摘要

VoIP(Voice over IP)是近几年发展起来的一种新的通信技术。它是网络分组化和网络业务综合化两大主流技术融合的结果。由于其通信成本低、支持的业务丰富，所以它的发展如火如荼，正在侵占传统的PSTN电话市场。但是，VoIP属于实时语音通信，需要一定的网络传输质量来保证。而目前的Internet服务并不能满足这一要求，因此导致当前的VoIP业务在稳定性和QOS上不尽人意。这已成为其进一步发展的主要障碍和挑战，也是IP电话亟待解决的重大难题之一。

对语音质量进行科学可靠的测量和评价是提高 VoIP 技术十分关键的前提。影响VoIP语音质量的因素是多方面的，这主要是由IP网络的固有特性所引起的，其中包括网络时延、抖动、丢包、乱序等。这些传输损伤会直接影响语音质量或通话者之间的交流，从而影响整个 VoIP 的通信。损伤参数在很大程度上依赖于接收端的播放缓冲机制。因此要提高 VoIP 的语音质量，就必须在 VoIP 通信系统中进行抖动处理。抖动缓冲既不能太长也不能太短，这样才能保证在丢包和延时之间取得较好的平衡。目前通信系统中大多采用自适应抖动缓冲算法来解决这一难题。

本文的主要工作和贡献表现在以下几方面：

1、 研究了 VoIP 通信中影响语音质量的几个关键因素，并分析了相应的解决方案。

2、 阐述了 VoIP 语音质量的客观和主观测试方法。并提出了一种新的测试语音质量的方法。在这个方法中，我们基于 PESQ 和 E-Model，将两者结合提出了一个用于不同 codec 的非线性回归模型。

3、 阐述了抖动的形成和对语音质量影响。重点研究分析 7 种已有的自适应缓冲算法，并在此基础上设计了一种改进的自适应抖动缓冲算法来更好的适应不同的网络环境，达到优化语音质量的目的。

4、 设计了一个简单有效的语音质量测试系统，对上述的 8 种算法进行了测试，比较分析了测试结果。结果表明，新的抖动缓冲技术具有更好的网络自适应性，能够更大程度地降低丢包、抖动、乱序等，表现出优越的性能。

关键字：VoIP，抖动，自适应抖动缓冲算法，语音质量

Abstract

VoIP (Voice over IP) is a new communication technology arisen in recent years. It is the result of the amalgamation by two mainstream technologies, one is packet network marked by IP, and the other is the integration of network service which is aimed at multimedia. Because of its low cost of communications, rich support of the business, it develops like a raging fire and is licking up the traditional PSTN phone market. However, real-time voice communications like VoIP needs some network transmission quality assurance, but the current internet can not meet this requirement, resulting its unsatisfactory of the current VoIP services on the stability and QOS. This has become the main obstacle and challenges to its further development as well as the major problem that IP phone technology has to deal with.

How to carry out scientific and reliable voice quality measurement and evaluation is the key issue of improving VoIP quality. There are many factors which impact VoIP voice quality, mainly caused by the inherent characteristics of IP networks, including the network delay and jitter, packet loss, as well as out-of-order. These impairments will directly influence the communication. The parameters for impairment rely on jutter buffer to great extent. Therefore, we have to deal with jitter in VoIP communication system to improve its quality. The buffer should be neither too long nor too short to achieve the trade-off between the loss and delay. Most of the current communication system using adaptive jitter buffer algorithm to solve this problem.

The main tasks and contributions of this paper are as follows:

1. Studied the key fators that affect voice quality in VoIP communication and analysed the corresponding solutions.
2. Illustrated the subjective and objective methods to evaluate voice quality of VoIP, and proposed a more efficency and accurate test method with the combination of PESQ and MOS.
3. Explained how the jitter comes into being and what its influence to voice quality. We focus on study the exsiting

7 buffer algorithms, and based on this we designed an improved adaptive jitter buffer algorithm to better adapt to different network environment and optimize voice quality.

4. We designed a simple but effective test system, and tested the above 8 algorithms. From the result, we can see that the new jitter buffer technology has better adaptiveness to different network environments, and it can reduce the packet loss, jitter and out-of-order to greater extent. This algorithm demonstrated superior performance.

Key words: VoIP, Jitter, Adaptive Jitter Buffer, Voice quality

学位论文独创性声明

本人所呈交的学位论文是我在导师的指导下进行的研究工作及取得的研究成果。据我所知，除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：李敏　日期：2008.6.1

学位论文使用授权声明

本人完全了解华东师范大学有关保留、使用学位论文的规定，学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版。有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆被查阅。有权将学位论文的内容编入有关数据库进行检索。有权将学位论文的标题和摘要汇编出版。保密的学位论文在解密后适用本规定。

学位论文作者签名：李敏　导师签名：

日期：2008.6.1　　日期：2008.6.5

Originality Notice

In presenting this thesis in partial fulfillment of the requirements for the Master's degree at East China Normal University, I warrant that this thesis is original and any of the techniques presented in the thesis have been figured out by me. Any of the references to the copyright, trademark, patent, statutory right, or propriety right of others have been explicitly acknowledged and included in the *References* section at the end of this thesis.

Signature: Li Min **Date:** 2008.6.1

Copyright Notice

I herein agree that the Library of ECNU shall make its copies freely available for inspection. I further agree that extensive copying of the thesis is allowable only for scholarly purposes, in particular, storing the content of this thesis into relevant databases, as well as compiling and publishing the title and abstract of this thesis, consistent with "fair use" as prescribed in the Copyright Law of The People's Republic of China.

Signature: Li Min **Date:** 2008.6.1

第一章 前言

1.1 课题背景及研究意义

VoIP(Voice Over IP)是近几年流行起来的一种新的通信技术。它是建立在IP网络上的分组化、数字化传输技术。VoIP的基本原理是：采用数字化处理技术对语音信息进行压缩处理，然后将语音分组经过IP网络传送到接收端，再次进行数字处理后，通过终端播放出来，从而实现端到端的双向通信【22】。VoIP技术节约了初始投资成本，降低了网络管理的复杂度，为用户提供了低廉的话音服务以及会议电视等多媒体通信服务。特别是近两年随着无线网络以及WiFi技术的完善和性能的提高，IP手机无疑成为VoIP市场的催化剂。这也是无线互联网与3G融合的一个体现。因此移动VoIP成为手机芯片生产厂商不得不考虑的问题。在过去的几年中，已经有很多的通讯设备生产厂商宣称正在研发VoIP手机，并将很快推向市场。据市场调查公司预测，到2008年底20%的手机都会具有WiFi功能。各大移动运营商也纷纷摩拳擦掌，以使自己能提供IP承载语音服务。可以预见在未来几年内， VoIP手机将取得快速的发展，赢得更多的市场份额，并逐步取代传统电话。

移动VoIP是未来发展的趋势，它的前景和巨大的商业价值使业界对其产生了浓厚的开发热情。但是由于IP网络诸多因素的影响，大大削弱了语音质量。与传统电话相比，VoIP的一个致命弱点就是QOS无法保证。IP网络并不是专为面向实时的通信而设计的，它的固有缺点在于，它是一种提供‘尽力而为’(Best Effort)服务技术的网络。数据在IP网中传播会有时延(Delay)，抖动(Jitter)，包丢失（Loss）及乱序（Out of order）等现象。它要保证的主要是数据的无错传输，并不强调其实时性。其重传时延对于一些非实时应用来说并不重要。而VoIP系统都是采用没有质量保证的UDP/IP协议进行传输，且要求语音数据实时传输，以往那种通过数据重传来弥补数据包丢失和错误的做法，会引入极大的延时，并使实时业务的性能降低到无法被接受的程度。由此IP网络所存在的一些问题就严重的影响了语音质量。这也是其发展的一个瓶颈。所以如何在IP网络中更好的传送实时语音信息，如何提高VoIP的QOS和语音质量，以在市场上夺取更高的占有率，已经成为人们研究的热点问题。

虽然VoIP技术在我国已经得到了应用，但应用范围相对于巨大的市场来说还是非常狭窄，该技术灵活丰富的业务形式也没有得到完全体现。另外，我国VoIP业务的经营许可也还尚未放开，仅有少数公司在个别地区试点，整个VoIP技术产业仍处于蓄势待发状态，我国发展VoIP技术已经是大势所趋，只是等待

政策放开的时间，而海外一些国家和地区该项技术的应用已经红红火火。可以预见，一旦政策放开，国内 VoIP 市场随之而来的必将是一场激烈的竞争。

语音质量是推动 VoIP 技术不断发展的关键，在整个 VoIP 系统中具有举足轻重的地位。因此，研究语音质量并对其进行改善，对于提高终端产品的竞争力，在激烈的市场竞争中取得优势有重要的意义。鉴于此，本文深入研究了影响 VoIP 语音质量的各个因素，并且针对网络抖动提出了一种改进的技术来提高话音质量。

1.2 研究范围及方法

1.2.1 论文的内容与创新点

本文首先回顾了 VoIP 系统概念。分析了网络损伤对 VoIP 系统中语音质量的影响，特别是丢包和时延。接下来阐述了如何去测试 VoIP 系统的语音质量。在此基础上对其关键技术抖动缓冲进行进一步研究。难点则是如何综合考虑丢包与抖动之间的相互制约，并选择一个好的折衷方案，来更好的自适应不同的网络环境，改善语音质量。在介绍抖动缓冲技术原理时，结合图例与公式，详细阐述消除时延抖动的原理和方法。本文介绍了当前几种主流的抖动缓冲算法，分析了各自的优缺点，提出了综合性能上比以前算法更好的自适应抖动缓冲算法，将所有算法一一实现，并与新的算法进行比较。

本课题主要创新点如下：

(1)提出了一种测试 VoIP 语音质量的新方法。在这个方法中，我们基于 PESQ 和 E-Model，将两者结合提出了一个用于不同 codec 的非线性回归模型。

(2) 研究了 7 种已有的抖动缓冲算法，提出了一种新的自适应缓冲算法，它针对不同的网络环境直接最大化 MOS 值，可以在尖峰期间有效的调整播放延时。我们实现了这些算法并比较了它们的性能。

(3)分析了 VoIP 系统的各个模块，设计了一个行之有效的测试系统，对 VoIP 系统的语音质量进行全面测试。

1.2.2 论文的组织结构

论文共包括六章内容：

第一章 简单介绍了 VoIP 系统的概念和课题研究背景，提出了本论文的研究思路和主要工作。

第二章 对 VoIP 系统做了总体介绍，包括它的协议 SIP 和 H.323、常用的编码方式、特点以及发展趋势。

第三章 分析了 VoIP 的 QOS 以及影响语音质量的几个网络损伤参数和相对应的几

个关键技术。我们发现这些损伤特征是依赖于 IP 网络的。最后详细分析了评价感知语音质量的不同方法，包括主观测试和客观的测试。对这些已有的评估感知语音质量的方法做了分析。然后我们将 PESQ 和 MOS 相结合，提出了一种简单有效的方法来测试语音质量。

第四章 分析了抖动缓冲技术，研究了抖动产生的原因以及对话音质量的影响。接下来研究了 7 种不同的缓冲算法【3】【4】【6】【7】【8】【9】【20】，并提出了一种新的自适应缓冲算法。它可以在给定的网络参数下最大化 MOS 指标，并且可以在不同的情况下有效的调整播放延时。

第五章 介绍了 VoIP 系统以及每一个模块的实现。介绍了各个模块的接口。并在此基础上对系统进行测试。从初步的结果可以看到，新的算法在大多数网络条件下都可以增强感知语音质量，并且更加适合真正的缓冲机制。

第六章 总结和展望。对全文进行总结，对整个系统进行了概述。在此基础上提出了论文有待改进的地方。

第二章　VoIP 系统介绍

这一章，我们先粗略的了解一下 VoIP 系统的概念，介绍相关的协议以及编码方式并与 PSTN 系统相比较。接下来描述一个典型的 VoIP 系统。最后，总结 VOIP 技术。

2.1 VoIP 的基本原理

IP 网络电话泛指在以 IP(Inernet Protocol)为网络层协议的计算机网络中进行话音通信的系统，它采用的技术通称 VoIP(Voice Over IP)。IP 技术允许多个用户共用同一带宽资源，改变了传统电话由单个用户独占一个信道的方式，节省了用户使用单独信道的费用。其次，由于技术和市场的推动，将语音转化成 IP 包的技术已变得更为实用、便宜，同时，IP 电话的核心元件之一数字信号处理器的价格在下降，从而使电话费用大大降低，这一点在国际电话通信费用上尤为明显，这也是 IP 电话迅速发展的重要原因。

第一个VoIP软件由1995年产生，在这相对来说很短的时间内，网络电话产业发展非常迅速。近年来涌现了很多VoIP嵌入式产品，并且随着无线移动技术的发展，越来越多的手机生产厂商开始研发WiFi与VoIP相结合的手机。图2.1为一VoIP移动终端解决方案【8】。

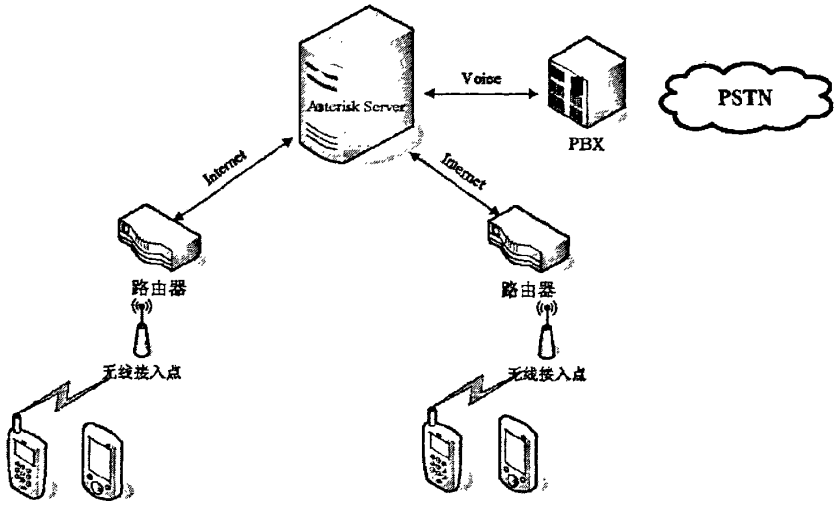


图 2.1　VoIP 移动终端解决方案

最简单的 VoIP 系统由两个或多个具有 VoIP 功能的设备组成，并通过 IP 网络连接。VoIP 模型的基本结构如图 2.2 所示。图中示出了 VoIP 设备如何把语音信号转换为 IP 数据流，并把这些数据流转发到 IP 目的地，IP 目的地又把它们转换回到语音信号。

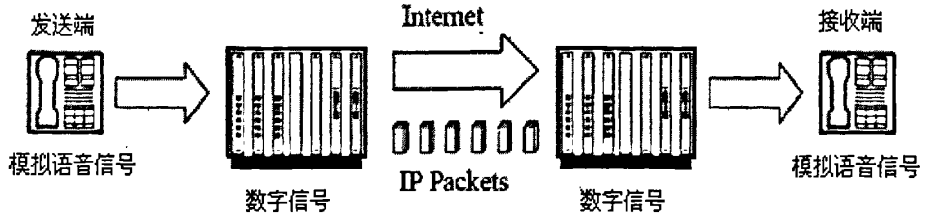


图 2.2 VoIP 基本结构

我们可以简单地将 VoIP 的传输过程分为下列几个阶段【22】:

1. 语音-数据转换

语音信号是模拟波形，通过 IP 方式来传输语音，不管是实时应用业务还是非实时应用业务，首先要对语音信号进行模拟数据转换，也就是对模拟语音信号进行量化。数字化可以使用各种语音编码方案来实现，目前采用的语音编码标准主要有 ITU-T G.711,G.729,G.723.1，AMR 等等。源和目的地的语音编码器必须实现相同的算法，这样目的地的语音设备才可以还原模拟语音信号。

2. 原数据到 IP 转换

一旦语音信号进行数字编码，下一步就是对语音包以特定的帧长进行压缩编码。大部分的编码器都有特定的帧长，典型值为 10～30ms。语音包通常由 60、120 或 240ms 的数据组成。若一个编码器使用 15ms 的帧，则把每一个 60ms 的包分成 4 帧，并按顺序进行编码。每个帧为 120 个采样点（采样率为 8kHz）。编码后，将 4 个压缩的帧合成一个语音包送入网络处理器。网络处理器为语音添加包头、时标和其它信息后通过网络传送到另一端点。

3. 传送

语音网络简单地建立通信端点之间的物理连接，并在端点之间传输编码信号。IP 网络不像电路交换网络，它不形成连接，它要求把数据放在可变长的数据报或分组中，然后给每个数据报附带寻址和控制信息，并通过网络发送，一站一站地转发到目的地。在这个通道中，网络从输入端接收语音包，然后在一定时间（t）内将其传送到网络输出端。t 可以在某个范围内变化，反映了网络传输中的抖动。网络中的节点检查每个 IP 数据附带的寻址信息，并使用这个信息把该数据报转发到目的地路径上的下一站。

4. IP 包-数据的转换

目的地 VoIP 设备接收这个 IP 数据并开始处理。网络提供一个可变长度的缓冲器，用来调节网络产生的抖动。该缓冲器可容纳许多语音包，用户可以选择缓冲器的大小。小的缓冲器产生延迟较小，但不能调节大的抖动。其次，解码器将经编码的语音包解压缩后产生新的语音包，这个模块也可以按帧进行操作，完全

和解码器的长度相同。若帧的长度为 15ms，是 60ms 的语音包被分成 4 帧，然后它们被解码还原成 60ms 的语音数据流送入解码缓冲器。在数据报的处理过程中，去掉寻址和控制信息，保留原始的原数据，然后把这个原数据提供给解码器。

5. 数字语音转换为模拟语音

播放驱动器将缓冲器中的语音样点（480 个）取出送入声卡，通过扬声器按预定的频率（例如 8kHz）播出。在像 IP 电话这样的全双工通信中，每一端既是发送端又是接收端。

简而言之，语音信号在 IP 网络上的传送要经过从模拟信号到数字信号的转换、数字语音封装成 IP 分组、IP 分组通过网络的传送、IP 分组的解包和数字语音还原到模拟信号等过程。

2.2 VoIP 与 PSTN 的比较

传统的电话网以电路交换方式传输语音，所要求的传输宽带为 64kbit/s。而 VoIP 是以 IP 分组交换网络为传输平台，对模拟的语音信号进行压缩、打包等一系列的特殊处理，使之可以采用无连接的 UDP 协议进行传输。

1. 传统 PSTN 话音业务

传统的话音通信网（PSTN）是采用面向连接及基于 64kb/s 信道的电路交换，通过信令来控制连接，传输上使用同步传输方式（STM）。PSTN 终端即可以是模拟的也可以是数字的。标准的电话机通过模拟接入网络(用户环路)连接到 PSTN。在模拟接入网络中，话音以 3kHz 带宽的模拟信号传输；在接入交换机中，该话音被数字化；其信令能力（如地址识别）开始时是通过带内的 DTMF(Dual Tone MultiFrequency)音来实现，后来用独立信令网来实现。数字核心网络使用的信令系统是七号信令系统。由于 PSTN 是专为语音通信而设计地，所以，它具有适合语音通信地优点。主要优点是业务质量(QOS)如延迟、语音地清晰度和自然度等有保障；控制相对简单。但随着通信业务需求地变化和技术进步，其不足之处日渐突出起来，其缺点如下【12】:

a) 呼叫建立需要时间并占用资源；
b) 每个连接的带宽固定，不能适应非 64kbps 速度和可变速度的业务需求；
c) 在对话静音期也占用资源，浪费带宽；
d) 在长途线路上，很少或不使用压缩，造成长途线路资源浪费；
e) 由于其网络及终端的固有属性，使得其不能容易实现一些增值业务；

2. VoIP 的特点

VoIP 使企业、服务供应商和电信运营商们看到了许多美好的前景。它的发展远远超过了 PSTN 电路交换网络的发展。这主要是由于 VoIP 所具有的以下特点决

定的：

(1)成本低， 远远低于传统电话的成本。特别是长途电话，国际长途电话。因为IP 网络没有地域的概念。国际间的 VoIP 数据的传输也只是增加了些传输延时。

(2)因特网可以同时支持话音业务和纯数据业务的应用。这使得它相比于传统电话交换网络具有极大的优势。

(3)支持多媒体业务的需求。

(4)随着 WiFi 技术的发展，能够将无线技术与移动技术相结合。进一步降低话费，实现真正的无缝通信。

正是因为 VoIP 具有低廉的费用，使用灵活，支持功能多等特点，越来越受到人们的欢迎，得到了快速发展。 由此我们可以知道，未来几年内，无论是在国外还是在国内，作为给用户提供的一种选择，VoIP 必将得到迅猛发展.

