

# 一个基于速率控制的 Internet 视频流服务方案<sup>1</sup>

黄伟红 张福炎

(南京大学计算机科学与技术系 南京 210093)

**摘要** 由于视频流服务对于网络服务质量有着较高的要求, 而现有的 Internet 所提供的是尽力而为的服务, 无法保证数据的实时传输。该文设计了一个用于 Internet 上视频流的端到端传输方案。整个方案设计的目的是在网络本身缺乏服务质量保证的条件下尽可能达到最好的视频传输质量。根据可用带宽估计和网络信息反馈, 系统对发送速率进行调整, 并提供两种视频流服务: 存储视频和实时视频。仿真结果表明方案的性能良好, 能满足 Internet 视频流的需求。

**关键词** 自适应传输协议, 视频流, 存储视频, 实时视频, 实况视频

**中图分类号** TP391

## 1 引言

视频传输是多媒体系统与应用的重点之一, 它对于网络带宽、传输延迟、抖动都有着严格的限制。最初的视频传输研究是在 ATM 上展开的, 随着 Internet 的飞速发展, Internet 逐渐成为视频流应用的主流。但是由于 Internet 本身缺乏 QoS 保证, 并不适合视频数据的实时传输。因此, 当前的视频流支持技术的研究与开发主要是从两个角度进行的: 一方面, 提出新的 QoS 保证机制, 如资源预留 RSVP(Resource reSerVation Protocol) 技术<sup>[1]</sup>与实时传输协议族 RTP/RTCP/RTSP<sup>[2]</sup>。目前的商业产品如 RealVideo<sup>[3]</sup>和 MS Media Player<sup>[4]</sup>, 支持 TCP 协议族和 RTP 协议族, 但是它们对于网络拥塞状况下的自适应能力不是很强。另外一方面, 完善端到端的视频流服务, 无需更改现有的 Internet 基础架构, 并试图提供高质量的视频展现。B.Smith 提出了一个称为 Cyclic UDP 的尽力而为协议<sup>[5]</sup>, 探讨了基于优先级与延迟反馈的 MJPEG 视频传输机制用于适应网络拥塞。H.Kanakia 提出了一种自适应的拥塞控制方案<sup>[6]</sup>, 使用分离出的网络反馈信息来调整信源的数据生成速率。Z.Chen 等人提出了一个数据报协议 VDP(Video Datagram Protocol)<sup>[7]</sup>用于视频应用与 web 的集成, 但是缺乏速率的控制机制。S.Cen 等人提出了一个用于多媒体流的 SCP(Streaming Control Protocol) 协议<sup>[8]</sup>, 修改了 TCP 的拥塞控制机制。1999 年 R.Rejaie 提出了一种速率自适应协议 RAP(Rate Adaptation Protocol)<sup>[9]</sup>, 用于解决端到端的拥塞控制问题, 但没有考虑平滑传输和实时视频流的情况。

本文提出了一个面向 Internet 视频流的端到端解决方案, 传输协议使用网络反馈来控制拥塞和发送速率, 综合处理输入视频流的所需带宽、可用带宽、接收方缓冲容量以及其它相关的问题。第 2 节讨论了端到端系统结构以及其中的关键技术, 第 3 节详细叙述了实时视频和存储视频服务, 第 4 节给出了仿真结果, 最后是结束语。

## 2 视频流系统设计

**2.1 端到端视频传输结构** 一个面向视频传输的传输层的端到端协议必须包含许多特征, 如某种形式的拥塞控制, 数据传输的实时限制, 一些测量可用带宽以及与所需带宽的比较的方法, 优先级的使用等。本文的视频流系统结构如图 1 所示。其中, 分组预处理器完成从视频数据到传输分组的分割封装任务, 为了保证在给定丢失率和网络拥塞情况下接收方最佳的质量, 传输分组按视频内容的 I 帧、P 帧、B 帧从高到低作优先级标记。分组发送器与接收器, 完成对等的协议对话。实时视频服务与存储视频服务的区别在于, 实时视频数据经过编码, 直接传送到缓冲器, 然后进入网络传输过程, 无需经过视频文件存储过程。由于缓冲器的存在, 使得分组重传机制可以在展现时间允许的情况下进行, 从而保证一定程度的差错恢复。接收方在逻

