

AVS 视频的分层重传优化策略与实现

吴佳莉^{1,2}, 李枚毅¹, 刘东华²

(1. 湘潭大学信息工程学院, 湘潭 411105; 2. 上海中科计算技术研究所, 上海 201203)

摘要: 针对网络抖动等因素会影响图像质量的问题, 采用差错控制与拥塞控制相结合的恢复技术, 改进 AVS 视频数据在传输层的 RTP 打包, 提出提高重传效率的 AVS 视频分层重传策略, 通过令牌发放的方式控制数据包重传的次數解决重传导致网络拥塞问题。实验结果表明, 在低带宽的网络下, 该策略对 AVS 视频传输相对于无令牌发放的重传方式图像质量有一定的提高。

关键词: 差错控制; AVS 视频; 分层重传; RTP 打包

Optimized Strategy and Implementation of AVS Video Layered Re-transmission

WU Jia-li^{1,2}, LI Mei-yi¹, LIU Dong-hua²

(1. College of Information Engineering, Xiangtan University, Xiangtan 411105;

2. Shanghai Institute of Computing Technology, Chinese Academy of Sciences, Shanghai 201203)

【Abstract】 Aiming at AVS video stream transmission on the Internet becomes a new focus. However, factors such as network jitter will bound to affect image quality. In this paper, the restoration technology of the error control and congestion control are combined to be produced. The technology of AVS video data RTP packing is improved in the transmission layer. The strategy of AVS video layered transmission is proposed to improve the efficiency of re-transmission and the way of token payment which can control the time of data packets re-transmission is used to solve the network congestion problem led by re-transmission. Experimental results show that: in the low-bandwidth network, this policy for AVS video transmission has a better image quality than that of delivering without token.

【Key words】 error control; AVS video; layered re-transmission; RTP packing

1 概述

AVS 作为我国自主技术的音视频编码标准, 相对于其他的信源标准具有其明显的优越性, 特别是编码效率高, 实现复杂度相对较低, 专利费用低等特点将有力的促进流媒体的发展^[1-2]。Internet 是一种“尽力而为”的传输网络, 网络状态会发生实时变化, 包括端到端的可用带宽, 网络丢包率等, 对实时视频业务的 QoS 会产生重大的影响, 为了提高 QoS, 采用实时适应网络状态变化的策略。

目前使用的策略有很多种, 围绕重传技术提高视频传输的 QoS 方案已经得到了大量的讨论和验证, 文献[3]提出了结合视频缓冲策略和重传技术的方法, 此外也有学者提出了根据视频帧类型来改善视频重传效果的策略^[4], 但是重传会导致网络拥塞, 这些方法只是考虑了重传策略, 却忽略了网络的拥塞控制, 但是这 2 种机制的结合可以改善总体性能。本文基于现有的组包算法, 改进了 AVS 视频 RTP 组包算法, 增加了有效的差错控制, 对传输的拥塞问题进行了一定的研究, 并提出了解决方案。

2 视频图像质量分析

在 AVS 压缩的视频流中, 帧内帧(如 P 帧和 B 帧)的解码还取决于与其相关的帧, 如 P 帧的解码还需要与其对应 I 帧的信息, 因此, 若在某个帧出现丢包情况, 就会导致该帧解码出错, 视频质量下降, 甚至会传递给相关的帧, 导致视频图像无法观看。比如, I 帧出错, 会影响到与之相关的 P 帧

和 B 帧解码。针对这种情况, 采用以下分析方式: 首先对经过网络传输后的视频数据进行解码, 解码后的图像质量通常采用峰值信噪比来表示其与原图的差别。当网络出现丢包时, 解码后的图像会与原图出现不同, PSNR 将随之变小, 而且是丢包越多, PSNR 也越小。

文献[3]中采用的用多条视频流(标准的 MPEG4 视频流)来做实验的结果, 表示丢包率与 PSNR 的关系。根据分析得出: 丢包越多, 图像质量也就越低。通常 PSNR 要大于 20 dB, 视频图像质量才认为可以接受, 即丢包率需要小于 1/250。

3 AVS 视频流内容重要性分析

AVS 比特流中采用的是分层结构, 高层包含低层的, 其由高到低的顺序依次为: 序列, 图像, 条带, 宏块和块。序列是比特流的最高层, 包括一个或多个连续的编码图像。一个视频序列的开始部分包含了视频序列头部, 视频对象头部, 视频对象层头部等配置信息, 这些信息对于后续视频流的解

