]. Proceedings of the 20th Annual Conference on Local Computer Networks[C]. Minneapolis, MN, 1995. 397 - 406.
- [3] Naoki Wakamiya, Masayuki Murata, Hideo Miyahara. On TCP-friendly video transfer with consideration on application-level QoS[A]. Proceedings of IEEE ICME 2000[C]. New York, NY, USA, 2000. 843 - 846.
- [4] Du David H C, Hsieh Jenwei, Lee Horng Juung, Taisheng Chang. PCR-Assist CBR for delivering pre-recorded MPEG-2 transport streams[A]. Proceedings of IEEE International Conference on Multimedia Computing and System[C]. Ottawa, Canada, 1997. 646 - 647.
- [5] J M McManus, K W Ross. Video on-demand over ATM: Constant rate transmission and transport[J]. IEEE JSAC, 1996, 14(6): 1087 - 1098.
- [6] Hong Fei, Wu Zhi Mei. Revised PCBR transmission of pre-recorded MPEG-2 transport stream[A]. Proceedings of IEEE SOFTCOM 2001[C]. Split, Croatia, 2001. 545 - 551.
- [7] 洪飞, 吴志美. MPEG-2 传送流传输方法的研究与改进[J]. 计算机学报, 2004, 27(3): 352 - 356.  
Hong Fei, Wu Zhi Mei. Research and revision on transmission of MPEG-2 transport stream[J]. Chinese Journal of Computers, 2004, 27(3): 352 - 356. (in Chinese)
- [8] 褚蓬飞, 张焕强, 方贵明. 视频会议中数据缓冲机制的研究[J]. 计算机应用研究, 2006, 23(5): 67 - 69.  
Chu Peng Fei, Zhang Huan Qiang, Fang Gui Ming. Research on the mechanism of data buffer management in video conference system[J]. Application Research of Computers, 2006, 23(5): 67 - 69. (in Chinese)

## 作者简介:



张源海 男, 1979 年 11 月生于吉林长春, 博士研究生, 主要研究方向是视频传输、视频编码和流媒体技术。  
E-mail: yuanhai02@ios.cn

钟琳华 男, 1979 年生于广东五华, 博士研究生, 主要研究方向是社区宽带综合业务网络中的 QoS 保障技术。