



无线视频通信中基于队列预测的自适应缓存管理算法^{*}

钱 超¹, 朱晓荣^{1,2}

(1. 南京邮电大学江苏省无线通信重点实验室 南京 210003;

2. 东南大学移动通信国家重点实验室 南京 210096)

摘 要: 无线网络中实时视频业务对网络环境的变化非常敏感, 因此通常需要在接收端采用合理的缓存管理策略来缓解网络波动对用户观看视频造成的影响。提出一种基于队列预测的自适应缓存播放管理机制, 该方法通过判断当前缓存队列的状态, 并根据实时视频到达率和端到端时延对缓存长度以及播放速率进行综合调整。实验结果表明, 所提算法可以随着网络环境的波动自适应地调整缓存大小和播放速率, 有效降低视频业务的中断频率和丢帧率。

关键词: 自适应缓存管理; 队列预测; 缓存状态; 时延控制

doi: 10.3969/j.issn.1000-0801.2014.07.011

An Adaptive Buffer Management Algorithm Based on Queue Prediction for Wireless Video Communication

Qian Chao¹, Zhu Xiaorong^{1,2}

(1. Wireless Communication Key Lab of Jiangsu Province,

Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China)

Abstract: The real-time video services in wireless networks are fragile to the dynamic network environment. Therefore, proper buffering management techniques are adopted to mitigate the impact of network fluctuation on user experiences. An adaptive buffer management algorithm based on queue prediction was proposed for wireless video streaming applications. By judging the current state of the buffer, the scheme adjusted the video playout rate and buffer size according to the real-time frame arrival rate and end-to-end delay. The experiments were conducted to validate the proposed algorithm. The experimental results show that this method is able to adaptively adjust buffer size and playout rate according to network fluctuations, which reduces the interrupt frequency and the number of lost frame effectively.

Key words: adaptive buffer management, queue prediction, buffer status, delay control

1 引言

近几年来, 由于智能手机和平板电脑等智能终端设备的快速发展, 移动视频业务已经呈指数增长并且开始占据

移动网络的大部分。然而由于用户的移动性和信道的复杂性以及多变性, 对实时的无线流媒体视频传输造成了很大的影响, 其中视频质量对分组丢失和时延抖动最为敏感^[1]。实际中, 移动视频应用通常采用缓冲技术来处理无线信道

^{*} 国家自然科学基金资助项目(No.61372125), 国家重点基础研究发展计划(“973”计划)基金资助项目(No.2013CB329104), 江苏省自然科学基金资助项目(No.BK2012831), 东南大学移动通信国家重点实验室开放研究基金资助项目(No.2013D01)

动态变化带来的各种问题。采用合适的缓存机制可以及时存储到达的分组,使视频可以平滑流畅地播放,有效地改善系统的性能。由于无线信道的时变性和随机性,传统固定长度的缓存机制并不适用于无线网络的视频传输,否则会因为数据的突发性造成缓存队列的过度拥塞或者过度空闲,造成严重的分组丢失或者视频播放中断^[2]。所以现在的实时视频系统采用的都是可变长度的缓存机制,即自适应的多媒体播放技术。

参考文献[1]提出一种新的自适应媒体播放系统,该系统充分地利用了分组时延预测算法,并采用了差异自动退避时延预测模型来做出估计,但算法比较复杂。参考文献[3]提出基于对缓冲区忙闲时间的测量的反馈控制方法,在适应短期内的分组突发情况反面表现较好。参考文献[4]通过改变分组内时延(inter-packet delay, IPD)结合缓存状态来控制发送速率,可以在有噪声的信道中为用户提供可接受的视频质量。参考文献[5]提出一种新的基于探索的自适应播放缓存方案。利用探索方案,收集即时网络状况,然后使用时延余量和时延抖动余量来调整播放缓存。参考文献[6]设计了一种基于下溢可能性估计的自适应多媒体播放方法,联合采用队列长度和信道状态参数来进行播放速率的控制,但是信道状态由于动态变化性一般很难准确得到。参考文献[7]提出了根据实际的时延变化来调整缓存的大小,并且采用 3 次样条平滑来重新设置缓存的大小,可以减小信号的噪声,但是该算法实现比较困难。本文提出一种基于队列预测的缓存管理算法,通过联合调整缓存区的大小和视频播放速率来为用户提供稳定的视频流,通过采用简单的调整方案,缓冲队列可以较好地适应当前的网络环境波动,而不用复杂的数学预测,简单易行,并且对端到端的时延也进行了一定的控制,较好地保证了视频业务的实时性。

