

# 基于实时流媒体传输系统的 H. 264 组包算法研究<sup>\*</sup>

魏聪颖 牛建伟 吉海星 胡建平

(北京航空航天大学计算机学院 北京 100083)

**摘要** 有线和无线网络上的实时流媒体传输技术已成为研究热点而其中组包策略的选取是其中的一个关键问题。本文首先设计了一种基于 RTP 的 H. 264 实时流媒体传输系统模型,在此基础上提出了适用于 H. 264 实时视频流传输的 RTP 组包算法 HMP。该算法针对 H. 264 NAL 视频流特点,在考虑到视频流关联性的同时加入了对重要信息的保护。实验结果证明,HMP 算法在网络丢包率较大的情况下仍能获得良好视觉质量。

**关键词** H. 264 NAL, RTP, 实时流媒体传输系统, 组包算法

## H. 264 Packetization Algorithm Research Based on Real-time Stream Media Transportation System

WEI Cong-Ying NIU Jian-Wei JI Hai-Xing HU Jian-Ping

(School of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083)

**Abstract** Real-time stream media transportation over wired/wireless network has become a focus of research. And the packetization algorithm is the key technology. This paper describes a H. 264 real-time transmission module based on RTP, and puts forward a RTP packetization algorithm for H. 264 video which is called Hybrid Mode Packetization (HMP). HMP considers video stream content correlation and unequal protection of important information with H. 264 NAL video features. Simulation results show that HMP can achieve good perceptual picture quality for H. 264 video under packet-loss network conditions.

**Keywords** H. 264 NAL, RTP, Real-time stream media transportation, Packetization

## 1 引言

随着 Internet 和多媒体技术的快速发展,网络视频的实时传输成为网络应用的热点之一。H. 264 作为新兴的编解码技术带来了技术上的改进,在同等压缩质量情况下,压缩输出码率大约为原来 MPEG-4 的二分之一。加之网络带宽的进一步提升,手机电视(Mobile TV)、数字电视(IPTV)、可视电话、网络游戏等宽带增值业务,将成为新的收益增长点 and 切入点。因此,研究基于 H. 264 的实时传输技术具有十分重要的现实意义和应用价值。视频的实时传输要求较低的时延和较小的丢包率,由于 TCP(Transmission Control Protocol)协议的重发机制带来较大的时延,UDP(User Datagram Protocol)协议本身又不提供任何服务质量(Quality of Service, QoS)保证,因此需要新的协议来满足网络视频实时传输的时延和丢包要求。国际标准化组织 IETF(Internet Engineering Task Force)制定了实时传输协议(Real time Transport Protocol, RTP)<sup>[1]</sup>来满足多媒体实时传输的需求。

要通过 RTP 进行 H. 264 视频传输,必须把 H. 264 视频数据封装成一个个 RTP 包。文[2~4]分别提出了 H. 261, H. 263, H. 263+ 视频流的 RTP 载荷格式及组包算法,但它们都是针对基于块的编码标准,不适用于 H. 264 所产生的视频流。文[5]提出了 H. 264 视频流的 RTP 载荷格式和组包规则但没有提出具体的组包算法。本文针对 H. 264 网络适配层(Network Abstraction Layer, NAL)所产生的视频流,提出了一种基于 RTP 的实时 H. 264 视频流的混合模式组包算

法(Hybrid Mode Packetization, HMP)。实验证明该算法在网络丢包率较大的情况下仍能获得良好视觉质量。

## 2 实时流媒体传输系统模型

### 2.1 传输系统模型

图 1 为本文提出的基于 RTP 的 H. 264 实时流媒体传输系统模型。如图 1 所示,通过摄像头实时采集的视频流采用 H. 264 视频编码标准进行压缩,H. 264 定义了视频编码层(Video Coding Layer, VCL)以及网络提取层(Network Abstraction Layer, NAL),以便能够适应不同的网络结构。VCL 中包括 VCL 编码器与 VCL 解码器,主要功能是视频数据压缩编码和解码;NAL 则用于为 VCL 提供一个与网络无关的统一接口,它负责对视频数据进行封装,经过封装的数据包在传送过程中具有更好的健壮性。编码器接口产生的 H. 264 视频流分别被封装上 RTP 包头、UDP 包头和 IP 包头,然后 IP 数据包通过有线、无线传输网络传输给接收端。接收端收到 IP 包后按相反的顺序将 RTP 包头和视频数据提取出来,根据 RTP 包头中的序列号进行排序后将视频数据通过解码器接口传入解码器,解码器对数据进行解码后进行播放,在解码的过程中可能还需要进行容错处理(如果有数据丢失)。