<sup>1</sup> 2000-05-28 收到, 2000-10-25 定稿

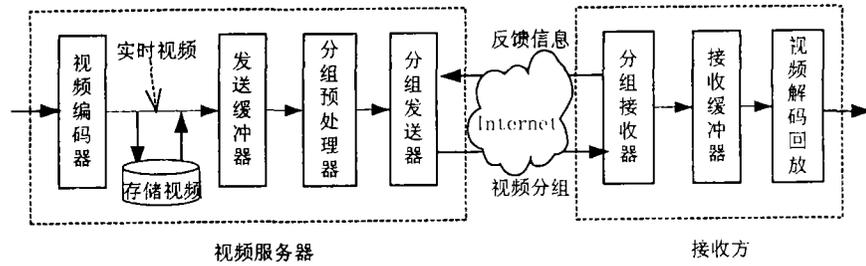


图 1 视频流服务系统结构

辑上被看作为由两个部分组成: 数据用户和分组接收器。分组接收器从网络上接收分组, 并存放在接收缓冲器中, 数据用户从接收缓冲器中读取分组, 进行解码并加以回放。

2.2 区间分割平滑传输 在视频传输中, 由于 Internet 状况的时变性, 经常会出现突发的拥塞, 为了降低突发对整个视频数据传输的影响, 我们采用区间分割的方法来代替连续不断的长时间传输, 以达到平滑传输的目的, 如图 2 所示。这种区间分割的方法在原理上虽然简单, 但是可以有效地增强传输的有效性。从机制上来说, 这种平滑不同于可变速率 (VBR) 数据用于降低负载变化而使用的平滑算法<sup>[10]</sup>, 与这些算法不同的是无需考虑将来帧的提前量, 简单易行, 而且并不依赖于发送方数据的类型, 支持恒定速率 (CBR) 和 VBR 视频的传输。

2.3 接收缓冲器设计 为了维持视频的平稳展现, 接收缓冲区的占用需要维持在一定的水平, 此时发送方的发送速率与接收方的接收速率相当, 那么缓冲的占用状况将得到维持。我们设计了一个环形接收缓冲器, 设有两个指针: 由分组接收器控制的接收器指针, 和数据用户控制的数据用户指针, 如图 3 所示。缓冲区使用的原则是保证接收器指针总是在数据用户指针之前, 两者之间的距离保证在一个极限区间内, 我们将这个差异称为缓冲器, 如果缓冲器的值大于最低极限值, 或者如果分组已经在前面的时间区间中丢失, 可以通过重传来恢复丢失的分组, 缓冲器的值作为一个反馈分组的内容发送到发送方, 从而改变发送速率。我们将缓冲器的两个极限分别定义为 MIN-THR 和 MAX-THR, 理想状况下, 缓冲器的值必须维持在这两个极限值之间, 并且尽量接近 MAX-THR, 这样如果有丢失的数据就有更多时间来恢复。MIN-THR 是一个距离数据用户指针的安全距离, MAX-THR 依赖于接收方的缓冲能力。

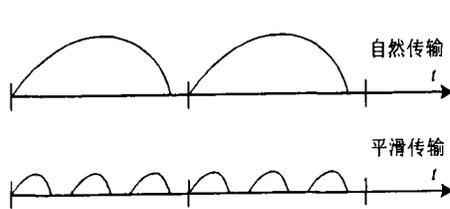


图 2 平滑传输与自然传输的效果对比

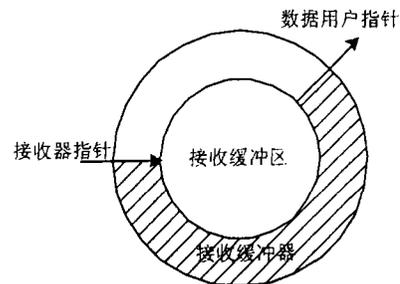


图 3 环形接收缓冲器

2.4 基于带宽估计的速率控制机制 传输系统的速率控制机制是根据网络带宽的反馈信息来进行的, 对可用网络带宽的准确估计有助于发送速率精确调整。最常用的速率自适应传输协议 TCP 和 TCP Vegas<sup>[11]</sup> 使用的是分组成对 (packet-pair) 的技术<sup>[12]</sup> 来估计网络带宽。考虑到协议的 TCP 友好性 (TCP-friendly), 本文也采用分组成对技术来进行带宽估计。我们将连续发送的两个标记分组经过传输后由于不同延迟 (传输延迟、传播延迟和排队延迟) 而到达接收方的时间差值记为  $\delta$ , 一段时间内, 重复的分组配对发送检测, 得到的最小的  $\delta$  值就表示最快