2 系统架构

一般视频通信系统接收端的基本架构如图 1 所示。由于无线网络的时变性,分组到达接收端的速率和其在发送端的发送速率会有较大的不同,所以当缓存将要出现上溢或者下溢时,可以根据当前的分组到达速率、视频播放速

率以及当前缓存占用率预测出下一时刻的缓存变化趋势,并及时做出调整,将缓存长度和播放速率调整为合适的大小。设上一时刻的缓存大小为 $q(t-1)$,则当前时刻缓存大小应设置为:

$$q(t)=q(t-1)+\Delta q \quad (1)$$

其中 Δq 为缓存的改变量,该值应由当前的分组到达速率、视频播放速率、当前缓存占用率以及时延参数共同决定。下面详细分析各参数对缓存大小调整的影响。

2.1 缓存状态

采用合适长度的缓存队列可以及时地存储到达的分组,不会造成数据分组的丢失,再加上对播放速率的控制,可以为用户提供更平滑的视频播放。缓存的长度不能设置得太大,否则会引起较大的时延和较低的队列利用率,也不能设置得太小,否则很容易造成到达的分组由于队列已满而被丢弃。所以提出的缓存管理方法要平衡消除时延抖动和降低端到端时延,为视频应用设置最合适的缓存队列长度。

为了更好地进行缓存管理,笔者根据当前移动设备的缓存队列占用情况将缓存分为 3 种明确的状态:underflow、normal、overflow。

- underflow,即下溢状态,也就是当前的缓存队列中分组数过少,不足以进行视频帧的播放,在这种状态下如果播放速率大于视频帧的到达速率,则会造成视频播放的中断。中断播放对视频业务的影响非常严重,有研究表明,如果一段视频的中断时间(或者称为缓冲时间)超过 2 s,就会有人开始放弃观看,之后缓冲的时间每增加 1 s,就会多 6% 的人放弃观看,并且随着中断播放的次数的增加,用户放弃观看的概率呈指数上升^[8]。所以说对实时视频来说,要尽量避免出现缓存发生下溢的情况。
- normal,在这种状态下,缓存队列中暂存的分组数为队列长度的正常比例数,视频帧的到达速率和播放速率基本维持在一个平衡的状态,视频播放流畅。
- overflow,即上溢状态,在这种状态下,缓存队列基本上被占满,这是由于大量的突发数据到达或者是播放速率小于帧平均到达速率,缓存队列很容易达到占满状态,致使新到达的数据由于来不及存储而被丢弃,造成队列的拥塞和数据的重传^[4]。所以视频播放的时候,一旦出现 overflow 或者 underflow 状态,可以通过调整缓存队列的大小和视频的播放速度来使缓存区尽量控制在 normal 状态。

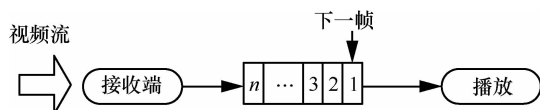


图 1 视频通信系统接收端架构



2.2 时延分析

无线网络中的流媒体视频通信对实时性的要求比较高,特别是对于视频会议、实况转播等一些实时视频应用来说,一般端到端的时延不能超过 250 ms,否则会引起较严重的用户体验下降,而用户可接受的时延应该控制在 200~500 ms^[5],所以一般的实时视频应用大多采用无需建立连接的 UDP 作为传输协议,可以大大减少端到端的时延,弊端是容易产生大量分组丢失,尤其是在无线网络中。因此在对视频质量要求比较高的业务中,且可以容忍一定的时延的情况下,往往采用可靠的 TCP。本文中采用的实验方法就是基于 TCP 的传输方法。

图 2 所示为一帧视频数据从流媒体服务器端采集、编码、发送,然后经过无线信道的传输,最后在接收端被缓存、解码、播放整个过程中所产生的时延分布情况,可以看出视频帧的端到端时延 D 可用式(2)表示(其中未讨论视频帧的分片和恢复时延):

$$D = D_{\text{encode}} + D_{\text{send}} + D_{\text{channel}} + D_{\text{recv}} + D_{\text{buffer}} + D_{\text{decode}} + D_{\text{playback}} \quad (2)$$

