

视频采集与实时传输系统的软件实现方法研究*

冯 琪, 裴海龙

(华南理工大学 自动化科学与工程学院 控制与信息工程系, 广东 广州 510641)

摘 要: 在简述 RTP/RTCP 协议原理和 MPEG-4 视觉编码标准的基础上提出了一种简单的基于 Linux 操作系统的视频采集、MPEG-4 编码以及实时传输方案, 并给出了一个基于 RTSP 协议的 WebCam 视频流传输实例; 最后, 还涉及到一些编写视频处理程序的技巧。

关键词: 视频采集; MPEG-4; RTP/RTCP; 实时传输

中图法分类号: TP311.11

文献标识码: A

文章编号: 1001-3695(2005)07-0188-02

Research on Video Capture and Real-time Transmission Using Software Implementation

FENG Qi, PEI Hai-long

(Dept. of Control Information Engineering, College of Automation, South China University of Technology, Guangzhou Guangdong 510641, China)

Abstract: Based on the principle of RTP/RTCP and MPEG-4 video encoding standard, this paper puts forward a feasible implementation of video capturing, encoding and real-time MPEG-4 video streaming on Linux. And then a Web camera video streaming example based on RTSP server/client structure is discussed. According to the scheme, the implementation methods are mainly introduced. Furthermore, some useful techniques in video stream programming are involved here.

Key words: Video Capture; MPEG-4; RTP/RTCP; Real-time Transport

1 引言

随着多媒体技术和互联网的快速发展, 数字视频的实时传输成为计算机领域研究的一个热点。本文对视频的采集和实时传输进行了探讨和研究, 并提出了一种基于 RTP/RTCP^[1] 可靠 MPEG-4^[2] 实时媒体流的实现方案。

传统的模拟信号由于抗干扰能力差而不利于传输, 而数字视频信号能够克服这个缺点, 但是采集的原始数字视频在传输过程中存在视频信息数据量巨大, 延迟敏感性强的特点, 不利于在带宽环境下的传输。因此, 必须对原始视频数据进行压缩, 使得在实时传输中能够保持较低的延迟。

目前流行的编码器 (Encoder) 基本上采用硬件实现, 存在价格高、灵活性差、不易修改等缺点。随着计算机处理速度的不断提高和内存空间的不断扩大, 通过软件方法来实现数字视频的实时传输已成为可能, 并且使用软件实现编 (Encode) / 解 (Decode) 码, 具有很大的灵活性, 根据用户的不同需要, 可以随时修改编 / 解码器, 这是硬件编码器无法做到的。

而采用 Linux 视频采集和实时传输的实现平台具有很好的特点, 如良好的开放性、稳定性、模块化、可扩充性、可移植性等诸多优点; Linux 有着强大的网络功能和丰富的软件资源; 同时, Linux 源代码开放, 使得修改操作系统以满足定制需要变得很容易。

2 MPEG-4 视觉编码标准

运动图像专家组 (Moving Picture Experts Group, MPEG) 隶属于 ISO/IEC 的一个专家工作组, 主要负责为数字音、视频编码算法开发和制定标准。MPEG-4^[2] 是 ISO 与 IEC 共同制定的编码标准, 它提供了用于甚低速率的音、视频编码, 实现了基于内容的压缩编码, 具有良好的兼容性、伸缩性和可靠性。MPEG-4 还制定了一个称为传输媒体集成框架 (DMIF) 的会话协议, 用来管理多媒体数据流。

MPEG-4 视觉是一种视觉编码标准, 它具有如下特征: 高编码效率、高纠错能力、基于多样性的、任意形的对象编码等。其速率范围介于数千位每秒到几兆位每秒, 并且它能适应从无差错网络到高错误率的移动网络等多种网络编码。考虑到 MPEG-4 具有上述如此优秀的特质, 我们将采用 MPEG-4 视觉作为实现视频实时传输的编码格式。

