

一种自适应的视频流化前向纠错算法*

梅 峥⁺, 李锦涛

(中国科学院 计算技术研究所,北京 100080)

An Adaptive Forward Error Correction Algorithm for Streaming Video

MEI Zheng⁺, LI Jin-Tao

(Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62565533 ext 9604, Fax: +86-10-82611846, E-mail: meiz@ict.ac.cn, <http://www.ict.ac.cn>

Received 2003-11-04; Accepted 2004-01-07

Mei Z, Li JT. An adaptive forward error correction algorithm for streaming video. *Journal of Software*, 2004,15(9):1405~1412.

<http://www.jos.org.cn/1000-9825/15/1405.htm>

Abstract: Video applications over network are often disturbed by data packet loss or errors as well as the insufficiency of network bandwidth. Some related studies have demonstrated that in many cases, the fluctuation of network bandwidth and packet loss rate are two key factors that influence the quality of video streaming. Therefore for guaranteeing video quality, FEC (forward error correction) can be adopted to improve the reliability of video data transmission; meanwhile, according to the current network state, the sender can adjust the sending rate of video data and optimally allocate the bandwidth resource between the video source data and FEC data. This paper analyzes the structure of video stream, and presents a frame decoding model that takes into account the frame types and the dependence among frames. On this basis, an optimal algorithm is proposed to allocate the bandwidth resource between the source video data and the FEC data. Experiments show that the model can effectively adapt to the fluctuation of network state, and optimally allocate network bandwidth so as to maximize the playable frame rate on receiver.

Key words: video streaming; FEC (forward error correction); playable frame rate; TCP-friendly

摘要: 网络视频应用经常会受到数据包丢失或错误以及网络带宽资源不足的干扰。相关研究表明:在多数情况下,动态变化的网络带宽和丢包率是影响视频流化质量的关键因素。因此,为了保证视频质量,可以采用前向纠错(forward error correction,简称 FEC)编码来提高视频数据传输的可靠性;同时,为了适应网络状态的变化,发送端可以调节视频数据的发送速率,并在视频源数据与 FEC 数据之间合理分配网络传输带宽。首先通过对视频流结构的分析,在充分考虑帧之间的依赖关系和帧类型的基础上提出了一种帧的解码模型。在此基础上,建立了用于在视频源数据和 FEC 数据之间分配网络带宽资源的优化算法。实验表明,该模型可以有效地适应网络状态的

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA114060 (国家高技术研究发展计划(863))

作者简介:梅峥(1975—),男,天津人,博士,主要研究领域为多媒体技术;李锦涛(1962—),男,博士,研究员,博士生导师,主要研究领域为多媒体技术,虚拟现实技术。

变化,并通过优化分配网络带宽资源来使接收端获得最大的可播放帧率.

关键词: 视频流化;前向纠错;可播放帧率;TCP友好

中图法分类号: TP316 文献标识码: A

多媒体数据压缩编码技术与计算机网络通信技术的相互结合产生了大量新的视频流化应用,并成为近年来研究和应用的一个热点领域.当前,视频压缩标准的目标是对视频信号进行压缩,为了实现这一目标大部分视频标准都采用了基于宏块的帧间预测和运动补偿技术.这些编码方法减少了大量的冗余信息,但同时也在压缩码流中不同部分的视频数据之间形成了很强的解码依赖性.由此产生的直接后果是:因网络传输差错造成的部分数据包丢失或损坏会导致另外一些与之相关的视频数据,即使被正确接收也无法使用.例如,通常视频帧有帧内(intra-frame)和帧间(inter-frame)两种编码方式.采用帧内编码方式的帧可以独立解码;而采用帧间编码方式的帧则必须在其所依赖的全部参考帧被正确接收后方可正常解码,反之将产生所谓的错误传播现象^[1].由此可见,不同类型的视频帧丢失对视频播放质量所造成的影响是不一样的.

为了减少视频数据丢失和错误对解码质量造成的不利影响,需要使用一些错误控制技术来提高视频数据在网络上传输的可靠性,其中通常采用的两种方式是:ARQ(automatic repeat request)和 FEC(forward error correction).ARQ 通过反馈应答方式来保证数据的可靠性,当接收端正确接收到数据后,必须向发送端发送确认信息,否则发送端将重传数据直至收到确认信息后再发送新的数据.这种方式的优点是可以保证数据的正确性,但是会消耗发送端很多资源而且延迟较长,因此不适合实时应用;FEC 通过产生一定的冗余数据来检测和纠正数据错误,虽然 FEC 会浪费一定的网络带宽,但是延迟短,所以 FEC 更适合于网络上的视频传输^[2].

