嵌入式SIP语音网关实时传真模块的设计和实现①

张京磊 [1,2] 孙建伟 [1] (1.中国科学院 沈阳计算技术研究所 网络与通信实验室 辽宁 沈阳 110171;
2.中国科学院 研究生院 北京 100039)

摘 要: 传真通信作为有一种传统的传输静态图像的手段,是现代通信技术的重要组成部分。分析了IP传真(FoIP)的现状,比较了存储转发和实时传真两种实现标准,然后在基于SIP的嵌入式语音网关基础上,给出了T.38实时传真模块的纯软件方案的设计和实现。该模块主要由T.30 Soft Modem和T.38协议栈两部分组成,分别处理PCM数据和IFP包,二者之间的数据交互通过数据泵完成,具有低耦合度和高可移植性,已投入实际应用并取得较好效果。

关键词: 嵌入式语音网关; 实时传真; T.38; 会话初始协议; Blackfin

Design and Implementation of Real-Time Facsimile Module in Embedded SIP Voice Gateway

ZHANG Jing-Lei[1,2], SUN Jian-Wei[1]
(1.Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110171, China;
2.Graduate University of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: Facsimile, as a traditional transmission method of static image file, is an important part of modern communication technology. This paper analyses the status of facsimile over IP (FOIP), compares the Storage- and-Forward and Real-Time implementation standards, then based on embedded SIP voice gateway gives the design and implementation of the pure software T.38 real-time facsimile module. This module is composed of T.30 Soft Modem, processing PCM data, and T.38 protocol stack processing IFP packets. These two parts exchange data through DATAPUMP. This module has weak coupling and high portability, and has been put into practice.

Keywords: embedded voice gateway; real-time facsimile; T.38; SIP; Blackfin

1 引言

传真通信作为有一种传统的传输静态图像的手段,是现代通信的重要组成部分。IP传真,是指利用IP网络全部或部分代替PSTN进行的传真通信方式。

IP传真主要有有两种实现方式:存储转发(Storage and Forward)和实时(Real Time)。存储转发方式类似于发送电子邮件,发送方首先将传真数据发送到本地传真网关,然后该传真网关通过IP网络传送给远端传真网关,最后远端传真网关将传真发送到特定的接收终端。存储转发方式主要有以下不足:没有发送确认,同传统意义上的传真业务相差较大;传真数据需要在网关存储,带来了一定的安全性和保密性问题。实时方式是指发送和接收传真终端进行实时对话、交换数据和传真报文。国际电信联盟(ITU)制定了T.38标准[1],对在IP网络中进行实时三类传真所采用的通信方式、报文格式、纠错方式以及部分通信流程均作了一定的描述和规定。在实时传真方式中,发送和接收传真终端之间所进行的呼叫建立、训练、报文传输以及呼叫拆除始终是实时的,使用方式同传统PSTN的传真业务一致。

T.30为PSTN上的传真传输协议与规程[2],它对三类传真机在普通电话网上的通信流程所采用的信号

① 收稿时间:2009-10-10;收到修改稿时间:2009-11-22

格式、控制信令以及纠错方式都作了详细的描述和规定。

实时传真的应用环境如图 1 所示，三类传真机连接到我们实现的嵌入式语音网关，传真模块作为网关软件的一部分，负责实时传真的发送和接收。一个传统的三类传真机连接到嵌入式语音网关，通过 IP 网络向另一个语音网关发送传真，接收网关则向另一侧的传统三类传真机发起 PSTN 呼叫。采用 SIP 作为 T.38 呼叫控制协议。两个传真机之间建立起 T.30 连接，而语音网关之间建立的则是 T.38 连接。

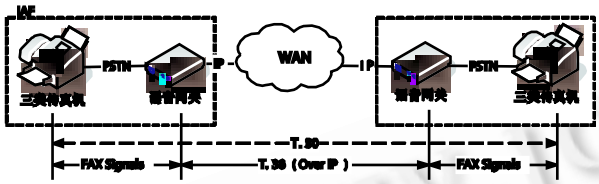

图 1　实时传真的应用环境

2　嵌入式语音网关简介

嵌入式语音网关采用 ADI 公司推出的 Blackfin ADSP BF532 处理器。这是集 MCU 和 DSP 于一体的 16 位定点 DSP，最高主频达 400MHz，具有 84kb 片内存储器，内建 2 个 MAC 运算单元和 4 个视频像素运算单元，峰值性能为 1600MIPS，专门设计用来满足要求较高计算能力并限制功率消耗的嵌入式音频应用需求[3]。