的服务速率。于是, 我们可以根据最小  $\delta$  值分割的分组情况来估计网络的最大带宽<sup>[12]</sup>。由于同一端到端连接上还可能存在其它的应用, 因此有

$$BW_{\text{可用}} = BW_{\text{最大}} - BW_{\text{已用}} \quad (1)$$

如果用  $\mu$  表示网络的使用情况, 那么可用带宽就可以表示为

$$BW_{\text{可用}} = BW_{\text{最大}} \times (1 - \mu) \quad (2)$$

对于 M/M/1 排队来说, 排队延迟为  $1/(1 - \mu)$ , 因此

$$BW_{\text{可用}} = BW_{\text{最大}} \times (\text{最小延迟} / \text{延迟}) \quad (3)$$

在实现中, 发送方可以使用回转时间 RTT, 因此可用带宽为

$$BW_{\text{可用}} = BW_{\text{最大}} \times (\text{最小 RTT} / \text{测量 RTT}) \quad (4)$$

在整个视频传输过程中, 发送方控制两个速率: 一个期望速率, 一个是估计速率。协议必须以两者之间的最低速率传输。如果期望速率比估计速率低, 足以用期望速率来发送分组, 相反, 如果估计速率比期望速率低, 发送方强制以估计速率发送, 以此来降低分组丢失、延迟超时和网络拥塞的风险。

### 3 存储视频服务和实时视频服务

3.1 存储视频服务 发送方的速率调整策略为: 如果缓冲器值  $<$  (MAX-THR), 那么

$$\text{期望速率} = \text{期望速率} \times \left[ 1 + \frac{\beta \times (\text{MAX-THR} - \text{缓冲器值})}{\text{MAX-THR}} \right], \text{ 否则}$$

$$\text{缓冲器值} = \text{MAX-THR} \quad (5)$$

其中常量  $\beta \leq 1$ , 用于期望速率的渐近修改。如果缓冲器的值大于 MAX-THR, 缓冲器的值不会增加, 这样, 缓冲器的值能够回落到稳定的区域。

接收方数据的管理, 实际上就是对接收缓冲器的管理, 维持在 (MIN-THR, MAX-THR) 的范围内, 避免上下溢出的出现。由接收方控制 (用于计算缓冲器的值) 的接收方指针只是在对应的当前时间区间内接收到特定的百分比 ( $x\%$ ) 的分组之后才变动, 参数  $x$  用于保证最基本的视频展现质量, 它提供了一个便捷的途径来平衡接收方的展现质量和停顿。当接收方指针现在指向  $i$ ,  $x\%$  的分组对应于  $(i + 1)$  时间区间中已经接收到的分组。接收方指针指向  $(i + 1)$ , 在这个时间段中, 如果分组相关于时间区间  $(i + 2)$ ,  $(i + 3)$  的已经接收, 它们只是拷贝到接收方缓冲中, 当  $(i + 2)$  区间中  $x\%$  的分组已经接收, 接收方指针移动到  $(i + 2)$ , 以此类推。

3.2 实时视频服务 每个传输区间被分割成多个子区间, 在每个子区间的开始, 期望速率通过公式来计算

$$\text{期望速率} = \theta \times (\text{要传输的数据量}) / \text{时间区间} \quad (6)$$

其中因子  $\theta > 1$  用于接收方重传的延迟调度, 我们取  $\theta = 1.25$ 。前一个区间的估计速率的测量用于在当前区间的开始。新的估计速率将用于下一个时间区间的目标速率的计算。目标速率以期望速率和估计速率之间的较小值来计算。在每个子区间中, 我们计算要传输的分组数目

$$\text{子区间中的分组数目} = \text{目标速率} \times \text{子区间} / \text{分组大小} \quad (7)$$

如上文所述, 我们使用分组配对技术得到可用带宽, 而实际速率从可用带宽中计算得到

$$\text{实际速率} = \alpha \times \text{实际速率} + (1 - \alpha) \times \text{估计带宽} \quad (8)$$

其中  $\alpha$  决定了历史估计速率与当前估计可用带宽之间的权值。

实时视频的接收方反馈给发送方的分组中包含 ACK 信息和分组序列号, 以此提供发送方对接收情况以及分组丢失的判断。在接收缓冲较大, 时间允许的情况下, 发送方在检测到分组丢失时重新进行分组重传, 重传分组被赋予比后来的分组更高的优先级。