2.3 VoIP 协议

信令技术保证电话呼叫的实现和通话质量，目前被广泛接受的 VoIP 控制信令体系包括 ITU－T 的 H.323 和 IETF 的 SIP、MGCP 和 H.248，但仍以 H.323 为主，然而 SIP 的支持将会成为今后的主流。

2.3.1 H.323

H.323【16】是 ITU-T 的多媒体通信协议系列 H.32x 中的一个。它提供了基于 IP 网络的传送声音、视频和数据的基本标准，并为 IP 网络上的多媒体通信应用提供了技术基础。它最初用于局域网（LAN）上的多媒体会议，后来扩展至覆盖 VoIP。该标准既包括了点对点通信也包括了多点会议。H.323 定义了四种逻辑组成部分：终端、网关、网守和多点控制单元（MCU）。

H. 323 并不依赖于网络结构，而是独立于操作系统和硬件平台之上，支持多点功能、组播和频宽管理。H.323 具备相当的灵活性，可支持包含不同功能节点之间的视频会议和不同网络之间的视频会议。

H.323 并不支持群播（Multicat）协议，只能采用多点控制单元构成多点会议，因而同时只能支持有限的多点用户。H.323 也不支持呼叫转移，且通过建立呼叫的时间也比较长。

早期的视频会议多支持 H.323 协议，如 NetMeeting、Intel Video Phone 等都是支持 H.323 协议的视频会议软件。

但是 H.323 协议本身具有一些问题，例如采用 H.323 协议的 IP 电话网络在接入端仍要经过当地的 PSTN 电路交换网。而之后制定出的 MGCP 等协议，目的在于将 H.323 网关进行功能上的分解，也就是划分成负责媒体流处理的媒体网关

（MG），以及掌控呼叫建立与控制的媒体网关控制器(MGC)两个部分。

2.3.2 SIP

SIP（Session Initiation Protocol）是由 IETF 所制定的应用层控制协议，定义了用户间交互式媒体会话的发起，修改和终止过程。它与 H.323 协议并列。SIP 主要借鉴了 Web 网的 HTTP 和 SMTP 两个协议。其特性几乎与 H.323 相反。主要为解决 Internet 网上的多媒体会议电视服务，用于建立互联网上主机之间的会话。这个协议最初以 HTTP 为基础，设计上遵循 Internet 的基本原则来架构 IP 电话业务网，协议很容易进行扩展，便于增加新业务，具有较强的互操作性。原则上它是一种比较简单的会话初始化协议，也就是说只提供会话或呼叫的建立与控制功能。与 H.323 协议为基础的 IP 电话相比，SIP 协议需要相对智能的终端。正是因为 SIP 系统有了相对智能的终端系统，所以它才有可能实现用户个性化的需要。

SIP 网络采用 internet 的 client/server 的工作方式，网络结构如图 2.5 所示。SIP 中有两个要素：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。网络服务器是处理与多个呼叫相关联信令的网络设备。

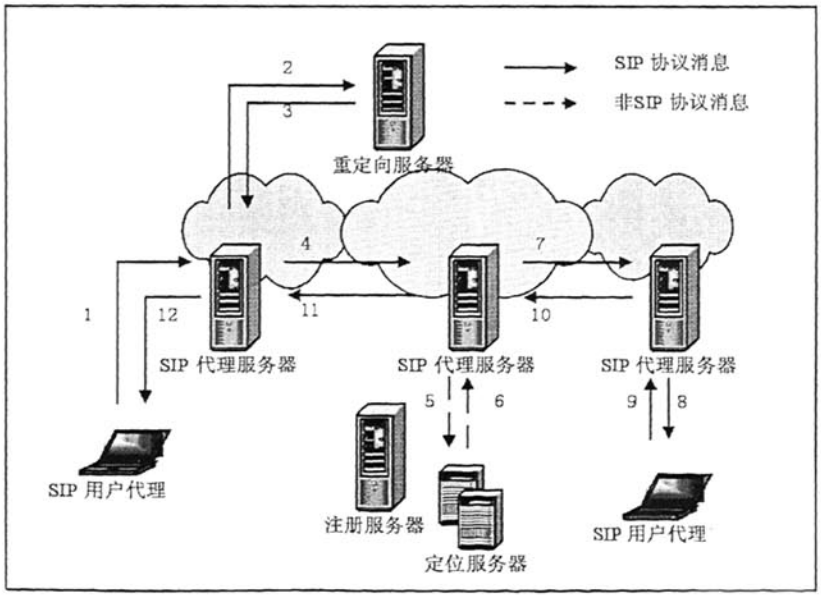


图 2.5 SIP 网络结构图

SIP 采用文本编码格式，其消息分为两种：UAC 到 UAS 的请求(Request)和 UAS 到 UAC 的响应(Response)，消息包括消息头和消息体两部分。

SIP 消息包括三个部分:一个起始行(Start-line、一个或多个字段(Field)组成的消息头、一个标志消息头结束的空行(CRLF)以及作为可选项的消息体(Message body)。起始行分请求行(Request-line)和状态行(Status-line)两种,其中请求行是请求消息的起始行,状态行是响应消息的起始行, 起始行位于消息的最开始。消息头分通用头 (general-header)、请求头 (request-header)、响应头 (response-header) 和实体头 (entity-header) 四种。消息头,描述消息的属性,类似于 HTTP 消息头的语法和语义。消息体主要是对消息所要建立的会话的描述。

SIP 同时支持单点播送 (Unicast) 及群播功能,也就是说使用者可以随时加入一个已存在的视频会议之中。在网络 OSI 属性上, SIP 属于应用层协议,所以可以透过 UDP 或 TCP 协议进行传输。

SIP 定义了下述方法:

1) INVITE:INVITE 方法用于邀请用户和服务参加一个会话。在 INVITE 请求的消息体中可对被叫方被邀请参加的会话作以描述。如主叫方能接收的媒体类型、发出的媒体类型及其一些参数。对 INVITE 请求的成功响应必须在响应的消息体中说明被叫方愿意接收哪种媒体,或者说明被叫方发出的媒体。服务器可以自动地用 200 OK 响应会议邀请。

2) BYE: BYE 用来终止呼叫上的两个用户之间的呼叫。呼叫的一方在释放 (挂断) 呼叫前必须发出 BYE 请求, 收到 BYE 请求的这方必须停止发媒体流给发出 BYE 请求的这方。

3) OPTIONS: OPTIONS 用于向服务器查询其能力。如果服务器认为它能与用户联系, 则可用一个能力集响应 OPTIONS 请求。

4) ACK: ACK 请求用于客户机向服务器证实它已经收到了对 INVITE 请求的最终响应。ACK 只和 INVITE 请求一起使用。对 2xx 最终响应的证实由客户机用户代理发出,对其它最终响应的证实由收到响应的第一个代理或第一个客户机用户代理发出。

5) REGISTER: REGISTER 方法用于客户机向 SIP 服务器提供地址解析的映射,让服务器知道其它用户的位置。

6) INFO:INFO 方法是对 SIP 协议的扩展, 用于传递会话中产生的与会话相关的控制信息, 如 ISUP 和 ISDN 信令消息, 以及 DTMF 数字等。

7) CANCEL 请求用于取消一个 Call-ID、To、From 和 Cseq (仅序列号) 字段值相同的正在进行的请求,但取消不了已经完成的请求(如果服务器返回一个最终状态响应, 则认为请求已完成)。

下图所示的场景为参与会话的两个 User agent, 从发起呼叫、建立链路、媒

体流传输、到拆除 SIP 信令链路的全过程：

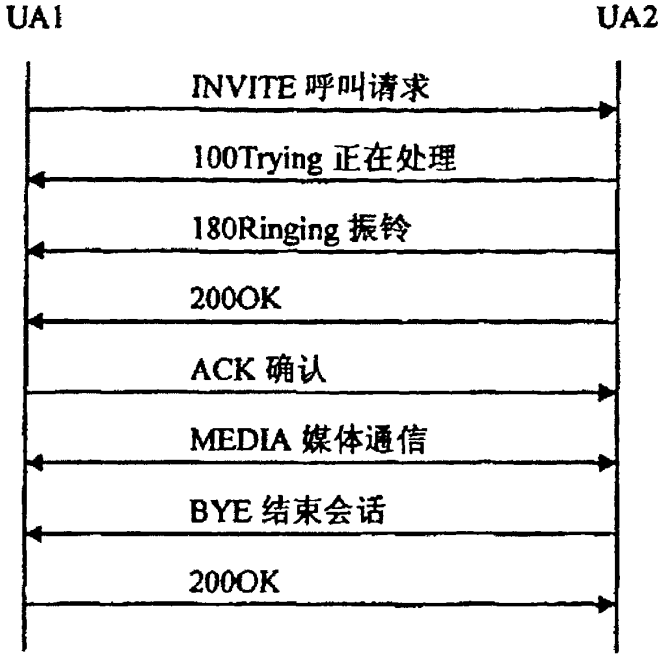


图 2.6 两个 UA 直接通信的呼叫流程

（1）首先，主叫向被叫发出 INVITE 请求。INVITE 请求的作用是发起并建立呼叫，邀请被叫加入主叫建立的呼叫。

（2）被叫收到请求后对主叫做出响应。接受请求方对请求的响应分为临时响应（状态码为 1xx）和最终相应（状态码为 2xx）。主叫只对最终相应做出回应。图 2.6 中，被叫做出的临时相应有 100Trying（尝试连接），180 Ringing（被叫振铃或进入受到请求状态），182 Queued（被叫可能有多个呼叫要处理，所以主叫请求需要排队等待）；被叫做出的最终响应是 200 OK，表示被叫接受并开始处理呼叫请求。

（3）为了向被叫证实主叫收到了最终响应，主叫收到响应后发送 ACK 请求。被叫收到主叫的 ACK 请求，标志呼叫建立阶段结束。

（4）主叫或被叫在呼叫建立后发起后续请求。后续请求可由参加呼叫的任一方发起。可发起 INVITE 请求，进行交互操作，并对当前呼叫进行修改；也可发起 BYE 请求终止当前呼叫。

2.3.3 SIP 与 H.323 比较与总结

SIP 和 H.323 作为 IP 电话的信令协议，分别是通信领域与 Internet 领域两大阵营推出的建议，它们的区别有以下几点【7】：

从信令协议的出发点来看,H.323 试图把 IP 电话当作是众所周知的传统电话，只是传输方式发生了改变，由电路交换变成了分组交换。而 SIP 协议侧重于将 IP 电话作为 Internet 上的一个应用，较其它应用(如 FTP、E-mail 等)增加了信令和 QoS 的要求。

从消息的编码方法来看，H.323 采用基于 ASN.1 和压缩编码规则的二进制方法表示其消息。ASN.1 通常需要特殊的代码生成器来进行词法和语法分析。而 SIP 是基于文本的协议，类似于 HTTP。基于文本的编码意味着头域的含义是一目了然的，如 From、To、Subject 等域名。这种几乎不需要复杂的文档说明的标准规范风格，其优越性已在过去的实践中得到了充分的证明。

从会话能力的协商和调整方法来看，H.323 是采用 H.245 协议来进行能力协商的会话控制的，而 SIP 的能力协商采用 SDP(Session Description Protocol)进行描述，SDP 中的每一项的格式为<type>=<value>，也比较简单。

从会话管理的方式来看，H.323 由于由多点控制单元(MCU)集中执行会议控制功能，所有参加会议的端点都向 MCU 发送控制消息，MCU 可能会成为瓶颈，特别是对于具有附加特性的大型会议；并且 H.323 不支持信令的多播功能，其单播功能限制了可扩展性，降低了可靠性。而 SIP 设计上就为分布式的呼叫模型，具有分布式的多播功能。其多播功能不仅便于会议控制，而且简化了用户定位、群组邀请等，并且能节约带宽。

在补充业务方面，H.323 中定义了专门的协议用于补充业务，如 H.450.1、H.450.2 和 H.450.3 等。SIP 并未定义专门的协议用于此目的，但它能很方便地支持补充业务或智能业务。只要充分利用 SIP 已定义的头域，并对 SIP 进行简单的扩展，就可以实现这些业务。对于通过扩展头域较难实现的一些智能业务，可在体系结构中增加业务代理，提供一些补充服务或与智能网设备的接口。

另外，H.323 中的呼叫建立过程涉及到三条信令信道：RAS 信令信道、呼叫信令信道和 H.245 控制信道。通过这三条信道的协调才使得 H.323 的呼叫得以进行，呼叫建立时间很长。在 SIP 中，会话请求过程和媒体协商过程等一起进行。尽管 H.323v2 已对呼叫建立过程做了改进(H.245 控制消息可以通过用 H.225.0 呼叫信道隧道来传送)，但较之 SIP 只需要 1.5 个回路时延来建立呼叫，仍是无法相比的。

H.323 的呼叫信令信道和 H.245 控制信道需要可靠的传输协议，而 SIP 独立于低层协议，一般使用 UDP 等无连接的协议，用自己应用层的可靠性机制来保证消息的可靠传输。

总之，H.323 沿用的是传统的实现电话信令的模式，比较成熟，已经出现了不少 H.323 产品。H.323 符合通信领域传统的设计思想，进行集中、层次式控制，

采用 H.323 协议便于与传统的电话网相连。SIP 协议借鉴了其它 Internet 的标准和协议的设计思想，在风格上遵循 Internet 一贯坚持的简练、开放、兼容和可扩展等原则，比较简单，但推出的时间不长，协议并不是很成熟。它的优点是同 Internet 结合，可以很方便地生成新的业务，如 Web 呼叫点击拨号等。

综上所述，SIP 协议凭借其简单、易于扩展、便于实现等诸多优点越来越得到业界的青睐，它正逐步成为 NGN（下一代网络）和 3G 多媒体子系统域中的重要协议，并且市场上出现越来越多的支持 SIP 的客户端软件和智能多媒体终端，以及用 SIP 协议实现的服务器和软交换设备。虽然 SIP 协议目前还不成熟，但可以预见 SIP 必定是将来网络多媒体通信中的明星。

2.4 语音编码技术

语音的编码及压缩技术是 VoIP 服务的关键技术之一，采取的编解码算法和压缩技术直接影响到 VoIP 的语音质量。多媒体通信中，编解码器统称为 Codec (Coder + Decoder)。语音编码器的主要功能就是把用户语音的 PCM（脉冲编码调制）样值编码成少量的比特（帧），然后经过专门的 DSP 芯片进行数据压缩，最后再形成 IP 包数据的形式，以适合 IP 网络上的传输带宽。在接收端，语音帧先被解码为 PCM 语音样值，然后再转换成语音波形。这种方法使得语音在产生误码、网络抖动和突发传输时具有健壮性（Robustness）。

2.4.1 语音编码器的分类

语音编码器分为三种类形【7】：（a）波形编码器；（b）参数编码器；（c）混合编码器。

波形编码技术力图使重建语音波形保持原语音信号的波形形状，即在编码端以波形逼近为原则对语音信号进行压缩编码，解码端根据这些编码数据恢复出语音信号的波形。它具有语音质量好、适应能力强、抗噪性能高等优点。波形基编码包含三个过程：抽样、量化和编码。波形编码不适应于低速话音编码，一般属于中高速编码。例如：ITU-G.711 规范（PCM）用的比特率为 64Kbps。

参数编码器是将语音信号用某种模型表示，仅仅对表示语音特征的参数进行编码。它力图使重建语音信号具有尽可能高的可懂性，从听感角度注重语音本身的重现。它通常都是基于某种语音产生模型，在编码端分析出该模型参数并选择合适的方式对其进行高效率的编码，解码端则利用这些参数和语音产生模型重新合成语音。参数基编码一般属于中低速编码。语音质量较差，而且对环境噪声比较敏感。

混合编码融入了波形编码器和参数编码器的长处，它的另一特点是它工作在非常低的比特率（4-6Kbps）。混合编码器采用合成分析（AbS）。它是在 VOIP 中常用的语音编码器。

下面是各种不同编码器的语音质量的比较：

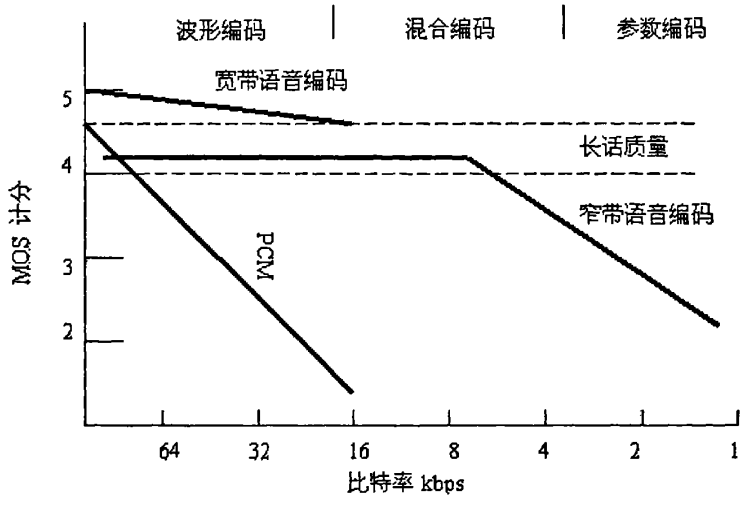


图 2.7 低比特率编码器的 MOS 得分--比特率关系曲线

2.4.2 语音编码方式

在选择压缩算法时，我们要综合考虑各种因素。例如，比特率高的能保证通话质量，但会占用较大的存储空间，耗费更多的系统资源；而比特率低的会增加延迟从而影响通话质量。所以，在较低比特率的前提下，保持较好的话音质量，是选择压缩算法的原则。ITU-T 在 G 系列建议中已经公布了一系列语音编码协议，目前最常见的有 G.711、G.723.1、G.726、G.729、G.729A。3GPP 的 AMR 系列应用也比较广泛。这些协议采用不同的压缩算法，具有不同采样率和编码速率。

下面我们简单介绍一下常用的不同编码算法的特点：

1. G.711【17】是使用最为普遍的语音编码技术。属于波形编码器。其输入数据直接来自 PCM,采样率为 8KHZ，编码速率为 64kbps，帧长度为 125us。G.711 有两种类型:A 律和 Mu 律。Mu 律编码器将 14 位线性 PCM 采样成 8 位压缩的 PCM，主要用在北美和日本；而 A 律将 13 位的线性 PCM 采样成 8 位压缩的对数形式，用于其它大多数国家。两者之间的区别主要在于非均匀量化的方式上。G.711 编码器由于编码速率高，所以获得的话音质量最好。

2. G.723.1【15】是一种为低比特率可视电话而设计的编解码器，具有两种不同的编码速率 5.3kbps（每帧 158 位，20 个字节）和 6.3kbps（每帧 189 位，24 个字节），是目前已标准化的最低速率音频编码算法。G.723.1 编码器的帧长为 30ms，还有 7.5ms 的前视（Look－ahead）。再加上编码器的处理时延，编码器的单向总时延为 67.5ms。它的主速率编码器使用多脉冲最大似然量化（MP-MLQ），低速率编码器使用代数码激励线性预测（ACELP，Algebraic-Code-Excited Linear-Prediction）算法，编码器在 6.3kbps 提供长话质量语音，在 5.3kbps 时提供的较低的语音质量。编码器和解码器都必须支持这两种速率，并能够在帧间对两种速度进行转换。