### 2.2 H. 264 NAL 视频流格式

NAL 层<sup>[6]</sup>产生的一个编码视频序列,由一串连续的存储单元组成,使用同一序列参数集,见图 2。每个视频序列可独立解码。编码序列的开始是即时刷新存储单元(IDR)。IDR 是一个 I 帧图像,表示后面的图像不用参考以前的图像。一

<sup>\*</sup> 基金项目:论文得到国家自然科学基金资助,基金编号:90612004,项目名称:个人化流动科研环境关键问题研究。魏聪颖 硕士,主要研究方向:无线环境下实时流媒体传输及视频编解码;牛建伟 副教授,硕士生导师;吉海星 硕士研究生;胡建平 教授,博士生导师。

个 NAL 单元流可包含一个或更多的编码视频序列。

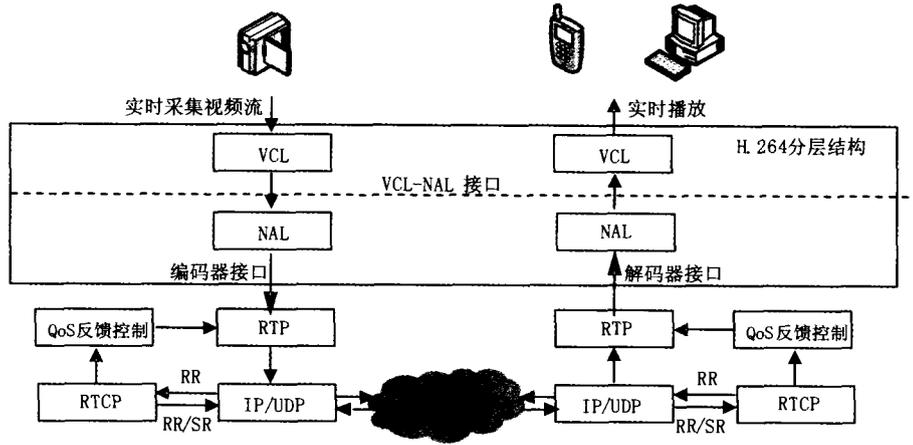


图 1 基于 RTP 的 H.264 传输模型



图 2 H.264 NAL 视频流序列

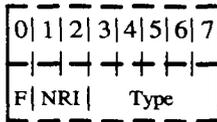


图 3 NALU 结构

NAL 层作为 H.264 与网络的接口,规定了视频序列的格式,NAL 单元(NALU)是 NAL 层的基本处理单元,一个 NALU 是一个包含一定语法元素的可变长字节字符串,它可以携带一个编码片,或者一个序列或图像参数集。每个 NALU 由一个字节的头和一个包含可变长编码符号的字节组成。头部含三个定长比特区:NALU 类型(T),重要性指示(R)和禁止位(F)。NALU 类型用 5bit 来代表 NALU 的 32 种不同类型。R 比特用于标志在重构过程中的重要性,值为 0 表示没有用于预测参考,值越大,用于预测参考的次数越

多。F 比特默认为 0,当网络检测到 NALU 中存在比特错误(在无线网络环境易出现)时,可将其置为 1,主要适用于异质网络环境(如有线无线相结合的环境),见图 3。

### 3 H.264 视频流传输控制算法

#### 3.1 H.264 视频流的组包模式

要通过 RTP 传输 H.264 视频流,需要对其进行打包(Packetization),加入时间戳、序列号等信息。文[5]对把 H.264 封装成 RTP 包提出了三种组包模式:

单一模式(Single NAL unit mode),即把一个 NALU 打入一个 RTP 包,主要适用于对话(conversation)应用系统;

非交错模式(Non-interleaved mode),按照编码出来的视频流的顺序进行组包,主要适用于低延迟的实时系统;