本文主要针对在网络上实时传输数据的视频应用.在这种情况下,视频帧的生成速度往往是固定的.受带宽和丢包率的影响,如果在带宽不足或者丢包率较高的情况下发送全部的数据帧,将会使接收端的可播放帧率(正确解码的帧率)明显低于发送端的发送帧率.本文通过对视频流中帧结构的分析,建立了能够正确描述视频帧之间解码依赖性和重要性的树形结构,并据此提出了网络传输过程中最大化接收端可播放帧率的 FEC 数据分配模型及相关算法.该算法可以根据网络带宽和丢包率,通过决定帧的发送模式在视频源数据和 FEC 之间优化分配带宽,使得可播放帧率最大.

本文第 1 节介绍相关基础知识和现有工作.第 2 节分析建立视频帧的解码依赖关系树,然后建立以接收端可解码帧数量为优化目标的数学模型.第 3 节给出相应的求解算法.第 4 节为仿真实验及分析.第 5 节总结全文.

1 相关知识和已有工作

1.1 视频帧结构

目前国际上流行的视频压缩标准主要出自 MPEG 和 ITU 两大国际组织,分别为 MPEG 系列和 H 系列.这两大标准虽然在技术细节上有所差异,但是其基本的原理都是相似的.下面以 MPEG-2 为例进行说明.

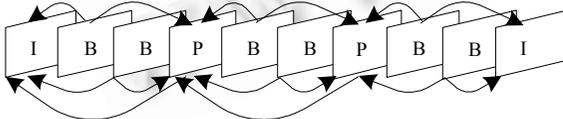


Fig.1 Structure of GOP

图 1 GOP 结构

MPEG-2 视频流是一个连续播放的视频帧序列,视频帧一共有 3 种编码类型,分别为 I 帧、P 帧和 B 帧.I 帧可以独立解码;而 P 帧和 B 帧必须参考相关的 I 帧和 P 帧才能解码.一定数量的帧形成一个视频片段 GOP(group of picture),每一个片段中的数据帧都按照固定的模式进行编码.如图 1 中 GOP 结构为 IBBPBBPBB.

在图 1 中,除第 1 帧(I 帧)采用帧内编码方式以外,其余帧都采用了帧间编码方式.为了压缩数据,目前的视频标准大都在帧间编码方式中使用了预测和运动补偿技术以减少空间和时间冗余信息.这些方法有效地压缩了数据,但同时也在前后帧之间形成了相互依赖关系.当优先级高的数据(比如 I 帧)丢失或发生错误时,容易造成一些相关的数据无法正常解码.尤其在网络上传输时,由于受带宽和丢包的影响,使得接收端播放质量大大降

低.文献[1,3]中对网络丢包给视频造成的影响有详细的描述:其中文献[3]的实验表明,当丢包率为 3%时,最多可以造成 1/3 的帧无法正常解码,所以应该通过纠错技术来提高网络视频数据传输的可靠性.

1.2 FEC信道编码

FEC 是一类用于有损信道通信的编码,通过增加冗余数据来提高抵御错误的能力,本文中的 FEC 使用了 RS(reed-solomon)编码.RS 编码是一种块编码,定义为 $RS(n,k)$,其中 n 为编码后的全部符号数量, k 表示原始信息符号的数量, $n-k$ 表示编码后产生的冗余信息符号的数量^[2].在视频流化过程中使用 RS 编码时,通常先将视频帧封装在若干个包中,然后用这些原始数据包生成冗余数据包后再传输.RS 编码最大可以纠正 $n-k$ 个错误,当经过网络传输后,如果有任意 k 个或更多的数据被正确接收,那么就可以从中恢复出原始的 k 个数据. k/n 称为编码率,表明了冗余数据的多少,很显然,当 k/n 越小时,纠错能力越强.RS 编码具有延迟短、纠错能力强的优点,已经被广泛用于网络通信.

当 n,k 确定时,就可以计算出全部数据的正确接收概率 $P(n,k)$,如公式(1)所示:

$$P(n,k) = \sum_{l=k}^n \binom{n}{l} (1-p)^l p^{n-l} \quad (1)$$

其中 p 为当前的平均丢包率.

