第5卷第1期 江南大学学报(自然科学版) Vol.5 No.1
2006年2月 Journal of Southern Yangtze University(Natural Science Edition) Feb. 2006

文章编号:1671-7147(2006)01-0050-04

SIP协议栈在TMS320DM642上的实现

万春新, 唐慧明*
(浙江大学 电气工程学院,浙江 杭州 310027)

摘 要:为实现基于TMS320DM642多媒体处理器的可视电话,采用相对简单的SIP协议作为信令协议;研究了SIP的结构、机制,并运用回调函数和UDP并发服务器算法实现了基于SIP的可视电话.实践证明,SIP协议结构简洁,易于扩展和适于Internet接入.

关键词:SIP;TMS320DM642;媒体处理器;可视电话
中图分类号:TP 311.1 **文献标识码**:A

Implementation of SIP Protocol Stack on TMS320DM642

WAN Chun-xin, TANG Hui-ming*
(College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: The paper presents the implement of video telephone based SIP using callback functions and UDP concurrent server . The application result shows that SIP is simple, prone to expandable, suitable to access to Internet.

Key words: SIP; TMS320DM642 ; media processor; video telephone

随着网络技术和多媒体技术的不断发展,多媒体通信业务逐渐在Internet应用中占据主导地位,其中以VOIP技术的应用尤为突出.VOIP即基于IP网络的语音通信,它不仅是狭义上的IP电话,更是一个能提供包括视频交互在内各种多媒体业务的综合性应用平台.VOIP系统一般基于SIP或H.323两种通信控制协议.在软交换技术即将成为下一代网络交换技术的大背景下,SIP以其诸多优势被认为可作为软交换技术的接入层核心协议,并被成功应用于诸多电信运营商的代表性系统中[1].

TMS320DM642是德州仪器公司(TI)发布的新型数字媒体处理器,该处理器是专门为视频与影像市场量身定制的,特别适用于视讯终端、视频点播(VOD)、多信道数字视频摄录像应用以及高品质视频编码与解码解决方案,该全套数字媒体处理器基于TI业经验证的TMS320C64x系列DSP架构,能与TI的其它C64x数字信号处理器进行代码兼容,并且可以同时连接多达6个符合BT 656的8位视频流.处理器可提供片上集成的视频端口、无缝以太网及多信道音频[2].

1 SIP协议

1.1 SIP协议的工作方式

会话初始化协议(Session Initiation Protocol, SIP)是由IETF(互联网工程任务组)提出的IP电

收稿日期:2004-10-14; 修订日期:2005-02-20.
作者简介:万春新(1970-),男,甘肃景泰人,通信与信息系统专业硕士研究生.
*通讯联系人:唐慧明(1963-),男,浙江桐乡人,副教授,工学硕士,硕士生导师.主要从事多媒体通信、视频压缩编解码及模式识别等研究.Email:tanghm@zju.edu.cn

话信令协议.它被用来创建、修改和终结一个或多个参与者参加的会话.SIP 具有许多优点,在下一代网络(NGN)和第三代移动通信(3G)中都有重要的应用.SIP 和会话描述协议(SDP)结合,可以在 IP 网上提供价格便宜的电话业务、数据业务甚至多媒体业务.

SIP 协议采用客户机/服务器的工作方式,SIP 网络包含两类组件:用户代理(User Agent)和网络服务器(Network Server).用户代理又分为用户代理客户端(UAC)和用户代理服务器(UAS),其中 UAC 负责发起 SIP 呼叫请求,UAS 负责对呼叫请求做出响应.网络服务器主要为用户提供注册、认证、鉴权和路由等服务.SIP 消息有两种类型:从 UAC 发到 UAS 的请求消息和从 UAS 发到 UAC 的响应消息.请求消息包含请求行、头、消息体 3 个元素.响应消息包含状态行、头、消息体 3 个元素.请求行/状态行和头定义了呼叫的特征,消息体独立于 SIP 协议并可包含任何内容[3].

SIP 是一个分层体系结构的协议,最底层是语法和编码层,它的编码使用扩展的巴科斯范式(ABNF)规定.第 2 层是传输层,它定义了网络上的客户机和服务器如何接收请求和发送响应.第 3 层是事务层,事务是 SIP 的基本元素,事务由客户机向服务器发送的请求和从服务器发回客户机的所有响应组成.事务层若处理消息超时,则重传并匹配响应到请求.事务层之上的层称为事务用户(TU),每个 SIP 实体(除了无状态代理)都是事务用户.当一个事务用户希望发送请求时,就创建一个客户机事务实例以发送此请求.SIP 提供用户定位、用户通信能力协商、用户可用性、建立呼叫和呼叫处理与控制等功能.SIP 不提供具体的业务,不定义会话将如何描述,只提供会话建立功能.SIP 协议清晰地将会话建立和会话描述区分开,使 SIP 可以成为 Internet 上多媒体会话中真正意义上的信令协议,可以和其他协议如 RTP、RTSP 和 MEGACO 协议一起使用,并提供不同的服务,使得 SIP 协议的适用范围很广.

