

# SIP

# 协议及其应用

张智江 张云勇 刘韵洁 编著  
博嘉科技 审



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



# 内 容 简 介

**入门要求低** 本书介绍了下一代网络、SIP、SIP最基本的知识,读者只需基本的电信及网络知识即可。

**完整性** 从SIP起源到SIP特征,从SIP基本结构到SIP应用,从SIP在固定软交换网络中的应用到移动领域中的应用,从SIP协议测试到SIP开发部署,从SIP产品现状到未来的展望。在附录中列出了常见网址和一些SIP类与接口层次,这对读者的学习将有很大的帮助。

**概括性** 本书每章的标题就是对该章内容的高度概括,在接下来的内容中对其进行的解释尽可能做到准确、翔实。

**实用性** 本书紧密结合应用,对具体应用的开发都进行了较详细的介绍。

**新颖性** 本书对下一代网络业务相关的最新的接口及协议、规范和国内外研究进展都进行了介绍,并对SIP的未来发展进行了展望。

**配套资源** 为充分展现本书编写特点,进一步提高对本书教学的使用效率,特建立本书使用指导联络方式, E-mail: [bojia@bojia.net](mailto:bojia@bojia.net); 提供本书配套的电子教案,为老师教学提供有益的参考和帮助, 网址: [www.bojia.net](http://www.bojia.net)。

本书贴有激光防伪标志,凡没有防伪标志者,属盗版图书。

ISBN 7-121-00643-X



9 787121 006432 >



责任编辑:王春宁

责任美编:欧美尼设计室

ISBN 7-121-00643-X

定价: 26.00 元

# SIP 协议及其应用

张智江 张云勇 刘韵洁 编著  
博嘉科技 审

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

SIP 是 IETF 提出的应用层控制协议, 用于在 IP 网上建立、修改以及终止多媒体会话或呼叫。

本书是目前此类技术方面较为系统的专著。结合作者多年研究和实践的经验, 从 SIP 起源到 SIP 特征, 从 SIP 基本结构到 SIP 应用, 从 SIP 在固定软交换网络中的应用到移动领域中的应用, 从 SIP 协议测试到 SIP 开发部署, 从 SIP 产品现状到未来的展望, 都进行了论述。本书在编写过程中, 既做到内容全面、叙述清楚, 又注意一些最新的协议、规范及学术界、工业界研究进展, 同时还非常注重实用性, 对具体的 SIP 应用、开发都作了较详细的介绍。

本书体系完整, 内容大致根据协议分层结构、按照自底向上的顺序组织; 全书理论翔实, 语言通俗易懂, 实例实用性和针对性强, 既适合作为高等院校通信、计算机专业本科生和研究生的教材, 也可供有关技术培训及工程技术人员自学参考之用。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

## 图书在版编目 (CIP) 数据

SIP 协议及其应用 / 张智江等编著. —北京: 电子工业出版社, 2005.1  
ISBN 7-121-00643-X

I. S… II. 张… III. 移动通信—通信协议 IV. TN915.04

中国版本图书馆 CIP 数据核字 (2004) 第 125278 号

责任编辑: 王春宁

印 刷: 北京东光印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 16 字数: 400 千字

印 次: 2005 年 1 月第 1 次印刷

印 数: 5 000 册 定价: 26.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。  
联系电话: (010) 68279077。质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。



## 《SIP 协议及其应用》编委会成员

北京邮电大学	陈俊亮院士、教授、博士生导师
	苏森副教授
	彭泳副教授
	邹华副教授
中国科学院计算技术研究所	李国杰院士、研究员、博士生导师
	李忠诚研究员、博士生导师
	谢高岗副研究员
电子科技大学	秦志光教授、博士生导师
	刘锦德教授、博士生导师
	吴跃教授、博士生导师
	谭浩副教授
	彭舰副教授
中国联合通信有限公司技术部	张智江总经理、教授级高级工程师
	齐力焕副总经理、教授级高级工程师
	刘韵洁主任、教授级高级工程师
	张云勇博士后
	王明会处长、副教授、高级工程师
中国联合通信有限公司数据部	杨征工程师
	杜子亭处长、高级工程师
	黄涛
国家科技部西南信息中心	朱宗元总编

# 前 言

在过去几年中,曾经飞速发展的电信业正在逐渐失去往日的辉煌,虽然电信业务的收入仍在增长,但增速已经下降,ARPU (Average Revenue Per Unit) 值也呈日益下降趋势。造成这种局面的一个重要原因是缺少适合人们需要的、能进一步刺激电信业发展的新业务与新应用。虽然智能网技术在为人们提供被叫付费(800 业务)、记账卡等增值业务的同时也为运营商带来了丰富的利润,但它存在业务开发和运营不开放、SCP 集中控制、与具体的承载网络绑定、业务客户化能力低等固有缺陷,这些缺陷已经严重阻碍了电信增值业务的进一步发展,成为困扰电信运营商的一大难题。

与此同时,以开放性、分布性和综合性为主要特征的下一代网络(NGN, Next Generation Networks) 技术正在深刻地改变着传统电信网的观念和体系结构,并让人们看到电信业振兴的希望。NGN 采用分层、开放的体系结构,其主要技术优势是各层间采用开放的协议或 API 接口,从而有利于打破传统电信网封闭的格局,实现多种异构网络间的融合。更为准确地说,NGN 体系通过将业务与呼叫控制分离、呼叫控制与承载分离来实现相对独立的业务体系,使得上层业务与底层的异构网络无关,从而为实现业务提供了更为广阔的空间。

在未来的电信业务中,可能不存在单一的“杀手业务”,受控管理的开放系统、灵活的计费策略、多样化的业务选择和由需求驱动的业务,才是主宰未来的杀手。在未来,“杀手业务”是建立在开放平台上的由众多内容提供商支持的个性化多媒体服务,而 SIP 协议的出现,使这种新型业务平台的出现成为可能。

SIP (Session Initiation Protocol, 会话发起协议) 是由 IETF (Internet 工程任务组) 提出的 IP 电话信令协议。基于 SIP 协议标准,整合了传统的语音及增值服务,并提供最新的即时通信服务以及 IP 网络上的视频服务,并且可以为其他更多的增值应用服务商提供一个标准的具有高扩展性的平台。系统平台完全采用 Internet 的分布式的体系结构,具有高度的灵活性和可扩展性。正是由于 SIP 这一系列优点,使得其广泛应用于下一代网络中。

与原有的协议相比, SIP 具有以下优势:

基于公开的 Internet IETF 标准,与 HTTP 协议类似,易与 Internet 兼容;可充分利用 Internet 开发工具和人员资源;可与 Voice XML, JSP, J2EE 等结合,快速开发增值业务;支持多媒体应用,如语音、视频、图像、音频、文字、数据等业务;可将语音、视频、Presence、短消息、Web 浏览、定位信息、Push、文件共享等业务结合起来,在语音、数据业务结合和互通方面具有天然优势;与接入无关,可跨越任何媒体和设备实现呼叫控制,支持丰富的媒体格式,可动态增/删媒体流;支持灵活的业务创建模式,甚至终端用户可通过 CPL 进行简单业务的创建;支持智能特性向业务和终端侧发展,减轻网络负担,方便业务开展;具有 Presence、Fork 特性,支持许多协议不能支持的新业务;业务网络和承载网络分离,两者可独立发展;业务网络可以融合现有的固网和移动网业务;协议简单,具有公认的扩展潜力;可方便地与 SOAP 融合,支持 Web Service 开发模式。

更为重要的是, SIP 的出现,使得电信网络 IP 化,也使得 IP 网络电信化。还使大量 IT 开发人员快速、有效地开发电信网络应用。从这个角度上讲,其功不可没。

本书力图给读者介绍全面、系统而深入的 SIP 协议及其应用相关知识,综观全书,本书有如下特点:

**入门要求低** 本书介绍了下一代网络、SIP、IMS 最基本的知识,读者只需基本的电信及网络知识即可。

**完整性** 从 SIP 起源到 SIP 特征,从 SIP 基本结构到 SIP 应用,从 SIP 在固定软交换网络中的应用到移动领域中的应用,从 SIP 协议测试到 SIP 开发部署,从 SIP 产品现状到未来的展望。在附录中列出了常见网址和一些 SIP 类与接口层次,这对读者的学习将有很大的帮助。

**概括性** 本书每章的标题就是对该章内容的高度概括,在接下来的内容中对其进行的解释尽可能做到准确、翔实。

**实用性** 本书紧密结合应用,对具体应用的开发都作了较详细的介绍。

**新颖性** 本书对下一代网络业务相关的最新的接口及协议、规范和国内外研究进展都进行了介绍,并对 SIP 的未来发展进行了展望。

本书体系完整,内容大致根据协议分层结构、按照自底向上的顺序组织;全书理论翔实,语言通俗易懂,实例实用性和针对性强。具体的内容安排如下:第 1 章至第 3 章,是 SIP 总体概述、SIP 基本知识;第 4 章至第 6 章,为 SIP 的应用;第 7 章和第 8 章为 SIP 协议测试及具体开发;第 9 章讨论了 SIP 产品的最新进展及未来发展。

本书在编写过程中,得到博嘉科技资讯有限公司王松先生的热情帮助,在此表示感谢。感谢中国联通有限公司技术部齐力焕女士、技术开发处处长王明会博士、裴小燕博士、杨征先生以及中国联通有限公司博士后科研工作站所有博士与作者的技术交流与讨论。感谢中国科学院计算技术研究所所长李国杰院士提出的若干宝贵意见,感谢中国科学院计算技术研究所信息网络研究室主任李忠诚研究员和谢高岗副研究员的大力支持。

感谢郭维娜在编写过程中所给予的启发和鼓舞。

在本书编写过程中,引用了部分材料,在此一并表示感谢。

本书由张智江、张云勇和刘韵洁担任主要编写工作。同时,参与本书编排的人员还有:邹素琼、王安贵、陈郭宜、程小英、谭小丽、卢丽娟、刘育志、吴淬砺、赵明星、贺洪俊、李小平、史利、张燕秋、周林英、黄茂英、李力、李小琼、李修华、田茂敏、苏萍、巫文斌、邹勤、栗德容、童芳、李中全、蒋敏、刘华菊、袁媛、李建康等,在此一并感谢。

由于编写时间仓促,书中疏漏之处在所难免,欢迎广大读者和同行批评指正。

**配套服务:**为充分展现本书编写特点,帮助读者深刻理解本书编写意图与内涵,进一步提高对本书教学的使用效率,我们建立本书使用指导联络方式,是读者与编者之间交流沟通的直通车。欢迎读者将图书使用过程中的问题与各种探讨、建议反馈给我们,本书编者会竭诚给你满意的答复。我们的联系方式 E-mail: [bojia@bojia.net](mailto:bojia@bojia.net)。

我们为读者提供本书配套的电子教案,为老师教学提供有益的参考和帮助。该服务请登录网址: [www.bojia.net](http://www.bojia.net),可在网站下载专区免费下载。

编 者



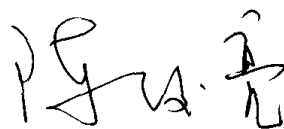
## 序

未来的网络是业务驱动的网络，业务的开发及快速部署显得极其重要，为此有关未来网络中的业务相关课题成为目前国内外电信界普遍关注的研究热点。在未来的电信业务中，可能不存在单一的“杀手业务”，在未来，“杀手业务”是建立在开放平台上的由众多内容提供商支持的个性化多媒体服务，而 SIP 协议的出现，使这种新型业务平台的出现成为可能。

中国联通作为一个不同于传统企业的新兴的运营商，通过各种创新，实现了自身的跨越式发展，也推动了整个通信产业的进步。同时，中国联通聚集了一大批优秀的技术人才，他们在网络建设队伍中始终站在前列，在多年的网络建设中积累了宝贵的工作经验，创造了丰硕的成果。他们把对 SIP 协议及其业务的深刻理解、获取的最新知识、宝贵经验，连同他们的智慧和热情，凝聚在本书中，奉献给社会和读者。作者们编写此书，其目的是给相关从业者从观念上、技术上给予启发，为未来的 SIP 业务全面铺开打下坚实的基础。

本书包含 SIP 起源、SIP 特征、SIP 基本结构、SIP 应用、SIP 在固定软交换网络中的应用、SIP 在移动领域中的应用、SIP 协议测试和 SIP 开发部署、SIP 产品现状及其未来的展望。本书在编写过程中，作者们既做到内容全面、叙述清楚，又非常注重实用性。既适合电信管理人员，也适合工程技术人员，同时也是一本很好的参考书。作者们既具有大型的 SIP 网络规划和建设经验，又紧跟 SIP 网络相关标准与规范的最新发展，因而对该项技术把握得比较准确、论述比较流畅。不仅技术性强，而且具有易读性和实用性。作者们从各自所在的研究项目与重大工程中抽出时间写作此书，其意义丝毫不亚于他们手头的一、两个项目。作者们为了整个书稿简洁、生动而不厌其烦地几易其稿，相信会对 SIP 网络及其业务的发展起到一定的指导、推动和参考作用。

中国科学院、中国工程院两院院士



## 本书作者简介

**张智江**，博士后，教授级高级工程师，国家 863 信息安全委员会专家组成员，国家 3G 实验专家组成员，长期从事我国电信网络的规划、建设、运营管理工作，有着丰富的经验。现为中国联合通信有限公司（总部）技术部总经理，负责电信新技术的研究与开发、管理工作。曾完成联通公司软交换网络重大实验工程，对国内外互联网技术发展现状及趋势、主要核心技术有着深厚的研究。曾获深圳市科技进步一等奖、广东省科技进步二等奖、中国联通科技进步一等奖。

**张云勇**，博士后，中国人工智能学会智能控制与智能管理专业委员会委员，中国人工智能学会可拓工程专业委员会委员。曾作为主要研究人员，参与了总装备部项目、教育部博士点基金、863、国家科技部项目的研究。另外还主持研究了证券监管系统，并获国家鉴定。

目前在中国联合通信有限公司（总部）技术部从事下一代电信网络与下一代互联网的研究，参与中国发改委 CNGI 项目与联通软交换网络的建设。在核心刊物以上级别发表了近 50 篇学术论文，在清华大学出版社、电子工业出版社出版五部论著，深受读者好评。另外还向国际组织递交了 10 余篇研究报告。

**刘韵洁**，教授级高级工程师，中国互联网应用与信息服务委员会主任，长期从事电信网络的规划、建设与运营管理工作，有着非常丰富的经验，曾担任数据所所长、中国电信总局数据局局长、中国联合通信有限公司副总裁等职务，为中国电信行业，尤其是互联网等数据网络技术的发展做出了杰出的贡献。在任中国电信总局副局长兼数据通信局局长期间，领导组织了我国公用数据网、计算机互联网、高速宽带网的网络建设、经营与管理工作，被业界称为“中国互联网之父”。已完成若干重大技术的研究与开发、管理工作，在国内外刊物上发表了若干有影响的学术论文，出版论著多部。

# 目 录

第 1 章 SIP 协议引言 .....	(1)
1.1 SIP 协议起源 .....	(1)
1.2 SIP 功能 .....	(2)
1.3 与其他协议比较 .....	(3)
1.3.1 H.323 协议族 .....	(3)
1.3.2 MGCP 协议 .....	(3)
1.3.3 SIP 协议 .....	(4)
1.3.4 H.323 和 SIP 协议的比较 .....	(4)
1.4 SIP 特点 .....	(6)
1.5 SIP 典型应用 .....	(8)
1.6 SIP 标准化 .....	(8)
1.6.1 SIP 论坛 .....	(8)
1.6.2 IETF .....	(8)
1.6.3 3GPP/3GPP2 .....	(9)
1.6.4 ITU-T .....	(9)
1.7 SIP 实现机制 .....	(9)
1.8 小结 .....	(11)
1.9 习题 .....	(11)
第 2 章 SIP 协议基本介绍 .....	(12)
2.1 SIP 系统基本组成 .....	(12)
2.2 SIP 的消息格式 .....	(13)
2.2.1 概述 .....	(13)
2.2.2 SIP 消息总体描述 .....	(13)
2.2.3 SIP 语法 .....	(21)
2.2.4 SIP 时钟 .....	(28)
2.3 SDP 及 SIP 对 SDP 的要求 .....	(29)
2.3.1 SDP 协议介绍与分析 .....	(29)
2.3.2 SIP 对 SDP 的要求 .....	(31)
2.4 实例 .....	(32)
2.4.1 请求消息 .....	(32)
2.4.2 注册/注销过程 .....	(33)
2.4.3 呼叫过程 .....	(33)
2.4.4 重定向过程 .....	(33)
2.4.5 能力查询过程 .....	(33)
2.5 SAP 协议 .....	(34)
2.6 小结 .....	(34)
2.7 习题 .....	(34)



第 3 章 SIP 协议相关扩展 .....	(35)
3.1 SIP 扩展介绍 .....	(35)
3.2 传送 ISUP 信令的扩展 .....	(35)
3.2.1 SIP-T 介绍 .....	(35)
3.2.2 SIP-I 协议 .....	(35)
3.3 穿越防火墙 .....	(36)
3.3.1 概述 .....	(36)
3.3.2 SIP 协议穿越 NAT/FW 存在的问题 .....	(36)
3.3.3 参考模型 .....	(36)
3.3.4 协议扩展 .....	(37)
3.4 临时应答的可靠传输 .....	(37)
3.5 用户偏好扩展 .....	(37)
3.6 INFO 扩展 .....	(38)
3.6.1 INFO 简介 .....	(38)
3.6.2 INFO 方法 .....	(39)
3.6.3 INFO 消息体 .....	(40)
3.6.4 利用 INFO 扩展的指导方针 .....	(41)
3.7 SIMPLE .....	(41)
3.8 SIP 多方会议扩展 .....	(42)
3.9 SIP 异步事件扩展 .....	(43)
3.9.1 基于会话启动协议的事件通告机制 .....	(43)
3.9.2 自动回叫业务示例 .....	(46)
3.9.3 事件通告机制的安全性考虑 .....	(47)
3.10 第三方呼叫控制及会话传递 .....	(47)
3.10.1 第三方呼叫控制 .....	(47)
3.10.2 会话传递 .....	(47)
3.11 SIP 计费 .....	(48)
3.12 SIP 安全 .....	(50)
3.12.1 AKA 机制 .....	(50)
3.12.2 SIP 私密性 .....	(51)
3.13 SIP QoS .....	(53)
3.13.1 在 SIP 中实现 RSVP 功能 .....	(53)
3.13.2 SIP COMET .....	(54)
3.13.3 策略服务 .....	(55)
3.14 SIP-H.323 互通 .....	(68)
3.15 SIP MIB 变量 .....	(69)
3.16 SIP 其他扩展 .....	(70)
3.17 小结 .....	(70)
3.18 习题 .....	(70)
第 4 章 SIP 在软交换网络中的应用 .....	(71)

4.1	软交换基础 .....	(71)
4.1.1	软交换简介 .....	(71)
4.1.2	软交换的系统结构、协议及业务 .....	(71)
4.2	软交换与 SIP 系统互通 .....	(73)
4.2.1	软交换与 SIP 系统互通的功能要求 .....	(73)
4.2.2	互通方式 .....	(74)
4.2.3	呼叫控制流程 .....	(74)
4.2.4	软交换位于 SIP 系统中与现有智能网的互通 .....	(79)
4.2.5	基于 SIP 的软交换互通 .....	(81)
4.2.6	软交换与应用服务器间的交互 .....	(84)
4.3	使用 SIP 开展软交换业务 .....	(85)
4.3.1	使用 SIP 开展 Parlay 业务 .....	(85)
4.3.2	利用 SIP 应用服务器来开展软交换业务 .....	(92)
4.4	小结 .....	(93)
4.5	习题 .....	(93)
<b>第 5 章</b>	<b>移动领域中的 SIP 协议及其应用 .....</b>	<b>(94)</b>
5.1	未来移动网发展趋势 .....	(94)
5.1.1	3GPP .....	(94)
5.1.2	3GPP2 .....	(96)
5.2	IMS 介绍 .....	(96)
5.2.1	IP 多媒体子系统概念 .....	(96)
5.2.2	IP 多媒体核心网子系统实体 .....	(97)
5.2.3	3GPP/3GPP2 异同及基于 IMS 的融合 .....	(99)
5.2.4	IMS 进展 .....	(101)
5.3	IMS 中的 SIP 协议 .....	(105)
5.3.1	简介 .....	(105)
5.3.2	IMS 中的 SIP 扩展 .....	(106)
5.4	IMS 中的 SIP 典型流程 .....	(107)
5.4.1	IP 多媒体子系统业务交互的过程 .....	(107)
5.4.2	服务 CSCF 相关流程 .....	(108)
5.4.3	应用服务器/MRFC 相关流程 .....	(109)
5.4.4	计费服务器流程 .....	(110)
5.5	利用 SIP 提供 OSA 业务 .....	(111)
5.5.1	OSA 简介 .....	(111)
5.5.2	MPCC 到 ISC(SIP)的映射 .....	(114)
5.5.3	利用 SIP 可开展的 OSA 业务 .....	(119)
5.6	利用 SIP 开展 CAMEL 智能业务 .....	(120)
5.7	利用 SIP 应用服务器提供业务 .....	(120)
5.7.1	LCS 业务 .....	(120)
5.7.2	IM/Presense 业务 .....	(122)

5.7.3	紧急会话建立 .....	(125)
5.7.4	利用 IMS 来支持会议 .....	(127)
5.7.5	PoC 业务 .....	(129)
5.8	SIP 压缩 .....	(132)
5.8.1	参数扩展 .....	(132)
5.8.2	压缩字典 .....	(133)
5.9	小结 .....	(151)
5.10	习题 .....	(152)
<b>第 6 章</b>	<b>SIP 其他应用 .....</b>	<b>(153)</b>
6.1	SIP 与 ENUM 的结合 .....	(153)
6.1.1	ENUM 简介 .....	(153)
6.1.2	ENUM 作用 .....	(153)
6.1.3	利用 SIP 和 ENUM 发起呼叫 .....	(154)
6.2	SIP 与 IPv6 的结合 .....	(158)
6.2.1	兼容 IPv6 的 SDP .....	(158)
6.2.2	DHCPv6 关于 SIP 的选项 .....	(159)
6.3	SIP 在 PINT 中的应用 .....	(160)
6.3.1	PINT 介绍 .....	(160)
6.3.2	PINT 对 SIP 的增强 .....	(161)
6.4	SIP 在 SPIRITS 中的应用 .....	(162)
6.4.1	SPIRITS 介绍 .....	(162)
6.4.2	SIP 在 SPIRITS 中的应用 .....	(163)
6.5	SIP 在 DCS 中的应用 .....	(165)
6.5.1	PacketCable 介绍 .....	(165)
6.5.2	基于 SIP 的基本呼叫流程 .....	(165)
6.6	SIP 在设备控制中的应用 .....	(166)
6.7	小结 .....	(168)
6.8	习题 .....	(168)
<b>第 7 章</b>	<b>SIP 测试技术 .....</b>	<b>(169)</b>
7.1	测试概述 .....	(169)
7.1.1	协议测试的基本概念 .....	(169)
7.1.2	协议测试的类别 .....	(169)
7.1.3	协议一致性测试的理论 .....	(170)
7.1.4	抽象测试集和 TTCN 简介 .....	(172)
7.1.5	协议一致性测试的研究现状 .....	(173)
7.2	SIP 测试现状 .....	(174)
7.2.1	ETSI .....	(174)
7.2.2	SIP 论坛 .....	(174)
7.2.3	SIP 协议互通测试结果 .....	(174)
7.2.4	可用于测试的公开 SIP 服务器 .....	(175)



7.3	小结 .....	(176)
7.4	习题 .....	(176)
<b>第8章</b>	<b>SIP 业务开发及部署 .....</b>	<b>(177)</b>
8.1	SIP CGI 技术 .....	(177)
8.1.1	SIP CGI 与 HTTP CGI 的区别 .....	(177)
8.1.2	SIP CGI 规范 .....	(178)
8.2	CPL .....	(178)
8.2.1	CPL 脚本结构 .....	(179)
8.2.2	CPL 框架结构 .....	(181)
8.2.3	CPL 的应用 .....	(184)
8.3	JAIN SIP Servlet .....	(185)
8.3.1	Servlet 基本知识 .....	(185)
8.3.2	SIP Servlet .....	(188)
8.4	J2EE 开发环境 .....	(195)
8.4.1	J2EE 引言 .....	(195)
8.4.2	EJB 模型 .....	(196)
8.4.3	EJB 角色 .....	(199)
8.4.4	EJB 和其他技术的关系 .....	(200)
8.4.5	常见 EJB 系统 .....	(200)
8.4.6	开发实例 .....	(201)
8.4.7	J2EE 小结 .....	(203)
8.5	Web 服务 .....	(203)
8.5.1	Web 服务简介 .....	(203)
8.5.2	SOAP .....	(204)
8.5.3	WSDL .....	(204)
8.5.4	UDDI .....	(205)
8.5.5	Web 服务安全 .....	(205)
8.5.6	WS-Routing .....	(205)
8.5.7	WS-Referral .....	(206)
8.5.8	DIME 和 WS-Attachments .....	(206)
8.6	SIP 开发实例 .....	(206)
8.7	SIP 业务部署 .....	(213)
8.7.1	SIP 业务部署模型 .....	(213)
8.7.2	SIP 业务 .....	(213)
8.7.3	SIP 业务发展步骤 .....	(214)
8.8	小结 .....	(215)
8.9	习题 .....	(215)
<b>第9章</b>	<b>SIP 产品现状及未来发展 .....</b>	<b>(216)</b>
9.1	SIP 产品现状 .....	(216)
9.1.1	具有代表性的 SIP 产品 .....	(216)

9.1.2 SIP 防火墙及 NAT ..... (218)

9.1.3 SIP 网关 ..... (219)

9.1.4 SIP 服务器 ..... (220)

9.1.5 SIP 服务 ..... (221)

9.1.6 SIP 组件 ..... (222)

9.1.7 SIP 软件工具 ..... (223)

9.1.8 SIP 用户代理 ..... (223)

9.2 SIP 未来发展 ..... (224)

9.3 小结 ..... (225)

9.4 习题 ..... (225)

附录 A 常见缩略语 ..... (226)

附录 B SIP 相关网址 ..... (229)

附录 C SIP servlet 类层次及接口层次 ..... (231)

附录 D jsip 类与接口 ..... (233)

附录 E SIP RFC 与草案 ..... (236)

参考文献 ..... (240)

# 第 1 章 SIP 协议引言

知识点:

- SIP 起源
- SIP 功能
- SIP 与其他协议比较
- SIP 特点
- SIP 应用领域
- SIP 标准化

随着 IP 电话应用的普及, 建立终端设备和网关的可扩展网络已成为业界面临的一大技术挑战。目前 IP 电话技术已完美实现了对语音和传真信息的编码和传输, 但仍需进一步发展为大型公司和服务提供平台的呼叫控制和地址管理技术。因此, 支持 IP 电话系统互通的多个协议已经出台。常用的 IP 电话协议 (如 H.323、MGCP 和 SIP) 的优缺点各异。其中, 秉承越简单越流行这一设计思想的 SIP 协议的应用前景无限。

## 1.1 SIP 协议起源

SIP (Session Initiation Protocol) 称为会话发起协议, 是由 IETF (Internet Engineering Task Force) 组织于 1999 年提出的一个在基于 IP 网络中, 特别是在 Internet 这样一种结构的网络环境中, 实现实时通信应用的一种信令协议。而所谓的会话 (session), 就是指用户之间的数据交换。在基于 SIP 协议的应用中, 每一个会话可以是各种不同的数据, 可以是普通的文本数据, 也可以是经过数字化处理的音频、视频数据, 还可以是诸如游戏等应用的数据, 应用具有巨大的灵活性。

作为一个 IETF 提出的标准, SIP 协议在很大程度上借鉴了其他各种广泛存在的 Internet 协议, 如 HTTP (超文本传输协议)、SMTP (简单邮件传输协议) 等, 和这些协议一样, SIP 也采用基于文本的编码方式, 这也是 SIP 协议同视频通信领域其他现有标准相比最大的特点之一。

SIP 协议的提出和发展, 是伴随着 Internet 的发展而发展的, 到目前为止它走过了以下几个阶段。

Mark Handley 和 Eve Schooler 开发了会话邀请协议 (Session Invitation Protocol) SIPv1, 于 1996 年 2 月 22 日递交 IETF。SIPv1 协议仅仅处理会话的建立, 一旦用户加入会话, 则信令就终止, 而且会议也无中间控制。

在 IETF 第 35 次会议上, Schulzrinne 提出了 SCIP (Simple Conference Invitation Protocol, 简单会议发起协议), 经过讨论, 将 SIPv1 和 SCIP 进行合并, 最终的协议保留了 SIP 作为名字, 但意义已经改为 Session Initiation Protocol。



随后 Mark Hanley、Schulzrinne 和 Schooler 于 1996 年将 SIPv2 递交给 IETF 第 37 次会议，这是一个基于 HTTP 新的 SIP。

1999 年 3 月，IETF 的多方、多媒体会晤控制（mmusic）工作组提出了 RFC2543 建议，供各厂商和机构讨论。

1999 年 9 月，SIP 工作组从 mmusic 中分离并独立出来，成立了 SIP 工作组，最初由 Joerg Ott、Jonathan Rosenberg 和 Dean Willis 担任主席，并于 2000 年 7 月发表了 SIP 的草案。

2002 年 6 月，IETF 的 SIP 工作组又发表了 RFC3261 建议，以取代 RFC2543。

由于网络环境以及相关多媒体技术的不足，在 SIP 协议首次提出的时候，仅仅针对各种文本应用，随着技术的发展，并通过和 IETF 中 IP 电话工作组（iptel）、IP 网中电话选路（trip）工作组等兄弟工作组配合工作，在 SIP 协议中大大加强了对多媒体通信的支持。

近年来，在建设下一代网络的过程中发生了一个很明显的变化：很多设备厂商都开始将其下一代网络的协议标准向 SIP 方向倾斜。SIP 成为追逐的标准，最大的优势就是开发简单，同时 SIP 具有继承性，对于拥有传统网络的运营商来说，SIP 在网络发展过程中，无疑是一个承上启下最好的切入点。

在 3GPP 中使用 SIP 标准来支持语音和数据是 SIP 协议得以发展的一个重要原因，SIP 可以对语音进行很好的优化，并且由于它的可编程性，使移动业务面临灵活性和多样性的变化时，有了很好的保证。

另外，基于 SIP 的终端丰富多彩有利于从网络接入到最终用户的完整产业链的完成。SIP 能够对手机、PDA 等移动设备提供良好的支持，对于在线即时交流、语音和视频数据传输等多媒体应用也能够很好地完成。

## 1.2 SIP 功能

SIP 被描述为用来生成、修改和终结一个或多个参与者之间的会话。这些会话包括 Internet 多媒体会议，Internet（或任何 IP 网络）电话呼叫和多媒体发布。会话中的成员能够通过多播或单播联系的网络来通信。SIP 支持会话描述，它允许参与者在一组兼容媒体类型上达成一致。它同时通过代理和重定向请求到用户当前位置来支持用户移动性。SIP 不与任何特定的会议控制协议捆绑。本质上，SIP 提供以下功能：

- 名字翻译和用户定位：无论被呼叫方在哪里都确保呼叫达到被叫方、执行任何描述信息到定位信息的映射，确保呼叫（会话）的本质细节被支持。
- 特征协商：它允许与呼叫有关的组（这可以是多方呼叫）在支持的特征上达成一致（注意：不是所有各方都能够支持相同级别的特征）。例如，视频可以或不可以被支持。总之，存在很多需要协商的范围。
- 呼叫参与者管理：呼叫中参与者能够引入其他用户，加入呼叫或取消到其他用户的连接。此外，用户可以被转移或置为呼叫保持。
- 呼叫特征改变：用户应该能够改变呼叫过程中的呼叫特征。例如，一个呼叫可以被设置为“voice-only”，但是在呼叫过程中，用户可以开启视频功能。也就是说一个加入呼叫的第三方为了加入该呼叫可以开启不同的特征。

## 1.3 与其他协议比较

### 1.3.1 H.323 协议族

1996 年, 创建了 H.323, 并于 1998 年 1 月推出版本 2。H.323 据称是一种包罗万象的标准, 因为它本身由众多从属协议构成。正因为如此, ITU 可以通过使用许多现有的数据和通信标准如 Q.931、G.711 和 G.723.1 对 H.323 进行定义。

H.323 最初由 Intel 和 PictureTel 提出, 该协议定义了一种能灵活应用于多媒体电话会议设备的通信方式, 并在 IP 栈上提供应用共享特性。设计人员提出了适用于多种设备的标准, 这些设备包括可视电话、台式电脑和大型多端口网关。因此, H.323 内容广泛, 并提供了应用于不同设备的多种媒体类型和压缩技术。

H.323 的核心优点在于其成熟性, 这有助于诸多软件供应商开发性能稳定的设备, 并且还有利于不同的供应商消除互操作性中出现的问题, 并在市场上推出各种支持 H.323 标准的设备。因为 H.323 标准包容了 Q.931 呼叫控制协议, 许多在现有 ISDN 电话技术上具有丰富经验的开发商对该呼叫控制模型也非常熟悉。实际上, 事件和参数通常能够直接通过 H.323 进入以前工作在 ISDN 下的应用系统。

在定义 H.323 时, 设计人员是从终端设备的角度入手, 而非从现有 PSTN 的内部设备入手, 因此 H.323 不能与 SS7 集成。另外, H.323 的扩展性在超大型应用中已证明确实存在问题, 设计人员在使用含有成千上万个端口的网关时会发现, 集中状态管理是其工作的瓶颈。

市场对 H.323 的反应表明, H.323 的最佳应用点应是位于或临近端点并带有 1~200 个端口的系统。H.323 在具有足够的处理能力实现呼叫控制和媒体处理的环境中工作良好。H.323 作为企业 IP 电话解决方案已得到了业界最强有力的支持。

### 1.3.2 MGCP 协议

媒体网关控制协议 (MGCP) 为众多 IP 电话网关的互联提供了一种解决方案, 能将这些网关连接成一个具备互操作性的整体。MGCP 假定呼叫代理 (CA) 完成所有呼叫控制处理, 而媒体网关控制器 (MGC) 完成所有媒体处理和转换。

MGCP 的规格是由不同的公司开发建立的 (如 Telecordia 和 Lucent 公司), 并由 IETF 以 RFC (RFC 2705) 信息文件的形式加以发布。MGCP 是简单网关控制协议 (SGCP) 和 IP 设备控制 (IPDC) 协议合并的结果。IETF 的 Megaco 工作组协同 ITUA 正致力于开发基于 MGCP 的推荐标准 H.248 (其前身为 H.gcp)。该核心文件及相关规格已于 2000 年 2 月完成, 并作为 IETF 的 RFC 标准文件发布。

当 H.323 网关提供媒体转换, 而 SS7 网关翻译呼叫控制信息时, MGCP 可与 H.323 网关和 SS7 网关协同使用。这种情况下, MGCP 将所有来自端点设备的呼叫控制信息传送至网络。使用这种结构方式, 开发人员能够补充 SS7 网络的能力, 并且与单用 H.323 相比, 能够构建更大的 IP 电话系统。

为协调单个呼叫的媒体路径和性能, MGCP 要依赖对话描述协议 (SDP), 该协议是 MGCP 规格的一部分。SDP 允许就实时协议 (RTP) 端口及端点 IP 地址、语音编码方法 (如 G.711 和 G.723.1)、报文分组周期和其他连接类型参数等进行协商。

MGCP 的优点包括: 特别适合于配置大型应用系统, 因为其本身就用于解决大型系统的

具体问题。应用 MGCP 可以实现与 SS7 网络良好的集成，为呼叫处理提供更大的控制和流通量。MGCP 将媒体处理和信令功能分离，从而可由多个设备提供商开发更简捷的系统。

MGCP 协议的部分缺陷包括：MGCP 对于小型应用系统来说过于复杂。MGCP 与 H.248/Megaco 标准存在竞争关系，而后者已于 2000 年初由 IETF 和 ITU 签署认可。这样，需要 MGC 的运营商既可选用 MGCP，也可选用 H.248。

因此，H.248 或许最终能取代早期的 MGCP 版本。MGCP 的归宿是电信运营市场，由它实现成千上万路 IP 电话的传送。

### 1.3.3 SIP 协议

对话启动协议提供了一种既可在终端设备或代理服务器之间，也可向网关设备传输呼叫控制信息的方法，像许多现有的 Internet 协议一样，SIP 中也包含了常用的 HTTP 协议。

SIP 被认为是一种轻量级协议，因为该协议使用的是能被终端设备轻易生成并分析的简单文本命令。SIP 只使用 6 个指令管理呼叫控制信息，SIP 协议的简单易行是极低成本应用系统选择该协议的重要原因。