1.3 已有工作

文献[4~10]中提出了一些在网络上传输视频数据时使用 FEC 来提高可靠性的方法.文献[4]将一个 GOP 的所有帧集中在一个数据块中使用 FEC 进行编码,在给定带宽的情况下,通过调节包的长度,给不同类型的帧分配不同的冗余带宽.这种方法对不同类型的帧给予不同的优先级,优先级越高,分配的冗余数据也越多.但是文献[4]中的方法没有考虑帧之间的依赖关系,而且对如何适应带宽变化也没有更多的研究.

文献[5,6]针对可扩展视频编码的传输进行了研究.文献[5]中的方法与文献[4]类似,只是不同的数据层有不同的优先级;而文献[6]中的方法主要是根据各层之间的依赖性来分配网络带宽,使视频数据传输过程中的平均丢包率最小.虽然文献[5,6]中的方法可以实现数据的分级保护,但是在如何适应网络带宽保持不变条件下进行的,对网络状态的变化适应性不强.文献[7]中的方法在传输时根据发送帧类型来附加冗余数据:不同类型的帧附加不同数量的冗余数据,而相同类型的帧附加等量的冗余数据.文献[7]中方法的优点是比较容易实现,但是它仅仅是根据帧的类型来改变冗余数据的分配,而且每种类型的帧分配的冗余数据是固定的,所以这种方法也不能动态适应网络状态的变化.文献[8]在 FEC 数据和 MPEG-2 源数据之间优化分配网络带宽,以提高解码的视觉效果,不过,文献[8]中的方法只是考虑了接收到的数据量与视频解码效果之间的关系,而对不同帧之间的相互关系涉及较少.文献[9,10]与本文中的工作有些相似,但是文献[10]中的模型对帧之间的依赖性分析不够充分;同时,文献[9,10]都对 GOP 的结构和冗余数据的分配方式做了一些假设,而本文中的方法不受这些条件的限制.

本文通过对视频结构的分析,提出了一种树形的 GOP 解码依赖模型,该模型很好地描述了帧之间的解码依赖关系和相对重要性.利用这个模型,发送端可以依据当前的网络状态,自适应地调节一个 GOP 中发送帧的数量,计算出最优的数据传输模式,以获得最大的可播放帧率;另外,为了保证算法对网络带宽资源使用的公平性,本文采用了 TCP 友好(TCP-friendly)的速率控制方法以便与基于 TCP 协议的应用相兼容.需要指出的是,虽然本文的研究主要以 MPEG 为例进行,但其结论对其他视频标准同样适用.

2 视频帧解码分析模型

根据前面的分析可知,在视频序列中,帧之间存在数据相关性.当发送端以一定的帧率发送视频数据时,受网络带宽的限制和丢包错误的影响,接收端正确解码的帧的数量小于发送的数量,即:接收端的可播放帧率低于发送端的发送帧率.由于 GOP 是视频序列中可以相对独立解码的视频片段,因此,如何提高接收端可播放帧率的研究也就变成了如何提高 GOP 中可解码帧的数量问题.

后面的讨论将根据 GOP 的结构特点结合 FEC 编码研究如何提高 GOP 中的可解码帧的数量.在本文的讨论中,视频流的 GOP 结构用参数 $(N_F, N_P, S_I, S_P, S_B)$ 来描述,其中 N_F, N_P 分别表示 GOP 中帧的数量和 P 帧个数,而

S_I, S_P 和 S_B 分别表示 I 帧、P 帧和 B 帧大小(以包计算);而网络状态用参数(R_{CUR}, t_{RTT}, p, S)来描述,其中 R_{CUR} 表示当前发送一个 GOP 的可用带宽(字节/秒), t_{RTT} 表示环路延迟时间, p 表示当前丢包率, S 表示数据包大小.

2.1 MPEG帧解码模型

第 1.1 节的内容表明,GOP 中视频帧根据类型的不同存在着不同的解码方式:I 帧可以独立解码;P 帧解码依赖前面的 I 帧或 P 帧;B 帧解码依赖前后两个参考帧.其中 B 帧根据其位置不同又分为两种情况:1) 若 B 帧位于 P 帧前面,其解码依赖于前后两个参考帧,而后一个 P 帧又依赖于前面的 I 帧或 P 帧,在这种情况下可以认为 B 帧的解码只依赖于后面的 P 帧;2) 若 B 帧位于 I 帧前面,其解码依赖当前 GOP 中的最后一个 P 帧和下一个 GOP 中的 I 帧.

