

端到端可伸展视频在因特网中 鲁棒传输的解决方法*

王贵锦¹ 张 黔² 朱文武² 林行刚¹

(1. 清华大学电子工程系, 北京 100084; 2. 微软亚洲研究院, 北京 100080)

摘要 提出了一种端到端的方法解决可伸展视频在因特网上鲁棒传输的问题. 传统方案由于没有同时考虑媒体本身的特性、网络信道状况以及优化的数据封装等因素, 所以在易发生信息错误的环境下(例如网络丢包)不能得到全局优化而保证高质量的视频传输. 针对可伸展视频的特点, 给出了一种端到端的网络自适应的拥塞控制和非均衡的错误控制因特网视频传输方案, 该方案根据网络状况动态调整发送速率, 具有抗网络噪声的特性. 考虑到多媒体传输的要求, 描述了媒体拥塞控制来平滑媒体流的发送速率和预测网络可用带宽的方案. 在数据包的层次上, 提出了具有非均衡数据交织封装的错误保护方案, 很好地减轻了网络丢包对接收视频的影响, 并且进一步发展了可伸展视频在因特网传输中的率失真理论, 从而给出了最优的比特分配方案, 可以在信源码流和错误控制之间优化地分配可用网络带宽资源. 大量的实验结果表明该方案能很好地解决可伸展视频在因特网中的传输问题.

关键词 拥塞控制 错误控制 视频传输 率失真 可伸展视频

随着因特网的不断普及和网络资源的日益丰富, 流视频近年来成为一种非常重要的网络应用. 然而因特网 IP 网络层协议并不保证数据包按时到达接收端, 如果网络发生拥塞导致数据包没有按时到达接收端(视频流需要连续地播放), 流视频应用则认为发生了丢包, 接收端视频质量会严重下降. 但另一方面流视频并不需要全部数据包都正确到达, 在一个时间段中统计意义上总会有一些数据包按时到达接收端. 这样流视频就可以根据这些视频数据进行解码和视频重建. 所以流视频应用所面临的挑战是如何处理丢包获得高质量的重建视频的问题.

近来可伸展的视频流在通信领域中越来越受到重视^[1,2], 其码流的可伸展性可以很好地根据网络带宽的变化调整传送速率. 可伸展的源编码器将视频信号压缩成多层码流: 基层是最重要的数据层, 携带着视频信号的基本信息, 例如运动矢量, 所以要求可靠地传输; 其他层数据被称为增强层, 可以根据当前的网络带宽有选择地进行传输.

到目前为止阐述如何在因特网中进行视频传输的文献大概分为两类: 拥塞控制和错误控制. 拥塞控制^[3~5]基本思想是根据网络拥塞的状况调整发送速率从而试图减少数据包的丢失率, 同时尽量保持发送速率的平滑性来避免速率调整产生视频质量抖动. 但是上述研究只考虑了如何进行网络传输中拥塞控制而没有讨论网络出现丢包情况下如何控制视频通信的质量, 而

进行视频的错误控制. 整个视频传输的流程是这样的: 在发送端存储的是可伸展视频码流, 包括基本层和增强层等多层视频数据. 根据各层的重要性和当前的网络状况, 每层视频流采用不同的错误保护力度. 视频数据经过打包后被发送到因特网中. 在接收端, 视频数据首先经过数据解包、信道解码, 然后送到视频解码器中进行信号的重建. 与此同时接收端的网络监控模块记录一些统计信息, 例如端到端的数据包丢失率和整个传输延时. 根据反馈信息预测模块动态地估计网络可用带宽, 从而目标比特分配模块可以在源编码和信道编码之间进行资源分配而得到最佳的媒体接收质量.

在这里我们提出的端到端可伸展视频传输系统中关键的模块包括有非均衡错误保护、目标比特分配、非均衡数据封装、拥塞控制和网络预测等模块. 接下来我们将按照所提出的传输框架逐一介绍我们提出的算法, 第 2 节介绍的是我们的 TCP 友好的拥塞控制策略, 第 3 节则给出了我们提出的非均衡数据封装和非均衡错误保护、目标比特分配等技术.

1.2 相关的背景知识

1.2.1 可伸展视频编码器

这里我们采用渐进精细颗粒可伸展 PFGS(progressive fine granular scalability)视频流^[1]作为整个因特网流视频的实现例子. PFGS 是一个基于 MPEG-4 精细可伸展 FGS(fine granular scalability) 版本^[2], 且具有更高编码效率的视频编解码器. 它可以输出从几十千比特/秒到几兆比特/秒中任意精度码率的视频流. 基层码流保存最重要的信息, 例如运动矢量. 增强层保存的则是基于当前帧的基本层和参考帧的预测信息. 这样增强层发生错误或丢失时不会发生视频质量抖动. 而另一方面 PFGS 的同一帧中层与层之间码流是相关联的, 高层信息能否正确解码依赖于其低层信息. 即在接收端如果低层码流有任何残留错误, 则其对应的高层视频数据不管正确与否都将被解码器视为无效数据.