3. G.726【24】是 ITU 前身 CCITT 于 1990 年在 G.721 和 G.723 标准的基础上提出的，把 64kbps 非线性 PCM 信号转换为 40kbps、32kbps、24kbps、16kbps 的自适应脉冲编码调制 ADPCM（Adaptive Differential Pulse Code Modulation）编码器。它属于波形编码技术。ADPCM 并不像 PCM 编码那样直接量化语音信号，而是量化语音信号和预测信号间的差分信号。

4. G.729【23】编码器是为低时延应用设计的，采用对称结构代数码激励线性预测 CS_ACELP 算法，它的比特率为 8 kbps，帧长 10ms，处理时延也是 10ms，再加上 5ms 的前视，这就使得 G.729 产生的端到端的时延为 25ms。G.729 又有不同的补充协议，如 G.729A，G.729A/B。这两个版本互相兼容但它们的性能有些不同，复杂性低的版本（G.729A）性能较差。两种编码器都提供了对帧丢失和分组丢失的隐藏处理机制，因此在因特网上传输语音时，这两种编码器都是很好的选择，其音频质量与 32kbps 的 ADPCM 相当。G.729A/B 主要用于数字蜂窝网和分组交换网。这种编码方式在一定的背景噪声下语音质量较好。其中 G.729B 具有静音消除技术，可以在通话的静音期间降低带宽占用率。

5. AMR 是由 3GPP 提出的应用于第三代移动通信系统中的一种自适应多码率（Adaptive Multi Rate）算法。研究这种算法的目的就是在低速率 GSM 无线网络中能够提供多速率的声码器。AMR 编码器根据无线信道环境在不同编码模式间进行切换。在好的传输环境下选择最优编码模式，在差的传播环境下选择最大抗噪编码模式。这种算法的压缩比比较大，但是相对与其它的压缩格式语音质量比较差，多用于人声、通话。AMR 又分为 AMR_NB(Narrow Band)、AMR_WB(Wide Band)、AMR_WB+(Wide Band Plus)。它们采用的主要技术基本相同的只是在带宽、编码速率和采样率之间有差别。此外 AMR_WB+还提供了对人声以外的音乐等的编码。

2.4.3 评估编码方式的指标

评估编码器的性能时主要考虑以下几个重要因素【8】【7】：

- 帧大小：帧的大小表示语音流量的时间长度，也称为帧时延。

• 处理时延：它表示在编码器中对一帧语音做编码算法处理所需时间。它通常简单计入帧时延。处理时延也称为算法时延。

• 前视时延：编码器为了对当前帧的编码提供帮助而检查下一帧的一定长度，此长度就称为前视时延。前视的想法是为了利用相邻语音帧之间的密切相关性。

• 帧长度：这个值表示经编码处理后的字节数（不包括帧头）。

• 语音比特率：当编解码器的输入是标准脉冲编码调制的语音码流（比特率为 64 kbit/s）时，编解码器的输出速率。

• DSP MIPS：此值是指支持特定编码器的 DSP 处理器的最低速度。

• RAM 需求：它描述了支持特定的编码过程所需要 RAM 的大小。

评价编码器性能的关键因素是编码器工作所需时间。它是实时通信的重要因素。这个时间是指编码器的缓存及处理时间，称为单向系统时延。其值等于：帧大小+处理时延+前视时延。显然，解码时延也非常重要。实际上，解码时延大约是编码时延的一半。表 2-1 比较了几种编码器的比特率、平均评价分 MOS 复杂性(以 G.711 为基准)，延时(帧大小及前视时间)等。

标准	编码类型	比特/(kbps)	MOS	帧大小	延时/ms
G.711	PCM	64	4.3	1	0.125
G.726	ADPCM	40,32,24,16	4.0	10	0.125
GSM	RAE_LPT	13	3.7	5	200
G.729	CS_CELP	8	4.0	30	10
G.723.1	ACELP	6.3/5.3	3.8	25	30

表 2-1 几种编码器的比较

第三章　VoIP 系统的 QOS 以及测试方法

3.1 VoIP 系统的 QOS

3.1.1 QoS 研究现状

VoIP 相关技术的研究包括很多方面，其中一个主要方面就是 VoIP 的服务质量保证（QoS）问题。所谓服务质量是指为保证所提供业务的质量达到相应标准而采取的一系列措施的技术总称。表 3.1 列出了对 IP 电话的指标要求。

表 3.1 IP 电话的指标

等级	最佳	高	中等	尽力而为
MOS 评估分	4.0-5.0	3.8-4.2	2.9-3.8	2.0-2.9
语音传送全程时延（ms）	0-150	150-250	250-450	450 以上
呼叫建立时间（s）	0-1	1-3	3-5	5 以上

然而，目前的 VoIP 应用还面临很多技术难题。因为 IP 网络依然是提供“尽力而为”服务的网络，没有 QOS 保障。对进入网络的业务流，都以先来先服务的方式对业务流分组进行服务。随着 Internet 和各种业务的迅猛发展，尤其是视频、话音等多媒体业务的迅猛增长，目前的 Internet 服务已不能满足多媒体应用的需求，传统的 IP 网络没有服务质量保障的弱点已经显示出来。如果想要很好地控制数据传输，就必须采用有效的 QoS 方案。因此虽然嵌入式 VoIP 产品是未来发展的趋势，但是其语音质量问题一直是其发展的绊脚石，为此，人们不得不研究 VoIP 的 QOS 来改善和提高语音质量，促进其进一步的发展。

3.1.2 影响 VoIP 语音质量的几个关键参数

VoIP 虽然发展迅速，但是实际应用时会感觉到通话过程中有明显的语音畸变和较频繁的断话现象。其服务质量之所以不能令人满意，是因为当初在设计 IP 网络时，为了提高网络的抗毁性和灵活性，就采用无连接的数据报来传送分组，这样虽然提高了网络利用效率，有利于各种语音数字处理技术的应用，但是并不能保证网络的服务质量。IP 网络平等对待网络上所有的分组，对所传送的分组只是尽最大努力交付。这意味着 IP 网络既不保证将数据可靠地交付给目的主机，

也不保证按序交付，更不保证交付的时限。它无法控制 IP 分组的丢包、延迟和抖动等参数，因此不适合传输实时语音、视频等通信业务。也正是 IP 网络这些固有的特性，影响了 VoIP 的语音质量。

那么我们应该如何提高 IP 电话的 QoS，保证语音质量呢？下面将具体讨论一些影响 VoIP 网络的 QoS 的因素和常用的提高 IP 电话服务质量的措施。【8】

1．时延

时延是指数据从源端到终端所需要的时间，即 IP 数据包在网络上传输所需要的时间。在实时语音通信系统中，语音编解码的时延同线路时延作用一样，对系统的通话质量有很大的影响。另外时延会引起回声，当时延比较小时，回声感觉不明显；当往返总时延超过 100ms 左右，发话者就能够听到自己的回声，如果回声传输路径损耗比较大，就能听到多次回声，从而严重影响通话质量。一般而言，端到端的延时由以下四部分组成：

（1）网络传输延时：指话音从一端到另一端通过网络的时间，由信号通过传播媒介的速度和传播的距离决定。它也是系统整个延迟的主要组成部分。

（2）算法延时：由语音编码器处理语音信号所引起的。它取决与使用的编码器，从单个采样时间（0.125ms）到几个毫秒不等。

（3）处理延时：由实际的语音编码过程以及采集已编码的语音采样进行打包发往整个分组网络传送过程所引起的。通常多个语音帧将集成为一个包在网络上传输，以减少包头对网络增加的负荷。

（4）抖动缓冲延时：指的是在接收端用来保持一个或多个接收的数据包的时间，用来克服数据包到达时间的变化，也就是克服抖动产生的延时。

研究表明，小于80ms 延时的音频质量相当PSTN；80～180ms延时满足一般商业通话要求且强于移动通信；180～230ms延时亦难以察觉，基本满足一般通信之需；单向延时超过250ms会产生回音，但是可以忍受；特别是超过400ms的延时会导致丢包或错序，从而严重影响语音的重现，难以保证正常通信。因此解决延时对提高VoIP音频质量至为关键。【1】

下图给出了IP电话网络中延时分布的情况。

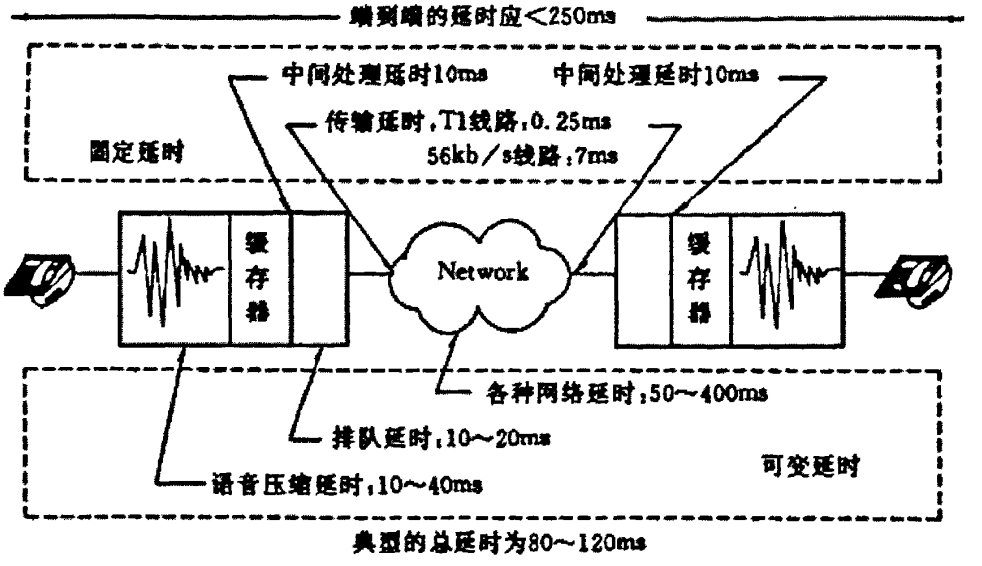


图 3.1 IP 电话网络中时延分布

2. 抖动

抖动也叫时延变化，是指由于各种延时的变化导致网络中的数据分组到达速率的变化。抖动是 VoIP 的一个必测项，它对语音产生的影响是非常明显的，如果网络抖动比较严重，那么有的话音包会因迟到而被丢弃，会产生话音的断续及部分失真，严重影响音质。为了消除时延抖动，通常采用抖动缓冲技术。通过增加抖动缓冲的数量，可以有效地降低抖动的影响。但是因为增加了抖动缓冲，所以也相应增加了网络延时。所以缓冲的大小是一个关键，我们必须在延时和丢包之间取得平衡。在第四章我们会详细讨论抖动缓冲技术，这里我们只给出一个评价抖动的指标：

表 3—2 抖动指标

好	可以接受	差
<75ms	75-125ms	>125ms

3. 包丢失【20】

包丢失是指语音包没有及时到达接收端而被丢弃。在语音传输过程中，为了提高传输效率，保证数据流的实时性，往往采用无连接的数据包传输方式，这就使得分组丢失无法避免，并且无法保证分组的有序传输。造成包丢失的原因主要有以下两个方面：

1. 语音包在网络传输过程中被破坏，由于网络拥塞、传输损伤、超过生

存周期而被丢弃等等。由于语音包主要是依靠面向无连接的UDP包来传输，虽然可以利用RTP报文头的序列号检查数据包的丢失和乱序，但没有相应的重发机制，所以语音信号在传输过程中不可避免会受到损害。

2. 语音分组达到受话端，然而由于延迟过大，超过抖动缓冲处理能力而被丢弃。

丢包是一个影响语音质量的关键因素。很显然包的丢失会使声音断断续续，影响正常通话。包的丢失率达到一定百分比时就会影响语音质量。对于IP包的丢失，典型的语音编码可以允许包丢失率为3%【2】。采取一些特殊措施后，包丢失率达到8%一10%尚可容忍【5】【3】【4】。网络主要有两种类型的丢包情况，一种是或多或少的随机丢包，当网络保持冲突碰撞时，就会偶尔有一个或两个数据包发生丢失；另一种是爆裂丢包，是指连续一个以上的数据包丢失，会显著地影响语音质量。当少量的丢包是随机地分布时，人耳并不容易感觉到较差的语音质量。

图3.2为不同的编解码方式在不同的丢包率情况下对MOS值的影响：

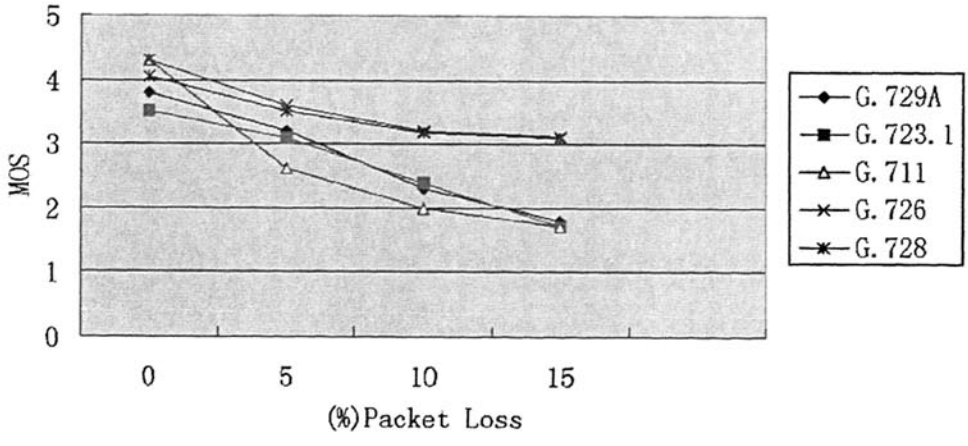


图3.2 不同编码方式在不同丢包率下对MOS值得影响

可见，G.711 的 MOS 值与丢包率的曲线最陡峭。受网络丢包率的影响最大，在网络丢包率达到 15%的条件下所能提供的语音质量已经下降到最低。G.726 编码器的曲线最为平滑，在网络丢包率达到 15%的情况下，仍然能够提供 MOS 值在 3 以上的话音质量。G.728 与 G.726 差不多。G.723.1 和 G.729A 两种编码技术的语音质量较之前三种有着明显的下滑，在不同的网络丢包率的条件下，语音质量均不如前三种。

4. 包乱序

当网络较差的时候，语音包在传输过程中很容易出现乱序现象，从而影响接收端播放。但是根据每个语音包的时间戳（Time Stamp），可以方便的判断出语音包的发送顺序，通常采用的解决方法同样是在接收端使用抖动缓存，对失序包进行调整，从而重现发端的顺序。

5.语音压缩编码技术

语音压缩编码技术是 IP 电话关键技术的一个重要组成部分。好的语音编码技术可以通过占用较小的带宽来传输优质的话音，特别是在网络环境较差的情况下，这种优势尤其明显。目前，语音编码技术使用比较多的是 ITU-T 定义的 G.xxx 系列语音编码标准。此外 Global IP Sound 公司的 iLBC 算法和 3GPP 的 AMR, AMR_WB 等也有不错的市场占用率。

3.1.3 提高语音质量的几个关键技术

前面介绍了影响QOS的几个主要参数，那么针对这些问题来改善和提高VOIP音频质量的技术亦是多方面的，关键技术有【8】：

1. 语音编码技术

前面我们已经介绍过语音编码是VoIP的基础。好的VoIP编码技术应具有音频编码动态转换功能，即网络拥塞时自动由高码速转换为低码速，网络条件较好自动由低码速转换到高码速以提高音频质量。实际选择编码算法时应综合考虑各种因素。此外语音编码器是在IP电话网中上产生延时的一个重要因素。

2. 抖动处理技术

IP网络的重要特征是网络抖动与网络延时，二者均可导致VoIP音频传输质量下降。 当网络抖动较严重时，部分音频数据包可能因迟到被丢弃，进而导致音频的时断时续及部分失真，严重影响VoIP音质。Jitterbufer抖动缓存，就是为减少抖动而产生的。在将包输出为声音流之前对延时抖动进行吸收。具体而言，收到语音包后不立即播放，而是暂时存储于抖动缓存中直到预定播放的时间，再将缓存中存储的数据包进行规则播放，有效补偿包抖动、包丢失等造成的不利影响，从而改善VoIP的音频质量。特别需指出的是这样处理虽然可使一些迟到的包得以规则播放，但却为早到的包引入了附加延时，因而具体应用时需均衡考虑二者：如果预定截止时间越晚，则可能重放的包越多，且丢包率越低，但缓存延时过高；如果缓存延时设得较低，较高的丢包率又会影响音频质量。

3. 音频优先技术

为确保高质量VoIP通信，在带宽不足的IP网络中可采用音频优先技术，即传输过程中IP网络路由器设置音频数据包为最高优先级。只要路由器发现有音频数据包就将延时对其它数据包的发送，转而传输音频数据包以减少其延时，这样，网络延时及网络抖动对音频质量的影响均将显著减低。此外，还可采用资源预留

协议RSVP为音频通信预留带宽，只要有语音呼叫请求，网络就根据规则为音频通信预留出设定带宽，直到通话结束，带宽才释放。

4. 静音检测技术（VAD, Voice Active Detector)【9】

又称话音活动侦测。人的音频话音突起（talkspurt）包括讲话的时间和静音的间隙，也就是所谓的开关模式。静音检测目的是从声音信号流里识别和消除长时间的静音期，以达到在不降低业务质量的情况下节省话路资源的作用。它是IP电话应用的重要组成部分。静音抑制可以节省宝贵的带宽资源，可以有利于减少用户感觉到的端到端的时延。研究表明，用户打电话时，并不是总在占用通话信道。根据传统电话业务的统计，一方用户实际占用通话信道的时间不会超过整个通话时间的40%。 当检测到突发的活动语音时才生成音频信号并加以处理、传输。采用静音检测技术能有效剔除VoIP中静默信号，既能显著提高网络资源利用率，又可有效提高VoIP音频质量。静音检测有两个技术难点：一是背景噪声问题，即如何在较大的背景噪声中检测静音；二是前后沿剪切问题。所谓前后沿剪切就是还原语音时，由于从实际讲话开始到检测到语音之间有一定的判断门限和时延，有时语音波形的开始和结束部分会作为静音被丢掉，还原的语音会出现变化，因此需要在突发语音分组前面或后面增加一个语音分组进行平滑以解决这一问题。此外，如果出现长时间的静默，会使用户感到很不自然。因此实际上接收端常常会在静音期间发送一些分组，从而生成使用户感觉舒服一些的背景噪声，即所谓的舒适噪声。

5. 回声消除技术

回声是音频信号通过网络时的反射。本地扬声器输出的模拟语音信号可能又被话筒接收，当信号被传回到源端时，就会产生不必要的回声。在传统固话网中，从4线交换到2线本地环路时的阻抗会导致回声，或者是由麦克风和扬声器或耳机之间的耦合效果不好也会导致回声。此外IP网中的呼叫需经过多个路由器和网关，其相当长的延迟会造成回声问题进一步恶化。ITU－T的G.168定义VoIP系统中回音消除技术。其功能是用相位补偿的方法抵消串入远端发送信号中的远端接收信号。其目标是消除延时超过45ms 的回声，因为当回音超过45ms 时，发话方就能够听到反射回来、滞后的自己的声音。如回声返回时间超过10ms 则人耳可感觉到较明显回声，通常IP网络延时40-50ms，显然对VoIP音频质量产生影响。回声消除技术主要利用数字滤波器消除对通话质量影响很大的回声干扰，分回声抑制和回声抵消两种VOIP网络设置回声消除器是建立在低比特率编码器中，且由DSP进行处理。该技术的应用能有效改善VOIP的音频质量。

6. 前向纠错技术（FEC）

前向纠错技术是PLC技术的一种，是用来保护丢失的包的一种机制。VoIP数

据包传输过程中必然存在损坏或丢失，其产生原因包括数据包通过网络传输过程中被破坏；由于网络拥塞网络节点队列已满，被丢弃；由于网络故障而丢失；由于到达接收端太晚而无法包括在重放语音中并被丢弃等。

