

基于无线网络的实时视频传输系统的设计

王 峰^{1,2}, 蔡敬菊¹, 魏宇星¹, 刘云峰^{1,2}

(1 中国科学院 光电技术研究所, 四川 成都 610209; 2 中国科学院 研究生院, 北京 100039)

摘 要: 通过对流媒体技术和网络协议的深入研究, 设计和实现了一种基于 RTP(实时传输协议)和 FEC(前向纠错)的无线视频实时传输系统. 该系统采用 C/S(Client/Server)架构, 服务器端基于 RTP 协议对数据包进行封装, 并加入 FEC 纠错机制; 客户端根据 RTP 数据包的头部信息, 重建包序列, 定位丢失的数据包, 并利用 FEC 算法对丢失数据进行恢复. 实际测试证明该系统能够在无线网络中有效地保证接收端数据的实时性和完整性.

关键词: 实时传输协议; TS 流; 前向纠错

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2012)05-0058-04

The Design of Real-time Video Transmission System Based on Wireless Network

WANG Feng^{1,2}, CAI Jing-ju¹, WEI Yu-xing¹, LIU Yun-feng^{1,2}

(1 Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China;

2 Graduate School of the Chinese Academy of Sciences, Beijing 100039, China)

Abstract: According to the research on streaming media technology and network protocol, this paper designed and implemented a real-time streaming media transmission system, which is based on RTP and FEC. Within the framework of C/S network architecture, the system was implemented as follows: the server encapsulated data packet through the RTP protocol while taking into the FEC mechanism; the client reconstructed the packet sequence, and then located the lost packets by RTP Packet's header information and recovered them by using FEC. The practical tests show that the system can effectively guarantee both the real-time performance and the integrity of the receiver's packets transmitted through the wireless network.

Key words: real-time transport protocol; TS streaming; forward error correction

1 引言

与文本信息的传输不同, 实时视频在网络传输中面临着一些特殊问题: (1) 对数据的实时性非常敏感, 如果数据包超过一定的时间段才到达, 就不可用了; (2) 由于视频数据量巨大, 因而传输的都是压缩率很高的码流, 而码流之间关系密切, 导致对错误极为敏感, 少量的错误就可能导致解码图像质量严重下降. 所以, 既要保证数据的实时传输, 又要保证接收数据的相对完整.

在网络传输协议中, TCP 协议能够保证数据的

相对可靠传输, 但传输前需要建立连接, 收到数据包后需要进行 ACK 确认, 并对丢失的数据进行重传, 这些机制导致 TCP 协议不能满足传输的实时性, 所以一般对实时数据进行传输时都采用 UDP 协议. 而 UDP 协议虽然能够保证数据的实时传输, 却无法处理网络拥塞、数据丢包等现象, 这样就会导致数据接收不完整. 基于这些原因, IETF(Internet Engineering Task Force)提出了实时传输协议(RTP), 它是专门针对 Internet 上实时数据流制定的一种传输协议, 为数据包提供了编号、时间信息和多点传送等功能^[1], 利用这些功能我们就可以针对传输过程

收稿日期: 2011-09-06; 修回日期: 2011-10-20

基金项目: 中国科学院科技创新基金项目(A08K001)

中出现的数据包乱序、丢失等现象,采取一定的算法恢复数据流的完整性。

本文提出了一种基于 RTP/FEC 的实时视频传输系统,对 MPEG2-TS 流在 3G 无线网络中进行传输.系统能够较好的克服无线传输中数据包的乱序、丢失现象,最大程度地保证客户端数据接收的实时性和完整性。

2 实时传输协议

RTP(实时传输协议, Real-Time Transport protocol)由 IETF 在 RFC3550 中定义,它在多点传送或单点传送的网络服务上,提供端对端的网络传输功能,适合应用程序传输实时数据,如音频、视频或者仿真数据. RTP 协议本身并不提供任何传输可靠性的保证和流量的拥塞控制机制,它不保证传输或防止乱序传输,不假定下层网络是否可靠,是否按顺序传输数据包,具体的实现需要依靠开发者来完成。

RTP 协议由两个相关的协议组成:实时传输协议 RTP(Real-time Transport Protocol)和实时传输控制协议 RTCP(Real-time Transport Control Protocol).实时传输协议主要负责数据的封装和传输,而实时传输控制协议主要负责视频的检测控制、对数据传输质量提供反馈以及提供不同媒体间的同步.本系统数据传输采用单播模式,所以仅使用了实时传输协议 RTP,视频质量将由 FEC 机制进行控制。