PFGS 提供了一些技术来增强码流抗差错的性能. 基层码流采用的抗差错技术包括再同步标志、头扩展编码、可逆变长码字、数据分离等, 可以在错误率 10^{-5} 以下得到可接受的视频图像^[11]. 增强层采用的抗错工具有再同步标志和头扩展编码方法. 如果增强层出现信息的错误或丢失, 解码器将丢弃两个同步码之间的比特数据, 搜索下一个同步码继续解码, 从而减少错误传播范围.

1.2.2 因特网数据包丢失模型

因特网中数据包丢失的测量表明网络丢包行为可以很好地用 2-状态 Markov 链来表示, 例如 Gilbert 模型^[12](见图 2). 如果数据包按时到达接收端则网络处于接收 R 状态, 而如果由于网络拥塞造成数据包丢失或者数据包到达的时间超过了规定时限, 则网络处于丢包 L 状态. 这个 2-状态 Markov 链的转移矩阵 A 可表示

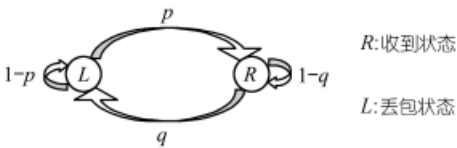


图 2 Gilbert 网络丢包模型

$$A = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix}, \tag{1}$$

其中 p 和 q 是状态 L 和状态 R 之间的转移概率参数, 可以通过最大似然方法得到

$$p = \frac{N_1}{N_1 + N_2}, \quad q = \frac{N_3}{N_3 + N_4}, \quad (2)$$

其中 N_1 (N_2) 是前一个包丢失后所成功收到(丢失)的数据包的个数; N_3 (N_4) 是前一个包成功接收到后所丢失(接收)的数据包的个数. 这些参数都可以通过接收端数据包统计得到.

2 网络自适应的拥塞控制

当发生拥塞时, 网络中的应用都应该自动调整它们的传输速率来减轻路由器的负荷使网络恢复正常状态. 这样才能保证网络的稳定性和各网络终端间传输的正常进行. 目前因特网通信流大部分是采用 TCP 协议的数据业务^[13]. 因此对于网络中的视频应用, 采用的传输策略必须要能与采用 TCP 协议的应用公平竞争网络带宽, 不独占网络资源. 为了避免 TCP 固有的网络丢包和带宽波动问题, 我们在传输层提出了多媒体传输 MSTFP 协议.

首先发送端需要根据反馈的信息来预测网络往返时间

$$RTT = \alpha \cdot \overline{RTT} + (1 - \alpha)(now - ST1 - \Delta RT), \quad (3)$$

其中 $\overline{RTT}$ 是当前的网络往返时间, RTT 是预测的网络往返时间, now 指示的是源端接收到反馈数据包的时刻, $ST1$ 是源端初始发送数据包的时刻, ΔRT 表示的是在接收端所耽搁的时间, α 是一个加权系数对所估计的参数进行平滑, 在这里设为 0.75. 这样, 网络带宽可估计为

$$AvailBW = \frac{PacketSize}{RTT \cdot \sqrt{2P_L/3} + 3 \cdot TO \cdot P_L \cdot \sqrt{3P_L/8} \cdot \sqrt{1 + 32P_L^2}}, \quad (4)$$

其中 $PacketSize$ 是发送数据包的长度, P_L 是网络丢包率, TO 表示的是数据包重发时需要等待的间隔^[14].

然后发送端就可以相应地调整自己的发送速率. 传输速率调整方案如下:

如果 ($AvailBW > \overline{currate}$)

$$multi = (now - lastchange) / RTT$$

限制 $multi$ 取值从 1 到 2

$$\overline{currate} = \overline{currate} + (PacketSize / RTT) \cdot multi$$

否则

$$\overline{curate} = \beta \cdot AvailBW + (1 - \beta) \times \overline{currate},$$

其中 $\overline{currate}$ 是当前正在传送的速率, $\overline{curate}$ 是更新的传送速率, $lastchange$ 是前一次更新的时刻, β 是一个加权系数, 同样设定为 0.75. 我们的拥塞控制的优势在于可以尽可能平滑地调整传输速率, 从而适合于连续视频流的传输, 并且对网络中的数据包丢失不敏感.

3 网络自适应的非均衡错误控制

尽管采用了 MSTFP 拥塞控制, 视频数据在网络中仍然可能经历丢包. 本节针对可伸展的视频提出了非均衡错误控制方案来减轻丢包对重建视频的影响, 其中包括非均衡交织数据封装、非均衡错误保护和比特资源分配三个部分.