前向纠错就是在原来的已经数字化的话音块（Chunk）上增加一些冗余信息，所付出的代价是增大了网络上传送的数据率。利用这些冗余信息，就可在还原时将丢失的话音块近似地或精确地重新构造出来。这里的“块”表示应用层的传输单位，而“分组”是网络层的传输单位。

前向纠错的一种方法是将 n 个话音分组编为一组，而每发送 n 个话音分组就增加一个冗余分组。冗余分组是将原来的 n 个话音分组进行异或相加得出。如果在这 n+1 个分组中只丢失了一个，那么在接收端就能够很容易地将丢失的分组恢复出来。当然当丢失的分组数超过一个时就无法恢复了。显然，减小 n+1 的数值就可多恢复出一些丢失的分组，但与此同时也使网络上传输的话音数据率增大得更多了。

前向纠错的另一种方法是在进行正常的话音编码（例如，64kbps 的标准 PCM 编码）的同时，再进行某种低速率的话音编码（例如，13kbps 的 GSM 编码）。然后将低速编码构成的话音块作为冗余数据与正常编码的话音块装配成一个分组一起发送出去。

当一连串丢失好几个话音分组时，可将前向纠错与其它几种恢复方法结合起来。例如，对最后丢失的一些话音分组，可用低速率话音块进行恢复。对一开始丢失的话音分组，可用重复已收到的话音分组进行恢复。对再后面的一些丢失的话音分组，可用静默分组（或相当于讲话的背景噪声）进行恢复。这样做多少可以弥补一下连续话音分组丢失的影响。

3.2 VoIP 系统语音质量的测试方法

IP电话网中的语音质量问题是普通电话网中不存在的特殊问题，前面介绍过影响VoIP语音质量的几个重要参数为时延、抖动、丢包等等。根据理论分析，若使IP电话的语音质量接近普通电话，单向延时的指标最好小于100ms ，最大不能超过250ms，而丢包率不能超过5%。随着IP网络技术的广泛应用，关于IP网络所能提供的业务的服务质量问题受到越来越多的关注，如何来对服务质量进行科学可靠的测量与评价是网络测量与网络规划设计中相当关键的问题。

目前对 IP 电话业务语音质量的评价分为主观评价和客观评价。主观评价方法主要是 MOS 模型（平均评定得分法），还包括判断满意度测量等方法；客观评价方法主要有 PSQM 模型（感知话音质量测量法）、PESQ 模型（感知话音评估法）和 E-Model 等。此外，有必要指出，平均主观值 MOS 是广泛认同的语音质量标准，

因此，无论采用何种方法所有测量方法都必须对应它们的结果对应到最终的平均主观值 MOS，以上各种方法均可以最终以 MOS 值表示。

3.2.1 MOS 模型

ITU-T P.830描述了一种对语音的主观评定方法MOS（Mean Opinion Score）方法。要求是请40～60位不同性别，不同年龄，不同语言的人来听一段相同的语音样本，然后对该样本经过IP电话传输后的语音质量进行投票评价。随着语音因语言、年龄、性别的变化，得分亦被赋予不同的意义。P.830对测试的要求非常严格，所有的操作都要严格地服从操作流程，对录音系统、语音采样、语音输入级别、听者级别、不同发话者(8男、8女、8儿童)、多发话者(多人同时讲话)、差错处理、不同语音编码方式的兼容性、过失、环境噪音、音乐等等，都作出了详细严格的规定。这是一种纯粹主观的定性测量。ITU-T选取在非常宽的听觉范围内，不同年龄、性别和语言组别的相同得分，作出语音质量的判别标准。

MOS评定以5－1的五级评分标准对语音质量进行评定,五级评分标准如表3－3所示

表3-3　MOS五级评分标准

级别	用户满意度
5（优）	很好，听得清楚，时延很小，交流流畅
4（良）	稍差，听得清楚，时延小，交流欠缺顺畅，有点杂音
3（中）	还可以，听不太清，有一定时延，可以交流
2（差）	勉强，听不太清，时延较大，交流重复多次
1（劣）	极差，听不懂，交流常不通

3.2.2 PSQM 模型

MOS 方法是一种模糊的评估方法，其测试结果很难对 VoIP 系统的改进和不同 VoIP 设备之间性能的比较作出有实际意义的判别。因此，有人提出借用 ITU-T 在 P.861 中建议的 PSQM (Perceptual Speech Quality Measurement)方法，用来作为客观质量度量的评估。

PSQM 的客观性是指模仿现实生活中主观声音的感知。PSQM 仿真实验中主观判断话音编码器的质量，通过把编码后的信号和源信号进行比较，PSQM 仍以 MOS 的 5 个级别作为评估结果。PSQM 方法并未摆脱原始的人类主观评估，只是作了进一步的说明。

图 3.3 为 ITU-T P.861 定义的 PSQM 算法的评价模型。首先选取符合条件的基准信号源，可以是真实的声音，也可以是规定的人工语音。把基准信号源和

经过网络的干扰后信号输入到感知模型，这个感知模型实际上是对信号进行时间-频率映射，以及频率和强度偏差处理。从感知模型输出得到的信号内部表现通过差别模型进行处理，为了获得主观和客观之间的较高关联性，再输入到认识模型，最后得到质量评分。从这个评价模型可以看出使用者对语音清晰度的评价主要取决于使用者的认识模型，而使用者的认识模型又是受其感知模型影响。

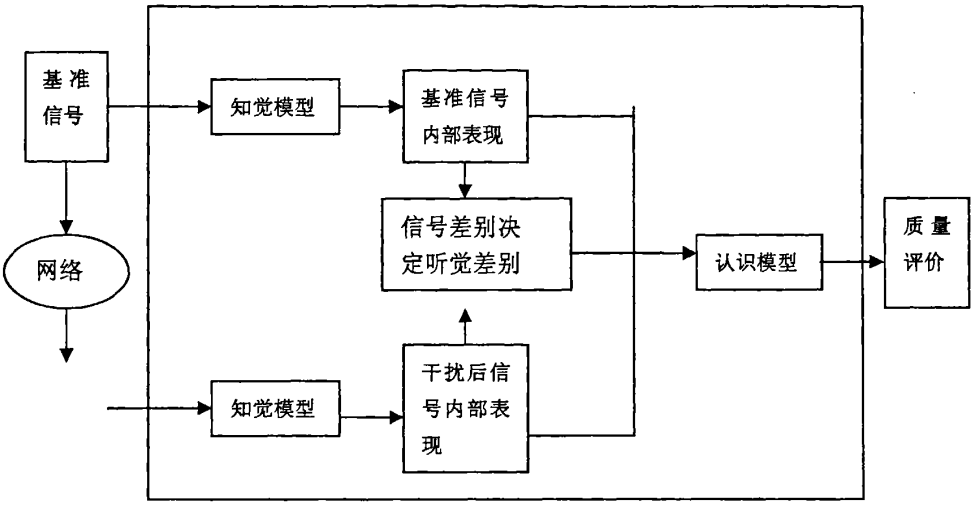


图 3.3　PSQM 算法的平价模型

3.2.3 PESQ

PESQ算法【11】提供了一种更加精确的测量方法。它将PAMS(感知分析测量系统)的时间排列技术和PSQM(感知语音质量测量)的精确的感知模型相结合，这是两种技术最好的特性的结合。

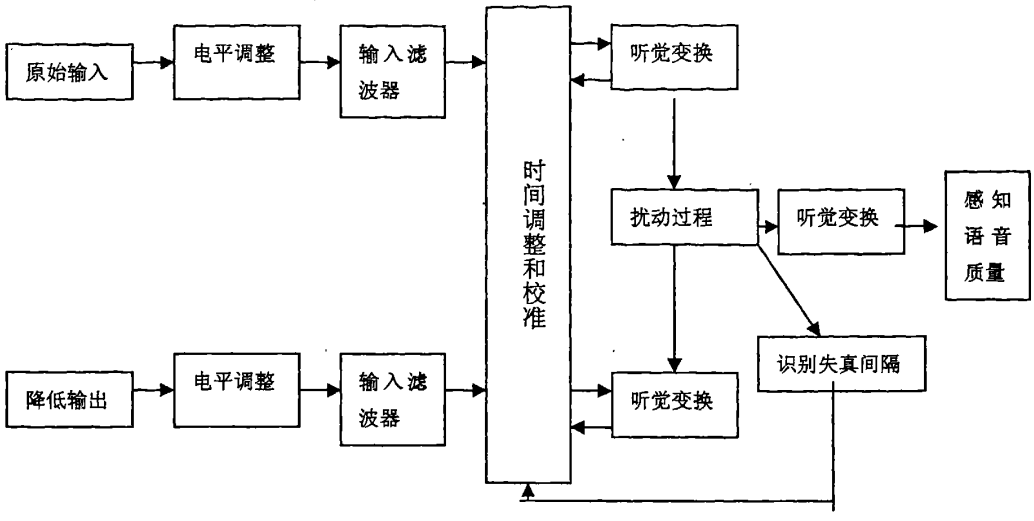


图3.4 PESQ算法模型

图3.4为PESQ的结构。开始时两个信号都通过电平调整，再用输入滤波器模拟标准电话听筒进行滤波(FFT)。这两个信号要在时间上对准，并通过听觉变换。这个变换包括对系统中线性滤波和增益变化的补偿和均衡。提取出两个失真参数，在频率和时间上总和起来，映射到对主观平均意见分的预测。

因为PESQ是插入的端到端的测量算法，需要一个参考语音信号，它只能预报单向的收听到的语音质量。像主观测试采用5点分数一样，PESQ与MOS映射，分数从-0.5到4.5，接近4.5表明语音质量非常好，接近-0.5说明语音质量非常差。

3.2.4 E－Model

PSQM无疑比MOS前进了一步，但其实质仍然是一种定性评估的方法，不能准确给出影响IP电话网语音质量的诸因素的量值如延时、抖动和丢包等，没有考虑网络故障对用户感觉造成的影响，单纯的从收发信号差异的角度分析网络语音问题，这些都是PSQM的局限和缺憾。为了克服这些缺点，国际电联的G.107标准提出了E-Model【10】，它克服了传统语音质量测试在数据网络测量中的不足。为了能够准确评估VoIP语音质量,在E-Model算法的基础之上,探讨了延时，延时抖动,噪声,回音,语音压缩等损伤因素对VoIP语音质量的影响。

E 模型的思想是将话音信号传输过程中若干因素对话质的负面影响综合为参数 R，用以评估该话音呼叫的主观质量。R 的值越大，表明话音质量越好。R 值的计算从没有网络和设备的损伤影响开始，此时语音质量是最好的，$R=R_o$。但是因为网络和设备损伤因素的存在，降低了通过网络的语音质量，R 值的基本计算为：$R=R_o-I_s-I_d-I_e+A$，其中参数 R_o 表示噪音带来的影响，如背景噪音

和电流噪音的干扰。参数 I_s 表示与语音信号同时产生的质量影响因素，如由量化、连接噪声和侧音过强带来的干扰。参数 I_d 表示由于时延造成的质量影响，包括由于通话回声和交互性丧失带来的干扰。I_e 包括由于使用特殊设备引入的质量损失，如低比特率编解码器的影响和分组丢失的影响。G.729A 的 I_e 为 10，G.723.1 在 5.3kbit/s 和 6.3kbit/s 码流速率下的 I_e 分别为 19 和 15。参数 A 为预期值，用以补偿由于用户采用某些带来便捷接入的设备而导致的话音质量的影响。对于传统电话，A 取值为 0；而 GSM 移动电话的 A 值为 10。

根据 E 模型确定可接受话音质量对应的 R 值。编解码器类型、通信模式和传输协议的不同，会使上式中的各个分量有不同的取值，从而得到不同的 R 值。

前面已经讲过，任何的测量方法，最终都将对应为 MOS 值标准，E-Model 也一样。

$MOS=1,$　　当 $R<=0$　　3－1a

$MOS=1+0.035R+R(R-60)(100-R)*7*10^{-6},$　　当 $0<R<100$　　3－1b

$MOS=4.5$　　当 $R>100$　　3－1c

对于等式 3-1b 来说，如果 R 值小于 6.5，那么 MOS 值将小于 1。因此，R 值通常限制在 6.5～100 之间。R 值与 MOS 相应的关系如图 3.5 所示。

图 3.5 语音质量的分类

E-Model 是非插入的不需要参考信号，适合检测实时通信。E-Model 质量评估通常置于抖动缓冲之后，消除设备之前，而 PESQ 置于解码之后，这时信号已经被消除。因此，由于一些类似错误（丢包）消除技术，当丢包率增加时，E-Model 可能并不是十分准确。除此以外，E-Model 用来评估不同编解码算法丢包率的公式是基于成千上万的主观测试，因此实际上不太可行也因此很难得到发展。总之，这两种方法用来评估语音质量的设置可以表示如下：

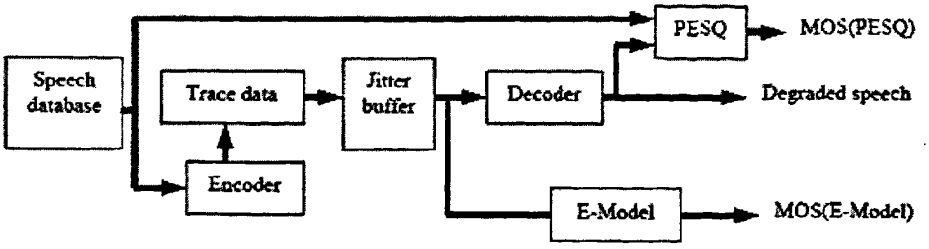


图3.6 VoIP语音质量设置

3.2.5 提出的评估语音质量的方法

PESQ 都有自身的确定或者优点。因此最好能将两者的优点结合起来，来提高语音质量测试的准确性和效率。E-Model 要求主观测试来得到模型参数，然而这非常耗时并且不切实际。因此，E-Model 只在有限的编解码器和网络状况下才适应。PESQ 给出了一个较好的测量语音质量的方法，但是却不适合用来优化。本论文中，提出了一种新的客观测试模型，即用 PESQ 来测试丢包的影响，用 E-Model 测试延时造成的影响。

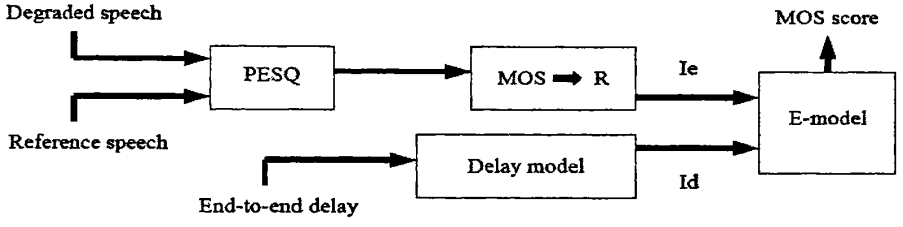


图 3.7 新的语音质量测试模型

图 3.7 表示了这种方法的概念【13】。被降低的语音信号和参考信号经过 PESQ 处理，来得到 MOS 值。E-model 的 R 值可以通过公式 3-1b 转化得到。接下来，I_e 值可以通过 R 值得到。最后，通话的 MOS 可以结合 I_e 和 I_d 通过 R 值得到。 因为这个方法是基于端到端的客观测量而非主观测量，它可以很容易的应用于新的编解码器和网络环境中。

为了方便的从 MOS 值得到 R 值，用简化的三次多项式 3-2 来表示【13】。

$$R = 3.026MOS^3 - 25.314MOS^2 + 87.060MOS - 57.336 \qquad 3\text{-}2a$$

$$I_e = R_o - R \qquad 3-2b$$

不像 I_e 那样是与编解码器无关的，延时损伤因子 I_ρ 对所有编解码器来说是一样的。I_d 可以通过下式得到【13】【14】:

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3) \qquad 3\text{-}3$$

$$\text{当}\begin{cases} H(x)=0 & \text{如果} x<0 \\ H(x)=1 & \text{如果} x\geq 0 \end{cases}$$

d 代表绝对延时（播放延时）。

通过用 I_d 和 I_e，我们可以用 E-model 模型来预测出语音质量。

$$I_e = 20.06\times\ln(1+0.1024\times \text{lossratio})+25.63 \qquad 3\text{-}4$$

本文用 4 种比较常用的编解码算法来解释这一模型，它们分别是 G.723.1, G.729, AMR 和 iLBC(15.2kb/s)。研究中，用到的参考语音数据库是来自 ITU-T 的数据集。

对 ITU-T 数据集中每一个语音的采样，都得到了 MOS 值。这一值是通过对多于 30 个不同的丢包位置（通过不同的随机种子的设置）取平均得到的，这样做的目的是去除丢包位置的影响。此外，每一种丢包率的 MOS 值是通过将所有的语音采样值取平均得到的（一个 16 个样值，8 个男声，8 个女声），这样就能去除性别的影响。4 种编解码算法中，平均 MOS 和丢包率（ρ）之间的关系如图 3.8 所示。

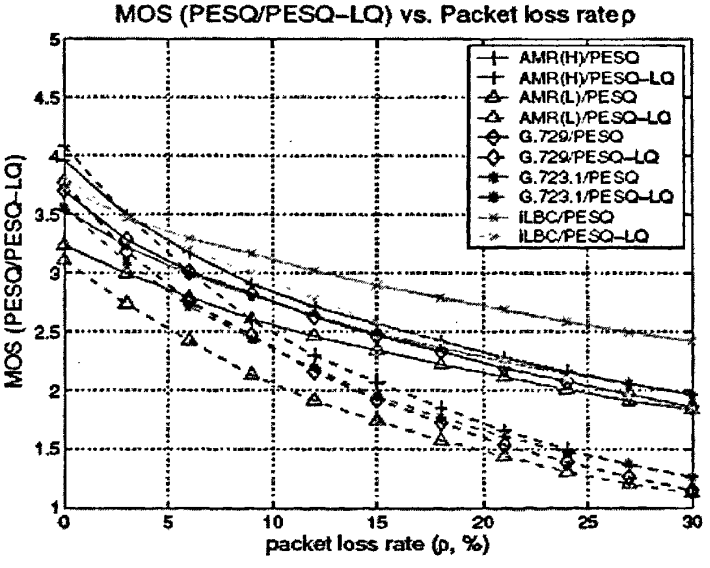


图 3.8 MOS VS 丢包率 ρ

从图 3.8 我们可以看出，当丢包率高时，MOS 值较低。在 ρ 值很高（大于 4 %）时，iLBC 的语音质量最好。当 ρ 为 0 时，AMR（H，12.2kbps）的 MOS 值最高。AMR（L，4.75kbps）不管有无丢包，它的语音质量都是最低的。

考虑到编解码损伤是固定的，I_e 可以表示为 $I_e = I_{ec} + I_{ep}$，I_{ec} 为丢包率为零的损伤，而 I_{ep} 为有丢包率时的损伤。I_{ep} 和 ρ 在 PESQ-LQ 下的关系如图 3.9。

