

1914

1915

1916

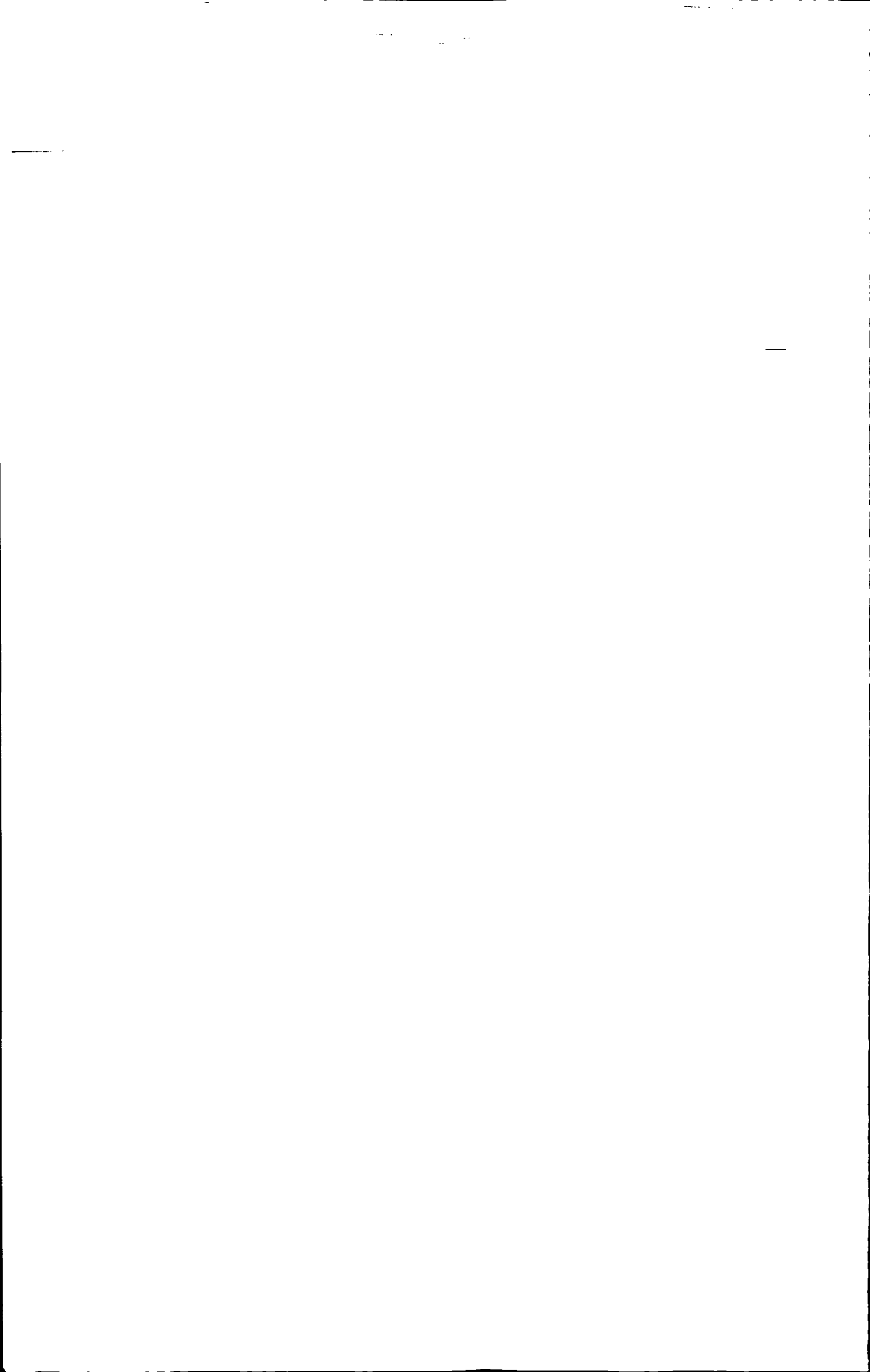
## 摘要

专网专用，价格居高不下，维护费高一直都是制约集群通信在国内大力发展的重要因素。本文以价格和性能的综合考虑为出发点，设计并实现了一种基于 IP 网络的多媒体集群通信系统。

根据功能、网络规模等需求，本文对该系统的结构、工作流程、控制协议进行了详细的分析、设计。该系统共分为会话服务器、应用层组播平台 NaradaBrokering、PC 机客户端以及 Android 手机客户端四部分。在对应用层组播平台 NaradaBrokering 进行分析、测试的基础上，本文分别对会话服务器和 PC 机客户端进行了具体的设计与实现，并与手机客户端进行联调。

该系统无需铺设专门网络、购买专用终端设备，因而成本大大降低。同时，在设计实现过程中采用了分模块、分层次的方法，使得系统后续功能的扩展和完善以及系统的维护相对简单方便。经过测试，此系统的功能和性能都可以满足用户要求，同时也达到了价格和性能的综合。

**关键字：**集群通信 多媒体 应用层组播 模块化



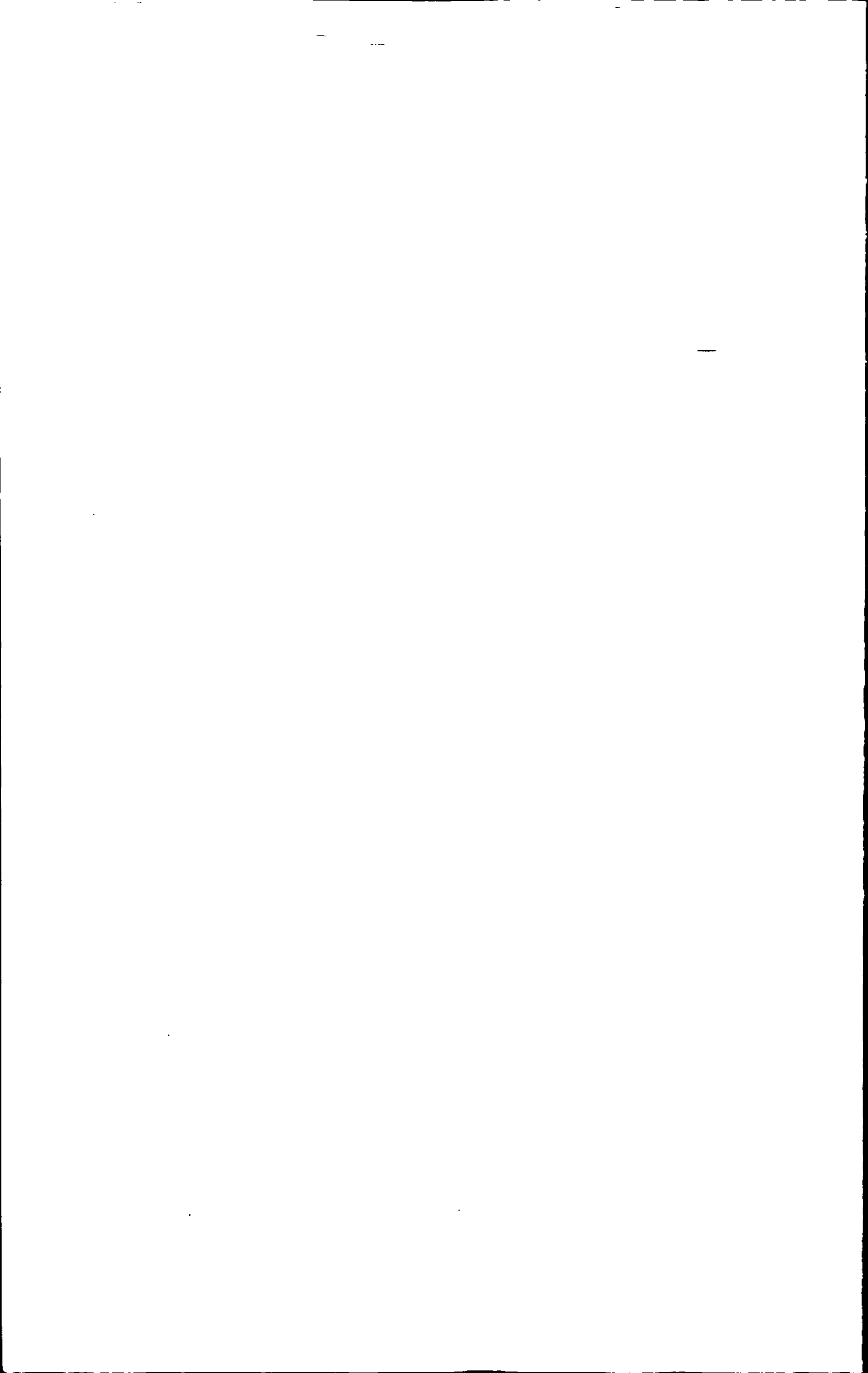
## **Abstract**

Dedicated private network, high price and high maintenance cost are always the restrictive factors for the trunked communication development in the domestic. Considering both the price and performance, this paper designs and implements an IP-based multimedia trunked communication system.

According to the function, performance and other requirements, the detailed analysis and design are carried out for the structure, workflow and control protocol of this system. The system is divided into four parts: the session server, the application layer multicast platform NaradaBrokering, the PC client and the Android mobile phone client. On the basis of analyzing and testing the application layer multicast platform NaradaBrokering, this paper designs and implements the session server and the PC client respectively, and debugs with the mobile phone client.

This system has no need of laying special network and purchasing dedicated terminal equipments, which reduces the cost greatly. Meanwhile, the usage of modularity and layering in the design process makes it relatively simple and convenient for the system to expand, improve and maintain its functions. Tested, the system can meet users' requirements in terms of function and performance, and also achieves the balance between price and performance.

**Keywords:** Trunked communication   Multimedia   Application layer multicast  
Modularization



# 目录

第一章 绪论.....	1
1.1 集群通信系统介绍.....	1
1.2 集群通信系统的发展现状.....	1
1.3 本文研究的意义及内容安排.....	4
第二章 基于 IP 网络的多媒体集群通信系统的总体设计.....	5
2.1 基于 IP 网络的多媒体集群通信系统的需求及特性分析.....	5
2.1.1 用户需求分析.....	5
2.1.2 功能需求分析.....	6
2.1.3 系统特性分析.....	7
2.2 基于 IP 网络通讯模式的系统结构的设计.....	9
2.2.1 基于 IP 单播模式的系统结构设计.....	9
2.2.2 基于 IP 组播模式的系统结构设计.....	11
2.2.3 基于应用层组播模式的系统结构设计.....	13
2.2.4 系统结构设计方案的选择.....	14
2.3 应用层组播平台的分析与选择.....	15
2.3.1 系统对应用层组播平台的需求分析.....	15
2.3.2 NaradaBrokering 平台的选择与分析.....	16
2.4 基于 NaradaBrokering 平台的多媒体集群通信系统的整体设计.....	20
2.4.1 系统整体结构的设计.....	20
2.4.2 系统工作流程的设计.....	22
2.4.3 系统控制协议的设计.....	23
2.5 本章小结.....	26
第三章 会话服务器和 PC 机客户端的设计与实现.....	27
3.1 设计原则.....	27
3.2 开发平台简介.....	28
3.3 会话服务器的设计与实现.....	29
3.3.1 会话服务器的模块化设计.....	29
3.3.2 服务模块的设计与实现.....	30
3.3.3 服务管理模块的设计与实现.....	35
3.3.4 消息管理模块的设计与实现.....	37
3.3.5 NB 系统发布/订阅模块的设计与实现.....	38
3.4 PC 机客户端的设计与实现.....	39

3.4.1 PC 机客户端的模块化设计.....	39
3.4.2 用户界面模块的设计与实现.....	40
3.4.3 信令处理模块的设计与实现.....	42
3.4.4 多媒体处理模块的设计与实现.....	43
3.4.5 NB 系统发布/订阅模块的设计与实现.....	45
3.5 本章小结.....	45
第四章 系统的测试.....	47
4.1 应用层组播平台的搭建.....	47
4.2 系统与手机客户端的联调.....	47
4.3 系统测试与分析.....	50
4.3.1 测试环境.....	50
4.3.2 功能测试.....	51
4.3.3 性能测试.....	52
4.4 本章小结.....	55
第五章 总结与展望.....	57
致谢.....	59
参考文献.....	61

## 第一章 绪论

本章概述了集群通信系统的概念以及发展历程,介绍了数字集群通信系统在国内外,尤其是在国内的发展现状,并给出了本文的研究意义与章节安排。

### 1.1 集群通信系统介绍

所谓集群通信系统,即系统所具有的可用信道可为系统的全体用户共用,具有自动选择信道功能。它是共享资源、分担费用、共用信道设备及服务的多用途、高效能的无线调度通信系统。

集群通信系统诞生于上个世纪 70 年代末、80 年代初,最早的实现是基于模拟系统。模拟的集群系统相互干扰强,功能有限,而且使用方需要单独建设一个集群调度专网系统,成本高、维护费用高、频率利用率低。90 年代后期,随着数字集群通信技术的日益成熟,集群通信由模拟走向数字成为大势所趋。数字集群通信技术改变了模拟集群系统功能单一、技术陈旧、效率低下等弊端,它具有全新的技术体制、灵活的通信构架和强大的服务功能,能提供话音、数据、图像等多种通信服务,因而渐渐成为市场的主流<sup>[1]</sup>。

同时,集群通信的组网方式也在由传统的专业网络向共网转变。专业集群通信网是指由专门部门组织建立的网络,部门内部的用户利用该网络建立相互呼叫,例如公安专用网等。它要求具有更严格的接入能力、可靠性、服务等级和安全性,但在紧急情况下多个部门不能协同工作,且会造成资源紧张,甚至由于阻塞造成系统崩溃。商业集群网(共网)是由一个机构建立并运营网络,其目的是为公共的或第三方组织的注册成员提供通信服务。它是由多个部门共用一个网络,支持的用户数多,覆盖范围大,但性能往往不如专网那么高<sup>[2]</sup>。

因此,集群通信系统的发展有两个明显趋势:由模拟向数字发展,由专网向共网发展。同时,随着 IP 技术在全世界范围内的广泛应用以及迅猛发展,数字集群通信系统向 IP 演进也成为一种必然趋势。

### 1.2 集群通信系统的发展现状

在国外,无线集群通信是一个可以和公共无线通信相抗衡的市场,尤其在美国等发达国家,集群通信的市场规模已经占到公网移动通信的 10%<sup>[3]</sup>。

我国于 1989 年引进模拟集群系统,1993 年得到较快发展,1995 年以后发展滞缓,1998 年开始发展数字集群通信。



目前,国内数字集群通信系统标准主要有欧盟开发的陆地集群无线通信 TETRA (Terrestrial Trunked Radio)、美国开发的集成数字增强型网络 iDEN (Integrated Digital Enhanced Networks) 以及由中兴推动发展的开放式集群结构 GoTa (Global open Trunking Architecture) 和 华为开发的 GT800。下面对这几种标准及其在国内的发展现状进行介绍:

### 1. iDEN:

iDEN 系统于 1994 年在美国洛杉矶问世,是由摩托罗拉公司提出的一种数字集群制式,工作于 800MHz 频段范围。它的主要特点是可以和全球移动通讯系统 GSM (Global System for Mobile Communications) 兼容,适合组成比较大的集群通信网。iDEN 系统的最初设计是做集群共网应用,因此除了以指挥调度业务为主外,还兼有双工电话互联、数据和短消息等功能。

目前 iDEN 在我国部署的有三个大系统,即福建集群电话公司从福州到厦门一线的网络、深圳市交通局下属的联通公司的网络和在上海的联通国脉公司的网络。

在共网 PTT (Push To Talk) 应用方面,iDEN 网络可以认为是迄今为止最成功的案例。然而,由于数字集群系统的专业性特点,国外通信巨头通过控制核心技术并设置专利等知识产权保护壁垒,导致现有国外数字集群系统的网络设备和终端的价格始终居高不下,不利于扩大数字集群系统应用的市场规模。

### 2. TETRA:

TETRA 是欧洲电信标准组织 ETSI (European Telecommunications Standards Institute) 制定的数字集群通信系统标准,基于传统大区制调度通信系统的数字化专用移动通信无线电标准,采用时分多址 TDMA (Time Division Multiple Access) 方式。TETRA 系统在指挥调度方面业务全面,可完成语音、电路数据、短数据消息、分组数据业务的通信及以上业务的直通模式,并可支持多种附加补充业务。

2003 年北京正通公司所建的北京政务移动通信网采用的是 TETRA 标准。同时,在轻轨、地铁等轨道交通系统以及一些城市应急联动系统的移动平台中 TETRA 也占有一定市场。

目前,TETRA 网络对外互联部分技术未完全公开,不同厂家的 TETRA 产品还没有实现完全的互联互通,不同网络之间的用户和业务的互通还存在问题。另外,TETRA 的系统设备采购和建网成本高,终端价格高,这些因素限制了 TETRA 难以在大范围推广应用。

### 3. GoTa:

GoTa 是中兴通讯独立推出的基于码分多址 CDMA (Code Division Multiple Access) 技术,面向未来技术演进的“新一代”数字集群通信系统。GoTa 数字集群通信系统,可以提供共网集群和专业调度统一的业务模式,提高网络综合竞争

能力,吸引更多专业和社会集群用户入网,创造更多运营收入。GoTa 作为专业集群通信系统,可供共网集群网络运营商建网运营,提供集群通信服务,也可应用于专业集群调度系统,例如地方的应急联动指挥调度系统。

与 TDMA 方式的数字集群系统不同,GoTa 数字集群通信系统对现有的 CDMA 技术进行优化和改进,使其适应了集群业务的需要,具有较大的技术和业务发展空间。GoTa 集群系统的无线侧基于成熟的 CDMA 空中接口技术,不改变空中接口的物理机制,一键通 PTT (PushToTalk) 呼叫利用分组数据方式解决了在 CDMA 上实现集群的关键技术,允许大量的 PTT 用户共享相对较少的通信资源,并使呼叫连接速度提高。GoTa 的实现方案简捷,投资少,部署方便。

#### 4. GT800:

GT800 是由华为公司提出的数字集群新技术。它采用时分多址技术,具有突出的覆盖优势,可为专业用户提供有保障的集群通信业务。GT800 覆盖能力强,通过技术增强,单基站覆盖半径可超过 70 公里,覆盖范围与小区内的用户容量、移动台与基站距离远近无关,不受“远近效应”等因素的影响,稳定可靠,可很好地满足集群应用的需求。

GT800 在开发设计时采用主流移动通信技术为基础,标准开放,具有雄厚产业支撑,在规范制定时成立产业联盟,形成多个厂家共同参与、共同发展的局面,保证产业链完整与成熟。在业务方面,GT800 在设计时参考了 TETRA 规范中的业务需求,并在此基础上有所扩充和发展,比如提供组内优先级、岗位号码和基于位置的路由等功能,更好满足指挥调度需要<sup>[4]</sup>。

综上所述,TETRA 和 iDEN 这两种标准在国内数字集群市场上的应用都不算是很成功,其主要原因首先是价位太高,当然,还有高速多媒体业务运作及前向发展潜力差,TETRA 多厂商运行环境时包括终端在内的互联互通能力差等问题。同时,它们在安全保密、特殊用户满意度及可信任度上也有一定问题。但是 iDEN 和 TETRA 都是经过长期研究开发和应用,已经比较成熟的系统,尤其这两个系统在国外已经广泛使用,得到了很大程度的认可,因此在国内还有一定的发展潜力。而 GoTa 和 GT800 标准是我国自主研发的数字集群新技术,虽然同 iDEN 和 TETRA 两系统相比,它们在组网规模、呼叫延时、技术演进、终端质量和产业链上还有一定差距,但是功能基本成熟并且与新技术有很好的兼容性,更重要的是,它们的研发成功打破了国外公司在国内数字集群领域的垄断,不仅能够带动集群领域的一系列创新,也可带动我国数字集群技术推向国外,提升我国移动通信技术在国际上的地位。

### 1.3 本文研究的意义及内容安排

集群通系统从 1989 年引入国内市场,发展一直并不理想。虽然从行业需求及用户数量角度考虑,中国都是一个拥有极大潜力的消费市场,但是专网专用,价格居高不下,维护费高等这一系列的因素都成为集群通信发展不可回避的障碍。特别是在一些有调度通信需求,但是对系统保密性、安全性等性能要求都不是太高的服务场所,如酒店、宾馆、小区、咖啡厅以及其他娱乐场所等,性价比才是他们最为关注的因素。如何能够为这些用户提供廉价、便利的集群通信系统,是本文的主要出发点。

为了降低集群通信成本,IP 网络成为最佳的选择。而基于 IP 的流媒体技术则为集群通信的实时性要求提供了有力的保障。在接入网的选择上,考虑到上述场所通常已经具备无线局域网热点条件,因此具备此项网络功能的智能手机终端(作为集群通信系统的客户端)可以通过无线相容性认证 WIFI(Wireless Fidelity)技术接入整个集群系统。这样,一个基于 IP 的集群通信系统的实现便成为可能。这样的集群通信系统,不仅可以进行语音,还可以传输发言者的实时视频,而由于采用半双工模式,既不会对整个网络带宽造成巨大影响,又能够充分考虑手机端资源有限的现实情况。

