

# 基于 G. 729A 语音实时传输系统的实现<sup>\*</sup>

## Implementation of Voice Real Time Transport System Based on G. 729A

王海霞,覃团发,刘运毅,陈跃波

Wang Haixia, Qin Tuanfa, Liu Yunyi, Chen Yuebo

(广西大学计算机与信息工程学院,广西南宁 530004)

(Coll. of Comp. & Info. Eng., Guangxi Univ., Nanning, Guangxi, 530004, China)

**摘要:**简述 G. 729A 编码和解码的原理,以及编码器和解码器的简化流程。语音实时传输系统由录音压缩、网络通信和解压播放 3 个模块组成,实现 G. 729A 编码器与解码器的封装、语音录制与播放、流式套接字和线性同步等功能。系统测试结果表明,基于 G. 729A 语音实时传输系统具有较好的语音通信效果。

**关键词:**语音实时传输 语音编码 套接字编程 G. 729

中图分类号:TN915

**Abstract:** A voice real time transport system based on G. 729A is developed. The construction, function modules, flow chart and implement methods of this system are revealed. Test results show that the system has good performance in voice communication.

**Key words:** voice real time transport, speech encoding, Windows Sockets, G. 729

G. 729 是国际电信联盟于 1996 年制定的语音编码标准,全称是“使用共轭结构代数码本激励线性预测的 8 kbit/s 语音编码”<sup>[1]</sup>。G. 729A 是 G. 729 的简化版本<sup>[2]</sup>,复杂度仅为 G. 729 的一半,两者在编解码器的比特流一级完全兼容。G. 729A 的码率为 8 kbit/s, MOS 分在 4.0 左右,达到全质语音标准。由于 G. 729A 编解码器具有较高的带宽利用率,且语音质量与 32 kbit/s 的 ADPCM 相当,算法时延仅为 15ms,所以迅速在许多原来采用 ADPCM 语音压缩算法的系统中得到运用,并被广泛应用于 DSVD、电视会议、IP 电话等领域<sup>[3]</sup>。

ITU-T 公布了 G. 729A 定点算法的标准 C 代码,利用该 C 代码可以方便的进行硬件开发。但是该 C 代码并不适用于 G. 729A 的软件实现,本系统基于 G. 729A 的算法,对编解码过程进行 VC++ 的面向对象封装,结合线程同步机制,并利用网络通信技术实现了局域网内的语音实时传输。

### 1 G. 729A 编解码算法<sup>[4]</sup>

在编码端,首先对输入的模拟语音信号进行电

话带宽滤波,然后以 8 kHz 的频率对其进行采样,再将其转换为 16 bit 线性 PCM 码,作为编码器的输入。编码器以帧为单位处理语音,1 帧为 10 ms,包括 80 个语音样本。编码器对每一帧输入语音信号进行分析,提取其中的 CELP 模型的参数,对这些参数进行编码和传输。在解码端,对接收到的参数进行解码,重新获得激励信号和合成滤波器,重构语音信号并对其进行后处理。图 1 和图 2 分别给出 G. 729A 编码和解码原理框图。

### 2 系统流程和功能

#### 2.1 G. 729A 编解码器流程

ITU-T 提供的 C 代码将编码和解码分为 2 个独立的程序:coder.c 和 decoder.c,编码实现将输入的 PCM 语音文件压缩为 G. 729A 的比特流文件,解码实现将输入的 G. 729A 的比特流文件解压为 PCM 文件。图 3 和图 4 分别是编码器和解码器的简化流程。

