

Software Design of Video Communication Server Based on TMS320DM8168

Yong Bai, Linbo Qing*, Xiaohai He, Xiong Fu

Image Information Institute, College of Electronics and Information Engineering, Sichuan University, Chengdu Sichuan
Email: 254548525@qq.com

Received: Nov. 13th, 2015; accepted: Nov. 28th, 2015; published: Dec. 1st, 2015

Copyright © 2016 by authors and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper implements a video communication server based on RTSP protocol using an embedded hardware platform TMS320DM8168. It specifically elaborates the hardware platform and development framework of TMS320DM8168, the realization of video communications server based on RTSP protocol as well. System captures the video flow through a camera firstly, then doing H.264 coding works. After that, the coded flow is packed as RTP packets; meanwhile, the system will establish a RTSP serve module. Finally, the server transfers the coded video flow to the receiver which displays the experiment results.

Keywords

DM8168, MCFW, H.264, RTSP

基于TMS320DM8168的视频通信服务器的设计

白 勇, 卿粼波*, 何小海, 付 雄

四川大学电子信息学院图像信息研究所, 四川 成都
Email: 254548525@qq.com

收稿日期: 2015年11月13日; 录用日期: 2015年11月28日; 发布日期: 2015年12月1日

*通讯作者。

文章引用: 白勇, 卿粼波, 何小海, 付雄. 基于 TMS320DM8168 的视频通信服务器的设计[J]. 图像与信号处理, 2016, 5(1): 1-8. <http://dx.doi.org/10.12677/jisp.2016.51001>

摘要

本文在嵌入式硬件平台TMS320DM8168上实现了一款基于RTSP协议的视频通信服务器，详细阐述了DM8168的硬件平台与开发框架，以及基于RTSP协议的视频通信服务器的实现。系统先通过摄像头对视频进行采集，然后对视频进行H.264的编码，再将编码后的H.264流进行RTP封包并建立好RTSP服务模块，将视频传输给接收端并进行了实验结果的展示。

关键词

DM8168, MCFW, H.264, RTSP

1. 引言

近年来，通信技术有了很大的进步，为人们越来越高的通信要求提供了技术支持。多媒体通信是通信技术的一个重要分支，它能够提供更语音、视频和数据的立体式服务，在远程教育、视频通话、视频监控等方面有着广泛的应用[1]。目前，国内外对流媒体技术进行了深入的研究，取得了很大的成就。同时流媒体技术也存在一些问题，例如压缩率不高、传输速率不够快等，并且目前的研究大部分都是在PC机上实现的，对设备的依赖性较高。基于以上原因，本文设计了基于嵌入式设备TMS320DM8168的视频通信服务器，对视频进行了H.264的编码，然后采用RTP协议进行打包传输，从而完成视频通信服务器的设计。同时由于嵌入式设备的便携性和易开发等特点，相对于PC上的流媒体服务器有更广泛的应用范围，基于嵌入式设备的流媒体通信有着重要的应用和研究价值，所以本文的设计是在嵌入式设备上实现的[2]。该视频服务器可以实现对视频的采集、处理和编码，通过网络接口传输到视频接收端，完成了视频通信中的一项重要工作，具有一定的研究价值。

2. 系统框架设计

本文旨在设计一款视频通信服务器，摄像头对视频进行采集后送入视频通信服务器，视频通信服务器对视频进行相应的处理，通过传输协议传送给PC、嵌入式设备、或者手机等终端进行播放。系统框图如图1所示。

2.1. 系统软硬件平台

本系统的硬件平台是TMS320DM8168以为核心处理器的高速图像处理平台。核心处理器TMS320DM8168实现图像处理和外设控制。除了核心处理器外，还有DR3、网口、视频输入输出、电源复位和时钟等外设。其中DDR3容量是2GB，主要用来提供操作系统运行和程序所需要的内存空间。模数转换芯片是将摄像头采集的模式视频转换成数字视频传给核心处理器处理。网口采用的是千兆网，主要用来进行数据传输。时钟电源和复位是为系统的运行提供支持。总体框图如图2所示。

TMS320DM8168是TI公司推出的一款高性能处理器，它采用的是多核异构模式，集成有四个处理器：Cortex A8、C674x、VIDEO M3和VPSS M3。四个处理器通过协同分工，独立完成各自的任务，然后再经过核间通信等方式进行协同从而完成复杂的功能，这种分工协同的开发方式是通过TI提供的多通道处理框架(MCFW)来实现的。为了进行图像处理，在MCFW中定义了许多Link节点，每个Link节点独立的完成一项处理，处理完成后调用需要的Link节点并将消息传递给下一个Link节点，下一个Link