SIP 并未对媒体传输机制加以定义，因此该协议能用在媒体传输作为专用设备的应用系统，这可以提高效率并降低成本。SIP 也允许通过任意数据报协议传输呼叫控制信息，使其在非 TCP/IP 环境（如 Novell 或其他专有协议）中也能有效使用。

SIP 的部分优点包括：该协议具有可扩展特性，可以轻松定义并迅速实现新功能。可以简单易行地嵌入廉价终端用户设备。该协议可确保互操作能力，并使不同的设备进行通信。便于那些非电话领域的开发人员理解该协议。

SIP 的缺点包括：SIP 问世不久，因此大多数应用尚处于原型阶段。该协议单独应用的范围较窄。但与其他协议协同使用时，具有较强的灵活性。SIP 只是完整解决方案的一小部分，还需要许多其他的软件来构建完整的 IP 电话产品。

低成本终端产品无疑是 SIP 最自然的应用了，像无线电话、以太网电话及其他带有有限计算和内存资源的设备都能使用该协议。由于 SIP 是一种优越的呼叫控制协议，因此是当前取代 MGCP 呼叫控制协议的首选。

上述每个协议强调了开发 IP 电话系统所需技术的不同方面，当前开发的许多系统中至少包含其中一种协议，而且这些协议通常要求互操作性。所有这些协议在构建复杂 IP 电话系统的过程中不断发展。各制造商都在尝试研制具备互操作性的系统，所以互操作能力将继续成为未来的一大挑战。由 IETF 和 ITU 衍生而来的新标准协议 MGC（H.248/Megaco）有望成为 MGCP 在传输市场最强有力的竞争对手。

### 1.3.4 H.323 和 SIP 协议的比较

H.323 和 SIP 分别是通信领域与 Internet 两大阵营推出的建议。H.323 企图把 IP 电话当做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，较其他应用（如 FTP，E-mail 等）增加了信令和 QoS 的要求，它们支持的业务基本相同，也都利用 RTP 作为媒体传输的协议。但 H.323 是一个相对复杂的协议。

H.323 采用基于 ASN.1 和压缩编码规则的二进制数据方式表示其消息。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 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。这种分布式、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(现在广为流行的邮件协议 SMTP 就是这样的一个例子)。SIP 的消息体部分采用 SDP 进行描述,SDP 中的每一项格式也比较简单。

在支持会议电话方面,H.323 由于由多点控制单元(MCU)集中执行会议控制功能,因而所有参加会议终端都向 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 不支持信令的组播功能,其单一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,具有分布式的组播功能,其组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。但是 H.323 是集中控制便于计费,对带宽的管理也比较简单、有效。

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 并未专门定义的协议用于此目的,但它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(如 Contact 头域),并对 SIP 进行简单的扩展(如增加几个域),就可以实现这些业务。例如,对于呼叫转移,只要在 BYE 请求消息中添加 Contact 头域,加入意欲转至的第三方地址就可以实现此业务。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。

在 H.323 中,呼叫建立过程涉及第三条信令信道: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 已对呼叫建立过程作了改进,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。H.323 的呼叫信令通道和 H.245 控制信道需要可靠的传输协议。而 SIP 独立于底层协议,一般使用 UDP 等无连接的协议,用高层的可靠性机制来保证消息的可靠传输。

标准的应用目标方面,H.323 标准是 ITU-T 组织 1996 年在 H.320/H.324 的基础上建立起来的,其应用目标是,在基于 IP 的网络环境中,实现可靠的面向音视频和数据的实时应用。目前,经过多年的技术发展和标准的不断完善,H.323 已经成为被广大的 ITU 成员以及客户所接受的一个成熟标准族。SIP 标准是 IETF 组织在 1999 年提出的,其应用目标是在基于 Internet 环境,实现数据、音视频实时通信,特别是通过 Internet 将视频通信这种应用大众化。由于 SIP 协议相对于 H.323 而言,相对简单、自由,厂商使用相对小的成本就可以构造满足应用的系统。例如,仅仅使用微软基于 SIP 协议的 MSN,就可以构造一个简单的,基于 Internet 应用环境的视频通信环境。这样网络运营商就可以在尽量少的成本基础上,利用现有的网络资源开展视、音频通信业务的扩展工作。

标准的体系结构方面,H.323 不是一个单一标准,而是一个关于在 IP 环境中实时多媒体应用的完整标准族,对于呼叫的建立、管理以及所传输媒体格式等各个方面都有完善而严格的规定。一个遵守 H.323 标准建立的多媒体系统,可以保证实现客户稳定完善的多媒体通信应用。SIP 标准严格意义上讲是一个实现实时多媒体应用的信令标准,由于它采用了基于文本的编码方式,使得它在应用上,特别是点到点的应用环境中,具有极大的灵活性、扩充性以及跨平台使用的兼容性,这一点使得运营商可以十分方便地利用现有的网络环境实现大规模的推广应用。但是 SIP 协议自身不支持多点的会议功能以及管理和控制功能,而是要依赖于别的协议实现,从而影响了系统的完备性,特别是对于需要多点通信的要求,应用单纯的 SIP 系统难以实现。针对这些不足,以 radvison 公司为首的 ITU-T sg16 小组提出了 SIP 的运用规范,

并实现了 SIP 和 H.323 之间的互通互连, 并成功地解决了 SIP 在多点环境下的应用难题。

系统的组成结构方面, 首先, 在系统主要组成成员的功能性方面进行类比, SIP 的 ua 等价于一个 H.323 的终端, 实现呼叫的发起和接收, 并完成所传输媒体的编解码应用; SIP 代理服务器、重定向服务器以及注册服务器的功能则等价于 H.323 的 gatekeeper, 实现了终端的注册、呼叫地址的解析以及路由。其次, 虽然在呼叫信令和控制的具体实现上不同, 但一个基于 SIP 的呼叫流程与 H.323 的 q931 相类似, SIP 所采用的会话描述协议(SDP)则类似于 H.323 中的呼叫控制协议 H.245。

系统实现的难易性方面, H.323 标准的信令信息是采用符合 asn.1 per 的二进制编码, 并且在连接实现全过程都要严格标准的定义, 系统的自由度小, 如要实现大规模的应用, 需要对整个网络的各个环节进行规划。SIP 标准的信令信息是基于文本的, 采用符合 ISO10646 的 utf-8 编码, 并且全系统的构造结构相对灵活, 终端和服务器的实现也相对容易, 成本也较低; 从网络运营商的角度考虑, 构造一个大规模视频通信网络, 采用 SIP 系统的成本要廉价许多, 而且也更具有可实现性。

总之, H.323 沿用的是传统的实现电话信令模式, 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 协议借鉴了其他 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, 但其推出时间不长, 协议并不是很成熟。

## 1.4 SIP 特点

下一代网络的一个重要目标是建立一个可管理的、高效的、可不断扩展的业务平台。SIP 作为应用层信令协议很好地满足了这一系列要求。

SIP 具有很强的包容性, 它既可以用于建立(如音频、视频、多方通话等)各种会话, 也可以被用来传送即时消息和文件, 这得益于它对 HTTP 等协议的吸收借鉴。这使运营商能通过统一的业务平台提供综合业务, 实现网络的融合。SIP 在灵活、方便地提供业务方面具有多方面优点。

### 1. 协议的可扩展性

SIP 协议处于不断丰富过程之中。SIP 的设计者在保持其核心协议简捷的同时, 为其建立了强大的扩充机制。协议扩充主要是在消息上做文章, 消息的三个基本部分: 消息类型、消息头、消息体都可以被不断扩充。SIP 基于文本的方式, 使各种扩充工作变得十分简便。

#### 消息类型的扩充

SIP 工作组在增加新的功能时, 更愿意定义新的消息类型, 而不是修补原有类型, 以保持每种类型目的单纯, 语义清晰。定义新的类型的工作很简单, 只需定义一个新的字符串即可。

一个被广泛接受的新的类型是 INFO 消息。INFO 消息被用来在会话参加者之间传递各种信息。INFO 消息可以用来传送通话中随机产生的各种信号, 也可以被用来在会话双方间传递即时消息。

还有新提出的 REFER 消息被用来实现呼叫转移的功能, 以及 COMET 消息用来检验能够用于会话的资源, 使用户代理能够根据资源的可用性情况, 决定是否接受一个呼叫。

## 消息头的扩充

SIP 消息包含了一系列消息头,对消息进行必要的描述。现在有 35 种标准的消息头。用户可以根据需要增加新的消息头以支持新的特性,可以结合新的类型定义新的消息头,可以对原有类型中的内容进行补充。

例如,上述为支持呼叫转移新增的 REFFER 类型消息,新增两个消息头 `referred-by` 用来指示发起转移的一方, `reffer-to` 用来指示会话被转移到的一方。

## 消息体的扩充

SIP 的消息体可以通过 MIME 定义的代码进行标识,携带各种类型的数据内容。一个例子是:在新一代网络中,会出现两个 PSTN 网络通过 IP 网络互连,IP 网络提供一个仿真的中继线,这时在 IP 网上采用 IP 电话信令(如 SIP)。对于两侧 PSTN 交互的传统电话信令(如 ISUP),如果要转换成 SIP 相应的内容,到另一侧再转换回来,则难免会造成信息丢失。针对这一问题,增加了一个用 `application/isup` 标识的消息体,将原始 ISUP 信令内容打包,原封不动地通过 SIP 消息携带到对端交换机,这样就可以方便而不失真地传递信令内容。

通过采用 MIME 方式, SIP 消息体可以根据需要任意扩展,携带音频、图像乃至 Java 小程序等各种内容。

可见,强大的扩充机制,使 SIP 的能力不断增强,与时俱进。同时也应看到,由于 SIP 修改扩充容易,各厂家在开发设备时难免有自由发挥的地方。如何保证各厂家设备功能兼容、互连互通,将成为 SIP 取得商业成功道路上面临的最关键的挑战之一。这需要标准组织及各方面协调统一,避免协议无节制地蔓延出各种分支。

## 2. 开放的业务生成环境

传统电话的增值业务是靠智能网来实现的,业务开发复杂、周期长,严重依赖电信设备厂商。面对用户需求日趋多变和个性化,运营商需要能方便地生成各种新的业务,以满足需求、扩大收益。

SIP 网络的业务提供主要由 Proxy Server 完成,生成一个业务就是设计一个业务逻辑从而对一个特定的消息流进行控制,或对消息请求做出相应的响应。这可以通过任何通用的程序语言编程来实现。IETF 起草定义了几种业务生成机制,其思路与 Web 服务器的方法十分接近。

IETF 将业务提供者分为可信任和不可信任两类。对于不可信任者,如终端用户,提供了 CPL (Call Processing Language) 语言,用以处理 INVITE 消息。通过 CPL 的程序,可以决定一个请求是否被拒绝、转移或转发。这里的终端用户可以是第三方提供的应用服务器,第三方业务提供者可以通过一个特服号提供一系列增值业务。使用者可以通过图形工具完成 CPL 程序的设计,由于可能面向缺乏经验的使用者,CPL 拥有严格的控制机制从而避免带来安全和性能问题。

IETF 还为信任用户(如服务器的管理员)提供了两种业务生成的途径: SIP-CGI 和 SIP Servlet。SIP-CGI 类似于互连网站普遍使用的 HTTP-CGI,它独立于编程语言,为复杂的程序处理提供了一个开放的接口。管理者可以使用任何程序语言实现复杂的处理程序将其运行于服务器,在呼叫过程中 SIP-CGI 作为程序调用及其信息输入、输出的接口,从而在呼叫过程中插入复杂的控制逻辑,对呼叫流程进行控制。这使提供高度个性化和智能化的业务变得十分方便。SIP Servlet 是用 Java 语言实现的,类似 Web 服务器常用的 Java Servlet,可以在有消息进入的时候被调用,指示 SIP 服务器如何处理并响应消息。

这些灵活、开放的机制，使业务提供者能使用通用的编程语言方便开发各种智能业务，而不必考虑服务器本身的实现。这与传统电话的智能网系统的私有编程方法和复杂的接口协议形成鲜明对比。业务开发者可以随时根据需要，在任何局部范围内开发、提供新的个性化业务，其好处不言而喻。

基于强大的业务生成手段和灵活的消息格式，SIP 既可以提供各种传统电话业务，如呼叫转移、遇忙转移、话务的排队和分发、800 号等各种智能业务；又可以提供语音与 Internet 互连（PINT）的业务，如单击拨号、语音邮件等；还可以提供特色业务，如即时消息、Presence（根据用户所处的地点和状态决定具体通信方式）等各种新兴而实用的业务。

### 3. 对移动性的支持

SIP 协议天然具有对移动性的支持。SIP 的动态注册机制，使用户端移动变得十分方便。用户端在接入 SIP 网络时，首先向当地注册服务器发出注册请求，将自己的 SIP URL 和 IP 地址告知注册服务器。如果注册服务器不是用户的属主，注册服务器将根据用户注册的 URL 找到并通知其属主：自己现在为用户服务。此后，对该用户的呼叫先到达用户的原属主服务器，再由其重定向到现在为用户服务的服务器上，从而有效实现对移动用户的支持。如果在会话进行中用户移动到新的位置，则可以通过 re-INVITE 重新建立新的会话来实现切换。

SIP 协议为实现固定和移动业务的无缝融合创造了条件。3GPP R5 版本已经选择 SIP 作为 3G 移动通信多媒体域的信令协议，来实现基于 IP 的移动话音和多媒体通信。

## 1.5 SIP 典型应用

SIP 应用有 IP 电话、移动游戏、在场显示与即时通信、视频与协同等。对典型应用的具体介绍见本书第 4 章至第 6 章。

## 1.6 SIP 标准化

### 1.6.1 SIP 论坛

SIP 论坛（[www.sipforum.org](http://www.sipforum.org)）是行业论坛，其目的是促进 SIP 协议的使用。它不是标准组织，所有与技术相关的问题都将反馈给 IETF 以便在合适的工作组内解决。

SIP 论坛具有以下四个工作组：市场工作组、业务提供工作组、认证工作组、SIP 设备工作组。

### 1.6.2 IETF

SIP 协议是由 IETF 工作组制定的，用于在 IP 分组网络上建立、修改以及终止双方或多方的多媒体会话。早期的协议版本是 1999 年提出的 RFC 2543。SIP 协议经过几年的发展，2002 年 6 月，IETF 重新制定并推出了 SIP 协议的 RFC 3261，同时也制定了 SIP 协议的一些相关协议，如：

- RFC 3262(June 2002) “Reliability of provisional responses in Session Initiation Protocol (SIP)”;
- RFC 3263(June 2002) “SIP: Locating SIP Servers”;



- RFC 3264(June 2002) “An Offer/Answer Model with SDP”;
- RFC 3265(June 2002) “Session Initiation Protocol (SIP) Specific Event Notification”。

此系列协议对 RFC 2543 进行了废弃, RFC 3261 也就成为 SIP 协议的新的标准版本。现在的固定网也正是以这个标准为基础进行开发的。

随后, IETF 工作组又制定了一些 SIP 协议的相关协议, 如:

- RFC 3311 (September 2002) “The Session Initiation Protocol (SIP) UPDATE method”;
- RFC 3312 (October 2002) “Integration of resource management and Session Initiation Protocol (SIP)”;
- RFC 3313 (January 2003) “Private Session Initiation Protocol (SIP) Extensions for Media Authorization”;
- RFC 3320 (January 2003) “Signaling Compression (SigComp)”;
- RFC 3323 (November 2002) “A Privacy Mechanism for the Session Initiation Protocol (SIP)”;
- RFC 3325 (November 2002) “Private Extensions to the Session Initiation Protocol (SIP) for Network Asserted Identity within Trusted Networks”;
- RFC 3327 (December 2002) “Session Initiation Protocol Extension Header Field for Registering Non-Adjacent Contacts”;
- RFC 3515 (April 2003) “The Session Initiation Protocol (SIP) REFER method” 等。

目前, 多个 IETF 工作组都在关注 SIP 协议的发展, 如 IP 电话工作组 (IPTEL)、IP 网中电话选路 (TRIP) 工作组、多用途 Internet 邮件控制 (MIME)、呼叫编程语言 (CPL) 工作组等。

### 1.6.3 3GPP/3GPP2

移动网在 IMS 域采用 SIP 协议时, SIP 已经在固网软交换中得到了比较充分的应用, 所以 3G 组织采用的 SIP 协议是以 IETF 的 SIP 标准为基础的, 二者基本一致。

移动网中直接采用 IETF SIP 协议标准中的 SIP 方法(SIP methods), SIP 头部(SIP headers), SIP 选项标签 (Option-tags), SIP 状态码 (Status-code), SIP 会话描述类型 (Session description types)。

### 1.6.4 ITU-T

SIP 已经开始被 ITU-T SG11 PINT, ITU-T SG16, ETSI TIPON(欧洲标准化组织), SIP BOF, IMTE 等各种标准化组织所接受, 并在这些组织中成立了与 SIP 相关的工作组。特别是作为 ITU-T SG16 主要成员, 在多年发展 H323 应用的基础上, 针对 SIP 应用在视频领域的特点, 提出了 SIP 的应用指导, 并推出了相应的 SIP 协议栈, 使得 ITU 的成员实现了这两种协议之间的互通性。

## 1.7 SIP 实现机制

SIP 是一个分层结构的协议, 这意味着它的行为根据一组平等独立的处理阶段来描述, 每一阶段之间只是松耦合。协议分层描述是为了允许功能的描述可在一个部分跨越几个元素, 它不指定任何方式的实现。通常所说的某元素包含某层是指它顺从该层定义的规则集。

不是协议规定的每个元素都包含各层。而且，由 SIP 规定的元素是逻辑元素，不是物理元素。一个物理实现可以选择作为不同的逻辑元素，甚至可能在一个个事务的基础上。SIP 的最低层是语法和编码。它的编码使用增强 BNF 形式语法来规定。

第二层是传输层。它定义了网络上一个客户机如何发送请求和接收响应，以及一个服务器如何接收请求和发送响应。所有的 SIP 元素都包含传输层。

第三层是事务层。事务是 SIP 的基本元素。一个事务是由客户机事务发送给服务器事务的请求（使用传输层），以及对应该请求的从服务器事务发送回客户机的所有响应组成。事务层处理应用层重传，匹配响应到请求，以及应用层超时。事务层具有客户机组成部分（称为客户机事务）和服务器组成部分（称为服务器事务），每个代表有限的状态机，它被构造来处理特定的请求。

事务层之上的层称为事务用户（TU）。每个 SIP 实体，除了无状态代理，都是事务用户。当一个 TU 希望发送请求，它将生成一个客户机事务实例并且向 TU 传递请求和 IP 地址、端口和用来发送请求的传输机制。一个 TU 既生成客户机事务也能够删除它。当客户机取消一个事务时，它请求服务器停止进一步的处理，将状态恢复到事务初始化之前，并且生成特定的错误响应到该事务。这由 CANCEL 请求完成。

SIP 通过 E-mail 形式的地址来标明用户地址。每一用户通过 URL 来标识，它通过诸如用户电话号码或主机名等元素来构造（例如：SIP:user@company.com）。因为它与 E-mail 地址的相似性，SIP URL 容易与用户的 E-mail 地址关联。

SIP 提供它自己的可靠性机制从而独立于分组层，并且只需要不可靠的数据包服务即可。SIP 可典型地用于 UDP 或 TCP 之上。

SIP 提供必要的协议机制以保证终端系统和代理服务器提供以下业务：

- 用户定位；
- 用户能力；
- 用户可用性；
- 呼叫建立；
- 呼叫处理；
- 呼叫前转，包括等效 800 类型的呼叫、无应答呼叫前转、遇忙呼叫前转、无条件呼叫前转；
- 呼叫号码传递，该号码可以是任何命名机制；
- 个人移动性，例如，通过一个单一的、位置无关的地址来到达被呼叫方，即使被呼叫方改变终端；
- 终端类型的协商和选择。呼叫者可以给出选择如何到达对方，例如，通过 Internet 电话、移动电话或应答业务等；
- 终端能力协商；
- 呼叫者和被呼叫者鉴权；
- 不知情和指导式的呼叫转移；
- 多播会议的邀请。

当一用户希望呼叫另一用户，呼叫者用 INVITE 请求初始呼叫，请求包含足够的信息用以被呼叫方参与会话。如果客户机知道另一方的位置，则它能够直接将请求发送到另一方的 IP

地址。如果不知道，客户机将请求发送到本地配置的 SIP 网络服务器。如果服务器是代理服务器，则它将解析被呼叫用户的位置并且将请求发送给它们。可以通过很多方法完成上步，例如，搜索 DNS 或访问数据库。服务器也可以重定向服务器，它可以返回被呼叫用户的位置到呼叫客户机，以便直接与用户联系。在定位用户的过程中，SIP 网络服务器当然能够代理或重定向呼叫到其他的服务器，直到到达一个明确的知道被呼叫用户 IP 地址的服务器。

一旦发现用户地址，请求就发送给该用户，此时将产生几种选择。在最简单的情况下，用户电话客户机接收请求，也就是用户的电话振铃。如果用户接受呼叫，则客户机用客户机软件的指定能力响应请求并且建立连接。如果用户拒绝呼叫，则会话将被重定向到语音邮箱服务器或另一用户。“指定能力”参照用户想启用的功能。例如，客户机软件可以支持视频会议，但用户只想使用音频会议，则只会启用音频功能。

SIP 还具有另外两个有重要意义的特征。第一个特征是有状态 SIP 代理服务器具有分割入呼叫或复制入呼叫的能力，从而可以同时运行几个扩展分支。第一个应答的分支接受呼叫。该特征在用户工作在两个位置之间（例如，实验室和办公室）或者同时对经理和其秘书振铃时是非常便利的。

第二个特征是 SIP 独特的返回不同媒体类型的能力。例如，当 SIP 服务器接收到客户机的连接请求时，它能够通过 Web 交互式语音响应页面来提供列表上的用户。单击适当的链接后将发送一请求到所选择的新用户，从而建立起呼叫。

## 1.8 小结

本章介绍了 SIP 协议产生的背景，SIP 所具有的优异特性以及与其他协议的比较。分析了 SIP 协议的功能、实现机理及典型应用场合，并给出了一些国际标准化组织，如 SIP 论坛、IETF、ITU-T 和 3GPP/3GPP2 关于 SIP 及其相关协议的标准化进展。

## 1.9 习题

1. 分析 SIP 诞生的背景以及为什么会得到重视。
2. 分析 SIP 的特点、功能以及适用场合。
3. 分析 SIP 的实现机制。
4. 简述 SIP 协议的进展情况。

## 第2章 SIP 协议基本介绍

知识点:

- SIP 系统组成
- SIP 消息格式
- SIP 时钟
- SDP 与 SDPng
- SIP 对 SDP 的要求

本章是 SIP 的基本介绍,介绍了 SIP 基本消息、基本头、响应、SDP 及其 SDPng 等内容。通过本章的学习,读者应了解 SIP 设计的思想和由此带来的先进特性。

### 2.1 SIP 系统基本组成

SIP 协议虽然主要为 IP 网络设计的,但它并不关心承载网络,也可以在 ATM、帧中继等承载网中工作,它是应用层协议,可以运行于 TCP,UDP, SCTP 等各种传输层协议之上。

SIP 用户是通过类似于 e-mail 地址的 URL 标识,例如: sip:myname@mycompany.com,通过这种方式可以用一个统一名字标识不同的终端和通信方式,为网络服务和用户使用提供充分的灵活性。

按逻辑功能区分, SIP 系统由 4 种元素组成: 用户代理、代理服务器、重定向服务器以及注册服务器。

#### 1. 用户代理

用户代理 (User Agent) 分为两个部分: 客户端 (User Agent Client), 负责发起呼叫; 用户代理服务器 (User Agent Server), 负责接受呼叫并做出响应。二者组成用户代理存在于用户终端中。用户代理按照是否保存状态可分为有状态代理、有部分状态用户代理和无状态用户代理。

#### 2. 代理服务器

代理服务器 (Proxy Server), 负责接收用户代理发来的请求, 根据网络策略将请求发给相应的服务器, 并根据收到的应答对用户做出响应。它可以根据需要对收到的消息改写后再发出。

#### 3. 重定向服务器

重定向服务器 (Redirect Server), 用于在需要将用户新的位置返回给呼叫方。呼叫方可根据得到的新位置重新呼叫。

#### 4. 注册服务器

注册服务器 (Registrar), 用于接收和处理用户端的注册请求, 完成用户地址的注册。

以上几种服务器可共存于一个设备, 也可以分布在不同的物理实体中。SIP 服务器完全是纯软件实现, 可以根据需要运行于各种工作站或专用设备中。

UAC, UAS, Proxy Server, Redirect Server 是在一个具体呼叫事件中扮演的不同角色，而这样的角色不是固定不变的。一个用户终端在会话建立时扮演 UAS，而在主动发起拆除连接时，则扮演 UAC。一个服务器在正常呼叫时作为 Proxy Server，而如果其所管理的用户移动到了别处，或者网络对被呼叫地址有特别策略，则它将扮演 Redirect Server，告知呼叫发起者该用户新的位置。

除了以上部件，网络还需要提供位置目录服务，以便在呼叫接续过程中定位被叫方（服务器或用户端）的具体位置。这部分协议不是 SIP 协议的范畴，可选用 LDAP（轻量目录访问协议）等。

理论上，SIP 呼叫可以只有双方的用户代理参与，而不需要网络服务器。设置服务器，主要是服务提供者运营的需要。运营商通过服务器可以实现用户认证、管理和计费等功能，并根据策略对用户呼叫进行有效的控制。同时可以引入一系列应用服务器，提供丰富的智能业务。

SIP 的组网很灵活，可根据情况定制。在网络服务器的分工方面：位于网络核心的服务器，处理大量请求，负责重定向等工作，是无状态的，它个别地处理每个消息，而不必跟踪纪录一个会话的全过程；网络边缘的服务器，处理局部有限数量的用户呼叫，是有状态的，负责对每个会话进行管理和计费，需要跟踪一个会话的全过程。这样的协调工作，既保证了对用户和会话的可管理性，又使网络核心负担大大减轻，实现可伸缩性，基本可以接入无限量用户。SIP 网络具有很强的重路由选择能力，具有很好的弹性和健壮性。

## 2.2 SIP 的消息格式

### 2.2.1 概述

SIP 是 IETF 提出的在 IP 网络上进行多媒体通信的应用层控制协议，可用于建立、修改、终结多媒体会话和呼叫，号称通信技术中的“TCP/IP”。SIP 协议采用基于文本格式的客户—服务器方式，以文本的形式表示消息的语法、语义和编码，客户机发起请求，服务器进行响应。SIP 独立于底层协议——TCP、UDP 或 SCTP，采用自己的应用层可靠性机制来保证消息的可靠传送。有关 SIP 协议的详细内容可参见 IETF RFC3261。

### 2.2.2 SIP 消息总体描述

SIP 消息有两种：客户机到服务器的请求（Request），服务器到客户机的响应（Response）。

SIP 消息由一个起始行（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）组成。其中，描述消息体（message body）的头称为实体头（entity header），其格式如下：

```
generic-message = start-line
                  *message-header
                  CRLF
                  [ 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。

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) 4 种。

### 1. SIP 请求消息

请求消息的格式如下：

```
Request = Request-Line
          *( general-header
            | request-header
            | entity-header )
          CRLF
          [ message-body ]
```

请求行 (Request-Line) 以方法 (method) 标记开始, 后面是 Request-URI 和协议版本 (SIP-Version), 最后以回车键结束, 各个元素间用空格键字符间隔。

```
Request-Line = Method SP Request-URI SP SIP-Version CRLF
```

SIP 用术语 “method” 来对说明部分加以描述, Method 标识是区分大小写的。

```
Method = "INVITE" | "ACK" | "OPTIONS" | "BYE"
        | "CANCEL" | "REGISTER" | "INFO"
```

SIP 定义了以下几种方法 (methods)。

#### INVITE

INVITE 方法用于邀请用户或服务参加一个会话。在 INVITE 请求的消息体中可对被叫方被邀请参加的会话加以描述, 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) 响应响应会议邀请。

#### 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, 对其他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。ACK 请求的 To, From, Call-ID, CSeq 字段的值由对应的 INVITE 请求的相应字段的值复制而来。

#### 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; 对于代理和重定向服务器只要转发此请求, 不用显示其能力。

OPTIONS 的 From、To 分别包含主被叫的地址信息, 对 OPTIONS 请求的响应中的 From、To (可能加上 tag 参数)、Call-ID 字段的值由 OPTIONS 请求中相应的字段值复制得到。

#### BYE

用户代理客户机用 BYE 请求向服务器表明它想释放呼叫。

BYE 请求可以像 INVITE 请求那样被转发, 可由主叫方发出也可由被叫方发出。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送媒体流给发出 BYE 请求的一方。

## CANCEL

CANCEL 请求用于取消一个 Call-ID, TO, From 和 Cseq 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)。

CANCEL 请求中的 Call-ID、To、Cseq 的数字部分及 From 字段和原请求的对应字段值相同,从而使 CANCEL 请求与它要取消的请求匹配。

## REGISTER

REGISTER 方法用于客户机向 SIP 服务器注册列在 To 字段中的地址信息。

REGISTER 请求消息头中各个字段的含义定义如下:

- To: 含有要创建或更新的注册的地址记录。
- From: 含有提出注册的人的地址记录。
- Request-URI: 注册请求的目的地址,地址的域部分的值即为主管注册者所在的域,而主机部分必须为空。一般,Request-URI 中的地址的域部分的值和 To 中的地址的域部分的值相同。
- Call-ID: 用于标识特定客户机的注册请求。来自同一个客户机的注册请求至少在相同重启周期内 Call-ID 字段值应该相同;用户可用不同的 Call-ID 值注册不同的地址,后面的注册请求将替换前面的所有请求。
- Cseq: Call-ID 字段值相同的注册请求的 CSeq 字段值必须是递增的,但次序无关系,服务器并不拒绝无序请求。
- Contact: 此字段是可选项;用于把以后发送到 TO 字段中的 URI 的非注册请求转到 Contact 字段给出的位置。如果请求中没有 Contact 字段,那么注册保持不变。
- Expires: 表示注册的截止期。

## 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,有关此方法的使用还有待标准化,详细内容参见 IETF RFC 2976。

## 其他扩展

其他扩展的含义如下:

- re-INVITE: 用来改变参数;
- PRACK: 与 ACK 作用相同,但是用于临时响应;
- SUBSCRIBE: 该方法用来向远端端点预定其状态变化的通知;
- NOTIFY: 该方法发送消息以通知预定者它所预定的状态的变化;
- UPDATE: 允许客户更新一个会话的参数而不影响该会话的当前状态;
- MESSAGE: 通过在其请求体中承载即时消息内容实现即时消息;
- REFER: 其功能是指示接受方通过使用在请求中提供的联系地址信息联系第三方。

## 2. SIP 响应消息

响应消息格式如下:

```
Response = Status-Line
          *(general-header
```



```
| response-header
```

```
| entity-header )
```

```
CRLF
```

```
    [ message-body ]
```

状态行 (Status-Line) 以协议版本开始, 接下来是用数字表示的状态码 (Status-Code) 及相关的文本说明, 最后以回车键结束, 各个元素间用空格字符 (SP) 间隔, 除了在最后的 CRLF 序列中, 这一行别的地方不许使用回车或换行字符。

Status-Line = SIP-version SP Status-Code SP Reason-Phrase CRLF

SIP 协议中用三位整数的状态码 (Status Code) 和原因码 (Reason Code) 来表示对请求做出的回答。状态码用于机器识别操作, 原因短语 (Reason-Phrase) 是对状态码的简单文字描述, 用于人工识别操作。其格式如下:

```
Status-Code    = 1xx (Informational)
                  | 2xx (Success)
                  | 3xx (Redirection)
                  | 4xx (Client-Error)
                  | 5xx (Server-Error)
                  | 6xx (Global-Failure)
                  | extension-code
```

状态码的第一个数字定义响应的类别, 在 SIP/2.0 中第一个数字有 6 个值, 定义如下:

- 1xx(Informational): 请求已经收到、继续处理请求。
- 2xx(Uccess): 行动已经成功地收到, 理解和接受。
- 3xx(Redirection): 为完成呼叫请求, 还须采取进一步的动作。
- 4xx(Client Error): 请求有语法错误或不能被服务器执行。客户机需修改请求, 然后再重发请求。
- 5xx(Server Error): 服务器出错, 不能执行合法请求。
- 6xx(Global Failure): 任何服务器都不能执行请求。

其中, 1xx 响应为暂时响应 (Provisional response), 其他响应为最终响应 (Final Response)。

### 3. SIP 消息头字段

SIP 协议的消息头定义与 HTTP 在语法规则和定义上很相似。每个头字段都遵循以下格式: 首先是字段名 (Field Name), 字段名不分大小写, 后面是冒号; 然后是字段值, 字段值与冒号间可有多个前导空格 (LWS)。其格式如下:

```
message-header = field-name ":" [ field-value ] CRLF
field-name     = token
field-value    = *( field-content | LWS )
```

#### 通用消息头 (General-header)

通用头字段适用于请求消息和响应消息, 包含的字段有:

```
general-header = Accept
                  | Accept-Encoding
                  | Accept-Language
```

```
| Call-ID
| Contact
| CSeq
| Date
| Encryption
| Expires
| From
| Organization
| Record-Route
| Timestamp
| To
| User-Agent
| Via
```

接下来，介绍通用头字段中各字段的含义：

- Accept, Accept-Encoding 和 Accept-Language 字段用于客户机在请求消息中给出其可接受的响应的媒体类型、编码方式以及描述语言。用于服务器在 415 响应（请确认）中表明其可理解的请求消息的媒体类型、编码方式以及描述语言。
- Call-ID 字段：用于惟一标识特定邀请或某个客户机的注册请求，一个多媒体会议可产生多个 Call-ID 不同的呼叫。
- Contact 字段：给出一个 URL，用户可以与此 URL 建立进一步的通信。
- Cseq 字段：用于标识服务器发出的不同请求，若 Call-ID 值相同，那么 Cseq 值必须各不相同。
- Date 字段：反映首次发出请求或响应消息的时间，重发的消息与原先的消息有相同的 Data 字段值。
- Encryption 字段：表明内容经过了加密处理，这种加密为端到端的加密。
- Expire 字段：它给出消息内容截止的日期和时间。
- From 字段：所有消息中都必须有 From 字段，此字段给出请求的发起者。
- Organization 字段：它给出发出请求或响应消息的实体所属的组织的名称。
- Record-Route 字段：它给出一个全局可到达的 Request-URI，用于标识代理服务器。
- Time-Stamp 字段：给出客户机向服务器发出请求的时间。
- To 字段：所有消息中都必须有 To 字段，此字段给出请求的目的收方。
- User-Agent 字段：含有与发起请求的用户代理客户机有关的信息。
- Via 字段：它给出请求消息迄今为止经过的路径。

### 实体头 (Entity-header)

实体头字段用于定义与消息体相关的信息。包含的字段有：

```
entity-header    = Content-Encoding
                  | Content-Length
                  | Content-Type
```

接下来，介绍实体头字段中各个字段的含义：

- Content-Encoding 字段：表明消息体上添加应用的内容编码方式。
- Content-Length 字段：表明消息体的大小。
- Content-Type 字段：表明消息体的媒体类型。

### 请求头 (Request-header)

请求头字段用于客户机上传附加信息到服务器，其中包括有关请求和客户机本身的信息。包含的字段有：

```
request-header = Authorization
                | Contact
                | Hide
                | Max-Forwards
                | Priority
                | Proxy-Authorization
                | Proxy-Require
                | Route
                | Require
                | Response-Key
                | Subject
```

接下来，介绍请求头字段中各个字段的含义：

- Authorization 字段：用于用户代理向服务器鉴定自身身份。
- Hide 字段：用于客户机表明其希望向后面的代理服务器或用户代理隐藏由 Via 字段构成的路径。
- Max-Forwards 字段：表明请求消息允许被转发的次数。
- Priority 字段：用于客户机表明请求的紧急程度。
- Proxy-Authorization 字段：用于客户机向要求身份认证的代理服务器表明自身身份。
- Proxy-Require 字段：用于标识出代理必须支持的代理敏感特征。
- Route 字段：决定请求消息的路由。
- Require 字段：用于客户机告诉代理服务器为了让服务器正确处理请求，客户机希望服务器支持的选项。
- Response-Key 字段：用于给出被叫方用户代理加密响应消息所采用的密钥需满足的要求。
- Subject 字段：提供对呼叫的概述或表明呼叫的性质，可用于呼叫过滤。

### 响应头 (Response-header)

响应头字段用于服务器向 Request-URI 指定的地址传送有关响应的附加信息。包含的字段有：