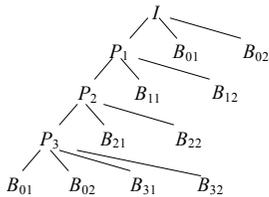


Fig.2 GOP dependency tree

图 2 GOP 依赖树

为了便于分析,下面对 GOP 中帧的依赖关系进行形式化描述:令 I_0 代表 I 帧; $P_i (1 \leq i \leq N_p)$ 表示 GOP 中的第 i 个 P 帧.对于 B 帧的描述略微复杂一些:通常 B 均匀在 I 帧和 P 帧之间,因此存在 $1 + N_p$ 个参考帧;不失一般性,令 $N_{B(i)}$ 表示依赖于第 i 个参考帧的 B 帧数量, B_{ij} 表示 GOP 中依赖于第 i 个参考帧的第 j 个 B 帧.例如:结构为 IBBPBBPBBPBB 的 GOP 可以表示为 $I_0 P_1 B_{11} B_{12} P_2 B_{21} B_{22} P_3 B_{31} B_{32} B_{01} B_{02}$.该 GOP 序列的解码关系依赖树如图 2 所示.图中 B_{01} 和 B_{02} 分别出现了两次,是因为它们位于 I 帧前面,其依赖关系跨越两个 GOP.这里,假设相邻两个 GOP 中的 I 帧具有相同的接收概率,因此将 B_{01}, B_{02} 的依赖关系简化在一个 GOP 中.图 2 中的树形结构很好地体现了 GOP 中帧之间的相互关系,即某一个结点的解码依赖其所有祖先结点的解码情况.

综合上述分析,GOP 中帧的解码关系形成了一棵以 I 帧为根的树,树中每个结点代表了一个 P 帧或 B 帧,子结点能否正确解码取决于其自身能否正确接收以及所有祖先结点的解码情况.树中任意结点的子孙结点数量代表了解码受该结点影响的帧的数目,子孙结点的数量越大,说明该结点越重要.显然,根结点的重要性最高,而叶子结点的重要性最低.

2.2 可用帧率的计算

依据 GOP 的解码依赖模型,首先分析计算当网络带宽资源充裕时,发送一个完整 GOP 中的可解码帧数量.假设一帧的接收概率为 $Pr(f)$,由公式(1)有:

$$Pr(f) = P(n(f), k(f)), f \in (I, P_i, B_{ij}) \tag{2}$$

其中 $n(f), k(f)$ 分别为帧 f 的经过 RS 编码后的全部数据包的数量和源数据包的数量.同时,假设 $f' \prec f$ 表示图 2 中结点间的依赖关系,其中 f' 为 f 的父结点.令 $E[f]$ 表示帧 f 的解码概率, $f \in (I, P_i, B_{ij})$,则 I 帧的解码概率为

$$E[I] = Pr(I), \tag{3}$$

而 P 帧的解码概率为

$$E[P_i] = E[P'] Pr(P_i) = E[I] * \prod_{j=1}^i Pr(P_j) = Pr(I) * \prod_{j=1}^i Pr(P_j) \tag{4}$$

其中 $P' \prec P_i (1 \leq i \leq N_p)$.

B 帧的解码概率根据其位置不同分两步计算:

1) 对于 P 帧之前的 B 帧 $B_{ij} (1 \leq i \leq N_p, 1 \leq j \leq N_{B(i)})$,其解码概率为

$$E[B_{ij}] = E[P_i] Pr(B_{ij}) = E[I] * \prod_{k=1}^i Pr(P_k) * Pr(B_{ij}) = Pr(I) * \prod_{k=1}^i Pr(P_k) * Pr(B_{ij}) \tag{5}$$

2) 对于 I 帧之前的 B 帧,其解码关系跨越了两个 GOP,分别依赖于前一个 GOP 中的最后一个 P 帧和后一个 GOP 中的 I 帧,故 $B_{ij} (i=0, 1 \leq j \leq N_{B(i)})$ 的解码概率为

$$E[B_{ij}] = E[I'] E[P_{N_p}] Pr(B_{ij}) = Pr[I] Pr(I') * \prod_{k=1}^{N_p} Pr(P_k) * Pr(B_{ij}) \tag{6}$$