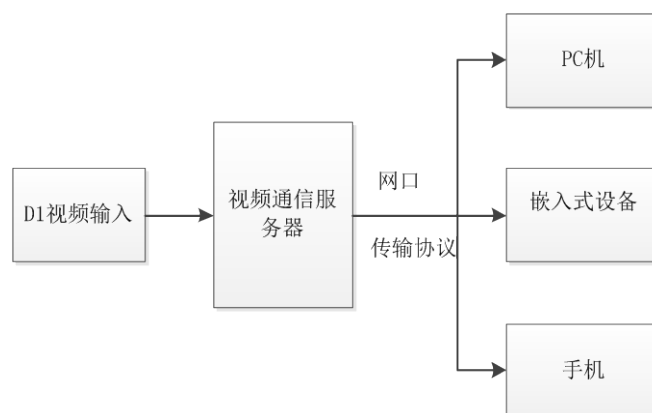


Figure 1. System architecture

图 1. 系统框架图

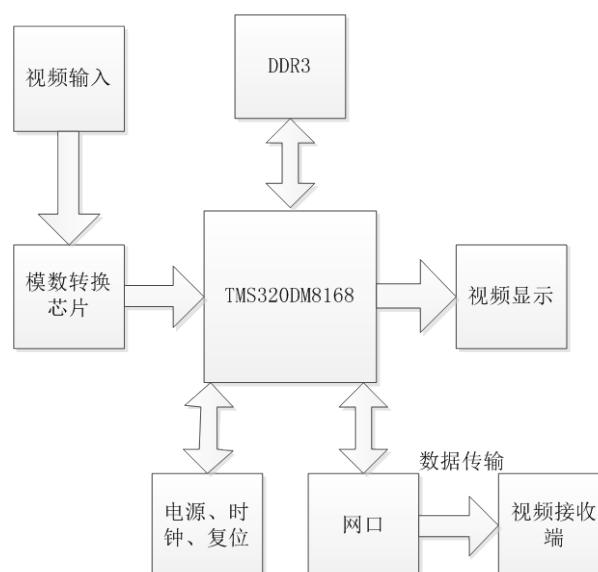


Figure 2. Hardware system architecture

图 2. 硬件平台框图

节点接着对图像进行处理。所以用户只要有序的创建和调用需要的 Link 节点就能完成复杂的图像处理操作。这种方式将复杂的图像处理操作分解为了许多简单的图像处理并调用 Link 节点实现，大大降低了开发难度，同时也提高了图像处理的效率[3]。

2.2. 传输协议

本文的传输协议采用的是实时流传输协议(RTSP)。RTSP 是 TCP/IP 协议体系中的一个应用层协议，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，它使用 TCP 或 UDP 完成数据传输[4]。

作为应用层协议，RTSP 用于若干个同步的媒体数据流连接的建立和控制，但它并不能传输媒体数据流；RTSP 与 RTP/RTCP 等更低层的协议一起配合使用，提供基于互联网的整套媒体流化服务。RTSP 不但为服务器和客户端建立会话控制连接，还实现媒体数据流的远程控制操作。因此客户端首先应该连接服务器的 RTSP 端口；客户端建立 RTSP 连接后发送 DESCRIBE 方法给服务器，方法中包含目标点播文件的 URL。客户端将 RTSP 包的 DESCRIBE 方法发送给目的服务器；服务器收到此方法信息后会传送

标准的媒体描述文件 SDP 到客户端播放器;客户端读取并分析 SDP 描述文件按具体内容来配置相应音视频数据解码的同步信息,例如:网络类型、RTP 传输通道的端口号、编码类型、采样率等。

3. 视频通信服务器设计与实现

3.1. 系统软件设计框架

为了实现视频通信服务器,首先对视频进行采集,然后将采集的视频进行基本的去噪处理,接着对视频进 H.264 的编码,再对编码后的码流进行 RTP 打包,最后建立 RTSP 服务器。服务器的软件设计是在 TI 提供的多通道框架(Multi Channel Framework, MCFW)中实现的。如图 3 所示。