```
response-header = Allow
                 | Proxy-Authenticate
                 | Retry-After
                 | Server
```

| Unsupported  
| Warning  
| WWW-Authenticate

接下来, 介绍响应头字段中各个字段的含义:

- Proxy-Authenticate 字段: 必须为 407 响应的一部分, 字段中的值给出适用于 Request-URI 的代理的认证体制和参数。
- Retry-After 字段: 可用于 503 响应中, 向发出请求的客户机表明服务预计多久以后可以启用, 用于 404、600、603 响应中表明被叫方何时才有空。
- Server 字段: 含用户代理服务器处理请求所使用的软件信息。
- Unsupported 字段: 列出服务器不支持的特征。
- Warning 字段: 用于传递与响应状态有关的附加信息。
- WWW-Authenticate 字段: 含于 401 响应中, 指出适用于 Request-URI 的认证体制和参数。

### 各种头使用场合

每一种头都有相应的使用场合, 具体如表 2-1 所示。

表 2-1 头使用范围

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Accept	R		-	o	-	o	m*	o
Accept	2xx		-	-	-	o	m*	o
Accept	415		-	c	-	c	c	c
Accept-Encoding	R		-	o	-	o	o	o
Accept-Encoding	2xx		-	-	-	o	m*	o
Accept-Encoding	415		-	c	-	c	c	c
Accept-Language	R		-	o	-	O	O	o
Accept-Language	2xx		-	-	-	O	M*	o
Accept-Language	415		-	c	-	C	C	c
Alert-Info	R	ar	-	-	-	O	-	-
Alert-Info	180	ar	-	-	-	O	-	-
Allow	R		-	o	-	O	O	o
Allow	2xx		-	o	-	m*	m*	o
Allow	r		-	o	-	O	O	o
Allow	405		-	m	-	m	M	m
Authentication-Info	2xx		-	o	-	O	O	o
Authorization	R		o	o	o	o	O	o
Call-ID	c	r	m	M	M	M	M	m
Call-Info		ar	-	-	-	O	O	o
Contact	R		O	-	-	M	O	o
Contact	1xx		-	-	-	O	-	-
Contact	2xx		-	-	-	M	O	o
Contact	3xx	d	-	o	-	o	O	O
Contact	485		-	o	-	O	O	O

续表

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Content-Disposition			o	o	-	O	O	O
Content-Encoding			o	o	-	O	O	o
Content-Language			o	o	-	O	O	o
Content-Length		Ar	t	t	t	T	T	t
Content-Type			*	*	-	*	*	*
CSeq	c	r	m	m	m	m	M	m
Date		a	o	O	o	o	O	o
Error-Info	300~699	a	-	o	O	O	O	o
Expires			-	-	-	O	-	o
From	c	r	m	m	M	M	M	m
In-Reply-To	R		-	-	-	o	-	-
Max-Forwards	R	amr	m	m	m	M	M	m
Min-Expires	423		-	-	-	-	-	m
MIME-Version			o	O	-	o	O	o
Organization		ar	-	-	-	o	O	o
Priority	R	ar	-	-	-	O	-	-
Proxy-Authenticate	407	ar	-	m	-	m	M	m
Proxy-Authenticate	401	ar	-	o	o	o	O	o
Proxy-Authorization	R	Dr	o	o	-	O	O	o
Proxy-Require	R	ar	-	o	-	O	O	o
Record-Route	R	ar	o	o	o	O	O	-
Record-Route	2xx 18x	mr	-	o	o	o	O	-
Reply-To			-	-	-	o	-	-
Require		ar	-	c	-	c	C	c
Retry-After	404							
	413		-	o	o	o	O	o
	480							
	486							
	500		-	o	o	o	O	o
	503							
	600		-	o	o	O	O	o
	603							
Route	R	adr	c	c	c	c	C	c
Server	r		-	o	o	o	O	o
Subject	R		-	-	-	o	-	-
Supported	R		-	o	o	m*	O	o

续表

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Supported	2xx		-	o	o	m*	m*	o
Timestamp			o	o	o	o	O	o
To	c(1)	r	M	m	m	m	M	m
Unsupported	420		-	m	-	m	M	m
User-Agent			o	o	o	o	O	o
Via	R	amr	m	m	m	m	M	m
Via	rc	dr	m	m	m	m	M	m
Warning	r		-	o	o	o	O	o
WWW-Authenticate	401	ar	-	m	-	m	M	m
WWW-Authenticate	407	ar	-	o	-	o	o	o

其中，Where 栏表示头可使用的地方，当为 R 时，表示仅可用在请求中；为 r 时，表示仅可用在响应中；为 c 时，表示将头从请求复制到响应中。

Proxy 栏描述代理对头采取的操作，当为 a 时，表示如果没有头，代理可以增加头；为 m 时，表示代理可以修改头；为 d 表示代理可以删除头；为 r 表示代理必须能读此头，所以此头不能加密。

接下来六栏中的 c 表示根据具体消息来决定是否需要头，m 表示必须要头；m\*表示应该发送头，但是在没有头的情况下，仍应能接收消息；o 表示头可选；t 表示应该发送头，但是在没有头的情况下，仍应能接收消息，如果使用基于流的协议（如 TCP）作为传输协议，则必须发送头；\*表示如果消息体非空，需要此头；-表示头不可用。

### 2.2.3 SIP 语法

以 ABNF 描述的 SIP 语法如下：

SIP-URL = "sip:" [userinfo "@" ] hostport url-parameters [headers]; 有两种类型，Address of Record (AOR) 用来标识用户，需要 DNS SRV 支持，另外一种 Fully Qualified Domain Name (FQDN)，用来标识特定设备。

userinfo = [user | telephone-subscriber] [":" password] ; 用户信息

user = \*( unreserved | escaped | "&" | "=" | "+" | "\$" | "," | ";" | "?" | "/" )

telephone-subscriber = <定义在 [RFC2806]>

password = \*( unreserved | escaped | "&" | "=" | "+" | "\$" | "," ) ; 密码

hostport = host [":" port] ; 端口号

host = hostname | IPv4address | IPv6reference ; 主机名或地址

hostname = \*( domainlabel "." ) toplabel [ "." ] ; 主机名

domainlabel = alphanum | alphanum \*( alphanum | "-" ) alphanum

toplabel = alpha | alpha \*( alphanum | "-" ) alphanum

IPv4address = 1\*3DIGIT "." 1\*3DIGIT "." 1\*3DIGIT "." 1\*3DIGIT ; 地址

IPv6reference = "[" IPv6address "]"

IPv6address = hexpart [":" IPv4address]

hexpart = hexseq | hexseq "::" [hexseq] | "::" [hexseq]

```

hexseq = hex4 *( ":" hex4 )
hex4 = 1*4HEX
port = 1*DIGIT ; 端口
url-parameters = *( ";" url-parameter )
url-parameter = transport-param | user-param | method-param | ttl-param | maddr-param |
other-param
transport-param = "transport=" ( "udp" | "tcp" | "sctp" | "tls" | other-transport ) ; 传输参数
other-transport = token
ttl-param = "ttl=" ttl ; 最大生存时间参数
ttl = 1*3DIGIT ; 0 to 255
maddr-param = "maddr=" host
user-param = "user=" ( "phone" | "ip" | other-user )
other-user = token
method-param = "method=" Method
tag-param = "tag=" UUID
UUID = 1*( HEX | "-" )
other-param = pname [ "=" pvalue ]
pname = 1*paramchar
pvalue = 1*paramchar
paramchar = param-reserved | unreserved | escaped
param-reserved = "[" | "]" | "/" | "?" | ":" | "@" | "&" | "+" | "$"
headers = "?" header *( "&" header )
header = hname "=" hvalue
hname = 1*( hnv-unreserved | unreserved | escaped )
hvalue = *( hnv-unreserved | unreserved | escaped )
hnv-unreserved = "[" | "]" | "/" | "?" | ":" | "@" | "+" | "$"
absoluteURI = <定义在 [RFC2068]>
SIP-message = Request | Response ; 消息
generic-message = start-line *message-header CRLF [message-body]
start-line = Request-Line | Status-Line
message-header = ( general-header | request-header | response-header | entity-header ) ; 消息头
message-body = <定义在 [RFC2068]> ; 消息体
Request = Request-Line*( general-header | request-header | entity-header ) CRLF
[message-body] ; 请求
Request-Line = Method SP Request-URI SP SIP-version CRLF ; 请求行
Request-URI = SIP-URL | absoluteURI
general-header = Accept | Accept-Encoding | Accept-Language | Call-ID | Contact | CSeq | Date
| Encryption | From | Organization | Record-Route | Require | Supported | Timestamp | To |
User-Agent | Via

```



entity-header = Allow| Content-Disposition| Content-Encoding| Content-Language| Content-Length| Content-Type| Expires

request-header = Authorization| Hide| In-Reply-To| Max-Forwards| Priority| Proxy-Authorization| Proxy-Require| Route| Response-Key| Subject

response-header = Proxy-Authenticate| Retry-After| Server| Unsupported | Warning| WWW-Authenticate

Method="INVITE"|"ACK"|"OPTIONS"|"BYE"|"CANCEL"|"REGISTER"| extension-method

extension-method = token

option-tag = token

Response = Status-Line\*( general-header | response-header | entity-header ) CRLF [message-body]

Status-Line = SIP-version SP Status-Code SP Reason-Phrase CRLF

SIP-version = "SIP/2.1" ; 版本

Status-Code = Informational | Success| Redirection| Client-Error| Server-Error| Global-Failure  
| extension-code

extension-code = 3DIGIT

Reason-Phrase = \*<TEXT-UTF8, excluding CR, LF>

Informational = "100" ; Trying

| "180" ; Ringing

| "181" ; Call Is Being Forwarded

| "182" ; Queued

| "183" ; Session Progress

Success = "200" ; OK ; 表示成功

Redirection = "300" ; Multiple Choices ; 重定向

| "301" ; Moved Permanently

| "302" ; Moved Temporarily

| "305" ; Use Proxy

| "380" ; Alternative Service

Client-Error = "400" ; Bad Request

| "401" ; Unauthorized

| "402" ; Payment Required

| "403" ; Forbidden

| "404" ; Not Found

| "405" ; Method Not Allowed

| "406" ; Not Acceptable

| "407" ; Proxy Authentication Required

| "408" ; Request Timeout

| "409" ; Conflict

| "410" ; Gone

| "411" ; Length Required

```

| "413" ; Request Entity Too Large
| "414" ; Request-URI Too Large
| "415" ; Unsupported Media Type
| "420" ; Bad Extension
| "480" ; Temporarily not available
| "481" ; Call Leg/Transaction Does Not Exist
| "482" ; Loop Detected
| "483" ; Too Many Hops
| "484" ; Address Incomplete
| "485" ; Ambiguous
| "486" ; Busy Here
| "487" ; Request Cancelled
| "488" ; Not Acceptable Here
Server-Error = "500" ; Internal Server Error
| "501" ; Not Implemented
| "502" ; Bad Gateway
| "503" ; Service Unavailable
| "504" ; Gateway Time-out
| "505" ; SIP Version not supported
Global-Failure = "600" ; Busy Everywhere
| "603" ; Decline
| "604" ; Does not exist anywhere
| "606" ; Not Acceptable
message-header = field-name ":" [field-value] CRLF
field-name = token
field-value = *( field-content | LWS )
field-content = < the OCTETs making up the field-value and consisting of either *TEXT-UTF8
or combinations of token, separators, and quoted-string>
Accept = <定义在[RFC2068]>
Accept-Encoding = <定义在 [RFC2068]>
Accept-Language = <定义在 [RFC2068]>
Alert-Info = "Alert-Info" ":" #( "<" URI ">" *( ";" generic-param ) )
Allow = "Allow" ":" 1#Method
callid = token ["@" token]
Call-ID = ( "Call-ID" | "i" ) ":" callid
Call-Info = "Call-Info" ":" #( "<" URI ">" [ ";" "purpose" "=" ( "icon" | "info" | "card" | token ) ]
*( ";" generic-param ) )
Contact = ( "Contact" | "m" ) ":" ( "*" | ( 1#( ( name-addr | addr-spec ) *( ";"
contact-params ) ) ) )
name-addr = [display-name] "<" addr-spec ">"

```

addr-spec = SIP-URL | URI  
 display-name = \*token | quoted-string ; 显示名称  
 contact-params = "q" "=" qvalue | "action" "=" "proxy" | "redirect" | "expires" "=" delta-seconds  
 | DQUOTE SIP-date DQUOTE | contact-extension  
 qvalue = <定义在 [RFC2068]>  
 contact-extension = generic-param  
 Content-Disposition = "Content-Disposition" ":" disposition-type \*( ";" disposition-param )  
 disposition-type = "render" | "session" | extension-token  
 disposition-param = handling-param | parameter  
 parameter = <定义在 [RFC2068]>  
 handling-param = "handling" "=" ( "optional" | "required" | other-handling )  
 other-handling = token  
 Content-Encoding = ( "Content-Encoding" | "e" ) ":" ; 内容编码  
 Content-Language = <定义在 [RFC2068]> ; 语言  
 Content-Length = ( "Content-Length" | "l" ) ":" 1\*DIGIT ; 长度  
 Content-Type = ( "Content-Type" | "c" ) ":" media-type ; 类型  
 media-type = <定义在 [RFC2068]>  
 CSeq = "CSeq" ":" 1\*DIGIT Method  
 Date = "Date" ":" SIP-date ; 日期  
 SIP-date = rfc1123-date  
 rfc1123-date = <定义在 [RFC2068]>  
 Encryption = "Encryption" ":" encryption-scheme 1\*SP #encryption-params  
 encryption-scheme = token ; 加密模式  
 encryption-params = generic-param  
 generic-param = token [ "=" ( token | quoted-string ) ]  
 Expires = "Expires" ":" ( SIP-date | delta-seconds )  
 From = ( "From" | "f" ) ":" ( name-addr | addr-spec ) [ ";" tag-param ] \*( ";" addr-extension )  
 tag-param = "tag=" token  
 addr-extension = generic-param  
 Hide = "Hide" ":" ( "route" | "hop" | other-hide )  
 other-hide = token  
 In-Reply-To = "In-Reply-To" ":" 1#callid  
 Max-Forwards = "Max-Forwards" ":" 1\*DIGIT  
 Organization = "Organization" ":" TEXT-UTF8-TRIM  
 Priority = "Priority" ":" priority-value  
 priority-value = "emergency" | "urgent" | "normal" | "non-urgent" | other-priority  
 other-priority = token  
 Proxy-Authenticate = <定义在 [RFC2068]>  
 Proxy-Authorization = <定义在 [RFC2068]>  
 Record-Route = "Record-Route" ":" 1#name-addr [rr-extension]

rr-extension = token ["=" ( token | quoted-string )]  
 Require = "Require" ":" 1#option-tag  
 Response-Key = "Response-Key" ":" key-scheme 1\*SP #key-param  
 key-scheme = token  
 key-param = generic-param  
 Retry-After = "Retry-After" ":" ( SIP-date | delta-seconds ) [comment] [";" "duration" "="  
 delta-seconds]  
 Route = "Route" ":" 1#name-addr [route-extension]  
 route-extension = generic-param  
 Server = <定义在 [RFC2068]>  
 Subject = ( "Subject" | "s" ) ":" TEXT-UTF8-TRIM  
 Supported = ( "Supported" | "k" ) ":" 1#option-tag  
 Timestamp = "Timestamp" ":" \*( DIGIT ) [ "." \*( DIGIT ) ] [delay] ; 时间戳  
 delay = \*( DIGIT ) [ "." \*( DIGIT ) ] ; 延迟  
 To = ( "To" | "t" ) ":" ( name-addr | addr-spec ) [ ";" tag-param ] \*( ";" addr-extension )  
 Unsupported = "Unsupported" ":" 1#option-tag ; 不支持  
 User-Agent = <定义在 [RFC2068]>  
 Via = ( "Via" | "v" ) ":" 1#( sent-protocol sent-by \*( ";" via-params ) [comment] )  
 via-params = via-hidden | via-ttl | via-maddr | via-received | via-branch | via-extension  
 via-hidden = "hidden"  
 via-ttl = "ttl" "=" ttl  
 via-maddr = "maddr" "=" maddr  
 via-received = "received" "=" host [ ":" port ]  
 via-branch = "branch" "=" token  
 via-extension = token ["=" ( token | quoted-string )]  
 sent-protocol = protocol-name "/" protocol-version "/" transport  
 protocol-name = "SIP" | token ; 协议名称  
 protocol-version = token ; 协议版本  
 transport = "UDP" | "TCP" | token ; 传送  
 sent-by = ( host [ ":" port ] ) | ( concealed-host )  
 concealed-host = token  
 ttl = 1\*3DIGIT ; 最大生存时间, 为 0 到 255  
 Warning = "Warning" ":" 1#warning-value  
 warning-value = warn-code SP warn-agent SP warn-text  
 warn-code = 3DIGIT  
 warn-agent = ( host [ ":" port ] ) | pseudonym  
 adding  
 pseudonym = <定义在 [RFC2068]>  
 warn-text = quoted-string  
 URI = SIP-URL ; URL 地址

```

digest-uri-value = Request-URI
WWW-Authenticate = "WWW-Authenticate" ":" "pgp" pgp-challenge
pgp-challenge = #pgp-params
pgp-params = realm | pgp-version | pgp-micalgorithm | pgp-pubalgorithm | nonce
realm = "realm" "=" realm-value
realm-value = quoted-string
pgp-version = "version" "=" DQUOTE DIGIT *( "." DIGIT ) *letter DQUOTE
pgp-micalgorithm = "algorithm" "=" ( "md5" | "sha1" | token | "ripemd160" | "MD2" |
"TIGER192" | "HAVAL-5-160" )
pgp-pubalgorithm = "pubkey" "=" ( "rsa" | "rsa-encrypt" | "rsa-sign" | "elgamal" | "dsa" | token )
nonce = "nonce" "=" nonce-value
nonce-value = quoted-string
Authorization = "Authorization" ":" "pgp" #pgp-response
pgp-response = realm | pgp-version | pgp-signature | signed-by | nonce
pgp-signature = "signature" "=" quoted-string
signed-by = "signed-by" "=" DQUOTE URI DQUOTE
Encryption = "Encryption" ":" "pgp" pgp-eparams
pgp-eparams = 1#( pgp-version | pgp-encoding )
pgp-encoding = "encoding" "=" "ascii" | token
Response-Key = "Response-Key" ":" "pgp" pgp-eparams
pgp-eparams = 1#( pgp-version | pgp-encoding | pgp-key )
pgp-key = "key" "=" quoted-string
double-return = ( CR CR ) | ( LF LF ) | ( CR LF CR LF )
OCTET = %x00-FF ; any 8-bit sequence of data
CHAR = %x00-7F ; any US-ASCII character (octets 0 - 127)
upalpha = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P"
| "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
lowalpha = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" |
"r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
alpha = lowalpha | upalpha
DIGIT = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
alphanum = alpha | DIGIT
CTL = %x00-1F | %x7F ; (octets 0 -- 31) and DEL (127)
CR = %d13 ; 回车
LF = %d10 ; 换行
SP = %d32 ; 空白键
HT = %d9 ; 水平键
CRLF = CR LF ; 回车加换行
unreserved = alphanum | mark
mark = "-" | "_" | "." | "!" | "~" | "*" | "'" | "(" | ")"
escaped = "%" HEX HEX

```

