

嵌入式流媒体客户端缓存技术^①

冯灵霞, 王学春, 张亚娟

(黄河科技学院 计算机科学系, 郑州 450006)

摘 要: 介绍了基于嵌入式微处理器 S3C2440 的嵌入式流媒体系统的硬件结构和工作流程. 服务器端通过 RTP/RTCP 协议将流媒体数据发送出去, 客户端对收到的数据进行解压并实时播放. 将接收缓存分成接收缓冲区、播放缓冲区和 DMA 缓冲区, 三个缓冲区的大小按 1:1:2 的比例设置, 通过平均速率、延时抖动和解码码率等参数来约束缓冲区的容量. 在接收缓冲区设置两个临界点, 通过对两个临界点的检测, 来辅助调节发送端的数据发送速率. 既可以避免网络拥塞, 又可以提高流媒体的传输质量.

关键词: 流媒体, 嵌入式, RTCP, 拥塞控制, 缓存

Buffer Memory Technology of Embedded Stream Media Client

FENG Ling-Xia, WANG Xue-Chun, ZHANG Ya-Juan

(Computer Science Engineering, Huanghe Science and Technology College, Zhengzhou 450006, China)

Abstract: The paper introduces the hardware structure and working process of the embedded microprocessor based on S3C2440 streaming media system. The server will be streaming media data transmitted by RTP/RTCP protocol. The client will decompress and playback of received data real-time. The receive buffer is allocated into the receive buffer, playout buffer and DMA buffer. The sizes of the three buffers are arranged according to the ratio of 1:1:2. To constrain the buffer zone by the average rate, delay jitter and decoding rate and other parameters of the capacity. In the receiving buffer, two critical points are set. Through the detection of the two critical points, we can assist the data sending rate adjustment of the sending end. Not only can avoid network congestion, but also can improve the quality of streaming media transmission.

Key words: streaming media; embedded; RTCP; congestion control; buffer memory

随着计算机网络技术和通信技术的迅速发展, 文件传输、电子邮箱、远程登录等这些传统的网络应用已经不能满足人们的需要, 而集音/视频和共享数据于一身的多媒体网络通信逐渐成为应用热点^[1]. 流媒体技术的产生满足了人们快速获取多媒体信息的需求. 流媒体指的是一种新的媒体传送方式—流式传输方式, 它是将动画、视音频等多媒体文件经过特殊的压缩方式分成一个个压缩包, 用户不必像非流式播放那样等到整个文件全部下载完毕, 而是只需经过几秒或几十秒的启动延时即可在用户的计算机上利用相应的播放器或其他硬件软件对压缩的动画、视音频等流式多媒体文件解压后进行播放和观看, 流媒体文件的剩余部

分将在后台的服务器上继续下载^[2]. 这种流式传输方式不仅大大降低了启动延时, 同时因为前面接收的片段在播放后直接从缓冲空间丢弃, 所以也大大降低了系统整体所需的缓冲空间, 并且采用此种方式来播放媒体文件, 最终不会在客户端留有播放文件副本, 使得有效的起到了保护作品版权的作用. 在宽带网逐渐普及的基础上, 流媒体应用领域逐步拓宽, 从视频点播、远程教学、远程医疗到网络直播以及视频监控等应用都涉及到流媒体技术.

而社会信息化的日益加强使得嵌入式系统已经全面渗透到日常生活的每一个角落. 嵌入式技术成为当前微电子技术与计算机技术中的一个重要分支, 嵌入

①基金项目: 郑州市嵌入式系统应用技术重点实验室资助项目(121PYFZX177)

收稿时间: 2013-09-28; 收到修改稿时间: 2013-11-08

式系统在应用数量上远远超过了各种通用计算机,嵌入式技术的应用更是涉及电信、网络、信息家电、医疗、工业控制、航天、军事等各个领域,并日益广泛。嵌入式技术将成为后 PC 时代的主宰。