3.1 基于非均衡交织数据封装的错误保护

为了抵抗网络中的数据丢包, 系统需要发送视频数据的同时伴随着校验冗余码. 本文采用的是建立在数据包级上的前向纠错技术. 其基本思想是接收端可以利用这些冗余的数据包

来恢复丢失的信息数据包. 我们所使用的信道编码是 Reed-Solomon (RS)码, 它非常适合用于恢复网络数据丢包^[15]. 这是因为 RS 码是 BCH 码中惟一具有最大码间距的码字, 也就是说, 不存在其他信道编码能用更少的冗余来重建丢失的数据包. 定义 $RS(n, k)$ 为长度为 n 的线性码, 其中包含 k 个源符号和 $n-k$ 个校验符号. 每个符号由 m 个比特组成, 码字长度 n 满足 $n \leq 2^m - 1$. 为了方便访问视频数据, 我们将符号长度设为 8 bit. 对于因特网流视频来说, 接收端可以通过数据包的序列号明确知道哪个数据包发生了丢失. 利用这些已知的错误位置信息, $RS(n, k)$ 可以恢复多达 $n-k$ 个符号错误. 对于数据包的每一列(以一个字节为单位), $RS(n, k)$ 将 k 个信息符号编码成 n 个符号, 从而生成 $n-k$ 个校验数据包(参见图 3).

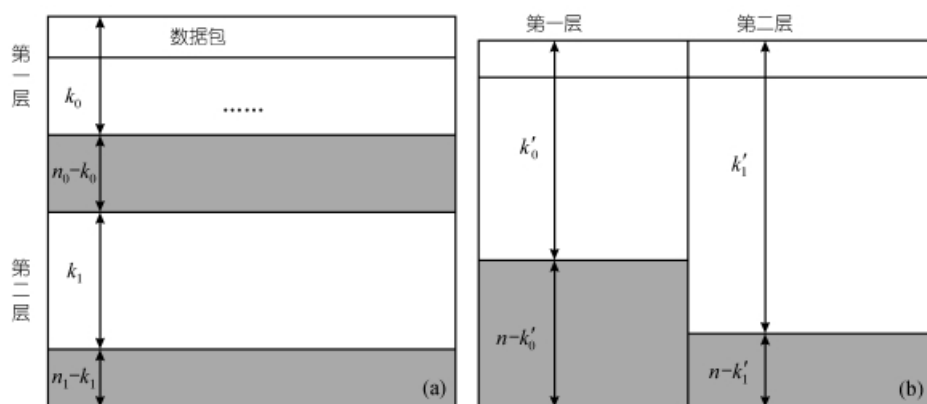


图 3 两种非均衡错误保护方案

(a) 传统非均衡错误保护方案, (b) 基于非均衡交织的错误保护方案

对于可伸展视频流, 第一层比第二层数据重要, 第二层又比第三层数据重要, 逐次类推. 根据码流具有优先级的特性, 我们在各层数据间使用非均衡丢包保护. 图 3 给出的是两种基于前向纠错的数据封装方式. 简单起见, 我们只列出了只有两层数据的情况. 在图 3(a)中, 当第一层数据 n_0 个数据包传输后紧接着发送第二层的 n_1 个数据包. 我们知道数据包在网络中的丢失状况呈现突发的特性. 如果在第一层发生突发错误(例如 $n_0 - k_0 + 1$), 即使第二层数据包能够完全恢复, 视频质量仍然会有严重下降. 因此我们提出交织的打包方式, 参见图 3(b). 其思路是通过对信息进行交织, 将错误进行随机化, 从而可以有效地进行错误恢复. 统一将一组帧(例如 20 帧)中所有层数据组装成一个 BOP 传输单位(block of packets), 每个数据包由各层数据块组成(每层一个). 对应图 3(a)和(b)中每层数据都有 n 个数据块, 其中 n 等于 $n_0 + n_1$, $k'_i \approx \frac{k_i}{n_i} n$.

当同样发生 $n_0 - k_0 + 1$ 个突发错误, 由于进行了交织, 各层数据所遭遇的突发丢失减少了, 接收端能够正确地利用冗余信息来恢复丢失的信息. 在我们的方案中可以用交织的深度来表明各层数据包的随机化的程度. 更大的交织程度, 则具有更小的突发丢包率, 从而更有利于前向纠错. 各层的交织的程度分别可以表示为 $(n_0 + n_1)/n_0$, $(n_0 + n_1)/n_1$. 对于可伸展码流, 增强层数据量逐渐增加, 因此低层码流具有更大的交织度, 从而得到非均衡的传输.

确定了数据封装方式, 以下小节讨论的是视频流各层前向保护的参数 $RS(n, k)$.

3.2 信源和信道保护之间的比特分配

在网络视频传输中, 我们考虑的系统失真 D_r 与传统的源编码中所涉及的量化失真不同的. 在有信息丢失的环境下, 端到端的失真 D_r 是一个随机变量, 其中包括信源失真 D_s (主要由于量化噪声引起) 和信道失真 D_c (主要由网络中的丢包所造成). 如果假定 D_s 和 D_c 是不相关的^[16], 则系统整体失真可以表示成 $D_r = D_s + D_c$. 定义随机变量 D_r 的期望值为系统的整体失真 D , 即

$$D = E\{D_r\} = E\{D_s + D_c\}, \quad (5)$$

其中 D_r 以一定的概率取 D_s 或 D_c , 这些概率主要取决于网络的状况. 前向纠错的思想就是通过传送冗余的数据包便于接收端重建丢失的信息. 为了补偿信道编码使用的网络带宽, 我们必须相应地减少视频信源的码率, 从而在发送端就引入了视频质量下降. 然而利用这些冗余的校验数据包, 视频解码端会得到比没有前向纠错时候低的数据包丢失率, 从而期望较好的重建视频质量. 所以关键在于如何在信源和信道之间分配好网络带宽以得到最小的端到端全局失真.