在数据分发平台的选择上,综合考虑了 IP 单播、IP 组播和应用层组播的优劣,最终选择了经过众多项目检验的开源应用层组播平台项目 NardaBrokering(简称 NB 系统),在对其进行细致研究的基础上,开发了一套由会话服务器、客户端(包括手机和 PC 机两种)和 Naradabrokering 平台组成的基于 IP 网络的多媒体集群通信系统,并进行了基本功能的测试,取得了较为理想的结果

本文共分为五章:

第一章介绍了论文的研究背景和意义,回顾了集群通信的发展现状,明确了论文的研究方向和章节安排。

第二章是本论文的重点,根据 IP 网络的三种通讯模式:IP 单播、IP 组播、应用层组播分别设计了三种系统结构,并通过比较、分析选择了第三种结构设计方案。同时,基于选择的应用层组播平台 NaradaBrokering 的特点对系统各部分的功能、系统工作流程及系统控制协议进行了详细设计。

第三章也是本论文的重点,对系统中的会话服务器和 PC 机客户端进行了分模块的设计与实现。通过各模块之间的协作,实现系统功能的有效运行。

第四章通过应用层组播平台搭建及与手机客户端的联调,实现了系统功能的完整性,并对实现结果进行系统测试,分析测试结果,总结系统性能,

第五章总结了本文的研发成果以及存在的不足,明确后续研发方向。

## 第二章 基于 IP 网络的多媒体集群通信系统的总体设计

本章首先分析了基于 IP 网络的多媒体集群通信系统的需求及特性, 根据 IP 网络的通讯模式 IP 单播、IP 组播、应用层组播分别设计了三种系统结构, 并通过对比、分析选择了以应用层组播技术为基础的结构设计, 最后基于这种结构对系统各部分的功能, 系统的工作流程以及系统的控制协议进行了详细设计。

### 2.1 基于 IP 网络的多媒体集群通信系统的需求及特性分析

#### 2.1.1 用户需求分析

集群通信系统的主要目标是为行业用户提供方便、快速、性能优越、覆盖范围广泛的指挥调度和协作功能。从这一点上讲, 国外的 TETREA、IDEN 以及国内的 GoTa、GT800 这四种数字集群通信系统都能很好地满足要求。但是, 这样的系统通常包括无线子系统、调度子系统、互联子系统三大部分, 每一部分又包括各自相应的组成设备, 因此网络构建复杂、维护费用高。而对于一些有协作、调度需求, 但是投入成本受限的行业, 如餐饮、娱乐等服务性行业, 价格却是需要考虑的首要因素。

随着 IP 网络技术的发展, 集群通信向 IP 网络演进已经成为一种必然趋势。在 IP 网络基础上构建的新一代集群通信系统, 集成了 IP 网络的优势, 可以达到性能与价格的统一, 扩大用户范围。

基于 IP 网络的集群通信系统的优势体现在:

##### 1. 无需铺设专门网络

基于 IP 网络构建的集群通系统中的移动用户通过现有无线局域网接入网络, 固定用户通过现有有线局域网接入, 所有的通信数据承载在 IP 网络之上进行实时传输, 因而不需另外铺设数据传输网络。

##### 2. 无需购买专用终端设备

系统将集群通信功能作为软件集成到手机端, 用户通过手机的无线接入功能接入网络, 因而无需另外购买终端设备。

##### 3. 无需基站以及交换机等设备

系统通过软交换进行所需调度, 会话服务器以软件的形式承载在计算机上, 无需铺设基站、交换机等大型设备, 仅需要搭建服务环境的计算机即可。

由于 IP 网络的时延问题以及数据包容易被捕获等特性, 未经数据加密处理的

该系统在整体性能, 安全性方面都不及专网专用的传统数字集群通信系统, 但是却做到了性能和价格的统一, 可以满足对安全性要求不高, 协作性要求较高, 并注重价格因素的服务行业的用户。而且, 随着互联网技术的发展和普及, 许多的商业场所都已经铺设了自己的无线局域网络, 拥有多个无线接入热点, 所以此系统仍然拥有庞大的潜在用户群体。

对基于 IP 网络的集群通信系统拥有需求的服务行业包括: 酒店、宾馆、娱乐场所、商务写字楼、社区服务等。他们对集群通信的需求主要分为:

#### 1. 调度需求

在酒店、宾馆等服务行业, 通常需要管理人员根据各个楼层, 各个地点的情况实时调配服务人员, 以满足不同用户的不同需求。

#### 2. 协作需求

服务人员除了服从管理者的调配, 还应该根据实时情况反馈自己所负责区域的信息, 进行分工协作;

#### 3. 会议需求

在酒店、写字楼等拥有多间会议室的场所, 参加会议的人员可以利用笔记本通过无线局域网接入这种基于 IP 网络的集群通信系统, 将会议发言者的实时画面传达给不同会议室, 甚至不同区域内的与会人员, 进行实时会议;

#### 4. 多媒体需求

在支持语音的同时, 如果能够进行视频的传输将使这些服务场所的调度、协作、会议进行得更加顺利、方便。

基于 IP 网络构建的集群通信系统通过软交换满足了用户进行调度、协作以及会议的需求, 而且由于 IP 多媒体技术的发展, 它也能很好地支持视频的传输。接收者可以看到发言人手机或者 PC 机摄像头所捕获的实时视频, 这样就能更直观地反映现场情况, 优化协作过程。

### 2.1.2 功能需求分析

基于 IP 网络的多媒体集群通信系统首先要具有集群通信系统的基本功能, 包括:

#### 1. 一对一呼叫功能

主叫用户与被叫用户建立一对一的通信通道, 采用半双工方式, 同一时刻只能由一位用户作为发言人, 另一位用户作为收听者。

#### 2. 分级管理功能

集群系统把用户分成若干个群组, 每个群组具有不同的优先级别。在每个群组内还可以为用户设置不同的优先级, 在通话过程中, 本组高优先等级的组成员

可以直接抢占低优先级组成员的话权。

### 3. 群组呼叫功能

用户可以对同一群组内的其他用户进行一对多的呼叫，建立一对多的通信，而普通的电话或者是手机则只能提供一对一的呼叫通话。

在这三种基本功能中，群组呼叫功能是集群通信最具特色的业务，它具体分为表 2.1 中几种方式：

呼叫方式	特点
组呼	一对多的对讲呼叫，通过专用按键申请发言权讲话，同时只能有一个成员讲话。
紧急呼叫	最高呼叫优先级，在最短时间内、以最便捷的方式将信息通报给预先指定的用户或者群组
广播呼叫	一种特殊的组呼，与组呼的不同点在于只有主叫可以讲话，而被叫方没有发言权

表 2.1 集群通信群组呼叫功能表

组呼和紧急呼叫功能都是以半双工的形式进行通信，除发起呼叫用户可以发言外，其他用户也可以在申请到发言权的情况下进行发言，而广播呼叫只有发起呼叫的用户可以发言。这几种功能分别适合不同的场合，例如广播呼叫比较适合调度应用，而组呼则比较适合协作以及会议应用，紧急呼叫则适用于有特殊要求的场合。

基于 IP 网络的集群通信系统除具备上述基本功能外，还可以在下面几个方面扩展功能：

#### 1. 图片功能

发言用户通过发送图片给所有收听用户，丰富自己的讲话内容。

#### 2. 视频功能

发言者从申请到发言权开始，实时发送通过手机或者 PC 机摄像头捕获的视频数据给所有接收用户，反映实时信息。虽然视频数据量比较大，但是由于采用半双工方式，只有发言人发送视频信息，因此不会过多占用系统带宽。

#### 3. 语音信箱功能

对于比较重要的发言，可以通过语音信箱存储起来，没有收听到或者需要再次收听的用户可以发起重播请求，重新收听。

### 2.1.3 系统特性分析

在传统的电路方式的数字集群系统中，底层的网络传输和上层的控制是融合在一起的。而基于 IP 的新一代数字集群通信系统的传输与控制分离，需要采用分

层的方法来进行网络结构的设计。传输层使用 IP 网络的传输协议,如传输控制协议 TCP (Transmission Control Protocol) 等进行数据传输,而控制层则负责各种指令的处理。这样的结构层次清晰,易于补充和扩展。

基于 IP 网络的数字集群通信系统具有以下特性:

#### 1. 网络拓扑多样,组网灵活

基于 IP 网络的架构设计较为灵活,可支持星型、环型、树型和混合型等各种网络。

#### 2. 互联互通性较好

由于采用了 IP 网络,数字集群网与互联网和 3G (3rd-generation) 等下一代网络的互联互通更加快捷容易。

#### 3. 标准互联,便于扩容和升级

网络采用路由器、交换机、网桥等标准计算机网络设备,同时各种网络的连接也是标准连接,如局域网、广域网、虚拟专用网等,因而方便未来的系统扩容和升级。

#### 4. 具有统一的开发平台

管理维护层通过采用标准的网络应用协议,便于各种扩展应用的实现,例如调度台软件、语音记录软件等,而且通过 IP 网络的软件升级维护更加简单方便。

#### 5. 建设网络所需的成本降低

网络的底层传输和交换系统中采用 IP 架构,与传统的电路交换相比,可以获得较低的购买成本和较低的维护费用。

当然,由于采用 IP 平台,构建数字集群通信系统必须考虑到如下三个方面的关键性问题:

#### 1. 接续延迟

如果网络中用户较多、使用频率增加或网络信号较差,呼叫接续时间会变长,而传统数字集群的接续时间都被控制在 1 秒或者更短的时间之内,这也是其核心的竞争力。在构建基于 IP 网络的数字集群系统时,必须考虑到 IP 化可能带来的较大的接续延时,因而有必要采取一定的技术和策略避免这种时延的发生。

#### 2. 带宽利用率

采用 IP 技术后,通过 IP 数据包来传输信息。传统的 IP 分组头至少是 40 字节,相对于分组净荷来说这是一个很大的数字。利用传统的 IP 分组传输数据势必会浪费有限的无线资源,造成系统的带宽利用率不高。因此在构建下一代数字集群系统时必须充分考虑到这一点,采取一定的解决策略。

#### 3. 服务质量 QoS (Quality of Service) 保证

QoS 一直是分组交换网络中的重要问题,其中包括通话时延和话音质量等。如果用户较多,QoS 的问题将会更加突出。另外,在跨网络互通时,性能和质量

也会下降<sup>[5]</sup>。

综合分析, 需要充分考虑引入 IP 所带来的优劣, 制定相应的策略, 扬长避短, 设计出符合要求的多媒体集群通信系统。

## 2.2 基于 IP 网络通讯模式的系统结构的设计

IP 网络的通讯模式主要分为 IP 单播、IP 组播、IP 广播以及应用层组播四种。前三种模式的实现位于 IP 网络的 IP 层, 通过路由器的存储、转发机制达到数据的发送目的, 最后一种模式的实现位于 IP 网络的应用层, 通过端系统的复制转发达到数据的发送目的。

基于 IP 网络构建的集群通信系统, 根据其功能需求首先必须具备会话服务器软件和客户端软件两部分, 以实现用户的接入和组呼叫的管理, 而由于基于 IP 网络, 所选择的通讯模式将直接影响系统整体结构的设计。下面根据上述不同通讯模式设计此集群通信系统的结构方案, 通过分析比较选择最优方案。由于 IP 广播禁止在 Internet 宽带网运行, 因此只设计了基于其他三种模式的方案。

基于 IP 网络的多媒体集群通信系统通过会话服务器与客户端之间进行的信令和音视频数据的交互最终实现集群通信功能。因此, 在设计基于每种通讯模式的系统结构的时候, 都需要综合考虑信令传输和音视频数据传输的有效性。

### 2.2.1 基于 IP 单播模式的系统结构设计

IP 单播: 主机之间“一对一”的通讯模式, 网络中的交换机和路由器对数据只进行转发不进行复制。如果 10 个客户机需要相同的数据, 则服务器需要逐一传送, 重复 10 次相同的工作。网络中的路由器和交换机根据其目标地址选择传输路径, 将 IP 单播数据传送到其指定的目的地。

基于 IP 单播模式设计的系统结构需要考虑下面两个问题:

#### 1. 信令的传输

由于会话服务器负责用户信息和组会话信息的管理, 每个用户的请求都需要交由会话服务器进行处理, 反馈结果, 通知其他用户状态更新。因此, 基于 IP 单播模式, 每个客户端都应该与会话服务器建立专门的信令链接, 用于与会话服务器之间进行信令信息的交互。

客户端的请求信息与会话服务器的反馈信息都是一对一的传输, 不需要额外复制分发。当会话服务器需要将相同的状态信息分发给多个用户时, 由于采用单播模式, 会话服务器需要将此信息复制多份, 通过与各客户端的信令链接进行多次发送。

#### 2. 音视频数据的传输



集群通信系统的通信采用半双工方式，同一时刻只有一个用户作为发言人发送音视频数据。基于单播模式，可以为音视频数据设计两种复制、分发方式：

### 1) 由发言者进行音视频数据的复制、转发

发言者需要从会话服务器得到所有需要此数据的用户名单，与每个用户建立音视频数据链接，然后将自己的音视频数据复制成多份，为每个用户发送一次。

### 2) 由会话服务器进行音视频数据的复制、转发

每个加入组会话的用户都与会话服务器建立专门的音视频数据链接。首先，发言者将音视频数据发送到会话服务器，然后会话服务器根据用户名单进行音视频数据复制，通过与每个用户的音视频数据通道进行多次发送。

第一种数据分发方式将所有的负载要求都集合在客户端，需要每个客户端拥有高处理能力和高带宽，这显然是不现实的。第二种方式则是将负载要求转移到服务器端，只要服务器拥有高配置，就可以完成整个数据的分发任务。因此，第二种方式更适合单播模式下音视频数据的有效传输。

综合上述分析，根据 IP 单播模式设计的基于 IP 网络的多媒体集群通信系统的系统结构如图 2.1 所示：

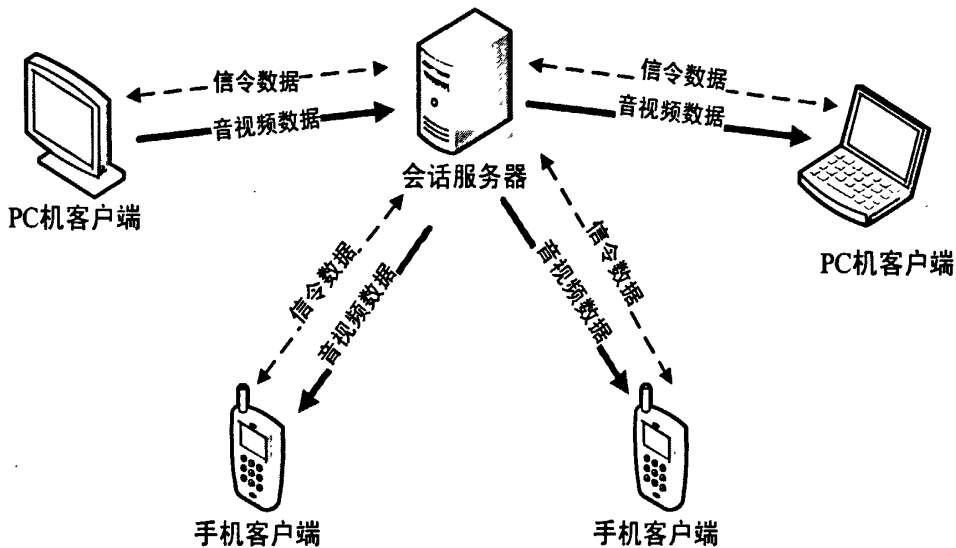


图 2.1 基于 IP 单播模式的系统结构图

从上图中可以看出，根据 IP 单播模式设计的基于 IP 网络的多媒体集群通信系统共有两类系统成员：会话服务器和客户端。他们的功能分别是：

#### 1. 会话服务器的功能

会话服务器是整个系统结构的核心，完成信令信息和多媒体信息的处理和分发。他与每个用户建立至少两个通道，分别用于信令和多媒体数据的传输。这样的会话服务器在处理能力和带宽方面要求通常都比较高。

#### 2. 客户端的功能

客户端与会话服务器分别建立信令和音视频数据通道, 向服务器发送用户请求, 处理服务器的反馈结果和通知信息, 并负责音视频数据的发送和接收。发言者将从本机获取的音视频数据发送到服务器, 接收者从会话服务器接收音视频数据, 展现给用户。

### 2.2.2 基于 IP 组播模式的系统结构设计

IP 组播: 主机之间“一对一组”的通讯模式, 也就是加入了同一个组的主机可以接受到此组内的所有数据, 网络中的交换机和路由器只向有需求者复制并转发其所需数据。主机可以向路由器请求加入或退出某个组, 网络中的路由器和交换机有选择的复制并传输数据, 即只将组内数据传输给那些加入组的主机。这样既能一次将数据传输给多个有需要的主机(加入组), 又能保证不影响其他不需要的主机(未加入组)的其他通讯<sup>[6]</sup>。

基于 IP 组播模式设计的系统结构需要考虑下面两个问题:

#### 1. 信令的传输

信令的传输主要是在组会话的开始和结束的过程中, 数据量小, 而且每个用户所需要交互的指令信息各不相同。在这样的情况下使用组播进行信令传输, 不仅达不到针对不同用户进行不同处理的要求, 而且每个组会话需要一个 IP 组播地址在会话初始和结束时使用, 造成地址浪费。

因此, 基于 IP 组播模式设计的系统结构中信令的传输依然采用 IP 单播模式。每个客户端与会话服务器建立专门的信令链接, 用于与会话服务器之间进行信令信息的交互。客户端的请求信息与会话服务器的反馈信息都是一对一的传输, 不需要额外复制分发。具体内容的分发形式与 IP 单播模式下的相同。

#### 2. 音视频数据的传输

由于音视频数据具有传输时间长、数据量大等特点, 使用 IP 组播具有突出优势。根据组播发送方的不同, 可以设计两种音视频数据分发方式:

##### 1) 服务器作为组播发送方

发言者客户端首先将多媒体数据发送到会话服务器, 然后由会话服务器根据配置好的组播地址进行发送。

##### 2) 发言者客户端作为组播发送方

发言者客户端从会话服务器获得组播地址信息进行组播初始化, 然后将多媒体数据发送到组播地址上。

无论是服务器作为组播发送方还是发言者客户端作为组播发送方, 首先都需要为此会话组建立一个组播组, 分配唯一组播地址, 所有参与此会话组的客户端需要进行组播初始化。

第一种方式中,发言者首先需要与服务器建立 IP 单播多媒体数据通道,将捕获的音视频数据通过此通道发送到服务器,由服务器进行 IP 组播发送。而第二种方式中,发言者不需要与服务器建立专门 IP 单播多媒体数据通道,获取发言权之后,他只需要进行组播初始化,将捕获的多媒体数据发送到组播地址上,然后由路由器完成数据的复制分发任务即可。由于 IP 组播中的发送方只需要发送一份数据,复制转发任务由路由器完成,不需要考虑负载和带宽问题,因此,第二种方案更简单可行。

基于上述分析,根据 IP 组播模式设计的基于 IP 网络的多媒体集群通信系统的系统结构如图 2.2 所示:

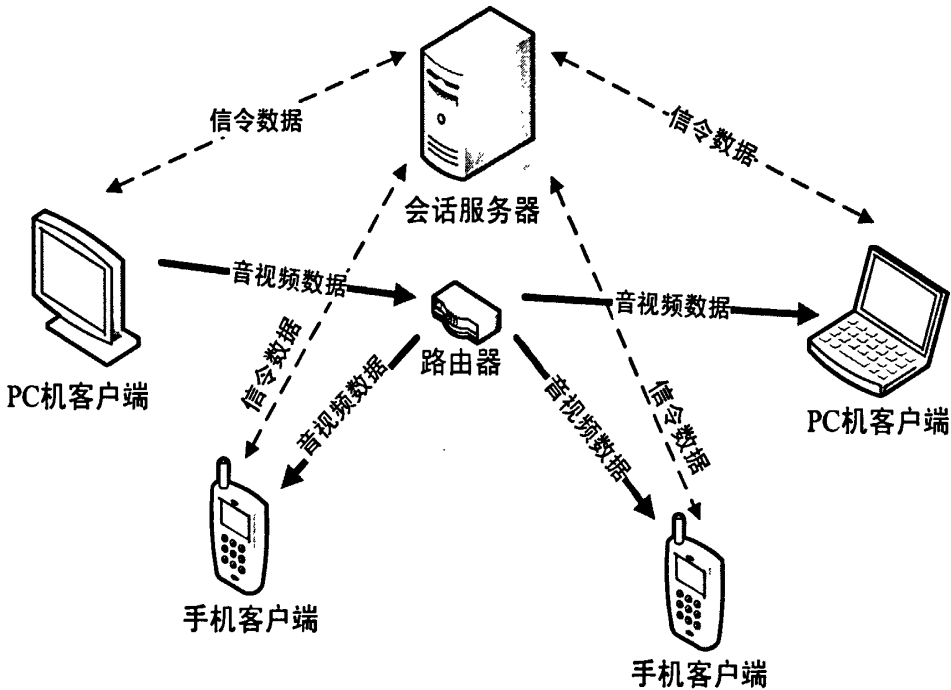


图 2.2 基于 IP 组播模式的系统结构图

从上图中可以看出,根据 IP 组播模式设计的基于 IP 网络的多媒体集群通信系统也分为两类系统成员:会话服务器和客户端。他们的功能分别是:

### 1. 会话服务器的功能

会话服务器仅负责信令信息的处理和分发。他采用 IP 单播方式与每个用户建立一个信令通道,处理用户请求,反馈处理结果,通知状态更新,并为每个会话组分配互不冲突的 IP 组播地址。

### 2. 客户端的功能

客户端除与会话服务器建立信令通道进行信令交互外,还需要根据服务器分配的组播地址进行音视频数据的发送和接收。发言者将从本机获取的音视频数据发送到此会话组的 IP 组播地址上,接收者通过参加此 IP 组播组接收音视频数据,展现给用户。

### 2.2.3 基于应用层组播模式的系统结构设计

应用层组播：是一种由主机终端在应用层实现组播通信的组播方案。参与组播通信的节点自组织地配置进叠加网络，端系统在组播网络中执行应用层组播路由算法来确定一条虚拟逻辑组播路径，数据包沿着这条逻辑链路转发。逻辑路径其实是两节点传统网络物理链路组成的通信路径，这条实际的物理路径对端系统而言是透明的。应用层组播参考 IP 组播的思想，但是数据是在主机上而不是路由器上实现复制和转发，数据报沿着逻辑链路转发<sup>[7]</sup>。

基于应用层组播构建的系统，应用层组播平台完成其所有的数据复制和分发任务。这样，信令和音视频数据的传输负担就从会话服务器和客户端转移到了应用层组播平台上。

基于应用层组播模式设计的系统结构需要考虑下面两个问题：

#### 1. 信令的传输

由于应用层组播平台完成信令的分发，各客户端和会话服务器只需要和应用层组播平台建立一条 IP 单播信令传输通道，将需要分发的信令信息发送到这条通道上即可，剩余的复制、转发任务交由应用层组播平台负责。客户端和服务端之间不需要建立直接连接。

#### 2. 音视频数据的传输

音视频数据的传输与信令相同。需要发送/接收音视频数据的客户端与应用层组播平台建立数据通道，发言者将数据发送到应用层组播平台上，由其完成复制分发任务，接收者通过与组播平台建立的数据通道接收多媒体数据。会话服务器不需要建立专门的多媒体数据链接，只需要对客户端的发送/接收进行信令控制即可。

根据应用层组播模式设计的基于 IP 网络的多媒体集群通信系统的系统结构如图 2.3 所示。从图 2.3 中可以看出，根据应用层组播模式设计的基于 IP 网络的多媒体集群通信系统分为三类系统成员：会话服务器、应用层组播平台以及客户端。他们的功能分别是：

#### 1. 会话服务器的功能

会话服务器负责处理用户请求，反馈处理结果，维护用户信息和会话组信息的更新等。会话服务器的功能仅仅是信息处理，而不再进行任何数据的复制、分发。它只是将需要发送的数据传送给应用层组播平台，由其进行分发处理。

#### 2. 应用层组播平台的功能

应用层组播平台完成系统中所有数据的复制、分发，包括信令信息和多媒体数据。它通过自己的地址分配和路由机制，将数据有效地发送到需要此数据的终端。

### 3. 客户端的功能

客户端的功能是发送处理过的信令和多媒体数据到应用组播平台，或者从该平台接收信令以及多媒体数据进行进一步处理。客户端与应用层组播平台需要建立至少两条链路，分别用于信令和多媒体数据的交互。

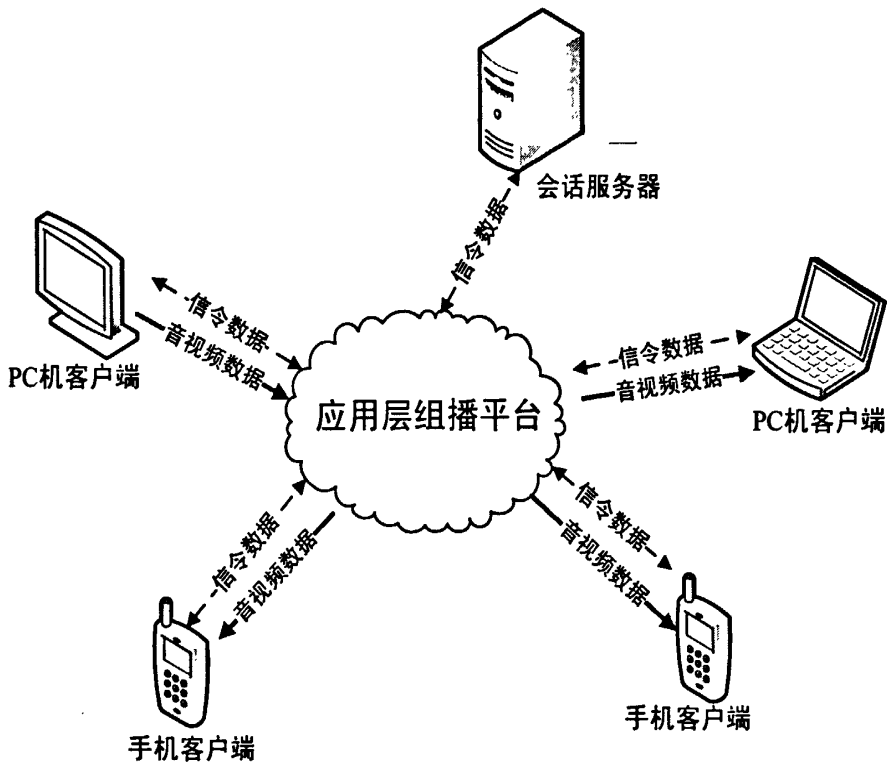


图 2.3 基于应用层组播模式的系统结构图

#### 2.2.4 系统结构设计方案的选择

上文根据 IP 单播、IP 组播以及应用层组播三种通讯模式分别设计了基于 IP 网络的集群通信系统的整体结构以及各部分的功能。由于这三种通讯模式各自的特点不同，以其为基础所设计的三种系统结构也各有优劣：

##### 1. 基于 IP 单播模式设计的系统结构

###### ● 优势

系统结构紧凑，是严格的服务器/客户端模式，开发相对容易；

###### ● 劣势：

服务器负责音视频数据的复制、分发，对其处理能力和带宽要求极高，一旦服务器出现问题，整个系统就会崩溃；服务器为每个用户维护多条链接，网络规模受限。

##### 2. 基于 IP 组播模式设计的系统结构

###### ● 优势

音视频数据只需一份在网络中传输, 由路由器负责复制, 对服务器要求大大降低, 同时节省网络系统资源。

- 劣势

服务器需要为每个用户维护一条信令链接, 同时, IP 组播技术需要路由器的协议支持, 在大于局域网规模的网络环境内较难推广, 这些都限制了系统整体规模的扩大。

### 3. 基于应用层组播模式设计的系统结构

- 优势

由应用层组播平台负责所有数据的复制分发, 对服务器的要求大大降低; 系统规模随应用层组播平台所支持的网络规模不断扩大。

- 劣势

相当于在 IP 单播模式设计的系统结构中增加了中继功能, 因而系统延迟增大。

通过对基于 IP 单播、IP 组播、应用层组播三种模式设计的集群通信系统结构的分析, 将三种结构的比较总结如表 2.2 所示:

系统要求 系统结构	会话服务器 配置要求	系统规模 可扩展性	系统延迟的决定性因素
基于 IP 单播设计的 系统结构	非常高	极其受限	服务器处理能力
基于 IP 组播设计的 系统结构	较高	较受限	服务器的处理能力
基于应用层组播 设计的系统结构	不高	随应用层组播网 的规模可扩展	应用层组播平台的 处理能力

表 2.2 三种系统结构比较分析表

会话服务器配置的高低直接决定整个集群通信系统的构建成本。同时, 为了能够容纳尽可能多的用户, 系统规模的可扩展性也是至关重要的。从上表的比较中可以看出, 基于应用层组播设计的系统结构具有构建基于 IP 网络的多媒体集群通信系统所需要的, 而其他两种结构无法解决的优势条件。在选择一个具有优越性能的应用层组播平台的基础上, 基于应用层组播模式设计的系统结构将拥有更大的优势, 因而成为这三种设计方案的首选。

## 2.3 应用层组播平台的分析与选择

### 2.3.1 系统对应用层组播平台的需求分析

应用层组播平台是基于应用层组播模式设计的系统结构的基础, 它的网络规

模和性能直接影响到基于 IP 网络的多媒体集群通信系统的规模和性能。因此,应用层组播平台应该具有下述特性:

### 1. 可扩展的网络拓扑

应用层组播网的可扩展性直接影响构建在该网络之上的集群通信系统的规模。应用层组播网是在 IP 网络基础之上构建的覆盖网,每个网络节点拥有全网唯一的逻辑地址,节点之间建立逻辑链路,进而构成网络拓扑。因为不需要配置底层 IP 网络,应用层组播网可以根据逻辑地址进行层次划分,分级管理,进而不断扩展整个网络的规模。

### 2. 有效的路由机制

有效的路由机制可以加快数据的传输,减少集群通信系统的网络延迟。网络规模越大,越需要有效的路由机制来进行数据的转发。如何动态更新应用层组播网中的网络节点的状态,维护整个网络的路由信息,防止数据无意义的分发,这将直接影响到构建在应用层组播平台之上的应用系统的性能。

### 3. 有效的数据分发机制

有效的数据分发机制可以简化客户端以及服务器软件的设计,优化数据传输过程。组播平台可以通过发送方的告知信息来确认接收用户,也可以通过接收用户的注册信息自动完成目的地址列表的更新,后者将目的地址的管理集成到组播平台上,简化了发送方程序的设计。

### 4. 支持数据实时性传输

集群通信系统的功能是提供组成员之间的实时通信。因此,作为数据中继的应用层组播平台必须支持数据的实时性传输,将传输延迟缩短到最小,以提供最优的性能。

### 5. 为时间敏感型数据提供缓冲等服务

由于集群通信系统传输的音视频数据均是时间敏感类型的数据,网络传输的延迟和抖动对音视频最后的播放都会产生很大影响。因此,组播平台需要提供缓冲等一系列服务来优化 IP 网络传输带来的不良影响。

综合对上述方面的思考,本文最后选择了 NaradaBrokering 项目作为应用层组播传输平台。

## 2.3.2 NaradaBrokering 平台的选择与分析

NaradaBrokering 项目(以下简称 NB 系统)是由美国印第安纳大学的社区网络实验室开发的,致力于大规模分布式流化系统研究的开源的通用的信息发送解决方案。NB 网络是一个分布式、分级管理的代理网络,提供有效的路由和消息分发策略,并能够在不依赖于单个代理是否可用的情况下提供可扩展性和可靠性。

该项目已经得到美国国家科学基金会、美国能源部以及英国开放中间件基础设施研究所的资助。

作为基于 IP 网络的多媒体集群通信系统的应用层组播平台，NaradaBrokering 的优越性主要体现在：

### 1. 在网络拓扑的可扩展方面

NB 网络是位于 IP 网络之上的覆盖网 (Overlay networks)。最小单元是每个 NB 代理节点 (Broker)。它在真实的 IP 网络拓扑之上构建了自己的逻辑虚拟网络拓扑，拓扑中的“节点”是端用户，边是“节点”之间的链接，并为每个“节点”分配此系统中的唯一的逻辑地址。

该系统拓扑框图如图 2.4 所示：

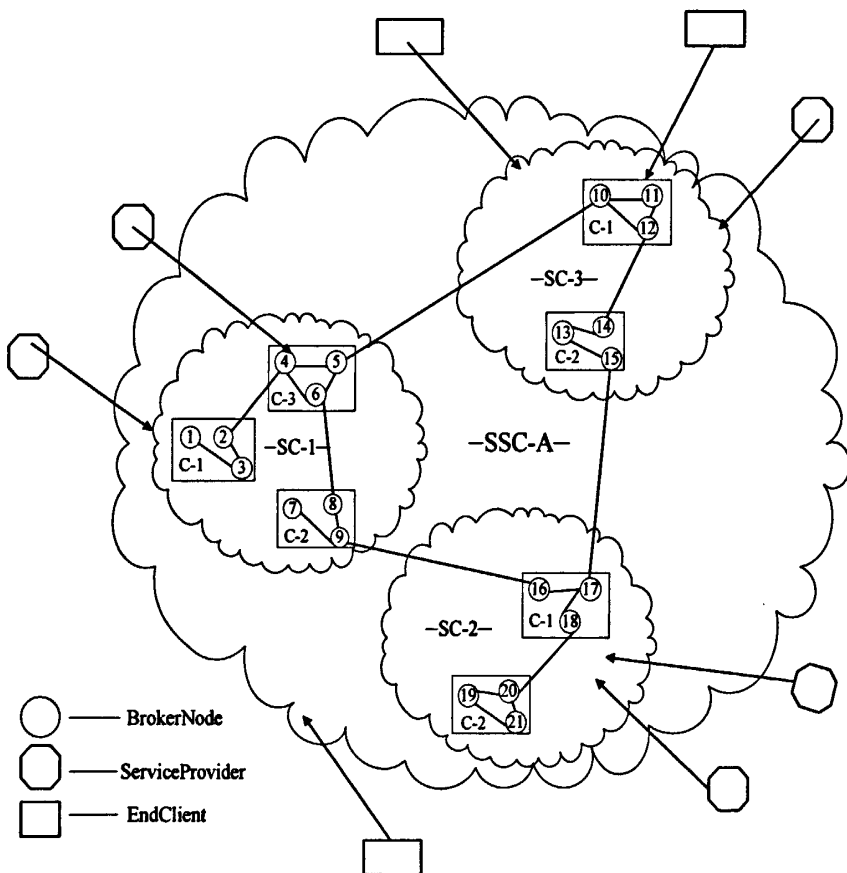


图 2.4 NB 系统拓扑图

从图 2.4 中可以看出这是一个分层的网络系统。代理节点 (Broker) 是最小的网络单元。符合某种规定条件的多个代理节点组成一个集群 (Cluster)，多个集群又组成一个超级集群 (Super-Cluster)，多个超级集群组成更加超级的集群 (Super-Super-Cluster)，依次类推，形成一个范围不断扩大的，级别不断升高的网络系统<sup>[8]</sup>。每个级别的集群中，至少存在一个控制者，管理此级别集群和其他同级别集群的连接、通信<sup>[9]</sup>。每个代理节点在加入此网络系统的时候，都会被指定或者从控制者那里获得一个全网唯一的逻辑地址。



## 2. 在路由机制方面

在 NB 网络中,不同代理节点之间的链接也具有不同的级别。根据网络拓扑和链接的分级特性,每个代理节点中维护着一个动态更新的代理网络图 BNM (Broker Network Map) 以及根据此图计算出的最短路由<sup>[10]</sup>。每个高一级别的链接的建立,都会促使其低级别的相应节点更新 BNM 和最短路由。

事件路由即是分发事件到相关客户端的过程。这个过程包括匹配内容,计算目的地址,路由内容到相关地址等。这里的相关地址指的是事件到达目的地址所必须经过的那些代理节点的地址。当事件在 NB 网络中传输的时候,与之关联的路由轨迹不断被修改来反映它的整个系统中的分发流程。事件经过的每个代理节点都是根据它的路由轨迹信息和计算过的地址来决定接下来的路由。

## 3. 在数据分发机制方面

NB 系统中的通信是异步的,系统通过将通信内容封装到 NB 系统特定事件 NBEvent 中进行交互。事件是 NB 系统的核心,它将信息封装在不同的层次中。

NB 系统通过发布/订阅模式来实现数据的分发。整个通信网络由三种实体组成:发布者、订阅者、代理节点。发布者和订阅者不直接相连。作为发布者或者订阅者的客户端可以创建以及发布事件,指明对特定主题的事件感兴趣,以及接收符合特定主题的事件<sup>[11]</sup>。

NB 系统的发布/订阅网络模型如图 2.5 所示:

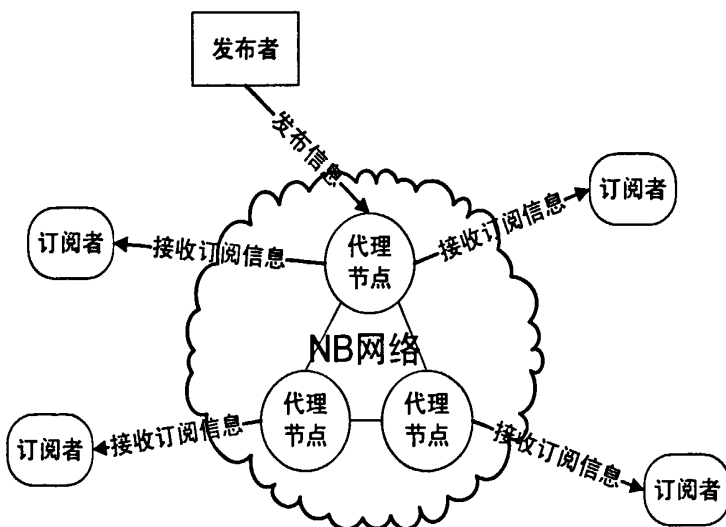


图 2.5 NB 系统发布/订阅网络模型图

在这样的网络体系中,订阅者首先要订阅某个特定的主题,将兴趣通知与其相连的代理节点。然后由该代理节点通过分级传输将此订阅信息通知到整个网络,完成订阅者的兴趣注册。这样,当发布者发布某个主题的事件到其连接的代理节点的时候,该节点会根据已经存储的订阅信息计算目的地址,再利用由 BNM 得到的最短路径选择下一跳的代理节点,转发事件<sup>[12]</sup>。

#### 4. 在对实时性传输的支持方面

NB 系统通过工厂设计模式设计了一种灵活的传输框架。通过这种框架结构，可以选择不同的传输协议，包括：TCP 协议、用户数据报协议 UDP (User Datagram Protocol) 以及实时传输协议 RTP (RealTime Transport Protocol) 等<sup>[13]</sup>。

NB 系统在保证可靠消息传送机制的同时，设计了一种专门用于实时多媒体流传输的事件类型 RTPEvent。该事件拥有较小的事件头和较小的计算损耗。RTPEvent 封装的媒体内容由四部分组成：事件头、媒体头、源信息、有效负载。如图 2.6 所示：

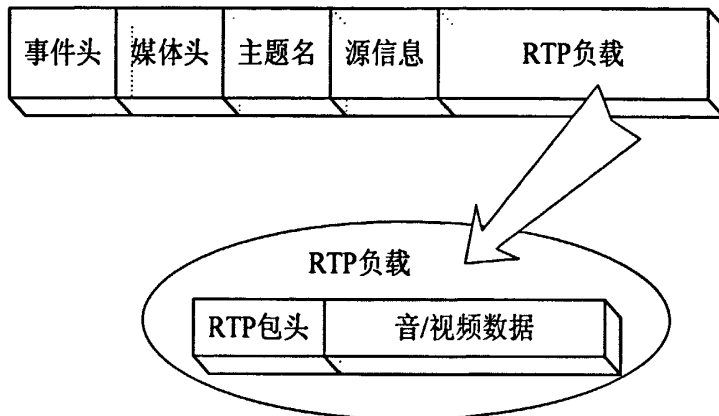


图 2.6 NB 系统 RTPEvent 结构图

事件头标识该事件的类型为 RTPEvent，用于区别 NB 系统中其他类型的事件，同时还有此产生的时间戳，用于消息缓存。媒体头标识该事件承载的数据的类型为音频、视频或者控制信息。源信息是为了避免事件在系统中路由的时候出现回环。最后，有效负载是发布客户端发布的多媒体数据，可以是任何数据类型。这样，RTPEvent 的头部长度的长度为 15 个字节<sup>[14]</sup>。也就是说，增加 15 个字节的数据到客户端的原始音视频流，就可以让这些流数据正确地在 NB 系统中完成分发。

#### 5. 在对时间敏感型数据的处理方面

NB 系统为时间敏感型数据提供了下面集中服务：

##### 1) 网络时间服务 NTS (Network Time Service)

通过实现 NTP (Network Time Protocol) 协议，NB 系统中的每个事件可以打上时间戳。由于采用的是统一的网络时间，所有代理节点的时钟都是同步的。这样就保证了事件在时间上的一致性<sup>[15]</sup>。

##### 2) 时间缓冲服务 TBS (Time Buffering Service)

在大型的分布式网络环境中，为了能够减少抖动的影响，NB 系统提供了一个容量可以根据客户需要定制的缓存。NB 系统的缓冲服务首先存储消息，然后将它们按照各自的时间戳进行排序，最后释放这些消息交由客户端处理。

##### 3) 可靠传送服务 RDS (Reliable Delivery Service)

NB 系统通过部署可靠传输服务节点来提供信息的可靠传送。该类型的节点在系统中可以存在多个,并通过数据库进行信息存储。例如,当代理节点或者链接失效,可靠传输服务节点会触发重传机制,使信息按照其他有效路径到达<sup>[16]</sup>。当订阅实体突然断开链接,而后又重新连接到 NB 系统的时候,可靠传输服务节点会把该实体订阅的信息从其存储数据库中取出,再次发送给该订阅实体。因此,这种可靠传输机制从本质上也就支持了重新播放和错误恢复服务。