RTP 协议规定数据包头部包含多项域标识,如图 1 所示.而在单播传送中,重要的标识有以下四个:

- (1) 版本号(V):2 比特,该域定义了 RTP 的版本,目前版本号为 2.
- (2) 负载类型(PT):7 比特,该域定义了负载的格式,负载类型为 MPEG2-TS 时 PT 值为 33.
- (3) 序列号(sequence number):16 比特,该域定义了数据包的序列号,每发送一个 RTP 数据包,序列号加 1,接收端可以据此检测丢包和重建包序列.
- (4) 时间戳(timestamp):32 比特,时间戳反映了 RTP 数据包中第一个字节的采样时间.

0	8	16	24	31
V=2	P	X	CC	M
Timestamp				
Synchronization Source (SSRC) identifier				
Contributing Source (CSRC) identifier				

图 1 RTP 数据包格式

有了以上这些标识,接收方就可以采取一定的措施保证数据接收的有序和完整.例如,根据 RTP 数据中的时间戳信息进行精确回放以及通过序列号进行顺序恢复;通过检测收到的 RTP 分组的序列号来判断是否有分组丢失,并可重新恢复发送时的分组顺序;通过提取 RTP 分组的时间戳字段可重新建立原始音频、视频的时序,确定数据到达时间的一致性变化等^[2]。

3 前向纠错机制

前向纠错(FEC)码是一种能有效、经济、可预测地提高传输数据可靠性的纠错方法.在通信系统中,纠错主要被用于纠正在传输过程中出现的误码;而在互联网中,传输底层的校验机制能将数据包的所有差错检测出来,并且将有差错的包都丢弃,通过传输层的协议,包的丢失位置可以得到.因而对应用层而言,前向纠错技术起的是纠删作用,用于恢复丢失的数据。

RS(Reed-Solomon)码是一类纠错能力极强的多进制 BCH 循环码,它是由 Irving Reed 和 Gusolomon 于 1960 年用 MS 多项式构造出来的一类多进制 BCH 码^[3].RS 码是前向纠错中应用最为广泛的一种编码方式,具有很强的恢复能力。

RS 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 为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.在视频流化过程中,使用 RS 编码时,通常先将视频流封装为若干个包中,然后用这些原始数据包生成冗余数据包后再传输^[4].其基本原理如图 2 所示。

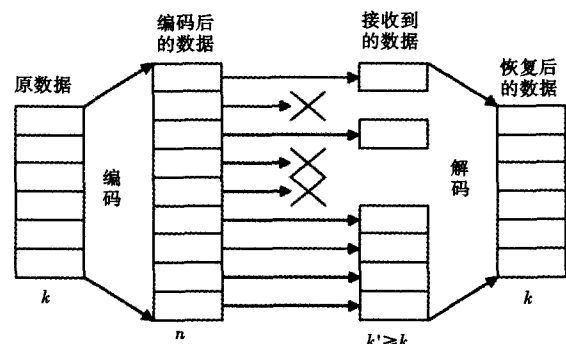


图 2 RS 编解码基本原理

设 $\vec{X} = \{X_1, X_2 \dots X_i \dots X_k\}^T$ 为源数据包向量,其中 X_i 为第 i 个数据包, G 为 $n \times k$ 生成矩阵,则 (n, k) 线性码可表示为 $\vec{Y} = G \vec{X}$,其中 $\vec{Y} = \{Y_1, Y_2 \dots$

$Y_1 \cdots Y_k)^T$ 为编码后的数据包向量. 设接收端接收到 $\vec{Y}$ 中的任意 k 个包, 其接收向量表示为 $\vec{Y}' = \{Y_{i_1}, \cdots, Y_{i_j} \cdots Y_{i_k}\}^T \subset \vec{Y}$, 其中下标 i_j 表示接收到的包号, 则有 $\vec{Y}' = G' \vec{X}$, 其中 $k \times k$ 矩阵 G' 为 G 的子矩阵, 其行向量由 G 中的第 $i_1 \cdots i_j \cdots i_k$ 行组成, 对应接收到的包号. 若 G' 是可逆的, 则有 $\vec{X} = G'^{-1} \vec{Y}'$.