上式中 I' 表示后一个 GOP 中的 I 帧,这里假设相邻两个 GOP 中的 I 帧其接收概率相同,所以公式(6)变成下面的形式:

$$E[B_{ij}] = Pr(I)^2 * \prod_{k=1}^{N_p} Pr(P_k) * Pr(B_{ij}) \quad (7)$$

综合式(3)~式(7),GOP 中全部可解帧数为

$$\begin{aligned} E[f_{gop}] &= E[I] + \sum_{i=1}^{N_p} E[P_i] + \sum_{i=1}^{N_p} \sum_{j=1}^{NB(i)} E[B_{ij}] + E[I]E[P_{N_p}] \sum_{j=1}^{NB(0)} E[B_{0j}] \\ &= E[I] + E[I] * \left(\sum_{i=1}^{N_p} \prod_{j=1}^i Pr(P_j) \right) + E[I] * \left(\sum_{i=1}^{N_p} \prod_{h=1}^i Pr(P_h) * \left(\sum_{j=1}^{NB(h)} Pr(B_{ij}) \right) \right) + \\ &\quad E[I]^2 * \left(\prod_{i=1}^N Pr(P_i) * \left(\sum_{j=1}^{NB(0)} Pr(B_{0j}) \right) \right) \\ &= Pr[I] + Pr[I] * \left(\sum_{i=1}^{N_p} \prod_{j=1}^i Pr(P_j) \right) + Pr[I] * \left(\sum_{i=1}^{N_p} \prod_{h=1}^i Pr(P_h) * \left(\sum_{j=1}^{NB(h)} Pr(B_{ij}) \right) \right) + \\ &\quad Pr[I]^2 * \left(\prod_{i=1}^N Pr(P_i) * \left(\sum_{j=1}^{NB(0)} Pr(B_{0j}) \right) \right) \end{aligned} \quad (8)$$

以上分析假设网络带宽足以传输整个 GOP 并能提供额外的冗余带宽.但是在实际情况中,网络状态的变化会导致带宽资源不足,这时候不允许发送整个 GOP 或者当发送整个 GOP 后冗余带宽提供的纠错能力不强.一种可行的方法就是对于一个 GOP 只发送部分帧,而将剩余的带宽用于传输更多的冗余数据以提高可靠性.因此,需要找一个合适的发送策略 $\pi(D, F)$ 使可解码帧数最大,其中 D 表示 GOP 中发送帧集合; F 代表对应于 D 中所有帧的 FEC 数据分配方案.假设 $fec(f)$ 是帧 f 的 FEC 数据,若 $f \notin D$, 则 $fec(f) = 0$.

2.3 解结构分析

根据第 2.2 节中的分析结果可知,寻找合适的发送策略 $\pi(D, F)$ 就是从解码依赖树的根出发,寻找一棵代表发送策略的子树,然后将剩余带宽作为冗余数据在子树的各结点间进行分配,使得在接收端,该子树的可解码帧数期望值最大.

对于寻找这样的一棵子树,可以通过穷举搜索来找出最优解.但是这样会加大运算量,而通过对 GOP 中解码依赖关系的分析,可以得到一些有用的结论而提高求解效率.

假设 $\pi_{opt}(D, F)$ 为最优解,则这个解的结构有如下几个性质:

性质 1. 对 GOP 任意一帧,只有其全部祖先结点都已经发送,该结点才能发送;否则,无法正常解码.

根据 GOP 的解码依赖关系,该结论显然成立.

性质 2. GOP 中只有前面的 B 帧全部发送完毕,后面的 B 帧才能开始发送,即:若 $B_{ij} \in D$, 则有 $B_{i'j'} \in D (1 \leq i' < i, 1 \leq j' < j, \text{ or } i' = i, 1 \leq j' < j)$.

证明:假设最优解 $\pi_{opt}(D, F)$ 的分配方案由图 3(a)所示, P_i 是 P_j 的父结点,即: $P_i \prec P_j$; 另外假设依赖于 P_i 和 P_j 解码的 B 帧数量均为 2. 由图 3(a)可知, P_j 的两个 B 帧都进行了分配,而 P_i 只分配了 B_{i1} 帧.