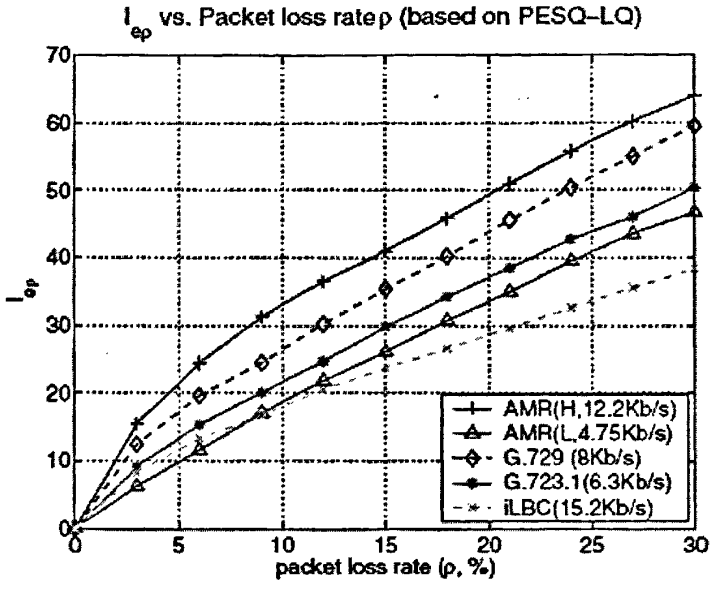


图 3.9　$I_{e\rho}$ vs. 丢包率

图 3.9 说明了一种编解码算法是怎样处理网络丢包的。从曲线我们可以看出，iLBC 的倾斜度最低，而 AMR(H)的最高。这进一步说明了 iLBC 对丢包率有明显的鲁棒性。AMR(H)在零丢包率下有最高的 MOS 值，但是处理丢包的能力却是最低的（质量随着丢包率增加急剧下降）。很显然，随着新的网络编解码器的不断出现，在对处理丢包延时能力方面的多样性会更大。因此，我们推荐对每一种编解码器用不同的模型，来更加精确的优化参数或控制质量。

3.2.6 结论

主观测试可以直接得到语音质量水平，它与用户的观点紧密相关。然而，评估检测真正的 VoIP 通信就会效率低下。相比于主观测试的缺点，客观测试可以几乎满足这些要求，同时又能确保评估的精确性。我们提出了插入和非插入两种方法来评估感官语音质量。PESQ 是一种插入方法， 端到端的测量算法，需要参考语音信号。它可以很好的预测接受的语音质量，但是只能单向。相反，E-Model 是非插入的方法，不需要参考信号，适合检测实时通信。然而 E-Model 用来评估不同编码算法丢包率的等式是基于成千上万的主观测试的，因此不可行。

我们提出将两种方法的优点相结合来提高精确度和效率的评估语音质量方法，特别是通话语音。它用 PESQ 算法来评估丢包率的影响，并用 E-Model 评估包延时。

3.3 本章总结

与其它通信技术相比，VoIP 具有低成本，高效率，节省带宽等优点。然而，由于 IP 网络的一些特性，VoIP 也面临一些挑战，特别是在 QOS 方面。延时，丢包，和延时抖动是三个主要的方面，它们决定了用户的感观质量，其性能取决于采用哪种方法和机制。因此只要这些参数存在，那么研究这些机制和方法就是有必要的，它可以进一步的提高 VoIP 的性能。下一章会具体研究时延抖动的处理方法。

第四章　自适应抖动缓冲算法

4.1 抖动缓冲技术原理

与传统的电话不同，丢包和延时等传输损伤会影响语音质量或通话者间的交流，从而影响整个 VoIP 通信的质量。这些损伤参数很大程度依赖于接收端的播放缓冲机制。这些机制是由卖主定制的而没有统一的标准，从而导致了很多不同的自适应或固定播放机制，每一种都有不同的参数设置。面对如此多的选择，VoIP 设计者和供应商需要有方法来对其进行优化和评估。本章的重点就是研究已有的抖动缓冲算法，并在此基础上提出一种新的算法。

4.1.1 抖动的产生和抖动缓冲

时延抖动（Jitter）是指由于各种延时的变化导致网络中的数据分组达到速率的变化。抖动是 IP 网络的普遍问题，是不可避免的，只能控制在一定范围内。造成抖动的主要因素是网络传输和网络拥塞。与传统的 PSTN 网络不同，在基于包交换的 IP 网络中，两个从相同源地址到相同目的地址的数据包，可以选取不同的路由。不同的路由路径可能经过拥塞状况不同的网络，并且不同的路由器也引入了不同的处理延时，最终可能使到达时刻不同，这样就产生时延抖动和乱序。

IP 网络中大的时延变化会使得接收端对语音信号的重建变得非常复杂。为了补偿抖动，典型的 VoIP 软件会在播放之前，对接收到的包进行缓冲，即在接收方设定一个缓冲区，配合 RTP 协议，通过一定的缓冲时间对乱序的包进行排队和控制。【3】【4】语音包到达时，除取得数据包外，还能得到该数据包的时间戳，通过判断接收缓冲区中语音包的时间戳，就能判定新接收包在队列中的位置，进而判断其是否在缓冲区的范围内，不是就简单丢弃，否则插入相应的位置。随后系统根据某种缓冲调度算法延迟一段时间再以稳定平滑的速率将话音包从缓冲池中取出，经解压后播放给受话者。对于那些晚到的包，缓冲就会将其丢弃。抖动缓冲的大小取决于网络的情况。如果丢包率在 10%以下，那么通话质量会比较好。

为了清楚起见，本文以图例介绍一下缓冲的形成与消除的基本原理【8】:

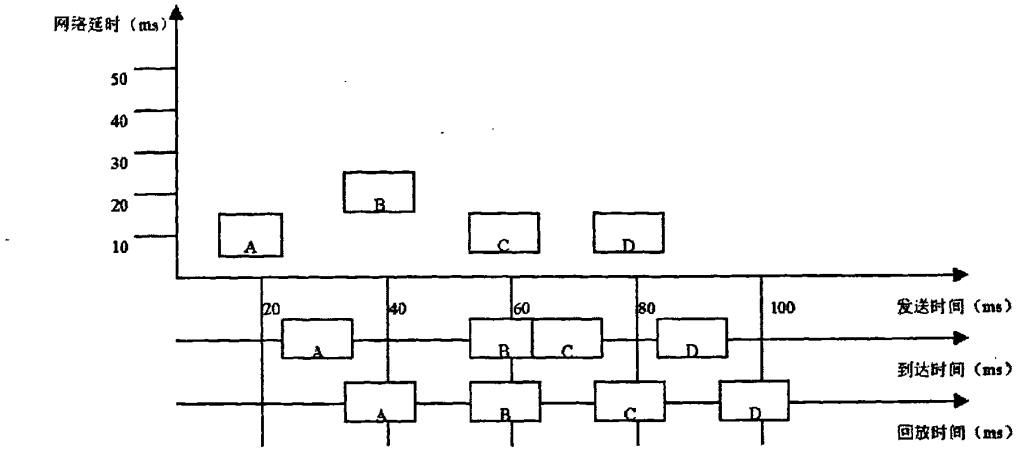


图 4.1 抖动缓冲的形成和工作原理

如上图所示，语音包 A 和 B 在网络上的传输延迟分别为 10ms 和 20ms，那么它们到达接收端也会相应的延时 10ms 和 20ms，如果在接收终端不做任何的处理就直接播放，将会产生语音包乱序播放的后果，进而影响到话端的服务质量。

抖动缓冲的工作原理如图 4.1 所示。A、B、C、D 四个语音包的发送时间分别为 20，40，60，80ms 时刻，经过网络传输后分别产生了 10，20，10，10ms 的网络时延。因此对应到达时间分别为 30，60，70，90ms 时刻。这样为了能够获得流畅的音质，就要求 A,C,D 三个语音包到达后在缓冲器中分别延迟 10ms 再播放。

在语音数据播放前的缓冲处理是与使整个系统延时最小的目标相悖的。如果缓冲时间太短，即使这些包最终到达接收端还是会被丢弃。如果缓冲时间太大，能抑制的网络抖动和乱序的能力就越大，但产生的额外延时也越大，这对用户来说是不能忍受的，所以需要在延时和抖动间取得较好的平衡。由此可见确定播放延时是一个难点。

4.1.2 消除时延抖动的模型

抖动缓冲技术主要组合了 3 种机制。

1、序列号:发送者在每个分组产生一个递增的分组序列号，RTP 分组中就是 16 位的序列号域。

2、时间戳:发送者在每个分组产生的时间。

3、在接收方延时播放:为了使大部分数据分组在播放前能够被接收，接收方的播放延时应该有一定的值。

为了确定话音包的正确时间间隔，在 RTP 的包头上提供了一个时间戳(time stamp)，用于记录包的到达时间以及本地的播放时间。

假设有两个数据分组，其序列号分别是 i 和 j（j > i），则对时间戳约定如下：

1. Si:数据分组 i 的帧头部时间戳(即发送时间)；
2. Ri:数据分组 i 到达本地的时间戳；
3. Pi:数据分组 i 播放的本地时间戳；
4. J：抖动缓冲区大小值；
5. JTi：表示数据分组 i 在接收端需要缓存的时间；
6. Jn(i,j)＝（Rj －Sj）－（Ri－Si）：数据分组 i，j 之间产生的网络时延抖动，它的大小由网络状况决定，并且是决定抖动缓冲区大小的重要参数；
7. Jp(i,j)＝（Pj －Sj）－（Pi－Si）：数据分组 i，j 之间产生的播放时延抖动；它应该尽可能的减小来保持语音流的连续性。
8. Jj(i,j)＝（Pj －Rj）－（Pi－Ri）：数据分组 i，j 之间产生的时延抖动；
9. Dr（j，i）＝Rj－Ri：两个数据分组 j 和 i 到达接收端时的时间间隔，它随网络抖动和语音包发送端间隔 Ds（j，j－1）改变。
10. Ds（j，i）＝Sj－Si：两个数据分组 j 和 i 在发送端的时间间隔，若在语音段中只有固定速率的语音流，则语音流传输间隔 Ds(j,j-1)总是一个固定值；
11. Dp(j,i) = Pj-Pi：数据分组 J 和 i 在接收端的播放间隔，理想情况下，Dp(j,i)应该等于 Ds（j，i）。

如果不存在网络时延抖动，应该得到如下关系式：即 Jn(i,j)=0。在这种理想情况下，到达接收端的语音包不需要进行缓冲处理便可直接播放(Pj－Pi＝Rj－Ri)。但在实际情况中，由于存在时延抖动，通常是 Jn(i,j)＝（Rj－Sj）－（Ri－Si）≠0，此时如果直接播放就会产生语音失真，所以必须对语音包 j 缓存 JTj 时间后再播放，即播放时刻应该是 Pj=Pi+(Rj-Ri)+JTj,这样才可以使语音得以真实复现。由此可见，进行抖动缓冲的目标就是:使语音分组播放的时间间隔等于它们实际发送的时间间隔，Pj-Pi=Sj-Si。即 Jp（i，j）＝0。

以图 4.1 为例，对于语音分组 C 和 D，由于二者的网络时延相同(均为 10ms)所以有 Rd-Rc=Sd-Sc，可以不作处理直接播放。但对于语音分组 B 和 C，由于二者的网络延迟不同，分别为 20ms 和 10ms，则两分组到达的时间间隔与发送的时间间隔不一样：（Rc－Rb＝10ms）≠（Sc-Sb=20ms），存在时延抖动 Jn（c，b）＝10ms。利用播放公式 Pc=Pb+(Rc-Rb)+Tn（b，c），如果不进行抖动缓冲处理而直接播放，即 JTc=0，就有下式:Pc-Pb≠Sc-Sb。很显然，这会导致播放过程与

实际发送不符，造成语音失真。所以必须在播放分组 C 之前先对其进行缓冲 JTc=10ms，这样 Pc-Pb=Rc-Rb+Tn(b,c)=Sc-Sb=20ms，语音可以完整重现。

在使用静态抖动缓冲控制时，缓冲区的大小是一个固定值。在上例中，如果令抖动缓存值 J=15ms，大于 A，B，C，D 四组语音包间产生的最大网络时延抖动 Jn(i,j)=10ms，则语音能够完整平滑地复现，但此时语音延迟较大；而如果令 J=5ms，那么语音分组 B 将会被丢弃，此时语音流将表现为断续地平滑。而在一个自适应抖动缓冲算法中，它的缓冲区大小是根据当前网络情况来动态调整的，所以不会存在像静态抖动缓冲控制中一旦缓冲区初始值设置不合理便出现丢掉大量语音分组的问题，且效率较高。

4.2 抖动缓冲常用算法介绍

4.2.1 抖动缓冲算法分类

一种好的播放算法应该能够将缓冲延时最小化，丢包最小化，以此来达到丢包和延时间的平衡。目前的缓冲算法研究基本上可以分为两大类：

一．固定缓冲算法

固定缓冲算法在一个语音会话持续期间为每个语音包都设定了固定的缓冲时间，并不会随着网络的变化而进行缓冲时间调整，如果在规定时间点上其对应语音包因时延抖动没有到达，则会被丢弃。其实现原理见上一小节的抖动模型。该算法的优点是：模型简单，易于实现，可靠性高，是 IP 电话最初的应用中的实现方法；缺点是：很难适应网络状况随时间的显著变化，实际应用中，经常会出现在网络延时较小的区间缓冲延时时间过长的状况，浪费了延时性能。而网络延时大的区间却用同样大小的固定延时，造成了严重丢包现象。不能提供 VoIP 这样的实时交互性语音应用所需的服务质量。

二．自适应抖动缓冲（AJB）算法

根据接收缓冲区中的数据包或 RTCP 提供的参考数据来衡量网络状况，在每一个话音突起(talkspurt)的开始调整延时播放时间。当网络状况好、抖动较小时，减小缓冲时间，以减少总体延时。反之则增加缓冲时间，以延时增加的代价来取得更好的抑制抖动的能力。该算法的优点是：具有较好的网络自适应性，会获得较好的延时和丢包平衡。 因此一个 VoIP 解决方案要想取得竞争性，那么 AJB 是唯一的选择。因此本论文主要研究自适应缓冲算法。

4.2.2 已有的抖动缓冲算法介绍

本节回顾和分析了【3】【4】【6】【20】中提出的 7 种自适应缓冲算法。主要研究尖峰检测和相关的播放调整。最后，讨论了缓冲算法的几个关键问题。

为了清楚的描述缓冲算法，我们定义了一系列参数，如图 4-2 所示。t_i 为发送第 i 个包的时间，a_i 为包到达的时间，p_i 为播放时间，n_i 为包的网络延时，d_i 为播放延时（端到端延时）。缓冲延时为 b_i。

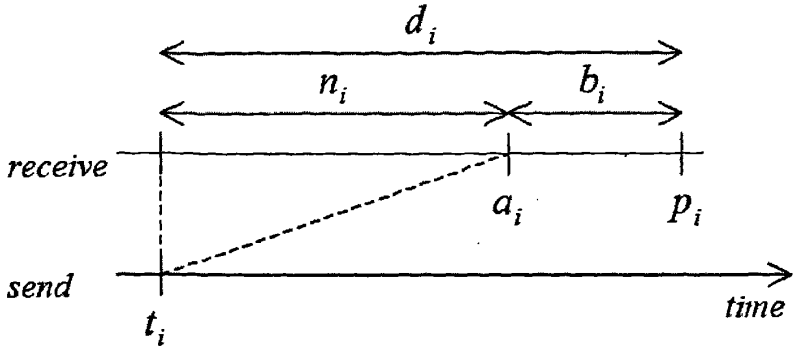


图 4—2 与第 i 个包相关的时间

算法一 指数平均（Exponential-Average）【3】

此算法中，第 i 个包的总延时 d_i 是通过 RFC793【19】计算得到的，单向延时 n_i 是由网络引入的。估计包延时 $\hat{d}_i$ 和流变化 $\hat{v}_i$ 用来估计下一个包的播放延时 $\hat{p}_i$

$$\hat{p}_i = \hat{d}_i + 4\hat{v}_i \qquad 4\text{-}1$$

$\hat{d}_i$ 和 $\hat{v}_i$ 分别为

$$\hat{d}_i = \alpha\hat{d}_{i-1} + (1-\alpha)n_i, \qquad 4\text{-}2$$

$$\hat{v}_i = \alpha\hat{v}_{i-1} + (1-\alpha)\left|\hat{d}_i - n_i\right|, \qquad 4\text{-}3$$

这个算法基本上是一个线性递归的滤波器，有一个权重因子 α，其值为 0.998002【3】。

算法二 快速指数平均 （Fast-Exponential-Average）【3】

这个算法与第一个相似。唯一的不同就是当当前包的网络延时 n_i 比 $\hat{d}_{i-1}$ 大时，$\hat{d}_i$ 等式为：

$$\hat{d}_i = \beta\hat{d}_{i-1} + (1-\beta)n_i, \qquad 4\text{-}4$$

根据【18】【3】可知 β 等于 0.7500000。

算法三 最小延时 (Min-Delay)【3】

这一算法的目的是将延时最小化。它用当前话音突起中所接收到所有的包的最小延时作为$\hat{d}_i$来预测下一个音频话音突起的播放延时。S_i代表音频话音突起期间收到的所有的包。$\hat{d}_i$由下式计算得出

$$\hat{d}_i = \min_{j \in s_i} \{n_i\} \qquad 4\text{-}5$$

算法四 尖峰检测(Spike Detection)【3】

这种算法基于尖峰检测机制。所谓尖峰就是端到端的网络延时突然大幅增加，随之大量的包几乎同时到达，从而导致出现尖峰。算法由图 4-3 给出。如果包到达的延时大于门限值，那么算法便会转为尖峰模式。当检测到尖峰时，延时估计就会跟踪它，并且调整它的倾斜度。当值很小时，算法就会切换回正常模式。如果不是尖峰模式，那么这种算法除了α值为 0.875,其它与算法一相同。

n_i：第i个包的网络延时
IF (mode == NORMAL)
　IF(abs($n_i - n_{i-1}$)>abs(v_i)*2+800)
{　var = 0;
　mode = SPIKE;}
　ELSE
　{ var=var/2+fabs($2n_i - n_{i-1} - n_{i-2}$)/8;
　　{IF(var<=63)
　　mode = NORMAL;
　　$n_{i-2} = n_{i-1}$;
　　$n_{i-1} = n_i$;
　　Return;
　　}
　}
IF (mode == NORMAL)
$\hat{d}_i = \alpha\hat{d}_{i-1} + (1-\alpha)n_i$;
ELSE
$\hat{d}_i = \hat{d}_{i-1} + n_i - n_{i-1}$;
$\hat{v}_i = \alpha\hat{v}_{i-1} + (1-\alpha)\left|\hat{d}_i - n_i\right|$;
$n_{i-2} = n_{i-1}$;
$n_{i-1} = n_i$;
return;

图 4-3 尖峰检测算法

算法五 窗【4】

和算法四相同，这种算法同样基于尖峰检测。此算法收集以前收到的包的网络延时用来估计播放延时。算法会记录下最后几个包的延时，延时分布用每个话音突起更新。算法用到一个增加的柱状图。当有新的包到达时，最老的包的延时将会被从柱状图中去掉，新的包的延时就会加进去。延时分布计算用到频率的累加和，并且只在新的话音突起才会进行计算。图 4-4 介绍了此算法的概念。尖峰期间，它只是简单用到此尖峰中当前音频话音突起（talkspurt）的第一个包作为这个音频话音突起（talkspurt）的播放延时。若不是尖峰模式，算法计算最后收到的包的第q个分位数来估计播放延时。根据【4】，q的值为 0.99，w为 10000。此外，【4】对两个相邻音频话音突起可能发生的碰撞进行了详细说明。

若发生碰撞，则增加播放延时确保后来音频话音突起中所有的包都不会被丢弃，因为相邻两个音频话音突起（talkspurt）的包可能有相同的播放时间。

```
IF (mode = = SPIKE)
    IF ( $\hat{d}_i^k \le tail * old\_d$ )  /*尖峰的结束*/
        mode = = NORMAL;
ELSE
    IF( $\hat{d}_i^k > head * \hat{p}_k$ )  /*尖峰的开始*
        mode =SPIKE;
        old_d =  $\hat{p}_k$ ;     /*保存 $\hat{p}_k$ 来检测后来的尖峰*/
    ELSE
        IF(delays[curr_pos]≤curr_delay)
            count -= 1;
        distr_fcn[delays[curr_pos]] - = 1;
        delays[curr_pos]= $\hat{d}_k^i$ ;
        curr_pos = (curr_pos+1) % w;
        distr_fcn[ $\hat{d}_k^i$ ] += 1;
        IF (delays[curr_pos]< curr_delay)
            count += 1;
        WHILE (count< $w \times q$ )
            curr_delay += unit;
            count += distr_fcn[curr_pos];
        WHILE (count> $w \times q$ )
            curr_delay - = unit;
            count - = distr_fcn[curr_pos];
```