G 的一种可能形式是采用单位矩阵扩展的范德蒙矩阵, 其元素为:

$$g_{ij} = \begin{cases} g_{ii} = 1 & i \leq k \\ g_{ii} = 0 & i \leq k, i \neq j \\ x_i^{i-1} & i > k \end{cases}$$

式中, x_i 为有限域 $GF(2^m)$ 中的元素, 只要所有的 x_i 不相同, 就可保证 G 中任意的 k 阶子方阵都是可逆的, 而接收到的包号可由包的序列号获得. 因此, 只要接收端接收到 n 个编码包中的任意 k 个包, 都能正确地恢复出原始数据. 若接收到的包少于 k 个, 图像所缺部分可由视频解码器用差错掩盖法补上^[5].

可以看到, 对于要传输的 k 个数据分组, 根据 RS 编码, 生成 $n(n > k)$ 个数据分组, 其中包含 $n-k$ 个冗余数据分组. 将这 n 个数据分组一起发送, 接收端接收到 k' 个数据分组, 当 $k' \geq k$ 时, 运用相应的译码方法就可以恢复出原始的 k 个数据分组. 我们将 $(n-k)/k$ 称为编码的冗余度, 一般冗余度越大, 恢复能力就越强, 但占据的带宽也越大.

4 系统设计实现方案

一般的实时视频系统包括 4 个部分: 视频采集子系统、编码压缩子系统、传输控制子系统和输出子系统. 而其中, 传输控制是实时视频系统的一个核心, 本文主要介绍传输控制子系统的设计和实现.

本系统基于 C/S 架构设计. 服务器端接收编码压缩子系统输出的 TS 视频流, 对 TS 流进行 RS 编码、RTP 封装、以及基于 UDP 协议传输; 客户端根据 RTP 信息判断数据包的乱序、丢失, 定位丢失的数据包, 并根据 RS 译码方法对丢失的数据包进行恢复, 最后重建包序列, 并将恢复好的数据流传输至显示端.

系统在 Linux 操作系统下开发, 并采用 Berkeley 套接字网络编程实现.

4.1 服务器端设计

服务器端接收 TS 流, 负责 RTP 封装, FEC 编码以及 UDP 数据传输, 并采用双缓冲和多线程方

案, 避免数据的丢失. 系统的流程图如图 3 所示.

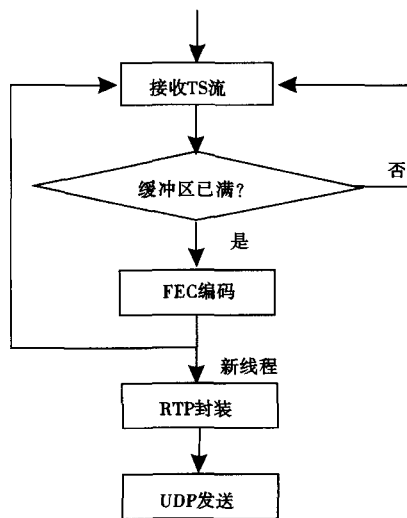


图3 服务器端流程图

系统初始时开辟两个线程和两个缓冲区, 线程和缓冲区用以接收编码器系统输出的 TS 流, 而单个 TS 流的长度固定, 为 188 字节, 这一特性给 RTP 封装^[6]和 RS 编码提供了便利, 为充分利用网络带宽, 使用 7 个 TS 流作为一个媒体数据包.

这里, 假设对 RS 编码为 (n, k) . 若使用单线程处理, 由于 RS 编码需要一定的时间, 有可能导致服务器端利用套接字接收数据流时, 数据流过大溢出套接字缓冲区而造成数据的丢失. 所以采用多线程方式, 数据系统启动时, 先用其中一个线程和一个缓冲区接收数据流, 当接收到 k 个媒体数据包后, 系统用另一个线程进行 RS 编码处理, 原线程将使用另一个缓冲区继续数据接收.

对 k 个媒体数据包的 RS 编码将生成 $n-k$ 个冗余包. 然后将 n 个媒体数据包依次封装为 RTP 数据包, 主要有两个步骤: (1) 生成 RTP 头部. (2) 将原始数据包作为 RTP 包的荷载, 生成一个完整的 RTP 数据包. 对于上述 n 个包进行 RTP 封装时, 首先对原始的 k 个包进行, 然后封装 $n-k$ 个冗余包. 每生成一个 RTP 包后, 下一个 RTP 包头部的序列号加 1. 这样, 每发送 n 个数据包, 序列号连续排列的前 k 个数据包是原始数据包, 最后的 $n-k$ 个数据包是冗余包.