对于给定的目标带宽 R (字节为单位), 我们选择固定的数据包长 P_L , 这样所要传输的数据包的个数 n 等于 R/P_L . 定义矢量 $\bar{k} = [k_1, \dots, k_L]$; $\bar{k}' = [k'_1, \dots, k'_L]$, 其中 k_i 是第 i 层信息块的个数; k'_i 表示这个信息块的大小 (以字节为单位). 这样比特分配问题可以描述为如何得到矢量 $\bar{k}$ 和 $\bar{k}'$ 使得端到端的失真最小:

$$\begin{aligned} (\bar{k}, \bar{k}') &= \arg \min_{(\bar{k}, \bar{k}')} (D_s(R_s) + D_c(R_c)), \\ \text{s.t. } R_s + R_c &\leq R, \quad R_c = \sum_{i=1}^L \left(\frac{n}{k_i} - 1 \right) R_i, \quad R_s = \sum_{i=1}^L R_i, \end{aligned} \quad (6)$$

其中 R_i 是第 i 层视频数据大小. 在 (6) 式中, 视频层的数目 L 和最高层数据量 R_L 取决于信源速率 R_s .

信源失真可以大致估算为 $D_s(R_s) = A2^{-2R_s}$, 其中 A 为常数. 由于层与层之间的依赖关系, 很难计算信道失真. 为了简化信道失真的分析同时抓住可伸展视频码流的本质特性, 我们做了两个假设:

- 同一层的数据块具有同样的重要性.
- 同一层的数据块具有同样的错误扩散的方式.

此外, 信道编码可以看成是扩展的信源编码. 这样我们在接下来的推导中不再区分视频块和校验块. 仿真结果显示我们的简化是非常有效的, 更便于洞察可伸展视频在因特网传输过程中的率失真关系.

经过数据交织封装 (参见图 3(b)), 假设所要传输的视频流包括 L 层数据, 定义这 L 层所用的前向纠错 RS 码字分别是 (n, k_1) , (n, k_2) , $\dots$, (n, k_L) . 考虑到可伸展视频各层数据的优先级, 各层的纠错码参数满足: $k_1 \leq k_2 \leq \dots \leq k_L$. 因此根据所丢失的数据包的个数 r ($0 \leq r \leq n$), 我们首先确认第一层的数据块是否能无误地解码, 然后再看第一层和第二层数据块是否都能正确解码, 如此类推.

根据网络中丢失的数据包的个数 r , 定义函数

$$c(r) = \arg \max_{j=0, \dots, L} \{(n - k_j) > r\}. \quad (7)$$

也就是说给定丢失的数据包个数为 r , 前 $c(r)$ 层都能完全正确解码不会产生信道失真; 而对于后面的 $L - c(r)$ 每层视频数据都有 r 个数据块丢失, 产生了信道失真. 这时信道失真等于

$$D_c(R_c) = \sum_{r=0}^n \{P(r, n) \cdot r \cdot \sum_{i=c(r)+1}^L D(i)\} / n, \quad (8)$$

其中 $P(r, n)$ 是在 n 个数据包中丢失 r 个的概率, $D(i)/n$ 表示在第 i 层丢失一个数据块在本层所造成的失真. 这样全局率失真关系可以表述为

$$D(R) = A2^{-2R_s} + \sum_{r=0}^n \{P(r, n) \cdot r \cdot \sum_{i=c(r)+1}^L D(i)\} / n. \quad (9)$$

将(9)式代入(6)式中, 比特分配的优化问题可以认为是在网络可用带宽限制下最小化系统整体失真 $E\{D(R)\} = E\{D_s(R_s) + D_c(R_c)\}$. 也就是说

$$\begin{aligned} (\bar{k}, \bar{k}') &= \arg \min_{(k, k')} (A2^{-2R_s} + \sum_{r=0}^n \{P(r, n) \cdot r \cdot \sum_{i=c(r)+1}^L D(i)\} / n), \\ \text{s.t. } R_s + R_c &\leq R, \quad R_c = \sum_{i=1}^L \left(\frac{n}{k_i} - 1 \right) R_i, \quad R_s = \sum_{i=1}^L R_i, \quad P_L = \sum_{i=1}^L n'_i. \end{aligned} \quad (10)$$

尽管经过简化处理, 搜索各层的最佳码率仍然是件困难的事情, 尤其这个端到端失真对于各视频层的码率并不是凸函数. 这可以给一个示意性的解释: 当网络情况非常恶劣的时候, 增加一个某一层校验包 $t_i = n - k_i$ 的时候, 虽然增加了保护强度, 但信道失真基本上没有减少, 而加大了信源失真, 这样整体失真反而增加. 所以需要采用迭代的算法来解决这个优化问题^[17,18](具有 $O(n^L)$ 复杂度). 接下来我们给出亚优化比特分配方法, 可以快速得到各层的信源信道速率.