基金项目: 国家自然科学基金资助项目(60773047); 湖南省自然科学基金资助项目(06JJ50106); 湘潭大学博士基金资助项目(05QDZ23); 上海市博士后科研计划基金资助项目(05R214207); 上海市青年科技启明星计划基金资助项目(05QMX1447)

作者简介: 吴佳莉(1984 -), 女, 硕士, 主研方向: 网络传输, 嵌入式系统, 网络智能化技术; 李枚毅, 教授、博士; 刘东华, 讲师、博士后

收稿日期: 2009-04-06 **E-mail:** wujiali84@126.com

码是非常重要的,因此在传输时应该保证这些信息可靠到达。

一个视频序列由多个图像组成,图像根据编码方式的不同分为3种类型:I帧,P帧和B帧。对于视频序列的第一帧及场景切换帧或者随机读取帧采用I帧(初始帧)编码方式。I帧也是极其重要的,若在传输中丢失,会影响视频图像的质量,不但影响自己解码,还会影响到后继P帧和B帧的解码,对应的情况就是产生花屏,所以有必要保证I帧传输的可靠性。对其余帧采用帧间编码方式,包括P帧(前向预测帧)和B帧(双向预测帧)。其中P帧是帧间预测中只使用前向预测解码的图像,即显示顺序上过去的参考图像对当前图像进行预测。B帧是帧间预测中使用双向预测解码的图像,即显示顺序上过去和将来的参考图像对当前图像进行预测。由于B帧的参考帧是双向的,其丢失后重构的可能性比P帧要高,因此,这两者应尽量保证P帧能够到达。

每个图像都是由一个个单独的条带组成的。条带是按光栅顺序的若干连续宏块行,每个条带中的宏块只能依靠本条带中的宏块进行解码,不参考其他条带中的宏块,所以条带与条带之间是没有联系的,相当于独立的个体。某条带丢失后,对其余条带是没有影响的,也就是某条带不依赖于前后条带进行解码的,它可根据自己的头部信息进行解码,其丢失相对于序列头和I帧、P帧、B帧来说,恢复得可能性更大,因此,它的优先级要低一些。

4 AVS 比特流的打包

AVS 传输流是由视频 PES,音频 PES 与其他辅助数据复用而成的。因此,要通过 RTP 传输 AVS 视频流,需要对其进行打包处理,再加上时间,同步等信息。通常使用的 RTP 对多媒体数据格式打包的方法有固定的包长来封装视频流,尽管此打包策略非常简单,但其有可能将一个宏块拆分打进2个 RTP 包中,增加包间相关性。也有的将一个宏块打入一个 RTP 包,改进了这个缺点,但是效率不高,因为使用较大的包长才能降低所需传输的总包数,减小系统开销。文献[4]中提出了 AVS-M 的打包方式,但是他对很重要的信息没有单独打包,导致如果丢失,对视频的影响将会非常巨大。AVS 第8部分规定了如何在 IP 网络上传输 AVS 定义了 AVS 视频的 RTP 净载格式^[5]。为了结合 AVS 的编码特性又可以提高网络传输效率,本文改进了原有的打包方式,采用了一种简单又方便的打包机制。

本文采用的基本数据流打包方法以帧为单位,即对于视频,每帧(I帧,P帧或B帧)编码数据作为一个 RTP 分组包,序列头单独打包。

为了不造成 IP 碎片,包长还不能超过该网络路径的 MTU(Maximum Transit Unit),考虑到传输的高效性和数据包传输的鲁棒性,取包长为当前帧大小与路径 MTU 值的较小值,采用如下组包机制:

(1)若当前帧能放入单个 RTP 包,就把此帧单独放入一个 RTP 包中。

(2)若有些帧映射到 RTP 包中时,导致访问单元超过 PATH-MTU,引起下层数据的分包,不利于 RTP 处理,此时也允许将过大的帧分割并映射到不同的 RTP 包中,采用条带作为包的分割点,分割后,一个 RTP 载荷是完整的一个或多个由条带逻辑划分的单元,由于条带与条带之间没有任何的关联性,并且每个条带的开始部分包含了条带头和起始码,这样分割后的 RTP 包可以独立解码,即使某个包丢失,也不

会影响后面包的解码,但必须把帧头部信息以及条带头复制到每个 RTP 包中;