式(2)中 D_{encode} 为系统编解码产生的时延,一般为固定值,随着移动设备硬件的提升,该时延相对于视频在信道中传输时所产生的时延 D_{channel} 可以忽略不计。 D_{send} 为服务器端的发送时延,包括分组的调度时延、排队发送时延等,对于用户端来说,该时延是不可控制的。 D_{recv} 为视频帧到达接收端并进入缓存区所产生的时延, D_{buffer} 为视频帧在缓存队列中等待的时间, D_{playback} 为视频帧从离开缓存区到提交到应用层进行播放之间的时延,与 D_{buffer} 相比, D_{recv} 和 D_{playback} 可以忽略不计^[9]。所以式(2)可以化简为:

$$D = D_{\text{send}} + D_{\text{channel}} + D_{\text{buffer}} + C \quad (3)$$

式中 C 为常数, D_{send} 由服务器端产生, D_{channel} 这里不做讨论,由于无线信道的时变性,也是动态变化且不可预测的,所以要减小视频帧的端到端时延,应在用户侧尽量减小分组数据在缓存区排队的时间,这就需要通过合理的缓存队列管理来实现,并且时延也应作为缓存队列调整的一项重要参数。

3 基于队列预测的缓存管理机制

基于上述分析,本文在考虑各项参数的影响下提出了一种基于队列预测的缓存管理机制。该方法对视频播放速率以及缓存长度进行联合调整来保证缓存队列尽量控制在 normal 状态,从而可以有效降低上溢和下溢概率。下面详细描述本文算法。

假定当前视频流到达接收端的速率为 R_a ,当前缓存队列总长度为 q ,视频播放速率为 R_p 。本算法中由于缓存队列的长度是时变的,所以缓存状态的设定不能采用固定的门限值,而是采用与缓存队列长度相比的相对值。本文中采用 B_l 、 B_h 来表示相对的门限值, B_l 、 B_h 分别表示下溢和上溢门限值,其中, $B_l = \alpha \times q$, $B_h = \beta \times q$, α 、 β 为系数因子,考虑到视频帧的突发性,缓存管理要来得及时对其做出调整,推荐设置为 $\alpha = 0.2$, $\beta = 0.8$ 。针对缓存的 3 种可能状态,设定 3 种实时播放速率, $R_p = \{R_l, R_n, R_h\}$,分别对应于较低、正常、较高速率,且有 $R_l < R_n < R_h$, R_n 是默认的 normal 状态采用的播放速率,当缓存中存储的帧数达到上溢门限时,说明播放速率已经来不及及时地播放缓存的视频,所以应该采用较大的播放速率 R_h 来加快视频帧的播放以及降低堆积的缓存帧的数量,不至于造成丢帧。当缓存帧的数目下降到下溢门限时,则说明视频帧的到达速率小于播放速率,应降低播放速率来防止缓存队列为空导致播放中断的情况,使缓存有足够的时间来缓存足够的帧来保持视频的流畅播放, R_l 对应较低的播放速率。

本文算法的具体思路如下:接收端每接收到一帧数据,首先根据当前的缓存长度计算对应的上溢和下溢门限 B_l 和 B_h ,然后判断缓存队列的状态,假设当前缓存占用长度为 Q ,若 $Q \leq B_l$,缓存队列状态为 underflow;若 $Q \geq B_h$,状态为 overflow;若 $B_l < Q < B_h$,则缓存队列状态为 normal。如果当前缓存队列状态为 normal 状态,说明视频播放正常,将当前帧存入至缓存队列中即可。如缓存队列状态为 underflow,则表示缓存即将发生下溢,则需要将播放速率减小,同时减小缓存长度以保持较高的缓存利用率,缓存