在(9)式中交换 r 和 l 的顺序, 信道失真也可以等价

$$D_c(R_c) = \sum_{r=0}^n \{P(r, n) \times r \times \sum_{i=c(r)+1}^L D(i)\} / n = \sum_{i=1}^L \{ \sum_{r=n-k_i+1}^n P(r, n) \times r \times D(i) / n \} = \sum_{i=1}^L D_{\text{channel}}(i), \quad (11)$$

其中 $D_{\text{channel}}(i)$ 是只与第 i 层相关的信道失真. 在这个表达式中, 当满足(10)式中的约束, 信道

失真可以分解为若干独立部分. 相应的率失真可以表述为 $D(R) = D_s(R_s) + \sum_{i=1}^L D_{\text{channel}}(i)$.

当各视频层取最优值时必要条件是

$$\sum_{i=1}^L \frac{\Delta D_{\text{channel}}(i)}{\Delta R_i} \leq \frac{\Delta D_s(R_s)}{\Delta R_s}, \quad (12)$$

也就是说冗余的校验数据包已经能很好地抵抗网络丢包, 这时候再增加信道码率只会使整体失真增大. 为了使视频传输时更加鲁棒, 我们加强(12)式右面的约束

$$\sum_{i=1}^L \frac{\Delta D_{\text{channel}}(i)}{\Delta R_i} \leq \frac{\Delta D_s(R)}{\Delta R} = D'_s(R) \leq \frac{\Delta D_s(R_s)}{\Delta R_s}, \quad (13)$$

其中 $D'_s(R)$ 是一个常数. 因为 $D_{\text{channel}}(i)$ 只依赖 $RS(n, k_i)$, 所以如果我们将 $D'_s(R)$ 均分 L 份, 这样就可以把(10)式中的矢量优化问题简化为若干个标量问题:

$$\begin{aligned} \hat{k}_L &= \arg \left(\frac{\Delta D_{\text{channel}}(L)}{\Delta R_i} \leq D'_s(R)/L, \forall k_L \leq \hat{k}_L \right) \quad \text{s.t.} \quad \hat{k}_L \leq n, \\ \hat{k}_i &= \arg \left(\frac{\Delta D_{\text{channel}}(i)}{\Delta R_i} \leq D'_s(R)/L, \forall k_i \leq \hat{k}_i \right) \quad \text{s.t.} \quad \hat{k}_i \leq \hat{k}_j, \quad \forall i \leq j, \\ \Delta D_{\text{channel}}(i) &= P(n - k_i + 1, n) \cdot (n - k_i + 1) \cdot D(i) / n, \quad \Delta R_i = \frac{n}{k_i - 1} R_i - \frac{n}{k_i} R_i. \end{aligned} \quad (14)$$

其中 L 可以就取其最大的可能值(当信源码率 R_s 等于 R). 这样每层的优化校验包数只需简单地 进行一维搜索就可以. 整个优化的复杂度降低到 $O(L \times n)$, 远远低于(10)式中的复杂度.

4 实验结果

本节给出了大量仿真实验来验证所提出的端到端网络自适应的视频传输方案在各种网络 信道环境下的性能. 实验中我们测试了四种传输方案: 1) 无交织数据封装的固定丢包保护, 表示为 FULP/I (BL 的保护比例是 40%, EL 的保护比例是 17%); 2) 交织数据封装的固定丢包保 护, 表示为 FULPI (BL 的保护比例是 36%, EL 的保护比例是 17%); 3) 本文的网络最优化错误 控制方案, 包括非均衡丢包保护、非均衡交织和优化比特分配, 其中采用的是迭代方法得到最 优的比特分配, 表示为 OULPI; 4) 本文的网络亚优化错误控制方案, 包括非均衡丢包保护、非 均衡交织和优化比特分配, 其中采用的是我们提出的亚优化分配方案, 表示为 SOULPI. 所有 方案中都采用了我们提出的拥塞控制方案进行监控预测网络状况来提高网络带宽使用效率.

在所测试的视频序列中, 除了第一帧是帧内编码外其他帧采用的都是帧间预测编码. 所 用的视频序列包括 Foreman 和 Coastguard (CIF 分辨率). 不影响方案的普适性, 视频离线编码 的帧率是 10 帧/s. 在仿真中我们考虑了各种网络情况, 网络带宽从 320 kbps 到 1 Mbps, 因特 网包丢失率从 0.5% 到 25% 之间变化, 网络平均突发的丢包个数为 2~3. 为了便于观察各方案在 网络中的性能, 我们进一步将网络丢包情况分为三种情况: (a) 高的网络丢包率从 15% 到 25%; (b) 中等网络丢包率从 5% 到 10%; (c) 低的网络丢包率从 0.5% 到 1%. 下面的实验结果如果没有 特别说明都是经过 30 多次 Monte Carlo 实验平均得到的.