将流媒体技术应用到嵌入式系统中就形成了嵌入式的流媒体系统^[3]。从全球范围来看,日本和韩国的嵌入式流媒体业务开展的较早,像韩国 SKT 的 NATE 和 June 业务及日本 NTTDoCoMo 的 i-mode 业务和 FOMA 业务等,这些业务主要涉及视频点播、视频会议、视频邮件和远程教育等。还有比如英国和美国的相关公司推出的多种 3G 视讯业务^[4-5]。我国嵌入式流媒体业务相对起步较晚,比较有影响力的有联通的“视讯新干线”系统和广州富年公司的 Funinhand 系统等。但是这些应用仍然存在很多不足,特别是流媒体对带宽的巨大需求使得用户获得的服务并不尽如人意。

流媒体的服务质量和客户端、服务器端及网络都有关联,在嵌入式环境中,由于客户端 CPU 及内存资源的限制,因此研究如何在客户端利用有限的资源保障流媒体的传输质量显得尤为重要。本文以嵌入式流媒体客户端上的缓存技术为研究对象,通过设置缓冲区的大小和控制策略来保证流媒体播放的 QoS(Quality of Service)。

1 系统体系结构

1.1 系统硬件结构

嵌入式系统是“量身定做”的“专用计算机系统”,它的核心部件是嵌入式微处理器^[6]。本文以三星公司的 S3C2440 为嵌入式微处理器, S3C2440 是三星公司专为手持设备和一般应用提供的高性价比和高性能的微控制器解决方案,特别适合低成本的嵌入式应用系统。S3C2440 内部集成有 IIS 总线接口,可以很方便地和音频播放设备相连^[7]。S3C2440 使用 ARM920T 内核,对外提供 117 个 I/O 端口,通过设置寄存器可以确定某个引脚用于输入、输出或其他特殊功能。现设计 S3C2440 通过 IIS 和 IIC 总线和音频编解码芯片 UDA1380 相连,其中 IIS 总线负责传输音频数据, IIC 总线负责对音频控制器进行初始化和相关控制。具体结构如图 1 所示。

1.2 系统工作流程

本系统的具体工作流程是:流媒体服务器端向客户端通过网络通信接口发送流媒体数据,客户端

对流媒体数据解压并整理成正确的顺序通过 IIS 总线发送给音频控制芯片,音频控制芯片对收到的数据进行播放。

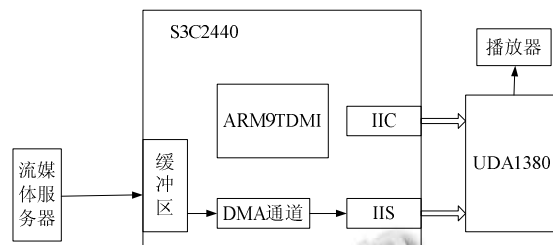


图 1 系统结构图

S3C2440 的 IIS 总线实质上就是一个音频播放接口,该接口提供三种数据传输模式:正常传输模式、DMA 传输模式和发送/接收模式^[8]。其中在 DMA 传输模式中,数据的发送或接收是由 DMA 控制器来完成的,不需要 CPU 干预。

1.3 流媒体的传输协议

Internet 在设计之初主要是为了传输单纯的文本性资料,所以不适合传输多媒体内容,如果在 Internet 网上传输流式数据就需要有合适的传输协议。在 Internet 上传输多媒体信息必须首先解决网络带宽、音/视频数据的实时传输和连续播放以及多媒体数据流的突发性强等问题,通过对所有数据通信进行分类,并且为不同的应用安排相应的机制即采用合适的协议可以有效解决这些问题。目前音/视频信号主要采用实时传输协议(RTP,Real-Time Transport protocol)进行封装传输^[9]。

RTP 协议为音/视频等实时数据提供了端到端的传输服务,在传输实时数据的同时传送用于恢复实时信号所需的定时和顺序信息并且向传输双方和网络运营商提供 QoS 检测信息。RTCP(Realtime Transport Control Protocol,实时传输控制协议)控制协议需要与 RTP 数据协议一起配合使用, RTP 本身并不能为按序传输数据包提供可靠的保证,也不提供流量控制和拥塞控制,这些都由 RTCP 来负责完成。通常 RTCP 会采用与 RTP 相同的分发机制,向会话中的所有成员周期性地发送控制信息,应用程序通过接收这些数据,从中获取会话参与者的相关资料,以及网络状况、分组丢失概率等反馈信息,从而能够对服务质量进行控制或者对网络状况进行诊断。RTSP(Real-time Streaming Protocol)可以对流媒体提供诸如播放、暂停、快进等