同时,作为成熟的项目,NaradaBrokering 已经被广泛应用于地震科学、粒子物理、环境检测、网络会议等方面的项目中,起到数据有效分发的功能<sup>[17]</sup>。

因此,选择 NaradaBrokering 作为文中所设计的集群通信系统的应用层组播平台不仅可以满足系统的功能需求,更能提高、优化系统的整体性能。

## 2.4 基于 NaradaBrokering 平台的多媒体集群通信系统的整体设计

### 2.4.1 系统整体结构的设计

由于应用层组播平台 NaradaBrokering 的选择,根据该平台采用的发布/订阅模式的特点及其提供的一系列服务,基于应用层组播模式设计的系统结构可以得到进一步的细节性设计。

下面从三个方面对基于 IP 网络的多媒体集群通信系统进行进一步设计:

#### 1. 系统各部分功能设计

系统成员依然主要分为三类:会话服务器、客户端(包括 PC 机和手机两种)以及应用层组播平台 NB 系统。

**会话服务器功能:**用户请求信息的处理和反馈,用户和会话组状态的维护及更新,每个会话组中发言权的分配及回收。

**客户端功能:**发送用户请求信息及捕获的音视频数据,并对接收的服务器反馈信息以及其他用户发送的音视频数据进行相应处理。

**NB 系统平台的功能:**分别与客户端和会话服务器建立基于不同传输协议的多个链接,并根据发布/订阅模式,将从这些链接上收到的所有数据进行有效的复制和分发。

#### 2. 信令的传输方式设计

所有加入该集群通信系统的客户端以及会话服务器都与 NB 系统中的代理节点建立一条用于信令传输的链接。为了保证信令传输的可靠性,该链接采用了面向连接的传输控制协议 TCP (Transport Control Protocol) 协议。

需要发送的信令信息首先在发送客户端或者是会话服务器端进行 NBEvent 的事件封装,将此信息与主题绑定在一起,然后通过 TCP 协议传输到临近的 NB 系统中的代理节点。代理节点根据自身存储的该事件所对应主题的订阅信息计算目

的地址, 将此事件进行 NB 网络分发, 最后发送到所有订阅了该主题的客户端或者服务器。

### 3. 音视频数据的传输方式设计

基于 NB 系统对实时传输的支持, 集群通信系统中音视频数据的传输采用应用广泛的实时传输协议 RTP/RTCP。

RTP 协议是针对多媒体数据流的实时传输协议, 一般用于一对一或一对多的传输情况下, 提供时间信息和实现流同步。RTP 通常使用用户数据报协议 UDP (User Datagram Protocol) 来传送数据, 但也可以在 TCP 或异步传输模式 ATM (Asynchronous Transfer Mode) 等其他协议上工作。

实时传输控制协议 RTCP (Real-time Transport Control Protocol) 和 RTP 一起提供流量控制和拥塞控制服务。在 RTP 会话期间, 各参与者周期性地传送 RTCP 包, 这些 RTCP 包中包含已发送的数据包的有效数据, 丢失的数据包数量等统计资料。因此, 服务器可以利用这些信息动态地改变传输速率, 甚至改变有效载荷类型<sup>[18]</sup>。RTP 和 RTCP 配合使用, 能有效地反馈和以最小的开销使传输效率最优化, 因而特别适合传输网络中的实时数据。

根据上述设计, 基于 IP 网络的多媒体集群通信系统的结构图如图 2.7 所示:

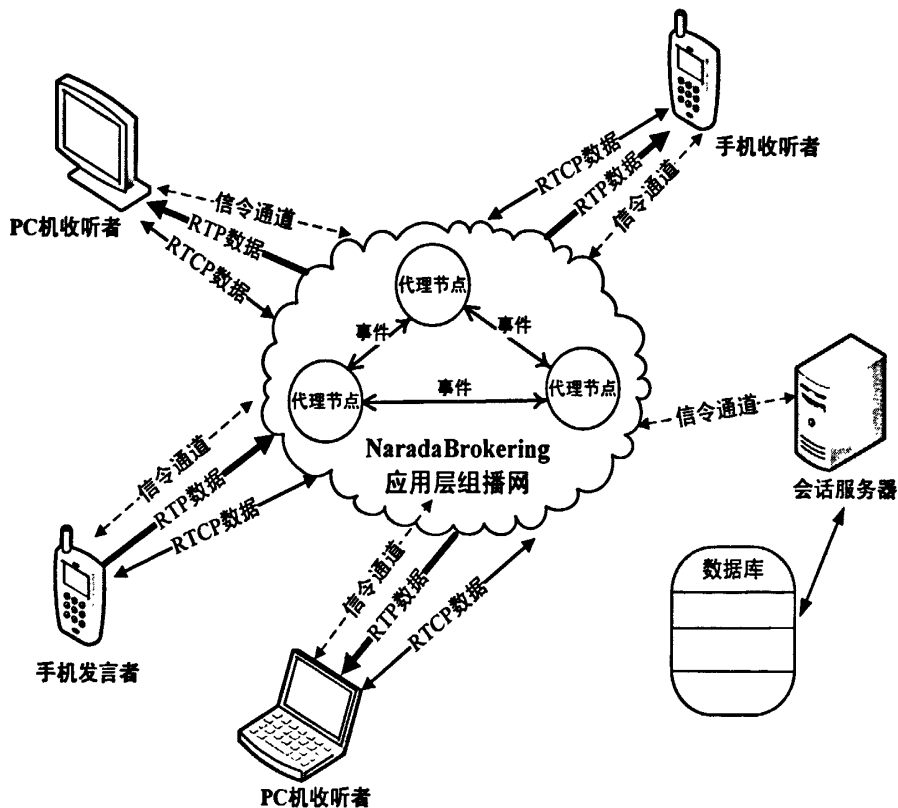


图 2.7 基于 IP 网络的多媒体集群通信系统结构图

从图 2.7 中可以看出, 每个客户端在会话组初始化之后都与 NB 系统建立两个

链接, 分别用于 RTP 数据和 RTCP 数据的发送和接收 (如果支持视频, 还需要另外建立两个链接)。当客户端将 RTP 数据通过链接发送到代理节点上, 代理节点为其进行 RTPEvent 封装, 将此音视频数据与整数型主题绑定在一起, 然后以发布/订阅的模式进行 NB 网络内的事件分发, 最终发送给每个订阅端用户。

由于 NB 系统的发布/订阅模式, 所有客户端都不需要和会话服务器建立直接连接, 而会话服务器也不需要为每个客户端的音视频流进行一对多的分发。这样, 对服务器的负载和处理能力的要求大大降低。会话服务器的主要功能就是用户信息处理和状态维护, 以及状态通知等。考虑到服务器可能出现异常、失效等情况, 在后续完善中, 可以建立备份的同步服务器, 当一个服务器出现失效, 同步服务器根据同步状态继续处理。由于是基于发布/订阅的主题消息机制, 用户甚至不会感觉到服务器的替换。

和服务器相连接的数据库用来存储用户信息, 包括用户名, 密码等, 以配合服务器在处理登录、创建会话等任务时的验证工作。

#### 2.4.2 系统工作流程的设计

为了缩短开发周期, 尽快验证系统的有效性和可用性, 作为系统设计与实现的最初版本, 文中设计的基于 IP 网络的多媒体集群通信系统只实现了集群通信中的一对一呼叫、组呼功能以及简单的分级管理功能。组内成员分为两个优先级: 主席和普通用户。主席具有创建会话组和邀请其他用户的权利, 而普通用户只能被动接受邀请。在系统设计实现的过程中会充分考虑系统的可扩展性设计, 这样就能为后续功能扩展工作提供便利。

组呼是集群通信区别于普通移动电话的特色业务, 能够提供组内成员的一对多的通信。每次组呼叫的建立称为一次会话。为了提供方便快捷的多方实时协作, 系统中通常还需要引入发言权控制策略。

图 2.8 是本系统从会话建立、邀请加入、分发发言权到发送多媒体数据, 再到结束会话的基本流程时序图 (忽略登录和退出过程)。用户 A 作为主席创建了会话 S, 成功邀请用户 B 加入 S, 并率先获得发言权进行发言。为了展示方便, 用户 A 只邀请了用户 B 的加入。如果考虑邀请多用户, 其他用户的工作流程与用户 B 相同。

从图 2.8 中可以看出, 用户 A、用户 B 以及会话服务器的所有信息交互都是通过 NB 网络转发实现的。用户的多媒体数据也是通过 NB 网络进行统一分发。当然, 上图并没有考虑各种用户请求被拒绝的情况。实际上, 用户的所有请求最终都是传递给会话服务器进行处理并根据结果反馈成功或者失败信息给请求用户的。在会话建立成功之后, 会话服务器会根据此会话的状态进行判断, 发送会话

空闲等状态信息给加入会话的所有用户。用户根据这些信息可以重新发送申请发言权或者释放发言权指令。当主席结束本次会话，会话中的其他成员将被强迫退出，发言人取消发言。

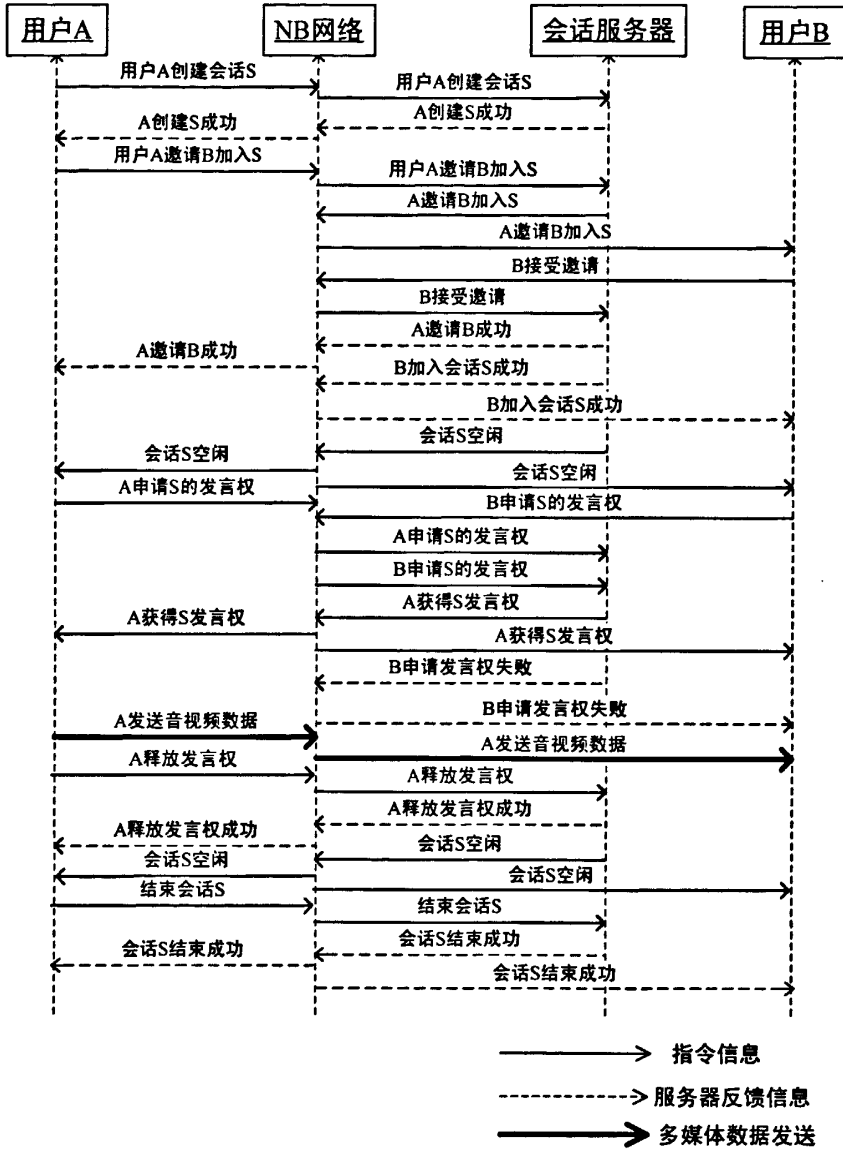


图 2.8 系统组呼流程图

### 2.4.3 系统控制协议的设计

系统控制协议规定了系统中各成员通信的具体信令内容，它的设计与实现直接影响系统协作的有效性以及系统后续功能的可扩展性。由于文中设计的基于 IP 网络的多媒体集群通信系统选择了应用层组播软件 NaradaBrokering 作为传输平台，如果集成现有控制协议，不仅需要考虑到协议功能的完备性，还需要更改协议

使其适合 NB 系统的发布/订阅机制。因此,基于本系统的功能需求设计一个可扩展的系统控制协议是更有效的方法。

可扩展标记语言 XML (Extensible Markup Language) 是标准通用标记语言 SGML (Standard Generalized Markup Language) 之一。它是 Internet 环境中跨平台的,依赖于内容的技术,是当前处理结构化文档信息的有力工具。扩展标记语言 XML 是一种简单的数据存储语言,使用一系列简单的标记描述数据,而这些标记可以用方便的方式建立。虽然 XML 文件比二进制数据要占用更多的存储空间,但极其简单、易于掌握、使用和扩展却使它倍受网络开发人员的青睐<sup>[9]</sup>。

通过分析 XML 语言的可扩展性以及小文件时的快速解析性,本系统在参考系统功能需求的基础上设计并实现了一套基于 XML 的集群通信控制协议 XTCCP (XML Based Trunked Communication Control Protocol)。

XTCCP 协议的消息格式分为两部分:

### 1. 消息头

由于采用 XML 为协议的描述语言,每个 XTCCP 消息成为一个小的 XML 文件。因此,XML 文件的文件头也是 XTCCP 协议的消息头。

消息头的格式为:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

这是 XML 文件的声明,消息接收端的解析器根据此格式判定消息为 XML 文件,然后进行内容解析。`version="1.0"`定义 XML 的版本是 1.0, `encoding="ISO-8859-1"`定义 XML 文件使用的编码为 ISO-8859-1,即 Latin-1/西欧字符集,这也是 JAVA 网络传输使用的标准字符集。

### 2. 消息体

因为 XML 是标记性语言,所以 XTCCP 协议的消息体由多个预定义的标签及其属性值组成,并按照树形排列。

如登录/退出请求指令的消息体为:

```
<LogIn>
```

```
<UserId>UserId</UserId>
```

```
<Password>Password</Password>
```

```
</LogIn>
```

LogIn 标签是消息体的根元素,表明该消息是登录/退出请求。UserId、Password 标签是 LogIn 标签的子元素,表明该消息包含的具体信息元素及其属性值。UserId 标签表示用户名,Password 表示用户密码。通过解析这些标签,获得标签属性值,就可以得到用户邀请登录/退出的全部信息,进行具体处理。

根据系统功能需求,XTCCP 协议包括的指令及其协议描述如表 2.3 所示:

指令 名称	XTCCP 协议 根标签描述	指令 名称	XTCCP 协议 根标签描述
注册请求	Registry	会话更新事件	SessionEvent
登录请求	LogIn	登录结果	LogInReply
退出请求	LogOut	退出结果	LogOutReply
创建会话请求	CreateSession	创建会话结果	CreateSessionReply
邀请请求	Invite	邀请结果	InviteReply
申请发言权请求	FloorRequest	释放发言权请求	FloorRelease
申请发言权成功	FloorGranted	申请发言权失败	FloorDeny
会话信息查询请求	SessionInfoQuery	会话信息更新事件	SessionInfoEvent
用户信息查询请求	UserInfoQuery	用户信息更新事件	UserInfoEvent
发言权空闲事件	FloorIdle	发言权被申请事件	FloorTaken
用户登录/退出事件	LogEvent	用户超时事件	TimeOutEvent
邀请事件	InviteEvent	释放发言权结果	FloorReleaseReply

表 2.3 XTCCP 协议信令消息列表

XTCCP 协议中一个完整的登录请求消息表示为:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LogIn>
<UserId>UserId</UserId>
<Password>Password</Password>
</LogIn>
```



与之对应的服务器的反馈结果消息表示为:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LogInReply>
<UserId>UserId</UserId>
<Result>success</Result>
</LogInReply>
```

根据 XTCCP 协议, 每个用户请求都有与之对应的服务器反馈结果。同时, 根据用户和会话状态的更新, 服务器还需要发送时间更新通知给相应用户。虽然 NB 系统的客户端通信属于异步通信, 但是在时延允许的情况下, 通过这种回复确认机制, 保证了消息的可靠性传输。而利用 XML 为协议的描述语言, 对于系统后续功能的扩展提供了方便的条件

## 2.5 本章小结

本章首先对基于 IP 网络的集群通信系统进行了用户需求、功能需求以及系统特性的分析, 指出构建这样一个集群通信系统的意义。然后根据 IP 网络的通讯模式 IP 单播、IP 组播、应用层组播分别设计了三种系统结构, 通过比较、分析, 最后选择应用层组播设计方案作为本系统的系统结构设计。本章在分析基于 IP 网络的多媒体集群通信系统对应用层组播平台的特性要求的基础上, 选择了开源项目 NaradaBrokering 作为系统应用层传输平台, 并根据此平台的特点对系统各部分功能、系统工作流程以及系统控制协议进行了详细设计。

## 第三章 会话服务器和 PC 机客户端的设计与实现

本章根据抽象化、模块化、可扩展原则将会话服务器分为服务模块、服务管理模块、消息管理模块以及 NB 系统发布/订阅模块, PC 机客户端分为用户界面模块、信令处理模块、多媒体处理模块以及 NB 系统发布/订阅模块, 然后对各模块进行了功能设计和具体实现。

### 3.1 设计原则

基于 IP 网络的多媒体集群通信系统中会话服务器以及 PC 机客户端的设计遵循以下原则:

#### 1、抽象化原则

软件设计的困难随着问题的规模和复杂性不断增大, 抽象是控制复杂性的基本策略。使用抽象技术便于人们用“问题域”本来的概念和术语描述问题, 而无须过早地转换为那些不熟悉的结构。

本文将会话服务器提供的所有服务抽象成一个服务模块, 以接口的形式实现, 统一提供请求信息的解析操作。请求与服务之间通过消息模式交互。会话服务器底层不需要根据特定请求类型调用特定服务, 只需要将消息交给服务模块的服务基类, 由其进行协议解析, 然后调用具体服务实例进行处理, 最后反馈处理结果。

#### 2、模块化

模块化是指将各功能需求采用模块方式实现, 模块是数据说明、可执行语句等程序对象的集合。模块可以单独被命名且可通过名字来访问。在面向对象的软件设计过程中, 模块统以接口和类的形式出现, 将对象的数据以及可对数据执行的操作封装在接口和类的实例中。

在解决问题的实践中, 如果把两个问题结合起来作为一个问题来处理, 其理解复杂性大于这两个问题被分开考虑时的理解复杂性之和。因此, 把一个功能复杂的软件系统划分成易于理解的比较单纯的模块结构, 可以降低软件的复杂度。

本文将会话服务器按其功能分为服务模块、服务管理模块、消息管理模块以及 NB 系统发布/订阅模块, PC 机客户端按其功能分为用户界面模块、消息处理模块、多媒体处理模块以及 NB 系统的发布/订阅模块。模块之间协调工作, 完成系统的整体功能的实现。各模块的具体功能将在后续章节作详细介绍。

#### 3、可扩展性原则

在软件的设计过程中应该充分考虑系统的可扩充能力。用户业务需求的增加及功能的追加, 以及技术的不断进步都要求系统必须具有扩展的余地。

本文在综合了 NB 系统异步消息机制的基础上, 在会话服务器的设计中采用了

消息接收与消息处理分模块进行，模块之间松耦合的方法。只要在会话服务器的服务模块增加一个继承自该模块的服务基类且实现了具体消息处理功能的服务子类，该服务子类就可以被服务模块调用，被服务管理模块有效管理，进而完成系统功能的扩展。

## 3.2 开发平台简介

基于 IP 网络的多媒体集群通信系统采用集成开发环境 NetBeans IDE 6.7 为应用开发工具，进行会话服务器和 PC 机客户端代码的开发调试

### 1. NetBeans IDE 简介

NetBeans 是由 Sun 公司建立的开放源码的软件开发工具，是一个开放框架，可扩展的开发平台，可以用于 Java、C/C++ 等的开发。NetBeans 是一个开发平台，可以通过扩展插件来扩展功能。它是一个全功能的开放源码的 Java IDE，可以帮助开发人员编写、编译、调试和部署 Java 应用，并将版本控制和 XML 编辑融入其众多功能之中。NetBeans 开发环境可供程序员编写、编译、调试和部署程序。另外也有巨大数量的模块来扩展 NetBeans IDE，它是一个免费产品，不限制其使用形式。