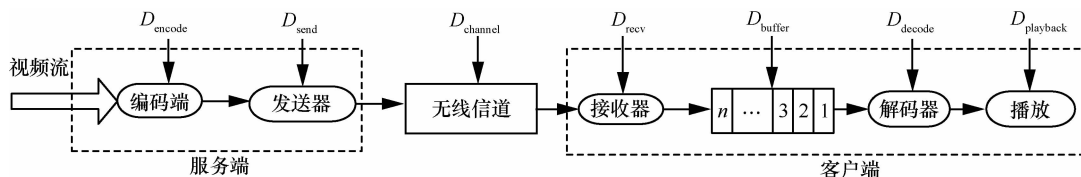


图 2 端到端的时延分析

的大小设置如下:

$$q(t) = q(t-1) - \lceil (R_p(t-1) - R_a(t-1)) \times k \rceil \quad (4)$$

其中, $\lceil \cdot \rceil$ 为向上取整的符号, $q(t)$ 为当前所要设置的缓存大小, $q(t-1)$ 为上一时刻缓存的大小值, 缓存的大小调整是基于上一时刻的缓存队列长度进行相应的微调, k 为调整因子, k 值不宜取太大, k 取值较大时, 即缓存调整的步长较大时, 偶尔会产生不必要的过多调整, 缓存的利用率较低, 而 k 值较小可以预防由于网络的突发性导致的帧到达速率的急剧变化的情况, 通过较小的调整来模拟预测缓存的实时变化情况, 具有较好的稳定性。本文中 $k=0.2$ 。另外, 如果缓存状态达到 overflow 门限, 则表示视频到达速率大于播放速率, 导致缓存区即将被填满, 所以需要及时扩大缓存来避免因缓存已满而造成的丢帧现象, 同时要增大播放速率, 来及时播放缓存中堆积的视频帧, 以预留出空间来存放接下来到达的帧序列, 此时缓存大小设置为:

$$q(t) = q(t-1) + \lceil (R_a(t-1) - R_p(t-1)) \times k \rceil \quad (5)$$

缓存的大小设置并不是无限制地增大减小, 都存在一定的界限, 缓存太大, 造成端到端时延过大, 不宜取, 缓存太小, 很难对视频起到一定的缓冲作用, 也容易造成上溢和下溢, 所以缓存自适应的调整也存在一定的上限和下限, 即 q_h 和 q_l , 应该保持缓存在 q_h 和 q_l 之间波动。算法如下。

算法 1 缓存管理算法

令当前缓存队列长度为 $q(t)$, 队列已占用长度为 Q

for every frame received

$B_l = \alpha q(t)$, $B_h = \beta q(t)$ // 计算缓存状态门限值 B_l 、 B_h

if ($Q \leq B_l$) // 则缓存即将下溢

if ($q(t) > q_l$)

$$q(t) = q(t-1) - \lceil (R_p(t-1) - R_a(t-1)) \times k \rceil \quad //$$

缓存队列调整

end if

$R_p = R_l$ // 播放速率调整

else if ($B_l < Q < B_h$)

$R_p = R_n$

else if ($Q \geq B_h$)

if ($q(t) > q_h$)

$$q(t) = q(t-1) + \lceil (R_a(t-1) - R_p(t-1)) \times k \rceil$$

end if

$R_p = R_h$

end if

end for

在播放每一帧的同时会对其端到端的时延 D 进行判断, 若其高于一定的门限值时, 说明端到端的时延太大, 可能会影响到用户的实时感受, 所以此时应及时做出调整, 将播放速率提高并减小缓存长度, 从而将时延控制在一定范围之内。具体的算法如下。

算法 2 时延控制算法

for every frame playout

计算当前播放视频帧的端到端时延 D ,

if ($D \geq D_l$) // D_l 为时延门限

$$q(t) = q(t-1) - \lceil (R_p(t-1) - R_a(t-1)) \times k \rceil$$

$R_p = R_h$

end if

end for

该算法简单实用, 通过实时地调整缓冲区的长度以及视频的播放速率, 减轻由于无线网络的动态变化造成的数据突发情况所引起的缓存上溢和下溢现象, 从而使用户端视频可以流畅的播放, 在一定程度上提高了用户体验。

4 仿真与测试

本节对上文所述算法进行了仿真, 并在实际的无线视频通信系统中模拟并测试。实验结构如图 3 所示。实验参数设置如下: 视频采集端为智能手机, 接收端为笔记本电脑, 视频采集的帧率为 25 帧/s, 传输的视频帧分辨率为 480×640 像素。为方便统计, 本实验中传输和接收数据、播放视频都以一帧视频为最小单位。测试时由于发送端和接收端的时间同步存在一定的误差, 并且由于无线网络的不稳定性, 所以实验的数据准确度也存在一定的误差, 本文在进行大量实验的基础上进行数据统计, 以减小误差。本文采用播放中断频率和丢失的帧数目作为评价算法的主要指标。视频播放中断的频率定义如下:

$$f = \frac{O(L)}{L} \quad (6)$$

其中, $O(L)$ 表示播放中断发生的次数, L 为视频播放的时长 [6, 10] s。具体统计结果如图 4~图 7 所示。

图 4(a) 和图 4(b) 给出了播放速率一定时, 不同的静态缓存下接收端的视频中断频率和分组丢失率。考虑到时延限制, 采用的静态缓存大小分别为 32、64、96、128 帧, 用 q 表示, 图 4(c) 表示了其对应的平均时延。为了接收端视频的正常观看, 播放速率分别采用了 24、26、28、30、32、34

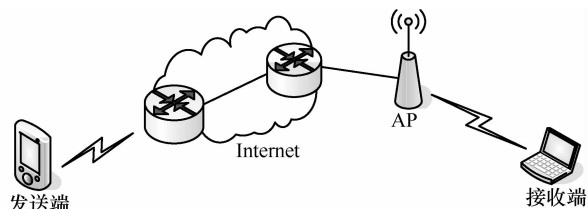


图3 实验结构

帧/s,用 r 表示。由图4可以看出,当采用较小的静态缓存(如 $q=32$ 帧或64帧)时,由于网络的波动和数据帧的突发性,播放中断频率较高,并且缓存队列也比较容易就被占满,导致之后的视频帧只能被丢弃,所以丢帧现象严重,且会产生大量的重传,导致进一步的网络拥塞。但在时延方面,则由于只缓存较少数量的帧,所以平均时延较低。相反,当采用较大的静态缓存($q=96$ 帧或128帧)时,由于缓存队列可以暂存更多的视频帧,缓存被占满的概率也比较低,所以其丢失的帧较少,并且由于可以缓存更多的视频帧,所以在帧到达率下降的时候不至于很快出现缓存队列为空的情况,所以视频中断频率较低,特别是在播放帧率较大的情况下,这种差别更加明显,但是采用较大的缓存时存在很大的时延,并且由于网络的波动,缓存在一定时间段内利用率较低。所以对于无线网络环境中的实时视频传输业务,不宜采用过大或者过小的静态缓存,同时还要考虑时延的限制,所以很有必要采用动态的缓存策略。

图5所示为采用本文算法与无控制下采用 $r=24$ 帧/s、 $r=30$ 帧/s两种常用的播放帧率的播放情况的对比。为了不影响视频观看效果,初始缓存大小为64帧,上溢门限比例为0.8,下溢门限比例为0.2,针对3种缓存状态的播放速率分别为24帧/s、28帧/s、32帧/s。图5(a)中所示为本文的自适应算法下缓存队列的大小与其实际占用情况的比较,由图可以看出本文算法可以很好地根据实时的视频帧到达速率和实际缓存占用率来动态调整队列长度和播放速率,本文算法的队列总长度可以随着缓存视频帧的增加和减少来自适应地调整,并且设置的是动态的门限,所以在缓存还有一定裕量,并且缓冲达到上溢和下溢门限时,就可以及时做出调整,这样可以尽量将缓存区的状态保持在normal状态,从而大大降低了缓存上溢和下溢的概率,即通过较小的调整来模拟预测缓存的实时变化情况,具有较好的稳定性。相比之下,图5(b)中不加控制的情况下,若网络波动,缓存很容易被占满,造成帧的丢失;而在网络阻塞的时候,缓存中的视频帧又很容易就播放完,导致视频中断。图5(c)中由于播放的速率较快,缓存区比较容易容易出现下溢,并且大部分时