操作，它负责定义具体的控制消息、操作方法、状态码等，此外还描述了与 RTP 间的交互操作。

RTP 协议在网络协议体系结构中的位置如图 2 所示，它位于 UDP 之上、应用层之下。应用层产生的实时数据先被封装成 RTP 格式，再被封装成 UDP 格式，之后通过传输层协议进行传送[10]。RTP 被定义为一对一或一对多的传输模式进行工作，其主要作用是提供信息和实现流同步。为了提供流量控制或拥塞控制，当应用程序开始一个 RTP 会话时会同时使用两个端口：一个给 RTP，一个给 RTCP。RTP 首部中的媒体类型、格式、序列号、时间戳以及是否有附加数据等重要信息为 RTCP 进行相应的检测和监控提供了基础。



图 2 RTP 在协议结构中的位置

2 客户端缓存区的设计

流式传输的实现需要缓存。适量的缓存是流式传输实现的基础。因为 Internet 是以包为单位进行异步传输的，流媒体数据在传输过程中被分解成多个包，由于在传输过程中各个包选择的路由不同，所以到达客户端的时间次序可能发生变化，甚至产生丢包的现象。缓存系统的主要作用是用来弥补延迟和抖动带来的影响，并按要求对数据包进行整序，保证媒体数据能流畅地输出，从而减少网络暂时拥塞时造成播放出现停顿的次数。

流媒体的客户端主要负责接收和播放流媒体。客户端通过网络接收来自服务器端的数据流，存入缓存区中，而后对缓存区中的数据进行解码，同步输出到播放设备上。由于流媒体大多采用有损编码，再加上因为网络拥塞造成的数据丢失等原因，致使在客户端恢复出来的数据都会存在一定程度的失真，所以需要在客户端检测丢包率、时延抖动等和服务质量有关的参数，及时反馈给服务器，同时还要在客户端采用相应的控制策略，从而保证流媒体的播放质量。在嵌入式流媒体的客户端，内存和处理器资源相对比较匮乏，所以除了

需要考虑平滑网络传输带来的影响外，还要关注内存资源的节约和高效利用。在嵌入式流媒体客户端合理地安排和设计流媒体的缓存区具有重要的意义。

2.1 缓存区大小的设置

本文中将流媒体客户端的缓存区分成三个部分：接收缓冲区、播放缓冲区和 DMA 缓冲区。缓冲区的结构如图 3 所示。接收缓冲区采用先进先出的队列结构，主要负责接收来自网络的包，接收处理程序把从网络上接收到的包插到接收缓冲区的队尾，同时在队首对包进行检查，如果是有效的包则传送到播放缓冲区，如果是无效的包则直接丢弃；播放区负责对包进行整序同时也是媒体解码程序的解码缓冲区；解码程序将解码后的数据传送到 DMA 缓冲区，DMA 缓冲区也采用队列结构，该缓冲区中的数据可以在 CPU 不参与的情况下直接传递给播放设备。根据各个缓冲区的作用，接收缓冲区、播放缓冲区和 DMA 缓冲区的容量按 1:1:2 的关系进行配置。

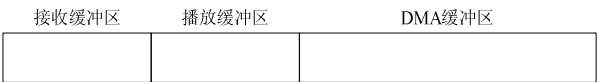


图 3 缓冲区的结构图

客户端需要一定数量的缓存区来应对丢包和失序现象，同时为客户端的流控提供条件，合理配置客户端缓存区的大小对系统性能有很大的影响。缓存区的容量如果设置过大，不仅会对系统中的宝贵资源造成浪费，同时也会使得播放时延随之增长；反过来，若缓存区容量设置过小，则会造成没有足够的存储空间来存放迟到的包。

缓存区的大小设置不是孤立的，与很多因素有关，比如：平均速率 X_{mean} 、延时抖动 T_{jitter} 、解码码率 B_{dec} 等参数都有关系。通过 RTP 的序列号、时间戳等相关字段，可以对相关参数进行测算。