NetBeans IDE 的功能特性包括：

- 易于使用的 Java GUI 生成器

通过在画布上放置组件并使其对齐，创建具有专业外观效果的 Java 图形用户界面(GUI)。

- 可视化的 Web 和 Java EE 开发

可视化地生成标准的 WEB 应用程序，提供完整的 EJB (Enterprise Java Beans) 开发工具套件。

- 可视化的移动开发

创建、测试并调试在移动电话、机顶盒和掌上电脑上运行的 GUI 应用程序。

- 可视化的 UML 建模

通过 UML 建模，将软件设计从实现中分离出来。

- 对 Ruby 语言和 Rails 框架的支持

强大的 Ruby 编辑器（具有代码完成功能）、调试器以及对 Rails 的全面支持。包括 JRuby 运行环境。

- 开发功能全面的 C/C++ 编辑器、调试器、项目模板以及对多项目配置的支持

### 2. NetBeans IDE 开发平台的配置

在安装 NetBeans IDE 程序之前，首先安装 Sun 公司的 JDK，即 Java 开发工具包 (Java Development kit)。JDK 包括了 Java 运行环境 (Java Runtime Environment)，

一系列 Java 程序编译、运行工具和 Java 基础的类库,是所有 Java 开发的基础。本系统要求 JDK 版本为 1.6。

然后下载并安装 NetBeans IDE 程序,并对相关运行参数进行配置。

### 3.3 会话服务器的设计与实现

#### 3.3.1 会话服务器的模块化设计

基于 IP 网络的多媒体集群通信系统中的会话服务器的功能主要包括:用户请求信息的处理和反馈;用户和会话组状态的维护及更新;每个会话组中发言权的分配及回收等。为了实现这些功能,会话服务器需要提供一系列的服务,像注册服务、登录服务、发言权控制服务以及邀请服务等,同时需要有效的机制对这些服务进行管理控制,以节省系统资源。

根据抽象化、模块化、可扩展性原则,将会话服务器按照功能分为四大模块:服务模块、服务管理模块、消息管理模块以及 NB 系统发布/订阅模块。每个模块完成抽象的系统功能,模块之间通过固定接口交互。

会话服务器设计的模块化如图 3.1 所示:

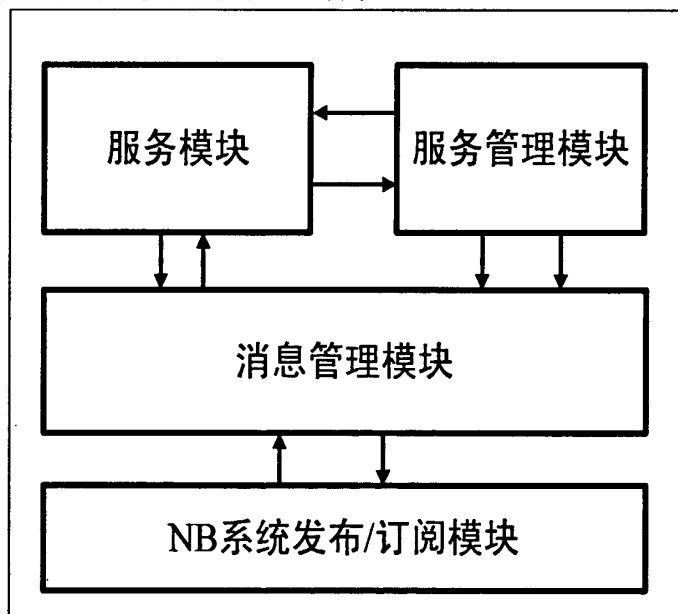


图 3.1 会话服务器模块图

图 3.1 不仅体现了会话服务器模块化的抽象,也展示了其系统层次的抽象。会话服务器的业务流程也是按照从 NB 系统发布/订阅模块→消息管理模块→服务模块或者是从服务模块→消息管理模块→NB 系统发布/订阅模块来完成的。通过这些模块的协调工作实现了会话服务器的所有功能。关于各模块的具体功能及其设计实现将在后续章节作详细介绍。

### 3.3.2 服务模块的设计与实现

服务模块的功能是实现会话服务器需要支持的各项服务，包括注册服务、登录/退出服务，创建/结束会话服务、邀请服务、查询服务、发言权控制服务以及用户状态维护服务。

由于同一时刻会有多个用户请求同一项服务，为了达到这些任务的并发进行，服务器端通常通过线程来实现。基于功能可扩展性的思考，在服务模块设计了抽象基类 `Service`，由其完成所有类型的服务都需要支持的一些初始化及与服务管理模块的配合工作。`Service` 类继承自 Java 语言的线程类 `Thread`，并将 `run` 方法定义为抽象方法。服务模块提供的所有类型服务的具体实现类都继承自 `Service` 基类，并通过覆盖 `Service` 类的 `run()` 方法具体定义各自的处理过程。

`Service` 类与其服务子类的类关系图如图 3.2 所示：

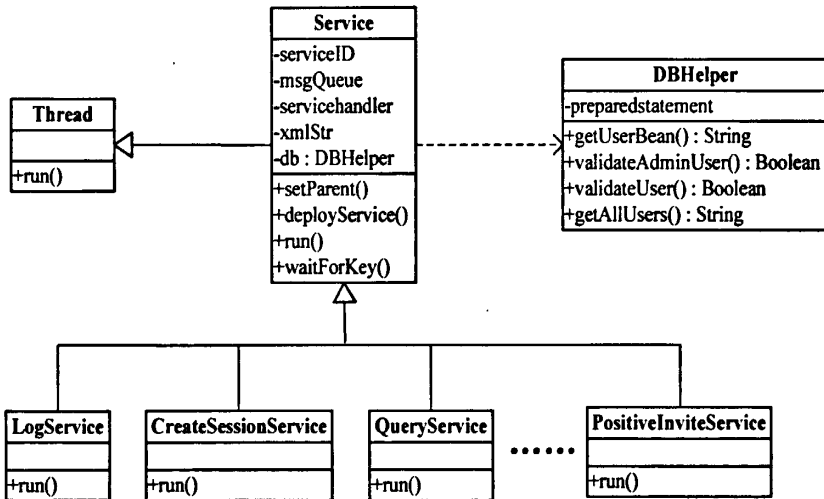


图 3.2 服务模块类关系图

从图 3.2 中可以看出，`LogService`、`RegisterService`、`CreateSessionService`、`QueryService`、`FloorControlService`、`HeartBeatService` 以及 `PositiveInviteService`，这七个服务子类均继承自基类 `Service`，而 `Service` 类继承于线程类 `Thread`，并依赖数据库处理类 `DBHelper` 实现数据库的查询。基于这种结构，可以很容易扩展其他服务。

所有继承于 `Service` 基类的服务子类只需要在 `run()` 函数中实现自己的具体服务处理细节，就可以被集成到整个服务模块之中，由服务控制模块进行管理。

下面对服务模块提供的各项服务进行具体设计实现。

#### 1. 注册服务

提供用户注册功能。XTCCP 协议中注册请求消息的具体元素包括 `UserId`（用户名），`Password`（密码），`UserArea`（用户所在区域）三项。其中，用户所在区域用来区分不同区域的用户，比如两个不同的公司。不同区域的用户不能加入同一个会话。注册服务实例从消息管理模块中取出消息体，进行分析，符合要求的情况

下在数据库用户列表中创建与其对应的新用户信息，用于验证等功能。

RegisterService 作为该项服务的具体实现类，在从消息管理模块取出消息体进行 XML 文件分析之后，遵循 XTCCP 协议进行解析，得到 UserId、Password、UserArea 三项的值，然后通过数据库连接类 DBHelper 连接数据库，查询是否有此用户名存在，存在则返回注册错误信息，否则将在用户列表中创建基于此 UserId 的数据项，存入密码和用户域名，最后返回注册成功信息。

## 2. 登录/退出服务

该服务处理用户的登录/退出请求，并根据处理结果进行组内状态更新。具体工作流程如图 3.3 所示：

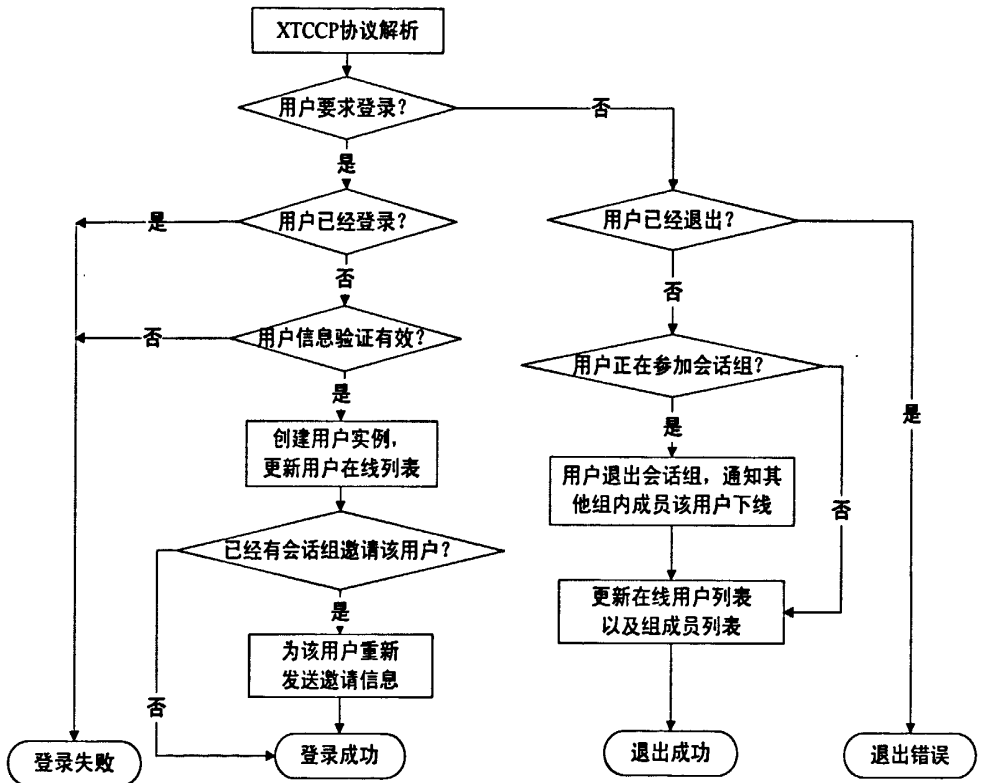


图 3.3 系统登录/退出服务流程图

根据图 3.3，登录/退出服务所需要完成的工作包括：用户请求解析、用户信息验证、会话组状态更新、用户列表更新等。

LogService 作为该项服务的具体实现类，在从消息管理模块取出消息体进行 XML 文件分析之后，遵循 XTCCP 协议进行解析，根据消息根标签为 LogIn 还是 LogOut 判断用户要求登录还是退出。然后，提取用户的 UserId 和 Password 通过 DBHelper 类进行验证。验证成功的用户，如果是登录请求，则为该用户创建一个与之对应的集群通信用户类 TrunkingClient 对象，并将其加入在线用户列表 online\_Users。然后从会话组列表 sessions\_table 中取出与该用户域对应的会话组，逐个判断是否已经有会话在用户登录之前对其进行了邀请。如果有，则为该用户生

成邀请事件 InviteEvent 并传给消息管理模块处理。没有, 则直接反馈登录结果。验证成功的用户, 如果是退出请求, 则首先在已下线用户列表 offline\_Users 中查询该用户是否已经退出。未退出的用户, 从 online\_Users 列表中取出与 UserId 对应的 TrunkingClient 对象, 通过该对象的 isActive() 方法判断用户是否正在参加其他会话组。如果是, 则需要从该会话中清除此用户信息, 并发送用户退出事件 LogEvent 通知组内其他成员该用户的退出。最后从 online\_Users 列表删除与 UserId 对应的 TrunkingClient 对象, 并将 UserId 加入 offline\_Users 列表, 反馈退出结果。

### 3. 创建/结束会话服务

拥有主席身份的用户通过该服务创建或者结束一个会话。该服务首先验证用户的身份和对应名称的会话是否存在。创建会话的情况下, 如果会话不存在, 则直接创建, 刷新会话列表, 然后反馈回复信息给请求用户。而结束会话的情况下, 首先要通知会话中成员关于会话的结束的信息, 然后删除会话, 刷新会话服务器的会话列表, 最后反馈回复信息给请求用户。

该项服务工作流程图如图 3.4 所示:

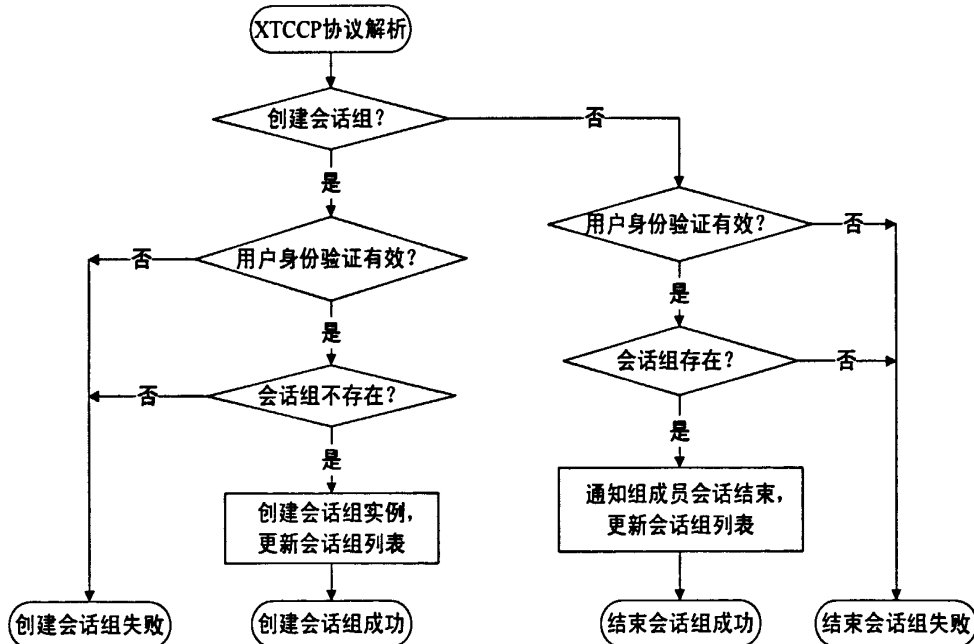


图 3.4 系统创建/结束会话服务流程图

CreateSessionService 作为该项服务的具体实现类, 在从消息管理模块取出消息体进行 XML 文件分析之后, 遵循 XTCCP 协议进行解析, 得到用户信息 UserId 和需要创建的会话组名称 sessionId。然后, 进行用户身份验证以及会话组是否存在判断。如果是创建会话组请求且会话组不存在, 则创建对应 sessionId 的集群通信会话类 TrunkingSession 对象, 更新会话组列表 sessions\_table 列表, 反馈创建结果。如果是结束会话组且会话组已存在, 则从 sessions\_table 中取出对应 sessionId 的 TrunkingSession 对象, 根据其 getOnlineUsers() 方法获得所有在线组成员名单, 通

知这些用户会话的结束，发送 SessionEvent，然后从 sessions\_table 列表中清除该 sessionId 对应的会话组信息，最后反馈结束会话请求结果。

#### 4. 邀请服务

会话主席可以在一次组会话开始的时候，同时邀请一个或者多人参加。被邀请用户可以在线也可以不在线。对不在线的用户，当其下次登录的时候，会话服务器会自动发送邀请信息给用户。

邀请服务收到客户请求并身份验证成功之后，首先判断被邀请用户是否在线，为在线的被邀请用户发送邀请事件 InviteEvent，并将不在线的用户存贮到会话的下线用户列表中，然后将结果反馈给请求用户。如果被邀请用户接受了邀请，邀请服务则为该会话中的所有用户发送 LogEvent，通知其新用户的加入。对于处于下线状态的用户，在其下次登录的时候，登录服务会为其发送邀请信息。

该项服务工作流程图如图 3.5 所示：

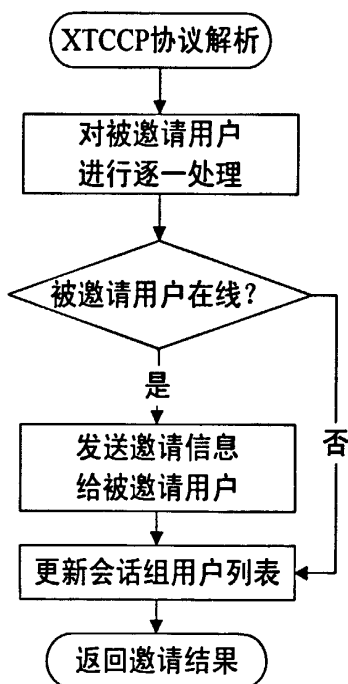


图 3.5 系统邀请服务流程图

PositiveInviteService 作为该项服务的具体实现类，在从消息管理模块取出消息体进行 XML 文件分析之后，遵循 XTCCP 协议进行解析，得到邀请用户的 UserId，会话组名称的 sessionId 以及被邀请用户的名单。在主动邀请用户的身份验证成功之后，通过在线用户列表 online\_Users 对被邀请名单中的所有用户进行逐一判断，如果被邀请用户已经登录，则为该用户生成邀请信息交给信息管理模块处理；如果被邀请用户未登录，则将其名称加入 sessionId 对应的 TrunkingSession 对象的 offline\_Users 列表中。最后根据结果反馈信息给用户。

#### 5. 发言权控制服务



集群通信采用半双工模式,即每个会话组同一时刻只能有一个人发言或者发送视频数据。发言权控制服务的功能就是控制每个会话的实时发言状态。

二进制发言权控制协议 BFCP (Binary Floor Control Protocol) 是 Internet 工程任务组 IETF (Internet Engineering Task Force) 组织在 RFC4582 文档中公布的标准协议,它为不同的多方网络协作系统之间定义了一整套通用的发言权控制通信方法、规范流程以及具体的信令格式<sup>[20]</sup>。由于该协议采用二进制序列进行信令描述,而文中设计的会话控制协议是基于 XML 语言描述的字符型协议,因此,本文在参考 BFCP 协议的基础上,将发言权控制的基本指令以 XML 描述的形式整合到会话控制协议 XTCCP 中,即以字符形式完成 BFCP 协议的基本功能。

发言权控制协议中的信令信息总结如表 3.1 所示:

名称	功能描述	信令发送方
FloorRequest	用户申请发言权请求	客户端
FloorRelease	用户释放发言权请求	客户端
FloorGrant	用户申请发言权成功	服务器
FloorIdle	用户可以申请发言权通知	服务器
FloorTaken	发言权已经被其他用户申请	服务器
FloorDeny	用户申请发言权被拒绝	服务器

表 3.1 发言权控制协议指令消息表

该服务一方面处理用户的发言权请求 (FloorRequest) 和发言权释放 (FloorRelease),将处理结果反馈给请求用户。另一方面,在预设置时间内不断轮询各个会话的发言权状态。当会话处于空闲时,广播该信息给此会话内所有成员。由于同一时刻可能有多个人申请发言,因此将时间作为判断标识。利用 NB 系统提供的时间服务,由其在 NB 网络传输接口层对收到的事件按时间戳进行排序。

FloorControlService 作为该项服务的具体实现类,订阅发言权控制主题,在从消息管理模块取出消息体进行 XML 文件分析之后,遵循 XTCCP 协议进行解析,取出用户的 UserId,会话组名称 sessionId 以及指令信息,并根据上表进行处理。如果是 FloorRequest 请求,则从会话组列表 sessions\_table 中取出对应 sessionId 的集群通信会话类 TrunkingSession 对象,通过该对象的 getSessionStatus()方法判断会话是否空闲,然后反馈 FloorGrant 或者 FloorDeny 结果给请求用户。如果是 FloorRelease 请求,则在判断会话状态之后,更新会话状态,然后反馈结果给用户。同时, FloorControlService 作为服务线程,不断对 sessions\_table 中的各个会话对象进行轮询,当会话状态为空闲时,则为会话组成员发送 FloorIdle 信息。

## 6. 用户状态维护服务

在本文所设计的集群通信系统中,每个用户都是通过与本地代理节点进行连

接下来完成与会话服务器的通信。这种异步通信机制使得会话服务器无法像以前的客户直连服务器一样通过 TCP 等连接的状态及相应机制来判断用户是否掉线从而进行状态更新，因此必须加入另外一种通知机制，来弥补这一不足。心跳包是实现这一功能比较简单且不浪费带宽和过多资源的方法。

心跳包就是客户端定时发送简单的信息给服务器，告知其依然在线的状态。代码实现就是每隔几分钟发送一个固定信息给服务器，服务器收到后回复或者只是更新维护的状态表。如果服务端在一段时间内没有收到客户端发送的信息，则可以判断用户已经断开或者掉线。

因此，在本文设计的系统中，客户端只要定时发送心跳包到会话服务器已经订阅的主题上，会话服务器就可以收到这些心跳信息，并根据这些信息更新所维护的用户状态以及会话状态等。当已经登录的用户几次（目前设定为 3 次）都没有发送心跳信息，用户状态维护服务则认为用户端超时，就会更新在线用户列表。如果此用户在某实时会话中，则更新会话组用户列表，发送用户超时通知给相应域内所有用户。

HeartBeatService 作为该项服务的具体实现类，通过 java 包提供的 Timer 和 TimerTask 类为每个登录成功的用户创建一个计时器。当计时器未到时，Timer 和 TimerTask 已经从消息管理模块收到该用户的心跳信息，它会刷新计时器重新计时。如果计时器已经到时，用户无心跳信息到达，则会启动 TimerTask 定义的任务，更新用户状态，并为其他域内用户发送用户超时事件 TimeOutEvent。

