

基于 Windows CE 的 SIP 软件电话的设计与实现

Design and Implementation of a SIP Soft Phone Based on Windows CE

朱承发 (桂林电子科技大学 信息与通信学院 广西 桂林 541004)

摘要: 本文提出了一种基于 SIP 协议的软件电话的软件结构和设计实现方案。该方案以嵌入式 Windows CE 为平台, 中间件采用开源的 SIP 协议栈 oSIP/eXosip, 通过协议栈的移植和在协议栈之上应用程序的开发, 实现了 SIP 软件电话。测试结果表明, 该软件电话在布置 Windows CE 的 PDA 上具有良好的语音通话质量。

关键词: SIP 协议 oSIP/eXosip 协议栈 Windows CE 软件电话

VOIP 的迅速普及, 出现了越来越多的 VOIP 的固定电话终端。随着 VOIP 的进一步发展, 为了满足移动用户 VOIP 通信的需要, 无线 VOIP 通讯将是未来的发展趋势。

现在手持 PDA 用户越来越多, 都具有无线上网功能, 如果能开发一个在 PDA 上运行的 VOIP 软件电话, 就可以满足这些 PDA 用户的无线 VOIP 通信的需求, 具有非常重要的实际意义。由于嵌入式 Windows CE(WinCE)是 PDA 最流行的操作系统, 而目前大多数 VOIP 终端都采用 SIP 协议, 因此, 本文提出了一种 Windows CE 平台下基于 SIP 协议的 VOIP 解决方案。该方案以嵌入式 Windows CE 为平台, eMbedded C++ 4.0 为开发工具, 中间件采用开源的 SIP 协议栈 oSIP/eXosip, 通过进行协议栈的移植和在协议栈之上应用程序的开发, 实现了 SIP 软件电话。

1 设计方案

SIP 软件电话的结构包括呼叫控制和语音通信两大模块。呼叫控制由 SIP 信令来完成, 主要实现呼叫的建立、修改和拆除; 语音通信模块由音频数据接口, 音频编解码和 RTP 传输三个子模块组成, 实现语音的采集, 编码, 传输和播放。

软件结构如图 1 所示。

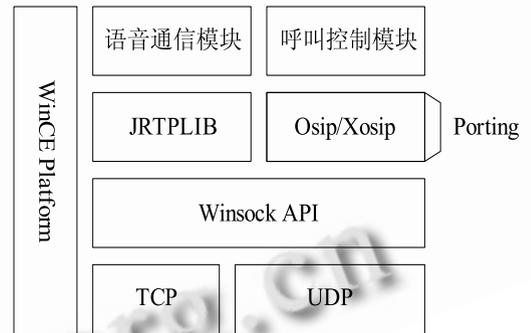


图 1 软件结构

音频数据接口模块实现语音的采集和播放, 采用 Windows CE 的低层音频服务, 因为低层音频服务中的回调机制为其提供了很大的方便; 音频编解码采用的是 G.729A 语音压缩技术, 它属于低比特率语音编码, 适合语音在 IP 网络传输, 语音采用 RTP 包传输。

对呼叫控制模块, 直接移植开源 oSIP/eXosip 协议栈到 WinCE 上, 通过调用协议栈的 API 函数, 实现对呼叫的控制。RTP 传输用开源 JRTPLIB, JRTPLIB 也需要在 WinCE 进行移植。

2 具体实现

基于上述方案，具体实现包括协议栈移植、呼叫控制和语音通信三部分。

2.1 协议栈移植

oSIP/eXosip 主要支持 Win32 及 Linux、VxWorks 等一些嵌入式系统，将 oSIP/eXosip 移植到 WinCE 上，可以在 Win32 基础上进行。WinCE 可以兼容大部分的 Win32 API，在 WinCE 上移植 oSIP/eXosip 主要有 3 种情况：与 Win32 兼容的 API，与 Win32 不兼容 API 和不同的头文件支持和预定义。与 Win32 兼容的 API 只要修改条件编译选项，与 Win32 不兼容 API 可以用其他的 WinCE 上的 API 来代替并作相关修改，另外，WinCE 和 Win32 的某些头文件和预定义也不同，也需要进行相关修改工作。表 1 给出了移植的一些示例。