#### 2.2 系统功能模块与整体流程

本系统按功能分为三大模块:录音压缩、网络通信、解压播放。录音压缩模块包括录制语音和编码器

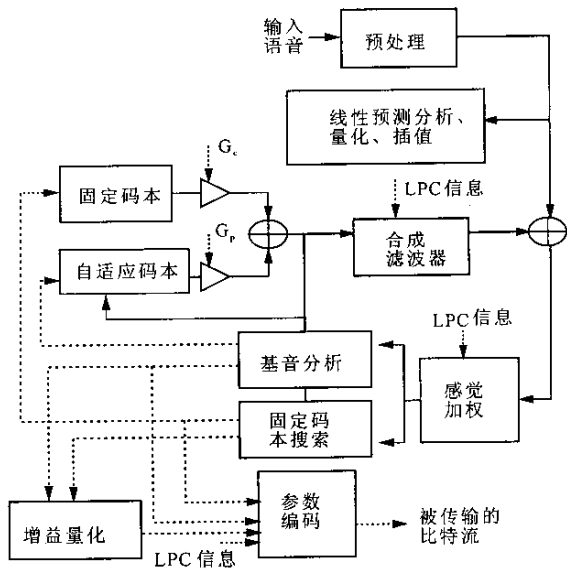


图 1 G. 729A 编码原理

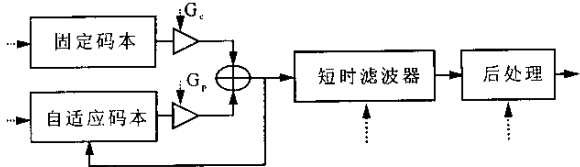


图 2 G. 729A 解码原理

2 个部分,其中录制语音从声卡采集数据并转换为 PCM 码,再通过编码器将其实时压缩为比特流数据。网络通信模块完成服务端的数据发送和客户端的数据接收。解压播放模块包括解码器和播放语音 2 个部分,其中解码器把比特流数据解压成 PCM 码,再由播放语音实现从声卡中播放。图 5 给出对每一帧语音信号进行处理时各个模块之间的关系。

系统整体流程如图 6 所示。

### 3 实现方案

#### 3.1 G. 729A 的 VC++ 面向对象封装实现

本系统采用 Visual C++ 6.0 为开发工具,在程序中将 G. 729A 编码器与解码器封装为 1 个类: ITUG729A,提供 4 个公有方法:①初始化编码器 (Init \_ Encoder());②编码处理 (Encoder (short \* speech, unsigned char \* bitstream));③初始化解码器 (Init \_ Decoder ());④解码处理 (Decoder (unsigned char \* bitstream, short \* speech))。其中编码处理与解码处理的输入参数 speech 是一帧 PCM 语音数据(大小为 80 的 short 数组),bitstream 是一帧 G. 729A 编码比特流(大小为 10 的 unsigned char 数组)。在进行编码、解码前必须调用初始化方法。

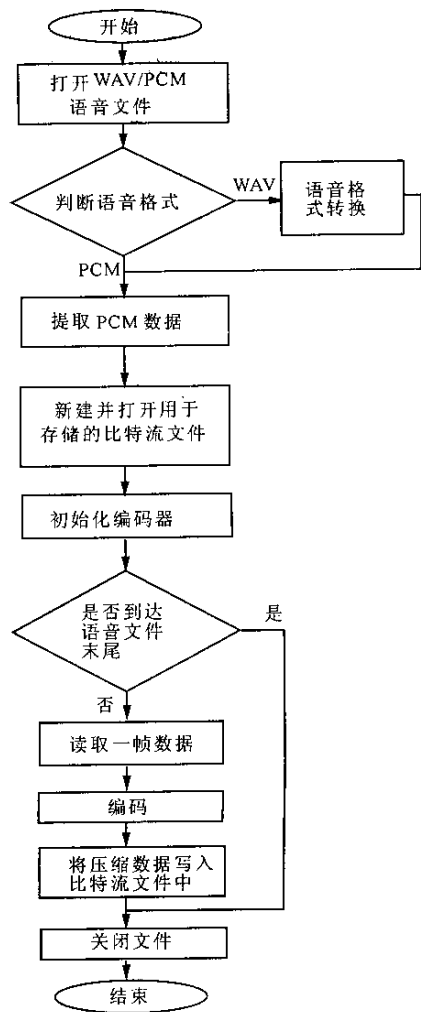


图 3 编码器流程

系统将封装好的 ITUG729A 这个类构造成为 1 个 DLL(动态链接库),提供上述 4 个公有方法为导出函数,以便不同的扩展应用(如:IP 电话系统,多媒体网络传输系统,视频会议等)能够方便的加载这个 DLL 并调用它的导出函数。

#### 3.2 语音的录制与播放

本系统对于语音的录制与播放采用 Windows 低级音频函数,因为它可以直接与音频驱动程序交互,通过窗口消息或回调函数来管理音频数据块的记录和播放,而且提供了与设备无关的接口。

录制是播放的逆过程,在此不再赘述,涉及的主要函数包括 waveInOpen(),waveInPrepareHeader(),waveInAddBuffer(),waveInStart()等。

#### 3.3 套接字网络编程

Windows Sockets 支持 2 种有效的套接字类型:流式套接字和数据报套接字<sup>[5]</sup>。流式套接字是可靠的、双向通信的数据流,提供 1 个面向连接、持续的