1.2 SIP 协议栈的结构与应用程序

SIP 协议栈组织分为 SIP/SDP 协议栈管理层、对话层、事务层、SIP/SDP 消息编码解析层和传输层 5 个层次,每层提供相应的 API.协议栈总体结构和应用程序与各层的关系见图 1.

1.2.1 *SIP/SDP 协议栈管理层*　负责系统配置、分配内存、管理资源、提供登录协议栈和进行管理的命令,以及所有其他层的初始化和关闭.应用程序使用本协议栈前,必须先调用该层的初始化接口以初始化要使用的层;在结束应用程序前,必须调用本层的关闭接口关闭相应的层.该层提供的 API 有资源管理、协议栈初始化和关闭接口.

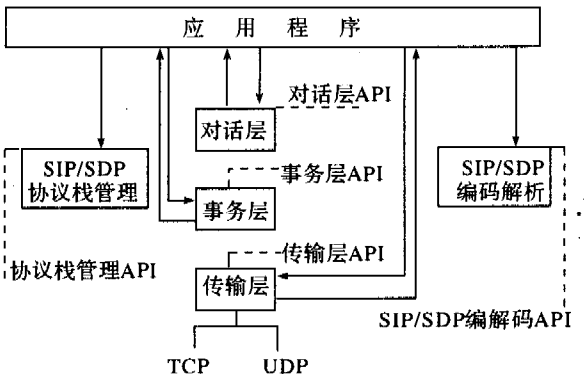

图 1　SIP 协议栈实现总体结构

Fig. 1　The structure of SIP stack

1.2.2 *对话层*　创建对话(Dialog),管理对话内的所有 SIP 消息,并对这些消息进行处理,创建请求并生成响应.该层也负责把从事务层来的事务映射到相应的对话.该层提供的 API 可以创建、终止和修改对话,还可以处理对话内的不同消息.

1.2.3 *事务层*　创建并管理事务对象.每个事务对象负责维持状态,收发消息和使用传输层重传消息.事务层也需要把从传输层到来的消息映射到相应的事务.该层提供收发不同类型消息的 API.

1.2.4 *传输层*　管理套接字(Socket)和网络连接,可以使用面向连接和无连接的协议,如 TCP 或 UDP 来传送数据.该层提供收发消息的简单 API.

1.2.5 *SIP/SDP 消息编码解析层*　SIP 请求和响应是基于文本的消息,该层处理 SIP 消息(包含 SDP 会话描述信息)的编码和解析.该层提供的 API 可以方便地操作 SIP 消息和 SDP 描述符的各个部分.对话层、事务层和传输层都要用到该层的功能实现它们的对外接口.

对话层、事务层、传输层是垂直方向上互相联系的 3 层.传输层提供最基本的收发消息的功能,对任何类型的消息提供的收发接口完全一样.事务层中,对于不同类型的 SIP 请求消息,提供不同的发送接口;收到不同的 SIP 请求消息后,提供不同的处理接口,另外还提供了收发响应消息的接口.事务层 API 的实现,要使用传输层中实现的功能.对话层中提供的 SIP,处理消息更加细分,其实现要使用事务层所提供的功能.

2　实现原理

该系统是在 DM642 的评估板上实现的,TI

DSP提供了一个实时操作系统DSP/BIOS,可实现实时调度、同步以及主机/目标系统通信和实时监测的应用.DSP/BIOS组件包括抢先式多任务内核、硬件抽象层、实时分析工具和配置工具[4].

文中的SIP信令模块建立在SIP协议栈的基础之上.一个完整的SIP协议栈包括核心协议栈和UA外围框架的实现.核心协议栈提供包括内容解析在内的一系列较低层的操作函数实现.一般以函数库的形式发布,其主要组成部分见图1.要实现完整的SIP UA功能,还必须提供核心协议栈的外包,即实现整个UA的外围构架.本实现的核心协议栈利用了GNU项目组的OSIP library,它是自由软件基金会(FSF)支持开发的一个开放源代码程序.其提供了SIP消息解析机制、有限状态机事务模型的处理、对话相关处理、SDP基本操作处理等较低层的操作函数实现,供SIP应用程序开发者使用.由于OSIP library是提供给Windows、Unix平台下的PC机或Vxworks实时操作系统下的嵌入式系统用的,所以欲在DSP/BIOS下使用,首先需进行移植,还要改变与操作系统有关的线程、信号量等实现方法,并要考虑在DSP/BIOS下网络开发的特点及特有的程序结构.