(3)若当前帧太小,此时为了提高 RTP 传输的有效性(减少包数和降低开销),可以把连续的几个帧按照解码的顺序放入一个 RTP 包中,并且第一个帧后的帧时间戳在解码时通过计算得到,此时 RTP 包头标志位位置 1 且 RTP 负载开始符合起始码。

组包流程如图 1 所示。

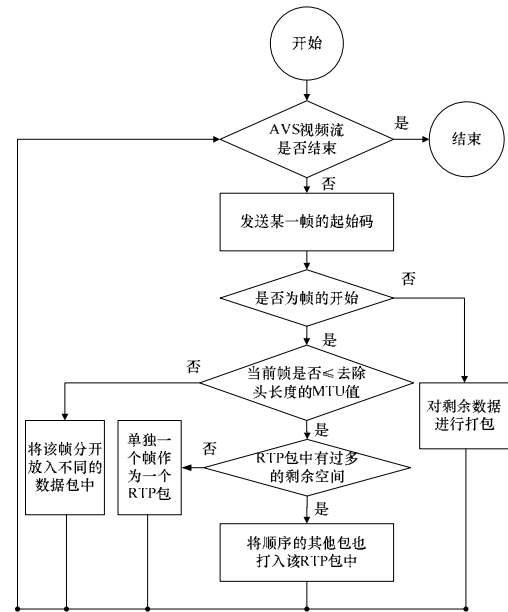


图 1 组包算法流程

该打包策略保证了存在丢包时重传能够高效率的进行:在低带宽环境下,由于带宽资源的有限,在条件允许重传时,一方面需要保证其按照信息的重要程度来进行,另一方面对于重要的信息不能占用太多的带宽资源,否则会影响正常顺序视频流的发送。

5 RTP 传输 AVS 视频流的恢复策略

在低带宽和丢包率比较高的网络环境下传输 AVS 视频流 RTP 包时,目的就是最大程度地利用现有带宽资源,按照主要程度的高低来使各部分信息到达接收方;同时利用限时重传策略来提高重建视频图像质量。

根据上文的分析:序列头是最重要的数据,如果其丢失将会使得所有的视频数据变得毫无意义,除此之外,I帧也是极其重要的,所以有必要保证I帧传输的可靠性。

针对这种情况,本文采取了重传的机制,在丢失的情况下重传数据包,保证序列头尽量不被丢失。在服务器端建立不同的优先级重传缓冲区,根据 RTP 包的重要性的不同,可以将包分为4个优先级别,含有序列头的数据包为最高重传优先级,含I帧帧头的包具有次重传优先级,含P帧帧头和B帧帧头的包为第三重传优先级,含有条带头的包为第四重传优先级。当没有足够的冗余带宽重传所有丢包时,系统按照不同的优先级来传输这些数据信息。

具体的算法思想是这样的:为了保证传输的可靠性,采用一种分层重传的 QOS 控制机制来尽量保证序列头和 I 帧传输的可靠性,接收方在接收 RTP 包时检测是否有包丢失,若第 N+3 号 RTP 包到来时,第 N 号包还未收到,认为第 N 号包已经丢失,这时接收端等待返回请求,然后对丢失的包进

行分层重传。首先检查数据包的类型，如果是包含序列头的数据包，则立即向发送方发送该数据包，仍然没有收到，继续发送直到收到为止；如果接收方接收请求是 I 帧，立即发送 APP 包，这个包中包含请求重发的数据包顺序号和 I 帧所在数据包的具体位置，发送方接收到重发请求时，若下一个 I 帧还没有生成，则立即重发这个数据包。如果不是最高优先级和次高优先级的请求，根据现有可用带宽而决定是否重传所请求的信息。如果可以利用现有可用带宽重传所请求的 RTP 包而不影响正常视频流的发送，则响应重传并优先发送请求重传 RTP 包；否则不予响应而继续进行正常视频流的服务。具体算法如下：

```
if(第 N+3 个包到来, 第 N 个包还没有收到)
```

```
{
  第 N 个包已经丢失;
}
```

```
else
```

```
{
  检查这是个什么包;
```

```
{
  if(包含序列头的数据包)
```

```
{
  do
```

```
  立即重传该数据包;
```

```
  while(收到)
```

```
}
  if(包含 I 帧的数据包)
```

```
{
  立即发送 APP 包;
```

```
  接收方检查下一个 I 帧是否生成了;
```

```
  if(还没有生成)
```

```
    重发该数据包;
```

```
  if(没有收到)
```

```
    检查网络状况;
```

```
  if(带宽允许)
```

```
    响应重传;
```

```
}
  if(是其他重传优先级数据包)
```

```
{
  if(现有带宽允许)
```

```
    {
  立即发送 APP 包;
```

```
    重传该数据包;
```

```
    }
```

```
  else
```

```
  {
```

```
    不响应重传, 进行正常视频服务;
```

```
  }
```

```
}
```

```
}
```