4.2 客户端设计

客户端负责 RTP 数据的接收^[7], 定位丢失的数据包, 并使用 RS 解码算法恢复丢失的数据, 详细流程图如图 4 所示.

在客户端, 设计一个链表来存放接收到的 RTP

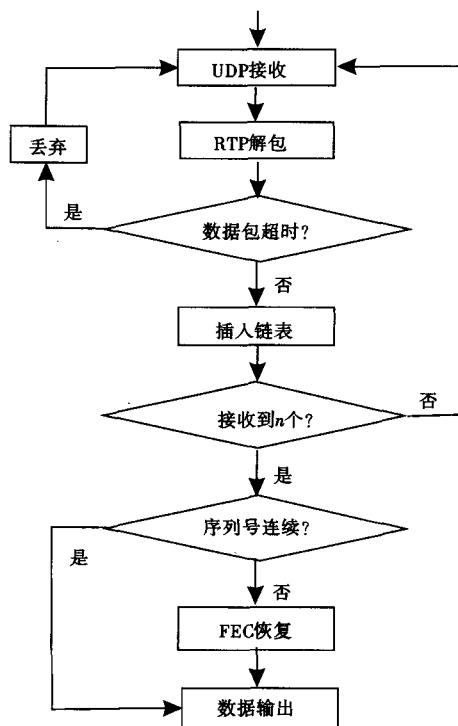


图 4 客户端流程图

荷载信息,链表的结点有数据区和序列号字段,链表是以序列号为基准的升序链表.每接收到一个 RTP 包,就取出其中的荷载信息和序列号复制到结点中,然后根据该序列号将该节点插入到链表的相应位置.

由于基于 UDP 传输时,数据包可能会乱序到达,所以我们对链表设置一个节点长度阈值,当链表的长度大于这个阈值后,才根据 RTP 序列号查找丢失的数据包,这样对那些虽然乱序但延时很小的数据包仍可以将其放置到正确的位置.而对于超过一定的时间段后才到来的数据包,由于这些数据包在前期已经当做丢包进行了恢复,这时直接丢弃即可.

当接收到的数据包的数量满足算法要求时,就可以使用 RS 解码算法对丢失的数据包进行恢复.最后,恢复好的数据按服务器端的打包顺序输送至显示端.

5 系统测试

前端利用摄像头采集视频,编码系统的视频平均输出码率为 512 kb/s,显示端使用 VLC 播放器直接接收传输系统客户端输出的 TS 流进行播放.

当系统运行在局域网中,基本不存在丢包现象时,VLC 播放器能够清晰流畅的播放视频,基本没有延时.因为没有数据包的丢失,客户端接收到的

RTP 序列号连续,RS 解码算法没有执行,系统只有 RS 编码和 RTP 打包过程,所以时延微小.

当系统运行在 3G 无线网络时,服务器端和客户端分属不同的网段,使用 3G 网络进行通信.而 3G 网络很不稳定,在传输过程中,很容易造成数据的丢包和乱序.

将视频同时传输到未采用数据包恢复算法和采用本文算法的两个系统,使用 VLC 播放器接收系统客户端输出的 TS 流产生的视频流在某一时刻的图像.可以看出,前者的画面产生了明显的马赛克;而使用本文所述算法后,马赛克现象得到了很好的克服.

通过在客户端查看接收到的 RTP 序列号,并与发送端序列号进行对比计算网络的丢包率.当采用本文算法后,网络丢包率在 20%左右时,VLC 播放器仍可以比较流畅的播放视频流,画面基本没有马赛克现象,根据 RS 算法冗余度的不同,会有 1~2 s 甚至更大的延迟;当丢包率大于 30%时,VLC 播放器中的视频会出现马赛克,并且时延也相对较大.对于网络环境更加恶劣的情况,可以适当调整 RS 编码的参数 n 和 k ,增加编码冗余度,仍可以较好的恢复数据,但视频时延会变得更大,系统运行具体比较见表 1.