### 7. 用户查询服务

登录的用户可以查询域内所有用户的在线/下线状态，而已经加入某会话的用户，则可以更新会话的当前状态。当某个更新消息因为网络问题没有及时到达用户的情况下，用户查询服务可以提供相应信息的更新，使用户维持会话系统的最新状态。

QueryService 作为该项服务的具体实现类，在从消息管理模块取出消息体进行 XML 文件分析之后，遵循 XTCCP 协议进行解析，根据消息体的根标签是 UserInfoQuery 还是 SessionInfoQuery 得出用户需要查询的是用户信息还是会话信息。如果查询会话信息，则从会话列表 sessions\_table 中取出对应会话名称 sessionId 的 TrunkingSession 对象，通过它的 getSessionInfo()方法得到该会话的所有信息反馈 SessionInfoEvent 给用户。如果查询用户信息，则取出对应用户的信息，反馈 UserInfoEvent 给用户。

### 3.3.3 服务管理模块的设计与实现

服务管理模块的功能是对服务模块的多个服务线程进行有效的管理和控制。

考虑到会话服务器的用户请求数量大、处理时间短的特性, 服务管理模块采取的模式是线程池模式。

线程池是一种多线程处理形式, 是预先创建线程的一种技术。线程池在还没有任务到来之前, 创建一定数量 ( $N1$ ) 的线程, 放入空闲队列中。这些线程都是处于阻塞状态, 不消耗 CPU, 但占用较小的内存空间。当任务到来后, 缓冲池选择一个空闲线程, 把任务传入此线程中运行。当  $N1$  个线程都在处理任务后, 缓冲池自动创建一定数量的新线程, 用于处理更多的任务。当系统比较空闲时, 大部分线程都一直处于暂停状态, 线程池自动销毁一部分线程, 回收系统资源。这样既减少了频繁创建、销毁线程所需要的系统资源, 也优化了系统的性能<sup>[21]</sup>。

服务管理模块通过线程池的方式对服务模块的服务线程进行控制管理。它首先负责产生多个服务线程, 在没有用户请求到来的情况下, 根据线程锁机制, 这些服务线程会相继进入阻塞状态。服务控制模块的管理线程不断访问消息管理模块, 如果有用户请求到来, 则唤醒相应数量的服务线程, 由其到消息管理模块取出消息数据, 进行具体处理。当处理完毕, 服务线程又进入阻塞状态。

ServiceHandler 是服务控制模块的管理线程类。每个 ServiceHandler 实例控制一种类型的服务, 对应一个服务类型号 ServiceID。它首先为对应类型的服务创建多个子服务线程实例, 然后通过线程锁机制对这些服务线程进行控制。而服务模块的实现中, 所有的服务子类都继承自基类 Service, 通过其提供的 waitForKey() 方法进行线程锁机制运行和消息管理模块的消息查询。因此, ServiceHandler 和 Service 类的线程锁机制的实现是服务控制模块的服务线程池机制的基础。

为了更好地说明 ServiceHandler 与 Service 各子类实例之间通过线程锁机制协调工作的过程, 下面以对应一个 ServiceID 的 ServiceHandler 实例和两个 Service 服务子类实例从服务被部署开始到线程不断被锁和被唤醒的过程来说明会话服务器如何提供服务线程池机制的实现, 流程如图 3.6 所示。

从图 3.6 中可以看出, ServiceHandler 作为 Service 子类实例的管理者 (通过线程锁机制), 不断查询消息管理模块的情况, 并在有对应 ServiceID 类型的消息存在的情况下唤醒一个被锁的服务子类线程, 由其从消息管理模块中取得消息实体进行处理。当 ServiceHandler 查询到消息管理模块中此类型消息的数目大于 1, 则 ServiceHandler 线程进入阻塞状态。被唤醒的服务线程在取到消息后, 唤醒下一个被锁服务线程, 由其继续处理消息管理模块中的该类型消息, 而自己在按照 XTCCP 协议处理完从消息管理模块取出的消息体后, 进入阻塞状态。如此反复, 直到消息管理模块中已经没有该类型消息存在, 唤醒链停止。当被唤醒的服务子类线程查询到服务管理模块中该类型服务的数目小于 1, 则由其唤醒 ServiceHandler 线程, 由 ServiceHandler 管理的新一轮查询开始。

按照这种机制, 消息管理模块中所有该类型的消息都可以得到处理, 而且,

根据线程的并发机制，多个该类型的消息可以通过并发的形式得到处理，从而完成处理同一时刻发起的多个用户请求。在消息管理模块中没有用户消息到来的情况下，各类型的多个服务子类线程相继进入阻塞状态，不再占有处理器资源。这样，通过线程锁机制的有效使用，ServiceHandler 完成了对多个服务线程的管理，真正起到了线程池管理者的作用。

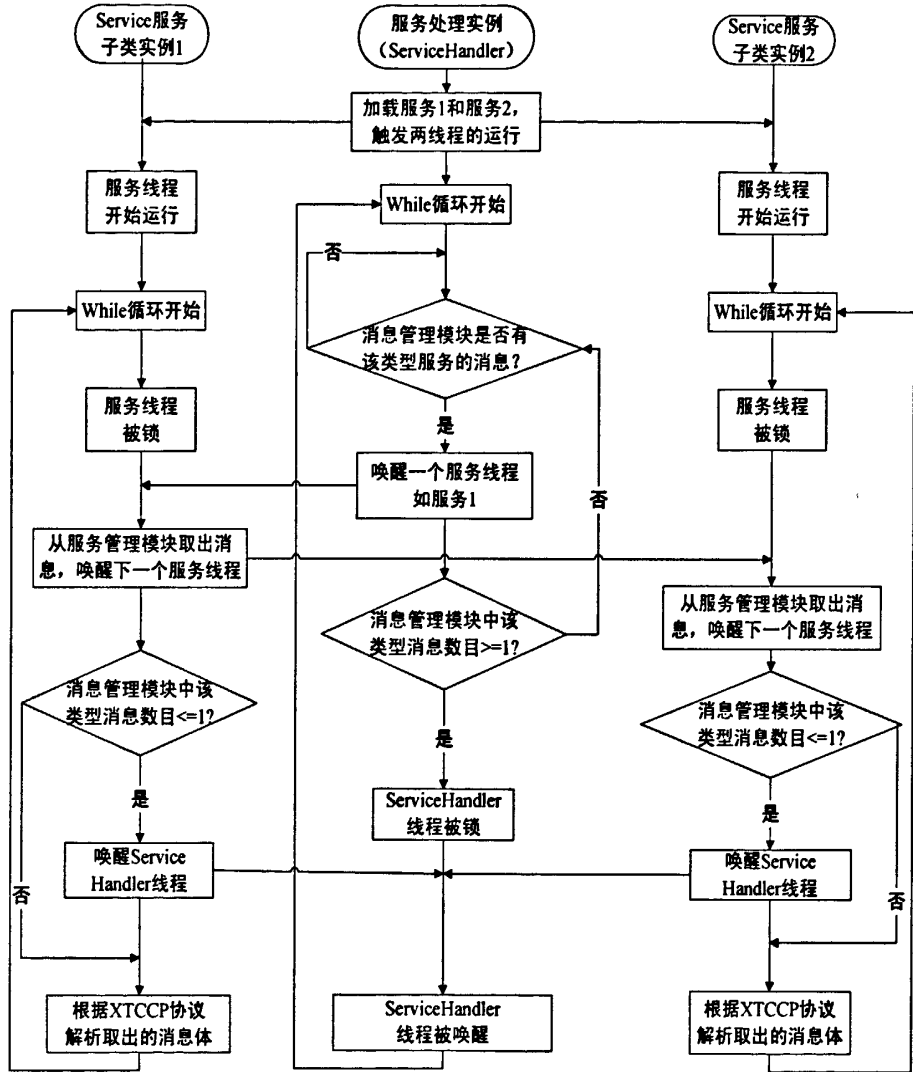


图 3.6 ServiceHandler 与 Service 子类实例线程锁机制流程图

### 3.3.4 消息管理模块的设计与实现

消息管理模块存储并管理着会话服务器中所有的消息，包括用户的请求信息以及服务器的反馈信息等。

由于服务模块中各类具体服务是以独立线程的形式实现，因此，消息管理模块需要能够有效地为不同类型的服务提供特定的消息，即为每种类型服务对应的消息定义为一种服务序列号 ServiceID。同时，为了服务管理线程的查询方便，为

每个消息产生一个唯一的消息号 key 与之对应。这样，当服务管理线程查询到有对应 ServiceID 的 key 存在时，具体的服务实例就可以对应 key 取到消息体进行处理。

消息队列实现类 MessageQueues 共包含了两个由同步之后的 HashMap 实现的队列：keys 和 queue。按照设计，每个消息对应的 Id 称为 key，通过系统时间的整数化来实现，以保证唯一性。而每种服务对应的 Id 称为 ServiceID，由代表该类型的字符串对象实现。

队列 keys 的结构如图 3.7 所示：

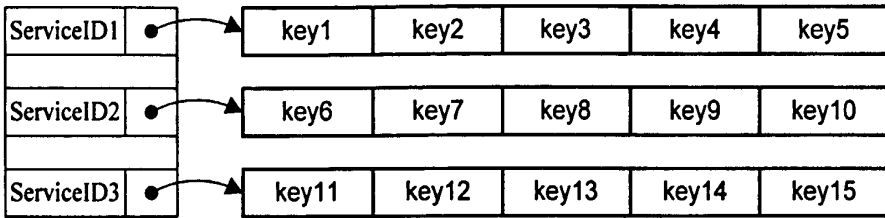


图 3.7 队列 keys 结构图

该 HashMap 中，每种服务 ServiceID 对应一个 KeyList 列表，包含着放入 MessageQueues 的所有该服务类型的消息的 key。

队列 queue 的结构如图 3.8 所示：

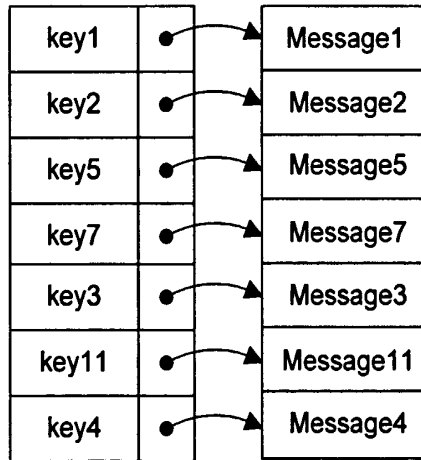


图 3.8 队列 queue 结构图

该 HashMap 为消息的真正存储区，并与其唯一的序列号 key 对应，可通过此 key 获得消息体。

通过 ServiceID 与 key 的对应机制，服务模块和服务管理模块可以有效地进行信息的查询和读取。

### 3.3.5 NB 系统发布/订阅模块的设计与实现

NB 系统发布/订阅模块的功能是从 NB 网络的代理节点接收已订阅的消息和向代理节点发送需要发布的消息。这些消息即是会话服务器和客户端交互的 XTCCP

协议描述的信令信息。

NB 系统发布/订阅模块是会话服务器与 NB 网络之间的通道,它从消息管理模块取出需要发布的消息,封装成 NBEvent 事件,并通过与 NB 代理节点之间建立的信令通道发送出去。同时,它又将从此通道上收到的 NBEvent 进行解析,取出消息负载交给消息管理模块进行存储、管理。

会话服务器在初始化的时候,必须要完成网络的初始化,即和 NB 网络代理节点建立特定链接。

NB 系统发布/订阅模块的实现类图如图 3.9 所示:

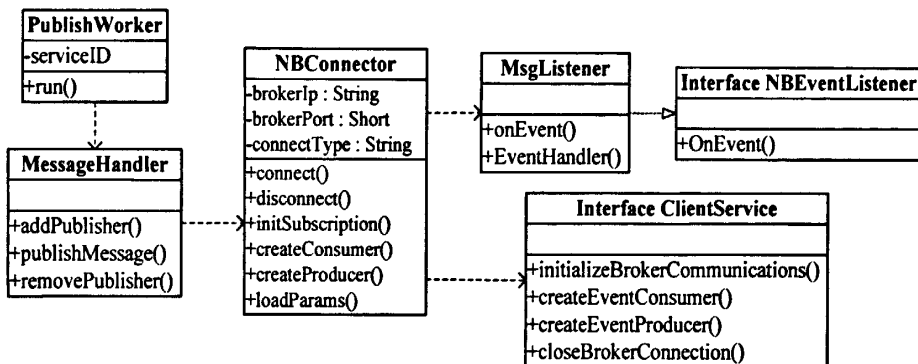


图 3.9 NB 系统发布/订阅模块实现类图

NBConnector 是该模块实现的核心类,通过 connect()和 initSubscription()方法完成会话服务器的网络和订阅的初始化。为了能够和 NB 网络代理节点通信,NBConnector 需要用到 NB 系统提供的客户端接口 ClientService,通过其实现类完成数据的底层封装和传送。MsgListener 是 NBConnector 在订阅初始化时需要创建的侦听类,实现对消息进行初步分析并传输到消息管理模块的操作。该类实现了 NB 系统提供的接口类 NBEventListener。

PublishWorker 作为发布信息线程类,不断从消息处理模块取出对应自己 ServiceID 的信息,交给消息处理类 MessageHandler 的 publishMessage()方法发布出去。而 MessageHandler 的 addPublisher()方法需要 NBConnector 来创建发布者类 Producer。这正体现了上图中的类依赖关系。

通过 NBConnector、MsgListener、MessageHandler 以及 PublishWorker 之间的协作,NB 系统发布/订阅模块实现了信息的双向传输。

### 3.4 PC 机客户端的设计与实现

#### 3.4.1 PC 机客户端的模块化设计

基于 IP 网络的多媒体集群通信系统中客户端的功能是完成用户与会话服务器之间的信令协作,发送/接收音视频数据,并以图形界面的形式为用户展现处理结果,播放实时多媒体数据。

基于 IP 网络的多媒体集群通信系统包括两种类型的客户端：PC 机客户端和手机客户端。手机客户端由项目组内其他研发人员进行开发，因此，本文只详细介绍 PC 机客户端的设计与实现。

根据抽象化、模块化、可扩展性原则，将 PC 机客户端按照功能设计为四大模块：用户界面模块、信令处理模块、多媒体处理模块以及 NB 系统发布/订阅模块。各模块之间的协作关系如图 3.10 所示：

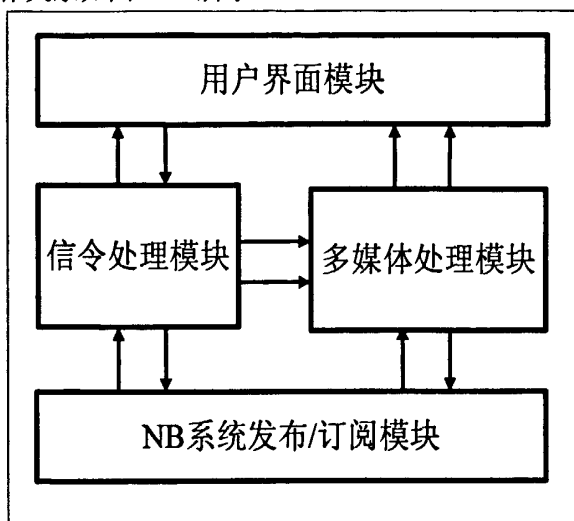


图 3.10 PC 机客户端模块图

图 3.10 中箭头的方向表示消息的走向。PC 机客户端共包括两种消息类型：信令消息和多媒体消息。因此，信令处理模块和多媒体处理模块是整个软件的核心。信令处理模块与用户界面模块和信令处理模块与 NB 系统发布/订阅模块之间的通信都是双向的，目的是完成用户请求的整理和发送以及服务器反馈信息的整理和用户呈现。同时，信令管理模块通过指令分析控制多媒体处理模块发送捕获的音视频数据或者接收其他用户发送的音视频数据，并将解码、播放的结果呈现给用户界面模块。

### 3.4.2 用户界面模块的设计与实现

用户界面模块是通过 NetBeans IDE 提供的基于 Swing 应用程序框架的 GUI 设计器实现的。

Swing 应用程序框架包括生命周期管理、Session 管理、Action、任务、资源五部分，基本涵盖了开发图形界面程序所需要的内容。开发人员可以在此基础上增加自己的应用、后台程序代码以及图片等资源，而不需要过多地将时间消磨在繁琐的界面框架设计上<sup>[22]</sup>。

用户界面首先应该包括登录/退出、联系人、创建/结束会话、设置、会话信息以及申请发言几个按钮，每个按钮触发相应的后台操作，等待会话服务器的反馈

结果并进行处理、呈现。登录后的用户，首先可以通过联系人列表了解自己所在区域内所有用户的在线/下线情况。然后，拥有权限的用户可以创建会话，邀请需要的用户参加会话。通过会话信息按钮用户也可以看到自己所参加的会话的实时情况，包括创建者，会话状态，发言人等。

PC 客户端的图形界面和联系人列表如图 3.11 所示（用户已经登录情况下）：

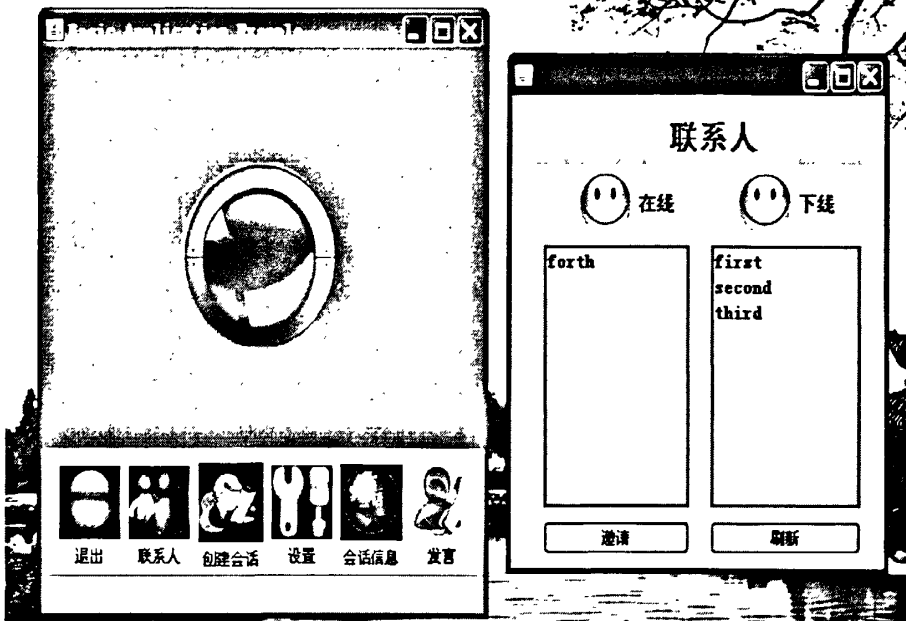


图 3.11 PC 客户端的图形界面及联系人列表

图 3.12 中 A 为创建会话框图, B 为用户被邀请通知框图, C 为会话信息框图, D 为会话服务器发布的会话空闲状态框图。会话服务器根据用户请求不断更新系统中维护的会话的状态, 并将相应信息通过 NB 系统发送到各个用户, 通过图形界面显示出来。

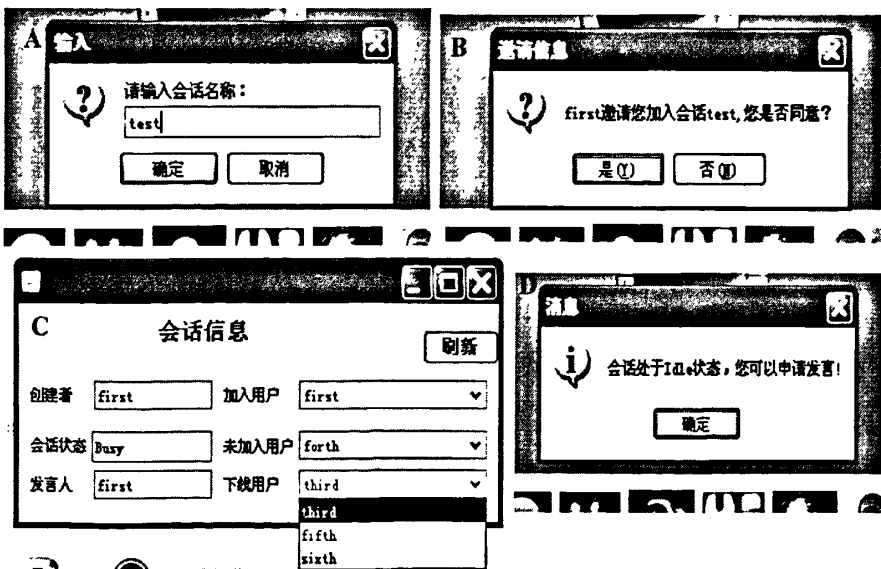


图 3.12 PC 客户端会话过程界面



### 3.4.3 信令处理模块的设计与实现

信令处理模块的功能主要分为三个部分：

1. 整理由用户界面模块得到的用户请求，生成 XTCCP 协议消息，交由发布/订阅模块封装、发送；
2. 从发布/订阅模块接收 XTCCP 协议消息，解析内容，整理结果交由用户界面模块展现；
3. 根据用户的发言权状态，控制多媒体处理模块的发送或者接收，以及音视频数据的编解码、播放等。