为了防止重传带来的网络拥塞，接收方采用发放令牌的办法控制重发请求的数量，设置最大可持有的令牌数 LP_{MAX} ，设置丢包率的一个阈值 L_{OPT} ，可以根据这个阈值来衡量丢包率的大小，从而判断网络状况的好坏。接收方每隔一段时间检查准备发送的 RR 包(RTCP 包的一种接收方报告)中的 RTP 包的丢包率 L 。当丢包率 L 小于阈值 L_{OPT} 则增加一个令牌，若大于阈值 L_{MAX} 则令牌数置零，发送方每发送一个请求重发的 APP 包(运用于特定功能的包)消耗一个令牌，然后根据令牌的数目去响应重传。用这样的方式就可以根据网络的状况动态的调节发送请求的次数。当令牌数小于或者等于零，则

说明现有带宽状态不佳，那么对于第三优先级和第四优先级的 RTP 包则不响应重传，而进行正常的视频流服务；如果令牌数大于零，则响应重传，但是同时消耗一个令牌数。对于第一优先级的 RTP 包一直重传已经设置的次数，不受带宽影响。第二优先级的 RTP 包，如果时间期限没有超过，即下一个该优先级的数据包还没有生成之前立即重发，不受网络带宽影响。

该策略一方面保证了在重传时间允许的条件下解码需要的重要信息能按重要程度的高低先后可靠地到达接收方，又考虑到了带宽的状态，从而不造成网络的拥塞，从而拥有更好的图像质量。

6 策略的实现及实验结果

本策略在局域网上搭建平台进行测试，平台分为两个工程，客户端程序和服务器端程序，发送端发送视频流的流程如下：(1)读取 AVS 视频流，根据 RTP 协议头部格式对正常视频流进行打包并发送，开启监听线程以接收有可能到达的重传请求报文。(2)若有重传请求报文到达，首先检查数据包类型，然后根据现有可用带宽来决定是否重传请求报文，若没有丢包，进行正常的视频服务。

发送流程如图 2 所示。

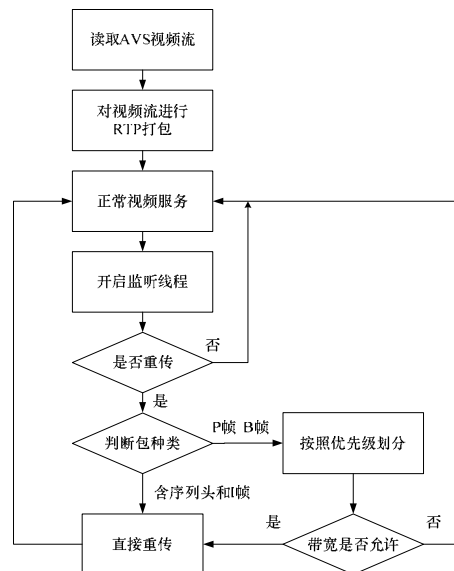


图 2 发送流程

接收端对于丢失的数据包，采取请求重发的差错控制策略。接收视频流时，同时开启视频流接受服务线程和丢失检测线程。服务器端接收到重传信息，判断时间期限是否超过，若没有，则重发客户端所需数据包。

本实验在网络带宽很低的情况下进行的。选择在带宽为 32 Kb/s 的环境下，针对 3 种策略对 2 个 AVS 压缩视频流进行测试。这 3 种策略分别为：本文所提出的有令牌分层重传策略，通常使用的无令牌重传策略以及无重传策略，测试结果如表 1 所示。

表 1 3 种策略的 PSNR 对比实验结果

视频流	帧数	丢包率/(%)	PSNR/dB (无重传)	PSNR/dB (无令牌重传)	PSNR/dB (有令牌重传)
Foreman	135	2.8	25.17	27.20	31.21
		9.8	21.35	25.45	30.96
Paris	170	3.0	29.05	32.09	39.85
		10.2	26.02	29.88	39.52

(下转第 178 页)