图 4-4 窗算法

此算法同样检测脉冲，当检测到脉冲时，就会停止收集包延时。如果在尖峰中有新的话音突起，它就会用话音突起的第一个包的延时作为那个话音突起的播放延时。记录下来的包延时 w 决定了算法随网络变化的灵敏度如何。如果太小，算法就可能对播放延时估计不对。如果太大，它就将跟踪大量不必要的历史数据。

算法六 增强的基于 MOS 的算法 （Enhanced-MOS-based）【6】

与其它算法不同，它的主要目的是将感知语音质量最大化。它将延时和丢包两个参数直接映射到 MOS 函数，表达式为

$$MOS = 4.10 - 0.195l + 2.64\times10^{-3}d - 1.86\times10^{-5}d^2 + 1.22\times10^{-8}d^3 \qquad 4-6$$

以此来获得播放延时的最优值d来最大化 MOS 值。丢包率l由下式得到

$$l = l_n + l_d \tag{4-7}$$

l_n为网络中由包丢失引起的丢包率，l_d为由抖动缓冲丢弃的晚到的包的丢包率。l_d可以表示为

$$l_d = 100\left(\frac{\hat{k}}{d}\right)^{\hat{\alpha}}, \tag{4-8}$$

$\hat{k}$为最后收到的w个包的网络延时最小值.(根据【6】，w的最大值为 10000)。$\hat{k}$可以表示为

$$\hat{k} = \min_{j \in w}\{n_i\} \tag{4-9}$$

α由等式 5—10 得到

$$\hat{\alpha} = w\left[\sum_{i=1}^{w}\log\left(\frac{n_i}{\hat{k}}\right)\right]^{-1} \tag{4-10}$$

因为只包括一个未知参数播放延时d（别的参数都可通过网络延时、包序列号等等已知数据推理或计算得到），d的值可以在 MOS 最大时得到。

算法七 最大化 MOS（Maximise-MOS）【20】

这一算法基于延时的历史信息，利用 MOS 函数来达到最好的平衡。它捕获播放延时和最后收到的w个包的丢包率作为滑动窗口来最大化以前的 MOS 值，然后获得相应的网络延时作为以后的播放延时，与算法六用泊松分布来预测丢包率不同。这种算法用非常简单而又灵敏的方法来检测尖峰。像算法五一样，同样包含碰撞处理部分。根据【20】，滑动窗设为 2 到 200 秒。

尖峰检测和窗口算法是用来检测尖峰和调整尖峰时的播放延时的。尖峰检测设置了一个门限值来检测尖峰的开始和结束。在窗算法中，如果新的话音突起发生在尖峰时刻，那么此话音突起的第一个包的网络延时就会被用做下一个话音突起的播放延时。对于可变的网络延时，此算法不能很好的调整尖峰因为算法中 head 的值是固定的。对播放缓冲预测的不足可能会导致包因为晚到而被丢弃。

对于这些缓冲算法来说预测缓冲延时的方法大概可分为两种：基于网络参数（1—5）的和 MOS 的（6，7）。最近的研究表明，缓冲算法的设计动机已经着眼于改进用 MOS 值来表示的感知语音质量。另外，预测方法也可以分为两种：总的延时调整和尖峰延时调整。有的算法只有前面一种如 1，2，3，6，而其余的两种都有。下一节将重点研究尖峰检测和尖峰相关的播放延时调整。

4.2.3 尖峰检测和播放延时调整

正如前面所提到的，尖峰检测，窗口，M-MOS算法都能够用来检测尖峰，调整尖峰期的播放延时。它们都设置门限值来检测尖峰开始和结束。这三种算法的检测尖峰开始和结束的方法可以总结如下：

```
IF (mode ==SPIKE)
{ IF (X<EXIT)
    mode = NORMAL;}
ELSE
{ IF (Y>ENTER)
    mode = SPIKE;},
```

EXIT和ENTER分别是检测尖峰开始和结束的标志。X和Y是由每一个算法分别定义的值。

对于尖峰检测算法来说，ENTER为“$2|\hat{v}_{i-1}|+E$”ms，Y为$|n_i-n_{i-1}|$ms。$\hat{v}_i$的值是为了跟踪不同的网络环境，但是因为其相对较小，ENTER的值基本上取决于E的值，【3】中E取值为800。这会使得尖峰检测的效率低下。当这个值比较大时，这个算法可能会引起大量的丢包，因为算法不能跟踪这么小的或者中等的延时变化。然而，如果将E设小，它可能会检测到错误尖峰过高估计了延时。因此，即便E是经过成千上万的试验结果得出的，如果改变网络环境，还是不能适应变化的网络延时。

对X来说，算法用到了一个var因子，

$$\mathrm{var}=\mathrm{var}/2+\left|(n_i-n_{i-1})/8+(n_{i-1}-n_{i-2})/8\right| \qquad 4\text{-}11$$

在一个尖峰中，每接收到一个包就会计算因子var。如果其值比EXIT设置的63小【3】，那么意味着尖峰的结束，var就会被置为0以检测下一个尖峰。在SPIKE模式，$\hat{d}_i$的计算变为

$$\hat{d}_i=\hat{d}_{i-1}+n_i-n_{i-1} \qquad 4\text{-}12$$

窗算法将ENTER定义为head*$\hat{p}_i$，$\hat{p}_i$是最新的播放延时，X为当前的包网络延时。对于尖峰的结束，EXIT被定义为tail*old_d。old_d因子是检测到尖峰时播放延时$\hat{p}_i$的最高记录。head和tail的值由【4】决定，head从2到20，tail从1到4。如果在尖峰中间有一个新的会话突起，会话突起的第一个包的网络延时将被作为这个话音突起的播放延时。

与尖峰检测的问题类似，窗算法的head值是固定的。如果平均播放延时很小，选择小的head值，那么ENTER值将会很小，这意味着门限值会非常低。反过来，当head的值选的很大，有大的平均延时时，ENTER值也会很大。因此，如果网络延时发生变化，这种算法就不能很好的调整尖峰。此外，此算法只考虑到尖峰的

一定的特性。最大的网络延时可能发生在尖峰期间。如果这样的话，即使窗算法检测到了尖峰，对播放延时预测的不足也会使得很多包因为晚到而被丢弃。

M-MOS和窗算法相似，都设置ENTER值。唯一的区别时它将head设为1。当网络延时变化很大时，这种检测尖峰方法的灵敏性可能引起算法的主要部分不能很好得工作。EXIT的设置和窗算法完全相同。然而，因为最大化的MOS函数的主要部分的特性，经过一个尖峰之后，播放延时将会慢慢减小。这就会使得接下来无尖峰的话音突起的延时变大，降低整个语音质量得性能。

4.2.4 总结和建议

本章研究了7种自适应缓冲算法。调整播放延时的方法可大体分为两种：整个延时调整和尖峰延时调整。一些算法只有第一部分如1，2，3，6，而其它的都有。

这些算法的预测播放延时主要基于历史的延时信息（有或没有丢包估计）和一些突发事件，如尖峰等。所有的播放延时的估计都着眼于丢包和延时平衡。最近的研究表明，现在缓冲算法设计的动机和性能的评估都注重语音质量，因为这个与用户的关系更加直接。

对总的延时估计来说，算法1-5没有考虑到和感知语音质量联系起来。算法6，7都用到了MOS函数来预测播放延时。然而，算法六基于假定包丢失和延时对语音质量的影响与MOS是成线性的，这一点是值得怀疑的。用线性损失函数也有可能是错误的。而算法七的问题在于尖峰检测和直接用到历史的网络延时作为播放延时，这可能使得估计不足或估计过大。另外，一个尖峰之后，播放延时的缓慢降低可能引起整个缓冲性能的不足。

在研究了这个概念和问题之后，本文总结出来一个好的缓冲算法应该包括以下这些方面：

1. 应该与感知语音质量相关。
2. 能够适当的跟随网络延时的变化。
3. 当可变延时小的时候能够取得好的平衡。
4. 有灵敏的尖峰检测机制。
5. 当发生尖峰时可以预测到合适的播放延时。
6. 两个连续的包或话音突起间可以避免碰撞。
7. 发生尖峰之后可以以适当的速度降低播放延时。
8. 能适应与任何网络环境。
9. 对每一个收到的包能够有高的计算速度。

4.3 提出一个自适应缓冲算法

前面的几节回顾讨论了以前的一些算法。因为还存在一些问题，本节提出了一种新的算法。

4.3.1 建立 MOS 函数模型

如第三章所述，R 因子表达式简化为 $R = R_o - I_e - I_d$，包括两它个参数：I_e 和 I_d. 如果将R代到等式3－1b中，可以获得一个新的等式：

$$MOS = 4.409 - 0.0194 \times (I_d + I_e) - 0.837 \times 10^{-3} \times (I_d + Ie)^2 + 7 \times 10^{-6} \times (I_d + Ie)^3 \qquad 4\text{-}13$$

当 $6.8 < I_d + I_e < 86.7$ 时能够较好的工作。编码算法为G. 723. 1时 I_d 可以用3-4表示，I_e 用3-5表示。

丢包率由两个参数组成：一个是由内部的丢包引起的，另一个是被抖动缓冲丢弃的。因此，丢包率可表示为

$$I = I_d + I_n \qquad 4\text{-}14$$

I_d 为被抖动缓冲丢弃的包的丢包率，I_n 为在网络中丢掉的包。

在实际网络通信中，播放延时 d 和丢包率 I_d 有一定的关系。它可以表示为泊松分布4-8。因子 k 和 α 的计算由4-9和4-10表示。

因为只有一个未知数播放延时 d（别的参数可以通过包网络延时，包序列号等已知数据推算出来）。当MOS最大时，可以得到 d 的值。然后，只有当话音突起开始时，d 值才会被设为相应的话音突起播放延时。

对真正的IP网络来说，播放延时 d 可以设为指定的值，但是有一个自适应的范围。限制 d 值是为了确保高的计算速度并且避免过高的估计播放延时。为了解决两个相邻的话音突起的碰撞问题，本文选择增加播放延时来避免后来的话音突起的丢包。【4】

这种不考虑尖峰的算法可以总结如下：

1. 测量每一个包的网络延时，记录最后的 w 个包中最小的网络延时。
2. 计算泊松分布参数 α，然后将 α 和 k 代入MOS函数。
3. 找到最理想的 d 值来最大化MOS值，当话音突起开始时将 d 设为播放延时
4. 返回步骤1

4.3.2 尖峰发生时调整播放延时的新方法

以前的Spike-Det，窗，M-MOS算法在网络条件变化的情况下，尖峰检测机制都不能很好的工作。检测尖峰时，这三种算法看起来都过于固定。为了克服这一问题，本文提出了一种新的调整尖峰的方法。该方法参考了Spike－Delay调整并基于MOS。

泊松分布中，参数 $\hat{\alpha}$ 的大小随延时变化的程度而改变。网络变化越小，$\hat{\alpha}$

值越大，反之亦然。如果用等式4-10来得到$\hat{\alpha}$，在稳定的网络环境中，其值接近于100，反之，在变化的网络中$\hat{\alpha}$值大多数时间都在5左右。

因为播放延时通常是在变化的延时情况下估计值高，在稳定的延时情况下估计值低，所以门限值应该适合这些变化。对于稳定的网络，可以将门限值设低；对于波动的网络可以将门限值设高。因此，参数$\hat{\alpha}$可以被用来设计自适应的门限值大小。

为了方便理解，用图4－5C语言伪代码来表示这一算法。对于SOMOSA算法来说，ENTER门限值设为

$$ENTER = \hat{k} - 0.006\alpha^2 + 118 + T(\hat{k})ms$$

$$\text{当}\quad \begin{cases} T(\hat{k}) = 0 & \text{若} k \le 150ms \\ T(\hat{k}) = 150 \times \ln(k/150) & \text{若} k > 150ms \\ \hat{\alpha} = 100 & \text{若} \hat{\alpha} > 100 \end{cases} \qquad 4\text{-}15$$

Y值伪当前包网络延时。

d_i:第i个包的播放延时

n_i:第i个包的网络延时

E_i：第i个包的估计播放延时

IF(mode = =NORMAL)

{IF (Y>ENTER && 是会话峰的第一个包) //检测尖峰的开始

{ var =0;

mode = SPIKE;

d_i = para * n_i;

重新收集除了$\hat{k}$以外的数据，保存之前w个包的数据以作为参考;

}

}

ELSE

{ var = var/2 + fabs($2n_i - n_{i-1} - n_{i-2}$)/8;

IF (X<EXIT) //检测尖峰的结束

{ mode = NORMAL;

IF($n_i \le door$×之前 NORMAL 模式下会话峰的播放延时)

重新收集除了$\hat{k}$以外的数据，保存参考数据作为分布函数;

}}

测量网络延时$\hat{k}$和$\hat{\alpha}$;

用 MOS 函数降低E_i;

IF （$d_i \neq$ para * n_i && 为会话峰的第一个包）

$d_i = E_i$;

图4-5 SAMOSA算法

ENTER值的设计主要是基于固定延时和可变延时间的某种关系。当固定延时增加时，可变延时的相对范围也会变大。当固定延时小时，可变延时的范围也相对小。另外，参数$\hat{\alpha}$给出了可变延时的级别，以此来避免ENTER值的设置太敏感或者太大而不适合检测尖峰。

在此必须注意，算法只在话音突起的开始检测尖峰。原因如下：

- 在会话峰中调整尖峰可能不会提高语音质量。有时候反而会拉长会话峰，引起碰撞或者下一个会话峰的延时。
- 如果尖峰发生在一个会话峰内，它可能不会跨越相继而来的几个会话峰。换句话说，下一个会话峰网络延时的变化可能是稳定的。

即使跨越了下面的会话峰，这一新的算法仍能检测到尖峰的发生。

当检测到尖峰时，算法重新收集网络延时，然后重新计算参数$\hat{\alpha}$，但是保留参数$\hat{k}$的值并且保存收集到的最后w个包的数据（网络延时和序列号）以作为参考。播放延时设为$para$×当前包的网络延时，这里参数para的值是可变的。

本文倾向于用LRT(Linear Regression Trendline)方法来预测网络延时，然后来推导参数para的值。计算LRT时选择最后n个包的网络延时。计算方法在【27】中有详细的描述，等式如下：

$$\phi=\partial T+\theta \tag{4-16}$$

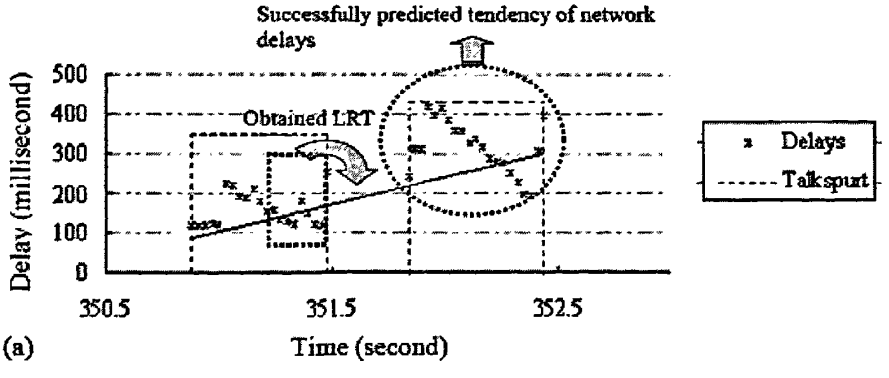
ϕ为网络延时的衰退，T为VoIP系统的运转时间，∂和θ可通过下式获得：

$$\partial = \frac{\sum_{i=1}^{n} xy - \frac{\sum_{i=1}^{n} x \sum_{i=1}^{n} y}{n}}{\sum_{i=1}^{n} x^2 - \frac{(\sum_{i=1}^{n} x)^2}{n}} \tag{4-17}$$

$$\theta = \bar{y} - \partial \bar{x}, \tag{4-18}$$

这里y为最后n个包的网络延时，x为每个包相应的发送时间。

下图展示了几个成功的预测后来的会话峰的例子。图中线性的线代表从上一个会话峰中最后n（n设为10）个包的网络延时中获得的趋势。



(a)

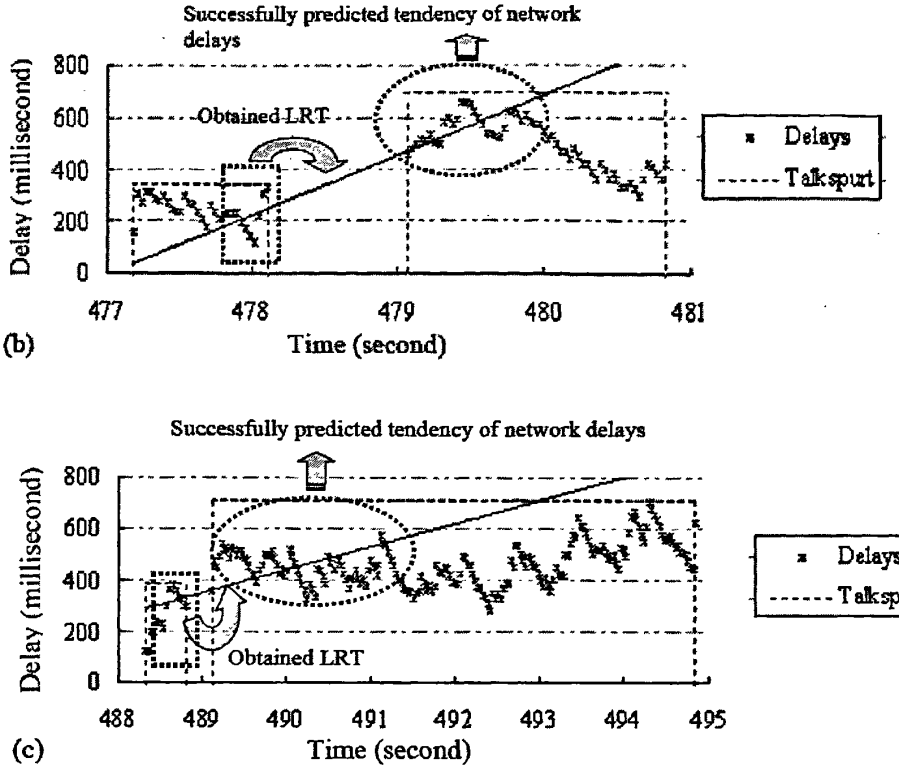


图4-6 预测下一个会话峰的网络延时的例子

为了实现这一方法，本文假设这样情况，即当前会话峰的第一个包的网络延时比预测的网络延时大或小20%以内，从最后N个包的数据来看LRT有明显的上升趋势。如果一个包满足了这两个条件，根据当前包和上一个会话峰的最后一个包的时间间隔，就可以得到para的值。计算方法如下：

$$\begin{cases} para = 1.7 - 0.4\times10^{-3}T & 若T \le 1500ms \qquad 4-19 \\ para = 1.1 & 若T > 1500ms \qquad 4-20 \end{cases}$$

T为时间间隔。如果其中任意一个条件部满足的话，para设为1.1。

自适应的设置相关播放延时是为了保证缓冲有足够的时间来播放会话峰的第一个包，以此来降低同一个会话峰中后来的包的丢包率。同时避免对播放延时估计过高。

检测尖峰结束的方法和尖峰检测算法类似。算法同样用因之var作为X。计算方法于【3】相似。EXIT的值设为20。当X<EXIT时，增加了一个条件来辨别它是一个瞬时尖峰还是会持续很长时间。思想表现如下：

IF (X<EXIT)

{ IF (当前的包网络延时>door x 前面一个NORMAL模式下的会话峰播放延时)

它可能是长时尖峰；
ELSE
瞬时尖峰；}

当算法检测到尖峰结束时，如果时瞬时尖峰，算法同样重新收集相关数据，同时用参考数据来计算参数$\hat{\alpha}$，但是保留参数$\hat{k}$的值来避免过高或过低的估计下一个会话峰的播放延时。如果不是瞬时尖峰，为了降低估计播放延时降低的速度，只有从尖峰一开始收集到的数据会被保留下来，以跟踪网络延时的趋势。