表 1 系统测试结果比较

		丢包率	未使用 RS 编码	使用 RS 码 ($n=10,k=8$)	使用 RS 码 ($n=12,k=8$)
局域网	接近于 0		视频很流畅清晰,基本没有延时.	视频清晰、流畅,有微小时延.	视频清晰流畅,由微小时延.
	无线网 <20%		出现马赛克,有微小时延.	基本没有马赛克,有 1~2 s 的时延.	没有马赛克,时延可接受.
无线网 >30%			出现较大面积的马赛克,有微小的时延.	出现轻微马赛克,有 1~2 s 的时延.	基本没有马赛克,时延可接受.

使用 RS 解码算法后,视频流仍可能出现马赛克,这是因为某一时段丢失的数据包过多,超过了本文算法所能恢复的极限,导致丢失数据无法恢复.如表 1 所示,在网络带宽、RS 编码冗余度不同的情况下,系统的实现效果也有所不同.但总体上来看,本系统可以基本满足视频流在 3G 无线网络中实时传输的要求.

(下转第 66 页)

参考文献:

- [1] Chen Chiaping, Chen Chusong. Image set compression through minimal-cost prediction structures[J]. Image Processing, 2004(2): 1289-1292
- [2] Imata H K. Multi-view video coding using reference picture selection for free-viewpoint video communication[J]. Picture Coding Symposium, 2004: 15-17.
- [3] Merkle P, Muller K, Smolic A, et al. Efficient compression of multi-view video exploiting inter-view dependencies based on H. 264/MPEG4-AVC[C]// IEEE international Conference on Multimedia and Expo. Toronto, 2006: 1717-1720.
- [4] Joint Video Team of ITU-T VCEG and ISO/IEC MPEG. Joint multi-view video model [R], Geneva, 2009.
- [5] Huaiyi Pan. Development of multi-view video coding using hierarchical B pictures[J]. Congress on Image and Signal Processing, 2008(1): 497-503.
- [6] Liu Hai. Inter-view scalability and prediction multi-view video coding scheme[C]// International Forum on Information Technology and Applications, 2010(1): 129-133.
- [7] Park P K, Oh K J, Ho Y S. Efficient view-temporal prediction structures for multi-view video coding[J]. Electronics Letters, 2008, 44(2): 102-103.
- [8] 韦耿, 吴乐南, 朱斌. 移动环境视频编码功率-码率-失真模型研究[J]. 微电子学与计算机, 2011, 28(3): 13-16.
- [9] 纪涛, 张太镒. 分布式视频编码中一种基于混合模型的边信息生成算法[J]. 微电子学与计算机, 2010, 27(9): 22-25.

作者简介:

韩宝英 男, (1985—), 硕士研究生. 研究方向为多视点视频编码.

郑江滨 男, (1971—), 博士, 教授, 博士生导师. 研究方向为计算机视觉、模式识别、图像处理、多媒体嵌入式处理系统.

(上接第 61 页)

6 结束语

本文提出了一种基于 RTP 实时传输协议和前向纠错机制的方法, 对实时视频流进行传输. RTP 协议和 FEC 机制的结合, 能够较好的解决实时视频无线传输中的丢包问题, 提高了系统的服务质量. 该方案目前已经应用于基于 Linux 平台的视频采集和实时传输系统中, 并取得了良好的效果.

参考文献:

- [1] Schulzrinne H, Casner S, Frederick R, et al. RFC3550 RTP: a transport protocol for real-time application[S]. IETF, 2003.
- [2] 景慧燕, 唐存琛, 马玉利. 基于 RTP 的 MPEG-4 视频监控系统的的设计[J]. 计算机工程与设计, 2006, 27(8): 1439-1441.
- [3] 程念科, 胡瑞敏, 王中元. 基于 RS 编码的视频差错恢复方法[J]. 计算机应用研究, 2006, 23(4): 66-67.
- [4] 杨宗凯, 彭杰, 余江. 实时视频通信中的自适应前向纠错方案设计[J]. 计算机工程与科学, 2007, 29(8): 43-45.
- [5] 陈敏. 网络实时视频传输研究[D]. 广州: 华南理工大学, 2004.
- [6] 方新勇, 马瑞芳, 刘萍芬, 等. 基于 RTP 的流媒体传输系统的设计与实现[J]. 微电子学与计算机, 2007, 24(11): 183-185.
- [7] 葛镜. 基于 RTP 协议的视频流媒体实时传输[D]. 武汉: 华中师范大学, 2007.

作者简介:

王 峰 男, (1989—), 硕士研究生. 研究方向为高速实时信号处理