若按照 $\pi_{opt}(D, F)$ 来发送数据, GOP 中可解码帧数为

$$E[f_{gop}] = \sum_{f \in D} E[f].$$

如果将分配策略 $\pi_{opt}(D, F)$ 改动为 $\pi'_{opt}(D', F')$, 如图 3(b)中箭头所示,即不传输 B_{j2} 而传输 B_{i2} . 此时,可解码帧数变为: $E[f'_{gop}] = E[f_{gop}] + E[B_{i2}] - E[B_{j2}]$, 其中 $E[B_{i2}]$ 和 $E[B_{j2}]$ 分别为

$$E[B_{i2}] = E[P_i]Pr(B_{i2}), E[B_{j2}] = E[P_j]Pr(B_{j2}) = E[P_i]Pr(P_j)Pr(B_{j2}).$$

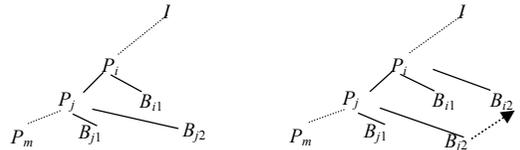


Fig.3 Optimal frame assignment
图 3 帧优化分配

显然, $E[B_{i2}] > E[B_{j2}]$, 即 $E[f'_{gop}] > E[f_{gop}]$, 而这与 $\pi_{opt}(D, F)$ 是最优解的初始假设相矛盾, 因此性质 2 可证. \square

性质 3. 当发送多个 P 帧时, 前面 P 帧的保护程度一定不小于后面的 P 帧, 即当 $P_i \in D, P_j \in D$ 且 $P_i \prec P_j$, 有 $fec(P_i) \geq fec(P_j)$.

证明: 与上面类似, 通过分析 P_i 和 P_j 的依赖关系, 使用反证法可以证明该结论, 这里不再给出详细的证明过程. 将此结论推广, 可得: 对 $P_i \in D (i=1, \dots, N_p-1)$, 若 $P_i \prec \dots \prec P_j \prec \dots \prec P_k$, 则 $fec(P_i) \geq \dots \geq fec(P_j) \geq \dots \geq fec(P_k)$. \square

性质 4. 若 D 为最优解发送帧集合, 则 D 中的帧一定是 GOP 依赖描述序列中从 I 帧开始的连续若干帧, 即: 若 $P_i \in D (1 \leq i < N_p)$ 或 $B_{ij} \in D (1 \leq i < N_p, 1 \leq j \leq NB_{(i)})$, 则 $P_k \in D (1 \leq k < i)$ and $B_{mn} \in D (1 \leq m < i, 1 \leq n \leq NB_{(i)} \text{ or } m=i, 1 \leq n < j)$.

这里不证明, 只举例说明. 以第 2.1 节中的 GOP 为例, 其依赖描述序列为 $I_0 P_1 B_{11} B_{12} P_2 B_{21} B_{22} P_3 B_{31} B_{32} B_{01} B_{02}$. 若 D 中包含 4 帧, 则发送帧集合一定是 $I_0 P_1 B_{11} B_{12}$, 而不是 $I_0 P_1 B_{11} P_2$. 通常在一个 GOP 中 B 帧的尺寸小于 P 帧, 因此可以通过分析两种情况下的可解码帧数得出该结论.

3 问题求解

综合第 2 节的讨论, 问题可以重新定义为: 当已知 GOP 结构和网络状态时, 选择合适的传输策略 $\pi(D, F)$, 使得:

$$\begin{cases} E[f_{gop}] = \max \sum_{f \in D} E[f] \\ \text{s.t. } BW(D, F) = \sum_{f \in D} Size[f] + \sum_{f \in D} fec[f] \leq \left\lfloor \frac{R_{CUR}}{S} \right\rfloor \end{cases} \quad (9)$$

其中 $Size(f)$ 代表帧 f 的大小, $BW(D, F)$ 代表当前分配策略使用的全部带宽.

公式(9)表述的是一个多元变量的优化问题. 使用穷举搜索可以找到正确答案, 但是将会消耗大量的时间, 而不能满足实时的要求, 因此需要另外的方法来进行计算. 这里给出一种快速的分配算法: 根据第 2.3 节的分析可知, 当丢包率和带宽确定时, 所发送的帧一定是按照其依赖描述序列顺序进行的, 因此可以先求出当前带宽所允许发送的各种帧组合的集合, 在每一个组合中对所有帧平均分配 FEC 带宽, 然后计算可解码帧数, 根据结果找出几种可能的发送帧组合; 最后根据第 2.3 节中的性质, 求出具体的分配结果. 对于一种组合, 搜索算法复杂度为

$$O\left(\left\lceil \frac{BW_{fec}}{N'} \right\rceil^{N'}\right),$$

其中 BW_{fec} 和 N' 分别为当前组合中的 FEC 带宽和帧数.