#### 4 仿真结果

本系统采用 C 语言编程, 本文的实验视频为一段片长 10min 分辨率为  $384 \times 288$  的 MPEG 流, Internet 端到端连接为从深圳电信局技术开发中心到南京大学计算机系。主要实验内容有: (1) 拥塞仿真。主要目的是仿真实际 Internet 流量下的拥塞状况, 观察协议的自适应能力。由于网络状况的随机性, 拥塞是不可重现的, 因此我们采用模拟的方法来进行。主要做法是, 在同一个连接上, 增加并行的 UDP 流, 随机发送一定量的数据, 然后随机停止发送, 如此循环。由于 UDP 不具有回退特性, 因此可以造成一定程度的拥塞。为了对比效果, 我们还观察了自然的 Internet 下的拥塞实验结果, 以此作为对比。(2) TCP/UDP 流应用并行的实验。主要考察的是协议间的公平性, 是否会造成其它协议的带宽枯竭, 以及本身对并行流的适应能力。实验方法是在同一端到端连接之间同时运行一个或多个并行的 TCP/UDP 流, 如 FTP 流等。

4.1 拥塞仿真实验 本实验中的仿真拥塞状况下的平均传输速率为 45kbps。由于视频传输速率的不断变化, 有时会低于或者高于这个速率。为了对比起见, 本文还进行了自然状况下的拥塞实验, 即在实际的 Internet 上进行传输, 不加任何其它的额外流, 实验表明我们的传输方案对于长时间的突发丢失调整的较好。图 4 给出了两种状况下(自然拥塞和模拟拥塞)存储视频传输和实时视频传输与 UDP 的对比。由观察可知实时传输对于低丢失率的情况差于 UDP, 而存储视频传输则较好, 因为它允许一定数量的分组重传。

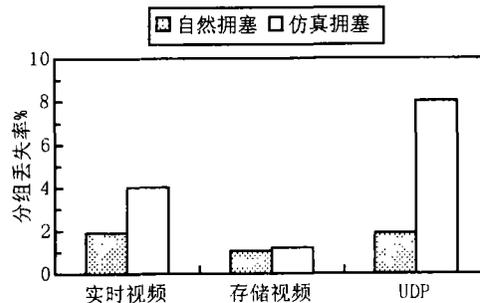


图 4 拥塞仿真中的丢失率

4.2 并行 TCP/UDP 会话实验 我们将实时视频和存储视频与多个 TCP 流(如 FTP 流)一块运行, 来观察带宽需求增加时对协议性能的影响。当多个 FTP 流运行时, 它们消耗了大量的带宽, 一个并行运行的自适应应用将试图尽量少占带宽, TCP 同样面向拥塞作调整。在模拟拥塞的环境下, 我们还将实时视频和存储视频与 UDP 一块运行, 为了比较的方便, 传输中使用在 FTP 传输中同样大小的数据。结合两个实验的数据结果, 对比结果如图 5 所示。

图 5 的结果表明我们的方案在并行 FTP 流的时候并没有导致许多的丢失, 实际上当并行的 FTP 流增加的时候, 每个 FTP 流的速率下降了, 对于 1 个 FTP 的情况, 观察得到的传输速率为 120 kbps, 而在 2 个和 3 个 FTP 的情况下, 观察速率分别为 110 kbps 和 110 kbps。文件传输协议 FTP 运行于 TCP 之上, 在网络拥塞的时候会降低发送速率。当我们的协议与 FTP 并行运行时, 我们的协议试图使用大量的带宽, 从而迫使 FTP 的速率降低。我们的协议在修改

发送速率之前, 在一段特定的时间(子区间)中维持计算得到的发送速率, 同时 FTP 检测到拥塞时降低速率来适应多个流。因为这样, 我们的协议没有导致许多的丢失, 甚至在多个像 FTP 一样的 TCP 应用并行的时候。从图 5 中我们可以发现并行的 UDP 连接比 TCP 会话导致更多的丢失。这是由于 UDP 无法实现对于拥塞的自适应, 我们的方案受到的网络带宽的压力时就较并行 FTP 之时为强。

