

P2P下的语音聊天软件实现

许 劲, 杨秀平

(长沙学院 计算机科学与技术系, 湖南 长沙, 410003)

摘 要: P2P是一种新型的分布式计算模式, 节点间可以交换信息. 通过对P2P技术下的语音通信原理的分析, 详细地阐述了P2P语音传输过程中应解决的一些关键性技术. 利用Java实现了语音聊天软件的开发, 给出了一个相应的简单实例.

关键词: P2P; 语音通信; Java

中图分类号: TP311

文献标识码: A

文章编号: 1672-7304(2005)02-0066-03

随着网络技术的快速发展和计算机技术的不断提高, 以网络为平台的各种多媒体音频和视频传输技术已经得到广泛的应用. 网络语音传输的基本原理是对模拟语音信号经过模数转换, 进行编码压缩后, 按一定的打包规则将压缩帧转换成 IP 数据包通过数据网进行传输, 在目的地经过数据解压、数模转换复原成语音, 从而达到语音通信的目的. 目前网络语音传输的应用方案一般有 2 种: 一是采用的客户/服务器 (C/S) 模型应用于数据广播的多对一传输, 其代表就是网络 IP 电话. 这种技术是使用嵌在网页上的 Java Applet 程序, 来实现拨号、通话等等基本的网络电话功能; 另一种方案则是点对点 (Peer-to-Peer) 模式下一对一的音频信号数据的传输. 相比之下, P2P 技术的特征之一就是弱化了服务器的作用, 甚至取消服务器, 允许处于网络上的任意两台计算机即扮演客户的角色, 又扮演服务器的角色, 实现对等访问. P2P 模式在数据发布、数据更新等方面要优于 C/S 模式. 所以在这种模式下的网络语音已经受到了工业界和学术界的关注.

1 存在的问题

在网络上点对点传输音频信号方面存在的问题主要可以归纳为以下几点.

1.1 双方之间的网络连接

要进行点对点的音频数据的传输, 首先就要建立数据连结. 在常用的通讯协议中, TCP 是使用最广、相对可靠的一种, 所以用于不允许数据丢失的应用中. 而 UDP 则偏向应用于处理速度要求较快、数据传输可靠性要求不是很高的应用上, 如数据广播. 通信协议的选择取决于我们所要做的应用的类型和要求. 因此怎样建立网络连接, 来保障稳定的接收和发送音频信号的数据流是关键.

1.2 音频信号的采集以及回放

在进行音频信号的采集中我们必须考虑到采样率的问题, 声音信号的采样率有 8 kHz、16 kHz、32 kHz、44 kHz 等, 每种数据采样率产生的数据量都不一样. 毫无疑问的

是, 越高的采样率越清晰, 但产生的数据量也越大, 所以这就要求我们所选择的采样率必须和网络的带宽相匹配.

1.3 音频数字信号的编码与解码

如果直接把采集到的音频信号数据流在网上进行传输, 它所占有的带宽也是十分大的. 以 8 kHz 的采样率采集 14 位的音频数据为例, 就可以得: $14 \times 8000 = 112,000$ (bps) or 112 k (bps)

从中我们可以看出直接传输所采集到的音频数据, 每秒需要向网络中发送 112k b 的数据. 所以从节省带宽缩减成本的角度考虑, 我们很有必要对这样的数据进行压缩. 对多媒体信号的压缩有许多格式可供选择, 如 mp2、mp3、GSM 等等. 同样, 在这里也存在一个对压缩格式进行选择的问题. 考虑到音频数据传输的及时性, 对传输的音频数据质量的要求, 各种压缩格式的压缩比率以及进行压缩和解压缩所要耗费的系统资源等方面问题, 选择合适的压缩格式就显得尤为重要.

2 解决方案

2.1 网络语音通信中的关键技术

网络语音传输总的目的是既要保证通话服务质量又要尽可能的减少带宽资源. 整个语音传输的过程为: 发送方先采集发话方声卡中的语音, 将模拟信号转换为数字数据后, 存入计算机内存中, 再通过语音压缩算法的运算将数据进行压缩, 最后打包成 IP 数据包发送出去. 接收方接收到 IP 数据包, 先解析 IP 数据, 接着采用解压缩算法将数据进行解压缩, 然后将解压缩后的数字数据转换为模拟信号. 在这一流程中, 语音的采集、数据的压缩和解压缩、数据的传输都是关键的技术. 其中采样率的大小直接决定语音的清晰度和采集的原始数字数据量的大小, 是网络语音通信要解决的首要问题. 音频数字压缩和解压缩在语音通信中占有很重要的地位, 直接影响到数据传输的效率和通话的质量. 数据的传输受限于网络带宽, 目前还未找到很好的解决方法.