数据传输服务,无重复的发送且按发送顺序接收,内设流量控制,避免数据流超限,数据被看作是字节流,无长度限制,因为它使用了TCP,所以流式套接字可以达到高质量的数据传输。数据报套接字提供1个无连接服务,数据包是以独立包形式被发送的,不提供无错保证,数据可能丢失或重复,并且接收顺序无序。

由于TCP考虑到了所有传输问题,它提供完全可靠性<sup>[6]</sup>。TCP验证数据的到达,对未到达的报文段会自动进行重传;它还计算数据的校验和,以保证数据在传输过程中没有损坏;它使用序号确保数据按顺序到达并自动忽略重复的分组;它提供流控制

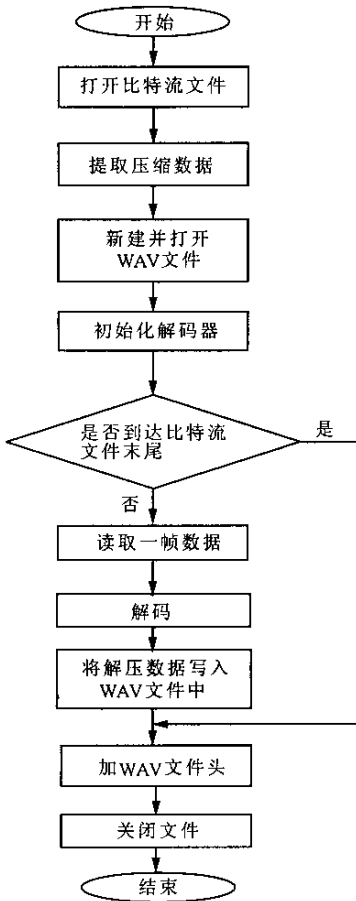


图4 解码器流程

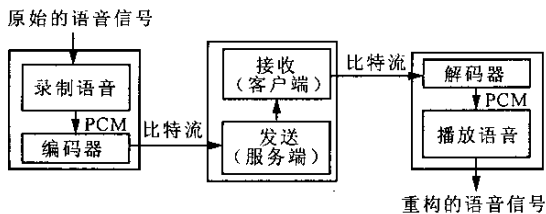


图5 模块之间的关系

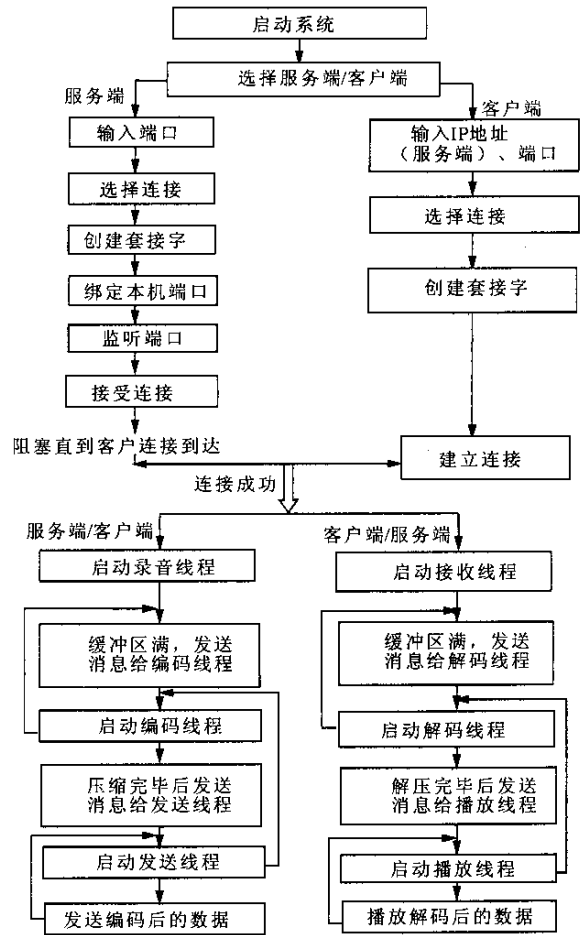


图6 系统流程

以确保发送方发送数据的速度不超过接收方的承受能力;最后,在下层网络因某种原因无法运作时,TCP会通知发送方。相反,使用UDP的客户和服务器在传输可靠性上没有任何保证。当客户发送请求时,这个请求可能丢失、重复、延迟或者交付失序,服务器发回给客户的响应也可能出现类似情况,因此使用UDP的客户或服务软件中必须包含检测和纠正这些错误的代码。通常差错只在1个广域网中通信时才会发生,在局域网看似工作得很好的应用软件如果放到广域网中使用,就可能失败或产生不正确的结果,这是因为在局域网中分组很少甚至从不被延迟、丢失或者交付失序。