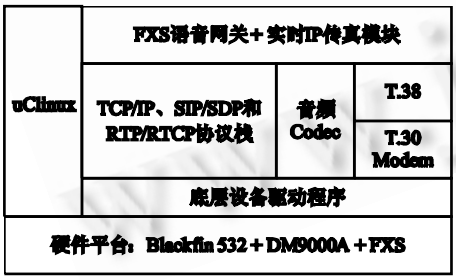

图 2　语音网关软件层次结构

存储设备包括 16M SDRAM、4M NOR Flash 和 64M NAND Flash。由两个 DM9000A 以太网控制芯片提供 10M/100M 以太网接口，实现网络数据的收发和路由。根据 Silicon Laboratories 公司提供的 QUAD PROSLIC 解决方案，以两组 Si3241+2 个 Si3203 组合的芯片组提供 8 个 FXS 接口。通过 CPU 对 FXS 接口电路芯片的控制，可以为外接话机或传真机提供直流馈路、振铃、摘机检测、铃流检测等功能。

软件方面，采用广泛应用于微控制领域的 uCli-nux 操作系统，分为底层驱动、协议栈和 Codec、应用程序三个部分，其中底层驱动主要包括 DM9000A 网卡驱动和 FXS 口驱动，分别负责 IP 数据包、模拟语音信号的收发，并向上层提供硬件访问接口；协议栈和 Codec 主要包括 TCP/IP、SIP/SDP、RTP/RTCP 协议栈和音频 Codec，SIP/SDP 和 RTP/RTCP 协议栈是语音网关和实时 IP 传真的实现基础，将基本操作封装成丰富的 API 供上层应用程序调用，而音频 Codec 目前已支持 PCMA、PCMU、GSM、Speex 和 G.723；应用程序则实现了接打电话和实时收发传真两大功能。

3　实时传真模块的设计

实时传真模块采用纯软件的实现方案，主要包括 T.30 Soft Modem 和 T.38 协议栈两大部分，分别负责处理 PCM 数据和 IFP 包，二者之间的数据交互通过数据泵完成转换和分发。为了降低耦合度并提高可移植性，整个传真模块完全封装，只对外提供经过精心设计的 API 接口，供上层应用程序调用。模块的输入输出数据为 PCM 语音信号和 IFP 报文。

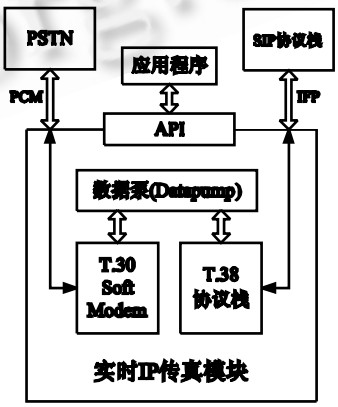

图 3　实时传真模块的结构

发送和接收传真的通信流程主要包括传真呼叫的建立和释放、兼容能力检测、机器工作状态的设定、通信线路的检测和监测、各种控制功能的确定以及在必要的时候进行传真操作的再次呼叫等。整个流程均

根据 T.30 和 T.38 协议的规定来完成。

采用 SIP(会话初始协议)作为承载协议。若建立 T.38 会话的协商失败或对方不支持 T.38，则尝试采用透传的方式在双方建立连接。所谓透传就是将传真的音频数据当作语音数据，以 PCM G.711A/U 的编码方式来进行传输。G.711A/U 和 PSTN 的带宽都是 64Kbps，因此是足够承载传真数据的。缺点是接收方会将传真音频当成普通 VoIP 音频处理，而由于少量丢包或错包并不会影响 VoIP 语音的质量，接收方一般不会采取严格的纠错校验机制，因而会造成传真数据的可靠性和完整性受到一定程度的影响。如果网络环境足够好，该方式也是可以接受的。

4 实时传真模块的实现

4.1 运行流程

传真模块的运行流程如图 4 所示，分为主程序、传真接收、传真发送和语音会话共四个部分，分别对应四个线程。其中，对 SIP 消息的封装、解析和响应等处理功能由主程序完成。主程序根据会话类型(语音或传真)的不同需要，调用相应的子程序进行处理，每个子程序对应一个线程。在会话建立之后并开始传输媒体时，还将单独创建媒体接收、媒体发送两个新的线程来进行媒体的处理，主要负责 FXS 驱动设备上的语音数据的读写，并向上层提供一致的接口，减少了上层模块对硬件的依赖，提高了传真模块的可移植性。