由于目前 Internet 上的应用大多是基于 TCP 协议的, 因此对于发送带宽 R_{CUR} 的计算可以采用文献[11]中的 TCP-Friendly 公式来进行计算, 以保证带宽资源使用的公平性:

$$T = \frac{s}{t_{RTT} \sqrt{\frac{2p}{3}} + t_{RTO} \left(3 \sqrt{\frac{3p}{8}} \right) p (1 + 32p^2)} \quad (\text{bytes/s}) \quad (10)$$

其中 t_{RTO} 为 TCP 重传超时阈值, 定为 $4 * t_{RTT}$; s 为包的大小. 根据帧的生成速度以及 GOP 的结构可以得到一个 GOP 允许使用的网络带宽. 在本文后面的实验中, 帧的生成速度为 30 帧/秒, 所以 $R_{CUR} = T / 2$.

4 仿真实验及分析

4.1 参数设置

本文仿真实验采用的 GOP 结构为 IBBPBBPBBPBBPBB, 其余所使用的参数分别为 $S_I, S_P, S_B, t_{RTT}, p, S$. 参照文献[9], 各参数的取值见表 1. 其中参数 p 的步长为 0.005. 硬件环境为 PIII667MHz, 256M 内存.

Table 1 Parameter settings

表 1 参数设置

Parameter	t_{RTT} (s)	S (bytes)	p	S_I	S_P	S_B
Value	0.05	1 500	0.01~0.05	20	12	3

4.2 实验结果及分析

为了对算法的性能进行分析,本节进行了相关的对比实验,另外研究了参数 t_{RTT} 和 S 的变化对算法性能的影响.在本节中,所有实验使用的时间平均大约为 150ms,因此该算法可以满足实时应用的要求.

图 4 是本文算法(adaptive FEC)与文献[7]中的静态分配冗余数据方法(static FEC)以及不使用 FEC(no FEC)的方法在相同条件下的实验结果.从图 4 的结果可以看出,本文的方法在相同的网络条件下可以显著提高 GOP 中的可解码帧数量.这说明本文的算法是有效的,能够根据当前的网络状态确定最优的数据发送模式.图 5 是图 4 实验中的 Adaptive FEC 方法在不同丢包率时,冗余数据在全部带宽中的比例.

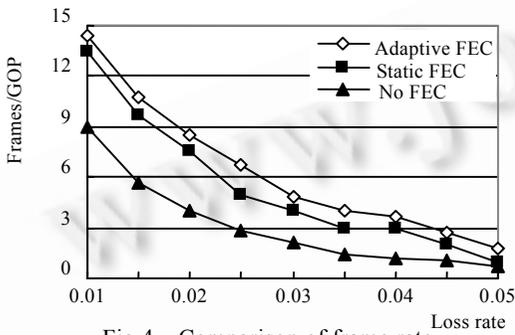


Fig.4 Comparison of frame rate
图 4 帧率比较

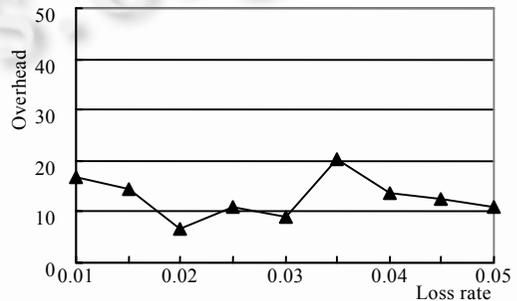


Fig.5 Comparison of FEC overhead
图 5 FEC 开销比较

图 6 的结果说明,参数 t_{RTT} 变化对算法的影响.在实际的应用中,除了丢包率 p 以外, t_{RTT} 也是动态变化的,根据文献[12]的测试结果,大部分的 t_{RTT} 分布在 0.035~0.055 之间.图 6 就是在此区间中以 0.005 为步长,丢包率 p 为 0.01s 时, t_{RTT} 变化对实验的影响.图 6 的结果表明,当 t_{RTT} 增大时,可解码帧数量会显著减少,这主要是由于 t_{RTT} 的增大导致有效带宽的减少造成的.

图 7 是包尺寸 S 的变化对可解码帧数量的影响,实验中数据包的长度分别为 800,100,1200,1400 字节,其余的条件与图 1 中的实验一样.从实验结果可以看出,数据包的大小对可解码帧率有一定的影响:当丢包率增大时,使用尺寸较小的包传输视频数据,可解码帧数相对较高,不过与其他结果相比差距不大.