由于流媒体数据并不是以恒定的速率进行传送和播放的，会随着节目源的变化而变化，另外网络的波动也会造成传输速率的不稳定，所以只能采用平均速率 X_{mean} 来表达媒体数据的传输速度。通过记录一个时间段 t 内接收到的数据包的个数和大小，就可以计算出媒体流的平均速率。平均速率与缓存区长度成正比。

延时抖动 T_{jitter} 是指同种类型数据包从发送方传送

到接收方延时的变化量, 取一个采样周期内最大的延时采样值和最小的延时采样值的差值来评价延时抖动。根据 RTP 协议字段中的时间戳可以计算延时采样值, 接收端接收到一个数据包的时间戳减去该数据包从发送端传送过来的时间戳相减即可。延时抖动与缓存区长度成正比。

解码码率 B_{dec} 是解码速率和平均帧长相结合的结果。解码速率指每秒钟对压缩数据进行解码的数量, 这是由解码器的实际解码能力决定的。流媒体数据在进行压缩编码时, 每帧的大小不是相同的, 所以选取平均帧长做为计算参数。解码码率和缓存区长度成反比。

本文利用平均速率 X_{mean} 、延时抖动 T_{jitter} 和解码码率 B_{dec} 三个参数来进行缓冲区长度的计算, 缓冲区的长度为平均速率和延时抖动的乘积除以解码码率并乘上一个通过实验得到的辅助系数。

2.2 基于接收端缓存的拥塞控制设计

IP 网作为尽力而为的网络, 它不对 QOS 提供任何保证。当多媒体通信量激增的时候, 常常会发生网络拥塞现象, 造成多媒体数据包的延迟和丢失, 进而导致媒体播放出现停顿、画面音质不清晰、音视频不同步等播放质量问题。所以, 必须在系统中使用拥塞控制算法, 以减少丢包率和缩短延迟时间。常见的有基于窗口的和基于速率的两种拥塞控制方法, 前者是根据接收端缓冲区的情况适时地调整发送窗口的大小, 后者是按每秒发送多少比特来控制数据发送。基于窗口的拥塞控制方法可以有效地解决带宽问题, 但是这种方法在网络发生拥塞时直接将发送窗口减半, 所以会使得发送速率产生抖动, 从而破坏了数据流的平滑性, 对接收端多媒体的播放质量产生影响。所以, 实时流媒体传输的拥塞控制常采用基于速率的拥塞控制方法。基于速率的拥塞控制方法既可以用在发送端也可以用在接收端或者同时用在接收端和发送端采用混合的方式, 其调整策略有基于探测和基于模型两种。

由于接收端缓存的状况也会对流媒体的播放质量产生影响, 所以本文将网络状况和接收端缓存状况结合起来, 共同对发送端的发送速率进行调整。在嵌入式系统中, 接收端(客户端)的计算和处理能力远比不上发送端(服务器端), 所以采用基于发送端的速率控制。

在客户端缓存区的接收缓冲区中设置两个临界点 C1 和 C2, 如图 4 所示。当接收缓冲区的容量小于 C1

时, 表示出现下溢危险, 此时发送端可采用基于探测的速率拥塞控制方法: 如果网络状况较好, 则逐步增大流媒体数据的发送速率, 尽可能地多传输数据流; 若网络出现拥塞情况, 则等到网络拥塞解除时, 再增大流媒体数据的发送速率, 直到接收缓冲区的容量达到临界点 C2。当接收缓冲区的容量大于临界点 C2 时, 表示出现上溢危险, 接收端应立即向发送端反馈这一消息, 发送端此时应尽快将发送速率调整到接收端的播放速率, 这样可以尽可能地在最短的时间在接收端储存尽量多的数据, 从而有效避免抖动与乱序, 保证媒体播放质量。

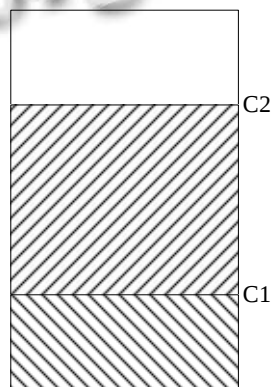


图 4 接收缓冲区临界点示意图

2.3 主动丢弃策略

主动丢弃策略指当接收缓冲区的容量大于临界点 C2 出现上溢危险时, 不仅要及时向发送端进行反馈, 同时还要在接收端采取的策略。因为从向发送端发送反馈消息到上溢危险解除是需要一段时间的, 所以在接收端采取主动丢弃的策略来保证缓冲区的高效率使用, 从整体上确保流媒体的播放质量。