交错模式(Interleaved mode),打乱编码出来的视频流的顺序进行组包,主要适用于对延迟要求较低的系统。

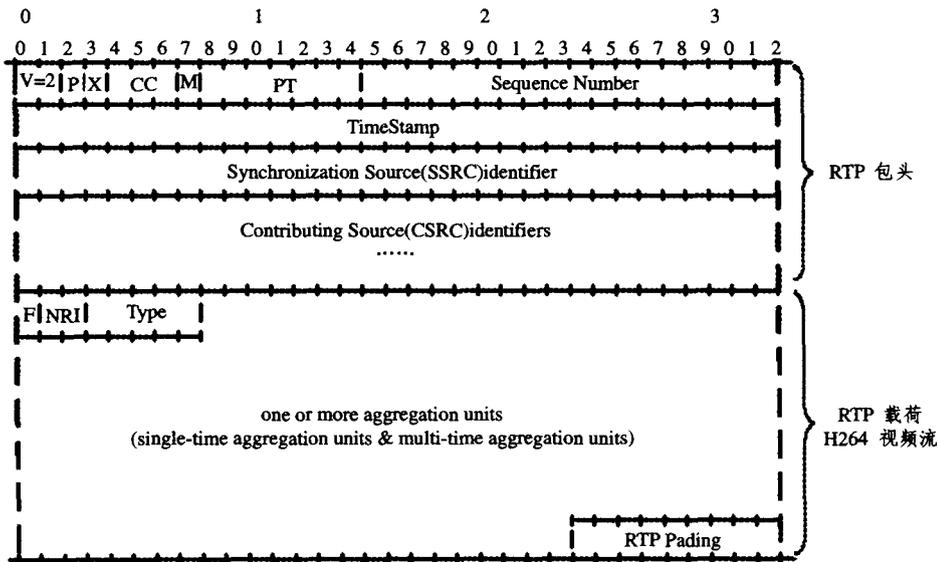


图 4 H.264 视频流的组包模式

本文提出的传输模型中视频流是通过实时采集、传输并实时解码播放的,初始化时延要求在 10s 以内。根据本系统的要求,本文的组包模式采用了非交错模式和交错模式结合的方法,对视频流按照解码出来的顺序进行组包,如图 4。

### 3.2 混合模式组包算法 (Hybrid Mode Packetization, HMP)

文[7]提出 FLP(Fixed-packet-length Packetization)组包算法用固定的包长来封装视频流,尽管此打包策略非常简单,但它有可能使一个 NALU 被拆分打进两个 RTP 包,增加包间相关性,因为只有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。同时为了不造成 IP 碎片,包长还须有个限制,即不能超过该网络路径的 MTU(Maximum Transit Unit)。MTU 是指数据包不需要合并或者分割能够在网络层以及应用层传输最大比特数。本文在考虑以上各方面的情况下提出了混合模式组包算法 (Hybrid Mode Packetization, HMP),在满足 MTU 对包长限制的同时最小化包间相关性。HMP 算法描述如下。

若当前帧组 (Group of Picture, GOP) 能放入单个 RTP 包,就把此 GOP 单独放入一个 RTP 包中;若放不下,则把 GOP 分成多个帧 (Picture),放入多个 RTP 包,此时须把 GOP 的序列参数集 (Sequence parameter set) 复制到每个 RTP 包,以去除包间的相关性,达到丢包的鲁棒性;为减小包数,降低开销,一包中能放入多少 Picture 就尽可能多地放入多少 Picture,但即使最后一个包中仍有剩余空间,也不能把另一 GOP 中的 Picture 放入此包中;若当前 RTP 包放不下一个 Picture,则把 Picture 分成多个切片 (Slice),放入多个 RTP 包,此时需把 GOP 的 Sequence parameter set 和 Picture 的图像参数集 (Picture parameter set) 复制到每个 RTP 包,以去除包间的相关性,达到丢包的鲁棒性。为减小包数,降低开销,一包中能放入多少 Slice 就尽可能多地放入多少 Slice,但即使最后一个包中仍有剩余空间,也不能把另一 Frame 中的 Slice 放入此包中。

```

Begin;
设 PS 为去除头部字段长度的 MTU 值;
While (! 遇到 NAL 视频流 End of Stream 标记){
If (发现当前 GOP 的 End of Sequence 标记){
If (当前分段长度 ≤ PS){
采用 Non-interleaved Mode;
把此段数据打入一个 RTP 包;}
else {