图 4 和 5 验证的是目标比特分配模块的性能, 给出的 Foreman 和 Coastguard 在各种网络 条件下各种方案的性能. 一方面可以看到我们给出的亚优化比特分配 SOULPI 在任何网络情

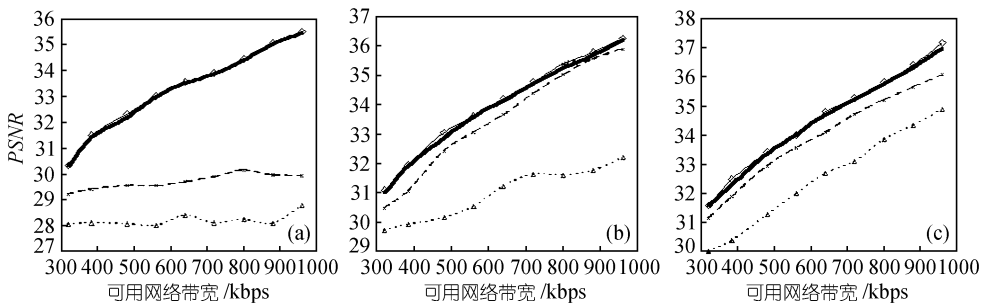


图 4 视频序列 Foreman 在各种比特率下各种方案的平均 PSNR
(a) 高网络丢包率, (b) 中等网络丢包率, (c) 低网络丢包率
--△--示 FULP/I, --×--示 FULPI, --◇--示 OULPI, ——示 SOULPI

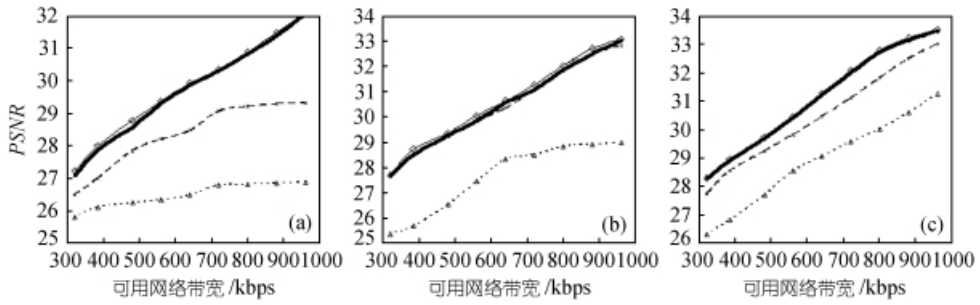


图 5 视频序列 Coastguard 在各种比特率下各种方案的平均 PSNR

(a) 高网络丢包率, (b) 中等网络丢包率, (c) 低网络丢包率

--△--示 FULP/I, --×--示 FULPI, --◇--示 OULPI, ——示 SOULPI

况下(不同网络带宽和网络丢包率)可以获得和最优比特分配方案 OULPI 差不多的重建视频质量, 而比特分配复杂度减少至 $O(L \times n)$, 具有很高的应用价值. 另一方面, 和固定丢包保护方案 FULPI 和 FULP/I 相比, 可以看出 OULPI 和 SOULPI 方案(结合提出的所有技术: 包括拥塞控制、非均衡丢包保护和数据交织封装技术及率失真比特资源分配)在各种网络条件下都取得了最好的系统性能. 并且网络可用带宽越大, SOULPI 获得的增益也越大(相对于 FULPI 和 FULP/I). 其中的原因可以解释为 OULPI 和 SOULPI 方案分配可以根据网络丢包情况和网络可用带宽以及各层视频层对整体质量的影响确定各层的保护力度, 从而获得最小的整体失真. 对于可伸展的视频流, 如果低层出现了错误, 其对应的高层数据则变成无效, 所以随着数据层的增高, 比特数据无效的概率也逐渐增加. 在网络丢包状况相同时, 可用带宽增大, OULPI 和 SOULPI 会相应调整(增强)低层的保护强度, 这样能够加大高层数据有效概率, 从而整个系统性能随着网络带宽增加而更好, 遵守了率失真的变化趋势. 而对于固定丢包保护方案, 不管当前网络带宽的多少, 各个比特数据无效的概率是固定的(在一定的网络丢包率下). 所以当比特的无效概率趋于 1 时, 传输方案的性能在一定网络带宽后就基本保持不变. 在(a)信道中, 对于 Foreman 和 Coastguard 序列 SOULP 相对于 FULPI 甚至可以获得高达 5.5dB 和 2.7dB 的增益(在 960kbps 网络带宽下). 在(b)类网络中, 所有方案都可以取得很好的性能, 尤其是 OULPI, SOULPI 和 FULPI 性能非常接近, 这表明这个固定丢包保护非常适合这样的网络丢包情况. 当网络状况进一步改善(参见图 4(c)和图 5(c)), SOULPI 将更多的比特资源分配给了信源从而可以在发送端获得更好的初始传输视频, 这样相对于 FULPI 方案分别获得高达 2.0 和 0.9 dB 的系统增益.