主动丢弃策略和传输过程中的丢包是两个完全不同的概念, 前者是有选择地主动地抛弃数据包, 而后者是不可控地随机地丢弃数据包。在出现上溢危险时, 将对播放质量影响小的, 不重要的数据包进行主动抛弃, 直至上溢危险解除。

3 系统性能分析

我们用一台 PC 机与嵌入式流媒体系统相连, 进行实际运行测试。PC 机是服务器端, 作为流媒体的发送端, 嵌入式设备作为接收端。由 PC 端选择要播放的文件通过网络发送给嵌入式设备, 进行实时播放。经

测试,发现流媒体的播放速率可达到350kbps左右,能基本上满足用户对流媒体的播放需要。

另外,本文也以时间间隔 0.5s 测算了视频播放期间的两个参数:缓存利用率和丢包率,缓存利用率的测量数据如图 5 所示。从图上可以看出,缓存利用率整体上可以达到 80%以上,这和预期的目标是吻合的。同时根据丢包率的测量结果发现,丢包率的基本上能维持在 1%左右,正常情况下,网络的丢包率大概在 1%到 5%之间,这意味着经过对缓存控制策略进行改进,缓存区中的有效数据比例得到了大大的提高。

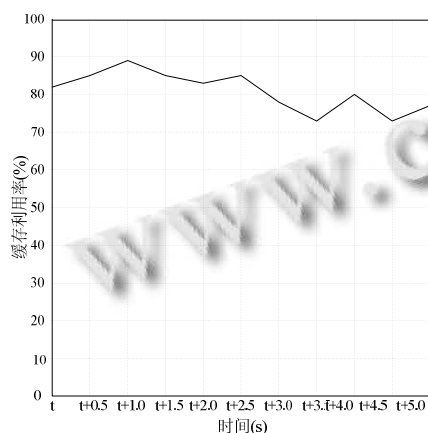


图 5 缓存利用率变化图

4 结论

将嵌入式设备和流媒体技术结合起来,已成为一个非常有意义的研究方向,是当前的研究热点。目前许多高科技的嵌入式设备都集成了在线的多媒体数据传输和播放功能。本文在基于 ARM 的嵌入式设备上

的播放性能提升为研究目的,研究了当前流媒体传输的一些关键技术,分析了常用的拥塞控制方法,通过对客户端缓存区的设计,提出了一种以接收端的缓存容量为基础的发送端速率控制方法,既可避免网络拥塞,又可以提高客户端流媒体的播放质量。

参考文献

- 1 张石.ARM 嵌入式系统教程.北京:机械工业出版社,2009.
- 2 秦贵和,徐云鹏,李宝玲.嵌入式流媒体系统的设计.微计算机信息,2007,23(8):67-68,28.
- 3 李立群,戚文芽,陈晓辉,赵昕.一种基于 Berkeley DB 的嵌入式流媒体存储系统的设计.微计算机信息,2006,22(8):17-20.
- 4 董静薇,穆英华,侯涛,孙博凯.RISC 嵌入式系统存储器管理的软件优化.哈尔滨理工大学学报,2011,16(3):72-76,81.
- 5 戴昱,黄德才.SQLite 的 SQL 语句高速缓存技术.计算机系统应用,2012,21(1):183-186.
- 6 王翠荣,高远,鄢任祥.视频点播系统的接收质量保证实现方案.计算机工程,2002,(7):52-53.
- 7 Chang SF, Sikora T, Purl A. Overview of the MPEG-7 standard. IEEE Trans. on Circuits and Systems for Video Technology, 2001, 11(6): 688-695.
- 8 Schulzrinne H, Casner S, et al. RTP: A transport protocol for real-time applications. IETF, RFC3350, 2003, (7): 4-18.
- 9 梁少波.基于嵌入式客户端的流媒体服务质量控制研究与实现[硕士学位论文].武汉:华中科技大学,2007.
- 10 徐波涛.基于 ARM 的流媒体传输方法的研究[硕士学位论文].南京:南京理工大学,2007.