尽管 MPEG-4 标准本身就是为实时传输设计的, 但是在实际应用中, 可能会由于遇到应用系统自身就已经具备流的管理功能, 不再需要使用 MPEG-4 系统的类似功能, 或者 MPEG-4 编码载荷需要与其他非 MPEG-4 编码载荷一起以通用方法进行处理与传输的情况。针对以上情况, 本文将探讨运用 RTP 协议对 MPEG-4 流媒体进行实施传输的方法。为了在 RTP 上进行 MPEG-4 流传输, MPEG-4 视觉载荷格式已被定义^[2]。

3 RTP/RTCP 协议

RTP (Real-time Transport Protocol) 与 RTCP (Real-time Transport Control Protocol) 协议是编写处理网络多媒体数据流

收稿日期: 2004-07-14; 修返日期: 2004-09-21

基金项目: 国家自然科学基金资助项目 (60374036); 广东省自然科学基金资助项目 (031407)

的基础。

3.1 RTP 协议

实时传输协议 RTP 是 Internet 上针对多媒体数据流的一种传输协议。RTP 被定义为在一对一或者一对多的传输情况下工作的,其目的是提供时间信息和实现多媒体数据流的同步。RTP 通常使用 UDP 来传送数据,但是 RTP 也可以在 TCP 或 ATM 等其他协议之上工作。当应用程序开始一个 RTP 会话时将使用两个端口,即一个端口分配给 RTP 和一个端口分配给 RTCP。RTP 本身并不能为按顺序传送数据包提供可靠的传送机制,也不提供流量控制或拥塞控制,即 RTP 不提供任何保证传输质量的机制,而这个工作是由 RTCP 协议来完成的,RTCP 协议则是提供传输质量以及可靠性等服务。通常 RTP 算法并不作为一个独立的网络层来实现,而是作为应用程序代码的一部分。

RTP 可以确定正在传输的数据的类型,确定数据包的正确顺序以便正常播放,还可以同步由不同数据源过来的媒体流,如接受播放 MPEG 流时的视频、音频的同步。RTP 数据包不一定按照发送时的顺序到达,甚至不一定能够到达,顺序的重新组织、丢失包的检测,都由接收端根据包头中提供的信息来完成。

3.2 RTCP 协议

实时传输控制协议(RTCP)是提供多媒体数据流量控制和拥塞控制服务的,在 RTP 会话期间,参与周期性地传送 RTCP 包。RTCP 包中包含已发送的数据包的数量、丢失的数据包的数量等统计资料,因此,服务器可以利用这些信息动态地改变传输速率,甚至改变有效载荷类型。RTP 和 RTCP 协议配合使用,能以有效的反馈和最小的开销使传输效率最佳化,因而 RTP 和 RTCP 协议特别适合传送网络上的实时数据。

3.3 实时流协议(RTSP)

实时流协议(Real-Time Streaming Protocol, RTSP)是由 Real Networks 和 Netscape 共同提出的。该协议定义了一对多应用程序如何有效地通过 IP 网络传送多媒体数据。RTSP 在体系结构上位于 RTP 和 RTCP 之上,使用 TCP 或 RTP 完成数据传输。HTTP 与 RTSP 相比,HTTP 传送 HTML,而 RTP 传送是多媒体数据。HTTP 请求由客户机发送,服务器作响应;使用 RTSP 时,客户机和服务器都可以发出请求,即 RTSP 可以是双向的。

4 视频采集和实时传输系统的设计

4.1 视频采集和实时传输系统的硬件架构设计

在视频采集和实时传输系统的设计过程中,采用服务器/客户机(Server/Client)结构。硬件部分(图 1)由四部分构建:

(1) USB 摄像头。采用 Logitech QuickCam Pro4000,它的特点具有高质量 VGA CCD 感测器,可以满足我们对图像采集实时性的高要求。