3.2. 视频的采集和 H.264 编码

本文所设计系统的服务器端软件设计是在 MCFW 框架内实现的。在 MCFW 框架中,进行图像处理时,先将复杂的处理分配给一个个 Link,通过调用各个 Link 完成相关功能,再通过 Link 的通信实现整个处理。要实现服务器端的设计,首先需要对视频进行采集和预处理和 H.264 编码,然后才能进行 RTP 打包发送,流程图如图 4 所示。

为了实现对视频的采集、编码,将这个过程拆分成了一个 Link,上游 Link 先实现自己的功能,然后将消息传递给下游 Link,这样完成整个处理流程。具体过程为:视频数据从 Capture Link 传给 Nsf Link,完成去噪处理,这个过程都是在 VPSS M3 核上完成的,所以不需要 Ipc link,直接通过 nextLink 和 prevLinkId 就可以实现。由于 Nsf Link 运行在 VPSS M3 核上,而 Encode Link 是运行在 VIDEO M3 上的,所以这两个 Link 之间的通信需要在中间添加另外两个用于核间通信的 Ipc Link,即 ipcOutVpssId 和 ipcInVideoId;视频数据先从 VPSS M3 出来之后,再进入 VIDEO M3 核中。从而实现两个核之间的通信。通过上面的 Encode Link 之后,视频数据被编码成了 H.264 文件格式,接下来需要把这些 H.264 文件传到 Cotex A8 中,等待 RTP 打包并通过网口发送出去[5]。

3.3. H.264 文件的 RTP 打包

视频经过 H.264 编码之后生成了一个个 NALU (网络传输单元),对 H.264 进行打包就是将一个个的 NALU 进行 RTP 打包,主要分为 RTP 头打包实现和 RTP 有效负载打包实现。

1) RTP 头的打包实现

对 RTP 头进行打包,就是对 RTP 报文头部的各项参数进行赋值,赋值的依据就是系统的要求。首先需要申请一个 1500 字节大小的数组 mysendbuf[1500],生成的 RTP 分组就保存在这个数组中。其前 12 个字节用于 RTP 分组首部,后面的部分用于 RTP 分组负载。在 RTP 报文头中,版本字段 V 固定为 2。本系统不需要填充比特和扩展首部,同时也没使用 CSRS 标识符,所以填充 P、X 和 CSRC 都赋值为 0。M 是边界标识符,所以在单一封包模式 M 赋值为 1,同时分片封包模式的最后一个分片 M 也赋值为 1,其它情况下赋值为 0。PT 在本系统指定为 100。序列号的起始值指定为 1,后面每发送一个 RTP 包序列号加 1。时间戳,一帧视频的所有 RTP 分组的时间戳都是相同的,后一帧视频的时间戳比前面一帧的时间戳增加一个时间戳增量。同步源标识符 SSRC 字段,在本系统中唯一即可,程序中将其指定为 1234 [6] [7]。

2) RTP 分组负载的实现

本系统对采集到的原始视频数据进行 H.264 编码后,输出的是一个一个的 NAL 单元。每个 NAL 单元都由三部分组成,NALU 的结构如下所示:

[开始码] [NALU 头] [NALU 有效负载]