由于本系统设计中 XTCCP 协议采用 XML 语言方式进行描述，因此，需要为每个用户请求生成相应的 XML 信息，这些信息将分别被会话服务器中服务模块的各项服务解析、处理。同时，会话服务器的反馈信息包括请求结果和事件通知两部分，需要针对这两类信息进行不同处理。

该模块实现类图如图 3.13 所示：

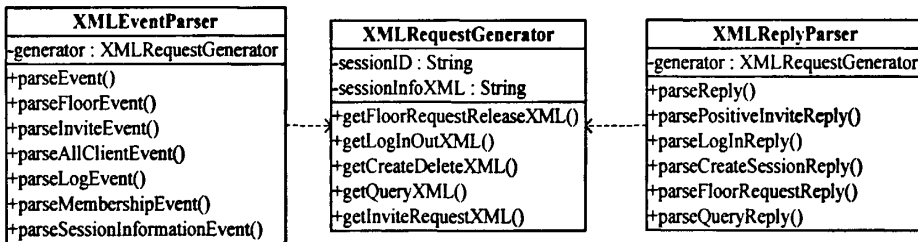


图 3.13 信令处理模块实现类图

XMLRequestGenerator 类是 PC 客户端所有请求信息的生成类，它根据用户提供的用户名 UserId，会话组名称 sessionId 等调用特定方法，生成与该要求类型对应的 XML 格式的请求信息。如 getFloorRequest()方法用于生成用户的发言权请求，getQueryXML()方法用于生成用户的查询请求等。

请求结果解析类 XMLReplyParser 和事件解析类 XMLEventParser 分别对用户收到的请求结果 (Reply) 和事件通知 (Event) 进行解析。XMLReplyParser 首先调用其 parseReply()方法解析出请求结果类型，然后交给对应类型的方法进行处理，如 parseLogInReply()解析用户登录/退出请求的结果，pareQueryReply()方法解析用户查询的结果。同样，XMLEventParser 首先调用其 parseEvent()方法解析出时间通知类型，然后交给对应类型的方法处理，如 parseFloorEvent()方法解析发言权事件的具体内容，parseLogEvent()方法解析其他用户的登录/退出事件。

根据解析结果，XMLReplyParser 和 XMLEventParser 判断是否需要再次调用 XMLRequestGenerator 产生新的请求信息，同时将处理结果通过用户界面模块展现出来。当请求结果和事件通知涉及到发言权问题，会根据信息的类型对多媒体控制模块进行不同控制。

### 3.4.4 多媒体处理模块的设计与实现

多媒体处理模块提供了 PC 客户端所有的音视频处理功能, 这些功能的实现主要是通过 Java 媒体框架 JMF (Java Media Framework) 来实现的。

JMF 是由 SUN、IBM、Intel 等公司共同开发, 针对基于时间的多媒体进行控制和处理的统一框架。它不仅支持本地媒体文件和媒体设备的采集、处理、存储、播放等功能, 还支持基于网络的压缩媒体流的同步、控制、处理和播放。通过对 JMF 的集成, 就可以实现网络传输音视频的功能。此外, JMF 2.1.1e 版本引入了插件结构, 使程序员能够简单快捷地对其框架结构进行扩展。

JMF 框架如图 3.14 所示。最上层为应用处理层, 用于用户开发各种 Java 应用。第二层为 JMF 展现和处理 API 层, 为用户的调用提供统一接口, 这些接口包括了 JMF 能够提供的所有功能。第三层为 JMF 的插件层, 它拥有一套标准的接口, 对于不同的功能, 这些接口将调用不同的底层实现, 如编码插件、解复用插件等。同时, 由于这种插件机制, 用户可以根据需要实现标准接口, 进而集成自己的功能。

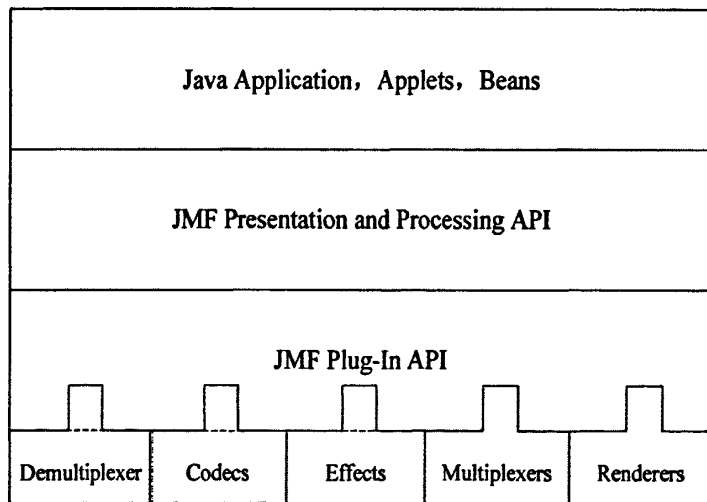


图 3.14 JMF 框架图

JMF 展现和处理 API 包括 JMF API 和 JMF RTP API 两个部分。前者的主要功能是捕捉、处理、存储和播放媒体; 后者主要是在网络上传输和接收媒体流。JMF 使用 RTPSessionManager (RTP 会话管理器) 来建立和管理用于传输实时流媒体的 RTP 会话。会话管理器跟踪会话的参与者和被传输的流, 从本地参与者的角度维护会话状态<sup>[23]</sup>。

JMF 对 RTP 的支持使它更加适用于流媒体实时传输的应用, 包括我们的集群通信系统。只要通过对 JMF 传输的 RTP 数据进行关于 NB 系统的 RTPEvent 封装, 就可以将 JMF 捕获的实时多媒体数据在 NB 网络内分发。为了使 JMF 能够与 NB 系统的传输模式相匹配, 需要考虑二者的接口问题。通过实现 JMF 的 RTP 连接器

RTPConnector 接口，可以把 NB 网络中收到音视频数据交给 JMF 进行解码，播放处理，也可以把 JMF 编码打包之后的音视频数据包交给 NB 传输接口进行网络分发。

多媒体处理模块的类实现图如图 3.15 所示：

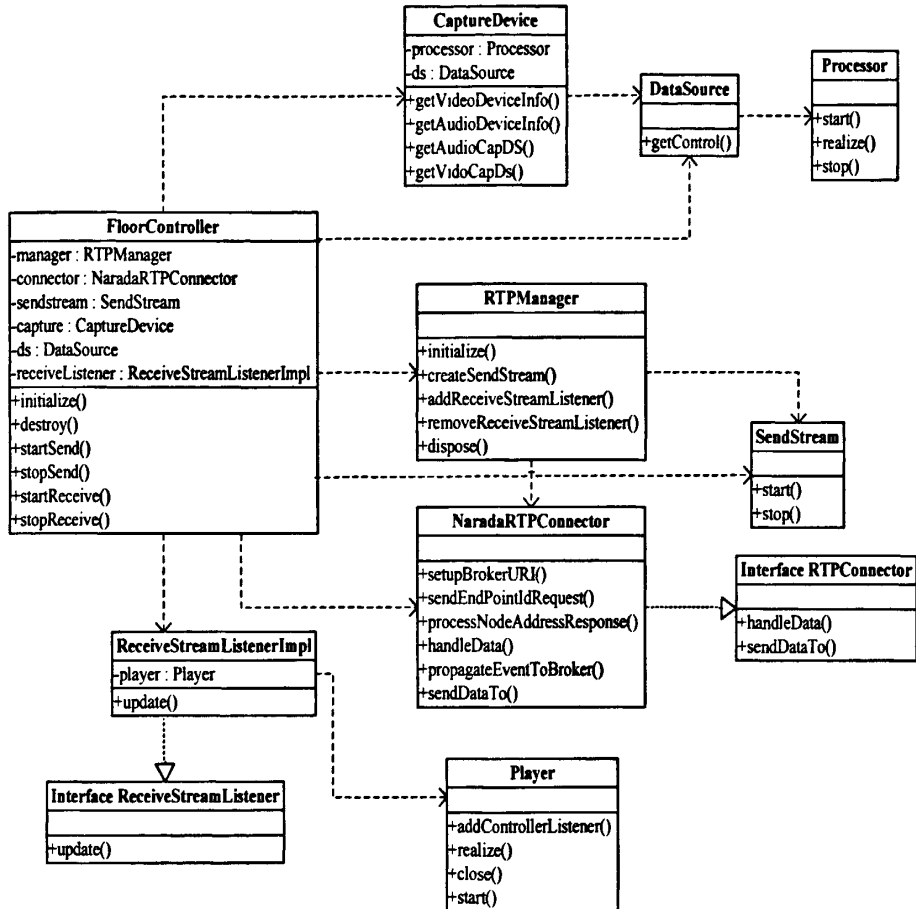


图 3.15 多媒体处理模块实现类图

FloorController 作为多媒体处理模块的管理类，负责接收指令处理模块的指令信息并触发相应处理，如开始发送 startSend()、开始接收 startReceive() 以及停止发送 stopSend() 和停止接收 stopReceive()。类 RTPManager (RTP 管理者)、DataSource (数据源)、Processor (处理器)、SendStream (发送流)、Player (播放器) 以及接口 RTPConnector (RTP 连接器)、ReceiveStreamListener (接收流侦听器) 都是 JMF 框架提供的 API: RTPManager 管理整个 RTP 会话的状态信息，DataSource 作为音视频数据源通过 Processor 进行管理，SendStream 控制发送流的开始、暂停和结束，Player 负责音视频的播放。

通过 NaradaRTPConnector 类 (RTPConnector 实现类) 实现 JMF 的 RTPConnector 接口，可以把由 JMF 捕获的数据以 NB 系统能够识别的方式发送出去，同时将从 NB 系统接收到的数据进行解码、播放。通过 ReceiveStreamListener Impl 类 (接收流侦听器实现类) 实现 JMF 的 ReceiveStreamListener 接口，当有其

他用户发送数据并由 NB 发布/订阅模块传递过来时, 就可以通过实现的 update() 方法对接收的数据进行各种处理。

### 3.4.5 NB 系统发布/订阅模块的设计与实现

PC 机客户端的 NB 系统发布/订阅模块的功能是对用户要发布的信息进行事件封装并发送, 对用户需要订阅的信息进行订阅、接收和解析。

由于不需要像会话服务器一样发布或者订阅类型很多且数量很大的信息, PC 机的客户端的 NB 系统发布/订阅模块的设计与实现要简化很多。它从信令处理模块取出需要发布的消息, 封装成 NBEvent 事件, 通过与 NB 代理节点之间建立的信令通道发送出去。同时, 它又将从此通道上收到的 NBEvent 进行解析, 取出消息负载交给消息管理模块进行处理。

PC 客户端在初始化的时候, 必须要完成网络的初始化, 即和 NB 网络代理节点建立特定链接。

PC 机客户端的 NB 系统发布/订阅模块的实现类图如图 3.16 所示:

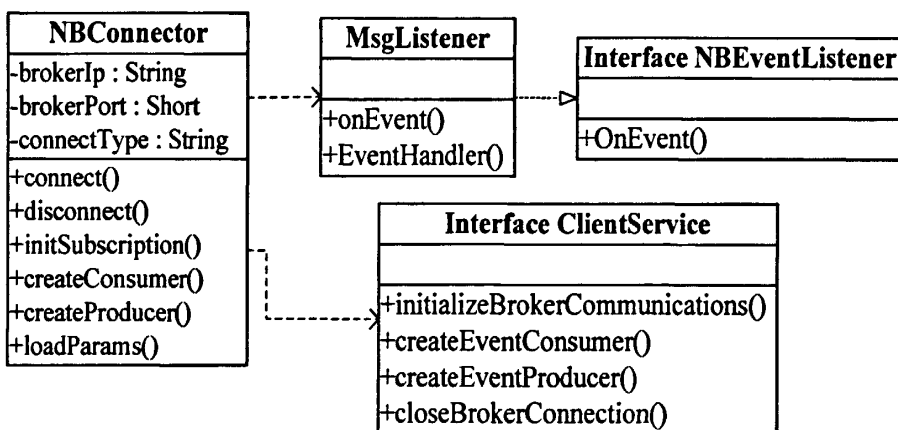


图 3.16 NB 系统发布/订阅模块实现类图

NBConnector 作为该模块的核心类, 封装了 NB 系统提供的底层接口, 如 ClientService (客户端服务接口) 等, 并通过 connect()方法与 NB 网络代理节点建立链接, 初始化客户端订阅。MsgListener (消息侦听类) 作为事件的侦听类实现了 NB 系统提供的 NBEventListener 接口 (NB 系统事件侦听接口), 对收到的事件通过 EventHandler()方法进行进一步处理。

## 3.5 本章小结

本章首先介绍系统的开发平台 NetBeans IDE, 然后根据抽象化、模块化、可扩展原则将会话服务器分为服务模块、服务管理模块、消息管理模块以及 NB 系统发布/订阅模块, PC 机客户端分为用户界面模块、消息处理模块、多媒体处理模块

以及 NB 系统的发布/订阅模块，并对各模块进行详细的功能设计。在功能设计的基础上，本章还给出了各模块的具体实现，包括流程图、类图等。通过本章的设计与实现，基于 IP 网络的集群通信系统完成了最基本的设计实现工作。

## 第四章 系统的测试

本章首先完成应用层组播平台的搭建，并与手机客户端进行功能联调，然后对整个系统进行功能和性能的测试，并对测试结果进行分析总结。

### 4.1 应用层组播平台的搭建

应用层组播平台 NaradaBrokering 是基于 Java 语言开发的系统软件，因此，在搭建该平台之前，需要为系统安装 Java 运行时环境。

应用层组播平台 NaradaBrokering 的搭建分为下面几部分：

#### 1. Java 运行时环境搭建

可以通过下载 Java 的 JDK 安装包或者是直接下载 JRE 安装包（Java Runtime Environment）的方法获得 Java 运行时环境安装包，安装到系统目录下，并在环境变量中增加一项 JAVA\_HOME，值为 JRE 的安装目录。

#### 2. NaradaBrokering 的环境变量设置

在系统的环境变量设置中增加一项名为 NB\_HOME，值为 NaradaBrokering 的存放目录。同时，为系统环境变量 path 增加一项为 %NB\_HOME%\dll 的路径，将 NB 系统运行所需要的动态链接库载入系统环境。

#### 3. NaradaBrokering 的配置文件设置

在 NaradaBrokering 存储目录的 conf 目录下找到 BrokerConfiguration.txt 文件并打开，设置该代理节点的协议端口如下：

- NIOTCPBrokerPort = 3045
- TCPBrokerPort = 5045
- UDPBrokerPort = 3045

同时，设置 AssignedAddress = true，为该节点分配逻辑地址为 NodeAddress = 1,1,1,1，为了支持 RTP 协议，设置 SupportRTP = true。保存文件。

这样就完成了一个 NB 网络代理节点的配置。其他节点的配置同理。当 NB 网络中存在多个代理节点的时候，还需要根据提示命令在节点之间建立不同传输协议类型的链接。

### 4.2 系统与手机客户端的联调

基于 IP 网络的多媒体集群通信系统中的手机客户端软件由项目组的其他研发人员进行开发。选择的手机端系统平台是 Android 系统。

Android 是由 Google 及 30 多家移动技术和无线应用领域领先企业在 2007 年

11 月发布的智能手机平台。它是一个真正意义上的开放性移动设备综合平台，包括操作系统、用户界面、中间件和应用程序，拥有移动电话工作所需的全部软件，同时其开放性保证该平台不存在任何阻碍移动产业创新的专有权障碍<sup>[24]</sup>。

基于 Android 系统的智能手机客户端与 PC 机客户端的设计与实现包括下述几种不同：

### 1. 多媒体处理框架不同

手机端对多媒体的处理采用了 Android 系统提供的多媒体核心 OpenCore 框架，而不是 JMF 框架。它利用 Android SDK 下 media 包中所提供的如 MediaRecorder（媒体记录）、MediaPlayer（媒体播放器）、AudioRecord（音频记录）等类，完成了对音视频的采集、编码、解码、播放、同步等一系列操作<sup>[25]</sup>。

### 2. 图形界面设计机制不同

手机端的图形界面根据 Android 系统的布局规则，使用 XML 标记语言来描述图形界面，而不是使用 Java 的 Swing 框架包进行设计。

### 3. 网络接入方式不同

手机端通过无线相容性认证 WIFI（Wireless Fidelity）技术接入无线局域网，而 PC 机端采用有线局域网进行接入。选择 WIFI 作为手机终端加入集群通信系统的接入手段，不仅可以很好地利用现有网络，节约成本，而且 WIFI 的高速率传输也能很好地满足多媒体传输的要求。

在搭建应用层组播平台 NaradaBrokering 的基础上，PC 机客户端、会话服务器与手机客户端进行了联调，以验证功能的完备性。

手机端用户界面如图 4.1，与 PC 端界面保持风格一致：

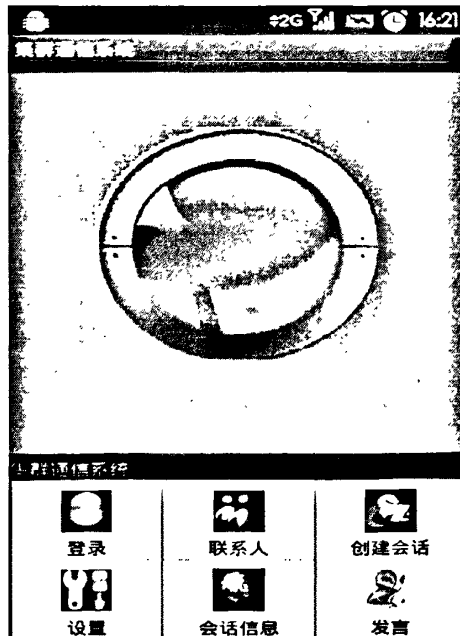


图 4.1 手机客户端图形界面

用户登录后,即可根据其权限创建会话,邀请其他用户加入会话。如图 4.2 所示:

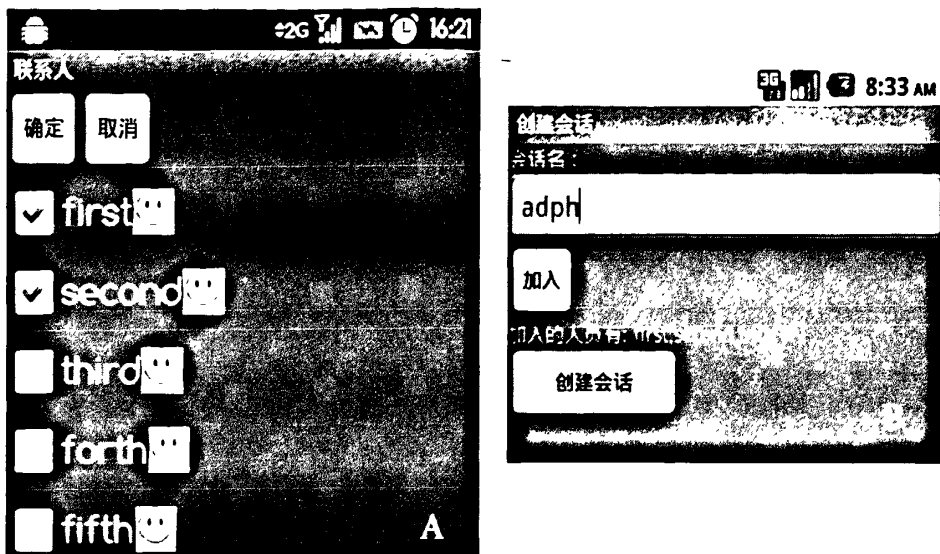


图 4.2 手机端会话创建界面

手机端发言者的界面如图 4.3 所示,用户首先登录,然后创建会话组,邀请其他用户加入会话,进而获得发言权,开始发送多媒体数据。同时,用户手机摄像头捕获的数据也在发言者手机端实时播放。



图 4.3 手机端用户发言界面



### 4.3 系统测试与分析

#### 4.3.1 测试环境

在带宽为 10M 的局域网中，部署了 22 台 PC 机以及 10 部 HTC 智能手机。PC 机的配置为奔腾 E5500 双核 CPU，主频 2.80GHZ，内存 2GB。HTC (HD2) 的配置为 Snapdragon(高通)QSD8250 的 CPU，主频 1024MHz，机身内存为 512MB ROM+448MB RAM。然后，在一台 PC 机上部署 NaradaBrokering 代码，作为 Broker 节点，一台 PC 机上部署会话服务器程序，作为集群通信系统的服务器，其余 PC 机每台部署 20 个 PC 客户端，而每部 HTC 手机部署一个手机客户端程序。

由于系统目前处于设计完成初期，以及网络条件有限，因此选择的环境为局域网环境。目的是首先完成小规模（用户少于 500 人）的性能和功能测试。随着后续工作的展开和完善，系统会在广域网内进行进一步测试。

搭建的测试环境如图 4.4 所示。由一位手机用户作为发言者 (Transmitter)，其他用户作为收听者 (Receiver)。所有会话指令及多媒体数据通过部署的代理节点 (BrokerNode) 进行分发。