(2) RTSP Server。使用安装 Linux 操作系统并且配置有 PCMCIA 无线网卡的笔记本电脑。

(3) RTSP Client。使用安装 Linux 操作系统并且配置有 PCMCIA 无线网卡的台式电脑。

(4) IEEE802.11b 无线网络。使用由 LINKSYS 的无线接入点(Access Point)构成的 10Mbps 的无线局域网。

以上系统的硬件配置原则还包括:RTSP Server 和 RTSP Client 需要有较好的数字图像处理能力,能高效地完成软件编/解码的工作。具体实施的硬件配置可以按需要进行调整和扩展,以满足不同的应用:对 RTSP Client 端的进行扩展可以实现 Multicast(一对多)的图像传输;类似地,对 RTSP Server 端的扩展可以实现多对一的效果。

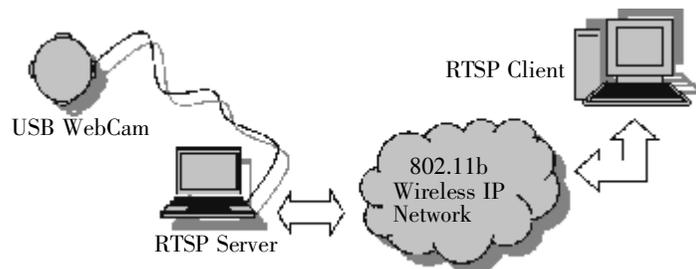


图 1 视频采集和实时传输系统的硬件架构

无线网络有着广阔的市场前景,正在成为嵌入式应用中一个快速增长点。基于 802.11b 网卡的应用,802.11b 网卡可以通过点对点方式与另一块网卡通信,也可以通过一个接入点(Access Point)接入局域网。在 Linux 下,现有的采用标准网络编程接口(Socket API)的应用程序都可通过 802.11b 网卡完成通信,如远程登录、FTP、Web 浏览器、Web 服务/数据库、各种语音/多媒体应用等都可以移植到嵌入式系统上,方便地构筑各种无线平台。对于中短距离的无线接入,IEEE802.11b 是一个理想的方案,传输距离可达 300m,而且可以通过增加接入点(Access Point)来扩大覆盖范围。本系统的设计完全可以应用到嵌入式系统当中,也正是我们设计该系统时的开发方向。

4.2 视频采集和实时传输系统的软件架构设计与实现

系统的软件架构按功能进行划分,如图 2 所示。

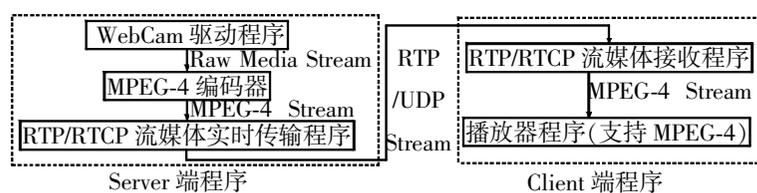


图 2 视频采集与实时传输系统软件结构图

基于服务器/客户机的结构,服务器端包括 USB WebCam 设备驱动程序、MPEG-4 编码器、RTP/RTCP 流媒体实时传输程序。系统初始化时,打开 USB 视频设备文件(如 /dev/video0),把读取的 YUV420P 格式的视频帧读入视频缓冲区,然后通过 MPEG-4 编码程序进行编码,最后把存在视频缓冲区的 MPEG-4 流媒体打包成 RTP 流,通过 UDP/IP 网络传输。其中比较关键的是摄像头设备类和编码器类的编写,如下:

```
#include < linux/videodev.h >
class Webcam
{
public:
Webcam( const string & deviceString );
//构造函数,打开设备 /dev/video0
virtual ~Webcam(); //析构函数
void GetGrabbedFrameData( void* * buffer, int* w, int* h );
//捕获 w* h 的视频帧到缓冲区
public:
string _deviceString; //设备文件 /dev/video0
...
};
class Encoder
```

(下转第 195 页)