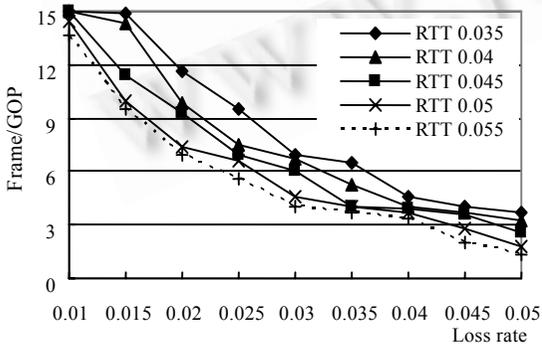


Fig.6 Frame rate vs RTT
图 6 帧率 vs RTT

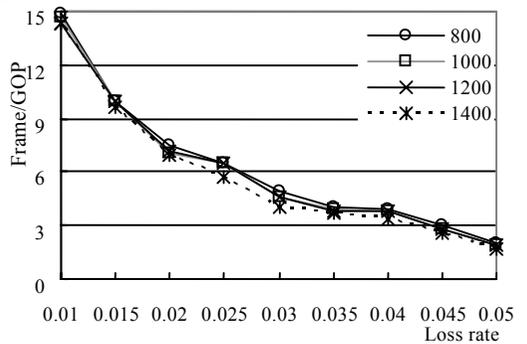


Fig.7 Frame rate vs packet length
图 7 帧率 vs 包长

5 结 论

本文通过对视频序列中帧解码关系的分析,建立了用于描述帧解码依赖关系和重要性的模型.在该模型基础上,提出了具体的带宽分配算法.实验证明,本文的算法可以找出最优的视频帧发送模式以及冗余数据的分配策略,使得接收端的可播放帧率最大.目前,本文中的工作主要是在平均丢包率的情况下进行的,没有考虑突发错误长度的影响,也没有使用重传技术,在以后的研究中将继续这一方面的工作.

References:

- [1] Feamster N, Balakrishnan H. Packet loss recovery for streaming video. In: Proc. of the 12th Int'l Packet Video Workshop. 2002. http://amp.ece.cmu.edu/packetvideo2002/final_program.htm
- [2] Johanson M. Adaptive forward error correction for real-time internet video. In: Proc. of the 13th Packet Video Workshop. 2003. <http://www.polytech.univ-nantes.fr/pv2003/papers/pv/papers/cr1001.pdf>
- [3] Boyce J, Gaglianella R. Packet loss effects on MPEG video sent over the public internet. In: Proc. of the 6th ACM Int'l Conf. on Multimedia. New York: ACM Press, 1998.181~190.
- [4] Storn R. Modeling and optimization of PET-redundancy assignment for MPEG sequences. Technical Report, TR-95-018, Berkeley: ICSI, 1995. <ftp://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95-018.pdf>
- [5] Horn U, Stuhlmuller KW, Link M, Girod B. Robust internet video transmission based on scalable coding and unequal error protection. Image Communication, Special Issue on Real-time Video over the Internet, 1999,15(1-2):77~94.
- [6] Zhang T, Xu Y. Unequal packet loss protection for layered video transmission. IEEE Trans. on Broadcasting, 1999,45(2):243~252.
- [7] Hartanto F, Sirisena HR. Hybrid error control mechanism for video transmission in the wireless IP networks. In: Proc. of the 10th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN) 1999. 1999. <http://www.fokus.gmd.de/research/cc/glone/employees/felix.hartanto/pubs/lanman99.pdf>
- [8] Frossard P, Verscheure O. Content-Based MPEG-2 structuring and protection. In: Tescher AG, Vasudev B, Bove V, Derryberry B. eds. SPIE Int'l Symp. on Voice, Video, and Data Communications. 1999.
- [9] Mayer-Patel K, Le L, Carle G. An MPEG performance model and its application to adaptive forward error correction. In: Proc. of ACM Multimedia. New York: ACM Press, 2002. 1~10.
- [10] Wu H, Claypool M, Kinicki R. A model for MPEG with forward error correction and TCP-friendly bandwidth. In: Proc. of the Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV). New York: ACM Press, 2003. 122~130.
- [11] Padhye J, Firoiu V, Towsley D, Kurose J. Modeling tcp throughput: A simple model and its empirical validation. In: Proc. of the ACM SIGCOMM. New York: ACM Press, 1998. 303~314.
- [12] Loguinov D, Radha H. Measurement study of low-bitrate internet video streaming. In: Proc. of the ACM SIGCOMM Internet Measurement Workshop (IMW). New York: ACM Press, 2001. 281~293.