在 WinCE 上对 JRTPLIB 进行移植与 SIP 协议栈的移植类似，注意 WinCE 下的 RTP 装载数据包最大 2K，和 PC 上的 64K 不同。

表 1 oSIP/eXosip 移植示例

	Win32	WinCE
兼容	#if defined WIN32(如Win32用_sprintf函数，其他支持的系统用sprintf函数) #if !defined WIN32(如除Win32外，其他支持的系统用strcasemp函数,Win32用_stricmp函数)	条件编译改成 #if (defined WIN32 defined WIN32_WCE) (WinCE也支持_sprintf函数) 条件编译改成 #if (defined WIN32 && !defined WIN32_WCE) (WinCE也支持_stricmp函数)
不兼容	time 函数	不支持time函数，用GetSystemTime函数加一些计算来代替。
头文件和预定义	#include <process.h> #include <sys/types.h> #define HAVE_SYS_TYPES_H 1	#include "Winbase.h" #define _beginthreadex CreateThread 等等

2.2 呼叫控制模块

呼叫控制模块用来向代理服务器注册以及进行会话的建立、修改和终止,实现对呼叫的控制。因此呼叫控制模块分为注册和呼叫两部分,都是通过底层的 SIP 协议栈来完成的。上层应用程序调用 SIP 协议栈提供的 API 函数，通知协议栈进行相应的操作，协议栈将底层检测到的事件以消息的形式报告给应用层，应用层收到 SIP 事件后作相应处理。

2.2.1 注册

注册的实现过程首先是由客户端向代理服务器发出 REGISTER 注册请求，并等待服务器返回响应消息。客户端必须在收到前一个 REGISTER 请求的最终响应

之后或者前一个 REGISTER 请求超时的情况下，才能发送一次新的注册请求，注册流程和对应的 SIP 协议栈 API 函数调用或协议栈的事件通知分别如图 2(B)和 (A)。

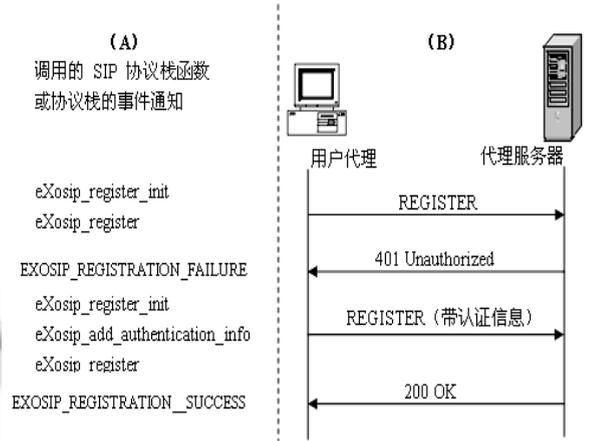


图 2 注册流程及对应的 SIP 协议栈函数调用或事件通知

2.2.2 呼叫

本设计的 SIP 软件电话有两种呼叫模式：注册呼叫和 Peer To Peer 呼叫。如果是呼叫一个公网的 SIP 用户或是同一个私网的 SIP 用户之间进行呼叫，就可以直接找到对方，而不必要通过代理服务器。下面以注册呼叫为例，终端注册成功后就可以进行呼叫。一个通过代理服务器的成功呼叫流程如图 3 所示。

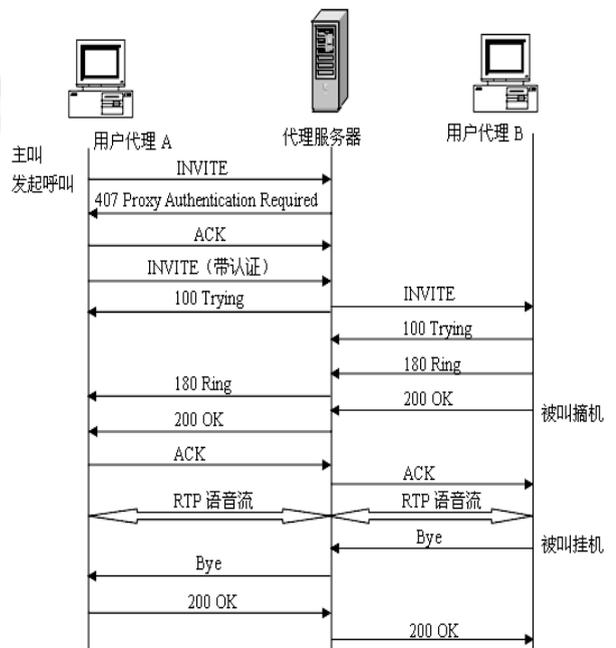


图 3 语音通信程序流程

oSIP/eXosip 对消息的报告有 2 种工作模式：事件模式和回调模式。本设计采用事件模式，通过函数 eXosip_event_wait 定时轮询获取 SIP 事件。SIP 信令的实现如表 2。