```
LWS = [CRLF] 1*( SP | HT )
TEXT-UTF8 = *( TEXT-UTF8char | LWS )
TEXT-UTF8-TRIM = *TEXT-UTF8char *( *LWS TEXT-UTF8char )
TEXT-UTF8char = %x21-7E | UTF8-NONASCII
UTF8-NONASCII = %xC0-DF 1UTF8-CONT | %xE0-EF 2UTF8-CONT | %xF0-F7
3UTF8-CONT | %xF8-FB 4UTF8-CONT | %xFC-FD 5UTF8-CONT
UTF8-CONT = %x80-BF
HEX = "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f" | DIGIT
token = 1*( alphanum | "-" | "." | "!" | "%" | "*" | "_" | "+" | "`" | "'" | "~" )
separators = "(" | ")" | "<" | ">" | "@" | "," | ";" | ":" | "\" | DQUOTE | "/" | "[" | "]" | "?" | "=" | "{" |
| "}" | SP | HT
comment = "(" *( ctext | quoted-pair | comment ) ")"
ctext = < any TEXT-UTF8 excluding "(" and ">">
quoted-string = ( DQUOTE *( qdtext | quoted-pair ) DQUOTE )
qdtext = LWS | %x21 | %x23-5B | %x5D-7E | UTF8-NONASCII
quoted-pair = "\" CHAR
DQUOTE = <定义在[RFC2234]>
delta-seconds = <定义在 [RFC2068]>
```

2.2.4 SIP 时钟

SIP 协议中有多种不同的时钟，具体见表 2-2 所示。

表 2-2 SIP 时钟

时 钟	值	含 义
T1	500ms（默认）	RTT
T2	4s	非 INVITE 请求与 INVITE 响应之间的最大重传间隔
T4	5s	消息在网络中滞留的最长时间
时钟 A	初始化为 T1	INVITE 消息重传间隔（UDP）
时钟 B	64*T1	INVITE 事务超时时钟
时钟 C	> 3min	proxy INVITE 事务超时时钟
时钟 D	> 32s（UDP） 0s（TCP/SCTP）	响应等待时间
时钟 E	初始化为 T1	非 INVITE 请求重传间隔（UDP）
时钟 F	64*T1	非 INVITE 事务超时时钟
时钟 G	初始化为 T1	INVITE 响应重传间隔
时钟 H	64*T1	ACK 等待时间
时钟 I	T4（UDP） 0s（TCP/SCTP）	ACK 重传等待时间
时钟 J	64*T1（UDP） 0s（TCP/SCTP）	非 INVITE 请求等待时间
时钟 K	T4（UDP） 0s（TCP/SCTP）	响应重传等待时间

## 2.3 SDP 及 SIP 对 SDP 的要求

### 2.3.1 SDP 协议介绍与分析

上面介绍的协议用到了 SDP，实际上 SDP 就是用来描述多媒体会话通告，多媒体会话邀请和其他形式的多媒体会话初始化的协议。SDP 包通常包括以下信息：

#### 1. 会话信息

会话信息主要有：

- 会话名和目的。
- 会话活动时间。

由于参与会话的资源是受限制的，因此包括以下附加信息是非常有用的。

- 会话使用的带宽信息。
- 会话负责人的联系信息。

#### 2. 媒体信息

媒体信息主要有：

- 媒体类型，例如视频和音频。
- 传输协议，例如 RTP/UDP/IP 和 H.320。
- 媒体格式，例如 H.261 视频和 MPEG 视频。
- 多播地址和媒体传输端口（IP 多播会话）。
- 用于联系地址的媒体和传输端口的远端地址（IP 单播会话）。

SDP 描述由许多文本行组成，文本行的格式为<类型>=<值>，<类型>是一个字母，<值>是结构化的文本串，其格式依<类型>而定。

#### 3. SDP 语法

SDP 语法如下：

announcement := proto-version origin-field session-name-field information-field uri-field  
email-fields phone-fields connection-field bandwidth-fields time-fields key-field attribute-fields  
media-descriptions

proto-version := "v=" 1\*DIGIT CRLF ; 协议版本

origin-field := "o=" username space sess-id space sess-version space nettype space addrtype  
space addr CRLF

session-name-field := "s=" text CRLF ; 会话名称

information-field := ["i=" text CRLF] ; 信息

uri-field := ["u=" uri CRLF] ; URI 地址

email-fields := \*("e=" email-address CRLF) ; 电子邮件

phone-fields := \*("p=" phone-number CRLF)

connection-field := ["c=" nettype space addrtype space connection-address CRLF] ; 连接

bandwidth-fields := \*("b=" bwtype ":" bandwidth CRLF)

time-fields := 1\*( "t=" start-time space stop-time \*(CRLF repeat-fields) CRLF)



[zone-adjustments CRLF]

repeat-fields := "r=" repeat-interval space typed-time 1\*(space typed-time)

zone-adjustments := time space ["-"] typed-time \*(space time space ["-"] typed-time)

key-field := ["k=" key-type CRLF]

key-type := "prompt" | "clear:" key-data | "base64:" key-data | "uri:" uri

key-data := email-safe | "~" | "

attribute-fields := \*("a=" attribute CRLF)

media-descriptions := \*(media-field information-field \*(connection-field) bandwidth-fields

key-field attribute-fields )

media-field := "m=" media space port [ "/" integer ] space proto 1\*(space fmt) CRLF

media := 1\*(alpha-numeric) ;为 "audio", "video", "application" 或者 "data"

fmt := 1\*(alpha-numeric) ;为 RTP 载荷类型

proto := 1\*(alpha-numeric) ;典型的值为 "RTP/AVP" 或者 "udp"

port := 1\*(DIGIT) ;范围应在 1024 到 65535

attribute := (att-field ":" att-value) | att-field

att-field := 1\*(alpha-numeric)

att-value := byte-string

sess-id := 1\*(DIGIT) ;应该惟一

sess-version := 1\*(DIGIT) ;0 表示新的会话

connection-address := multicast-address | addr

multicast-address := 3\*(decimal-uchar ".") decimal-uchar "/" ttl [ "/" integer ] ;多播地址范围

为 224.0.0.0 到 239.255.255.255

ttl := decimal-uchar

start-time := time | "0"

stop-time := time | "0"

time := POS-DIGIT 9\*(DIGIT)

repeat-interval := typed-time

typed-time := 1\*(DIGIT) [fixed-len-time-unit]

fixed-len-time-unit := "d" | "h" | "m" | "s"

bwtype := 1\*(alpha-numeric)

bandwidth := 1\*(DIGIT)

username := safe

email-address := email | email "(" email-safe ")" | email-safe "<" email ">"

email := ;定义在 RFC822

uri := ;定义在 RFC1630

phone-number := phone | phone "(" email-safe ")" | email-safe "<" phone ">"

phone := "+" POS-DIGIT 1\*(space | "-" | DIGIT)

nettype := "IN"

addrtype := "IP4" | "IP6"

addr := FQDN | unicast-address

FQDN := 4\*(alpha-numeric"-"|"|".")  
unicast-address := IP4-address | IP6-address  
IP4-address := b1 "." decimal-uchar "." decimal-uchar "." b4  
b1 := decimal-uchar ; 小于 224, 且不能为 0 和 127  
b4 := decimal-uchar ; 不能为 0  
IP6-address :=  
text := byte-string  
byte-string := 1\*(0x01..0x09|0x0b|0x0c|0x0e..0xff) ; any byte except NUL, CR or LF  
decimal-uchar := DIGIT | POS-DIGIT DIGIT | ("1" 2\*(DIGIT)) | ("2" ("0"|"1"|"2"|"3"|"4")  
DIGIT) | ("2" "5" ("0"|"1"|"2"|"3"|"4"|"5"))  
integer := POS-DIGIT \*(DIGIT)  
alpha-numeric := ALPHA | DIGIT  
DIGIT := "0" | POS-DIGIT  
POS-DIGIT := "1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"  
ALPHA := "a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|"n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|"A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|"N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z" e  
mail-safe := safe | space | tab  
safe := alpha-numeric | "" | "" | "-" | "." | "/" | ":" | "?" | "" | "#" | "\$" | "&" | "\*" | ";" | "=" | "@" |  
 "[" | "]" | "^" | "\_" | "`" | "{" | "|" | "}" | "+" | "~" | "  
space := %d32  
tab := %d9  
CRLF := %d13.10

#### 4. 下一代 SDP

最新的 SDPng 扩展了 SDP, 使其可用于流服务、会议邀请、主从设备控制。与 SDP 相比, SDPng 提供了更为丰富的会话描述、可以更好地协商能力。但即使扩展了 SDP, 也不会影响其设计的简明性这一特征。

### 2.3.2 SIP 对 SDP 的要求

按照 rfc2327 的标准, 各消息的 SDP 部分应该有以下文本行。

#### 1. 协议版本

格式: v=0

意义: 给出 SDP 的版本号。

#### 2. 会话源

格式: o=<用户名><会话标识><版本><网络类型><地址类型><地址>

意义: 给出会话的发起者(用户名和用户地址)以及会话标识和会话版本号。

其中, 用户名为用户在主机的登录名, 可为“-”; 会话标识建议用 NTP 的时间戳; 版本号也建议用 NTP 的时间戳; 网络类型规定“IN”表示 Internet; 地址类型为“IP4”; 地址可以为域名或点分十进制数 IP4 地址表示形式。

### 3. 连接数据

格式: c = <网络类型><地址类型><地址>

意义: 给出连接数据。

其中, 网络类型规定 “IN” 表示 Internet; 地址类型为 “IP4”; 地址可以为域名或点分十进制数 IP4 地址表示形式。

### 4. 时间描述

格式: t=<起始时间><终止时间>

意义: 给出会话集获得时间区段。

其中, 起始时间和终止时间为 NTP 时间值的十进制数表示, 单位为秒。

### 5. 媒体级描述格式

格式: m=<媒体><端口><传送层><格式列表>

其中, 媒体可以是音频、视频、应用、数据、控制; 端口如果是 UDP 协议, 端口的取值范围 1024~65 535; 传送层定义了两种类型 RTP/AVP、udp; 格式列表对于音频和视频来说, 就是 RTP 音频/视频应用文档定义的媒体净荷类型。

### 6. 属性

格式: a = <属性>

a = <属性> : <值>

其中, 属性可以是会话级属性, 也可以是媒体级属性。常用的有 ptime, sendonly, recvonly 和 sendrecv 等。

## 2.4 实例

### 2.4.1 请求消息

以下为一请求消息的格式:

INVITE sip:bob@acme.com SIP/2.0 向 sip:bob@acme.com 发起呼叫, 协议版本号

Via: SIP/2.0/UDP alice\_ws.radvision.com 通过 proxy: alice\_ws.radvision.com

From: Alice A. 发起呼叫的用户标识

To: Bob B. 所要呼叫的用户

Call-ID:2388990012@alice\_ws.radvision.com 对这一呼叫的惟一标识

CSeq: 1 命令的序列号, 标识一个事件

Subject: Lunch today. 呼叫的名字或属性

Content-Type: application/SDP 消息体的类型

Content-Length: 182 消息体的字节长度

一个空白行标识消息头结束, 消息体开始

v=0 SDP 协议版本号

o=Alice 53655765 2353687637 IN IP4 128.3.4.5 会话建立者和会话的标识, 会话版本, 地址的协议类型, 地址

s=Call from Alice. 会话的名字

c=IN IP4 alice\_ws.radvision.com 连接的信息

M=audio 3456 RTP/AVP 0 3 4 5 对媒体流的描述：类型、端口，呼叫者希望收发的格式

通过以上的例子，可以对 SIP 协议有一个基本认识。除了在建立会话时进行各种消息交互外，SIP 终端还可以在会话过程中，发出消息改变或添加会话的某些属性。例如，用户在进行语音通话的过程中，想增加视频通信，可以在不中断通话的情况下，发送新的 INVITE 消息，打开双方的视频媒体，使通话变成可视。这为用户的使用带来很大的灵活性。

### 2.4.2 注册/注销过程

SIP 为用户定义了注册和注销过程，其目的是可以动态建立用户的逻辑地址和其当前联系地址之间的对应关系，以便实现呼叫路由和对用户移动性的支持。逻辑地址和联系地址的分离也方便了用户，它不论在何处、使用何种设备，都可以通过惟一的逻辑地址进行通信。

注册/注销过程是通过 REGISTER 消息和 200 成功响应来实现的。在注册/注销时，用户将其逻辑地址和当前联系地址通过 REGISTER 消息发送给其注册服务器，注册服务器对该请求消息进行处理，并以 200 成功响应消息通知用户注册/注销成功。

### 2.4.3 呼叫过程

SIP IP 电话系统中的呼叫是通过 INVITE 邀请请求、成功响应和 ACK 确认请求的三次握手来实现的，即当主叫用户代理要发起呼叫时，它构造一个 INVITE 消息，并发送给被叫。被叫收到邀请后决定接受该呼叫，就回送一个成功响应（状态码为 200）。主叫方收到成功响应后，向对方发送 ACK 请求。被叫收到 ACK 请求后，呼叫成功建立。

呼叫的终止通过 BYE 请求消息来实现。当参与呼叫的任一方要终止呼叫时，它就构造一个 BYE 请求消息，并发送给对方。对方收到 BYE 请求后，释放与此呼叫相关的资源，回送一个成功响应，表示呼叫已经终止。

当主、被叫双方已建立呼叫，如果任一方想要修改当前的通信参数（通信类型、编码等），可以通过发送一个对话内的 INVITE 请求消息（称为 re-INVITE）来实现。

### 2.4.4 重定向过程

当重定向服务器（其功能可包含在代理服务器和用户终端中）收到主叫用户代理的 INVITE 邀请消息，它通过查找定位服务器发现该呼叫应该被重新定向（重定向的原因有多种，如用户位置改变、实现负荷分担等），就构造一个重定向响应消息（状态码为 3xx），将新的目标地址回送给主叫用户代理。主叫用户代理收到重定向响应消息后，将逐一向新的目标地址发送 INVITE 邀请，直至收到成功响应并建立呼叫。如果尝试了所有的新目标都无法建立呼叫，则本次呼叫失败。

### 2.4.5 能力查询过程

SIP IP 电话系统还提供了一种让用户在不打扰对方用户的情况下查询对方通信能力的手段。可查询的内容包括：对方支持的请求方法（methods）、支持的内容类型、支持的扩展项、支持的编码等。

能力查询通过 OPTION 请求消息来实现。当用户代理想要查询对方的能力时，它构造一个 OPTION 请求消息，发送给对方。对方收到该请求消息后，将自己支持的能力通过响应消

息回送给查询者。如果此时自己可以接收呼叫，就发送成功响应（状态码为 200），如果此时自己忙，就发送自身忙响应（状态码为 486）。因此，能力查询过程也可以用于查询对方的忙闲状态，看是否能够接受呼叫。

## 2.5 SAP 协议

SAP 的全称是会话通告协议，其目的是为了通知一个多播的多媒体会议或其他多播会话而将相关的会话建立信息发送给所期望的会议参与者。SAP 协议本身并不建立会话，它只是将建立会话所必要的信息（例如，将所采取的视频或音频编码方式）通知给其他在一个多播组内的参与者，当参与者接收到该通知数据包后就可以启动相应的工具，并设置正确的参数向该会议的发起者建立会话了（建立会话可以使用 SIP 协议）。

通知的发起者并不知道各参与者是否收到了会话通知，也就是说每个参与者并不向通知发起者回复“我收到了通知”的确认。因此，通知发起者只能够通过周期性地发送这个会话通知从而最大可能地使参与者收到通知。

SAP 并不是向每个参与者一一发通知数据包，它是通过多播的机制（multicast）向一个已知的多播地址和端口一次性发送一个通知数据包，该多播组内的成员如果工作正常的化就会收到该通知数据包。因此，为了使会议的参与者都能够接收到通知，就要确保其参加到该多播组内。

一个通知数据报除了可以通知某会话将要发起外，还可以通知该会话取消了或该会话的某些通信参数已被修改了。当然，这需要相应机制使这几个通知都是针对同一会话的。

那么 SAP 如何描述会话的相关信息，这就需要借助 SDP 协议了。在 SAP 数据包的 payload 字段中一般情况下填充的就是 SDP 数据，它描述了建立会话所必要的基本信息。

## 2.6 小结

本章介绍了 SIP 的基本知识，如 SIP 基本消息、基本头、响应及基本的 SIP 系统组成。分析了基本的 SIP 和 SDP 语法，还介绍了 SDPng 等内容。最后详细分析了一些典型的 SIP 消息格式。

## 2.7 习题

1. 简述 SIP 的系统构成。
2. 简述 SIP 基本消息、基本头及响应。
3. 分析 SIP 对 SDP 的要求。
4. 试用抓包软件（如 ethereal）在 SIP 网络进行抓包，并分析 SIP 报文。

## 第3章 SIP 协议相关扩展

知识点:

- SIP 扩展原则
- SIP 事件扩展
- SIP QoS 扩展
- SIP 安全扩展
- SIP IM 扩展
- SDP 相关扩展
- SIP 其他扩展

SIP 协议的基本设计思想是将协议的基本功能与功能扩展功能相分离,前者构成稳定而相对简单的 SIP 基础标准,后者源于增值业务的需要,对应为一系列草案文件,经过一定时间的运行试验,成熟的扩展功能将纳入基础标准。本章将介绍 SIP 扩展的原则及其具体的 SIP、SDP 扩展。

### 3.1 SIP 扩展介绍

当扩展 SIP 时,必须遵循一定的扩展原则。扩展 SIP 时,不能破坏 SIP 简单性和可管理性;不能破坏 SIP 对等关系;必须保持 SIP 会话建立过程与 SIP 会话描述之间的独立性;不要改变原有方法的语义。

### 3.2 传送 ISUP 信令的扩展

#### 3.2.1 SIP-T 介绍

SIP-T (SIP for Telephones) 是基于 SIP 的一种扩展协议,用于在 IP 网间传送 ISUP 信令,以解决 PSTN 信令在 IP 网上的协议转换和传送问题。SIP-T 将传统电话网信令(目前仅对 ISUP 消息)通过“封装”和“翻译”转化为 SIP 消息,提供了用 SIP 实现传统 PSTN 网与 SIP 网络的互连机制。

#### 3.2.2 SIP-I 协议

SIP-I (Session Initiation Protocol with Encapsulated ISUP) 由 ITU-T SG11 制定, SIP-I 协议系列包括 TRQ、BICC/ISUP SIP 和 Q.1912.5。前者定义了 SIP 与 BICC/ISUP 互通时的技术需求,包括互通接口模型、互通单元 IWU 所应支持的协议能力集、互通接口的安全模型等。后者根据 IWU 在 SIP 侧的 NNI 上所应支持的协议能力配置集 A、B、C,详细定义了 3GPP SIP 与 BICC/ISUP 的互通,一般情况下 SIP 与 BICC/ISUP 的互通,等等。

SIP-I, 基本上与 SIP-T 相同。但从 ISUP 封装角度讲, SIP-I 不需要使用 S/MIME; SIP-I 需要接收方能够理解 ISUP (必须有 Content-Disposition 处理参数); 在 SIP overlap 信令中, 封装在每个 INVITE 中的是更新过的 IAM, 而不是 IAM+SAM。

从 SIP/ISUP 消息映射来看, cause code 映射不相同; Q.1912.5 使用 pre-conditions, 而 RFC3578 没有使用; 关于 ISUP 参数, Q.1912.5 比 RFC3578 更详细。

最重要的是, SIP-I 协议系列具有 ITU-T 标准固有的清晰性、准确性, 可操作性非常强。

## 3.3 穿越防火墙

### 3.3.1 概述

为了尽可能地减少对 IPv4 地址的耗尽, 通常, SIP 用户代理处于私网内部, 配置为私有 IP 地址, 网络出口配置了 NAT/FW 来提供网络地址和端口转换功能。私网可以是一级私网或多级私网, 在多级私网模型下, 第  $N$  级以上级私网通常为提供 Internet 接入的 ISP 网络, 且 SIP 用户代理可位于不同级私网内部。因此, 处于私网内部的 SIP 用户代理与网外的其他终端用户或媒体服务器发生通信时, 应实现 SIP 协议的 NAT/FW 穿越和 RTP/RTCP 流的 NAT/FW 穿越。

根据 NAT 类型不同, NAT 可分为静态 NAT、动态 NAT 和 NAT。根据 NAT 工作方式不同, NAT 类型可分为 Full Cone 方式 NAT、Restricted Cone 方式 NAT、Port Restricted Cone 方式 NAT 和 Symmetric Cone 对称方式 NAT, 具体可参见 RFC3489。因此, SIP 协议的 NAT 穿越应支持多种 NAT 类型和多种 NAT 工作方式的穿越。

### 3.3.2 SIP 协议穿越 NAT/FW 存在的问题

#### 1. 数据包地址与载荷部分地址不一致

在 SIP 协议的呼叫建立过程中, 消息接收方始终按照协议载荷内的源地址和端口信息来回传响应消息和发送相应的媒体流。当 H.323 终端和 SIP 终端通过 NAT/FW 进行地址转化时, NAT/FW 会将 IP 包头中的源 IP 地址和源端口号转换为 NAT/FW 的网络出口地址, 从而造成载荷内的地址和端口 (如 SDP 部分中的地址和端口) 与 IP 包头的源地址和源端口的不一致。此时, 消息接收方无法将响应消息正确回传到消息发送方, 主被叫方无法建立正常通信连接。

#### 2. 媒体流端口动态分配

SIP 终端所使用的媒体流端口需要通过 SIP 信令呼叫流程来进行动态协商。因此, 动态创建的媒体流端口之间的连接将被防火墙阻断。

#### 3. 防火墙端口主动开放

防火墙配置是允许内部用户主动访问外部, 不允许外部用户主动访问内部。因此, 在不采用相应策略的情况下, 网外用户发起到网内的呼叫将被拒绝, 即网内用户只允许做主叫不允许做被叫。如果允许网内用户做被叫, 则防火墙需要增加相应策略, 开放几乎所有网外 IP 地址, 从而给内部网络带来严重的安全隐患。

### 3.3.3 参考模型

支持 SIP 协议的 NAT/FW 穿越可分为信令和媒体穿越两部分, 信令穿越和媒体穿越的功

能可以分别由信令中继功能（SRF）和媒体中继功能（MRF）来完成，当 SRF 和 MRF 采用独立的物理实体来实现时，SRF 和 MRF 之间的接口协议可采用 MIDCOM 协议或其他协议（其他协议待定）。当采用信令和媒体穿越分离的模型时，SRF 功能可以集成于服务器中，MRF 功能可以集成于 NAT 设备中。

信令和媒体穿越不分离时，可采用代理技术和隧道技术。

### 3.3.4 协议扩展

SIP 终端利用私有 IP 地址有两种情况，一种为主叫用私有 IP 地址发起请求和接受响应，另一种为被叫通过私有 IP 地址接受请求和回送响应。

第一种情况，当位于 NAT 后的 SIP 终端与外界通信时，需要在 SIP 终端发出的 INVITE 请求中加入扩展字段 `rport`，表明需要穿透 NAT。经过 NAT 处理后的消息送到 SIP 代理服务器，代理服务器发现 SIP 消息中含有 `rport` 字段，从而确认 INVITE 请求经过了 NAT，SIP 服务器记录经过 NAT 后临时分配的合法 IP 地址和端口号。这时需要另外一个扩展字段 `received`，并将 NAT 临时分配的合法 IP 地址和端口号分别放到 `received` 和 `rport` 中。以后当 NAT 网关收到代理的消息后，将查找相应 NAT 对应表，将 UDP 包中的目的地址和端口替换成内部的私有地址，再转发给内部网络。

第二种情况，则需要添加新的字段 `Translate`，以起翻译作用。

## 3.4 临时应答的可靠传输

SIP 应答分为正常应答和临时性应答，对于前者可保证可靠传输，而对于后者，RFC 3261 并未保证可靠传输，RFC 3262 定义了用于可靠传输临时应答的 SIP 扩展。这种新的扩展称为 PRACK（Provisional Response ACK），UAS 会不断重传临时应答，直到收到 UAC 的确认为止，一旦 UAS 收到 UAC 收到的 PRACK，便停止重传。但 PRACK 自身也需要确认，所以 UAS 也必须为 PRACK 发送一个 200 应答，表明 PRACK 成功。目前，IANA 为 PRACK 分配了标志：100rel。”

## 3.5 用户偏好扩展

不同的呼叫者可能具有不同的个人偏好，为此 SIP 扩展中定义了三个新的标题头和 Contact 参数，其适用范围如表 3-1 和表 3-2 所示。

表 3-1 Accept-Contact、Reject-Contact 和 Request-Disposition 使用范围 1

头 域	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Accept-Contact	R	ar	o	o	o	o	o	-
Reject-Contact	R	ar	o	o	o	o	o	-
Request-Disposition	R	ar	o	o	o	o	o	o

表 3-2 Accept-Contact、Reject-Contact 和 Request-Disposition 使用范围 2

头 域	where	proxy	PRA's	UPD	SUB	NOT	INF	MSG	REF
Accept-Contact	R	ar	o	O	o	o	O	o	o
Reject-Contact	R	ar	o	o	o	o	o	o	o
Request-Disposition	R	ar	o	o	o	o	O	o	o



三个头的 BNF 语法描述如下:

### 1. Request-Disposition

Request-Disposition = ("Request-Disposition" / "d" ) HCOLON directive \*(COMMA directive)

其中, directive=proxy-directive/cancel-directive /fork-directive/recurse-directive /parallel-directive / queue-directive)

proxy-directive= "proxy" / "redirect"

cancel-directive= "cancel" / "no-cancel"

fork-directive = "fork" / "no-fork"

recurse-directive = "recurse" / "no-recurse"

parallel-directive= "parallel" / "sequential"

queue-directive = "queue" / "no-queue"

### 2. Accept-Contact

Accept-Contact = ("Accept-Contact" / "a") HCOLON ac-value\*(COMMA ac-value)

其中, ac-value = "\*" \*(SEMI ac-params)

rc-value = "\*" \*(SEMI rc-params)

ac-params = feature-param / req-param/ explicit-param / generic-param

rc-params = feature-param / generic-param

req-param = "require"

explicit-param = "explicit"

### 3. Reject-Contact

Reject-Contact = ("Reject-Contact" / "j") HCOLON rc-value \*(COMMA rc-value)

各参数的含义同 Accept-Contact。

## 3.6 INFO 扩展

### 3.6.1 INFO 简介

#### 1. INFO 目的

没有通用的目标机制来在会话过程中沿着 SIP 信令通路承载会话控制信息。INFO 消息的目的是沿着 SIP 信令通路携带应用层消息。INFO 方法并不是用来改变 SIP 呼叫的状态或会议 SIP 的初始化状态参数。它仅是用于发送通常与会议有关的应用层的可选信息。

会议中的信号信息传过会议后的 SIP 信令通路的建立是必要的。这个通路是 SIP 的 re-INVITEs、BYEs 和其他与一个独立会话联系的 SIP 请求所采用的。它允许 SIP 代理服务器接收并潜在地对会议中的信号信息起作用。

#### 2. 用法举例

以下是一些 INFO 消息的可能应用:

- 在 PSTN 网关之间传送呼叫中的 PSTN 信令消息。
- 传送 SIP 会议中生成的 DTMF 数字。

- 传送无线信号强度信息以支持无线移动应用。
- 在会议的参加者之间传送影像或其他的信息。

### 3.6.2 INFO 方法

INFO 方法被用于沿着呼叫的信令通路进行会议中信令消息间的通信。INFO 方法并不是用于改变 SIP 呼叫的状态，也不是用于改变被 SIP 初始化的会议状态。然而，它提供增加的选项信息可以进一步加强 SIP 的应用程序功能。

INFO 方法的信令通路是呼叫建立之后建立的信令通路。这可以是呼叫方与被呼叫方用户代理之间的直接信令，也可以是包括牵涉到呼叫建立和自己增加到初始 INVITE 信息记录路由头部的 SIP 代理服务器的信令通路。

会议中信息能够在 INFO 信息头部或作为一个消息体的一部分来进行通信。消息体和/或消息头部的定义被用来传送在本文档讨论范围之外的会议中信息。

#### 1. INFO 请求方法的响应

如果服务器收到一个 INFO 请求，其必须发出一个最后的回应。如果 INFO 请求被现有的呼叫成功地接收到，则 UAS 必须发送一个 200 OK 没有消息体的回应给一个 INFO 请求。除此之外，不需要其他的操作。INFO 相关的头使用范围如表 3-3 所示。

表 3-3 INFO 相关头使用范围

头部 (Header)	地方 (Where)	信息 (INFO)
接收 Accept	R	o
接收-编码 Accept-Encoding	R	o
接收-语言 Accept-Language	R	o
允许 Allow	200	-
允许 Allow	405	o
认可 Authorization	R	o
呼叫号 Call-ID	gc	m
连接 Contact	R	o
连接 Contact	1xx	-
连接 Contact	2xx	-
连接 Contact	3xx	-
连接 Contact	485	-
连接-编码 Content-Encoding	e	o
内容-长度 Content-Length	e	o
内容-类型 Content-Type	e	*
CSeq	gc	m
数据 Date	g	o
加密 Encryption	g	o
期满 Expires	g	o
From	gc	m
隐藏 Hide	R	o
最大-向前流 Max-Forwards	R	o
组织 Organization	g	o
优先权	R	o

续表

头部 (Header)	地方 (Where)	信息 (INFO)
Proxy 验证	407	o
Proxy 验证	R	o
Proxy-需求	R	o
请求	R	o
重试之后	R	-
重试之后	404,480,486	o
重试之后	503	o
重试之后	600,603	o
回应-关键字	R	o
记录-路由	R	o
记录-路由	2xx	o
路由	R	o
服务器	r	o
主体	R	o
时间戳	g	o
To 到	gc(1)	m
不支持的	420	o
用户代理	g	o
Via	gc(2)	m
告警	r	o
WWW-验证	401	o

如果 INFO 请求与任何现存的呼叫 leg 不匹配, 那么一个 481 呼叫 Leg/Transaction 不存在的消息必须在一个 UAS 中被发送。

如果一个服务器收到一个其能理解消息体的 INFO 请求, 但是它又对与 INFO 过程有关的消息体规则没有一点了解, 那么这个消息体可能被翻译并显示给用户。这个 INFO 被一个 200 OK 所回应了。

如果 INFO 请求包括一个服务器不能理解的消息体, 那么在 INFO 相关的消息体的进程规则缺乏时, 服务器必须回应一个 415 不支持的媒体类型消息。

那些在 SIP 呼叫状态中或被 SIP 初始化后的会议中完成一个改变的消息体不能被放在一个 INFO 消息中发送。

其他请求失败 (4xx)、服务器失败 (5xx) 和全局失败 (6xx) 回应将被送给 INFO 请求。

## 2. SIP 用户代理的行为

INFO 请求的协议规则控制了标记 (tags) 的用法。

如果一个最终的回应没有被送给 INFO, 那么一个 UAS 的取消 (CANCEL) 将用一个 “487 请求已取消” 回应给 INFO。然而, INFO 消息决不许改变 SIP 呼叫的状态或 SIP 初始化的回应。

### 3.6.3 INFO 消息体

INFO 消息的目的是在 SIP 用户代理间传送会议中的信息。这一信息尽管能够在 INFO 信息头中传送, 但它一般将被放在消息体中传送。

另外, INFO 方法并不定义确保按顺序传送的附加机制。

### 3.6.4 利用 INFO 扩展的指导方针

以下是在定义利用 INFO 方法的 SIP 扩展时必须考虑的几个方面:

- 被 INFO 消息传送的消息体的大小必须要考虑。由于消息将可能在 UDP 上传送, 并且可能重组一个大的消息, 消息体应该保持较小。
- 有一种可能是 INFO 消息能被一个 SIP 代理服务器创建。完成这一 INFO 消息中的信息的创建过程需要被考虑。
- 当定义被 INFO 消息传送的消息体时, 大部分消息体的应用将会很有用。
- 用 INFO 消息的扩展不允许依靠 INFO 消息做那些影响 SIP 呼叫的状态或相关会议的状态的事。
- 本文档定义的 INFO 扩展不依赖于请求或代理请求头部的应用。用 INFO 消息的扩展名可能需要应用这些机制。然而, 有可能请求或代理请求的应用最好能避免, 以便在 SIP 实体间可以互操作。

## 3.7 SIMPLE

即时消息/存在技术目前到了一个转折点。缺少基于标准的可互操作的 IM/存在 (Presence) 系统造成难于控制和监测这种流行业务工具的部署。专有网络和协议也阻碍 IM 用户与所在单位之外的其他人相互通信。

用户需要的是一种实现 IM/存在技术的统一的协议, 其作用正如简单邮件传输协议、HTTP 和实时协议 (RTP) 对于电子邮件、Web 和语音流的作用。解决方案是“即时消息与存在利用扩展会话初始协议” (SIMPLE)。

Internet 工程任务组的 SIMPLE 工作组受命定义一套 SIP 扩展。该小组预计将于今年公布一项建议标准。IETF 的 IM 与存在协议工作组在 RFC 2778 和 2779 中公布了对 IM/存在技术的一般要求和它的模型。

### 1. 简单方式

SIMPLE 在本质上与 SIP 相同: SIP 没有采用 GET 和 POST 等数据索取方式, 而采用 INVITE 和 BYE 等信令方式来启动和结束一次呼叫或会话。

SIMPLE 增加了一种叫做 MESSAGE 的新的请求方式来发送所谓寻呼模式 IM 的一次性 IM。SUBSCRIBE 被用于请求将存在信息发送给请求方, 而 NOTIFY 则被用于传输存在信息。

通信各方在一段时间交换多条消息的更长时间的 IM 会话利用 INVITE 和一种名为“消息会话中继协议” (MSRP) 的传输协议来启动。当与 SIMPLE 一同使用时, MSRP 用于传送 IM 的文本, 就像在 SIP 中 RTP 用于传送一次 IP 电话呼叫中的语音包。

IM/Presence 基础设施中的许多部分可重复利用 SIP。例如, 一个 IM 客户机向 SIP 注册服务器发送 REGISTER 消息, 通知它可以接收 IM。正如在普通 SIP 系统中一样, 注册服务器处理来自端点的登录。

### 2. 传播消息

IM 客户机直接或通过 SIP 代理服务器和 SIP 重定向服务器, 向其他每一个 IM 客户机发

送实际的 IM 流和最新的存在信息。SIP 代理服务器在 SIP 系统元素（如 SIP 电话）之间转发 SIP 请求，重定向服务器则被用于告之客户机有关已经移动的通信方的信息。

IM 客户机利用 MIME 发送多媒体请求。由于 SIP 被设计为可方便地像对一个端点传送信令那样，向一组端点传送信令，因此多方 IM 和聊天室已经得到了支持。

IM/存在与 SIP 的关系就像是 SMS 与移动电话信令系统的关系。SMS 利用移动电话网搭载文本消息，IM/存在则搭载在 SIP 上，SIP 就是 Internet 格式的电话信令。

### 3.8 SIP 多方会议扩展

多媒体会议系统中传输的信息流包含音频、视频、数据和控制信息，会议系统主要包含两个部分内容：（1）使得参与者可以加入或者离开会议；（2）会议控制，其中包括会议管理（如创建、修改和删除会议）、用户管理（增加和删除会议参与者，修改他们的属性）和底层（Floor）管理。会议系统中各种元素的不同组织方式构成了不同类型的会议模型，一般会将会议系统分为 3 种类型，即集中型、分散型和混合型。

图 3-1 为基于多代理服务器的会议模型，为了方便起见，以会议模型中的拨入模型讨论会议的一致性。对于媒体服务器而言，它接收多个 Proxy 的控制消息，并维护与每个加入的 Proxy 的连接。在有参与者加入会议时，Proxy 通知媒体服务器有新的 UA 加入，以便媒体服务器组织媒体流和数据信息，同时要让新的参与者知道这是一个会议，有若干个用户代理（UA）在会议中。这里不采用中心会议成员服务器，当 UA 呼叫某个会议时，Proxy 对定位服务器查询，确定处于会议中的其他 Proxy，然后通过与其他 Proxy 的信息交流，通知其他成员有新成员加入。当然，仅使用原来的 SIP 方法是不够的，必须对 SIP 功能扩展，以使得 SIP 支持多代理的多媒体会议。

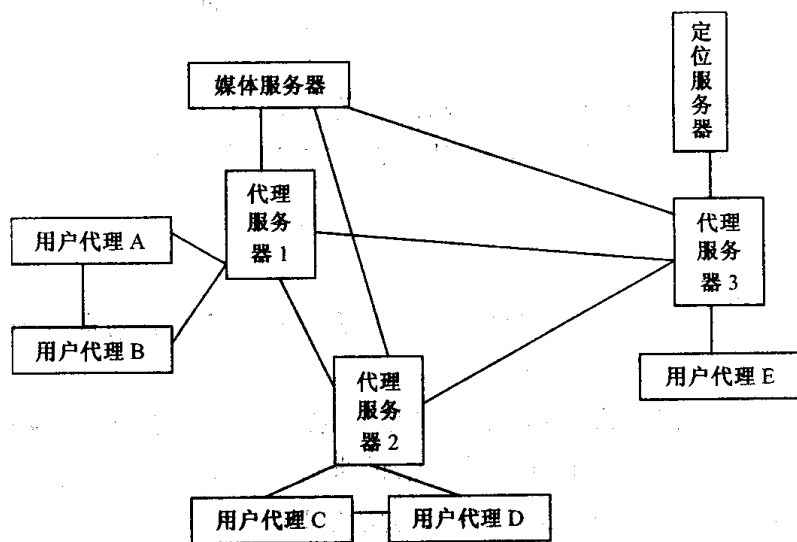


图 3-1 代理多方会议模型

在由 A、B、C、D 和 4 个 UA 组成的会议中，Proxy1、Proxy2 也为会议的组成部分，当 UA E 要加入会议时，E 向其本地 Proxy3 发出 INVITE 消息，Proxy3 向 Location Service 定位媒体服务器的地址，在 E 加入会议之后，E 将向媒体服务器发送媒体流，并从媒体服务器接收媒体流。

Proxy3 同时向 Proxy1 和 Proxy2 发送用户 E 加入会议的消息，Proxy1 和 Proxy2 向各自的状态成员发送 E 加入会议的消息，在得到各自参与者的 ACK 后，再向 Proxy3 发送 ACK。如

果某个状态成员在一定时间内没有返回 ACK, 则认为该用户不愿意让其他参与者知道其存在。

## 3.9 SIP 异步事件扩展

### 3.9.1 基于会话启动协议的事件通告机制

#### 1. 事件通告机制的概念

所谓事件通告机制是指网络中的一些实体可以订阅网络中某些资源或呼叫的状态信息, 当那些被订阅的资源的状态发生改变时, 负责这一资源的网络实体将向订阅者发送通告, 通报当前资源状态的变化情况。

为了实现这一机制, Internet 工程任务组 (IETF) 的 SIP 工作组对基本的会话启动协议进行了扩充, 提出了基于会话启动协议的事件通告机制规范: RFC 3265。在规范中定义了两个扩展方法: 订阅 (SUBSCRIBE) 和通告 (NOTIFY)。SUBSCRIBE 方法用于发起订阅请求, NOTIFY 方法用于通告当前资源状态。

会话启动协议事件通告机制涉及以下几个概念:

#### ◆ 订阅者

订阅者负责接收 NOTIFY 消息的会话启动协议用户代理 (SIP UA)。这些 NOTIFY 消息中包含订阅者订阅的资源信息。订阅者典型的动作是向通告者发送 SUBSCRIBE 消息以请求创建一次订阅关系。

#### ◆ 通告者

通告者负责产生 NOTIFY 请求的 SIP UA。通告者在 NOTIFY 消息中向订阅者回馈当前资源的状态。通告者典型的动作是接收 SUBSCRIBE 消息并创建相应的订阅关系。

#### ◆ 订阅

所谓订阅就是一组与某个对话相关联的应用状态的集合。订阅关系既存在于订阅者中, 又存在于通告者中。

#### ◆ 事件包

事件包是通告者向订阅者发送的一组资源的状态信息。RFC 3265 中给出了抽象的事件包模板定义, 对应具体业务可定义相应的事件包类型, 例如: 在席事件包、对话事件包等, 这些事件包可使用不同的语法并具有各自的语义。这种框架赋予会话启动协议事件通告机制极大的生命力和灵活性, 有助于快速提供新的业务。

#### 2. 事件通告机制的流程

典型的会话启动协议事件通告机制流程如图 3-2 所示。

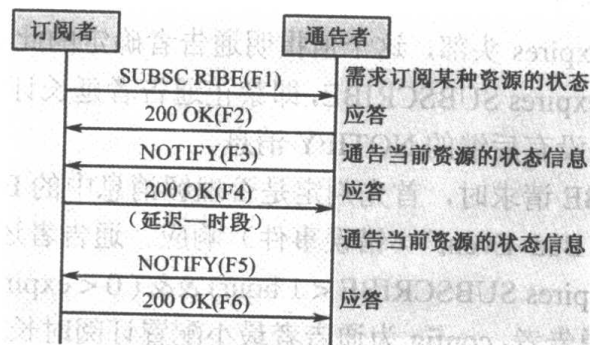


图 3-2 基于 SIP 的事件通告流程

**SUBSCRIBE** 方法和会话启动协议基本规范中定义的邀请 (**INVITE**) 方法都可以创建一个对话。当订阅者想得到网络中某一资源的状态时, 便向负责这一资源的会话启动协议实体发起 **SUBSCRIBE** 请求, 如图 3-2 中的 F1 所示。**SUBSCRIBE** 消息中的请求统一资源标识符 (**Request-URI**) 就是所要请求的资源的统一资源标识符 (**URI**), 这一 **URI** 同时还为会话启动协议代理服务器路由请求提供线索。**SUBSCRIBE** 请求中必须包含一个扩展的 **Event** 头部, 其中注明要订阅的事件类型, 即事件包标记, 如, **dialog** (用于代答业务)、**refer** (用于呼叫转交) 等。还可包含扩展的 **Allowed-Event** 头部, 指示本节点能够支持的事件包类型。如果在一个对话中有多次订阅, 则如图 3-3 所示, 在 **Event** 头部还要增设标识参数 **ID** 予以区分。

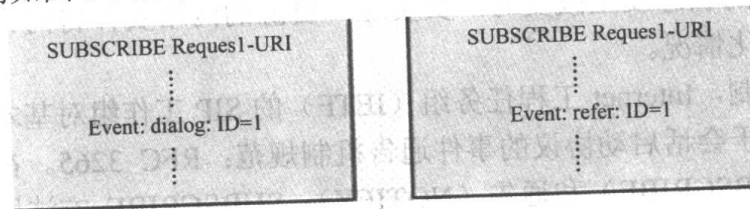


图 3-3 同一会话中多个订阅请求

对于订阅者来说, 它总是在一定的时间段内对它感兴趣的某一资源进行观察。因此, **SUBSCRIBE** 消息中应包含 **expires** 头部, 这一头部值表明订阅者期望的有效订阅时长。为了延长某一订阅的时间, 订阅者可以在有效期内再次发送 **SUBSCRIBE** 消息来刷新这一订阅。具体某次订阅的有效时长, 最终是由对 **SUBSCRIBE** 请求的 **2XX** 响应中的 **expires** 头部值或 **NOTIFY** 消息中的 **Subscription-State** 头部的 **expires** 参数决定的。**expires** 头部值等于 0 的 **SUBSCRIBE** 请求表示撤销订阅。如果订阅关系能够建立, **SUBSCRIBE** 消息将会触发通告资源状态的 **NOTIFY** 消息立即回送。订阅者想要获得的资源状态信息封装在后继通告消息 **NOTIFY** 的消息体中, 为了能够正确地解释这部分信息, 订阅者应该向通告者指明自己支持的消息体格式, 因此, 在 **SUBSCRIBE** 消息中应携带 **Accept** 头部, 例如, **Accept: application/dialog-info+xml**, 这表明订阅者支持用可扩展标识语言 (**XML**) 描述的对话事件包, 实际上就是一种通用 Internet 邮件扩展 (**MIME**) 格式消息体。如果 **SUBSCRIBE** 消息中没有携带 **Accept** 头部, 则通告者根据 **SUBSCRIBE** 消息中 **Event** 头部指明的事件包标记选择默认的格式传送资源状态信息。

**SUBSCRIBE** 请求通过 **2XX** 响应确认, 如图 3-2 中的 F2 所示。不同的 **2XX** 响应具有不同的语义:

- 200 OK 表示订阅已被接受且用户已被授权订阅请求的资源。
- 202 Accepted (接受) 是事件通知机制扩展的响应码, 表示订阅请求已被理解, 但是否授权给订阅者未确定。

**2XX** 响应中必须包含 **expires** 头部, 这个值指明通告者确定的此次订阅的有效时长, 且必须满足:  $\text{expires } 200 \text{ OK} \leq \text{expires SUBSCRIBE}$ , 即禁止通告者延长订阅时长。如果返回非 **2XX** 响应, 则表示订阅失败, 将没有后继的 **NOTIFY** 消息。

通告者收到 **SUBSCRIBE** 请求时, 首先判定是否理解消息中的 **Event** 头部, 如果不理解, 则通告者返回扩展的 “489 Bad Event” (错误事件) 响应。通告者还会检查消息中的 **expires** 头部, 如果其值满足:  $(0 < \text{expires SUBSCRIBE} < 1 \text{ hour}) \ \&\& \ (0 < \text{expires SUBSCRIBE} < \text{Min 通告者-config})$ , 其中, **Min 通告者-config** 为通告者最小配置订阅时长, 则通告者向订阅者返回 “423 Interval too small” (时间间隔太短) 响应, 表示提出的订阅时长太短。此外, 通告者还

应根据本地策略对提出 SUBSCRIBE 请求的用户进行鉴权。

与订阅相关的信息由 NOTIFY 消息传送。通告者在成功创建订阅关系后,必须立即发送 NOTIFY 消息,向订阅者通告当前订阅资源的状态,如图 3-2 中的 F3 所示。通告者使用 SUBSCRIBE 消息中 Accept 头部明确允许的或者 Event 头部隐含指定的消息体格式将资源的状态信息或指向该资源状态的 URI 封装在消息中。消息也可包含扩展的 Allowed-Events 头部,指示本节点能够支持的事件包类型。NOTIFY 消息中必须包含扩展的 Subscription-State 头部,指示创建的订阅状态。共有 3 种订阅状态,分别是:

- active: 订阅已被接受且授权成功。
- pending: SUBSCRIBE 请求已收到,但还没有足够的信息决定接受或拒绝此次订阅。
- terminated: 订阅未激活,或创建的订阅关系终止。

对应 active 状态或 pending 状态,该头部还带有 expires 参数指示此次订阅的有效时长。对应 terminated 状态,该头部应包含 reason 参数指示订阅被终止的原因,或者包含 Retry-After 参数,指示订阅者过一段时间后可重新发起订阅请求。

订阅者收到通告者 NOTIFY 请求后,将进行匹配检查。

如果找到相应的匹配,且 NOTIFY 消息中的 Subscription-State 头部指示的订阅状态是 active 或 pending,则订阅者创建新的订阅或对话,并对 NOTIFY 请求回送 200 OK 响应,如图 3-2 中的 F4 所示。如果匹配失败,则发送“481 Subscription does not exist”(订阅不存在)响应。

在订阅有效期内,如果资源状态发生变化,则通告者使用 NOTIFY 请求及时将变化信息通告订阅者,如图 3-2 中的 F5 和 F6 所示。

扩展的 SUBSCRIBE 方法和基本的 INVITE 方法都可以创建对话。在某一对话中,SUBSCRIBE 请求将创建和该对话关联的订阅,而该对话有可能是由 INVITE 建立起来的。因此,如果订阅终止,且当前无其他应用状态(如由 INVITE 请求建立起来的应用状态)和该对话关联,则该对话结束。如果仍有订阅和该对话关联,虽然其他的应用状态已结束,但该对话并没有结束。换句话说,由 INVITE 创建的对话并不会因为收到或发送了再见请求而结束,因为仍有订阅关系和此对话相关联。与此类似,当多个订阅和同一对话关联时,必须当与此对话相关联的所有订阅都结束时,该对话才结束。这一概念非常类似于程序设计里对文件描述符的引用,其中文件描述符相当于对话,对文件描述符引用的进程相当于建立的订阅关系。

SUBSCRIBE 请求可以触发通告者对其资源状态的立即回送,因此,订阅者可以利用这一特性实现对资源状态的轮询。当订阅处于激活状态时,订阅者在 SUBSCRIBE 请求的 expires 头部写入当前剩下的订阅有效时长的秒数,这样,能够立即触发通告者产生 NOTIFY 消息,将当前资源的状态通告给订阅者。需要指出的是,这种对资源的轮询会导致网络、通告者和订阅者负荷的增加。在如移动通信这样的特定应用中,订阅者一般是数据处理能力较慢、需要额外供电的移动终端设备,随着事件通告频度的增加和通告事件包的增大,将消耗很多宝贵的带宽资源,造成网络拥塞和订阅者的过载。因此,订阅者需要对通告者的状态通告频率做出限制。另外,订阅者还可以在 SUBSCRIBE 消息中指定一些事件包的过滤规则,使得通告者能够根据这一过滤规则产生通告事件,而不是任何状态发生变化时都发起通告。

一般说来,订阅者使用 SUBSCRIBE 请求建立一次订阅,称为显式订阅创建,与此相对应的还有一种隐式的订阅创建,即订阅者不是通过 SUBSCRIBE 请求来创建订阅。而是通过转交(REFER)方法,一种由 RFC 3515 定义的用于实现呼叫转交等业务的方法。REFER 请求隐



式地在被转交用户处创建订阅，所要观察的资源是转交请求的状态。

### 3.9.2 自动回叫业务示例

如前所述，会话启动协议的事件通告机制非常灵活，针对不同的应用可以定义不同的事件包，事件包被封装在 NOTIFY 消息中通告资源状态中，这一机制为实现各种功能强大的业务提供了坚实的基础。现给出使用这一机制实现的自动回叫业务流程。

#### ◆ 业务描述

用户 A 呼叫用户 B，而 B 正在会话，A 希望 B 在通话结束后能够立刻通知他，这样 A 就能够及时地和 B 建立会话了。这里，A 使用事件通告机制来获取 B 的会话状态。

#### ◆ 业务流程

如图 3-4 所示，用户 A 向用户 B 发起呼叫请求 INVITE (F1)，此时 B 正在通话，因此返回“486 Busy Here”（现在正忙）响应 (F2)。A 希望 B 能够在通话结束后通知他，于是 A 向 B 发起 SUBSCRIBE 请求 (F4)，其 Event 头部指明事件类型为 dialog，即订阅者 B 正在进行的对话状态；Accept 头部指明 A 支持使用 XML 语言封装的 dialog 事件包。

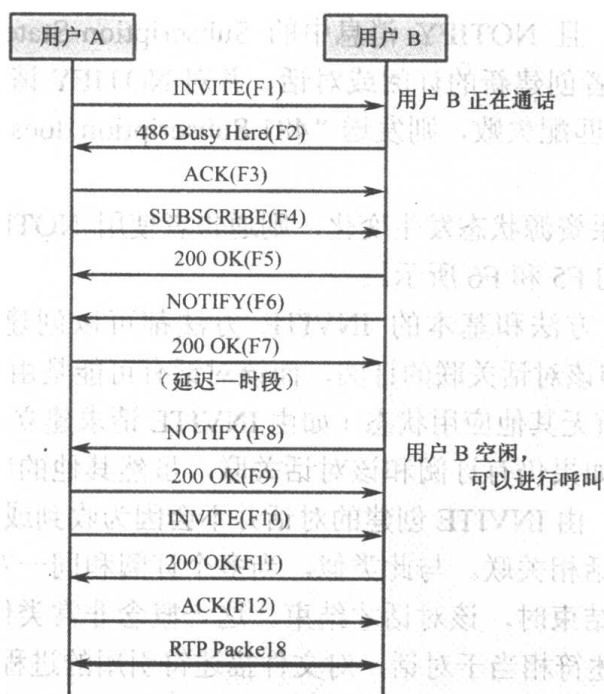


图 3-4 自动回叫呼叫流程

SUBSCRIBE 的消息片断如下：