4.4 总结

本章我们研究了抖动的形成和解决方法。研究分析了以前提出的几种常用的抖动缓冲算法，在此基础上我们提出了一种新的自适应缓冲算法来适应不同的网络环境。

新的算法包括两个模式：正常模式（NORMAL）和尖峰模式（SPIKE），分别用不同的方法来处理。对于正常模式，我们改进了以前的概念，基于感知语音质量来研究。此外，对已有的三种尖峰算法的研究，使我们获得了尖峰模式下设计尖峰检测和延时调整的全方面的概念。

第五章　系统设计和测试

前几章介绍了VoIP的基本原理和关键技术,以及影响其通话质量的主要因素,并针对抖动这一现象提出了一种新的抖动缓冲算法来提高语音质量。本章将重点介绍构成VoIP媒体播放系统的各个模块，然后在这个基础上对提出的自适应抖动缓冲算法进行测试。

5.1 VoIP 系统框图

图5.1为本文设计的整个VoIP系统的接口，其中主要包括以下几个部分：

- VoIP SDK驱动
- 媒体处理器
- VoIP网络服务（TCP/IP协议栈）
- 驱动接口API
- 媒体处理器API

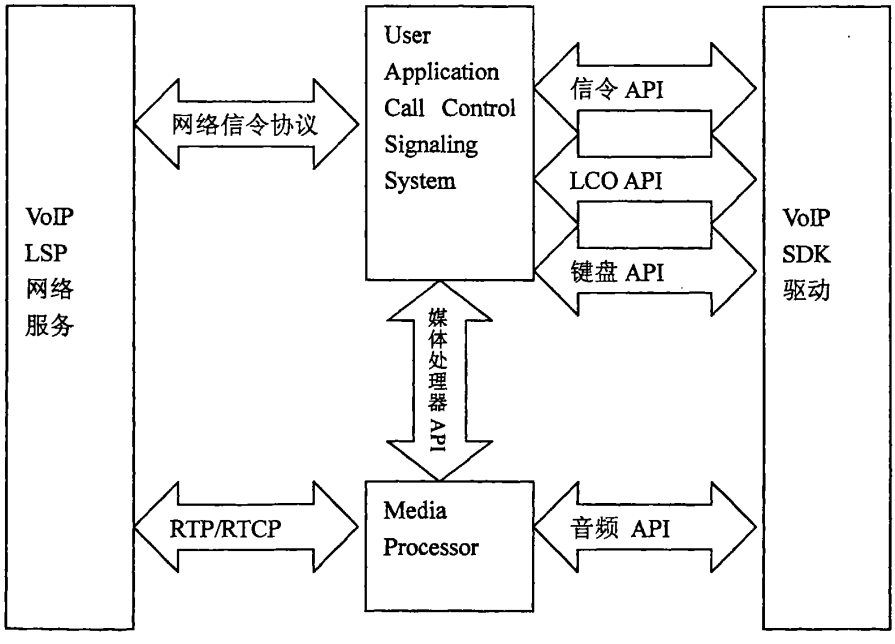


图5.1 VoIP系统接口

5.1.1 MP（Media Processor）

因为本论文研究的主要是VoIP的语音质量，所以我们将着重描述其媒体处理

器部分，如图5.2 所示。

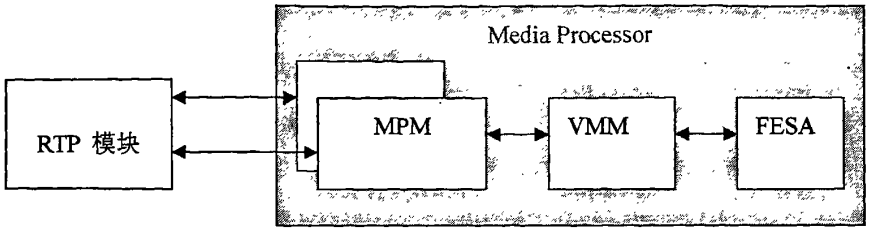


图5.2 MMP构成

媒体处理器与RTP/RTCP模块相连，主要由三个部分组成：

• 一块或多块DSP算法模块，即媒体处理模块（MPM，Media Processing Module),它包括了AJB(Adaptive Jitter Buffer),PLC(Packet Loss),VAD(Voice Active Detector)，CNG(Comfortable Noise Generation)，Codec，DTMF等功能。

• 语音混合模块（Voice Mixed Module）。

• 一个或多个FESA(Front End Specific Module)，包括AGC,AEC,EQ等功能。

其中本文要研究的自适应抖动缓冲算法就位于MPM模块中。下面来看一下这个模块的构成。

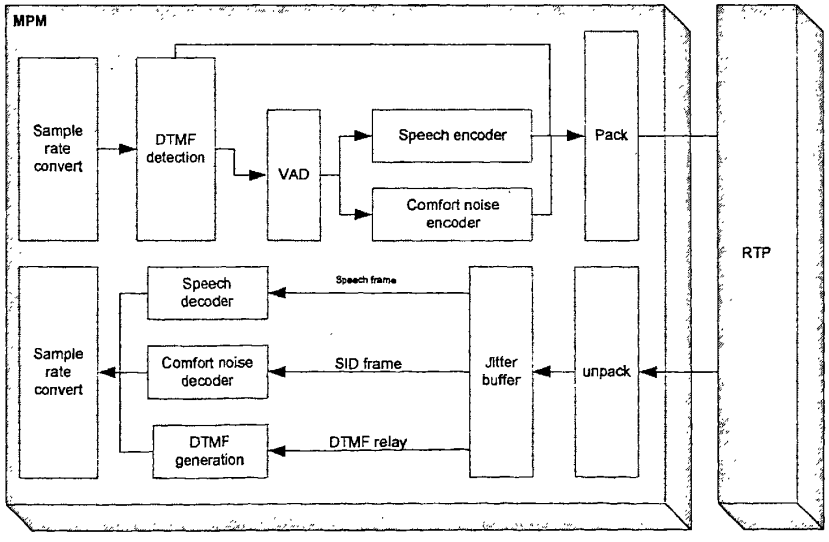


图5.3 MPM模块

MPM模块与RTP紧密相连，但是它们相互之间却是独立的实体。在正方向，来自语音混合模块的16位线性采样值被送入VAD中进行检测。如果没有检测到声音，语音包就会被送到舒适噪声产生器，来采样静音值并产生一个静音插入描述符

（SID, Silence Insertion Descriptor）帧。如果检测到声音，采样值就会被送到语音编码器，产生语音帧。语音，DTMF或非语音帧就会被打包成RTP格式发送到RTP模块。

反方向，RTP包头被去掉，语音帧就会被发送到抖动缓冲（JB，Jitter Buffer）。

抖动缓冲的输出会被识别为三种情况，分别发送到不同模块：

- 如果是语音帧则送到语音解码
- 如果是SID帧并且非G.729和G.723就送到舒适噪声解码器（CND）
- DTMF重现

5.1.2 抖动缓冲框图设计

上一节介绍了MPM的基本构成和工作流程，但是如何对其中的AJB，Codec和PLC模块进行具体合理的设计，使其可以更好的提高质量和性能，是不得不考虑的一个问题。设计时有以下几个选择：

1．三个模块是独立的，数据流方向为：网络－AJB－解码－PLC－扬声器；

2．PLC与语音解码合为一体。数据流方向为：网络－AJB-解码（内嵌PLC）－扬声器；

3．PLC与AJB作为一体。数据流方向为：网络－AJB 预处理－解码－AJB后处理－扬声器；

4．所有的模块合在一起。

方案1在提高质量方面没有任何优点，因为所有的三个模块是独立的。

方案2比较常用，较易理解，但是达不到最好的语音质量，因为AJB和解码器是串行连接的。

方案3：优点：

- PLC不仅可以用到以前的帧信息，也可以用到以后的帧信息。
- PLC和播放缓冲延时调整可以无缝的建在一起，所以有更好的性能和质量。
- PLC和播放调整可以在语音参数域完成（预处理）或者在PCM域（后处理），或者AJB中的PLC可以禁用，那么就可以用编解码内嵌PLC。
- 解码器可以由任何模块提供。因此如果没有足够的时间开发所有的语音编解码器，至少可以提供给别人最好的PLC。

缺点：

- 可能达不到最好的质量；
- 如果PLC在预处理时完成，其余帧的打包和解包就必须完成。

方案4不够灵活，因为语音编解码器是不可代替的。并且与方案3比起来，质量上没有什么区别。综上所述，本文认为方案3具有最好的质量，并且在延时和

丢包间能取得灵活的平衡。图5.4为其框图和外部接口。

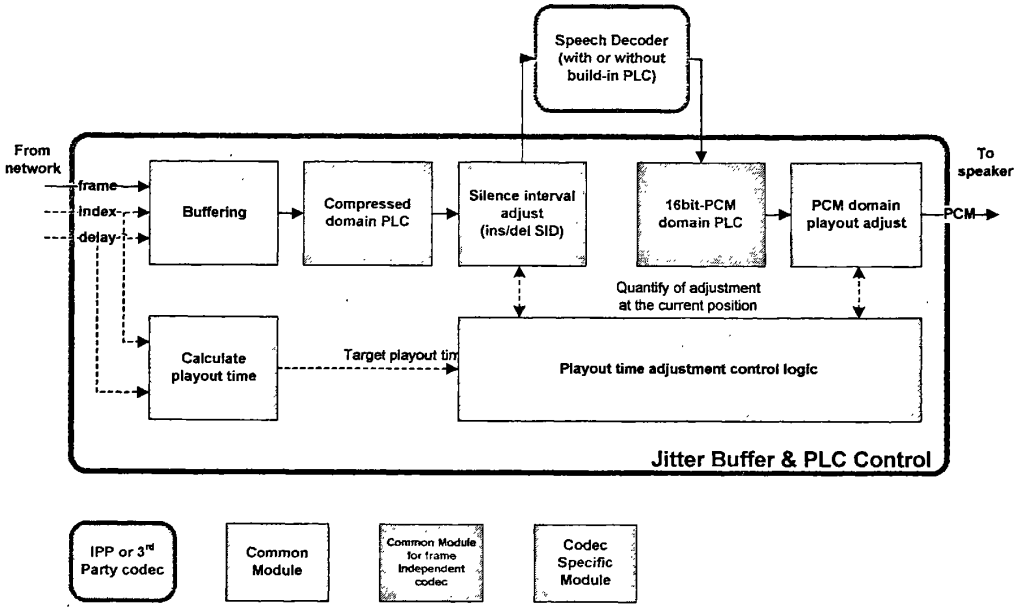


图5.4 Jitter Buffer设计框图

模块的输入包括解包过的数据包到达的时间和时间戳。这些数据都是从网络上转发过来的，且AJB没有反馈数据或控制数据给网络模块。输出是以8khz采样的16位PCM，单声道。

AJB和PLC内部主要有五个模块：

- 缓冲，这个模块对输入的帧数据进行缓冲，处理乱序的帧。
- 计算播放时间，这个模块用来预测网络延时，对总的延时和抖动作出折衷。这一折衷方法是基于动态网络测量的。它不采用编解码器，PLC或播放模块产生的动态信息。
- 播放时间控制逻辑，这个模块获得信息来决定当前点应该插入或删掉多少帧或采样，以此来使失真最小化。处理时应该遵循以下准则：改变播放时间的速率是受限制的。对不同的部分来说，播放时间的延长或者缩减，允许的最大变化速率不同。时间的延长可以在短时间内完成，但是播放时间的缩短需要较长的时间。播放时间在静音期可以很快的变化。但是在有话音时是受限的，要有稳定的幅度。所以话音期间幅度不稳定或者基音周期不稳定时，变化时间应该更短。
- 压缩域插值，
- PCM域插值，它只适合帧无关的编解码器。对于依赖帧的编解码器不适合。

5.1.3 Jitter Buffer 的接口函数

1. JB的数据结构如下：

```
Typedef struct
{
    Uint32 sequence; //RTP包的序列号
    Uint16 payloadtype; //RTP包的载荷类型,由RTP定义，并告诉MPM用哪种编解码算法来解码载荷
    Uint16 payloadlength; //载荷长度
    Uint8 *payload; //实际的载荷
    Uint32 timestamp; //包的时间戳
}jbData_t;
```

2. 定义的变量数据

```
Typedef struct
{
    Int32 avgJitter;//平均Inter Arrival 抖动
    Int32 maxJitter;//最大Inter Arrival 抖动
    Int32 minJitter;//最小Inter Arrival 抖动
    Int32 varJitter; //抖动变化
    Int32 avgJbSize; //平均抖动缓冲大小
    Int32 maxJbSize;//最大抖动缓冲大小
    Int32 minJbSize;//最小抖动缓冲大小
    Int32 lostPcktCnt;//网络丢包数目
    Int32 discardPcktCnt;//缓冲丢弃包数目
    Int32 dropPcktCnt; //网络丢包数目
} msi_jbStatistics_t;
```

3. 将包放入缓冲区的函数接口

```
Int32 msi_jbPutPacket(Uint32 mpmid, jbData_t *data)
```

这个函数实现将语音包放入抖动缓冲中。

4. 将包从缓冲区取出的函数接口

```
Int32 msi_jbGetStat(mpm_t *mpm, msi_jbStatistic_t *jbStat)
```

5. 设置延时周期的降低

```
Int32 msi_jbSetDecDlPeriod (mpm_t *mpm, Int32 decDlPeriod)
```

变量decDlPeriod控制缓冲大小降低的频率，用来帮助从网络拥塞或延时的包中恢复语音数据。它的可调整范围是从0到1000ms。

6. 获取抖动缓冲数据

Int32 msi_jbGetStat (mpm_t *mpm, msi_jbStatistics_t * jbStat)

7. 同样，MPM还包括一些外部API接口来设置和控制抖动缓冲

Int32 msi_jbSetPrime(mpm_t * mpm, Int16 prime)

Int32 msi_jbSetThre(mpm_t * mpm, Int32 incThre, Int32 decThre, Int32 adjMethod, Int32 adjInterval)

Int32 msi_jbFreeze(mpm_t * mpm, Int32 prime)

Int32 msi_jbAdapt(mpm_t * mpm, Int32 incThre, Int32 decThre,Int32 adjMethod, Int32 adjInterval)

Int32 msi_jbSetAdaptMode(mpm_t * mpm, Int32 adjMethod)

Int32 msi_jbGetStat(mpm_t * mpm, msi_jbStatistics_t * jbStat)

5.2 系统测试

5.2.1 测试系统搭建

为了测试语音质量，本文建立了一个测试系统来自动得到不同环境下的数据信息并保存下来。测试所需仪器和设备有：一到两个待测试的终端设备，DSLA语音质量测试仪，主机和SIP服务器。

在众多的语音质量测试设备中，我们选择了DSLA【21】。DSLA是Digital Speech Level Analyser的缩写，是由英国Malden公司生产的用于语音信号强度及质量测试的高精度工具。之所以选择DSLA，是因为它包含了以下基本的功能：

- 能够通过脚本或程序控制，包括TCL, Perl, Python等；
- 能够自动的从待测试设备中接收语音流；
- 可以完成语音质量测试(PESQ，PSQM, PAMS)等，以及延迟、抖动、切割、DTMF分析等。
- 能够测试所有的语音质量数据并且自动的保存记录；
- 能够通过脚本控制自动跳转到下一个测试环境来测试语音质量。

测试系统框架如下所示：

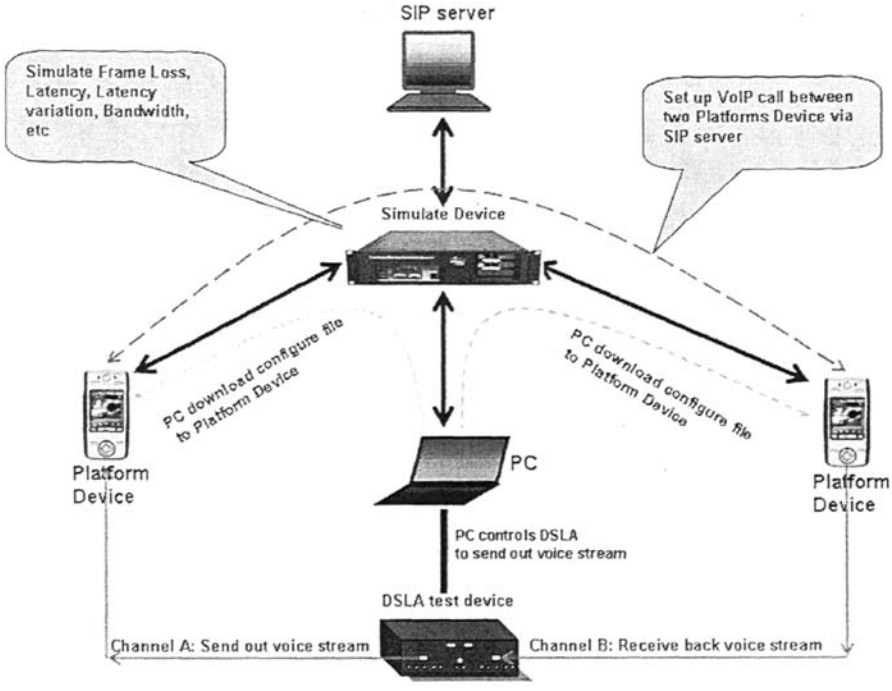


图5.5 系统测试结构图

5.2.2 测试要求与流程

作为一个实时通信的工具，VoIP的性能数据要求为在丢包率为3%的情况下

- 单向延时：小于250ms
- 延时抖动：20ms～30ms
- 电话建立时间：小于5s
- G.711的PESQ值：大于4.0

自动化脚本的测试流程：

1. DVK、DSLA和其它外围设备初始化。比如，DSLA上电和自初始化，DVK上电

2. 启动配置，SIP、TFTP服务器配置，DVK固定IP地址的设置，WiFi程序上电和设置等等。

3. 产生VoIP连接，包括判断连接是否成功，从服务器端下载VoIP程序，远程登陆到DVK来拨通一个VoIP电话，远程登陆到别的DVK来接受一个VoIP电话，挂断电话等等。

4. DSLA的自动测试，包括DSLA通过DVK的输入脚（line in port）发送测试流，DSLA自动接收来自DVK的输出流，DSLA的语音质量测试以及输出测试结果，判断DSLA输出测试结果。

5. 测试结果的自动记录，包括结果的格式，判断DSLA有无错误等等。
6. 测试结束的判断。

下面是测试流程图：

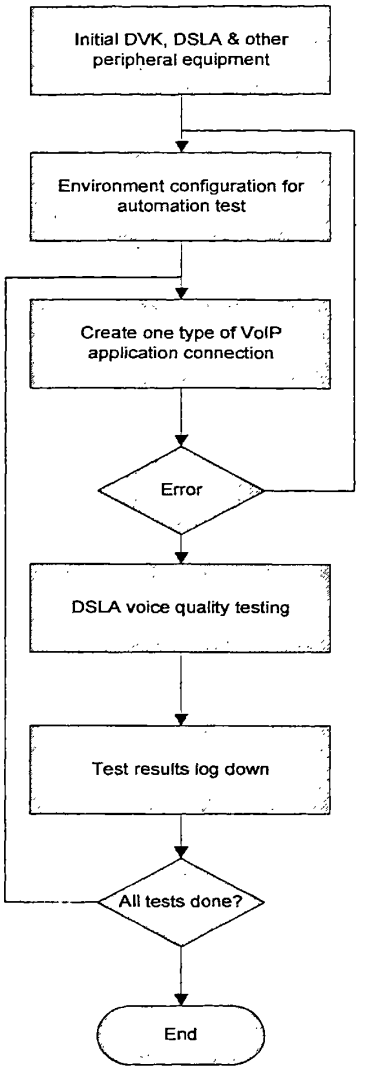


图5.6 测试流程图

5.3 测试结果

为了比较八种算法，我们选择G.723.1在四种网络环境下进行仿真。并且只在每一个话音突起的开始调整播放延时（缓冲大小）。每种环境下我们选择30分钟的参考数据。

对于窗和E-MOS算法，选择最后的10000个包作为滑动窗可能并不适合真正的

缓冲机制，因为这需要很大的存储空间。因此新的算法中选择小的滑动窗口1000个包。

表5-1比较了每种算法的平均播放延时（Avg-P-D）,因为晚到而丢弃包的数量（A-Discard）,丢包率（P-Loss ratio）和平均意见得分（MOS）。前面两种网络是变化比较大的网络，后面两种网络是比较稳定的网络。从表5-1中可以看出，快速指数平均算法在后两种网络中表现较好，但是在前两种网络中会引起很大的时延。因此在网络情况不稳定的情况下不推荐。其他算法类似。各种算法的比较常见的特性如表5-2所示。然而，我们提出的算法在这两种网络下都有不俗的表现。它可以自适应不同的网络环境，在丢包和延时之间取得了较好的平衡。