收稿日期: 2005-03-15

作者简介: 许 劲 (1978-), 男, 湖南长沙人, 助教, 主要从事计算机网络技术研究.

2.2 网络语音通信实现

就前面提出的 3 个问题讨论解决的办法。

(1) 双方之间的网络连接问题的解决。Java 在网络连接上有其独特的优势，因为 Java 提供了丰富的网络类库的支持，可以轻松编写多种类型的网络通信程序。笔者开发的软件中就使用了 TCP/IP 协议，通过 Java 的 Socket 类进行编程就成功的实现了点对点的连接。

(2) 音频信号的采集和回放以及音频数字信号的编码与解码问题的解决。在解决这两个问题的时候，笔者是采用的 Answer Machine 的源代码。在这个程序代码中，有几个解决我们问题所需要的类，而且作者将这些类封装的很好，我们基本不需要做什么改动，只需要屏蔽其中的调试信息的输出就行了，更可贵的是它还封装了几种常见的音频格式。其中的 GSM 格式可以将 128 kbps 的音频数据流（16 bit 通过 8 kHz 的音频采样）压缩为 13 kbps 的音频数据流，非常适合语音信号的传送。解释几个重要概念：

① AMAudioFormat 类

AMAudioFormat 类(在 AMAudioFormat.java 文件中封装)封装了 CD、FM、TELEPHONE、GSM 这四种质量的音频格式的参数，使用起来也非常简单，在使用 Java Sound API 时就不用自己去写那些复杂的代码了，但为了明白 Java Sound API 的原理，需要对它的代码做一下分析。它使用了 Java Sound API 中的 AudioFormat 这个类，这个类非常重要，在 Java 中对任何音频数据的使用都要实现通过它指定所需要使用的音频格式，AudioFormat 类有一个嵌套的类 AudioFormat.Encoding，实际上大部分对 AudioFormat 类的使用都是使用的这个嵌套的类。

AMAudioFormat 类的重要方法：

名称：getLineAudioFormat

调用格式：getLineAudioFormat(整型音频格式代号)

返回值：根据传递音频格式代号生成的 AudioFormat 对象。

② AudioCapture 类

AudioCapture 类(封装在 AudioCapture.java 文件中)封装了从音频硬件捕获音频数据并自动编码为 GSM 音频压缩数据的过程，并且通过它的 getAudioInputStream()方法提供给我们一个音频数据输入流，我们就可以直接将这个流发送到网络中。

AudioCapture 类的重要方法：

名称：getAudioInputStream

调用格式：getAudioInputStream()

返回值：AudioInputStream 对象

③ AudioPlayStream 类

AudioPlayStream 类(在 AudioPlayStream.java 文件中封装)与 AudioCapture 类刚好相反，它封装了 GSM 压缩音频数据的解码和音频信号的回放过程，提供给我们一个音频信号输出流。AudioCapture 类用到的 Java Sound API

中的类它也基本都用到了，只是它使用了 SourceDataLine 接口而不是 TargetDataLine 接口。

④ Debug 类

Debug 类(封装在 Debug.java 文件中)主要用来在调试时输出讯息，代码很少，后来我把其中输出信息的语句都屏蔽了，对程序运行没有影响。

为了方便使用以上几个类，需要对它们进行编译和打包，编译时需要设置相关的编译环境。将以上提到的 Java 源码文件放在了 am 目录下，编译之后可以得到一个 am.jar 文件，所需要做的就是程序中引用这个包。

3 程序设计实现

基于以上思路，笔者成功地开发了语音聊天测试软件。该开发平台为 Windows XP；全双工声卡；开发工具为 Java。

3.1 程序的结构

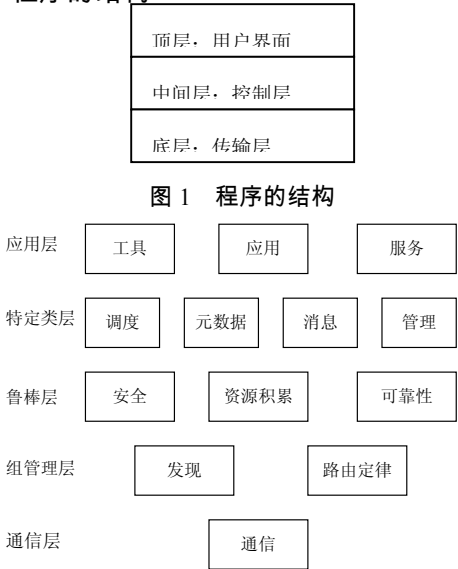


图 1 程序的结构

图 2 P2P 的层次模型

3.2 程序有2个主要的类

程序的主类 jphone 使用了 Runnable 和 ActionListener 接口，主类除了基本的几个方法之外，还具有方法 initAudioHardware()、ShowMSG、startPhone 分别用于初始化 AudioCapture 类与 AudioPlayStream 类、显示程序状态和开始程序。主类 jphone 具有两个子类 VoiceSender 和 CallLink。