2.1 任务的创建与调度

基于在SIP library中使用了POSIX线程的概念,所以在DSP/BIOS下要用任务(task)及相关的函数来实现.如:TSK_create()创建线程;TSK_setpri()设置线程的优先级;TSK_exit()退出线程等.

互斥与同步机制的实现,需要用DSP/BIOS下的信号灯(Semaphores)及相关函数.如:SemCreate()创建信号量;SemDelete ()删除信号量; SemPend ()等待信号量以及SemPost()触发信号量等.

2.2 网络程序开发方法

针对网络开发提供了NDK (Network Developer's Kit)套件,NDK通过抽象的编程接口而与本地操作系统和底层的硬件层相隔离.本地操作系统被抽象为操作系统适配层(OS. lib),而定制的硬件通过硬件抽象层库(HAL. lib)来支持.这两个库用来连接DSP/BIOS栈和系统外设,作为NDK与它们之间的接口.NDK的5个主要组成部分见图2[5].

开发网络程序时必须注意以下几点:

1) 必须包含5个组成部分,即以下库文件:hal_eth_dm642. lib, hal_ser_stub. lib, hal_timer. lib, hal_userled. lib, netctrl. lib, nettool. lib, os. lib, stack. lib以及evmdm642bsl. lib.

2) 在CCS的配置文件中,必须包含初始化EVM板子的程序dm642. init.

3) TCP/IP协议栈的基本初始化过程为:① 调用NC_SystemOpen()初始化操作系统环境; ② 通过CfgNew()创建新配置; ③ 建立新配置,或通过CfgLoad()装载先前的配置; ④ 调用NC_NetStart()启动协议栈,加入3个回调函数指针给"start","stop"和IP地址变化选项.

经过初步初始化后,NC_NetStart()创建一个调用用户提供的"start"回调函数的线程,而"start"回调函数创建网络需要的线程,用户网络应用在此编写.正常情况下除非调用NC_NetStop(),否则网络不会停止.调用NC_NetStop()后,最初的NC_NetStart()线程调用用户提供的"stop"回调函数关闭所有操作,释放资源,但这时TCP/IP协议栈的函数都无法调用了.NC_NetStart()以NC_NetStop()调用的返回值返回.应用程序可再次调用NC_NetStart(),重启(reboot)TCP/IP协议栈.当系统准备好最终关闭时,在NC_NetStart()返回和会话结束后,调用CfgFree()释放配置句柄,最后调用NC_SystemClose()关闭系统.

4) 在使用套接字(sockets)时,必须调用fdOpenSession(),结束任务时调用fdCloseSession().

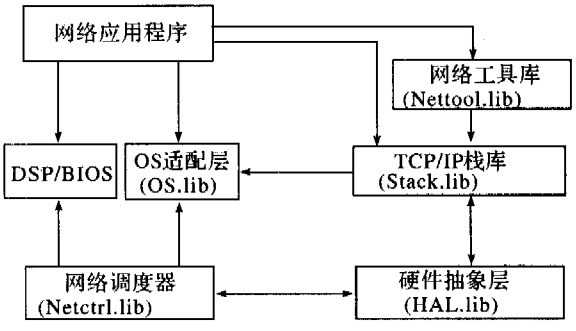

图2 TCP/IP协议栈控制流

Fig. 2 TCP/IP stack control flow

2.3 C/S(客户/服务器)体系的算法实现

客户/服务器模型是解决会聚点(rendezvous)问题的一种方式.它要求在任何一对进行通信的应用进程中,有一方必须在启动执行后(无限期地)等待对方与其联系,它减少了下层协议软件的复杂性.因为,下层协议不必对收到的通信请求作出响应.由于SIP终端既包含客户又包含服务器,必须同时实现客户端和服务器端的算法[6].

采用UDP传输、并发服务器算法同时处理多个连接请求,算法流程见图3.在具体的实现代码中,要用NDK提供的API代替相应的sockets API,主要是Network Tools Library和DNS两个

模块里的函数[7].