考虑到使用UDP在非本地环境中需要进行额外的可靠性处理,语音系统采用的是流式套接字用以实现网络通信模块。

本系统通过使用MFC类库中的CSocket类直接进行数据通信,利用CSocketFile类和Archive类实现。实现步骤如下:

服务器: Construct -> Create -> Bind ->

Listen—>Accept—>Send—>Close;

客户机:Construct—>Create—>Connect—>

Receive—>Close。

### 3.4 线程同步

由于语音传输的实时性要求很高,本系统采用发送消息的方式来实现各个线程(录音、编码、发送、接收、解码和播放)之间的同步,具体流程参见图 6。限于篇幅,这里不展开说明。

### 3.5 实现程序的简化流程

(1) 新建一个 CNewSocket 类,父类为 CSocket。

```
class CNewSocket:public CSocket{...};
```

构造一个 CNewSocket 对象。

```
CNewSocket * m_ServerSocket;m_
```

```
ServerSocket=new CNewSocket;//服务端
```

```
CNewSocket * m_ClientSocket;m_
```

```
ClientSocket=new CNewSocket;//客户端
```

(2) 使用这个对象的 Create 成员函数产生一个 socket 句柄。

```
m_ServerSocket—>Create(m—Port);//服务端
```

端

```
m_ClientSocket—>Create();//客户端
```

(3) 服务端套接字调用 Listen 开始监听来自客户端的连接请求,构造一个新的 CNewSocket 对象,收到连接请求后调用 Accept 接受请求,建立连接。

```
m_ServerSocket—>Listen();m_ClientSocket
```

```
=new CNewSocket;
```

```
m_ServerSocket—>Accept(* m_
```

```
ClientSocket);
```

客户端套接字调用 Connect 与服务端套接字进行连接。

```
m_ClientSocket—>Connect(m_IPAddress,m
```

```
_Port);
```

(4) 构造一个 CSocketFile 对象。

```
CSocketFile * m_file;
```

```
m_file=new CSocketFile(m_
```

```
ClientSocket);//服务端
```

```
m_file=new CSocketFile(m_
```

```
ClientSocket);//客户端
```

(5) 构造一个 CArchive 对象用于服务端与客户端之间的数据传递。

```
CArchive * m_arOut;CArchive * m_arIn;
```

```
m_arOut=new CArchive(m_file,CArchive::store);
```

```
m_arIn=new CArchive(m_file,CArchive::load);
```

```
* m_arOut<<send[i];//发送数据
```

```
* m_arIn>>recv[i];//接收数据
```

(6) 通信完毕后,销毁 CArchive、CSocketFile 和 CNewSocket 对象。

```
delete m_arOut;delete m_arIn;delete m_file;
```

```
delete m_ServerSocket;delete m_
```

```
ClientSocket;
```

## 4 结束语

本系统根据 ITU-T 提供的 C 代码,对 G. 729A 编码解码过程进行面向对象封装,利用线程同步技术,并选用流式套接字进行网络通信编程,在局域网中的实际测试结果表明,系统具有良好的话音实时通信效果,可以方便的扩展应用到各种类型的通信软件。

参考文献:

- 1 ITU-T. Recommendation G. 729 Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear-prediction,1996.
- 2 ITU-T. Recommendation G. 729 Annex A. Reduced complexity 8 kbit/s CS-ACELP speech codec,1996.
- 3 颜彦,陈健. ITU-T G. 729A 语音编解码器的实时实现. 数据采集与处理,2001, 16(4):408~411.
- 4 黄永峰,等编著. IP 网络多媒体通信技术. 北京:人民邮电出版社,2003. 93~94.
- 5 贾斌,等编著. 网络编程技巧与实例. 北京:人民邮电出版社,2001. 7~10.
- 6 Douglas E Comer 著. 用 TCP/IP 进行网际互联. 第 3 卷. 客户-服务器编程与应用. 赵刚,等译. 北京:电子工业出版社,2001. 8~11.

(责任编辑:黎贞崇)