表 1 程序的 2 个类

类名	描述
CallLink	网络传输层，用于接收或发送音频数据。
VoiceSender	作为第二个启动的线程提供从音频硬件捕获并编码好的数据给网络传输层。

子类 VoiceSender 同样使用了 Runnable 接口，它在程序中作为第二个启动的线程负责发送捕获到的音频数据。CallLink 子类就是负责建立 socket 连接，并且负责接收或发送网络数据、监听网络连接等功能的实现。它具有

主要的方法是 `getInputStream()`、`getOutputStream()`、`listen()`、`open()`、`close()`等。

3.3 程序的基本工作流程

当程序启动时首先执行建立当前主类的实例,当按下呼叫按钮的时候执行 `startPhone()`方法,`startPhone()`方法通过调用 `initAudioHardware()`方法建立 `AudioCapture` 对象和 `AudioPlayStream` 对象的实例 `PhoneMIC` 和 `PhoneSPK`,紧接着在建立 `CallLink` 子类的实例 `curCallLink` 来与具有目标 IP 地址的计算机进行 socket 连接后, `startPhone()`方法又将子类 `VoiceSender` 作为 `secondThread` 线程启动,然后又调用 `run()`方法。`run()`方法通过已经建立的 `CallLink` 子类的实例 `curCallLink` 监听网络上的数据(也就是等待别人的呼叫),一旦有音频数据到来 `curCallLink` 实例就为 `AudioPlayStream` 对象 `PhoneSPK` 提供网络传来的音频数据,而 `PhoneSPK` 在一个循环中不断的将音频数据转换为音频信号,完成类似电话听筒的功能。

子类 `VoiceSender` 就作为第二线程启动的时候, `startPhone()` 方法传递给它的参数是实例化的 `CallLink` 子类 `curCallLink`, 子类 `VoiceSender` 通过实例化的 `AudioCapture` 对象 `PhoneMIC` 将音频信号压缩成 GSM 数据,并通过 `curCallLink` 将音频数据发送到具有目标 IP 地址的计算机上,完成类似电话受话器的功能。在这里实例化的 `CallLink` 子类 `curCallLink` 就相当于两个电话之间的电话线。

其中的音频数据发送线程和音频数据接收线程是同步的,不过考虑到网络的因素,可能在声音传输上有一些延迟,不过由于延迟比较小对及时听到对方话语影响不大。

3.4 编写代码摘要

```
boolean    complete    false;
byte[ ]    gsmdata new byte[bufSize];
int numBytesRead 0;
.....
PhoneSPK.start();
.....
```

```
complete false;
while(!Thread.currentThread().interrupted())
{
    try
    {
        numBytesRead=playbackInputStream.read(gsmdata);
        if(numBytesRead == -1)
        {
            complete=true;
            break;
        }
        PhoneSPK.write(gsmdata, 0, numBytesRead);
    }
    catch (IOException e)
    {
        System.exit(1);
    }
}
```

程序系统可以在两台或者两台以上的 PC 上运行,通过输入对方的 IP 和选择不同的端口,就可以通过麦克风实现语音对话了,并且语音对话的质量效果还不错。

4 结束语

该系统利用 Java 技术,在 P2P 模型下设计了一个语音聊天软件。当然对于该软件仍需要进一步改进和加强。

参考文献:

- [1]侯俊杰. JAVA 2 程序设计教程与上机实训[M]. 北京: 中国铁道出版社, 2004.
- [2][美]埃克尔侯捷. Java编程思想(第2版)[M]. 北京: 机械工业出版社, 2003.
- [3]苗长云. 现代通信原理及应用[M]. 北京: 电子工业出版社, 2005.
- [4]<http://www.jsresources.org/apps/am.html>[EB/OL].

Voice Chat Software Based on P2P

XU Jin, YANG Xiu-ping

(Computer Science Department, Changsha College, Changsha, Hunan 411103, China)

Abstract: P2P is a new kind of distributed computing mode that enables information exchange between different nodes. By means of analyzing the principles of voice communication employing P2P techniques, this paper expounds on the solutions to some vital problems in the P2P voice transfer. Both a developing plan of the voice chat software based on Java and an instance are also put forward in it.

Key words: P2P; voice communication; Java

(责任编辑: 向旭宇)