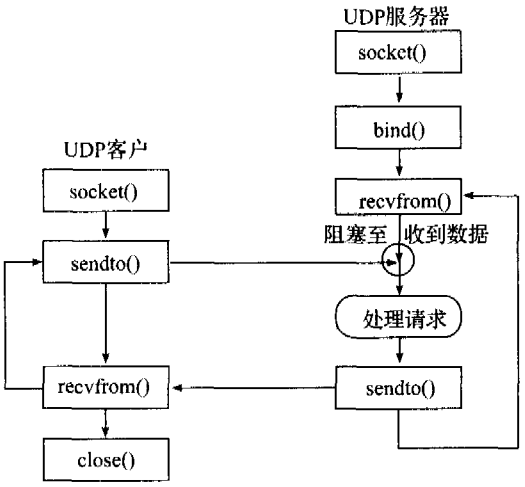

图 3　UDP 客户/服务器程序算法流程

Fig. 3　UDP C/S arithmetic flow

2.4　回调函数机制

从网络上接收到 SIP 消息时,采用回调函数的方式为应用程序提供接口.采用该方式,用户可方便地使用 SIP 协议栈,在回调函数中实现自己的应用逻辑,亦即实现自己的应用.另外,协议栈可以通过提供更多的回调函数,方便扩充对外接口.这些机制使协议栈非常灵活,方便使用.

当从网络上收到 SIP 请求消息时,协议栈先对消息进行解析,根据 SIP 消息的类型进行相应的处理.在实现协议栈内部的处理后,使用一个函数指针调用某个特定的函数,这个函数指针亦即回调函数指针.在应用程序中,需实现这个函数,在初始化协议栈时,把这个函数指针作为协议栈的初始化参数传入协议栈.这样,在协议栈对消息的处理过程中,就会"嵌入"用户所实现的应用.当接收到不同类型的 SIP 消息后,需要让用户进行不同的处理,于是协议栈针对不同的消息,提供给用户不同的回调函数.用户实现这些函数后,在初始化协议栈时将函数指针作为参数传入即可.在代码中结合应用重新实现了自己的 invite_accepted_cb、bye_cb、faillure_cb、invite_cb、informateve_cb、set_audio_offer、accept_audio_offer、read_audio_answer 这些回调函数,完全实现了 SIP 信令协议的功能.

2.5　音、视频的接入方法

初始化时,注册音/视频编解码器(G711/G723.1;H.263,MPEG4):register_codecs(),并将其加入到对应的链表中 list_append().利用 SIP 的 offer/answer 模型机制协商会议参数(即音、视频编解码器算法),这是通过 set_audio_offer(),set_vidio_offer(),accept_audio_offer(),accept_vidio_offer(),read_audio_answer(),read_vidio_answer()这些回调函数实现的.而媒体流的通信是在成功建立起 SIP 双方对话后开始的,发起 Invite 请求的主叫方(UAC)在收到 200 OK 应答后,用 Invite_accepted_cb()回调函数中启动音、视频数据流,而被叫方(UAS)在接受 Invite 请求后启动音、视频数据流,这样通信双方就可以进行双向媒体通信了.:audio_stream_start()和 vidio_stream_start()这两个函数启动了音、视频编解码处理流程:音视频数据采集->编码->RTP 打包发送(主叫方);和 RTP 接收拆包->解码->音视频数据回放(被叫方).

3　结　语

将 SIP 协议栈成功移植到了 TI TMS320-DM642 处理器上,并在其上实现了基于 SIP 的 VOIP 可视电话,达到了与 Windows 下的 SIP 软件终端的视音频互通.实践证明,SIP 协议结构简洁、易于扩展并适于 Internet 接入,有良好的应用前景.

参考文献:

[1] 雷正雄,朱晓民,廖建新. SIP 协议栈的设计与实现[J]. 现代电信科技, 2004,(3):17-20.

[2] Texas Instruments Incorporated. TMS320DM642 Technical Overview[Z]. USA: Texas Instruments Incorporated,2002.

[3] Rosenberg J,Schulzrinne H. SIP:Session Initiation Protocol.RFC3261[Z]. USA: IETF,2002.

[4] Texas Instruments Incorporated . TMS320 DSP/BIOS User's Guide[Z]. USA: Texas Instruments Incorporated ,2003.

[5] Texas Instruments Incorporated . Texas Instruments TMS320C6000 TCP/IP Network Developer's Kit User's Guide[Z]. USA: Texas Instruments Incorporated ,2003.

[6] Douglase Comer,David L Steviens. 用 TCP/IP 进行网络互联(第 3 卷)客户-服务器编程与应用(Linux/POSIX 套接字版)[M]. 赵刚, 林瑶, 蒋慧译. 北京:电子工业出版社,2001.20-50.

[7] Richard Stevens W. UNIX 网络编程(第 1 卷)套接口 API 和 X/Open 传输接口 API[M]. 施振川, 周利民, 孙宏晖,等译. 北京:清华大学出版社,1999.180-202;620-648.

(责任编辑:彭守敏)