表 2 SIP 信令的实现

SIP信令	主叫	方向	被叫
invite	eXosip_build_initial_invite、eXosip_initiate_call	↓ →	EXOSIP_CALL_NEW (新呼叫到来) ↑
Trying	EXOSIP_CALL_PROCEEDING (对方正在处理)	↑ ←	——
Ringing	EXOSIP_CALL_RINGING (对方正在响铃)	↑ ←	eXosip_answer_call(180) ↓
200 OK	EXOSIP_CALL_ANSWERED (对方摘机)	↑ ←	eXosip_answer_call(200), 被叫摘机 ↓
ACK	——	→	EXOSIP_CALL_ACK ↑
Bye	EXOSIP_CALL_CLOSED (对方挂机)	↑ ←	eXosip_terminate_call, 被叫挂机 ↓
200 OK	——	→	——

注：↓ 表示向下调用协议栈API函数，↑ 表示协议栈向上报告事件给应用层；
→ 和 ← 分别表示主叫向被叫方向和被叫向主叫方向；
—— 表示由协议栈底层自动完成

2.3 语音通信模块

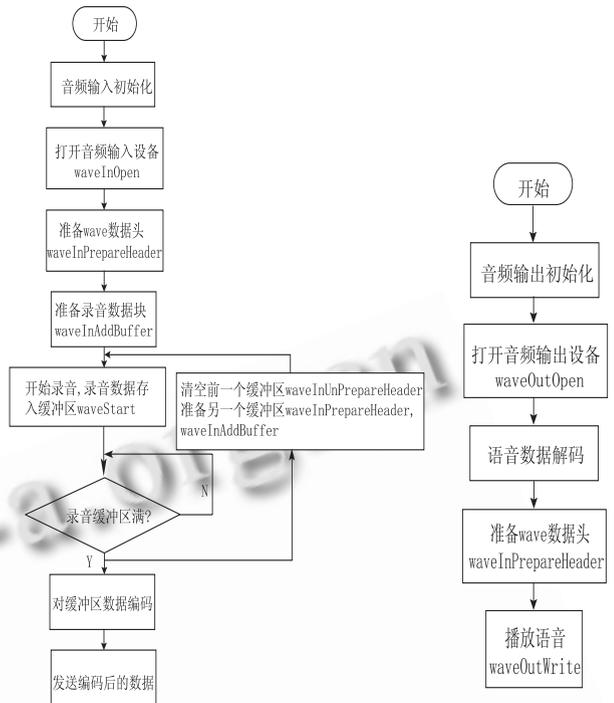
语音通信模块主要是对语音进行采样、编码压缩等处理,并封装成能在 IP 网络上传输的 RTP 包,接收端收到后解码,播放。

语音通信模块程序流程如图 4。

RTP 传输模块通过使用开源的 RTP 协议栈 JRTPLIB 来实现,我们只需调用 JRTPLIB 的少数几个函数就可以了。在通过 SIP 信令建立会话后,准确地说是在主叫得知被叫应答即获取到事件 EXOSIP_阶段 CALL_ANSWERED 后及被叫收到主叫的 ACK 消息即获取到事件 EXOSIP_CALL_ACK 后分别启动语音通信模块。会话结束(由呼叫模块控制)则关闭语音输入输出设备,结束语音流传输。

3 结束语

本文提出了一种 Windows CE 平台下基于 SIP



(a)语音采集、编码、发送流程

(b)语音解码、播放流程

图 4 语音通信程序流程

协议的 VOIP 软件电话终端的解决方案,并对方案的各模块及实现进行了详细的介绍。该软件电话实现了基于 Windows CE 操作系统的 PDA 用户的无线 VOIP,通话话音清晰,延时较小,具有很好的实用价值。作为后续工作,还将在 SIP 软件终端中增加视频功能。

参考文献

- 1 Rosenberg J, Schulzrinne H, Camarillo G. IETF RFC 3261.SIP:Session Initiation Protocol,June 2002.
- 2 田延丽,余厚全.基于 SIP 协议的 IP 电话服务器的设计.微计算机信息,2007,23(3):138 - 140.
- 3 朱小军.SIP 协议在 WiFi 智能手机中的应用与实现.成都:电子科技大学,2006.