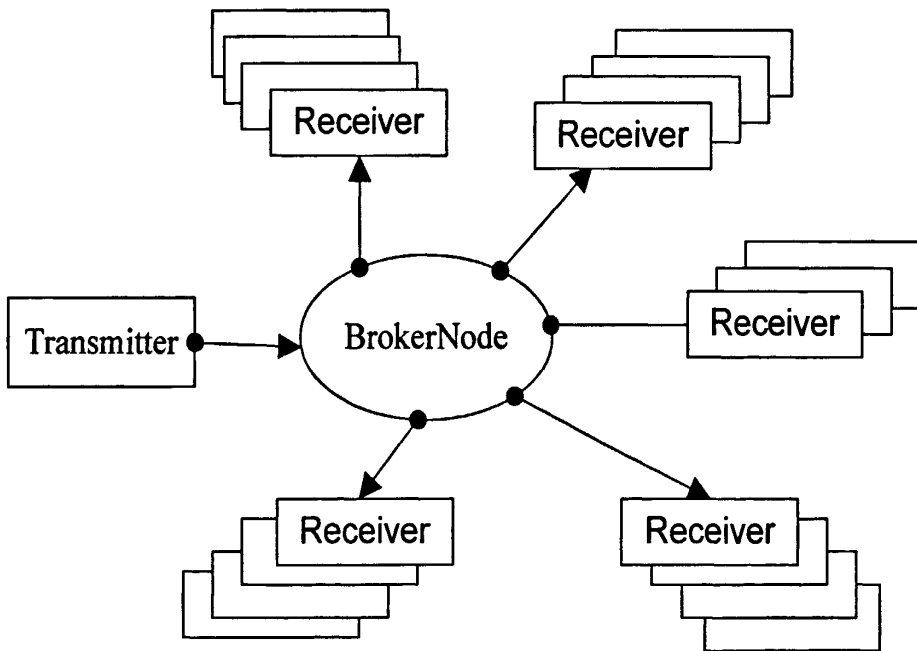


图 4.4 测试环境示意图

由于在局域网内部测试，因此只部署了一个 BrokerNode，由其完成所有会话信息和多媒体信息的分发任务。在这样的测试环境中，BrokerNode 的负载能力和计算能力成为会话时延以及多媒体性能的主要瓶颈。随着用户数目的不断增加，BrokerNode 需要处理的数据不断增多，性能也会不断下降。

系统测试画面如图 4.5 所示。手机用户作为发言者将通过手机获取的音频和

视频数据实时发送给所有参与此会话的收听者。



图 4.5 测试界面

#### 4.3.2 功能测试

功能测试主要针对开发的集群通信系统所提供的所有功能，它们包括：

- 注册功能；
- 登录/退出功能；
- 会话创建/结束功能；
- 会话邀请功能；
- 发言权控制功能；
- 用户状态维护功能；
- 客户端查询功能。

这些功能的实现主要是通过会话服务器所提供的一系列服务，因此也是对服务器端程序的测试。经过测试，服务器端服务线程池以及消息队列的设计能够很好地节约系统资源，最大限度地处理客户端请求。在同时有 50 个用户发送登录请求的情况下，线程锁机制依然能够很好地工作。当在线用户达到 200 个的时候，服务器的处理速度有所下降，但是依然在用户感应可接受范围内。

### 4.3.3 性能测试

性能测试主要针对开发的集群通信系统的会话通信延迟, 音视频传输延迟、抖动。

#### 1. 会话通信延迟:

测试项目	两个用户同时发起请求获得服务器确认时延 (ms)	50 个用户同时发起请求获得服务器确认时延 (ms)
登录	9.5-10.2	25.2-29
创建会话	11.4-13.8	30.8-40.2
邀请加入	12.1-18.7	39.2-50.3

表 4.1 会话延迟测试结果

由表 4.1 可以看出, 会话通信延迟随着用户数目的增加在不断增大。由于在通信指令交互过程中, 数据包数目较少, 且包数据量较小, 因此 NB 网络中 BrokerNode 的处理时延基本可以忽略。而导致延时增加的原因主要是会话服务器的处理时延在增加。在 50 个用户同时发起登录、创建会话、邀请加入等请求的时候, 会话服务器内的多个服务线程要协同工作, 分配系统资源。登录和创建会话只需要进行相关验证即可回复, 而邀请其他用户加入则涉及到验证、用户在线/下线状态判断、会话信息更新等一系列操作, 因此延迟较大, 但均控制在 100ms 之内, 属于用户可接受范围。

#### 2. 音视频传输延迟:

时延是处理和传输时导致数据不能按时到达的延迟, 是影响流媒体数据传输的一个主要因素。时延的产生主要包括: 编解码器引入的时延、打包时延、去抖动时延、承载网上的传输节点中排队、服务处理时延等。这些时延累计的总和将影响音/视频质量, 导致回声干扰和交互性的劣化。对于 VoIP 系统, 规定时延一般控制在 150ms 内<sup>[26]</sup>。

在本文所设计的集群通信系统中, 音视频的传输时延主要是指发送者发出 RTPEvent 到接收者收到该数据包的时间, 它包括网络传输时延和 BrokerNode 处理时延两部分, 而由于测试环境为局域网, 因此后者是影响整个时延的主要因素。整个传输时延过程如图 4.6 所示。

从图 4.6 中可以看出, RTPEvent 从 Transmitter 到 Receiver 的时间  $T_{AB}$  即为所求时延。而 RTPEvent 将其生成时间  $T_A$  作为时间戳记载在事件头内, 这样, 在 Receiver 端就可以根据其收到 RTPEvent 的时间  $T_B$  来计算  $T_{AB}$ , 从而有  $T_{AB}=T_B-T_A$ 。

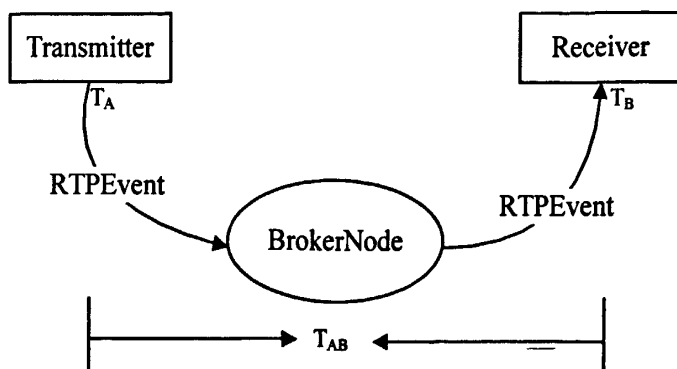


图 4.6 延时计算示意图

随着系统中接收用户数目的不断增多, BrokerNode 的处理量将不断增大, 而 PC 机的性能有限, 则整个时延也将不断增大, 从而影响整个集群通信系统的性能。因此, 按照上述计算方法, 不断增加该系统的用户数量, 并在每个用户接收端计算时延的平均值, 最后将所有接收用户时延再取平均, 得到音频和视频的时延性能如图 4.7 所示。

由图 4.7 可以看出, 在音视频同传的情况下, 随着用户数目的增加, 音视频时延在不断增大。而由于视频包的数据量比较多, 发送速率比较快, 延时也比音频要大许多。当系统用户数目在 150 之内时, 延迟在 30ms 之内, 而用户数目达到 400 的时候, 视频延时达到 60ms 左右。这样的传输延时, 再加上终端的处理延时, 都可以控制在 100ms 之内, 满足音视频的传输要求。

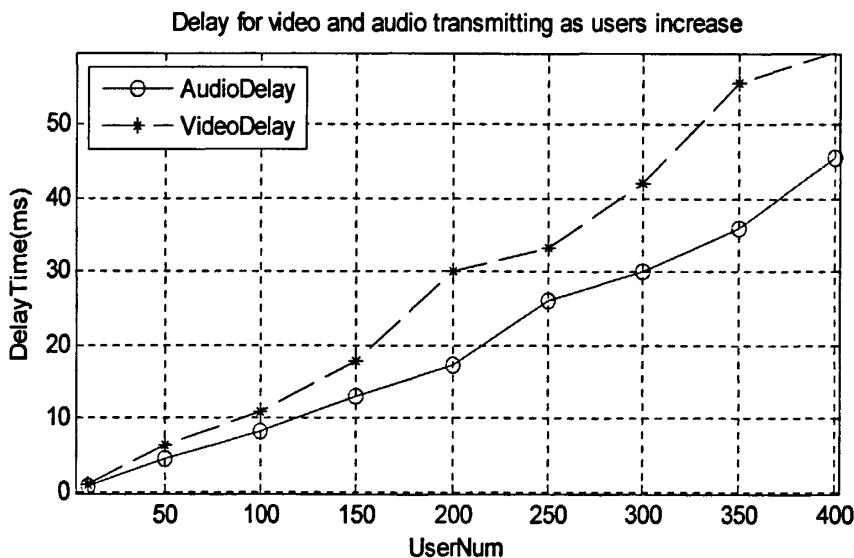


图 4.7 音视频延时测试结果

### 3. 音视频传输抖动:

抖动也是衡量多媒体传输性能的一个重要标尺。在音/视频数据的传输过程

中, 由于传输延迟的不稳定而造成相邻数据包接收时刻间隔不稳定, 从而产生抖动。本系统中音视频的传输采用的是 RTP 协议, 而该协议利用时间戳机制对抖动进行了详细的分析和计算, 并将计算结果通过 RTCP 协议呈现给用户。根据 RTP 协议, 在一个发送者和一个接收者的环境中, 包  $i$  和包  $j$  之间增加的延迟差(二阶效应)计算公式如下: 设  $R_j$  代表第  $j$  个包的接受时刻,  $S_j$  代表第  $j$  个包的 RTP 时戳值, 则第  $i$  个 RTP 报文与第  $j$  个 RTP 报文间的抖动为  $D(i, j)$ :

$$D_{(i,j)} = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad \text{式(4-1)}$$

在生成 RTCP 报文时, 其应当传送的时延抖动的值可用以下公式进行递推计算:

$$J_{(i)} = J_{(i-1)} + [|D_{(i-1,j)}| - J_{(i-1)}] / 16 \quad \text{式(4-2)}$$

其中:  $J$  为要传送的时延抖动值。对后一项除以 16 是为了消除连带噪声<sup>[27]</sup>。

在接收者端经过上述公式计算得到的这个抖动值记载在 RTCP 协议接收者报文 RR (ReceiverReport) 中。因此, 在每个接收端不断获取 RR 中的抖动值, 并取平均值, 然后再在各个用户的平均值的基础上去平均, 即可得到当前用户数目下音视频抖动的平均值。

增加系统用户数目, 观察用户不断增加情况下音视频传输抖动性能的下降, 如图 4.8 所示。

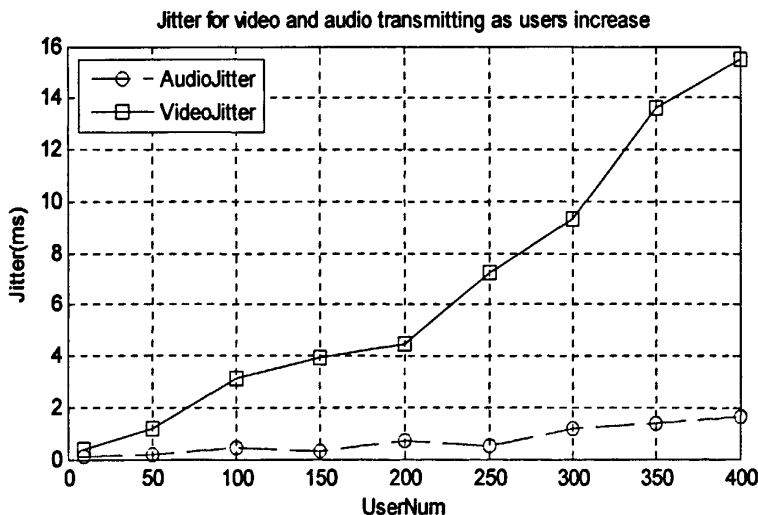


图 4.8 音视频抖动测试结果

从图 4.8 中可以看出, 音视频的抖动相差极大。在用户数目为 400 的时候, 音频的抖动控制在 2ms 之内, 而视频抖动已经达到 16ms。但这些抖动值均在音视频播放可以接受的范围内。而且 JMF 和 Android 系统的 openCore 框架都增加了缓冲功能, 来缓解抖动对音视频播放造成的影响。

经过一系列的性能测试, 系统的音视频延时以及抖动虽然随着用户数目增加有明显增大, 但都属于用户可允许范围。本文设计的基于 IP 网络的多媒体集群通

信系统取得较理想的结果。

#### 4.4 本章小结

本章首先进行了应用层组播平台 NaradaBrokering 的搭建与配置,然后与手机客户端进行系统联调。这样,基于 IP 网络的多媒体集群通信系统中的会话服务器、PC 机客户端以及手机客户端的基础开发完成,可以进行相关测试。

本章的最后,对设计的系统进行了整体测试,包括功能测试和性能测试两部分。通过对测试结果分析,本文设计的基于 IP 网络的多媒体集群通系统在局域网测试环境中取得了较为理想的性能。



## 第五章 总结与展望

随着 IP 网络技术的发展以及下一代移动网络的全 IP 化, 数据集群通信系统正在以各种不同的方式向 IP 网络演进。本文即是在此基础上提出的基于 WIFI 智能手机的一项增值业务——集群通信功能。这种集群通信功能并不像传统的集群通信系统一样, 需要专用终端, 部署专门网络, 而是基于已经得到广泛应用的 IP 无线局域网, 由具有 WIFI 入网功能的智能手机作为用户终端, 因而具有平台统一, 价格低廉、扩展简便等一系列优势。

本文所做的工作主要包括:

1. 基于 IP 网络的三种通讯模式 IP 单播、IP 组播以及应用层组播设计了三种系统结构, 并通过比较、分析, 最终选择第三种方案进行系统的具体设计。

2. 通过分析文中设计的集群通信系统对应用层组播平台的需求, 选择了开源项目 NaradaBrokering 作为传输平台, 并对 NaradaBrokering 项目进行了深入的研究和测试, 对其提供的功能进行了进一步验证, 并根据本文所设计的系统进行相应改善。

3. 在充分考虑底层传输平台应用层组播特性的基础上, 对系统中各部分的功能、系统工作流程以及系统控制协议进行了具体设计。

4. 对系统中的会话服务器和 PC 机客户端进行了分模块的设计与实现。会话服务器分为服务模块、服务管理模块、消息管理模块以及 NB 系统发布/订阅模块, PC 机客户端分为用户界面模块、信令处理模块、多媒体处理模块以及 NB 系统的发布/订阅模块。

5. 搭建应用层组播平台, 并在此基础上与手机客户端进行了功能联调。

6. 对实现的系统进行功能及性能测试, 并对结果进行分析, 以判定系统的可用性。

根据测试结果, 本文设计的基于 IP 网络的多媒体集群通信系统实现了集群通信所需要的基本功能, 性能较好。在这样的基础上, 该系统基于 IP 网络所拥有的平台优势以及价格优势将成为其亮点。

当然, 由于时间原因, 该系统的设计与实现都属于完成初期, 有许多方面考虑得并不完善, 后续研发工作还需要继续。主要工作包括下面几点:

1. 系统的服务质量控制方面: 可以利用 RTP/RTCP 的协议特性, 进行信息反馈, 由会话服务器进行综合评定, 动态控制发送者的发送速率以及编码速率, 以提高网络性能。

2. 网络规模方面: 由于条件限制, 该系统的测试环境选择的是局域网, 系统的整体规模受限。而根据 NB 系统的性能, 能够支持大规模网络实时传输, 因此, 后期应该配置广域网环境, 进行研发测试, 从而使该系统能够在更大范围内



得到应用。

3. 系统功能方面：目前系统的功能只是基础的会话功能，应该在此基础上进行扩展，比如增加语音信箱功能、优先级功能等。

## 致 谢

在本论文最终完成之际，首先要感谢我的导师陈彦晖教授。在这两年多的时间里，陈老师为我提供了宝贵的学习平台和丰富的项目实践，令我受益匪浅。在完成论文的过程中，陈老师给予了细心的指导，从最初选题、设计改进、论文的撰写直到最后的修改，都提出了宝贵的意见和建议，而且在我遇到困难时给予了很大的帮助。陈老师严谨的治学态度、渊博的学术知识、宽广的胸襟和对学生的关怀，都给了我深深的启迪。在此特别感谢！

感谢 101 宿舍我的姐妹们：李林、王继荣、种雕雕，在两年半的学习生活中，朝夕相处，你们给予了我莫大的支持、鼓励和帮助。

感谢 614 实验室与我一起学习、进步的兄弟姐妹，特别是于文斌、胡伟、李斌、赵恩峰、徐磊、钟安红、程冰，在论文的撰写和修改的过程中给予了我很多宝贵的意见。

感谢我的家人以及父母，他们的关怀和鼓励永远是我前进的动力。

感谢评审论文和出席答辩会的各位专家和老师，在百忙之中给予的悉心指导！

最后，再次向以上所有关心、支持、帮助过我的老师、同学、亲人和朋友致以最衷心的祝愿！



## 参考文献

- [1] 张乃通等编著 . 集群移动通信系统. 北京市: 人民邮电出版社, 1994. pp18-20.
- [2] 李佳骏 . 探析集群移动通信共网的发展. 数字通信世界. 2006 年, (3). pp40.
- [3] 徐小涛 . 数字集群移动通信系统原理与应用. 北京市: 人民邮电出版社, 2008. pp12.
- [4] 徐小涛 . 数字集群移动通信系统原理与应用. 北京市: 人民邮电出版社, 2008. pp14-16.
- [5] 姜维, 王海超 . 基于 IP 的下一代数字集群通信系统. 通信世界 B. 2009 年, (44) . pp26.
- [6] 刘莹, 徐恪著 . Internet 组播体系结构. 北京市: 科学出版社, 2008. pp2.
- [7] 张文, 罗宇, 曾嵘等 . 基于覆盖组播的应用层组通信服务系统. 计算机工程与设计. 2005 年 10 月, 26 (10) . pp2591.
- [8] Geoffrey Fox and Shrideep Pallickara . "A Scaleable Event Infrastructure for Peer to Peer Grids". Community Grid Labs, Indiana University, 501 N. Morton St, Suite 224 Bloomington, IN-47404. USA. 2007.
- [9] Amey Dharurkar, Shrideep Pallickara and Geoffrey Fox . "A Topology Viewer for Distributed Brokering Systems". Community Grids Laboratory, Indiana University. 2005.
- [10] Geoffrey Fox and Shrideep Pallickara . "NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids". Community Grid Labs, Indiana University, 501 N. Morton St, Suite 224 Bloomington, IN-47404. USA. 2007.
- [11] Geoffrey Fox, Shrideep Pallickara . "On the Matching of Events in Distributed Brokering Systems". Community Grid Labs, Indiana University. 2008.
- [12] Harshawardhan Gadgil, Geoffrey Fox and Shrideep Pallickara . "On the Discovery of Topics in Distributed Publish/Subscribe systems". Community Grid Labs, Indiana University. 2006.
- [13] Pallickara and Shrideep Fox . "A Transport Framework for Distributed Brokering Systems". Electrical Engineering and Computer Science. 2003. pp79.
- [14] Wu, Wenjun and Geoffrey . "Service Oriented Architecture for VoIP Conferencing". Electrical Engineering and Computer Science. 2006. Paper 144.

- [15] Hasan Bulut, Geoffrey Fox and Shrideep Pallickara . “Implementing a NTP-Based Time Service within a Distributed Middleware System”. Community Grid Labs, Indiana University. 2009.
- [16] Shrideep Pallickara and Geoffrey Fox. “A Scheme for Reliable Delivery of Events in Distributed Middleware Systems”. Community Grid Labs, Indiana University. 2008.
- [17] Geoffrey Fox and Shrideep Pallickara . <http://www.naradabroking.org/deployments/index.html>.
- [18] 卢官明 . 移动流媒体技术. 北京市: 电子工业出版社, 2010. pp12-13.
- [19] 耿祥义 . XML 基础教程. 北京市: 清华大学出版社, 2006. pp2-6.
- [20] 钱屹, 侯义斌 . 基于多播的实时多媒体协作系统中发言权控制的研究. 计算机工程与应用. 2004 年, 40(11). pp146.
- [21] 孙卫琴 . Java 网络编程精解. 北京市: 电子工业出版社, 2007. pp65-74.
- [22] (美) 波德鲁, (美) 图拉赫, (美) 威尔兰格, 叶亮译 . NetBeans 富客户端编程权威教程. 北京市: 清华大学出版社, 2008. pp173-178.
- [23] 彭波, 孙一林编著 . Java 多媒体技术. 北京市: 清华大学出版社, 2005. pp12.
- [24] E2ECloud 工作室编著 . 深入浅出 Google Android. 北京市: 人民邮电出版社, 2009. pp6-12.
- [25] 韩超 . Android 系统原理及开发要点详解. 北京市: 电子工业出版社, 2010. pp216.
- [26] 张明敏 . 多媒体技术与应用. 北京市: 清华大学出版社, 1998. pp60-61.
- [27] H. Schulzrinne, S. Casner and R. Frederick . “RTP: A Transport Protocol for Real-Time Applications”. Network Working Group Request for Comments: 3550. 2003. pp33.