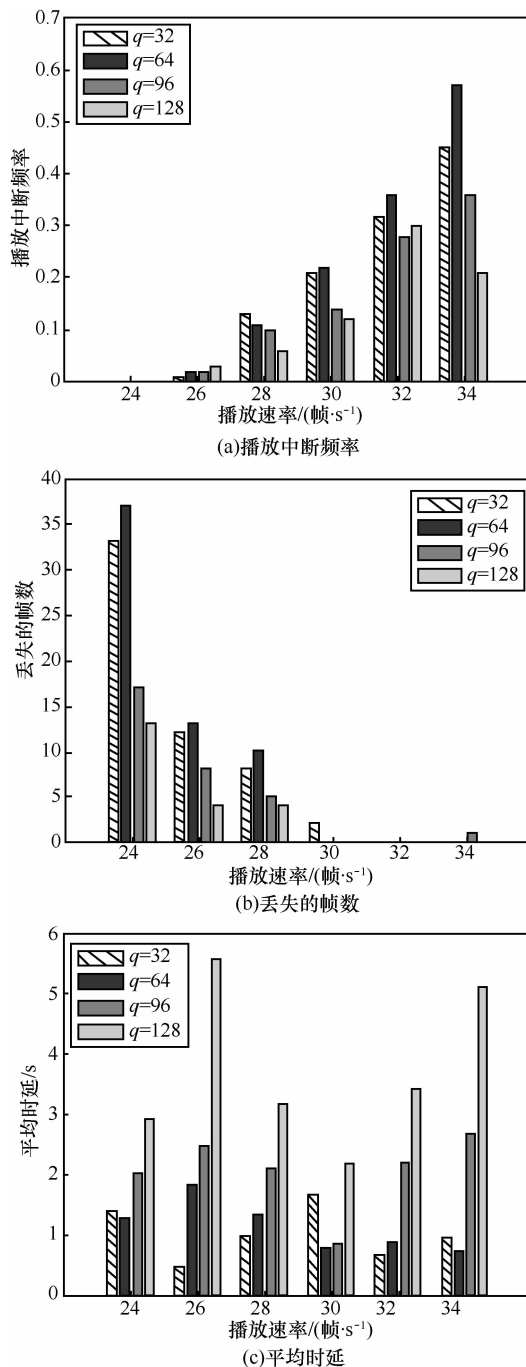


图4 播放速率一定时不同静态缓存的各项数据比较

间,对缓存的占用率非常低,造成系统资源的浪费。图5(d)为3种情况下缓存实际平均占用率的对比,由图可看出,不加控制时,速率越大,缓存的占用率越低,即资源利用率较低,而速率越小则缓存队列就要存储更多的视频帧,缓存占用率就比较高,但是存在一定的丢帧。而本文算法在通过及时调整缓存队列的长度来保证较高的利用率的同时,尽量降低丢帧率和中断概率。