```

```

回到当前 GOP 开始处;
While(! 当前 GOP 的 End of Sequence 标记){
if(发现下一个 Picture 的起始码){
If(当前分段长度 ≤ PS){
当前分段长度 = 0;
while(当前分段长度 ≤ PS){
发现当前 Picture 的 Picture delimiter;
当前分段长度 += 当前 Picture 长度;
搜索下一个 Picture 起始码;}
if(此段包含 Picture 数目 = 1){
采用 Non-interleaved Mode;
把此段数据打入一个 RTP 包;}
else{
采用 Interleaved Mode;
把此段数据打入一个 RTP 包;}}
else{
回到当前 Picture 开始处;
While(! 当前 Picture 的 Picture delimiter 标记){
当前分段长度 = 0;
while(当前分段长度 ≤ PS && ! Picture delimiter){
发现当前 Slice 的结束符;
当前分段长度 += 当前 Slice 长度;
搜索下一个 Slice 起始码;}
采用 Non-interleaved Mode;
把此段数据打入一个 RTP 包;
}}}}}}
End

```

### 4 实验结果

实验的编解码采用的是 TML 颁布的软件包编解码器测试程序 JM95,选择 QCIF 分辨率的 Foreman 视频序列和 CIF 分辨率的 News 视频序列作为测试序列,视频序列 YUV 格式为 4:2:0, GOP 大小 N=100, I/P 间距离 M=10。帧率为 10fps,帧数为 30 帧。

仿真时的一些参数设置如下:

- (1) 1/4 像素精度的运动估计技术;
- (2) 运动补偿均基于宏块;
- (3) 仅仅采用一个单独的参考帧预测,而没有采用多帧预测。

针对同一视频序列在不同比特率及网络丢包率情况下采用不同组包算法的 PSNR 值进行比较来对本文提出的 HMP 算法的性能进行评估。对丢失的包采用最简单的错误隐藏方法,即用前一正确收到的帧来替代当前丢失的帧。表 1 显示了在 3%、5%、10% 不同网络丢包率情况下,对变化的码流测试视频序列采用不同组包算法所达到的客观视觉效果进行比较的结果。从表 1 中可以看出, HMP 组包算法优于 FLP 组包算法,峰值信噪比 (PSNR 值) 大约提高 0.01~10dB。

表 1 HMP 组包算法与 FLP 组包算法实验结果比较

视频序列	分辨率	QP	比特率 (kbits/s)	原 PSNR	组包算法	不同丢包率下的 PSNR		
						3%	5%	10%
Foreman	176 * 144	24	62.98	38.52	FLP	33.19	33.01	19.39
					HMP	34.14	33.98	27.15
		20	128.37	41.60	FLP	36.12	34.87	30.05
					HMP	37.28	35.06	34.96
		16	240.93	44.95	FLP	36.96	35.12	32.26
					HMP	38.49	36.29	35.09
News	352 * 288	24	120.88	40.83	FLP	35.61	35.35	24.05
					HMP	35.69	35.57	34.17
		20	200.00	43.28	FLP	37.54	35.75	31.21
					HMP	38.08	36.98	34.66
		16	362.80	45.82	FLP	39.96	36.02	31.30
					HMP	40.73	37.24	34.96

### (1)网络丢包率不同的情况

本文通过改变丢包率来对所提出的算法的性能进行验证,选择的丢包率参数分别为3%、5%、10%。随着网络丢包率的增大,必将导致视频测试序列的图像质量的下降。从表1中可以看出,采用HMP算法后视频测试序列的PSNR值相较FLP算法有所提高,并且随着丢包率的增大,这种性能上的差别就更加显著。这主要是由于HMP在打包时尽量使每个包包含一个相对独立的内容,减少包间的相关性,把由于前一包的丢失而对后续包所带来的影响减到最低,同时该算法还加入了对重要信息的保护,不会出现由于重要信息的丢失而造成收到数据也不能解码的情况。

### (2)视频流比特率不同的情况

本文通过改变视频测试序列的量化参数(QP)进而改变视频测试序列的比特率,针对相同视频测试序列,本文选择的QP参数大小分别为24、20、16。从表1中可以看出,采用HMP算法可以很好地适应各种码流,这主要是由于当QP相对较小时,该算法组包采用以切片为单位,并尽量放入多个切片;当QP相对较大时,该算法组包采用以图片或帧组为单位,同时严格限制包长在MTU范围内,从而可以适应各种码流的视频序列,同时提高了视频流序列在网络上传输的效率。

### (3)视频测试序列类型不同的情况

本文采用的视频测试序列都是标准的测试序列,选择了有场景转换的Foreman测试序列和无场景转换的News测试序列。针对两种不同类型的测试序列,在分别采用HMP算法后图像质量在性能上都相比采用FLP方法有所提高。由此可以看出,对不同类型的视频序列HMP算法都能起到一定的作用。