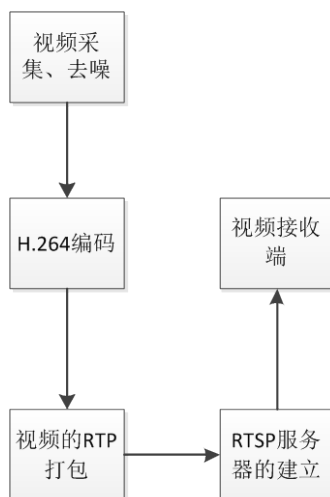


Figure 3. Software design of video server

图 3. 视频服务器软件设计方案

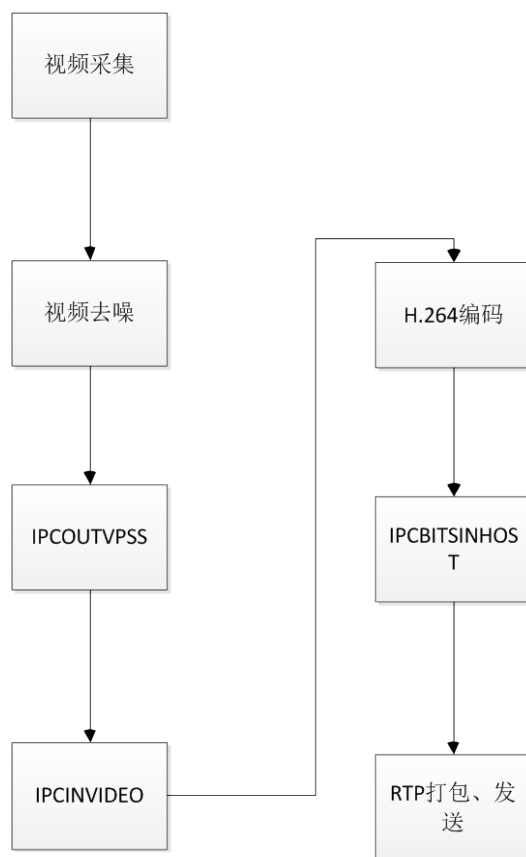


Figure 4. Video processing flow

图 4. 视频处理流结构

本系统开始码赋值为 00 00 00 01。NALU 头部占用 1 字节，NALU 有效负载为多个字节的 RBSP 数据。本系统的最大传输单元(MTU)设为 1400。根据 NALU 大小来区分：尺寸很小采用组合封包，小于 1400 字节采用单一封包，大于 1400 字节采用分片封包[6]，如图 5 所示。

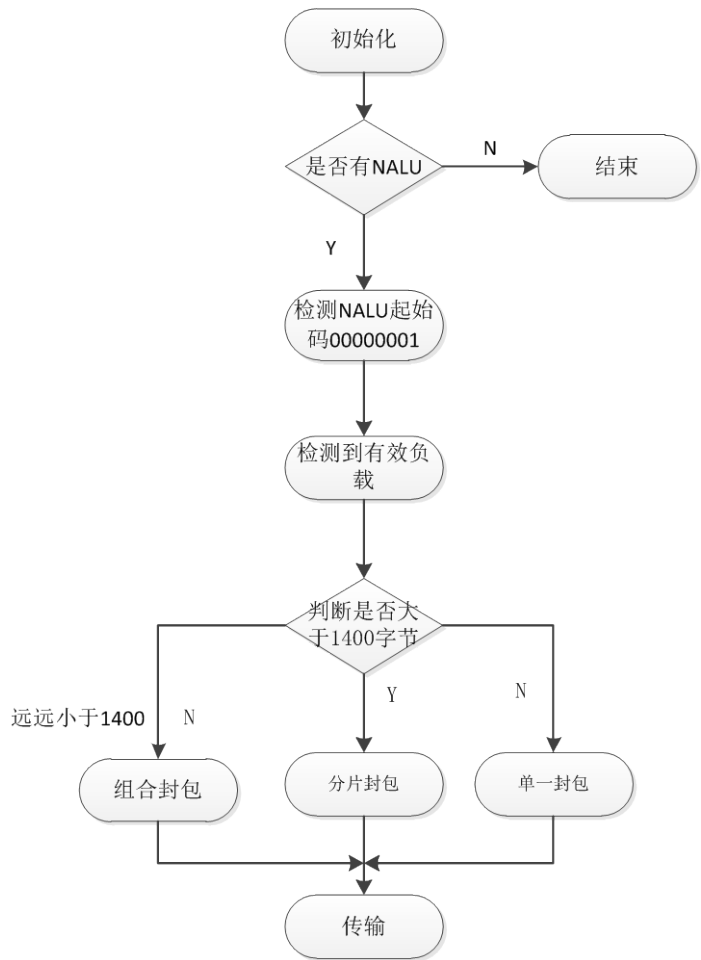


Figure 5. The implementation of RTP payload
图 5. RTP 分组负载的实现

RTP 打包流程：首先检测到 NALU 包，检测 NALU 的起始码 00 00 00 01，去掉 NALU 头部，检测到有效负载。判断 NALU 有效负载的大小，分别采用分片封包、单一封包和组合封包，将之写入创建的数组中，等待网络发送。

3.4. RTSP 服务模块

当 RTSP 客户端需要服务时，它先要对 RTSP 服务器发起连接请求，然后 RTSP 服务器对请求进行响应。对客户端的响应是由 RTSP 服务模块完成的(图 6)。

在本系统服务器模型中，为了处理客户端的 RTSP 请求，首先分析 RTSP 请求报文各头字段信息，然后程序判断 RTSP 请求的命令码再分别对其进行处理，并回复 RTSP 应答报文。当服务器正确处理 PLAY 请求后，会将该用户的 RTSP 状态设置为 STATE_PLAYING，然后开始向客户端发送封装有 H.264 视频数据的 RTP 分组。