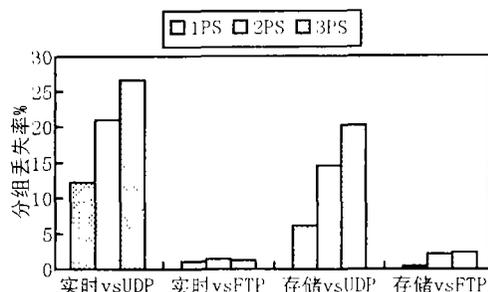


图 5 多个并行会话 (PS) 的 UDP 和 FTP 的丢失率

## 5 结论

为了实现 Internet 环境下的视频流的高效传输和展现, 本文提出了一个端到端的视频流方案。方案中采用了传输区间分割的平滑传输技术和基于分组建配的带宽估计技术, 分别提供存储视频服务和实时视频服务。在实时视频服务中, 考虑到信息的及时性, 我们没有像存储视频服务中一样考虑接收方的展现停顿, 在拥塞的时候图像质量较差。两种服务都采用基于速率的自适应机制。仿真实验结果表明本方案能够较好地适应适度的网络拥塞状况, 并占用较少的网络资源, 在与具有拥塞控制机制的流应用(如 FTP)并行时, 具有一定的鲁棒性, 但是, 在无拥塞控制流应用(如 UDP)并行时, 视频传输受到较大的影响。进一步的研究包括传输协议扩展(自适应参数调整)和向多播(Multicast)应用研究延伸等问题。

## 参 考 文 献

- [1] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala. RSVP: A new resource reservation protocol. IEEE Network, Sept. 1993, 7(5), 8-18.
- [2] AudioVideo Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889: RTP: A transport protocol for realtime applications, Jan. 1996, Status: Proposed Standard.
- [3] RealNetworks. <http://www.real.com/>
- [4] Microsoft Media Player. <http://www.microsoft.com/>
- [5] B. C. Smith. CyclicUDP: A priority driven best effort protocol. tech.rep. Cornell University, 1994. (<http://www.cs.cornell.edu/Info/Faculty/bsmith/nossdav.ps.gz>)
- [6] H. Kanakia, P. Mishra, A. Reibman, An adaptive congestion control scheme for realtime packet video transport, IEEE/ACM Trans. on Networking, 1995, 3(6), 671-682.
- [7] Z. Chen, S-M Tan, R. H. Campbell, Y. Li, Real time video and audio in the world wide web, The Fourth International World Wide Web Conference, Boston, Dec. 1995, 333-348.
- [8] S. Cen, C. Pu, J. Walpole, Flow and congestion control for Internet streaming applications, in Proc. Multimedia Computing and Networking, San Jose, California, January, 1998, 26-28.
- [9] R. Rejaie, An end-to-end architecture for quality adaptive streaming applications in the Internet, Ph.D. Thesis. University of Southern California, Dec 1999.
- [10] Z. L. Zhang, J. Kurose, J. D. Salehi, D. Towsley. Smoothing, statistical multiplexing and call admission control for stored video, IEEE J. on SAC, 1997, 15(6), 1148-1166.

- [11] L. S. Brakmo, S. W. O'Malley, L. L. Peterson, TCP vegas: New techniques for congestion detection and avoidance, in Proc. 1994, SIGCOMM Conference, London, UK, Aug. 31st-Sept, 24-35.
- [12] S. Keshav, Packet-pair flow control. tech.rep. AT& T Bell Laboratories, Murray Hill, New Jersey, 1994.

## AN APPROACH FOR INTERNET VIDEO STREAMING BASED ON RATE CONTROL

Huang Weihong    Zhang Fuyan

*(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)*

**Abstract** Video streaming requires high quality of service (QoS) guarantee of transport system, but current Internet only provides best-effort service, and is not suit for real-time transmission of video data. To address the issues of meeting delay-constrained video streaming over Internet, this paper presents an end-to-end adaptive transmission approach without any modification to the Internet infrastructure. The design emphasis is on trying to obtain the best quality of video reception possible in the absence of network QoS guarantees. According to the estimation of available bandwidth and network feedback information, the video steam sender could adjust the sending rate, and then achieve the best transmission efficiency. Two kinds of streaming services have been considered in proposed protocols: stored video and live video. Simulation results show the well performance of proposed system for Internet video streaming.

**Key words** Adaptive transmission protocol, Video streaming, Stored video, Real-time video, Live video

黄伟红: 男, 1975 年生, 博士生, 研究方向为网络多媒体.

张福炎: 男, 1939 年生, 教授, 博士生导师, 研究方向为多媒体信息处理, 数字图书馆.