仿真结果显示本文提出的HMP算法,具有传输的高效性和丢包的鲁棒性,在丢包率较大的网络状况下仍获得良好视觉质量。

**结论** H.264实时视频传输是多媒体网络应用的重要课题。本文针对H.264编码特性提出了适用于H.264视频流实时传输的RTP传输模型,并在此基础上提出了混合模式组包算法HMP。实验结果证明,本文提出的HMP算法具有传输的高效性和丢包的鲁棒性,在丢包率较大的网络状况下仍获得良好视觉质量。

## 参考文献

- Schulzrinne H, Casner S, et al. RTP: A Transport Protocol for Real-Time Applications. RFC3550, July 2003
- Turletti T, Huitema1 C. RTP Payload format for H. 261 Video Streams [S]. RFC2032, Oct. 1996
- Zhu C. RTP Payload Format for H. 263 Video Streams [S]. RFC2190, Sept. 1997
- Bormann C, Cline L, Deisher G, et al. RTP Payload Format for the 1998 Version of ITU2T Rec1H. 263 Video (H. 263+)[S]. RFC2429, Oct. 1998
- Wenger S, Hannuksela M M, et al. RTP Payload Format for H. 264 Video. RFC3984, Feb. 2005
- Wenger S. H. 264/AVC Over IP. IEEE Transaction on Circuits and System for Video Technology, 2003, 7(13): 645~656
- Martini M G, Mazzotti M, Chiani M. Fixed-packet-length Transcoding for Error Resilient Video Transmission over WCDMA Radio Links. In: Proc. of Packet Video 2003, Nantes, April 2003

(上接第17页)

下面,我们以数值模拟程序Lared-S(辐射多群扩散流体力学界面不稳定性程序)计算的能量物理量第1000个时间步的数据为例,使用二维9点最优内插预测算子及ILU-GMRES来测试压缩/解压缩效果。

从表中我们可以看出,随着 $\epsilon$ 的减小,数据精度损失越来越小,压缩比也越来越小,而压缩与解压缩总的开销均呈递增的趋势。

### 3.3 算例3

Lorenzo预测算子<sup>[5]</sup>是外插预测算子,文<sup>[5]</sup>证明了它成形如:

$$F(x, y) = ax + by + c$$

$$F(x, y, z) = ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + jz + k$$

的多项式数据具有最高的预测精度,下面我们用本文的内插预测算子与其做一对比。

从以上内插与外插的压缩测试结果可以看出,使用内插预测的压缩比大大优于使用外插预测时的情形,但是使用内插预测的解压缩时间开销远大于使用外插预测时的情形,因此二者各有优缺点。

**总结** 本文所得到的图2所示的内插预测算子具有较高的预测精度,基于该预测算子的科学数据压缩算法可以根据用户对科学数据的不同精度要求,灵活调整 $\epsilon$ 的值,以达到无损甚至无损压缩的目的,测试结果表明该方法取得了很好的压缩效果。此外,该压缩算法非常鲜明的一个特点就是其压

缩时间远远小于解压缩时间,这为大规模科学数据的实时、在线压缩提供了有效手段。

## 参考文献

- Engelson V, Fritzsion D, Fritzsion P. Lossless Compression of high volume data from simulation. IEEE Computer Society, March 2000. 754~765
- Klimenko S, Mitselmakher G. Lossless Compression of LIGO data. Technical Note, California Institute of Technology, 2000
- Ibarria L, Lindstrom P, Rossignac J, et al. Out-of-core compression and decompression of large n-dimensional scalar fields. Eurographics, 2003, 22
- Trott A, Moorhead R, McGinley J. Wavelets applied to lossless compression and progressive transmission of float point data in 3D curvilinear data. IEEE Computer Society Press, Oct. 1996. 385~388
- Gamitoa M N, Dias M S. Lossless Coding of Floating Point data with JPEG 2000 Part 10. 2004
- <http://hdf.ncea.uiuc.edu>
- Ibarria L, Lindstrom P, Rossignac J. Scientific data compression. <http://www.llnl.gov/isr/guests/students/FY05-posters/IbarriaLawrence.pdf>
- 吴乐南,徐孟侠.数据压缩.北京:电子工业出版社,2000
- 陈虹,夏芳,宋磊.三维等离子体粒子模拟程序的数据模型和IO性能改进.计算机工程与应用,2004,20:104~107
- 徐树方.矩阵计算的理论与方法.北京:北京大学出版社,2001