```
SUBSCRIBE sip: userB@foo.bar SIP/2.0
```

```
...
```

```
Event: dialog
```

```
Accept: application/dialog-info+xml
```

```
...
```

B 的通告者根据本地策略对 A 进行鉴权后授权 A 订阅这一资源，返回 200 OK (F5)，并立即回送通告当前对话状态的 NOTIFY 消息。当 B 结束通话时，A 订阅的对话资源状态发生了变化，从确认状态迁移至了终结状态，B 的通告者生成 NOTIFY 消息，向 A 通告当前 B 的

对话状态 (F8)。A 得到这一信息后, 立即向 B 发起 INVITE 请求, 与 B 建立新的会话。

### 3.9.3 事件通告机制的安全性考虑

#### 1. 接纳控制

在事件通告机制中, 通告者需要将资源的状态信息通告给订阅者。然而, 在通告的事件包中, 有些信息对于用户来说是敏感的或者是隐私, 因此, 通告者应能够根据订阅者的身份, 选择性地拒绝某些用户的订阅请求, 并使用标准的会话启动协议鉴权机制对用户进行鉴权。订阅请求的最终决定权应该由用户 (自然人) 作出。

#### 2. 通告者的策略安全

在某些情况下, 对 SUBSCRIBE 请求回应 200 OK 或 4XX、6XX 响应可能会导致通告者某些敏感策略信息的泄露。因此, 在这些情况下, 通告者应总是对 SUBSCRIBE 请求回送 202 Accepted 响应。后继的 NOTIFY 中可能并不携带真正的资源状态, 但从订阅者的角度来看, 这并无任何异常。例如, 在“在席”应用中, 用户 A 向用户 B 发送 SUBSCRIBE 请求获取用户 B 是否在席的信息, B 在席, 但他拒绝了 A 的订阅请求, 因为 B 并不想让 A 知道他正在网上。如果 B 直接使用 4XX 或 6XX 响应拒绝, 则会泄露 B 的策略。为此, B 可对 SUBSCRIBE 回送 202 Accepted 响应, 在随后的 NOTIFY 消息的 Subscription-State 头部指明订阅终结, 因为没有资源, 即 A 想订阅的资源不存在, 其 Subscription-State 头部为: Subscription-State: terminated; reason=noresource, 这样既拒绝订阅请求又不会暴露通告者的策略信息。

#### 3. 防止恶意攻击

随着事件通告机制的进一步应用, 要逐步考虑应对各种 Internet 上惯用的恶意攻击。例如, 拒绝服务 (DoS) 攻击, 在当前的基于会话启动协议的事件通告模型中, 一个 SUBSCRIBE 请求将触发一个或多个通告资源状态的 NOTIFY 消息, 通告者接收到 SUBSCRIBE 请求, 要创建相应的状态, 消耗一定的系统资源。有恶意的订阅者会利用这一点发起过多的订阅请求, 这就有可能使通告者因创建过多的订阅而资源耗尽。为了减少这种攻击的机会, 通告者的实现必须包含鉴权, 以过滤恶意的订阅请求。

## 3.10 第三方呼叫控制及会话传递

### 3.10.1 第三方呼叫控制

有时, 某个用户想在另外两个用户之间建立一个会话, 而自己不参与进去, 这就需要第三方呼叫控制扩展来完成。

如图 3-5 所示, 控制器发送一个无任何会话描述符的 INVITE, A 按照通常的 SIP 处理过程向控制器含会话描述符的 200 OK 做出应答。控制器接着使用此会话描述符来邀请 B, B 将向 A 发送音频流。

### 3.10.2 会话传递

在第三方控制信令中, 媒体在其他实体之间进行交换。如果第三方自己不想再控制信令, 而想借助其他实体进行控制, 就需要使用 SIP 会话传递技术。

如图 3-6 所示, A 本来和 B 在通信, 如果 A 此时有事情外出, 而让其他人 (如 C) 和 B 通信, 就向 B 发送 REFER 原语, REFER Refer-To 标题头中含有 C 的 URL, Referred-By 标题头包含 B 的 URL。B 收到后, 将 Referred-By 标题头复制到新的 INVITE 中, 向 Refer-To 标题头中包含的 SIP URL 发送。当 C 收到 INVITE 后, 知道是 A 转接的结果, 因为 Referred-By 标题头中含有 A 的 SIP URL。当 INVITE 完成后, B 发送一个 NOTIFY 指明转接的结果。

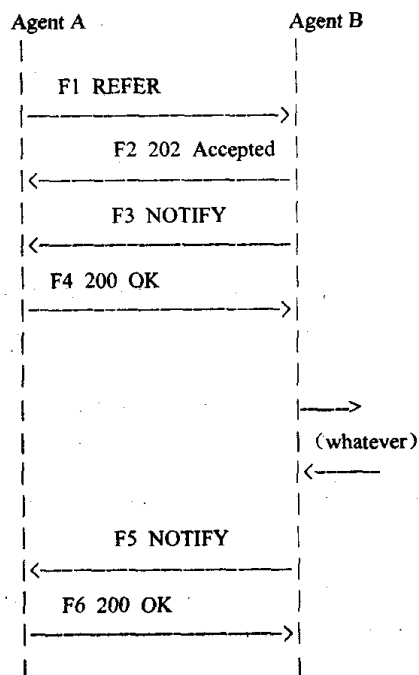


图 3-5 第三方呼叫控制流程

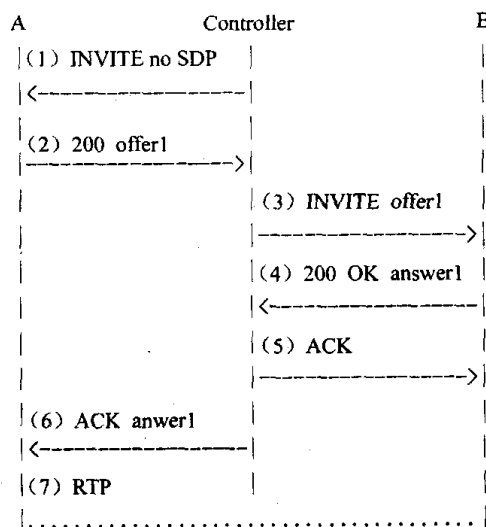


图 3-6 会话传递流程

### 3.11 SIP 计费

SIP-AAA 接口应支持发现功能、SIP 服务器更新、用户信息更新、认证、可靠性、安全等功能。一个典型的 SIP AAA 流程如图 3-7 所示。

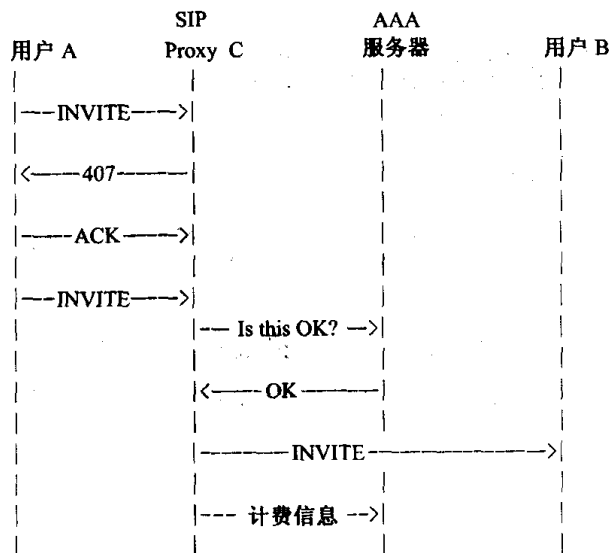


图 3-7 SIP AAA 流程

## 1. Device-Reboot-Indication AVP 扩展

DIAMETER SIP 使用 Extension Id 6。

## 2. DIAMETER-Command AVP 扩展

DIAMETER 命令 AVP 扩展如表 3-4 所示。

其中：

<Admission Control Request> ::= <DIAMETER Header><Command AVP><Host-IP-Address>  
<SIP-Sequence><SIP-Call-ID><SIP-To><SIP-From> [<SIP-Entire-Msg>]<Timestamp AVP>  
<Initialization-Vector AVP>{<Integrity-Check-Vector AVP> ||<Digital-Signature AVP> }

<Admission Control Response> ::= <DIAMETER Header><Command AVP><Host-IP-Address>  
<Result-Code> [<Error-Code>]<SIP-Sequence><SIP-Call-ID> [<SIP-To>] [<SIP-From>]<Timestamp  
AVP><Initialization-Vector AVP>{<Integrity-Check-Vector AVP> || <Digital-Signature AVP> }

<Accounting Request> ::= <DIAMETER Header> <Command AVP><Host-IP-Address>  
<Timestamp><SIP-Sequence> <SIP-Call-ID><SIP-To> <SIP-From> [<SIP-Entire-Msg>] <Timestamp  
AVP><Initialization-Vector AVP> {<Integrity-Check-Vector AVP> ||<Digital-Signature AVP> }

<Accounting Response> ::= <DIAMETER Header> <Command AVP> <Host-IP-Address>  
<SIP-Sequence> <Result-Code> [<Error-Code>]<SIP-Call-ID> [<SIP-To>] [<SIP-From>]<Timestamp  
AVP><Initialization-Vector AVP>{<Integrity-Check-Vector AVP> ||<Digital-Signature AVP> }

<Termination Request> ::= <DIAMETER Header> <Command AVP><Host-IP-Address>  
<SIP-Sequence><SIP-Call-ID><SIP-To><SIP-From> <Timestamp AVP><Initialization-Vector  
AVP>{<Integrity-Check-Vector AVP> ||<Digital-Signature AVP> }

<Termination Response> ::= <DIAMETER Header> <Command AVP><Host-IP-Address>  
<Result-Code> [<Error-Code>]<SIP-Sequence><SIP-Call-ID> [<SIP-To>] [<SIP-From>]<Timestamp  
AVP><Initialization-Vector AVP>{<Integrity-Check-Vector AVP> ||<Digital-Signature AVP> }

表 3-4 DIAMETER 命令 AVP 扩展

命 令 名	命 令 代 码
SIP-Admission-Request	600
SIP-Admission-Response	601
SIP-Accounting-Request	602
SIP-Accounting-Response	603
SIP-Termination-Request	604
SIP-Termination-Response	605

## 3. DIAMETER 错误代码 AVP 扩展

DIAMETER 错误代码含义如表 3-5 所示。

表 3-5 DIAMETER 错误代码含义

0x6001	在请求中缺少 Call-ID
0x6002	在请求中缺少 To
0x6003	在请求中缺少 From
0x6010	禁止的呼叫者
0x6011	禁止的被呼者

#### 4. SIP 特定的 AVP

SIP 特定的 AVP 属性名及其代码对应关系如表 3-6 所示。

表 3-6 SIP 特定 AVP

Attribute Name	AVP Code
SIP-Sequence	600
SIP-Call-ID	601
SIP-To	602
SIP-From	603
SIP-Entire-Msg	604

### 3.12 SIP 安全

SIP 安全有很多措施，如 SIP Digest Authentication, S/MIME, SDP & RTP security（增加了一个属性  $k=encryption\ key$ ）等。

#### 3.12.1 AKA 机制

AKA（Authentication and Key Agreement）机制，其流程图见图 3-8 所示。

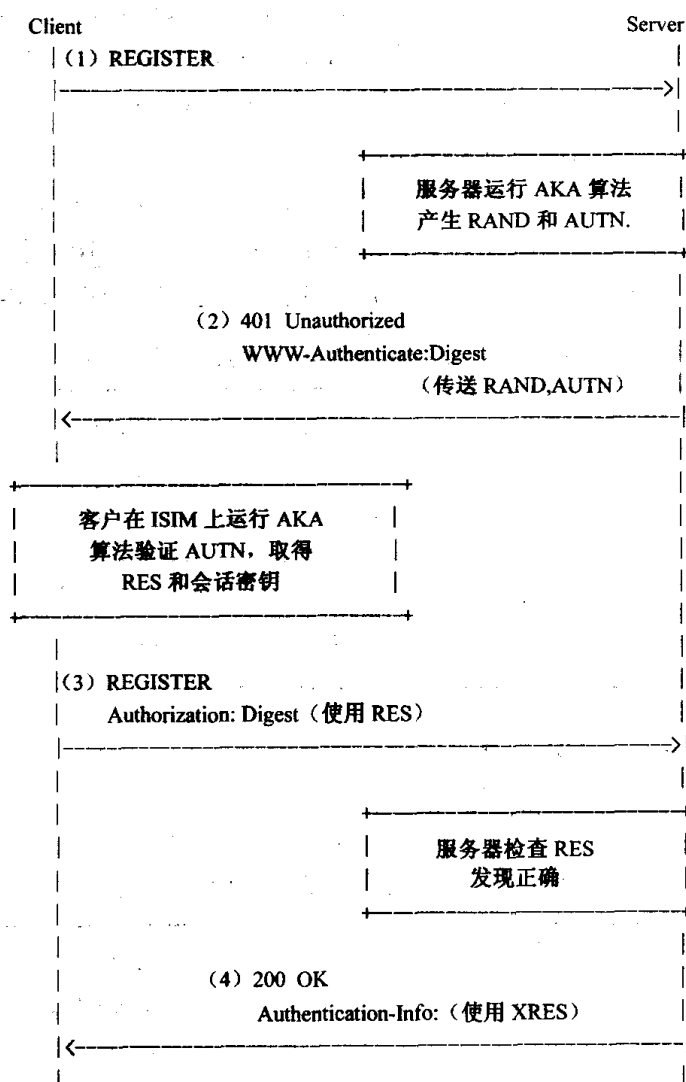


图 3-8 成功认证的消息流程

### 1. Initial request

消息如下:

```
REGISTER sip:home.mobile.biz SIP/2.0
```

### 2. Response containing a challenge

消息如下:

```
SIP/2.0 401 Unauthorized
WWW-Authenticate: Digest
    realm="RoamingUsers@mobile.biz",
nonce="CjPk9mRqNuT25eRkajM09uTl9nM09uTl9nMz5OX25PZz==",
    qop="auth,auth-int",
    opaque="5ccc069c403ebaf9f0171e9517f40e41",
    algorithm=AKAv1-MD5
```

### 3. Request containing credentials

消息如下:

```
REGISTER sip:home.mobile.biz SIP/2.0
Authorization: Digest
    username="jon.dough@mobile.biz",
    realm="RoamingUsers@mobile.biz",
nonce="CjPk9mRqNuT25eRkajM09uTl9nM09uTl9nMz5OX25PZz==",
    uri="sip:home.mobile.biz",
    qop=auth-int,
    nc=00000001,
    cnonce="0a4f113b",
    response="6629fae49393a05397450978507c4ef1",
    opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

### 4. Successful response

消息如下:

```
SIP/2.0 200 OK
Authentication-Info:
    qop=auth-int,
    rspauth="6629fae49393a05397450978507c4ef1",
    cnonce="0a4f113b",
    nc=00000001
```

## 3.12.2 SIP 私密性

### 1. 扩展头及使用范围

扩展头及其使用范围见表 3-7、表 3-8 和表 3-9 所示。

表 3-7 Privacy 头使用范围

Header field	Proxy	ACK	BYE	CAN	INV	OPT	REG	SUB	NOT	PRK	INFO	UPD	MSG
Privacy	amrd	o	o	o	O	o	o	o	o	o	o	o	o

表 3-8 P-Asserted-Identity 头使用范围

Header field	Proxy	ACK	BYE	CAN	INV	OPT	REG	SUB	NOT	REF	INFO	UPD	PRA
P-Asserted-I identity	adr	-	o	-	o	o	-	o	o	o	-	-	-

表 3-9 P-Preferred-Identity 头使用范围

Header field	Proxy	ACK	BYE	CAN	INV	OPT	REG	SUB	NOT	REF	INFO	UPD	PRA
P-Preferred-I identity	adr	-	o	-	o	o	-	o	o	o	-	-	-

## 2. 实例

本例中, proxy.company.com 利用从 SIP 摘要认证消息中发现的身份创建 P-Asserted-Identity 头, 然后转发给可信代理, 再转发给可信的网关。

### 1) useragent.company.com -> proxy.company.com

消息如下:

INVITE sip:+14085551212@company.com SIP/2.0; 发起请求

Via: SIP/2.0/TCP useragent.company.com;branch=z9hG4bK-123

To: <sip:+14085551212@company.com>; 收端

From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748; 发端

Call-ID: 245780247857024504; 呼叫 ID

CSeq: 1 INVITE

Max-Forwards: 70

Privacy: id

### 2) proxy.company.com -> useragent.company.com

消息如下:

SIP/2.0 407 Proxy Authorization

Via: SIP/2.0/TCP useragent.company.com;branch=z9hG4bK-123

To: <sip:+14085551212@company.com>;tag=123456; 收端

From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748; 发端

Call-ID: 245780247857024504; 呼叫 ID

CSeq: 1 INVITE

Proxy-Authenticate: .... realm="sip.company.com"

### 3) useragent.company.com -> proxy.company.com

消息如下:

INVITE sip:+14085551212@company.com SIP/2.0

Via: SIP/2.0/TCP useragent.company.com;branch=z9hG4bK-124

To: <sip:+14085551212@company.com>; 收端  
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748; 发端  
Call-ID: 245780247857024504; 呼叫 ID  
CSeq: 2 INVITE  
Max-Forwards: 70; 最大转发次数  
Privacy: id  
Proxy-Authorization: .... realm="sip.company.com" user="fluffy"

#### 4) proxy.company.com -> proxy.pstn.net (trusted)

消息如下:

INVITE sip:+14085551212@proxy.pstn.net SIP/2.0  
Via: SIP/2.0/TCP useragent.company.com;branch=z9hG4bK-124  
Via: SIP/2.0/TCP proxy.company.com;branch=z9hG4bK-abc  
To: <sip:+14085551212@company.com>; 收端  
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748; 发端  
Call-ID: 245780247857024504; 呼叫 ID  
CSeq: 2 INVITE  
Max-Forwards: 69; 最大转发次数  
P-Asserted-Identity: "Cullen Jennings" <sip:fluffy@company.com>  
P-Asserted-Identity: tel:+14085264000  
Privacy: id

#### 5) proxy.pstn.net -> gw.pstn.net (trusted)

消息如下:

INVITE sip:+14085551212@gw.pstn.net SIP/2.0  
Via: SIP/2.0/TCP useragent.company.com;branch=z9hG4bK-124  
Via: SIP/2.0/TCP proxy.company.com;branch=z9hG4bK-abc  
Via: SIP/2.0/TCP proxy.pstn.net;branch=z9hG4bK-a1b2  
To: <sip:+14085551212@company.com>; 收端  
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748; 发端  
Call-ID: 245780247857024504; 呼叫 ID  
CSeq: 2 INVITE  
Max-Forwards: 68; 最大转发次数  
P-Asserted-Identity: "Cullen Jennings" <sip:fluffy@company.com>  
P-Asserted-Identity: tel:+14085264000  
Privacy: id

## 3.13 SIP QoS

### 3.13.1 在 SIP 中实现 RSVP 功能

在 SIP 中实现 RSVP 的步骤如图 3-9 所示。



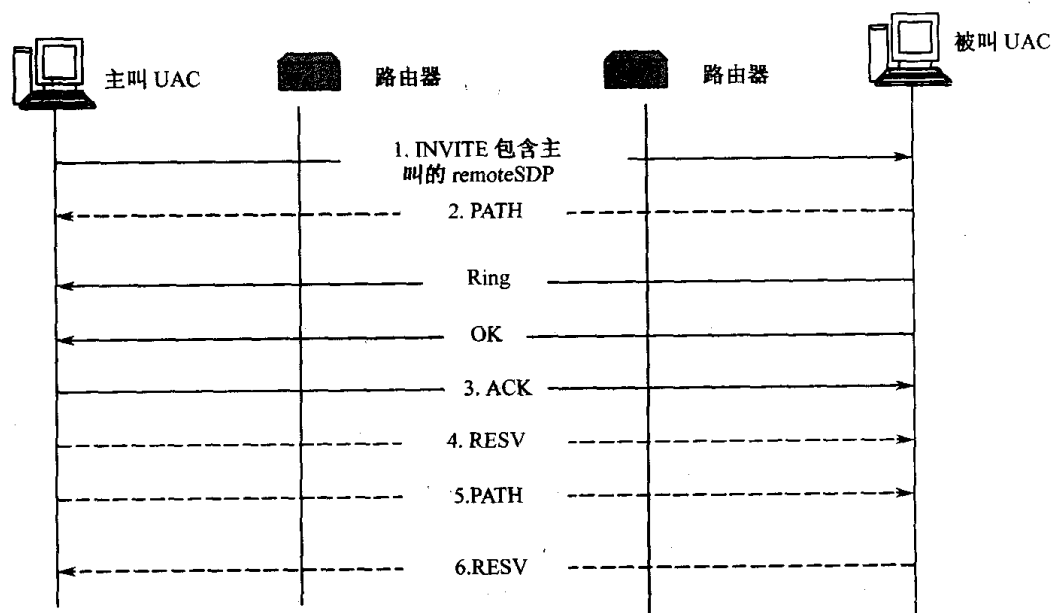


图 3-9 SIP RSVP 流程

图 3-9 中实线的部分是 SIP 命令，虚线部分是 RSVP 消息。首先创建一个 RSVP 的会话实例，如果成功则返回一个非零的数值用于表示建立的会话 ID 号，创建会话是为了发送 RSVP PATH 或者是发送 RSVP RESV。接下来的流程如下：

(1) 主叫部分发送 INVITE 命令，命令中包含有主叫的会话描述（这里称为 Remote SDP）。

(2) 被叫部分此时处于 OpRing 的状态中接收到主叫的 INVITE 消息以后，根据主叫的 INVITE 消息和主叫的 SDP，得到主叫的地址和主叫的 RSVP 端口（主叫的 RTP 端口）；被叫发送包含有数据流标识和数据业务流特征的 PATH 消息到主叫。

(3) 在完成了一系列 SIP 命令和状态的交换（RING，OK 过程）以后，主叫方开始准备发送 ACK 消息了，主叫在下一个过程（收到 ACK 消息后，打开 RTP 通道之前）的时候，已经保证了所有的主叫到被叫之间的路由器都已经收到了 PATH 预留消息。

(4) 5 和 6 两个消息是主叫端点向被叫端点发送 PATH 消息，并且接收对端的 RESV 消息的过程。

### 3.13.2 SIP COMET

只有当所有条件满足时，才会发送 COMET。

#### 1. SDP 扩展

扩展 SDP 的 BNF 描述如下：

```

qos-attribute="a=qos:" strength-tag SP direction-tag[SP confirmation-tag]; 服务质量属性
strength-tag= ("mandatory" | "optional" | "success" | "failure")
direction-tag= ("send" | "recv" | "sendrecv")
confirmation-tag = "confirm"
security-attribute="a=secure:"SP strength-tag SP direction-tag[SP confirmation-tag]
  
```

#### 2. SDP 实例

以下为从 A 到 B 的 SIP INVITE 中的 SDP 描述。

```

v=0
o=mhandley 2890844526 2890842807 IN IP4
126.16.64.4
s=SDP Seminar
i=A Seminar on the session description
protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sd
p.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
m=audio 49170 RTP/AVP 0; 媒体类型
a=qos:mandatory recv confirm
m=video 51372 RTP/AVP 31; 媒体类型
a=secure:mandatory sendrecv
m=application 32416 udp wb
a=orient:portrait
a=qos:optional sendrecv
a=secure:optional sendrecv

```

### 3. SIP 流程

SIP 的工作流程如图 3-10 所示。

#### 3.13.3 策略服务

##### 1. 模型

策略服务模型如图 3-11 所示，其中，issll 为 Intergrated Services over Specific Lower Layers，APS 为应用策略服务器。

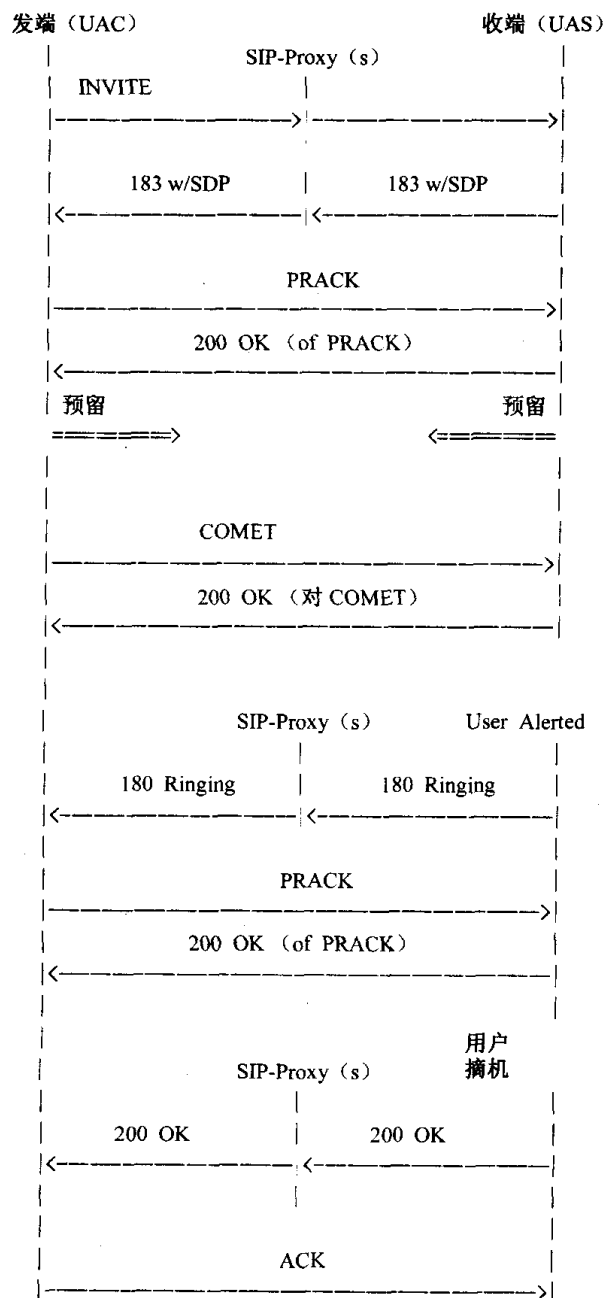


图 3-10 基本呼叫流程

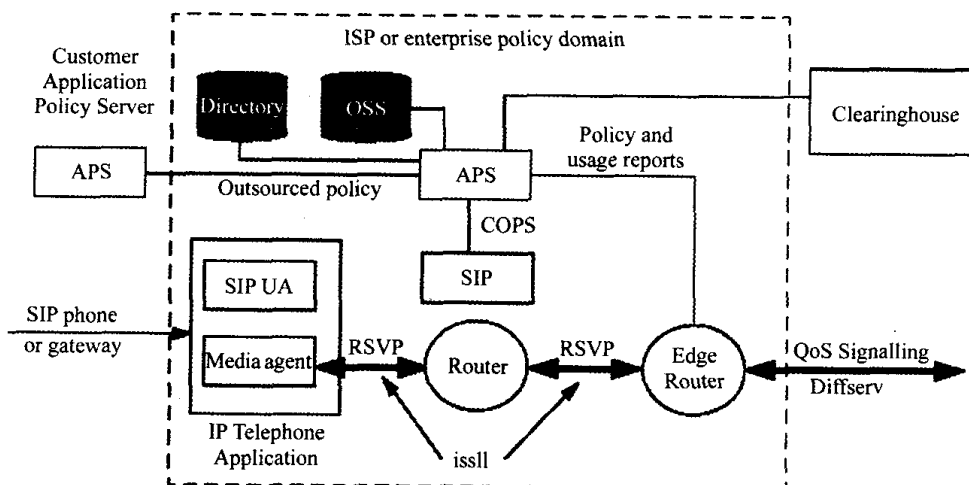


图 3-11 策略服务模型



### 1) SIP 消息

消息如下:

```
INVITE sip:+1-972-555-5555@sip.domain2.com;user=phone SIP/2.0
Via: SIP/2.0/UDP phone1.domain1.com:5060
From: Henry Sinnreich henry.sinnreich@phone1.domain1.com; 发端
To: <sip:+1972-555-5555@sip.domain2.com;user=phone>; 收端
Callid: 123456@domain1.com
Cseq: 1 INVITE
Contact: phone1.domain1.com
```

...

SDP

QoS required

### 2) COPS 消息

消息如下:

REQ OSP =

(Common Header, Client Handle, Context, ClientSI: OSP)

Client Handle = "123456@domain1.com"

Context = "Incoming & Outgoing", "OSP"

ClientSI: OSP = (Called Number, Calling Number)

Called Number = "To: <sip:+1972-555-5555@sip.domain2.com;user=phone>; 收端

Calling Number = "From: Henry Sinnreich; 发端

henry.sinnreich@phone1.domain1.com"

### 3) OSP 消息

消息如下:

<AuthReq>

<?xml version=1.0?>

<Message messageId="123454321" random="12345678">

<AuthorizationRequest componentId="9876567890">

<Timestamp>

1999-10-24T17:03:00Z

</Timestamp>

<CallId encoding="base64">

YT64VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756t

</CallId>

<SourceInfo type="henry.sinnreich@phone1.domain1.com">

</SourceInfo>

<DestinationInfo type="e164">

1-972-555-5555

</DestinationInfo>

```

<Service/>
<MaximumDestinations>
    5
</MaximumDestinations>
</AuthorizationRequest>
</Message> <AuthReq/>

```

#### 4) OSP 消息

消息如下:

```

<AuthRsp>
<?xml version=1.0?>
  <Message messageId="123454321" random="12345678">
    <AuthorizationResponse componentId="9876567890">
      <Timestamp>
        1999-10-24T17:03:01Z
      </Timestamp>
      <Status>
        <Code>
          200
        </Code>
        <Description>
          success
        </Description>
      </Status>
      <TransactionId>
        67890987
      </TransactionId>
      <Destination>
        <DestinationSignalAddress>
          [172.16.1.2]:112
        </DestinationSignalAddress>
        <Token encoding="base64">
          YT64VqpFyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756t
          HGTrfvbnjn8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfH
          6ghyHhHUujpFyF47GhIGfHfYT64VQbnj
        </Token>
        <ValidAfter>
          1999-04-24T17:01:01Z
        </ValidAfter>
        <ValidUntil>
          1998-04-24T17:11:01Z
        </ValidUntil>
      </Destination>
    </AuthorizationResponse>
  </Message>
</AuthRsp>

```

```

</ValidUntil>
<CallId encoding="base64">
YT64VqpFyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756t
</CallId>
<UsageDetail>
  <Service/>
  <Amount>
    24
  </Amount>
  <Increment>
    3600
  </Increment>
  <Unit>
    s
  </Unit>
</UsageDetail>
</Destination>
</AuthorizationResponse>
</Message>
<AuthRsp>

```

### 5) COPS 消息

消息如下:

```

DEC = (Common Header, Client Handle, Context,
Decision Flag, Decision: ClientSI Data: OSP)
Client Handle = "123456@domain1.com"
Context = "Incoming & Outgoing"
Decision Flag = "Install"
Decision: ClientSI Data: OSP = (Called Number, Authorization Token)
Called Number = "To: <sip:+1972-555-5555@sip.domain2.com;user=phone>"
Authorization Token = "<Token encoding="base64">
YT64VQpFyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756t
HGTrfvbnjn8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfH
6ghyHhHUujpFyF47GhIGfHfYT64VQbnj
</Token>"

```

Opt. SIP

SIP/2.0 100 Trying

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>

Callid: 123456@domain1.com

Cseq: 1 INVITE

## 6) SIP 消息

消息如下:

INVITE sip:+1-972-555-5555@sip.domain2.com;user=phone SIP/2.0

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>

Callid: 123456@domain1.com

Cseq: 1 INVITE

Contact: phone1.domain1.com

Record-Route: sip.domain1.com

...

SDP

QOS required

## 7) SIP 消息

消息如下:

INVITE sip:+1-972-555-5555@sip.domain2.com;user=phone SIP/2.0

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>

Callid: 123456@domain1.com

Cseq: 1 INVITE

Contact: phone1.domain1.com

Record-Route: sip.domain1.com

...

SDP

QOS required

## 8) SIP 消息

消息如下:

SIP/2.0 183 Session Progress

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

...

SDP

QoS required

### 9) COPS 消息

消息如下:

REQ AssuredLDP=

(Common Header, Client Handle, Context, ClientSI: ConfigLocalPolicy)

Client Handle = "123456@domain1.com"

Context = "Configuration Request", "ConfigLocalDecisionPolicy"

ClientSI: ConfigLocalDecisionPolicy = ( Caller Media Address, Caller Media

Port, Caller SDP info, Callee Media Address, Callee Media Port , Callee SDP

info )

### 10) COPS 消息

消息如下:

DEC =

(Common Header, Client Handle) (Decision) | (Error)

Client Handle = "R1001"

Decision = (Context, Decision Flag, Named Decision Data: Config Local DecisionPolicy)

Context = "Configuration Request", "ConfigLocalDecisionPolicy"

Decision Flag = "Install"

Named Decision Data: (Binding Count, PRID, BPD)

Binding Count = 1

PRID = "1.2.3.4"

BPD = (Ace of Caller flow, Caller Token rate, Caller Peak rate, Caller Token Bucket Size, Caller Min Policed Unit, Caller Max Packet Size, Caller Reservation Style, Caller DSCLASS, Caller POLICY)

| (Ace of Callee flow, Callee Token rate, Callee Peak rate, Callee Token Bucket Size, Callee Min Policed Unit, Callee Max Packet Size, Callee Reservation Style, Callee DSCLASS, Callee POLICY ))

### 11) COPS 消息

消息如下:

RPT =

(Common Header, Client Handle, Report-Type, ClientSI: ConfigRPT)

Client Handle = "PEP1001"

Report-Type = "Installed"

ClientSI = (PRID)



PRID = "1.2.3.4"

## 12) COPS 消息

消息如下:

DEC =

(Common Header, Client Handle, Context, Decision Flags)

Client Handle = "123456@domain1.com"

Context = "Incoming&Outgoing", "ConfigLocalDecisionPolicy"

Decision Flag = "Install"

## 13) SIP 消息

消息如下:

SIP/2.0 183 Session Progress

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

...

SDP

QoS required

## 14) COPS 消息

消息如下:

REQ

AssuredLDP= (Common Header, Client Handle, Context, ClientSI:  
ConfigLocalPolicy)

Client Handle = "123456@domain1.com"

Context = "Configuration Request", "ConfigLocalDecisionPolicy"

ClientSI: ConfigLocalDecisionPolicy = ( Caller Media Address, Caller  
Media Port, Caller SDP info, Callee Media Address, Callee Media Port ,  
Callee SDP info )

## 15) COPS 消息

消息如下:

DEC =

(Common Header, Client Handle) (Decision) | (Error)

Client Handle = "R1001"

Decision = (Context, Decision Flag, Named Decision Data: Config Local  
DecisionPolicy)

Context = "Configuration Request", "ConfigLocalDecisionPolicy"

Decision Flag = "Install"

Named Decision Data: (Binding Count, PRID, BPD)

Binding Count = 1

PRID = "1.2.3.4"

BPD = (Ace of Caller flow, Caller Token rate, Caller Peak rate, Caller  
Token Bucket Size, Caller Min Policed Unit, Caller Max Packet Size,  
Caller Reservation Style, Caller DSCLASS, Caller POLICY)

| (Ace of Callee flow, Callee Token rate, Callee Peak rate, Callee  
Token Bucket Size, Callee Min Policed Unit, Callee Max Packet Size,  
Callee Reservation Style, Callee DSCLASS, Callee POLICY ) )

#### 16) COPS 消息

消息如下:

RPT =

(Common Header, Client Handle, Report-Type, ClientSI: ConfigRPT)

Client Handle = "PEP1001"

Report-Type = "Installed"

ClientSI = (PRID)

PRID = "1.2.3.4"

#### 17) COPS 消息

消息如下:

DEC =

(Common Header, Client Handle, Context, Decision Flags)

Client Handle = "123456@domain1.com"

Context = "Incoming&Outgoing", "ConfigLocalDecisionPolicy"

Decision Flag = "Install"

#### 18) SIP 消息

消息如下:

SIP/2.0 183 Session Progress

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

...

SDP

QoS required

#### 19) RSVP 消息

消息如下:

PATH =

(Common Header, [INTEGRITY], SESSION, RSVP\_HOP, TIME\_VALUES, [ (POLICY\_DATA) +], [sender descriptor]

Sender descriptor = (SENDER\_TEMPLATE, SENDER\_TSPEC, [ADSPEC] )

Note: Messages 19 - 31 are for Caller to Callee flow.

**20) RSVP 消息**

见消息 19)。

**21) RSVP 消息**

见消息 19)。

**22) RSVP 消息**

RESV = ( Common Header, [INTEGRITY], SESSION, RSVP\_HOP, TIME\_VALUES, [RESV\_CONFIRM], [SCOPE], [(POLICY\_DATA) + ] STYLE, flow descriptor list)

**23) RSVP 消息**

见消息 22)。

**24) RSVP 消息**

见消息 22)。

**25) RSVP 消息**

消息如下:

RESV-CONF = (Common Header, [INTEGRITY], SESSION, ERROR\_SPEC, RESV\_CONFIRM, STYLE, flow descriptor list )

**26) RSVP 消息**

见消息 25)。

**27) RSVP 消息**

见消息 25)。

**28) RSVP 消息**

见消息 19)。

**29) RSVP 消息**

见消息 19)。

**30) RSVP 消息**

见消息 19)。

**31) RSVP 消息**

见消息 22)。

**32) RSVP 消息**

见消息 22)。

**33) RSVP 消息**

见消息 22)。

**34) RSVP 消息**

见消息 25)。

**35) RSVP 消息**

见消息 25)。

**36) RSVP 消息**

见消息 25)。

**37) SIP 消息**

消息如下：

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

...

SDP(可选)

**38) SIP 消息**

消息如下：

SIP/2.0 183 Ringing

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

...

SDP(可选)

**39) SIP 消息**

消息如下：

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

...

SDP(可选)

#### 40) SIP 消息

消息如下:

SIP/2.0 200 OK

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain1.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

Contact: 972gw.domain2.com

Record-Route: sip.domain1.com, sip.domain2.com

...

SDP (可选)

#### 41) SIP 消息

消息如下:

SIP/2.0 200 OK

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

Contact: 972gw.domain2.com

Record-Route: sip.domain1.com, sip.domain2.com

...

SDP (可选)

#### 42) SIP 消息

消息如下:

SIP/2.0 200 OK

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

Contact: 972gw.domain2.com

Record-Route: sip.domain1.com, sip.domain2.com

...

SDP (可选)

#### 43) SIP 消息

消息如下:

ACK sip.domain1.com SIP/2.0

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

Route: sip.domain2.com, gw972.domain2.com

...

SDP (可选)

#### 44) SIP 消息

消息如下:

ACK sip.domain2.com SIP/2.0

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

Route: gw972.domain2.com

...

SDP (可选)

#### 45) SIP 消息

消息如下:

ACK gw972.domain2.com SIP/2.0

Via: SIP/2.0/UDP sip.domain2.com:5060

Via: SIP/2.0/UDP sip.domain1.com:5060

Via: SIP/2.0/UDP phone1.domain1.com:5060

From: Henry Sinnreich <henry.sinnreich@phone1.domain1.com>

To: <sip:+1972-555-5555@sip.domain2.com;user=phone>;tag=134159

Callid: 123456@domain1.com

Cseq: 1 INVITE

...