接下来我们分析数据交织封装技术给系统性能带来的增益. 数据交织也可以帮助恢复丢失数据包, 通过数据交织数据封装技术, 网络噪声能够被进一步离散化, 接收端更容易利用前向纠错重建丢失的信息; 从另外一个角度来看, 经过数据交织以后, 每一层的码流所使用的信道编码 RS 码字都比原来未交织前大, 信道编码的效率得到提高, 从而整体的性能变得更好. FULPI 在传输过程中相比较 FULP/I 的增益来自两部分: 信源增益和信道增益. 在基本层中, FULPI 和 FULP/I 两个传输方案均有很强保护力度, 分别是 36%和 40%. 由于 FULPI 采用了数据交织封装, 所以用比 FULP/I 少 4%的校验就可以基本保证基本层能够在网络中无丢失的传输. 这样 FULPI 信源增益就来自于基本层少用的 4%校验数据. 而在增强层上两个方案都使

用了 17% 相同的保护. 所以 FULPI 所获得的信道增益就只来自二者数据封装的差异: 数据交织技术. 举个简单的例子, 在图 4(b) 中, FULPI 在 960 kbps 带宽下获得 3.6 dB 增益(相比较 FULP/I). 通过图 4(c) 和图 5(c), 我们可以观察到 FULPI 在所有网络比特速率时候都比 FULP/I 高出 1 dB 左右, 此时 FULPI 和 FULP/I 都可以很好抵抗网络中的丢包, 整个端到端失真主要是信源失真, 也就是说由于信源本身带来的增益是 1 dB. 这样我们可以推测出在该网络条件下由于数据交织导致的丢包恢复方面的系统改善是 2.5 dB 左右.

综合实验结果, 我们总结出以下三条结论. 第一, 网络带宽越大, 相比固定的非均衡丢包保护方案, 具有率失真资源分配的方案性能更好. 在优化的比特分配中我们考虑了各层的优先级/重要性以及变化的网络条件等因素, 而固定丢包保护方案则因为没有考虑网络条件的变化从而不能很好地适应网络传输. 第二, 数据交织技术同样对因特网视频传输系统有着重要的影响. 数据交织有利于系统对网络丢失的数据包的恢复, 这个增益取决于网络的状况, 甚至可以高达 3 dB. 第三, 即使信道条件很好的情况下, 率失真分配的方案相比较固定非均衡丢包方案仍然可以获得不小的整体增益. 这是因为 SOULPI 在网络条件佳的时候分配更多的资源给信源, 从而获得高质量的发送初始视频.

图 6 给出的是 Coastguard 在 480 kbps 网络带宽下各种传输方案实时传输的 PSNR 对比结果. 横坐标是接收端重建视频的帧号, 它们与编码端发送的时刻只相差一个应用起始延时, 也就是一个 BOP 的延时. 在我们的实验中一个 BOP 包含 20 帧视频, 所以接收端的初始延时是 2 s. 可以看出我们所提出的网络自适应拥塞控制和错误控制的方案均优于其他对比方案. 此外 SOULPI 方案具有更平滑的重建视频质量, 更符合人们的主观感受. 对应的固定非均衡丢包保护方案由于没有考虑到网络状况不能很好地抵抗网络丢包, 往往会造成视频质量有比较大的

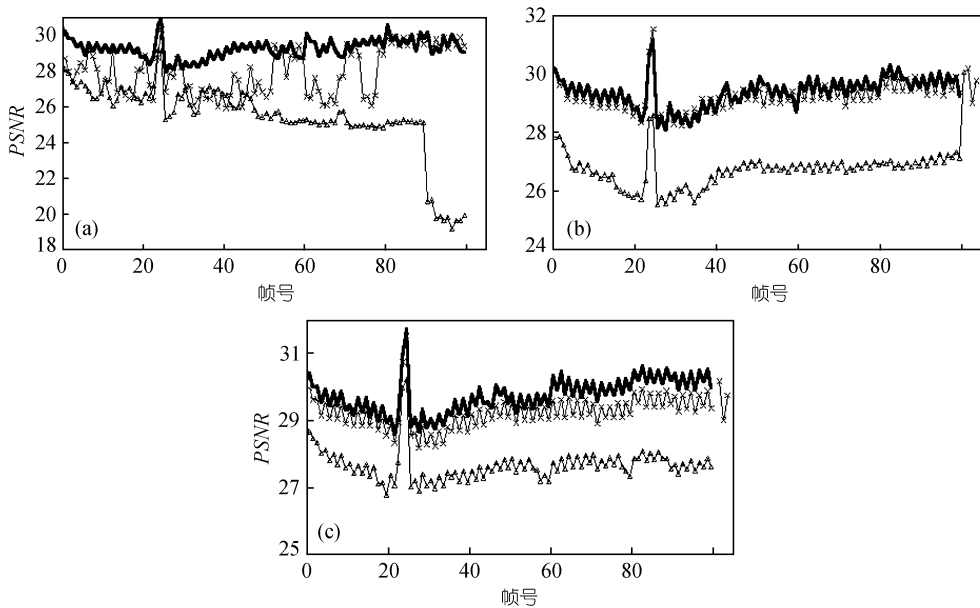


图 6 在网络带宽 480 kbps 时各种方案实时传输 Coastguard 序列 PSNR 对比结果

(a) 高网络丢包率, (b) 中等网络丢包率, (c) 低网络丢包率

—△—示 FULP/I, —×—示 FULPI, —示 SOULPI

抖动. 此外 SOULPI 可以根据预测的网络状况, 对 BL 进行很好地保护, 所以视频可以很好地在接收端重建. 然而对于固定的非均衡丢包保护方案, BL 信息可能会被损坏并且错误不能被后处理所掩盖, 这样就会导致很差的视频质量甚至会发生解码器的中断退出严重后果.