RTSP 服务器模块初始化后，程序会一直监听系统的 554 端口，等待客户端连接。客户端与服务端建立连接后，客户端会按顺序发送 RTSP 请求报文，服务端会根据请求报文内容做相应的处理，并回复 RTSP 响应报文。然后双方完成 RTSP 协商，服务器开始发送封装有 H.264 频数据的 RTP 分组，客户端接收到 RTP 分组后解码显示。最后，服务结束，关闭连接[7]。

4. 服务器端实验效果展示

通过第二节设计实现了视频通信服务器，为了验证正确性，采用 PC 机作为视频接收端。在 PC 机安装开源播放器 VLC，连接网线，在 VLC 中输入 IP 地址：`rtsp://192.168.196.128/steam1.264`，请求服务器播放视频，服务器响应请求，将视频传输到 PC 机端并在 VLC 中显示。图 7 是系统硬件平台的连接图，图 8 是 PC 机所接到的视频在 VLC 上的显示画面。

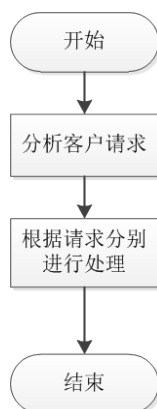


Figure 6. Server processing
图 6. 服务器模块处理过程

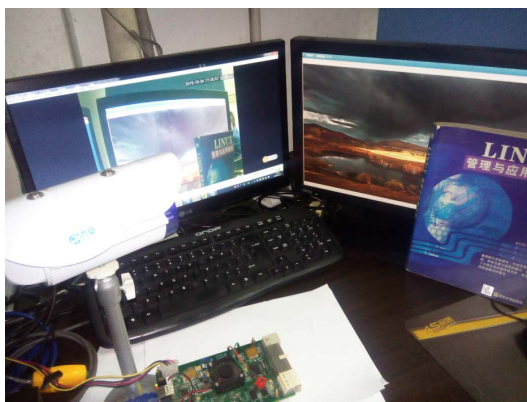


Figure 7. The result picture 1
图 7. 实验效果图 1

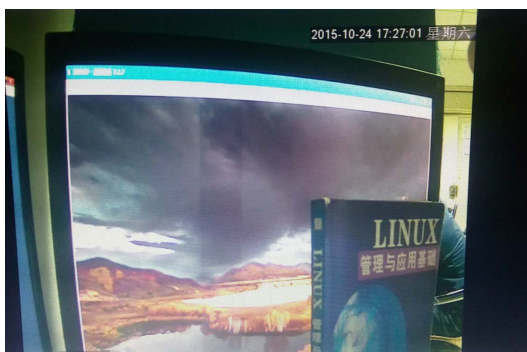


Figure 8. The result picture 2
图 8. 实验效果图 2

5. 结论

本文针对快速实时的传输视频数据的应用,设计与实现了一款基于 TMS320DM8168 的视频通信服务器,实现了嵌入式设备对视频进行采集、编码和传输,并建立了 RTSP 服务器。本文通过 MCFW 框架对视频进行采集和 H.264 的编码,再对编码过的视频流进行 RTP 打包,建立好 RTSP 服务器,从而完成了视频通信服务器的设计,并在最后测试了系统设计的正确性。由于受到硬件系统的限制,本文系统处理的视频是 D1 格式的,如果想实现更加清晰的播放效果可以考虑采用高清视频实现。

基金项目

国家自然科学基金(61201388)。

参考文献 (References)

- [1] 龙舟. 基于 RTSP 的智能视频监控系统的研究与实现[D]. 长沙: 中南大学, 2013.
- [2] Cai, X., Ouyang, G. and Zhang, X. (2014) The Design of Streaming Media Video Terminal Based on Embedded Linux. *Future Generation Communication and Networking (FGCN)*, 2014 8th International Conference on IEEE, 68-71.
- [3] 管庆, 朱海, 王凯, 等. 基于 TMS320DM8168 的视频监控跟踪系统[J]. 数据采集与处理, 2013(6): 652-657.
- [4] RFC2326. Realtime Streaming Protocol (RTSP).
- [5] 郑亮. 基于 DM8168 多路视频监控系统研制[D]. 杭州: 杭州电子科技大学, 2014.
- [6] 李校林, 刘利权, 张杰. 基于 RTP 的 H.264 视频流实时打包传输的研究[J]. 计算机工程与科学, 2012, 34(5): 168-171.
- [7] 宋建勋. 基于 DaVinci 技术的多平台网络视频监控系统的设计与实现[D]. 南京: 南京邮电大学, 2011.