SDP (可选)

### 3. OSP 相关头部使用范围

OSA 相关头部使用范围如表 3-10 所示。

表 3-10 OSP 相关头部使用范围

头 部 域	Where	Enc	e-e	ACK	BYE	CAN	INV	OPT	REG
OSP-Authorization-Token	R	N	H	-	-	-	o	-	-

3.14 SIP-H.323 互通

1. IWF 要求

IWF 应该具有以下功能:

- 呼叫序列映射;
- 地址解析;
- 终端能力事务;
- 打开和关闭媒体通道;
- 映射媒体算法;
- 呼叫资源保留与释放;
- 能够提供呼叫状态;
- 可完成呼叫状态机;
- 中间呼叫信令处理;
- 业务互操作逻辑。

2. 流程

1) H.323 到 SIP 终端

H.323 到 SIP 终端的流程如图 3-13 所示。

2) SIP 终端到 H.323 终端

SIP 终端到 H.323 终端流程如图 3-14 所示。

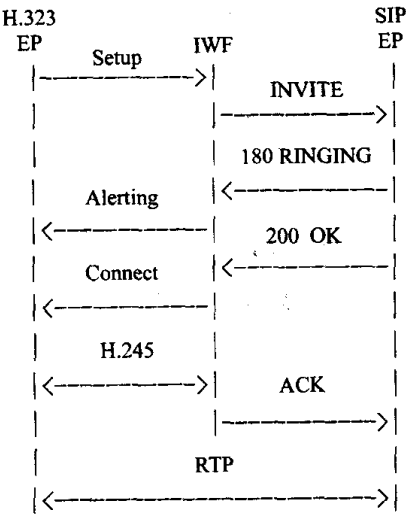


图 3-13 H.323 到 SIP 终端呼叫流程

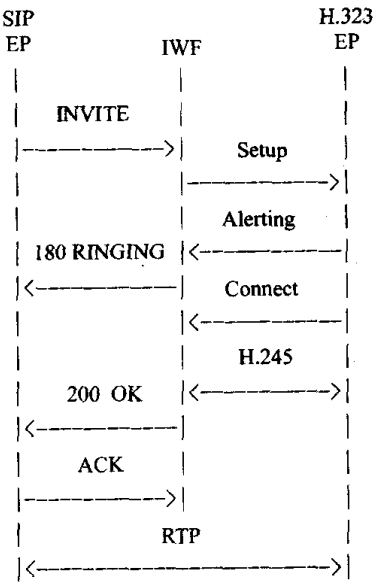


图 3-14 SIP 终端到 H.323 呼叫流程

## 3.15 SIP MIB 变量

SIP-COMMON-MIB, SIP-SERVER-MIB, SIP-UA-MIB 和 SIP-TC.SIP-COMMON-MIB 包含在所有 SIP 实体中应用的公共对象；SIP-SERVER-MIB 包含 Registrars, Proxies 和重定向服务器特定的对象；SIP-UA-MIB 包含用户代理特定的对象。

### 1. SIP-COMMON-MIB

有以下表项：

- sipCommonCfgBase: 包含 SIP 版本、操作与管理状态、SIP 组织名称、可管理的最大 SIP 会话数等。
- sipCommonCfgTimer: SIP 用户代理和有状态代理实体的时钟。
- sipCommonCfgRetry: SIP 用户代理和有状态代理实体的重试计数器。
- sipCommonCfgExpires: SIP 用户代理和有状态代理实体的头失效值。
- sipCommonStatsSummary: SIP 请求/响应数、事务数统计值。
- sipCommonStatsMethod: Invite, Ack, Bye, Cancel, Options 以及 Register 方法请求/响应数。
- sipCommonStatusCode: 每种响应 (1xx, 2xx 等) 统计。
- sipCommonStatsTrans: 等待事务的统计。
- sipCommonStatsRetry: 每种方法重试次数。
- sipCommonStatsOther: 被不支持 URI 接收的 SIP 请求统计。
- sipCommonNotifObjects: 通告对象。
- sipStatusCodeNotif: 指示发送或接收的特定状态码。
- sipStatusCodeThreshExceededNotif: 指示特定的状态码已经超量发送或接收。

### 2. SIP-SERVER-MIB

有以下表项：

- sipServerCfg: 主机配置信息 (如地址)。
- sipServerStats: 主机状态, 目前为空。
- sipProxyCfg: 代理服务器配置。
- sipProxyStats: 代理服务器状态。
- sipRedirCfg: 重定向服务器配置。
- sipRedirStats: 重定向服务器状态。
- sipRegCfg: 注册服务器配置。
- sipRegStats: 注册服务器状态。

### 3. SIP-UA-MIB

有以下表项：

- sipUACfgSipServer: 用户代理相关配置。



## 3.16 SIP 其他扩展

SIP 协议的基本设计思想是将协议的基本功能与扩展功能相分离,前者构成稳定而相对简单的 SIP 基础标准,后者源于增值业务的需要,对应为一系列草案文件,经过一定时间的运行试验,成熟的扩展功能将纳入到基础标准中。常见的 SIP 扩展有: SIP 负载控制、MIME 扩展、SIP cookies 等。

在描述被呼能力时,虽然 Allow, Accept, Accept-Language and Supported 可进行一定的描述,但不全面。draft-ietf-sip-callee-caps-02 里定义了完整的描述能力的框架。

UPDATE 允许客户更新会话参数,如媒体流编码格式,但是对对话状态没有影响。和 re-INVITE 不同,它可以在 INVITE 完成之前发送。具体描述如下:

draft-ietf-sip-authid-body 定义了 SIP Authenticated Identity Body (AIB)格式。

draft-ietf-sip-congestsafe 定义了如何有效地减少拥塞的机制。

draft-ietf-sip-connect-reus 定义了重用连接的机制。

draft-ietf-sip-content-indirect-mech 描述了传送内容链接的过程。

draft-ietf-sip-identity-01 改进了用户身份的认证与管理。

draft-ietf-sip-join 新增了 JOIN,用来加入会议。

draft-ietf-sip-publish 描述了发布事件状态的过程。

draft-ietf-sip-replaces 定义了替换对话的机制。

draft-ietf-sip-resource-priority 定义了通信优先级。

draft-ietf-sip-scvrtdisco 定义了注册期间进行路由的机制。

draft-ietf-sip-session-timer 允许通过 re-INVITE 或 UPDATE 周期性地刷新 SIP 会话。

draft-ietf-sipping-early-media 定义了两种模型,用来判断是从 UAS 接收还是自己产生回铃音。

## 3.17 小结

本章介绍了 SIP 扩展的原则,并介绍了典型的一些 SIP 扩展,如 SIP 事件扩展、SIP 服务质量扩展、SIP 安全性扩展、SIP 即时消息扩展和 SIP 会议扩展,随后还介绍了 SDP 的扩展及其 SIP 网络管理。

## 3.18 习题

1. 分析 SIP 扩展的原则及其带来的好处。
2. 简述常见的 SIP 扩展。
3. 简述 SDP 扩展。
4. 简述 SIP 有哪些网管变量。

## 第4章 SIP在软交换网络中的应用

知识点:

- 软交换基础
- 基于 SIP 的智能网业务
- 基于 SIP 的软交换互通
- 基于 SIP 的软交换业务

软交换（Softswitch）技术是实现新一代语音、通信及多媒体和数据交互的核心技术。软交换技术体现了控制/业务与承载分离的思想，作为控制实体的软交换机，完成呼叫接续、业务控制和用户管理等功能，而业务媒体流的传送由 IP 基础网络完成。在整个软交换体系中，完成接续、控制功能的信令协议起到核心作用，决定了业务提供能力的强弱。IETF 制订的新一代会话控制协议（Session Initiation Protocol, SIP），具有简单、开放、灵活、可扩展等多方面明显优点，已成为下一代网络软交换体系的重要技术。本章介绍 SIP 在软交换中的应用。

### 4.1 软交换基础

#### 4.1.1 软交换简介

软交换（Softswitch）技术是一种近年来在电信界受到广泛关注的技术。自贝尔实验室 1997 年提出软交换的概念以来，在短短的几年时间内，国际上已经先后经历了实验室阶段、市场推广阶段、应用阶段这三个发展阶段。在众多制造商和运营商的共同推动下，软交换产品逐步趋于成熟，功能日益丰富，标准化过程正稳步推进，从而使得软交换技术开始走向市场。在电路交换网中，呼叫控制、业务提供以及交换矩阵均集中在一个交换系统中，而软交换技术的主要思想是业务层面、控制层面和承载层面/接入层面相分离，各实体之间通过标准的协议进行连接和通信，以便在网上更加灵活地提供业务。其中的软交换设备实际上是一个基于软件的分布式交换/控制平台，是实现传统程控交换机“呼叫控制”功能的实体。它将呼叫控制功能从传统的网关中分离出来，公开业务、控制、接入和交换间的协议，从而真正实现多厂家的网络运营环境，并可以方便地引入多种业务。

与电路交换技术相比，利用软交换技术构建的网络成本低，且由于软交换技术采用开放式平台，各接口之间采用标准协议，所以更有利于提供新的业务和应用。

#### 4.1.2 软交换的系统结构、协议及业务

##### 1. 软交换的系统结构

软交换的主要功能包括：呼叫控制功能、地址解析/路由功能、业务提供功能、网管及计费功能。软交换系统的逻辑结构如图 4-1 所示，包括软交换设备、应用服务器、计费和账务系统、操作维护管理系统、认证系统、各种网关设备以及各种终端设备。

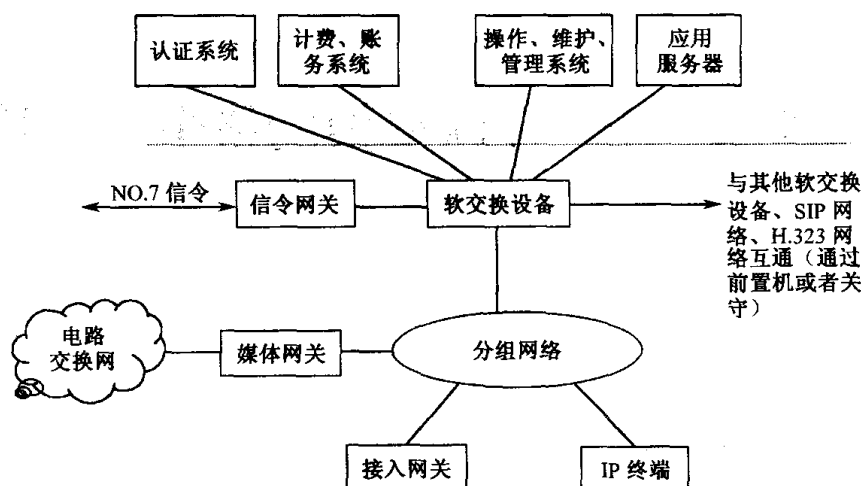


图 4-1 软交换的系统结构

软交换设备主要是完成一些基本的呼叫控制功能，同时控制媒体网关和终端设备进行承载链路的建立。在软交换设备中也可以实现一些基本的业务，但原则上所有的业务应该尽量推向边缘，利用应用服务器或其他类型的服务器来实现。

信令网关对 No.7 信令的底层承载协议进行转换，以便在分组网内能够继续传送 No.7 信令，而高层协议一般保持不动，直接送到软交换设备中进行处理。

媒体网关是为了实现电路交换网与软交换系统的互通。随着通信网络向软交换体系的演进，现有的 PSTN/ISDN/CDMA/GSM 等传统的电信网络将逐渐被推向网络的边缘，然后通过媒体网关接入到软交换系统中，利用软交换来进行呼叫中继或将呼叫终结在分组网内。媒体网关主要完成媒体流的转换。

应用服务器和其他类型的服务器（如资源服务器、多媒体服务器）上具体实现各种业务，同时在应用服务器上应该提供可编程的接口，以便提供给第三方开发新的业务。

终端设备可分为 SIP 终端、H.248 终端、MGCP 终端或者 H.323 终端，其中包括软终端。终端设备可以直接接入软交换设备，也可以通过接入设备如接入网关和综合接入系统（IAD）接入软交换设备。

软交换系统中还需要认证系统对用户进行认证以防止非法用户盗用系统资源，计费和账务系统对呼叫进行计费，操作维护管理系统对软交换系统的设备进行配置和管理。

## 2. 软交换通信协议

软交换设备是一个开放的、多协议的实体，因此必须采用标准协议与各种网关、各种网络进行通信。

软交换设备和媒体网关（也称为中继网关）之间的通信协议可以有 H.248/Megaco 和 MGCP，目前更侧重于 H.248 协议，MGCP 协议只能支持 IP 承载的分组网络，H.248 能同时支持 IP 和 ATM 承载的分组网络。

软交换设备和接入设备（包括接入网关、综合接入系统、用户驻地网关等）之间的通信协议可以有 H.248/Megaco 和 MGCP。

信令网关的功能可以内置于软交换设备也可以外置。当信令网关内置于软交换设备中时，可采用专有协议，但要组建大的网络，还得采用外置信令网关的方式，此时信令网关的处理能力和性能都比内置信令网关的能力更强。信令网关和软交换设备之间的通信协议的用户适配层协议可以采用 M3UA、M2PA，同时信令网关还需要支持 SCTP，以便保证 M3UA 或 M2PA

信令消息能够在 IP 内进行可靠的传输。信令网关的另一端需要和现有的 NO.7 信令网相连, 所以信令网关还需要支持 NO.7 信令网中的相关协议, 如 MTP 三层协议。

为了支持不同类型的终端, 软交换设备同时还需要支持 SIP 协议、MGCP 协议、H.248/Megaco 协议, 具体能够支持哪些协议取决于软交换设备能够支持的终端类型。

软交换设备和应用服务器之间一般采用 SIP 协议。SIP 协议是一种基于文本的协议, 该协议可扩展性好, 对业务的描述也比较灵活, 同时解码的速度较快, 有专门的文本类解码器, 而在传统电路交换网中, 应用层协议一般采用 ASN.1 标准进行描述, 此时应用协议的解码要经过两级, 首先要对 ASN.1 进行解码, 然后是二进制解码。SIP 协议类似于 HTTP, 随着 Internet 的发展和 HTTP 的广泛应用, 人们对文本协议使用得更加灵活, 应用服务器采用 SIP 协议将有利于新业务的提供。

同时为了实现业务的开发能够向第三方开放, 应用服务器还可以提供各种开放的 API (如 Parlay、Jain、3GPP 的 OSA 和 OMG 的 TSAS)。应用服务器是一个独立的组件, 与控制层的软交换设备无关, 从而实现了业务与呼叫控制的分离, 有利于新业务的引入。

当软交换设备与现有 IP 电话网互通时, 软交换设备还需要支持 H.323 协议。软交换设备和 SIP 网络之间可以直接通过 SIP 协议进行互通。

不同软交换设备之间可以通过 SIP-T 或 BICC 协议进行互连, 目前从技术发展的趋势来看, 各厂家都比较认可 SIP-T 协议。但目前 SIP-T 协议还没有正式规范, 各个生产软交换设备的厂家大都是将 ISUP 协议装载到 SIP 协议中来实现。

软交换设备应该具有 SSF (业务交换功能), 以便能够进行智能业务的触发, 在分组网内实现智能网。同时, 为了实现和智能网的互通, 软交换设备应该支持 INAP 协议, 以便和业务控制网关功能 (SCGF) 直接相连。

软交换设备和认证系统、计费和账务系统以及操作维护管理系统之间采用的一般都是一些已经标准化的协议。

### 3. 软交换提供的业务

软交换的一大优势就是将业务提供和呼叫控制相分离。软交换设备仅负责基本的呼叫控制, 而将各种业务放到应用服务器上提供, 同时在应用服务器上还可以提供开放给第三方的 API 接口, 从而使业务的提供将更加灵活, 使网络对各种业务的适应能力更强。

软交换系统除能够提供语音业务外, 还能够提供数据业务和多媒体业务。软交换系统提供业务时, 首先要考虑到业务的继承性, 即软交换系统应该完全能够提供现有电路交换网中向用户提供的基本语音业务, 同时也应继续向用户提供各种补充业务, 如呼叫前转、呼叫限制等, 不仅能向传统电路交换网中的用户提供, 同时还应能向分组网中的各种终端提供这些业务。软交换对现在向用户提供的各类智能网业务也应该具有继承性, 如目前广泛应用的各类卡类业务、800 业务和 VPN 业务, 使用户不会因为使用了不同的网络技术而不能继续使用这些业务。另外还应提供一些特色业务。

## 4.2 软交换与 SIP 系统互通

### 4.2.1 软交换与 SIP 系统互通的功能要求

为完成软交换与 SIP 系统的互通, 软交换应具备以下功能:

- SIP 用户代理 (User Agent) 功能包括: SIP 用户代理客户机 (User Agent Client) 和用户代理服务器功能 (User Agent Server)。主要是代表 PSTN/ISDN 侧的非 SIP 终端向 IP 一侧发出 SIP 呼叫请求和对来自 IP 一侧的 SIP 呼叫做出响应。
- SIP 代理功能: 转发 SIP 请求和响应消息。
- 支持 SIP-T 协议: 实现 PSTN/ISDN 侧的 SS7 信令和 IP 侧的 SIP 信令的映射和转换。

### 4.2.2 互通方式

软交换与 SIP 系统的互通主要分为以下 3 种方式。

#### 1. 方式一: PSTN/ISDN—软交换网—SIP 网

该方式表示呼叫自 PSTN/ISDN 网发起, 终结于 IP 网。发端的软交换接收来自发端 PSTN/ISDN 网的 SS7 信令消息, 利用 SIP-T 协议将 SS7 信令消息转换封装成 SIP 消息, 通过中间的 SIP 网络直接传送给收端的 SIP 终端。其工作方式如图 4-2 所示。

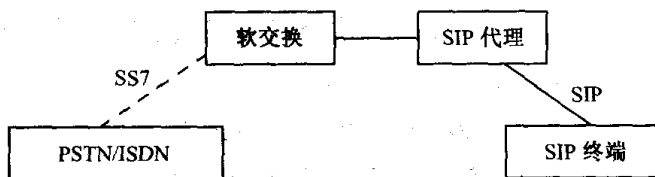


图 4-2 PSTN/ISDN-SIP 互通方式

#### 2. 方式二: SIP 网—软交换网—PSTN/ISDN

该方式表示呼叫自 IP 网发起, 终结于 PSTN/ISDN 网。发端的 SIP 终端发出 SIP 消息, 经过 SIP 网络将消息路由至收端的软交换, 软交换将 SIP 消息转换封装为 SS7 消息传送给收端的 PSTN/ISDN, 如图 4-3 所示。

#### 3. 方式三: SIP 网—软交换网—SIP 网

该方式表示呼叫自 IP 网发起, 终结于 IP 网, 此种是纯 SIP 网的情形; 发端的 SIP 终端发出 SIP 消息, 经过 SIP 网络将消息路由到收端的软交换, 如图 4-4 所示。

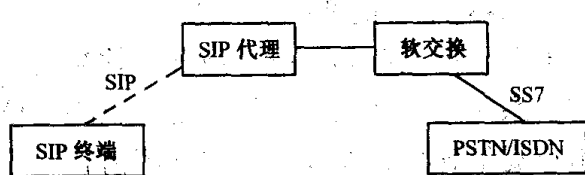


图 4-3 SIP-PSTN/ISDN 互通方式

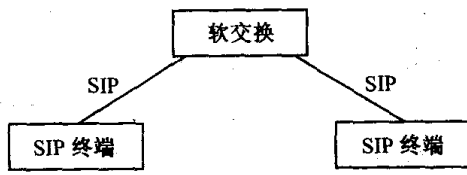


图 4-4 SIP-SIP 互通方式

### 4.2.3 呼叫控制流程

#### 1. PSTN/ISDN 端到 IP 端的呼叫建立和释放建立流程

PSTN/ISDN 端到 IP 端的呼叫建立和释放建立流程如图 4-5 所示, 本流程示例基于以下约定:

- 七号信令以 ISUP 为例;
- 连接主叫用户的发端局发出的 ISUP 信令发给图中的软交换;
- 代理服务器为被叫用户即 SIP 终端的代理服务器;

- SIP终端指具有SIP用户代理功能的实体。

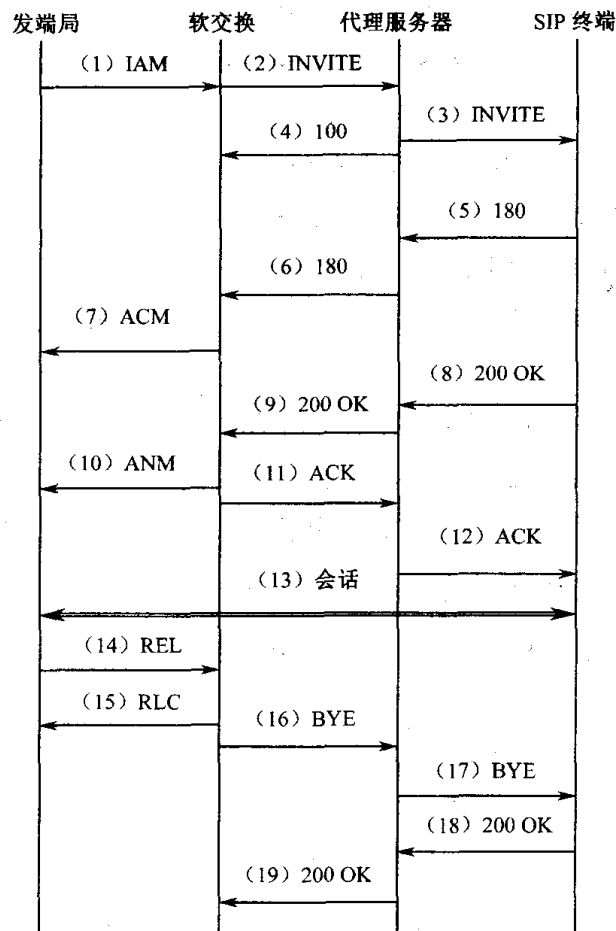


图 4-5 PSTN/ISDN 端到 IP 端的呼叫建立和释放建立流程

其中各个步骤含义如下：

- 步骤（1），ISDN 端的发端局收到主叫用户发出的呼叫建立请求消息，生成初始地址消息 IAM 送给软交换。
- 步骤（2），软交换收到 IAM 消息后，利用 SIP-T 将 IAM 消息组装成 SIP INVITE 请求消息发出，此 INVITE 消息将按照 SIP 系统的路由方式路由至代理服务器。
- 步骤（3），代理服务器将 INVITE 请求消息发送给被叫用户代理，即 SIP 终端。
- 步骤（4），代理服务器同时发送 100 Trying 响应给软交换，表明已收到 INVITE 请求，呼叫建立请求正被转发至目的地，但尚在进行中。
- 步骤（5），SIP 终端收到 INVITE 请求，向代理服务器发 180 Ringing 响应，表明其正在通知被叫。
- 步骤（6），代理服务器将收到的 180 响应转给软交换。
- 步骤（7），软交换收到 180 响应后，用 SIP-T 协议将 180 响应生成 ACM 消息送给发端局，消息中含被叫的当前状态信息。
- 步骤（8），被叫用户应答呼叫，SIP 终端向代理服务器发 200 OK 响应。
- 步骤（9），代理服务器将 200 OK 响应转给软交换。
- 步骤（10），软交换收到 200 消息后，将 200 消息转换成 ANM 消息发给发端局，发端

局将通知主叫用户。

- 步骤 (11), 软交换同时发 ACK 给代理服务器。
- 步骤 (12), 代理服务器将 ACK 消息转给 SIP 终端, 至此呼叫建立成功。
- 步骤 (13), 主叫、被叫进入通信阶段。
- 步骤 (14), 呼叫释放可由通信双方中的任一方发起, 假定由主叫方发出, 发端局收到主叫方送出的释放请求消息, 向软交换发 REL 消息。
- 步骤 (15), 软交换回送 RLC 消息给发端局。
- 步骤 (16), 软交换同时将 REL 消息转换成 BYE 消息发给代理服务器, BYE 消息表明主叫方释放呼叫。
- 步骤 (17), 代理服务器将 BYE 消息发给 SIP 终端。
- 步骤 (18), SIP 终端回送 200 OK 消息, 表明被叫释放呼叫。
- 步骤 (19), 代理服务器将 200 OK 响应转给软交换, 至此释放完成。

2. IP 端到 PSTN/ISDN 端的呼叫建立和释放建立流程

IP 端到 PSTN/ISDN 端的呼叫建立和释放建立流程如图 4-6 所示。本流程示例基于以下约定:

- 七号信令以 ISUP 为例;
- 代理服务器为主叫用户即 SIP 终端的代理服务器;
- 连接被叫用户的收端局发出的 ISUP 信令发给图中的软交换;
- SIP 终端为具有 SIP 用户代理功能的实体。

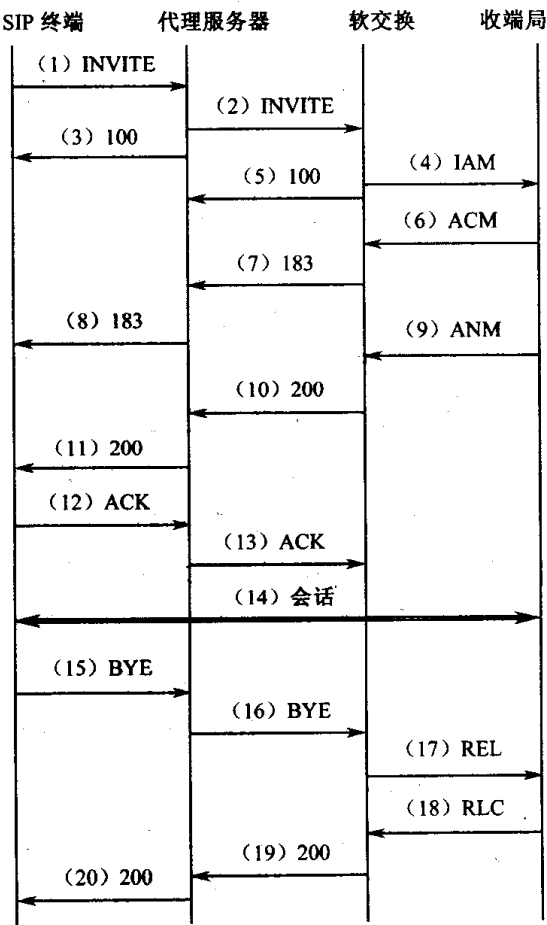


图 4-6 IP 端到 PSTN/ISDN 端的呼叫建立和释放建立流程

其中各个步骤含义如下:

- 步骤(1), IP侧的SIP终端向代理服务器发出呼叫建立请求INVITE消息。
- 步骤(2), 代理服务器收到INVITE请求, 转发INVITE请求, 此INVITE请求将按照SIP系统的路由方式路由至软交换。
- 步骤(3), 代理服务器同时向SIP终端发100 Trying响应, 表明已转发INVITE请求, 但尚在进行中。
- 步骤(4), 软交换收到INVITE消息, 利用SIP-T协议将INVITE消息封装成IAM消息发出, 此IAM消息将被送至被叫所在的收端局。
- 步骤(5), 软交换同时向代理服务器回送100 Trying响应, 表明已转发INVITE请求至目的地, 但尚在进行中。
- 步骤(6), 收端局收到IAM消息后, 分析被叫用户号码, 检查被叫的情况, 向软交换发送地址全消息(ACM), ACM消息中含有被叫的当前状态等附加信息。
- 步骤(7), 软交换收到ACM消息后, 利用SIP-T协议生成SIP 183 Session Progress响应传送给代理服务器, 此时, 183消息含呼叫建立期间的状态信息。
- 步骤(8), 代理服务器转发183响应给SIP终端。
- 步骤(9), 被叫用户应答呼叫, 收端局收到被叫用户发送的连接消息, 向软交换发送应答消息(ANM)。
- 步骤(10), 软交换收到ANM消息后, 利用SIP-T将ANM消息转换成200 OK响应后发出, 此消息将按SIP系统的路由方式发送给代理服务器。
- 步骤(11), 代理服务器转发收到的200消息给SIP终端。
- 步骤(12), SIP终端收到200响应后, 发送ACK消息给代理服务器, 表明其知道被叫应答呼叫。
- 步骤(13), 代理服务器将ACK消息转发给软交换, 至此呼叫建立成功。
- 步骤(14), 主叫、被叫进入通信阶段。
- 步骤(15), 呼叫释放可由通信双方中的任一方发起, 假定由SIP终端发出, SIP终端向其代理服务器发出BYE消息, 表明其释放呼叫。
- 步骤(16), 代理服务器收到BYE消息后, 将其转发给软交换。
- 步骤(17), 软交换收到BYE消息后, 将BYE消息转换成REL消息发送给收端局。
- 步骤(18), 收端局收到REL消息后, 向软交换回送释放完成消息(RLC), 表明被叫释放呼叫。
- 步骤(19), 软交换收到RLC消息后, 将RLC转换成200 OK响应发给代理服务器。
- 步骤(20), 代理服务器将200响应转给SIP终端, 至此释放完成。

### 3. SIP终端到SIP终端的呼叫建立和释放流程

SIP终端到SIP终端的呼叫建立和释放流程分别如图4-7、4-8所示, 其中软交换应包含代理服务器功能、重定向服务器功能和/或定位服务器功能。

图4-7为经过代理服务器的成功邀请所建立的流程图, 图4-8为经过重定向服务器的成功邀请所建立的流程图。



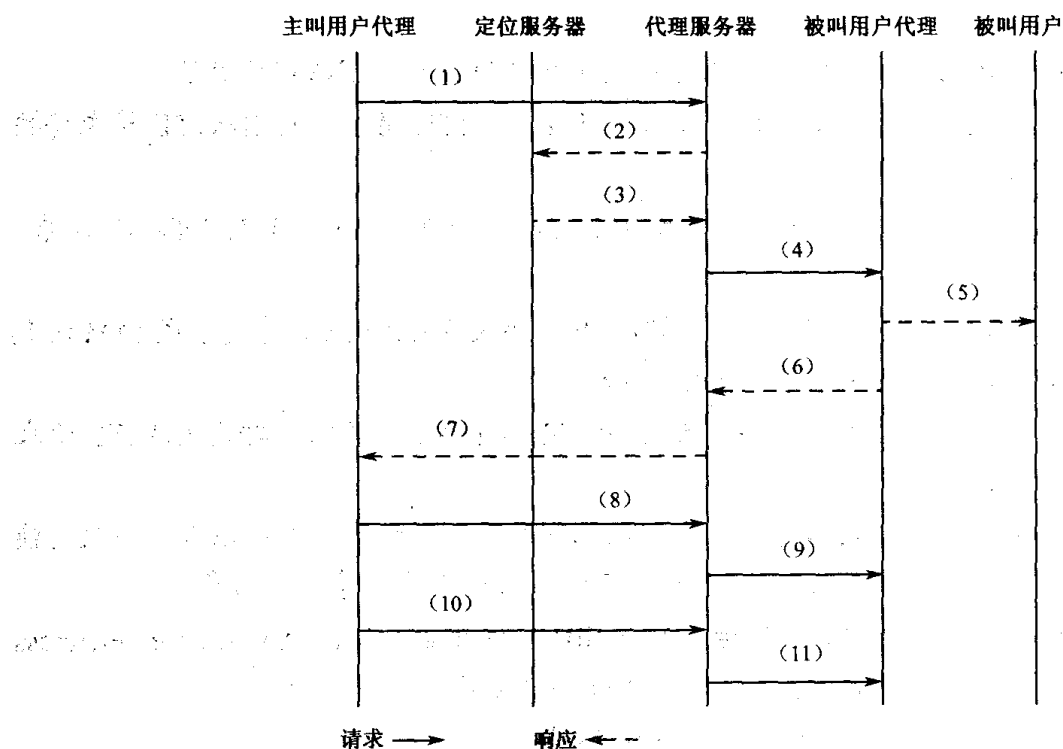


图 4-7 成功邀请建立的流程图 (经代理服务器)

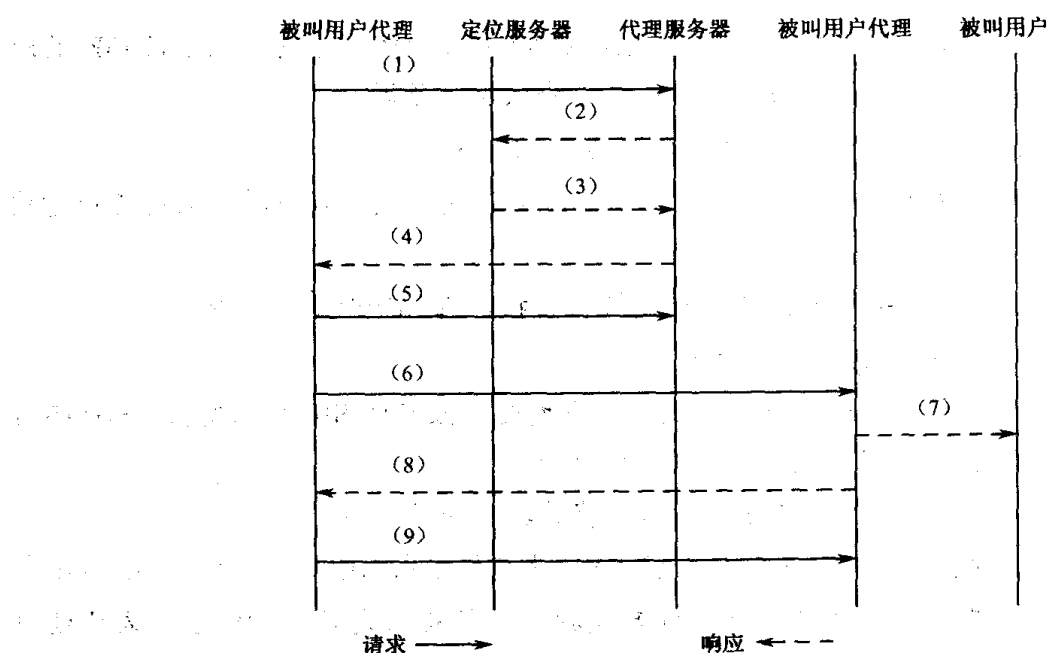


图 4-8 成功邀请建立的流程图 (经重定向服务器)

其中各个步骤含义如下:

- 步骤 (1), 主叫用户代理发送 INVITE 请求到代理服务器。
- 步骤 (2), 代理服务器收到 INVITE 请求后, 连接定位服务器。
- 步骤 (3), 定位服务器向代理服务器返回被叫用户的准确位置。
- 步骤 (4), 利用定位服务器返回的地址, 代理服务器发送 INVITE 请求给被叫用户代理。
- 步骤 (5), 被叫用户代理收到请求后, 提醒被叫用户。
- 步骤 (6), 被叫用户代理向代理服务器发送 200 成功响应。

- 步骤(7), 代理服务器将 200 成功结果传送给主叫用户代理。
- 步骤(8)、(9), 主叫用户代理向被叫用户代理发 ACK 请求。
- 步骤(10)、(11) 若会话中的任一方想终止会话(假设为主叫), 通过自己的用户代理向代理服务器发出 BYE 请求, 代理服务器再将 BYE 请求转给对方的用户代理。

其中各个步骤含义如下:

- 步骤(1), 主叫用户代理发 INVITE 请求到重定向服务器。
- 步骤(2), 重定向服务器收到 INVITE 请求后, 连接定位服务器。
- 步骤(3), 定位服务器向重定向服务器返回被叫用户的准确位置。
- 步骤(4), 重定向服务器用 302 响应将被叫用户地址发给主叫用户代理。
- 步骤(5), 主叫用户代理向重定向服务器发 ACK 请求进行确认。
- 步骤(6), 主叫用户代理直接向被叫用户代理发 INVITE 请求。
- 步骤(7), 被叫用户代理收到请求后, 提醒被叫用户。
- 步骤(8), 被叫用户代理发送 200 成功响应给主叫用户代理。
- 步骤(9), 主叫用户代理向被叫用户代理发 ACK 请求。

#### 4.2.4 软交换位于 SIP 系统中与现有智能网的互通

在 SIP 系统中, 如果与 SIP 终端相连的 SIP 代理服务器不具备 SSF 功能, 无法触发 IN 业务, 则可通过与该 SIP 代理服务器相连的软交换来完成 IN 业务的触发。如果软交换直接与 SIP 终端相连, 则也由软交换完成 IN 的触发。

下面以与 SIP 终端相连的 SIP 代理服务器不具备 SSF 功能为例, 说明互通流程。

本流程示例基于以下约定:

- 七号信令以 ISUP 为例;
- 代理服务器为主叫用户即 SIP 终端的代理服务器;
- 连接被叫用户的收端局发出的 ISUP 信令发给图中的软交换;
- SIP 终端为具有 SIP 用户代理功能的实体。

具体流程如图 4-9 所示。

其中各个步骤含义如下:

- 步骤(1), IP 侧的 SIP 终端向代理服务器发出呼叫建立请求 INVITE 消息。
- 步骤(2), 代理服务器收到 INVITE 请求后, 转发 INVITE 请求, 此 INVITE 请求将按照 SIP 系统的路由方式路由至软交换。
- 步骤(3), 代理服务器同时向 SIP 终端发 100 Trying 响应, 表明已转发 INVITE 请求, 但尚在进行中。
- 步骤(4), 软交换向 S GF 发送 INAP/IP 操作 IDP, 报告触发 800 业务, SGF 向 SCF 发送 INAP/TC 操作 IDP, 向 SCF 报告触发 800 业务。
- 步骤(5), SCF 向 S GF 发送 INAP/TC 操作 RRBE 和 CONNECT, 要求软交换监视遇忙、无应答和挂机事件, S GF 将这两个操作转换为 INAP/IP 操作, 然后向软交换发送。
- 步骤(6), 软交换收到 INVITE 消息, 利用 SIP-T 协议将 INVITE 消息封装成 IAM 消息发出, 此 IAM 消息将被送至被叫所在收端局。

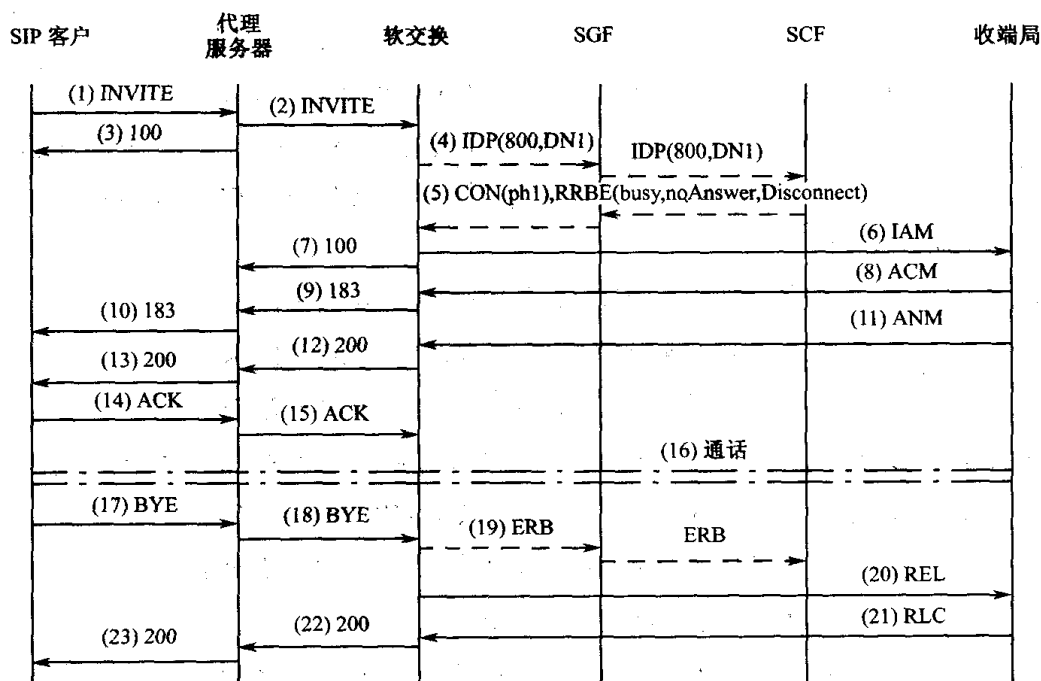


图 4-9 SIP 终端发起 IN 呼叫

- 步骤 (7), 软交换同时向代理服务器回送 100 Trying 响应, 表明已转发 INVITE 请求至目的地, 但尚在进行中。
- 步骤 (8), 收端局收到 IAM 消息后, 分析被叫用户号码, 检查被叫的情况, 向软交换发送地址全消息 (ACM), ACM 消息中含有被叫的当前状态等附加信息。
- 步骤 (9), 软交换收到 ACM 消息后, 利用 SIP-T 协议生成 SIP 183 Session Progress 响应送给代理服务器, 183 消息含呼叫建立期间的状态信息。
- 步骤 (10), 代理服务器转发 183 响应给 SIP 终端。
- 步骤 (11), 被叫用户应答呼叫, 收端局收到被叫用户发送的连接消息后, 向软交换发送应答消息 (ANM)。
- 步骤 (12), 软交换收到 ANM 消息后, 利用 SIP-T 将 ANM 消息转换成 200 OK 响应后发出, 此消息将按 SIP 系统的路由方式发送给代理服务器。
- 步骤 (13), 代理服务器转发收到的 200 消息给 SIP 终端。
- 步骤 (14), SIP 终端收到 200 响应后, 发 ACK 消息给代理服务器, 表明其知道被叫应答呼叫。
- 步骤 (15), 代理服务器将 ACK 消息转发给软交换, 至此呼叫建立成功。
- 步骤 (16), 主叫、被叫进入通信阶段。
- 步骤 (17), 呼叫释放可由通信双方中的任一方发起, 假定由 SIP 终端发出, SIP 终端向其代理服务器发出 BYE 消息, 表明其释放呼叫。
- 步骤 (18), 代理服务器收到 BYE 消息后, 将其转发给软交换。
- 步骤 (19), 软交换向 SGF 发送 INAP/IP 操作 ERB, SGF 向 SCF 发送 INAP/TC 操作 ERB, 向 SCF 报告发生挂机事件。
- 步骤 (20), 软交换收到 BYE 消息后, 将 BYE 消息转换成 REL 消息发送给收端局。
- 步骤 (21), 收端局收到 REL 消息, 向软交换回送释放完成消息 (RLC), 表明被叫释放。

呼叫。

- 步骤 (22), 软交换收到 RLC 消息后, 将 RLC 转换成 200 OK 响应发送给代理服务器。
- 步骤 (23), 代理服务器将 200 响应转给 SIP 终端, 至此释放完成。

4.2.5 基于 SIP 的软交换互通

图 4-10 是软交换之间的呼叫流程。

下面是软交换之间 SIP 消息的格式:

◆ INVITE

消息如下:

INVITE sip:66500002@191.169.1.116:5060;User=phone  
SIP/2.0  
Allow:INVITE,ACK,BYE,CANCEL,PRACK...; 允许  
From:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 发端  
To:<sip:66500002@191.169.1.116:5060;User=phone>; 收端  
Supported:100rel  
Content-Type:multipart/mixed;boundary=unique-boundary-1; 内容  
Call-ID:01F01A709DA1400000000001@191.169.1.112; 呼叫 ID  
CSeq:1 INVITE  
Expires:180; 过期时间  
Via:SIP/2.0/UDP 191.169.1.112:5061; branch=  
Record-Route: <sip: 191.169.1.112;lr>  
Contact:<sip:7670000@191.169.1.112:5061>  
Max-Forwards:70; 最大转发次数  
Content-Length:597; 内容长度  
--unique-boundary-1  
Content-Type:application/SDP  
v=0  
o=SoftX 1073741827 1073741827 IN IP4 191.169.1.112  
s=Sip Call  
c=IN IP4 191.169.1.45; 连接  
t=0 0  
m=audio 14380 RTP/AVP 8 0 4 18; 媒体类型  
a=rtpmap:8 PCMA/8000  
a=rtpmap:0 PCMU/8000  
a=rtpmap:4 G723/8000  
a=rtpmap:18 G729/8000

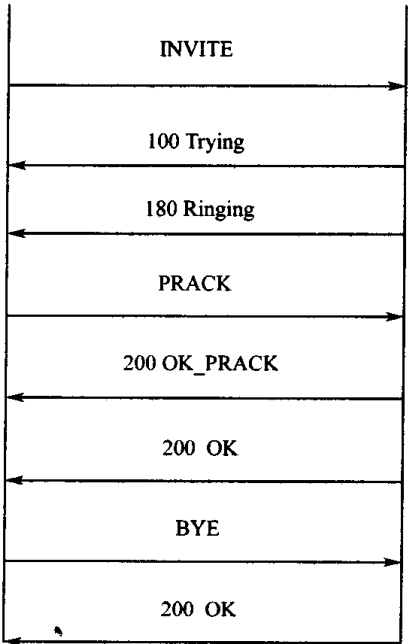


图 4-10 软交换之间的呼叫流程

--unique-boundary-1

Content-Type:application/ISUP; version=itu-t92+;base= itu-t92+

Content-Disposition: signal; handling=optional

01 00 20 00 0A 03 02 09 07 83 90 66 05 00 20 0F 08 01 00 0A 06 81 13 67 07 00 00 00

--unique-boundary-1--

◆ SIP/2.0 100 Trying

消息如下:

Call-ID:01F01A709DA1400000000001@191.169.1.112

CSeq:1 INVITE

From:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 发端

To:<sip:66500002@191.169.1.116:5060;User=phone>; 收端

Via:SIP/2.0/UDP 191.169.1.112:5061;branch=

Content-Length:0

◆ SIP/2.0 180 Ringing

消息如下:

Allow:INVITE,ACK,BYE,CANCEL,PRACK

Require:100rel

Supported:100rel

Via:SIP/2.0/UDP 191.169.1.112:5061

Call-ID:01F01A709DA1400000000001@191.169.1.112

CSeq:1 INVITE

From:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 发端

To:<sip:66500002@191.169.1.116:5060;User=phone>;tag=5358336B534F7A5F0021649B; 收端

RSeq:633013647

Contact:<sip:66500002@191.169.1.116:5061>

Content-Length:4; 内容长度

Content-Type:application/ISUP

06 C2 00 00

180 中任选包含 SDP 信息

◆ PRACK

消息如下:

PRACK sip:66500002@191.169.1.116:5060;User=phone SIP/2.0

Call-ID:01F01A709DA1400000000001@191.169.1.112

CSeq:2 PRACK

From:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 发端

To:<sip:66500002@191.169.1.116:5060;User=phone>;tag=5358336B534F7A5F0021649B; 收端

RAck:633013647 1 INVITE

Via:SIP/2.0/UDP

191.169.1.112:5061;branch=898E03EE1DBA24C581E03DB69E23FF01.80000001

Max-Forwards:70; 最大转发次数

Content-Length:0; 长度

◆ SIP/2.0 200 OK

消息如下:

Call-ID:01F01A709DA1400000000001@191.169.1.112

CSeq:2 PRACK

From:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 发端

To:<sip:66500002@191.169.1.116:5060;User=phone>;tag=5358336B534F7A5F0021649B; 收端

Via:SIP/2.0/UDP

191.169.1.112:5061;branch=898E03EE1DBA24C581E03DB69E23FF01.80000001

Content-Length:0; 内容长度

◆ SIP/2.0 200 OK

消息如下:

Content-Type:multipart/mixed;boundary=unique-boundary-1

Via:SIP/2.0/UDP 191.169.1.112:5061

Call-ID:01F01A709DA1400000000001@191.169.1.112; 呼叫 ID

CSeq:1 INVITE

From:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 发端

To:<sip:66500002@191.169.1.116:5060;User=phone>;tag=5358336B534F7A5F0021649B; 收端

Contact:<sip:66500002@191.169.1.116:5061>

Content-Length:298; 长度

--unique-boundary-1

Content-Type:application/SDP; 内容类型

v=0

o=SS 1073741851 1073741851 IN IP4 191.169.1.116

s=Sip Call

c=IN IP4 191.169.1.35; 连接

t=0 0

m=audio 30000 RTP/AVP 8; 媒体类型

a=rtpmap:8 PCMA/8000

--unique-boundary-1

Content-Type:application/ISUP; version=itu-t92+;base=itu-t92+

Content-Disposition: signal; handling=optional

09 01 11 03 C6 00 00 00

--unique-boundary-1--

◆ ACK sip:66500002@191.169.1.116:5061 SIP/2.0

消息如下:

Call-ID:01F01A709DA1400000000001@191.169.1.112; 呼叫 ID

CSeq:1 ACK

From:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 发端

To:<sip:66500002@191.169.1.116:5060;User=phone>;tag=5358336B534F7A5F0021649B; 收端

Via:SIP/2.0/UDP 191.169.1.112:5061

Max-Forwards:70; 最大转发次数

Content-Length:0; 长度

#### ◆ BYE

消息如下:

BYE sip:7670000@191.169.1.112:5061 SIP/2.0

Call-ID:01F01A709DA1400000000001@191.169.1.112; 呼叫 ID

CSeq:1 BYE

From:<sip:66500002@191.169.1.116:5060;User=phone>;tag=5358336B534F7A5F0021649B; 发端

To:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 收端

Via:SIP/2.0/UDP 191.169.1.116:5061

Max-Forwards:70; 最大转发次数

Content-Length:6

Content-Type:application/ISUP; version=itu-t92+;base= itu-t92+; 内容类型

Content-Disposition: signal; handling=optional

0c 01 02 65 9f 00

BYE 中应该封装 Rel 的信息,接收方如果在接收到不封装 Rel 的 BYE 时应该正常地接收处理。

#### ◆ SIP/2.0 200 OK

消息如下:

Via:SIP/2.0/UDP 191.169.1.116:5061

Call-ID:01F01A709DA1400000000001@191.169.1.112; 呼叫 ID

CSeq:1 BYE

From:<sip:66500002@191.169.1.116:5060;User=phone>;tag=5358336B534F7A5F0021649B; 发端

To:<sip:7670000@191.169.1.112:5061;User=phone>;tag=446B5358336B534F00009820; 收端

Content-Length:0

### 4.2.6 软交换与应用服务器间的交互

软交换与应用服务器之间使用 SIP 协议进行通信。SIP 协议具有建立、拆除和管理端点间会话的功能,所以软交换既能建立也可取消至应用服务器的呼叫,同样,应用服务器也能建立和取消至软交换的呼叫,并且应用服务器还具有转换主叫和被叫方的信息,保持和恢复连接的功能。目前,对 SIP 协议功能正进行扩展工作,未来 SIP 将能支持多方功能。

软交换和应用服务器都可作为 SIP 用户代理,它们可以直接通过代理服务器进行通信。软交换可利用注册机制动态得知应用服务器的存在,也可通过在软交换上配置应用服务器的地址信息,静态得知应用服务器的存在。源报头用于通知应用服务器主叫方的身份,而终报头用于通知应用服务器增值业务将应用的那一方的身份。如图 4-11 所示说明了 SIP 实现各种增值业务控制流的流程。

软交换和应用服务器之间控制流交互的一般流程如下:

- 软交换触发决定呼叫是否切换至应用服务器进行增值业务处理。触发可基于主叫方地址、被叫方地址或其他软交换机制。

- 软交换根据触发信息确定应用服务器的地址，并通过发送 SIP 请求信息（包括适当的呼叫信息）将呼叫转至目标应用服务器。

目标应用服务器收到 SIP 请求后，调用相应的增值业务。在此处，应用服务器可进行许多动作：

- 重定位—向软交换发送一个新的目的地址，重定向呼叫（3XX 重定向响应中包含一个新的目的地址）。这种机制可用于面向地址转换和路由的业务。
- 接收和传送—分配媒体资源，命令软交换将至媒体资源的路径连接好（用 200 OK 响应表示）。用户与媒体资源的交互结束以后，应用可将呼叫传至新的目的地，应用退出呼叫（BYE 的头中包含有新的目的地址）。此机制可用于面向媒体的业务，例如，卡号和传真存储/转发业务。
- 代理—通过软交换将呼叫返回。使得应用服务器可监视所有的并发呼叫事件。此机制可用于面向事件的业务，如记账卡和计时业务。

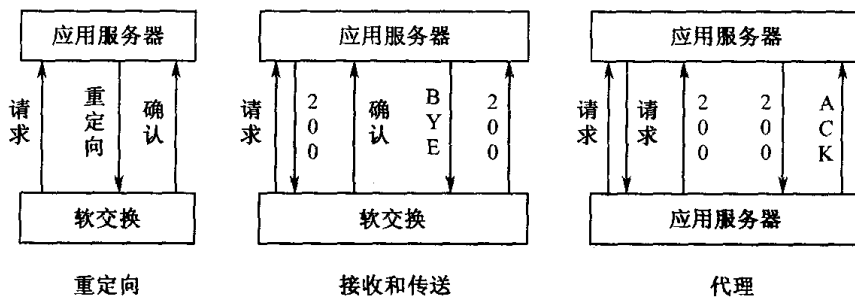


图 4-11 控制流

基本的 SIP 功能和扩展的呼叫控制功能结合在一起，可使软交换将呼叫转至应用服务器进行增值业务处理，处理完以后，应用服务器通过软交换将呼叫转回，并将自己从呼叫中退出。SIP 协议可使应用服务器进入所有的呼叫控制活动并能传送、重定向和代理呼叫。由于 SIP 的通用性和灵活性，使得在软交换网络中可以非常有效和容易地实现增值业务。

## 4.3 使用 SIP 开展软交换业务

### 4.3.1 使用 SIP 开展 Parlay 业务

#### 1. 呼叫建立流程

呼叫建立流程如图 4-12 所示。

其中各个步骤含义如下：

- 步骤 1)，SIP PPS 应用调用 enableCallNotification，请求应用服务器监视网络信息；
- 步骤 2)，用户 A 拨打 SIP PPS 业务接入码，软交换发送 Invite 消息到应用服务器；
- 步骤 3)，应用服务器给软交换回送 100 响应；
- 步骤 4)，应用服务器调用 eventNotify API，向应用报告用户 A 的拨号信息；
- 步骤 5)，应用调用 createUICall，要求应用服务器准备建立到媒体服务器的连接；
- 步骤 6)，应用调用 sendInfoAndCollectReq，放音请用户选择提示语音的种类；
- 步骤 7)，应用服务器向媒体服务器发送 Invite 消息并携带需要放音的参数；



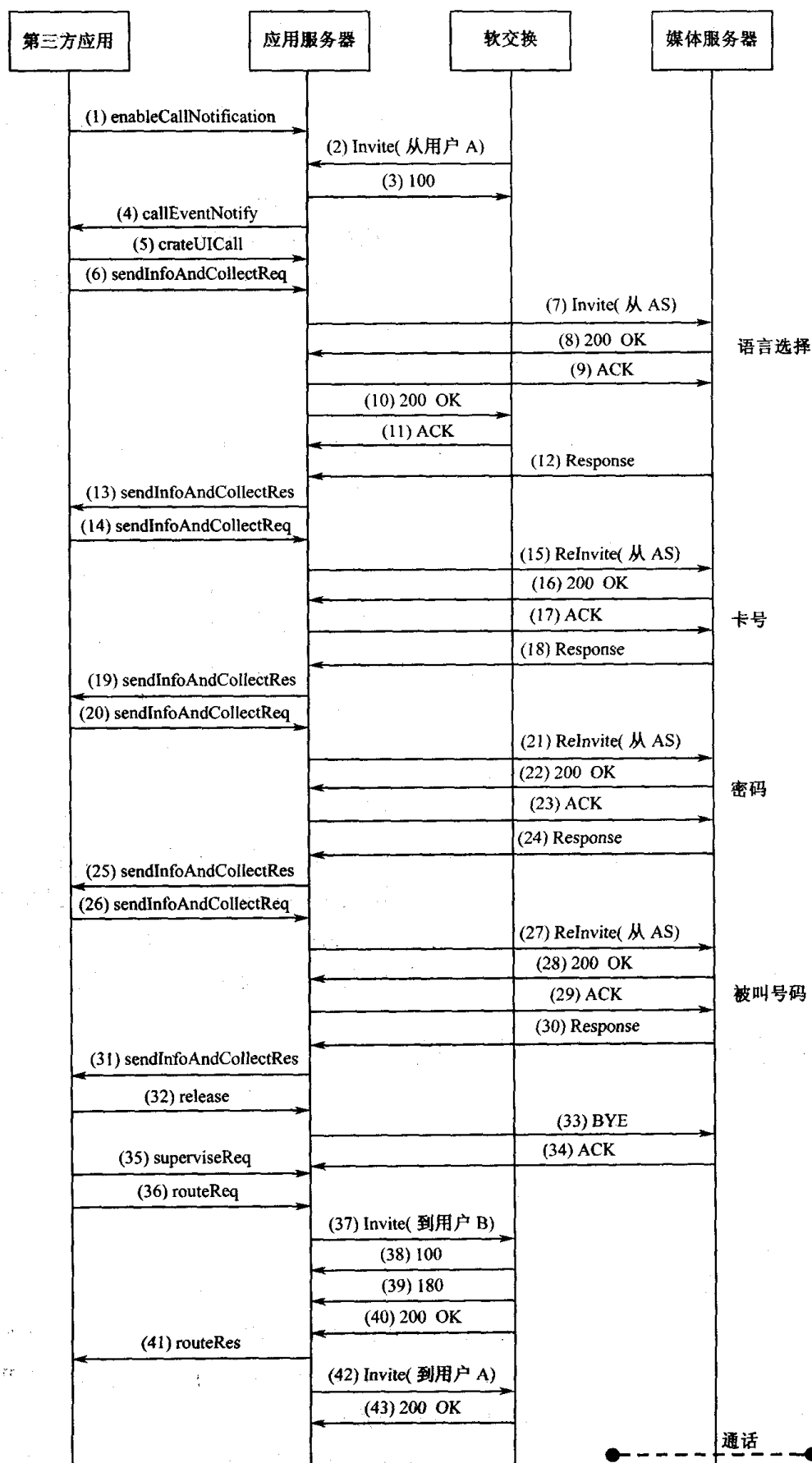


图 4-12 呼叫建立流程图

- 步骤 8), 媒体服务器向应用服务器回送 200 OK 响应;

- 步骤9), 应用服务器向媒体服务器回送 ACK 响应, 表示步骤7)中软交换发送的 Invite 建立成功;
- 步骤10), 应用服务器向软交换回送 200 OK 响应;
- 步骤11), 软交换向应用服务器回送 ACK 相应, 表示步骤2)中软交换发送的 Invite 建立成功;
- 步骤12), 媒体服务器向应用服务器发送 Response, 通知放音完成, 并把收集到的用户输入信息通知给应用服务器;
- 步骤13), 应用服务器回送 sendInfoAndCollectRes, 将用户的选择传送给应用;
- 步骤14), 应用调用 sendInfoAndCollectReq, 放音请用户输入卡号;
- 步骤15), 应用服务器发送 (Re) Invite 到媒体服务器, 要求媒体服务器放音并收集用户输入信息;
- 步骤16), 媒体服务器向应用服务器回送 200 OK 响应;
- 步骤17), 应用服务器向媒体服务器回送 ACK 响应;
- 步骤18), 媒体服务器向应用服务器发送 Response, 通知放音完成, 并把收集到的用户输入信息通知给应用服务器。
- 步骤19), 应用服务器回送 sendInfoAndCollectRes, 将用户输入的卡号送给应用;
- 步骤20), 应用调用 sendInfoAndCollectReq, 放音请用户输入密码;
- 步骤21), 应用服务器发送 (Re) Invite 到媒体服务器, 要求媒体服务器放音并收集用户输入信息;
- 步骤22), 媒体服务器向应用服务器回送 200 OK 响应;
- 步骤23), 应用服务器向媒体服务器回送 ACK 响应;
- 步骤24), 媒体服务器向应用服务器发送 Response, 通知放音完成, 并把收集到的用户输入信息通知给应用服务器;
- 步骤25), 应用服务器回送 sendInfoAndCollectRes, 将用户输入的密码送给应用;
- 步骤26), 应用调用 sendInfoAndCollectReq, 放音请用户输入被叫号码;
- 步骤27), 应用服务器发送 (Re) Invite 到媒体服务器, 要求媒体服务器放音并收集用户输入信息;
- 步骤28), 媒体服务器向应用服务器回送 200 OK 响应;
- 步骤29), 应用服务器向媒体服务器回送 ACK 响应;
- 步骤30), 媒体服务器向应用服务器发送 Response, 通知放音完成, 并把收集到的用户输入信息通知给应用服务器;
- 步骤31), 应用服务器回送 sendInfoAndCollectRes, 将用户输入的被叫号码送给应用;
- 步骤32), 应用调用 Release, 准备释放媒体服务器通道;
- 步骤33), 应用服务器发送 BYE 给媒体服务器, 释放通道;
- 步骤34), 将媒体服务器回送 200 OK;
- 步骤35), 应用调用 superviseReq, 请求监视呼叫时长;
- 步骤36), 应用调用 routerReq, 要求应用服务器建立到被叫用户 B 的连接;

- 步骤 37), 应用服务器向软交换发送 Invite, 请求软交换建立到用户 B 的连接;
- 步骤 38), 软交换回送 100;
- 步骤 39), 软交换回送 180;
- 步骤 40), 软交换回送 200 OK;
- 步骤 41), 应用服务器回调 routeRes, 报告应用被叫 B 已经应答;
- 步骤 42), 应用服务器发送给用户 A 的 Invite 消息到软交换, 指示将用户 A 和 B 的媒体流相连;
- 步骤 43), 软交换回送 200 OK; 用户 A 和用户 B 进入通话流程。

## 2. 呼叫过程监视流程

呼叫过程监视流程如图 4-13 所示。

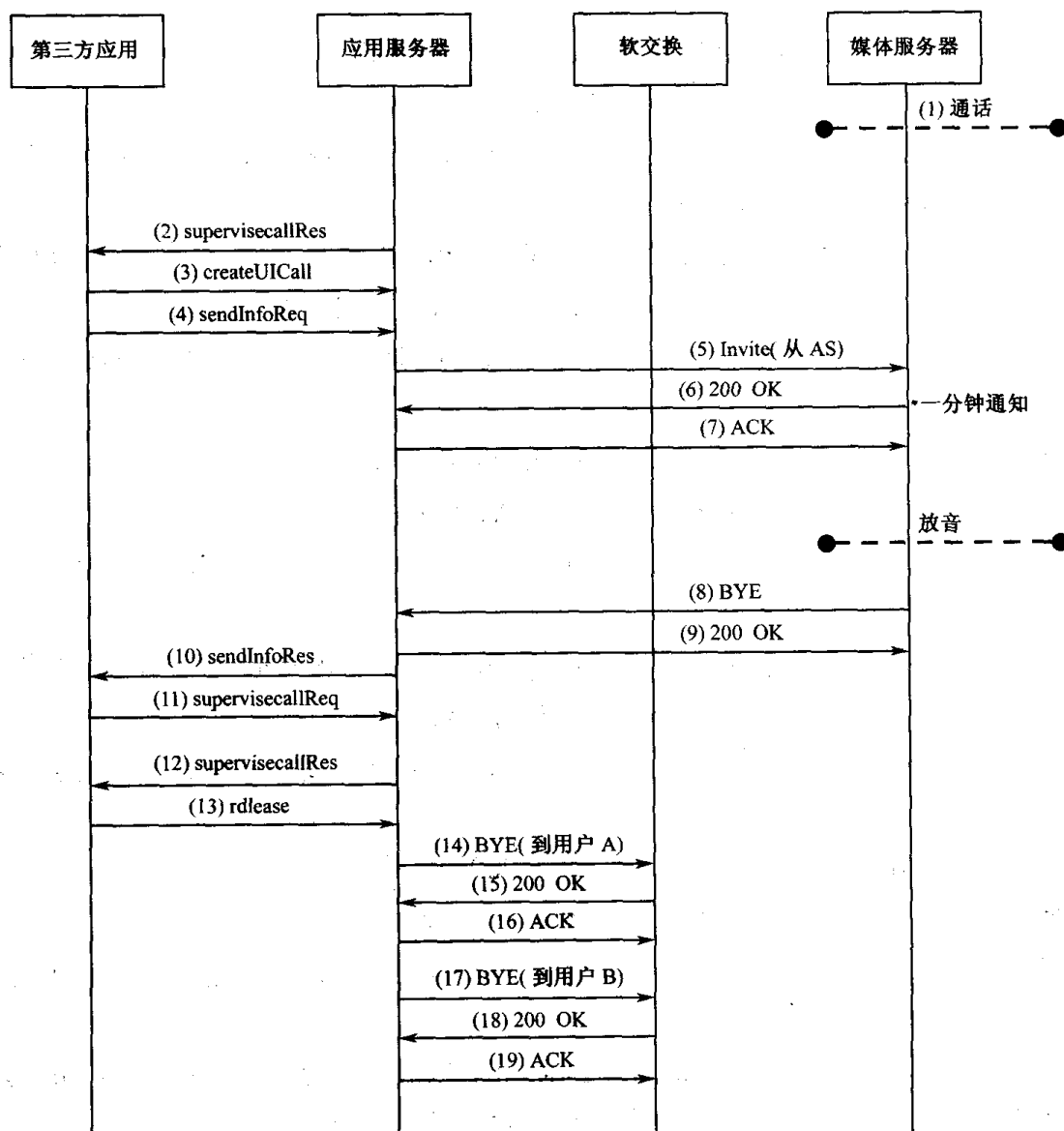


图 4-13 呼叫过程监视流程

图中各个步骤的含义如下:

- 步骤 1), 通话;
- 步骤 2), 应用服务器监视呼叫时长, 当请求监视的时间到达时, 回送 superviseCallRes;

- 步骤3), 应用调用 createUICall, 要求应用服务器建立到媒体服务器的连接;
- 步骤4), 应用调用 sendInfoReq, 放音提示用户通话时间仅能持续1分钟;
- 步骤5), 应用服务器向媒体服务器发 Invite, 要求媒体服务器放音;
- 步骤6), 媒体服务器向应用服务器回送 200 OK 信息;
- 步骤7), 应用服务器向媒体服务器发送 ACK;
- 步骤8), 媒体服务器放音完毕发送 BYE;
- 步骤9), 应用服务器回送 200 OK;
- 步骤10), 应用服务器回送 sendInfoRes;
- 步骤11), 应用调用 superviseCallReq, 请求监视呼叫时长;
- 步骤12), 应用服务器监视呼叫时长, 当请求监视的时间到达时, 回送 superviseCallRes;
- 步骤13), 应用调用 realease, 请求释放呼叫;
- 步骤14), 应用服务器向软交换发送 BYE 给用户 A;
- 步骤15), 软交换向应用服务器回送 200 OK;
- 步骤16), 应用服务器回送 ACK;
- 步骤17), 应用服务器向软交换发送 BYE 给用户 B;
- 步骤18), 软交换向应用服务器回送 200 OK, 呼叫释放;
- 步骤19), 应用服务器回送 ACK。

### 3. 被叫正常挂机流程

被叫正常挂机流程如图 4-14 所示。

图 4-14 中各个步骤的含义如下:

- 步骤1), 被叫用户 B 挂机, 软交换向应用服务器发送 Bye 消息;
- 步骤2), 应用服务器回送 superviseCallRes, 通知呼叫时长;
- 步骤3), 应用服务器调用 routeRes, 通知应用, 被叫已挂机;
- 步骤4), 应用调用 createUICall, 应用服务器建立到媒体服务器的连接;
- 步骤5), 应用调用 sendInfoAndCollectReq, 放音通知用户被叫用户已挂机, 请用户拨再需要呼叫的被叫号码或按 # 键挂机;
- 步骤6), 应用服务器向媒体服务器发送 Invite 到媒体服务器, 要求媒体服务器放音并收集信息;
- 步骤7), 媒体服务器向应用服务器回送 200 OK 信息;
- 步骤8), 应用服务器向媒体服务器发送 ACK;
- 步骤9), 媒体服务器向应用服务器回送 Response 信息;
- 步骤10), 应用服务器回送 sendInfoAndCollectRes, 将用户选择送给应用;
- 步骤11), 应用服务器发送 BYE 到媒体服务器, 释放通道;
- 步骤12), 媒体服务器回送 200 OK;
- 步骤13), 应用调用 realease, 请求释放呼叫;
- 步骤14), 应用服务器向软交换发送 BYE 给用户 A, 释放呼叫;

- 步骤 15), 软交换向应用服务器回送 200 OK, 呼叫释放;
- 步骤 16), 应用服务器回送 ACK。

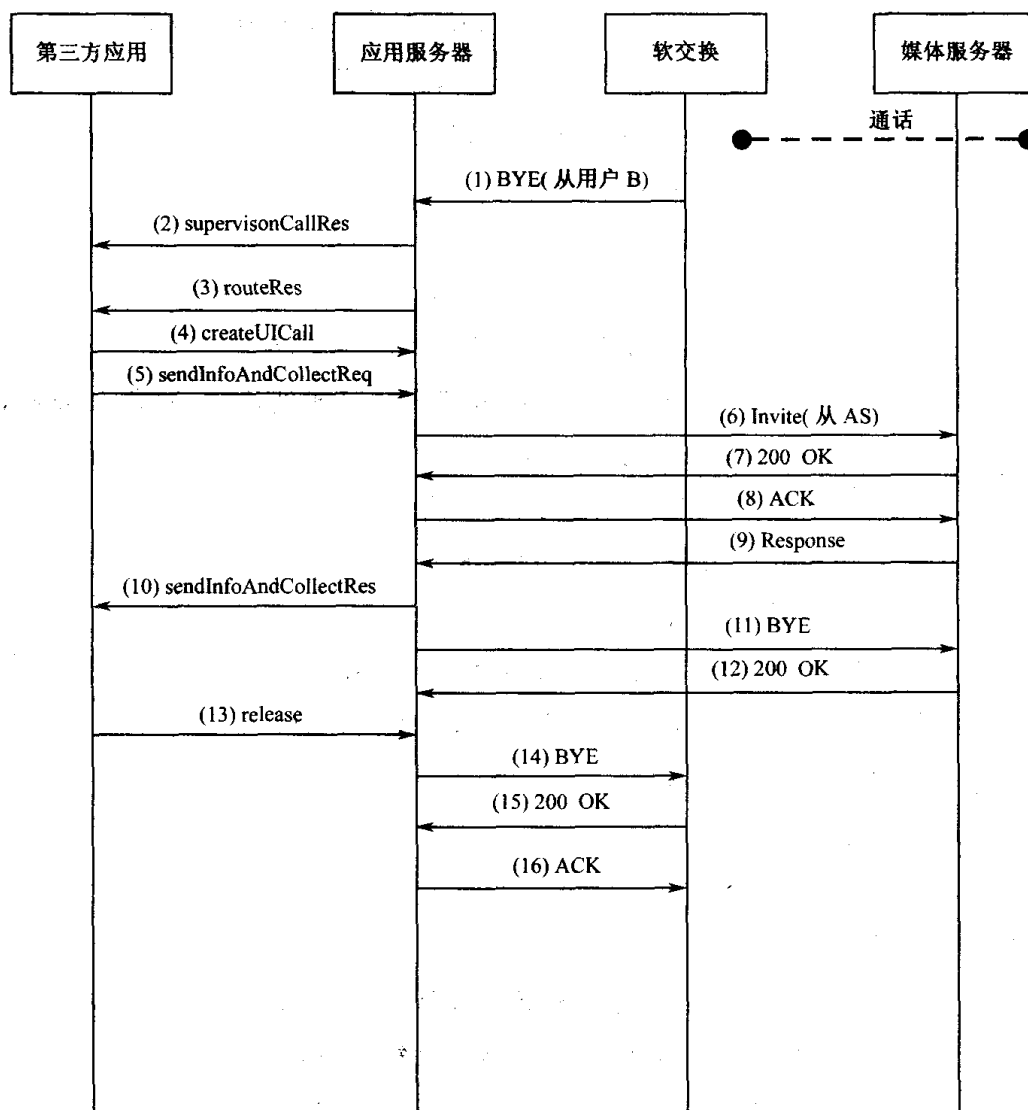


图 4-14 被叫正常挂机流程

#### 4. 主叫正常挂机流程

主叫正常挂机流程如图 4-15 所示。

图中各个步骤的含义如下:

- 步骤 1), 主叫用户 A 挂机, 软交换向应用服务器发送 Bye 消息;
- 步骤 2), 应用服务器回送 superviseCallRes, 通知呼叫时长;
- 步骤 3), 应用服务器调用 routeRes, 通知应用, 主叫已挂机;
- 步骤 4), 应用调用 realease, 请求释放呼叫;
- 步骤 5), 应用服务器向软交换发送 BYE 给用户 B, 释放呼叫;
- 步骤 6), 软交换向应用服务器回送 200OK, 呼叫释放;
- 步骤 7), 应用服务器回送 ACK。