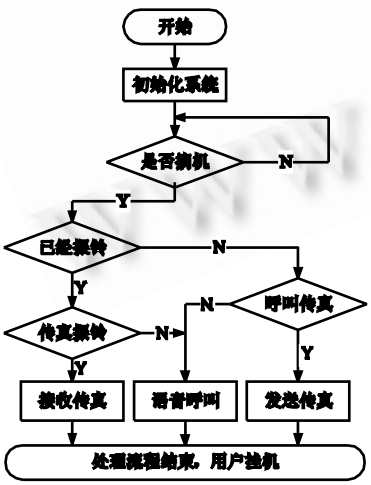

图 4 程序运行流程

4.2 T.38 协议主结构体和 IFP 包的定义

```
struct t38_core_state {
    /* IFP 包发送函数*/
    t38_tx_packet_handler    *tx_packet_
handler;
    /*指向需发送数据的不透明指针*/
    void *tx_packet_user_data;
    /* IFP 包接收函数*/
        t38_rx_data_handler    *rx_data_
handler;
    /*指向所接收数据的不透明指针*/
    void *rx_user_data;
    /*定义数据速率管理的方法*/
    int data_rate_management_method;
    /*定义使用的协议，如 UDPTL 或 RTP*/
    int data_transport_protocol;
    /*给出 T.38 协议版本*/
    int t38_version;
    /*给出信道所支持的最快数据速率*/
    int fastest_image_data_rate;
    ...
};
```

一个 IFP 包由“类型(TYPE)”和“数据(DATA)”两部分组成。“类型”单元描述 HDLC 帧的前导标志、传真训练、调制方式和速率，以及传真模拟信号(CED 和 CNG)；“类型”单元分为两种，其一是“T.30 指示符”(T.30-INDICATOR)，它指示前导标志、传真训练以及传真模拟信号；其二是“T.30 数据”(T.30-DATA)，它指示传送传真信号和报文的调制方式和速率。“数据”单元包含传真网关从传真终端接收到的传真信号和报文数据。“数据”单元由一个或多个字段组成，每一个字段有两部分：字段类型(Field-Type)和字段数据(Field-Data)。字段类型指出后随的字段数据的性质；字段数据包括传真通信中的使用 HDLC 传送的规程信号。

```
struct IFPPacket {
    int    t30_indicator;      /*T.30指示符*/
    int    t30_data;           /*T.30数据*/
    int    t30_field_type;     /*T.30域类型*/
    const  uint8_t  *msg;      /*指向消息内容*/
    int    len;                /*包长度*/
    uint8_t    *buf;           /*数据缓冲区*/
    uint16_t   seq_no;         /*序号*/
```

```
    ...
};
```

4.3 纠错协议 UDPTL 和冗余纠错机制的实现

为了保证传输的实时性，通常在 IP 层以 UDP 协议为基础。但 UDP 协议并不具备流控制、差错恢复等机制，所以 T.38 协议规定 UDPTL(用户数据包传输层)协议为基于 UDP 的 IFP 封装形式，在 UDP 保证实时性的基础上提供额外的纠错与重传机制[4]。

UDPTL 由 UDPTL 头部和 UDPTL 负载两部分组成。UDPTL 头部是指 UDPTL 序列号，一个以八位字节对齐的整形数字，用于标识封包以及封包中的原始数据域的发送次序，保证了包的有序收发。后者作为 UDPTL 负载，包含需要发送的 IFP 报文和冗余纠错或前向纠错信息。UDPTL 负载的第一个 IFP 称为主数据(primary)，承载实时数据，其后为用于纠错和丢包恢复、可选的附加数据(secondary)，承载用于封包校验与纠错的冗余信息(redundancy)或前向纠错(FEC)消息这两种附加信息。

结构体和相关接口定义如下：

```
struct UdptlPacket {
    /*保存 udp sicket 信息*/
    udp_socket_info *udptl_sock_info;
    /*保存原始数据*/
    unsigned char rawdata[MAX_LENGTH];
    /*用于差错校验*/
    int error_correction_scheme;
    int error_correction_entries;
    int error_correction_span;
    /*本地和远程终端的最大数据包尺寸*/
    int far_max_datagram_size;
    int local_max_datagram_size;
    int tx_seq_no;        /*发送序号*/
    int rx_seq_no;        /*接受序号*/
    int rx_expect_seq_no;  /*期望接收的序号*/
    /*接收和发送缓冲区*/
    udptl_fec_rx_buffer rx[UDPTL_BUF_ LEN];
    udptl_fec_tx_buffer tx[UDPTL_BUF_ LEN];
    ...
};
/*UDPTL 包的接收和发送函数*/
int udptl_rx_packet(struct UdptlPacket *s,
uint8_t *buf, int len);
    int udptl_tx_packet(struct UdptlPacket *s,
uint8_t *buf, uint8_t *msg, int msg_len);
```