表5-1 不同算法仿真结果比较

环境	算法	Avg-P-D(ms)	A-Discard	P-Lossratio(%)	MOS
环境1(总的语音包：29695)	Exp-Avg	312.14	1091	4.60	1.94
	F-Exp-Avg	737.65	74	1.17	1.00
	Min-D	265.58	1890	7.29	2.08
	Spike-Det	247.74	2715	10.07	2.05
	Window	368.08	761	3.48	1.68
	E-MOS	294.49	1296	5.29	2.01
	M-MOS	261.12	1396	5.62	2.21
	SAMOSA	220.28	2137	8.12	2.34
环境2(总的语音包：29683)	Exp-Avg	252.21	1029	6.37	2.22
	F-Exp-Avg	637.96	8	2.93	1.00
	Min-D	184.87	2418	11.05	2.42
	Spike-Det	195.42	2935	12.80	2.27
	Window	277.19	402	4.26	2.20
	E-MOS	212.35	1750	8.80	2.35
	M-MOS	212.90	1491	7.93	2.40
	SAMOSA	179.58	2535	11.45	2.44
环境3(总的语音包：29916)	Exp-Avg	21.84	464	5.36	3.01
	F-Exp-Avg	53.32	91	4.11	3.06
	Min-D	20.61	413	5.18	3.02

	Spike-Det	21.37	949	6.98	2.90
	Window	30.84	335	4.92	3.03
	E-MOS	31.88	164	4.35	3.07
	M-MOS	35.93	131	4.24	3.07
	SAMOSA	24.25	289	4.44	3.07
环境4(总的语音包：29933)	Exp-Avg	52.55	489	1.64	3.26
	F-Exp-Avg	60.10	0	0.01	3.41
	Min-D	47.86	1260	4.22	3.06
	Spike-Det	48.11	1204	4.03	3.01
	Window	48.88	292	0.99	3.33
	E-MOS	76.80	0	0.01	3.39
	M-MOS	50.47	107	0.37	3.39
	SAMOSA	51.01	31	0.11	3.41

5-2 不同算法的特点

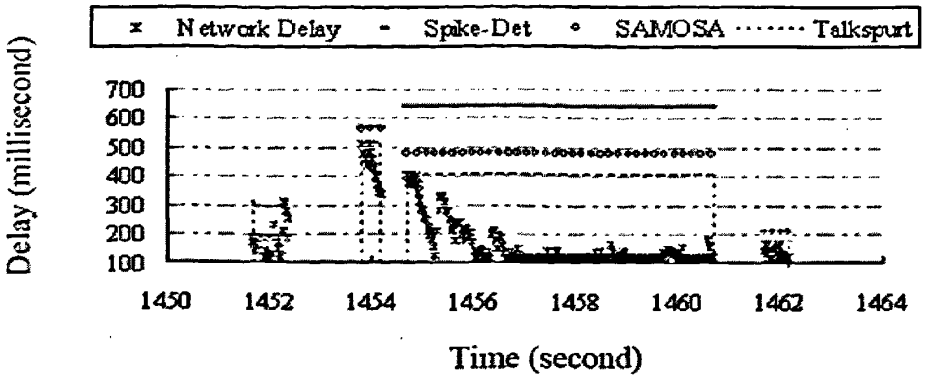
算法	播放延时预测		对可变延时的敏感程度
	整个过程	尖峰期间	
Exp-Avg	估计不足	估计不足	低
F-Exp-Avg	过高估计	过高估计	高
Min-D	估计不足	估计不足	低
Spike-Det	估计不足	过高估计	取决于门限值的设置
Window	过高估计	取决于尖峰的幅度	取决于门限值的设置
E-MOS	过高估计	取决于尖峰幅度	低
M-MOS	相对估计过高	估计不足	高

既然我们提出了新的算法来克服上述的不足，那么通过图5-7和图5-8我们可以看出SAMOSA算法在不同条件下都能够演绎出合适的播放缓冲，因为

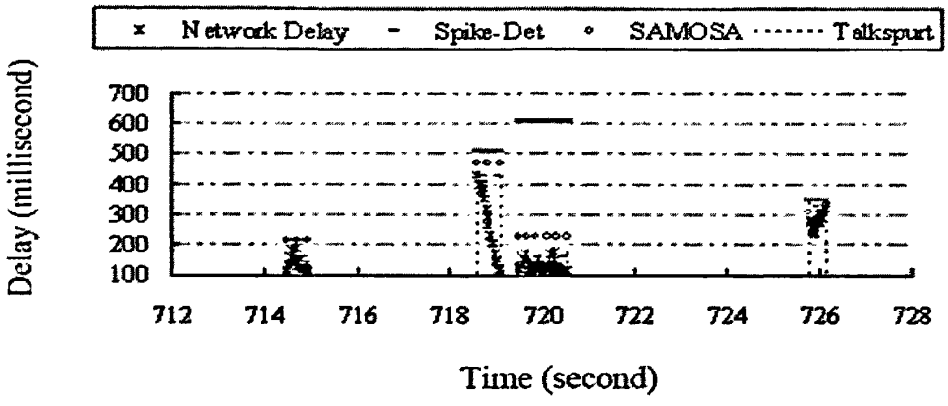
1．在网络延时降低时，它可以以适当的速度降低播放延时（图5-7）

2．它可以预测网络延时的变化趋势(图5-8(a))

3．它可以判断尖峰是否应该调整(图5-8(b))

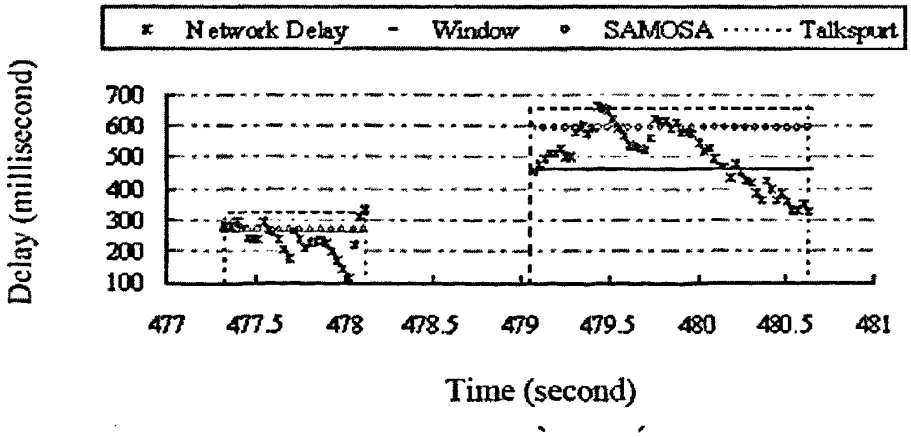


(a)

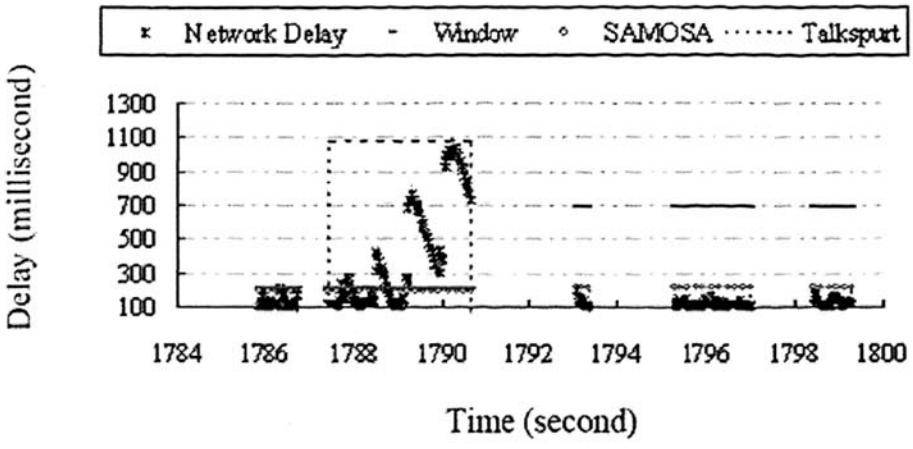


(b)

图5-7 Spike-Det和SAMOSA算法的性能比较



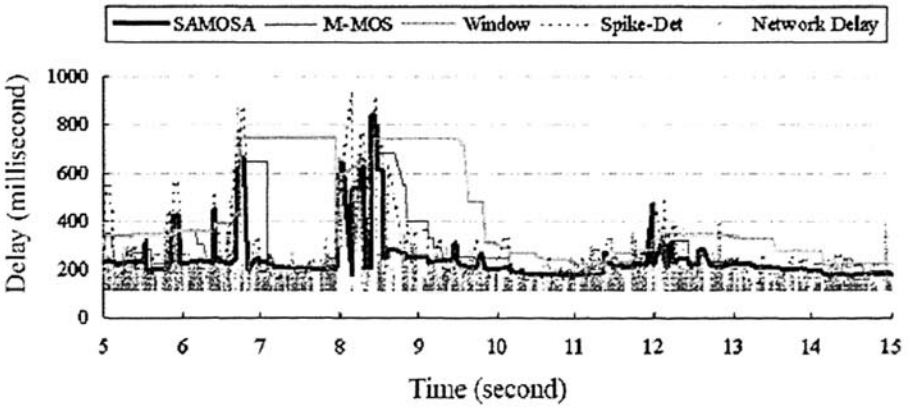
(a)



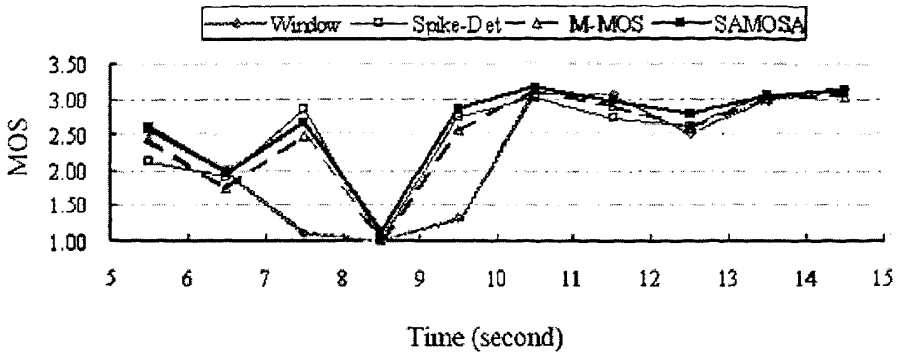
(b)

图5-8 Window和SAMOSA算法的性能比较

我们同样用 10分钟长的时间来研究这四种算法。图5-9说明了在尖峰期间各种算法的性能。可以看出SAMOSA算法能够很好的跟踪网络延时的变化趋势，从而在每个时间间隔都能得到比其他算法更高的MOS值。



（a）播放延时的比较



（b）MOS值的比较

图5-9 四种算法的性能评估

5.4 总结

本章我们介绍了整个系统和框架。并设计了一个行之有效的测试系统来测试我们所研究的几种算法，并对结果进行了分析。从比较的结果可以得出，我们提出的新算法在大多数网络环境下都能够很好的提高感知语音质量。因为它在四种网络环境下，不管是从整个过程来说还是从尖峰延时调整来说都表现不俗。特别是他可以跟踪网络延时变化，在每个话音突起时都能够给出适当的播放延时。因此，我们的缓冲算法更加有效，更加适合实时的缓冲机制。

第六章 总结与展望

6.1 论文工作总结

VoIP业务作为基于IP技术的一种新应用已经得到了广泛的关注和发展，对传统的电话业务提出了强有力的挑战。但是IP电话的质量还不尽如人意。如何建立一个有质量保证的实时通信平台，如何提高VoIP的语音质量成为近几年研究的热点。

本论文首先概述了VoIP的基本概念和影响语音质量的主要因素。这些因素包括时延、抖动以及编码方式。接下来本文分析了相应的解决方案。其中对语音包进行缓冲可以是固定的也可以是自适应的。因此本课题研究了当前主流的自适应抖动缓冲算法，分析了各自的实现机制和性能优劣，接着通过提出一个比较实用而且综合性能优良的抖动缓冲算法来消除时延抖动，这也是本论文的重点工作所在。最后，论文给出了系统设计的基本架构和设计特点，描述了测试模型和流程。在程序中实现了8种抖动缓冲播放算法。实验表明使用了该自适应抖动缓冲算法后，语音质量有明显提升，具有更好的延时、丢包特性。因为其计算复杂度小，可控性良好、使用独立性较强等特点，此算法在VoIP终端设计中具有很大的实用性。

综上所述，本论文的主要成果可以总结如下：

- 介绍了感知语音质量评估的方法论，并将PESQ和E-Model相结合，提出了一种新的基于客观测试的评估方法。本课题用G.723.1编码器来实现了对这种新的方法的评估。结果可以看出，新的方法简便易行，并且评估准确有效。
- 分析了7种已有的缓冲算法，针对它们存在的问题，基于感知语音质量和丢包延时特性，提出了一种新的自适应缓冲算法。
- 用C语言实现了上述8种算法，并对8种算法进行了方针和比较。结果显示，和已有的算法相比，在大多数网络环境中，新的算法更能增强感知语音质量，并且更加有效。

6.2 未来工作展望

VoIP的QOS是一个复杂的问题，本论文中还有很多未涉及或有待改进的地方。

- 虽然我们在四种网络环境下进行了几种算法的比较和研究，这四种环境也能够代表大多数的网络环境。但是IP网络的情况远比这个复杂，

以后的研究中需要考虑更多的网络环境和网络性能。另外，丢包的一些方面，如丢包突发；以及网络延时分布，如泊松分布也值得进一步研究。

- 对感知语音质量评估来说，主观测试需要和客观测试方法结合起来，以得到更加精确结果。此外，丢包的位置也会影响到语音质量，因此这两者之间的关系也是本课题以后要研究的一个方面。
- 从编解码的角度来说，不同的编辑码器可能会导致不同的包长度和语音治理。因此，它与延时丢包、丢包特性紧密相关。而本文研究的算法都是基于G.723.1。因此需要进一步研究更多的编解码器来证明和提高提新提出的方法。
- 本文中所研究的算法都是在话音突起的开始调整播放延时。因此有效性收到话音突起长度的限制。
- 研究其它影响语音质量的关键技术，来进一步提高IP电话的通话质量，比如说丢包恢复问题、回声抵消、静音检测问题等等。

参考文献

1. International Telecommunication Union, *ITU-T Recommendation G.114. Technical report*, 1993.
2. Jayant, N.S. (1980) *Effects of packet loss on waveform coded speech.* In: Proc. Fifth Int. Conference on Computer Communications, Altanta, Ga., pp. 275-280.
3. Ramjee, R., Kurose, J., Towsley, D. &Schulzrinne, H. (1994) *Adaptive Playout Mechanism for Packetised Audio Application in Wide-area Networks.* IEEE, Toronto, Canada.
4. Moon, S.B., Kurose, J. &Towsley, D. (1998) *Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms.* Acm/Spring Multimedia Systems, vol.5, pp.17-28.
5. Mills, D. (1983) *Internet Delay Experiments.* ARPANET Working Group Requests for Comment, RFC 889.
6. Fujimoto, K., Ata, S. & Murata, M. (2002) *Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Application.* IEEE Globecom, 2002.
7. VoIP中音频编解码选择与QoS研究,邹军，Feb 2006.
8. 蒋欢，VoIP语音质量关键技术的研究，Mar 2006.
9. 周康，VoIP移动终端关键技术研究及实现，Jan 2007.
10. ITU-T Recommendation G.107(2000). The E-model, A Computational Model for *Use in Transmission Planning.*
11. ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ), *An Objective Method for End-to-end Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs.*
12. 钱俊,VoIP系统中静音检测的设计和实现，Apr 2004.
13. Sun, L.F. & Ifeachor E.C. (2003) *New Methods for Voice Quality Evaluation for IP Networks.* ITC 18. Berlin, Germany, 2003, pp.1201-1210.
14. Cole, R.G. & Rosenbluth, J.H. (2001) *Voice over IP Performance*

Monitoring. Journal on Computer Communications Review, vol. 31, no. 2.

15. International Telecommunication Union, *Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*, ITU-T Recommendation G.723.1, Mar 1996. H.323 International Engineering Consortium http://www.iec.org/online/tutorials/h323/topic01.html
16. Davidson, Jonathan and Peters, James, *Voice over IP Fundamentals*, Cisco Pree.
17. International Telecommunication Union, *Pulse Code Modulation (PCM) of Voice frequencies*, ITU-T Recommendation G.711, Nov 1998.
18. Postel, Jon, *Transmission Control Protocol specification*, ARPANET Working Group Request for Comment, RFC793, September 1981.
19. Mendenhall, William and Sincich, Terry, *Statistics for Engineering and the Sciences*, Prentice Hall, 4th edition.
20. 蒋金弟，IP网络语音技术及其应用研究，浙江工业大学研究生学位论文.
21. 北京瑞森新谱科技有限公司，DSLA语音质量测试仪器.
22. Rosenboerg, J., Qiu, L. & Schulzrine, H. (2000) *Internet Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet*. Proc. of IEEE Infocom 2000.
23. International Telecommunication Union, *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Liner-Prediction (CS-ACELP)*, ITU-T Recommendation G.729.
24. International Telecommunication Union, *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)*, ITU-T Recommendation G.726, Dec 1990.

攻读学位期间发表的学术论文目录

[1] 李敏，刘锦高 RFID系统中低功耗标签的研究与设计 现代电子技术 2007.11

致谢

时光匆匆，3年的研究生生活即将过去。老师们的谆谆教导和同学们的无私帮助都历历在目。我想这段时光将是我一生中最美好的回忆。每当想起这一切，我心里都油然生出无限的感激之情。

首先我要感谢我的恩师刘锦高教授。刘老师渊博的学识和脚踏实地的工作作风深深感染了我。我从刘老师那里学到的不仅仅是专业知识，更重要的是严谨的治学态度和对事业孜孜不倦的追求。刘老师在学习和生活上都给予了我悉心的指导和深切的关怀，教给我很多做人处事的道理。正是刘老师的教诲和关心激励我完成了本论文的研究工作，使我顺利结束了研究生阶段的学业。在此临别之际，我再次表示感谢，谢谢您对我的栽培和关心。

同时我要感谢翁墨颖教授。是他在科研工作中给我指导和点拨，不断给我研究的机会，让我的动手能力得到了很大的锻炼。他对我严格的要求和悉心指点一直鞭策着我前进。他那永远进取，永远创新的精神时时激励着我，让我终生受益。在此我衷心的感谢翁老师在工作、学习和生活上对我的关心和鼓励。

感谢1201寝室的所有同学。正是因为生活在这样一个温暖的家庭中，才使我的生活充满了乐趣；也正是因为大家的相互鼓励和学习，才使得我有动力去完成各项任务。很庆幸自己生活在这样一个寝室。

感谢实验室所有的师兄师姐同学们对我的关心和鼓励，感谢电子05级研所有同学在生活学习中给我的友情和美好的回忆。

感谢我的家人，感谢他们一直以来对我的无私关爱，他们是我不断前进的动力。

感谢各位审稿老师在百忙之中抽出宝贵的时间审阅论文，感谢你们提出的宝贵意见。