从图 4~6 等实验结果可以看出本文提出的端到端可伸展视频传输方案在各种网络条件下(不同的网络带宽和不同的丢包率), 无论是主观效果还是客观指标都可以获得高质量鲁棒的接受效果.

5 结论

因特网中流视频在过去几年来日益成为非常重要的应用, 例如视频点播. 在这个应用中存在的挑战是如何处理网络带宽的抖动和网络丢包, 在接收端获得高质量的视频重建. 本文提出了一个新颖的可伸展视频在因特网上进行端到端鲁棒的传输方案. 本文主要的贡献有:

- 联合考虑了系统中各个关键模块: 网络自适应拥塞控制、非均衡数据交织封装、非均衡错误保护和比特资源分配.
- 在数据包的层次上我们提出了非均衡数据交织和错误保护方案, 能很好地减轻网络丢包对重建视频的质量造成的恶化.
- 发展了可伸展视频流在因特网传输的率失真理论, 同时给出了相应的最优的比特分配方案, 在信源码流和错误保护之间分配可用的网络带宽资源.

大量仿真实验表明本文的方案对可伸展视频在因特网中的传输系统得到全局优化和高质量的重建视频.

参 考 文 献

- 1 Li S P, Wu F, Zhang Y Q. Study of a new approach to improve FGS video coding efficiency. ISO/IEC JTC1/SC29/WG11, MPEG99/m5583, December 1999, Maui
- 2 Li Weiping. Overview of fine granularity scalability in MPEG-4 video standard. IEEE Trans on Circuits and Systems for Video Technology, 2001,11(3): 301~317
- 3 Zhang Q, Zhu W, Zhang Y Q. Network-adaptive rate control with TCP-friendly protocol for multiple video objects. In: IEEE International Conference on Multimedia and Expo (ICME), July, 2000, New York. 2000
- 4 Rejaie R, Handley M, Estrin D. An end-to-end rate-based congestion control mechanism for realtime streams in the internet. In: Proceedings of INFOCOMM 99. 1999
- 5 Wu D, Hou Y T, Zhu W, et al. On end-to-end transport architecture for MPEG-4 video streaming over the internet. IEEE Trans on Circuits and Systems for Video Technology, 2000,10(6): 923~941
- 6 Zhu W, Zhang Q, Zhang Y Q. Network-adaptive rate control with unequal loss protection for scalable video over internet. In: IEEE International Symposium on Circuits and Systems, 2001, Australia. 2001
- 7 Mohr A E, Riskin E A, Ladner R E. Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction. IEEE Journal on Selected Area in Communications, 2000, 18(6): 819~828
- 8 Stuhlmuller K, Link M, Girod G, et al. Scalable Internet Video Streaming with Unequal Error Protection. New York: Packet Video Workshop 99, 1999
- 9 Zhang T T, Xu Y. Unequal packet loss protection for layered video transmission. IEEE Trans on Broadcasting, 1999, 45(2): 243~252
- 10 Schulzrinne H, Casner S, Frederich R, et al. RTP: A transport protocol for real-time applications. RFC 1889, Internet

Engineering Task Force, Jan 1996

- 11 Gringeri S, Egorov R, Shuaib K, et al. Robust compression and transmission of MPEG-4 video. In: ACM MM 2000 Electronic Proceedings, June 2000
- 12 Bolot J C, Turetti T. Adaptive error control for packet video in the internet. In: Proc IEEE ICIP'96. Sept 1996
- 13 Thompson K, Miller G J, Wilder R. Wide-area internet traffic patterns and characteristics. IEEE Network, 1997, 11(6): 10~23
- 14 Padhye J, Firoiu V, Towsley D, et al. Modeling TCP throughput: A simple model and its empirical validation. In: Proceeding of SIGCOMM'98. 1998
- 15 Blahul R E. Theory and Practice of Error Control Codes. Reading, MA :Addison Wesley, 1983
- 16 Stuhlmuller K, Farber N, Link M, et al. Analysis of video transmission over lossy channels. IEEE Journal on Selected Areas in Communications, 2000, 18(6): 1012~1032
- 17 Cheung G, Zakhor A. Optimal bit allocation for joint source/channel coding of scalable video. IEEE Transactions on Image Processing, 2000, 9(3): 340~357
- 18 Ramchandran K, Ortega A, Vetterli M. Bit allocation for dependant quantization with applications to multiresolution and MPEG video coders. IEEE Trans On Image Processing, 1994, 37: 533~545