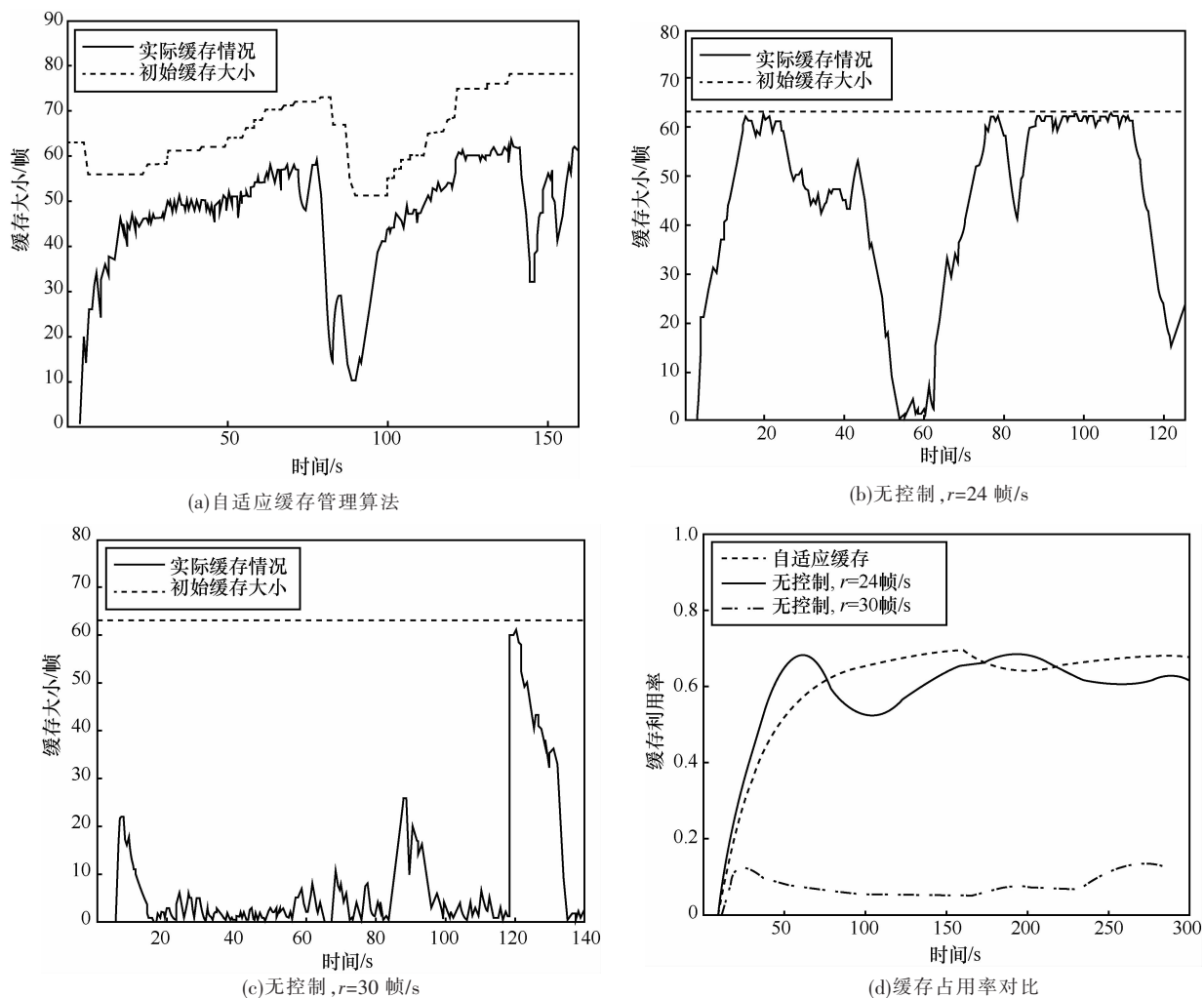


图 5 不同策略下实际缓存大小及其占用情况对比

图 6 为针对上述 3 种情况下多次测试的统计结果比较,共测试了 4 组数据,每组测试 10 次并取平均值作为统

计结果。图 6(a)中为下溢概率即播放中断频率的比较,可看出不加控制时,速率越大则出现中断的概率越高,而采用

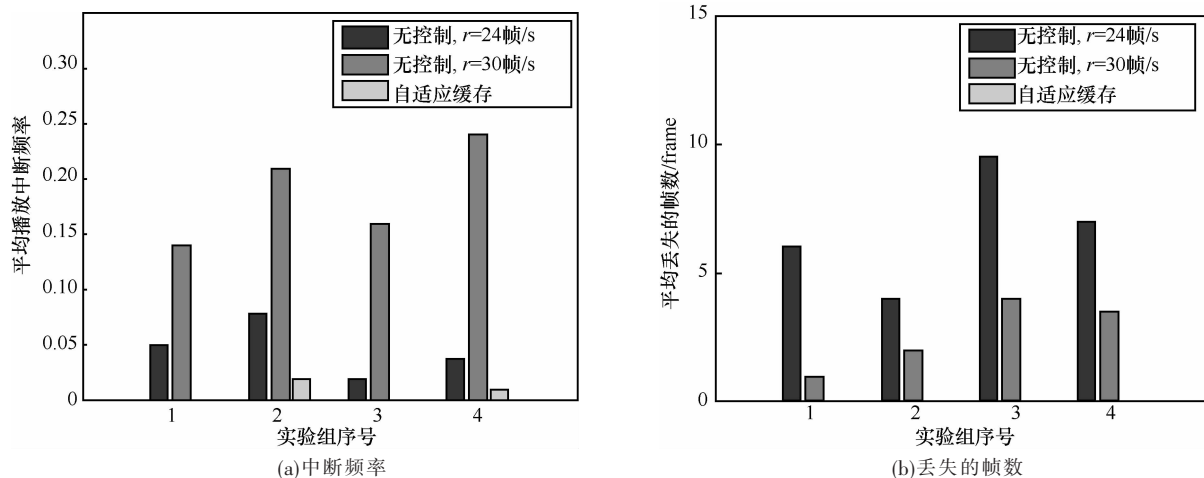


图 6 各项指标统计



本文的算法时可以大大降低下溢概率。如图 6(b)中所示,不加控制时,播放速率较低则缓冲容易被占满,较易出现上溢现象,相比之下,采用本文算法则基本上不会出现丢帧现象,通过同时对播放速率以及缓冲区大小的联合调整,在保证视频流畅播放的同时,可以有效地降低下溢和上溢概率。