图 5 给出了 IFP 被封装到 UDPTL 中的顺序。在同一个封包中，冗余信息和前向纠错消息是可选的。在线路协商过程中，两个 T.38 网关会彼此通告自身所支持的纠错模式，然后在连接时选择二者都支持的模式完成实际传输过程中的纠错工作。

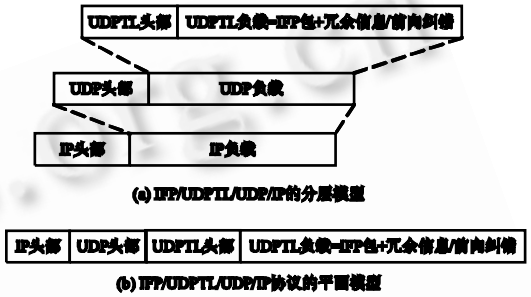

图 5　顶层 UDPTL/UDP/IP 的封包结构

本文实现的传真模块支持冗余信息的纠错方式。如当前需要发送的主 IFP 报文的序列号为 M，则序列号为 M-1 至 M-N-1 的 IFP 报文将作为封装的冗余数据被一并发送出去，因此每一个 UDPTL 封包都含有一个当前 IFP 数据并附加前 N 个 IFP 数据。这样即使 N 个连续数据域全部丢失也是可以恢复的，缺点是只能发送具有连续序列号的 IFP 封包数据块。在编码方面，实现了一个先进先出的缓冲区循环队列，保存当前及前 N 个 IFP 报文。在信道质量较好时，N 的取值一般为 2 或 3，随着信道质量的降低而不断增加。

4.4 对 SDP 的扩展

SDP 协议是 RFC2327 中定义的多媒体会话描述协议。它描述了从会话信息到可能的会话参加者的格式，包括一个或多个媒体流的相关参数说明，还包括与会话整体相关的通用信息。

根据 T.38 协议的附录 D，扩展 SDP 属性完成对实时传真的媒体和能力的协商，主要有 T38FaxRateManagement、T38FaxVersion、T38MaxBitRate、T38FaxFillBitRemoval、T38FaxTranscodingMMR、T38FaxTranscodingJBIG、T38FaxMaxBuffer、T38FaxMaxDatagram、T38FaxUdpEC 和 T38VendorInfo 等[5]。

首先传真发送网关向接收网关发出一个 INVITE 消息，请求建立语音连接。连接建立之后，若发送传真则发送含有 T.38 SDP 的 ReINVITE 消息，在传真发送网关和接收网关之间建立起 T.38 连接，该连接将之前建立的语音连接替换。最后发送 BYE 消息断开连接。

建立 T.38 连接时的 SDP 实例如下:

```
v=0
o=SIPGW 61579 1 IN IP4 192.168.139.152
s=SIP CALL c=IN IP4 192.168.139.152
t=0 0
m=image 20000 udptl t38
a=T38FaxVersion:0
a=T38MaxBitRate:14400
a=T38FaxRateManagement:transferredTC
a=T38FaxUdpEC:t38UDPRedundancy
```

5 结论

本文设计并实现了一个基于 SIP 的在嵌入式语音网关上运行的实时 IP 传真模块。采用纯软件的实现方案，进行分层和模块化设计，主要由 T.30 Soft Modem 和 T.38 协议栈构成，具有较低的耦合度和较好的可移植性，在实现方面充分使用多线程，最大程度的利用了嵌入式处理器的性能。

实时 IP 传真模块作为 SIPSYS 的蓝蜻蜓系列 SIP/IMS 语音网关的一部分，在实际应用中取得了较好的效果，目前该网关产品已进入批量生产阶段。

参考文献

1 ITU-T Recommendation T.38: Procedures for real-time Group 3 facsimile communication over IP networks. 2007,4.

2 ITU-T Recommendation T.30: Procedures for document facsimile transmission in the general switched telephone network. 2005,9.

3 Devices A, Inc. ADSP-BF531/BF532/BF533 Data Sheet. Analog Devices, Inc.2008,11.

4 刘辉,刘玉贵,张权利.实时 IP 传真纠错协议 UDPTL 的应用.现代电信科技, 2002,8:15－19.

5 IETF Internet Draft: SIP Real-time Fax Call Flow Examples and Best Current Practices. 2002, 4.