图 7 为本文算法与不加控制下的平均时延对比,此时延为端到端的时延,由图可知,本文算法可以在时延较大的时候,通过调整播放速率来加快帧的播放,从而使其时延保持在一定的门限内波动,而不加控制时,则平均时延会由于缓存中帧的堆积或者因队列占满而进一步增大丢帧,在实时视频业务中这一点是不允许的,会对用户的体验造成一定的影响。本文算法可以根据当前的缓存状态来及时调整缓存大小和播放速率,能保持视频的时延处于用户的可接受范围内,在不影响视频观看的同时,较好地提高了用户体验。

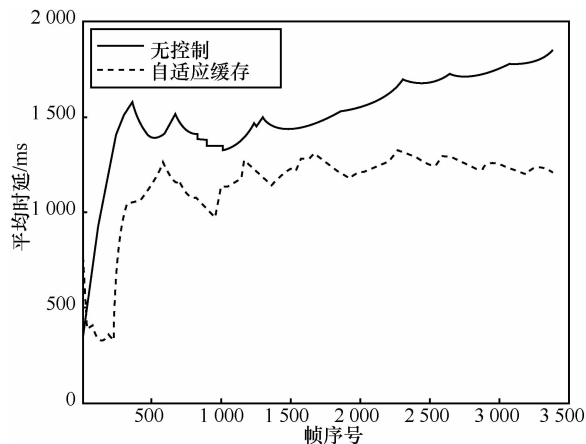


图 7 平均时延的对比

5 结束语

本文提出一种基于队列预测的自适应缓存管理方法,该方法可用于无线网络中的实时视频传输业务。实验表明,本文算法可以较好地解决由于网络环境的变化对用户观看视频带来的影响问题,通过联合调整缓存队列的长度以及视频播放的速率,从而有效降低了视频中断频率和丢帧率,并保证了端到端时延在可接受的范围之内。下一步的工作将对时延控制算法进一步改善,尽量将时延保持在较低的状态。

参考文献

1 Rui C, Zhengkun M, Liangbao J, *et al.* Delay controlled wireless

video playout system. IET Communications, 2010, 4 (11): 1348-1355

- 2 Sequeira L, Fernandez-Navajas J, Casadesus L, *et al.* The influence of the buffer size in packet loss for competing multimedia and bursty traffic. Proceeding of the 2013 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Toronto, 2013
- 3 Tianji L, Leith D. Adaptive buffer sizing for TCP flows in 802.11e WLAN. Proceeding of the Third International Conference on Communications and Networking in China, Hangzhou, 2008
- 4 Youn-Sik H, Ji-Hong K, Yong-Hyun K. A buffer-controlled adaptive video streaming for mobile devices. Proceedings of the International Conference on Convergence Information Technology, Gyeongju, 2007
- 5 Wanqing T, Weijia J. Adaptive playback buffer for wireless streaming media. Proceeding of the 12th IEEE International Conference on Networks (ICON 2004), Singapore, 2004
- 6 Jina Y, Han H, Hongsheng X, *et al.* Online buffer fullness estimation aided adaptive media playout for video streaming. IEEE Transactions on Multimedia, 2011, 13 (5): 1141-1153
- 7 Repele L, Muradore R, Quaglia D, *et al.* Improving performance of networked control systems by using adaptive buffering. IEEE Transactions on Industrial Electronics, 2013, PP (99)
- 8 林靖东. 研究称视频缓冲超过 2 秒就会有用户放弃观看. <http://tech.qq.com/a/20121112/000013.htm>, 2012-11-12
- 9 Sanghyun C, Womack B F. Qos-based optimal adaptive playout-buffer scheduling using the packet arrival distribution. Proceedings of the Military Communications Conference, Boston, MA, 2009
- 10 陈俊杰, 倪宏, 孙鹏. 基于 G/G/1 排队系统的播放缓冲控制机制. 网络新媒体技术, 2012, 1(3): 1-7

[作者简介]



钱超,男,南京邮电大学硕士生,主要研究方向为无线通信和多媒体通信。



朱晓荣,女,博士,南京邮电大学副教授,主要研究方向为异构网络融合、泛在网络、无线传感器网络等无线资源管理,跨层优化算法及协议设计。

(收稿日期:2014-03-18)