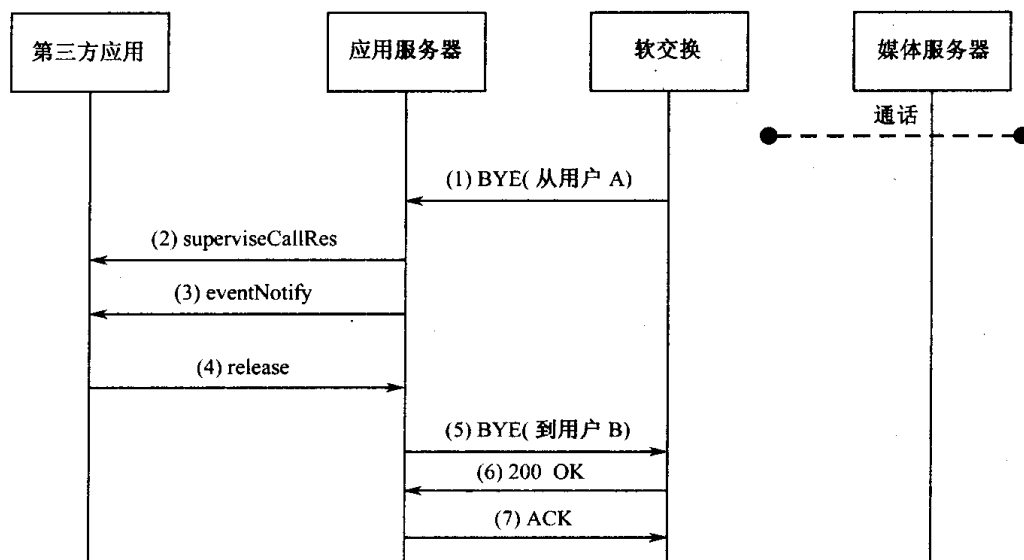


图 4-15 主叫正常挂机流程

### 5. 被叫忙流程

被叫忙流程如图 4-16 所示。在此假定卡号和密码鉴权已经通过（具体流程参见呼叫建立流程）。

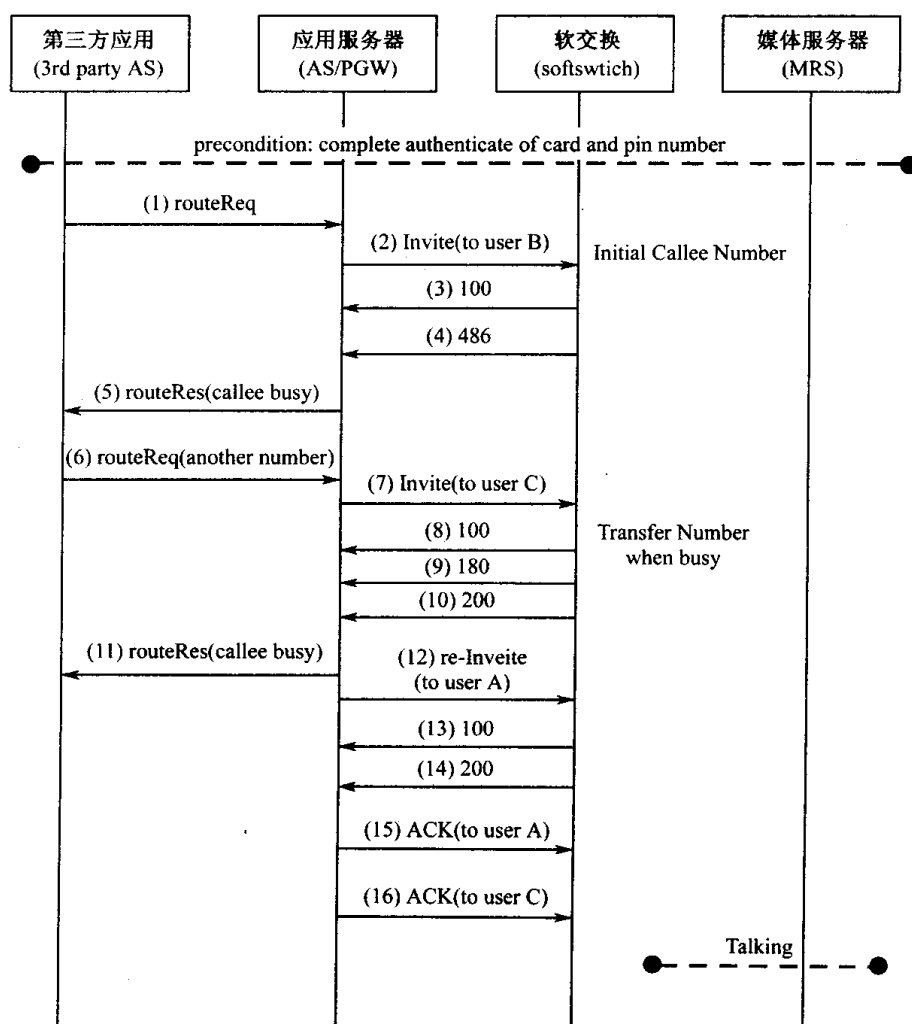


图 4-16 被叫忙流程

图中各个步骤的含义如下：

- 步骤 1)，第三方应用调用 routeReq，要求应用服务器建立到初始被叫用户 B 的连接；
- 步骤 2)，PGW 发送 Invite 到软交换，呼起被叫 B；
- 步骤 3)，软交换回送 100；
- 步骤 4)，用户 B 遇忙，软交换发送 486 给 PGW；
- 步骤 5)，PGW 调用第三方应用的 routeRes 接口，将被叫遇忙事件通知应用；
- 步骤 6)，第三方应用根据业务逻辑，查找所配置的到遇忙转移号码，发送 routeReq 给 PGW；
- 步骤 7)，PGW 发送 Invite 到软交换，呼起遇忙转移的被叫 C；
- 步骤 8)，软交换回送 100；
- 步骤 9)，被叫振铃，软交换回送 180；
- 步骤 10)，被叫应答，软交换回送 200；
- 步骤 11)，PGW 通过 routeRes 接口通知第三方应用，被叫应答事件；
- 步骤 12)，PGW 发送 re-Invite 给主叫 A，将用户 C 的媒体信息发送给 A；
- 步骤 13)，软交换回送 100；
- 步骤 14)，软交换回送 200 OK；
- 步骤 15)，PGW 发送 ACK 给用户 C；
- 步骤 16)，PGW 发送 ACK 给用户 A，A 和 C 的通话建立。

### 4.3.2 利用 SIP 应用服务器来开展软交换业务

Presence 业务流程图如图 4-17 所示。

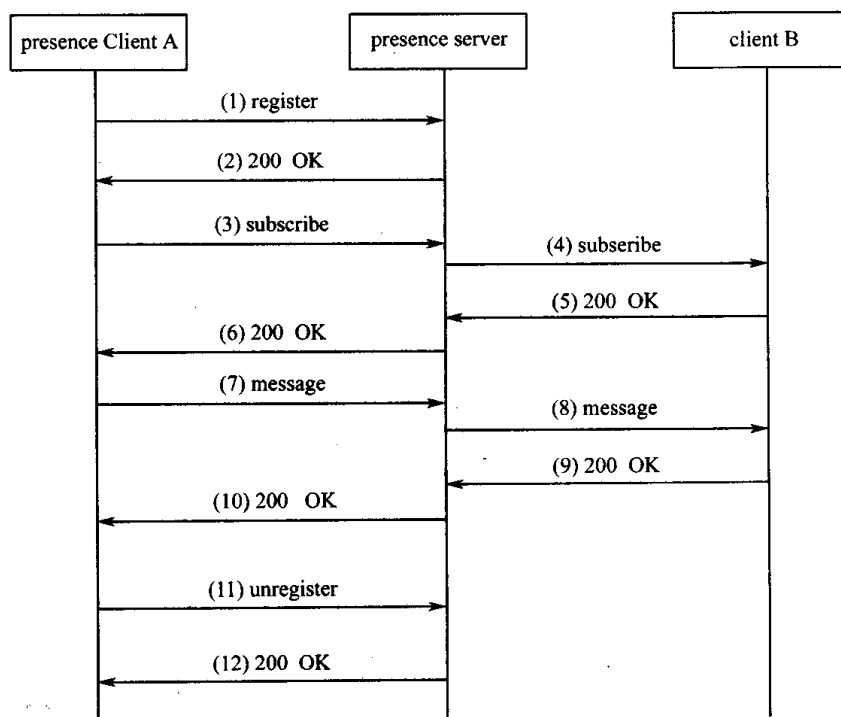


图 4-17 Presence 典型流程图

其功能如下：

### 1. 用户身份验证

用户使用 Messenger 软件登录系统时,需要输入自己的用户名和密码提交服务器进行用户身份验证。通过身份验证后,用户方可正常登录。

### 2. 设置用户状态

用户使用 Messenger 软件登录系统后可以随时改变自己的用户状态。用户状态改变后,系统将同步更新该用户所有订阅者的联系人状态列表,使该用户的订阅者能够随时了解到该用户的状态变化。

### 3. 订阅联系人

用户可以订阅其他用户,将被订阅者添加到自己的联系人列表中。订阅成功后订阅者和被订阅者可以互相看到对方状态,并可互发即时消息。

### 4. 收发即时消息

互为订阅的用户,可以随时发送/接收即时消息,若一方离线,则可发送离线消息。

### 5. 消息群发

用户可以同时向指定的多个用户群发即时消息。

### 6. 系统消息广播

系统管理员可以向系统中的所有用户发送广播消息。

### 7. 历史消息记录

系统将用户发送和接收过的即时消息自动保存,用户可以在需要时随时查看。

## 4.4 小结

本章首先介绍了软交换网络的基本知识,随后介绍了利用 SIP 协议进行软交换网络实体的互通和利用 SIP 协议开展软交换网络业务,重点分析了相关典型流程,如呼叫建立流程、被叫正常挂机流程、主叫正常挂机流程和被叫忙流程等。

## 4.5 习题

1. 简述软交换网络特点及其组成。
2. 简述 SIP 协议在软交换网络中的典型应用。
3. 简述利用 SIP 协议在软交换网络中,可以开展哪些业务。



## 第5章 移动领域中的 SIP 协议及其应用

### 知识点:

- 未来移动网发展趋势
- IMS 介绍
- SIP 在 IMS 中的应用
- SIPIMS 业务

未来移动网络将是分组化、多媒体化的 IMS 系统,本章将着重介绍 IMS 系统框架、SIP 协议在 IMS 系统中的应用及其利用 SIP 开展的 IMS 业务。

### 5.1 未来移动网发展趋势

#### 5.1.1 3GPP

3GPP 组织制定了 WCDMA 相关标准。

WCDMA 的网络演进分成 R99、R4、R5 等阶段。由兼容传统的 GSM、GPRS 网络逐渐向 ALL IP 的网络架构进行演进,最终将形成 ALL IP 的网络架构。

WCDMA 的 R99 版本于 1999 年 4 月形成第一个版本,2000 年 3 月全部完成。R99 的核心网继承了 GSM 以及 GPRS 核心网的网络特征,空中接口采用了 WCDMA 技术,在 RAN 与 CN 之间使用 ATM 承载方式。

3GPP R99 版引入了一套新的空中接口标准,运用了新的无线接口技术即 WCDMA 技术,引入了适于分组数据传输的协议和机制,数据速率可支持 144kbps、384kbps、2Mbps。R99 的核心网仍以演进的 GSM 核心网为基础。在业务应用上,3GPP 的标准中为业务的开发提供了三种机制,其一是针对 IP 业务的 CAMEL 功能;其二是开放业务结构(简称 OSA);其三是会话起始协议(简称 SIP),并在不同的版本中对其作了相应的不同程度的定义。R99 对 GSM 中的业务有了进一步的增强。从传输速率、频率利用率和系统容量上都有了很大提高。在业务方面,除了支持基本的电信业务和承载业务外,也可支持所有的补充业务。另外还可支持基于定位的业务(LCS)、号码可携性业务(MNP),支持一个用户在同一个连接上同时进行多个业务,支持 64kbp/s 的电路数据承载、CS 域的多媒体业务等。R99 采用开放式业务结构。

R4 于 2001 年 3 月完成定稿版本。R4 最大的变化是将 MSC 拆分成 MSC Server 和 MGW 两个网元,实现了呼叫控制与承载的分离,开始向 ALL IP 的网络架构演进。

R4 在业务方面对 R99 又做了进一步的增强。可支持 CS 域的多媒体消息业务、紧急呼叫的增强/基于 CS 域的呼叫、无 callback 功能 MexE 的增强以及 3 种实时传真业务。R4 允许运营商完全或根据要求在分组数据协议建立阶段来闭锁用户接入。在 R4 无线网络技术规范中没有网络结构的改变,增加的只是一些接口协议的增强功能和特性,主要特性包括低码片速率 TDD、UTRA FDD 直放站、TDD NodeB 同步、对 Iub 和 Iur 上的 AAL2 连接的 QoS 优化、Iu

上 RAB (无线接入承载) 的 QoS 协商、Iur 和 Iub 的 RRM (无线资源管理) 的优化、Iub、Iur 和 Iu 上的传输承载的修改过程、WCDMA1800/1900 和在软切换中的 DSCH 功率控制的改进。R4 在核心网上的主要特性为电路域 (CS 域) 的呼叫与承载的分离。核心网内的 No.7 信令传输第三阶段 (Stage3) 可支持 No.7 信令, 在 2 个核心网络功能实体间以基于不同的网络方式来传输, 包括基于 MTP、IP 和 ATM 的网络传输。

R5 阶段引进了 IMS 域, 增加了 CSCF, MGCF, BGCF 等网络实体, IP 承载成为核心承载方式, 形成了无线接入网络和核心网 ALL IP 的网络架构。

3GPP R5 最重要的工作是完成 IP 多媒体子系统的定义, 诸如路由选取以及多媒体会话的主要部分。随着 R5 的完成, 将为转向全 IP 网络的运营商提供一个可以开始建设的基础。在业务应用方面, R5 主要包括支持基于 IP 的多媒体业务即 IMS (包括确立 IMS 子系统的网络结构业务需求)、CAMELPhase4、全球文本电话 GlobalTextTelephony (GTT), 并实现了 OSA 的进一步改进。R5 还支持会话起始协议 (SIP), 支持扩展的透明端到端分组媒体流业务和宽带电话业务 (AMR 编码技术)。在无线部分的主要特性有: UTRAN 中的 IP 传输、HSDPA (高速下行分组数据接入)、对 Iub/Iur 的无线资源管理的优化、进一步的 RAB (无线接入承载) 的增强功能、相同域内的不同 RAN 节点与多个 CN 节点的连接。在核心网方面完成了由 M3UA (SCCP-UserAdaptation) 来传输 No.7 信令、IMS (IP 多媒体子系统) 第一阶段的网络结构和协议的定义。

到了 3GPP R6 阶段, 网络架构方面已没有太大的变化, 主要是增加了一些新的功能特性, 以及对已有功能特性的增强。3GPP R6 计划推出以下功能。考虑到版本冻结时间, 一些功能可能推迟, 成为后续 R7 版本的任务。具体包括以下方面:

- 在新业务方面, R6 进一步提出了许多新的业务需求, 这些新业务包括: IMS 阶段 2 (包括 IMS 本地业务、IMS 分组管理、IMS 消息业务、IMS 会议业务)、PUSH 业务、增强 MMS 能力、数字版权管理、WLAN-UMTS 互通、语音识别和语音驱动业务、用户认证、用户优先级业务、统一用户信息数据管理 (GenericUserProfile)、蜂窝通信系统上推出对讲机功能 (PushToTalkoverCellular)、网络共享功能等。在原有的业务生成和原有业务性能的增强方面, R6 同样提供有增强的功能, 主要是 OSA 的增强, 包括用户数据的管理/安全性管理、接入用户资料、提取被访网络的特性、接入 IPSESSION 信息、多媒体消息功能、增强 LCS 用户的私密性、Presence 信息管理、提取终端信息等能力。
- 在增强网络和空中接口的特性、优化无线资源的能力等方面, 包括基站分类、多输入多输出 (MIMO) 技术、改进 FDDUE 接收特性、波束赋型的增强、为在 RAN 里支持 IMS 所提供的 Iu 增强功能、跨 RNS 和 RNS/BSS 的无线资源管理的改进、远端天线电子倾角控制等。
- 在核心网方面, R6 将主要进行原有 R5 计划完成而未完成的内容, 如 IMS 与其他 IP 网络的互操作性、IMS 与 CS 网络的互操作性、Mm 接口 (CSCF 到外部的 IP 多媒体网络)、Mg 接口 (BGCF 到 MGCF 与 CS 互操作性)、Mn 接口 (IM-MGW 到 MGCF) 的增强、Mp 接口 (MRFC-MRFP) 协议的定义。
- 在网管方面, R6 也提出了不少工作内容, 包括性能管理、用户设备管理、网络设施管理、跟踪管理、收费管理。
- 在终端方面有新业务的引入、MexE, 有关 (U)SIM 工具箱的增强内容包括对 JAVA API

的测试规范、C 语言嵌入 USIM 应用程序接口 (API)、与 WLAN 互通有关的 USIM 方面的支持等。

### 5.1.2 3GPP2

CDMA2000 相关标准由 3GPP2 标准组织负责制定。

基于 CDMA2000 的移动软交换架构是基于 3GPP2 TSG-S 工作组制定的 ALL IP NAM (IP Network Architecture Model for cdma2000 Spread Spectrum Systems) 标准 S.R0037。该标准早在 2000 年就开始讨论,目前的最新版本是 2003 年 8 月 21 日发布的 Rev3.0 版本。

关于从现有网络向 ALLIP 演进的 roadmap, 3GPP2 还发布了一个演进标准 (Evolution Document) S.R0038。目前的最新标准是 2004 年 2 月 13 日发布的 Rev. 2.0 版本。在这个演进标准中定义了 ALLIP 网络的演进阶段。

ALLIP 网络的演进共分为 4 个阶段。

- Phase 0: 它是向 ALLIP 网络演进的起点。核心网标准为 ANSI41-D, 接入网标准为 IOS4.0, 空中接口标准为 cdma2000 Release 0, 分组网络标准为 P.S0001-A。
- Phase 1: 它是向 ALLIP 网络演进过程中的增强型网络。分组网络能力扩大。核心网标准为 ANSI41-D, 接入网标准为 IOS4.1, 空中接口标准为 cdma2000 Release 0 或 cdma2000 Release A, 分组网络标准为 P.S0001-B。接入网和分组网络信令和承载开始分离, 信令用 IP 进行传输。
- Phase 2: 它是向 ALLIP 网络演进的第一步。信令和承载开始独立演变并采用 IP 进行传输, 核心网和接入网也开始分离。这个阶段引入了传统 MS 域 LMSD (Legacy MS Domain) 在 IP 核心网中支持传统的终端, 以及多媒体域 IMS (IP Multimedia Subsystem) 的一些实体。空中接口仍旧采用 cdma2000 系列标准。
- Phase 3: 是 ALLIP 网络的最终目标。空中接口将 IP 化。LMSD 域将逐渐消失, 最终由 IMS 完全取代。

## 5.2 IMS 介绍

### 5.2.1 IP 多媒体子系统概念

IP 多媒体核心网系统是由所有能提供多媒体服务的核心网功能实体组成, 包括了与信令和承载相关的功能实体的集合。IP 多媒体业务是基于 IETF 定义的会话控制能力, 利用 PS 域和多媒体承载来实现的。

为了实现接入的独立性和支持无线终端与 Internet 互操作的平滑性, IP 多媒体子系统尽量采用与 IETF 一致的 Internet 标准。因此, 定义的接口跟 IETF 的 Internet 标准也是尽可能的一致, 如采用了 IETF 的 SIP 协议。

IP 多媒体核心网子系统使 PLMN 的运营商能给他们的用户提供基于 Internet 的应用、服务和协议的多媒体业务。这里并不是要把这些业务变成 IP 多媒体子系统的标准, 而是为了让 PLMN 的运营商和第三方的业务提供者来发展这些业务。IP 多媒体核心网子系统能集中语音、图像、消息、数据和基于 Web 的技术来为无线用户服务, 并把 Internet 的发展和无线通信的发展结合起来。

支持 IP 多媒体应用的全套解决方案是由终端、GERAN 或 UTRAN 无线接入网、GPRS 核心网和 IP 多媒体核心网子系统的一些特殊的功能单元。这些功能单元包括呼叫会话控制功能 (CSCF)、媒体网关控制功能 (MGCP)、IP 多媒体网关功能 (IM-MGW)、多媒体资源功能控制器 (MRFC)、多媒体资源功能处理器 (MRFP)、签约定位功能 (SLF)、出口网关控制功能 (BGCF)、应用服务器 (AS)、信令网关功能 (SGW)。

所有的功能实体被认为在不同的逻辑结点中实现,如果在同一个物理设备中实现两个逻辑结点的功能,那么这两个逻辑结点的接口就成为该设备的内部接口。

### 5.2.2 IP 多媒体核心网子系统实体

IP 多媒体子系统像 CS 域、PS 域子系统一样,可以完成呼叫的发起、保持、释放等功能。另外,它还要对多媒体流进行转换控制以及对多媒体业务的支持,所以包含更多的功能实体,它们分别完成不同的功能。

#### 1. 呼叫会话控制功能 CSCF (Call Session Control Function)

根据 CSCF 在网络中所处的位置不同,它所承担的作用也不一样,它可以分为如下三种类型:

##### 1) 代理 CSCF (P-CSCF)

在移动终端获得 IMS 服务时,代理 CSCF 是第一个联系节点。用户设备通过一个“本地 CSCF 发现流程”来得到 P-CSCF 的地址。P-CSCF 的作用就像一个代理服务器,它把收到的请求和服务进行处理或转发。P-CSCF 不会对 SIP 的 INVITE 消息中请求的 URI 进行修改。P-CSCF 也可以看做是一个用户代理,在异常情况下,它可以中止或独立地产生 SIP 的事务。

P-CSCF 的主要功能有:

- 把 UE 发来的 SIP 注册请求消息前转给 I-CSCF,该 I-CSCF 由 UE 提供的域名决定。
- 把 UE 发来的 SIP 消息前转给 SIP 服务器,该服务器的名字由 P-CSCF 在该 UE 发起注册规程时得到。
- 生成计费记录。
- 支持 UE 与 P-CSCF 间的加密过程。
- 对 SIP 消息进行压缩与解压缩。
- 承载资源的鉴权和 QoS 管理。

##### 2) 查询 CSCF (I-CSCF)

查询 CSCF 既可以充当网络所有用户的连接点,也可以用做当前网络服务区内漫游用户的服务接入点。在一个网络中可以有多个 I-CSCF。它的主要功能是:

- 注册。为用户指定一个 S-CSCF 来执行 SIP 注册。
- 处理会话相关与会话不相关的消息流。包括:将另一个网络收到的 SIP 请求路由到 S-CSCF;从 HSS 获得 S-CSCF 的地址;转发 SIP 请求或响应给 S-CSCF。
- 生成计费记录。

##### 3) 服务 CSCF (S-CSCF)

服务 CSCF 执行会话控制功能。它可以根据网络运营商的需要,维持会话状态信息。在同一个运营商的网络中,不同的 S-CSCF 可以有不同的功能。但在一个呼叫过程中,S-CSCF 要

完成的功能有:

- 注册。它可以作为一个注册机,也就是接收注册请求后通过位置服务器来使该请求的信息生效。
- 处理会话相关与会话不相关的消息流。包括:为已经注册的会话终端进行会话控制;可以作为一个代理服务器,也就是接收请求后,进行内部处理或是把它转发;可以作为一个用户代理,也就是它可以中断或是独立发起 SIP 事务;与服务平台交互来向用户提供服务;提供终端相关的服务信息。
- 当代表主叫的终端时,根据被叫的名字(如电话号码或 SIP URL)从数据库中获得为该被叫用户提供服务的网络的 I-CSCF 的地址。如果被叫在另一个网络,就把 SIP 请求或响应前转给该 I-CSCF;如果被叫与主叫在同一个网络,就把 SIP 请求或响应前转给该网络中的 I-CSCF。根据运营策略,把 SIP 请求或响应前转给 IP 多媒体核心网子系统外的 ISP 的 SIP 服务器。当呼叫要路由到 PSTN(或 CS 域)时,就把 SIP 请求或响应转发给 BGCF。
- 当代表被叫的终端时,如果用户在归属网络中,就把 SIP 请求或响应前转给 P-CSCF;如果用户在拜访网络中,就把 SIP 请求或响应前转给 I-CSCF。根据 HSS 和业务控制功能的交互作用,把要路由到 CS 域的入局呼叫的 SIP 请求进行修改。当呼叫要路由到 PSTN(或 CS 域)时,就把 SIP 请求或响应转发给 BGCF。
- 生成计费记录。

## 2. 媒体网关控制功能 MGCP (Media Gateway Control Function)

媒体网关控制功能 MGCP 包括如下功能:

- 控制 IMS-MGW 中的媒体信道的连接。
- 与 CSCF 通信。
- 根据路由号码,为从传统网络来的入局呼叫选择 CSCF。
- 执行 ISUP 协议和 IMS 呼叫控制协议间的转换。

## 3. IP 多媒体网关功能 IM-MGW (IP Multimedia-Media Gateway Function)

一个 IM-MGW 可以终止来自电路交换网的承载信道和来自分组网的媒体流(例如,IP 网中的 RTP 流)。IM-MGW 可以支持媒体转换、承载控制和负荷处理(例如,多媒体数字信号编解码器、回声消除器、会议桥)。它包含如下功能:

- 与 MGCF 交互来进行资源控制。
- 拥有并维护回声消除器等资源。
- 可能需要多媒体数字信号编、解码器。

IMS-MGW 要提供必要的资源来支持 UMTS/ GSM 的媒体传输。还需要对 H.248 协议进行进一步的调整来支持额外的多媒体数字信号编、解码器等。

## 4. 多媒体资源功能 MRF (Multimedia Resource Function)

多媒体资源功能分成两部分,包括多媒体资源功能控制器 MRFC (Multimedia Resource Function Controller)和多媒体资源功能处理器 MRFP (Multimedia Resource Function Processor)。

- 多媒体资源功能控制器 MRFC 的主要功能如下:控制在 MRFP 中的媒体流资源;翻译来自 AS 和 S-CSCF 的信息(例如,会话标识符),并相应地对 MRFP 进行控制;产生

计费记录。

- 多媒体资源功能处理器 MRFP 的主要功能如下：控制 Mb 接口点的承载；提供 MRFC 需要的资源；混合输入的媒体流，如用于多方会议；发出多媒体流，如用于多媒体广播；处理多媒体流，如语音编码转换、媒体分析。

#### 5. 签约定位功能 SLF (Subscription Locator Function)

签约定位功能 SLF 的主要功能如下：在注册和会话建立期间，被 I-CSCF 查询，SLF 向 I-CSCF 提供存储用户具体数据的 HSS 的名字；通过 Dx 接口来接入 IMS。在单一的 HSS 环境中，并不需要 SLF。

#### 6. 出口网关控制功能 BGCF (Breakout Gateway Control Function)

出口网关控制功能 BGCF 用来选择与 PSTN (或 CS 域) 接口点相连的网络。如果 BGCF 发现自己所在的网络与接口点相连，那么 BGCF 就选择一个 MGCF，该 MGCF 负责与 PSTN (或 CS 域) 的交互。如果接口点在另一个网络，那么 BGCF 就把会话信令转发给另一个网络的 BGCF。BGCF 在选择与 PSTN 相连的网络的时候，会利用收到的其他协议的信息和管理信息。BGCF 的主要功能如下：

- 收到 S-CSCF 的请求后，为呼叫选择一个适当的 PSTN (或 CS 域) 接口点。
- 选择一个与 PSTN (或 CS 域) 相连的网络。如果本网络没有与 PSTN 相连，那么 BGCF 就把 SIP 信令转发给与 PSTN (或 CS 域) 相连的网络的 BGCF。
- 在与 PSTN (或 CS 域) 相连的网络中，选择一个 MGCF，并把 SIP 信令前转给 MGCF。
- 生成计费记录。

#### 7. 信令网关功能 SGW (Signalling Gateway Function)

信令网关 SGW 完成传输层的信令转换，把基于 SS7 的信令与基于 IP 的信令进行转换 (也就是在 Sigtran SCTP/IP 和 SS7 MTP 间进行转换)。SGW 不对应用层的消息进行解释，但必须对底层的 SCCP 或 SCTP 消息进行解释来保证信令的正确路由。

### 5.2.3 3GPP/3GPP2 异同及基于 IMS 的融合

#### 1. 异同

3GPP 与 3GPP2 IMS 存在以下的异同：

- 智能卡：3GPP IMS 终端支持 UICC 卡和 ISIM 应用，而 3GPP2 暂不支持；
- 临时的公共/私有标识符：3GPP 采用 E.212，而 3GPP2 为基于 MIN 的 IMSI；
- IPv4/IPv6：3GPP IMS 只支持 IPv6，而 3GPP2 IMS 既支持 IPv6，又支持 IPv4；
- 网络安全：接入安全 (Gm 接口) 相同，但安全网关 (SEG) 在 3GPP2 里尚不支持；
- HSS/AAA：在 3GPP2 IMS 中，HSS 只在 PS 域中采用，其余采用 AAA 和数据库，而 3GPP 中 PS、CS 域都采用 HSS；
- SLF、Dx 接口：在 3GPP2 IMS 中尚未支持；
- 默认 Codec：3GPP2 IMS 没有定义默认 Codec；
- 策略及 PDF：3GPP2 中尚处于阶段 1；
- P-CSCF&PDSN/GGSN 位置：在 3GPP 中，GGSN 和 P-CSCF 位于同一网络中，而在

3GPP2 中, PDSN 和 P-CSCF 可位于不同网络中, 如 PDSN 处于访问网络中, 而 P-CSCF 处于归属网络中;

- GGSN 锚点: 在 3GPP 中, 对于给定的会话, GGSN 锚点确定; 而在 3GPP2 中, PDSN 可改变;
- 计费: 3GPP2 IMS 使用 AAA 来计费, 代替了 CCF, 使用 AAA 传递计费信息, 而非通过计费系统;
- 位置信息交换: 3GPP R5 阶段位置信息通过 Sh 接口来交换;
- IMS 对象命名: 3GPP 中采用与接入网无关的命名方式。

## 2. IMS 融合

如图 5-1 所示, IMS 融合包括以下方面的统一。

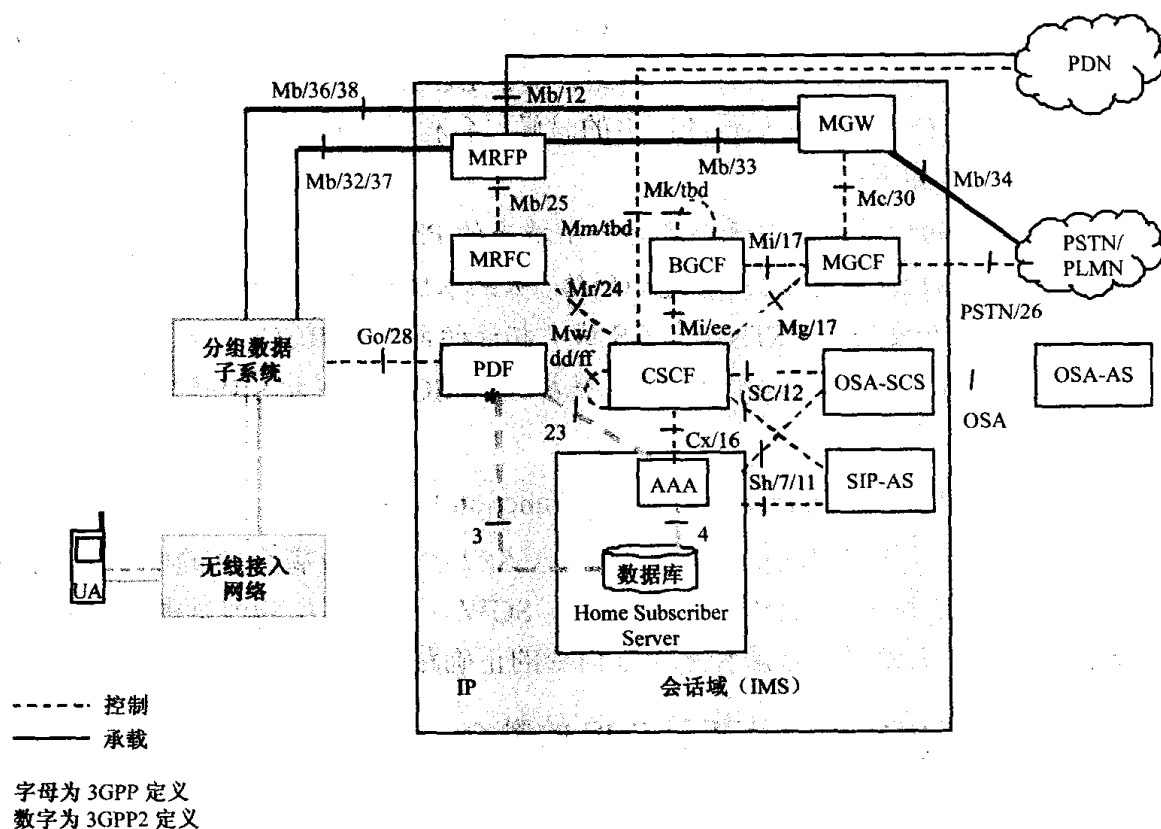


图 5-1 IMS 统一图

### ◆ 3GPP/3GPP2 参考模型术语的统一

3GPP2 与 3GPP 术语的对应关系如下:

- MMD Subset -> IMS (IP Core Network Multi-media Session Domain);
- CQM, PCF -> PDF (Policy Decision Function);
- X-SCM -> X-CSCF (Call Session Control Function, X = P, I, S );
- L-SCM -> BGCF (Breakout Gateway Control Function);
- NCGW -> OSA-SCS (OSA Service Capability Server)。

### ◆ 功能实体和接口的统一

3GPP/3GPP2 应当对公共实体采用相同的功能; 3GPP/3GPP2 应当对公共接口采用相同的过程和协议。

### 5.2.4 IMS 进展

3GPP-IMS 当前的进展情况和热点讨论中的问题主要有: IMS phase 2、网络融合和 IMS commonality and interoperability、IP flow based bearer charging evolution 和 QoS。这里着重介绍 IP Flow Based Charging 技术。

#### 1. IMS stage 2

IMS stage 2 主要研究 IP 多媒体核心网子系统业务实现规程及相关 SIP 会话呼叫模型, 目前 3GPP 对 IMS stage 2 的研究参见 R6 规范“3GPP TS 23.228 V6.4.1 (2004-01) TSG-Services&System IP Multimedia Subsystem (IMS) – Stage 2(Relase 6)”, 3GPP2 对 IMS 阶段 2 的研究参见“3GPP2 X.S0013-002-0 IP Multimedia Subsystem – Stage 2”, 3GPP2 的 IMS 相关标准参照了 3GPP 的标准, 其标准的成熟度要比 3GPP 弱。最新的 3GPP IMS stage2 标准相对于 3GPP2 来说, 主要区别在于:

- 增加了 IMS 群组管理的概念;
- 提供对增强型的 SIP 能力的支持;
- 增强了 QoS 功能及基于业务的本地策略处理功能;
- 会话刷新功能;
- 增强了与 IPv4 SIP 网络互连的考虑;
- 增加了 PSI (Public Service Identities) 配置和路由原则;
- 增加了与外部 SIP client 互连时的情形处理;
- 增加了对路由信息的询问处理规程;
- 增加了基于会话的消息处理。

在 IMS 阶段 2 标准中, 首先介绍了一些与 IMS 相关的基本概念, 例如, 与 CS 域和 IP 连接网络间的关系、与 IMS 业务相关的概念、域名和编址概念、信令、与移动性相关的概念、CSCF 的功能、媒体资源功能处理器、IMS 安全概念、计费等, 着重介绍了与 IMS 相关的操作规程, 主要包括:

- CSCF 相关的操作规程;
- 应用级的注册及注销规程;
- IP 多媒体会话相关的规程;
- Serving-CSCF/MGCF 与 serving-CSCF/MGCF 之间的规程;
- 发起会话呼叫和终结会话呼叫的规程;
- 与路由信息查询相关的规程;
- 会话过程中的信令路由规程;
- 会话释放规程;
- 增强型的多媒体业务规程 (主要是一些常见的补充业务规程);
- 呼叫未注册用户或呼叫不存在的用户的规程;
- 多媒体紧急呼叫规程;
- 与 MRFC/MRFP 相关的操作规程;
- IMS 消息概念及操作过程;



- 不同 IP 版本的网络间互操作体系架构。

## 2. IMS 阶段 2

### 1) 3GPP IMS phase2 进展情况

#### ◆ R6 IMS Phase 2 内容

- Stages1(22.228) 和 Stages2 (23.228) 已完成。
- IMS 本地业务: Stage3 尚未开始。
- IMS 和 CS 网络的互通: Stage3 (29.163) 在 CN#21 获得通过。
- IMS 与非 IMS IP 网络的互通: CN1 Stage3 (SIP/SDP); CN3 SIP/SDP 已完成; CN3 IPv4/v6 互通方面已完成。
- Mn 接口 (IM-MGW - MGCF) 增强: TS 29.332 Stage3 (CN4), TS 29.163 Stage3 (CN3) 已完成。
- Mp 接口 (MRFC - MRFP) 协议定义: TS 29.333 Stage3 (CN4)已完成。
- R6 结构的合法监听, 正在制定。

#### ◆ 其他方面

- Gq 接口 (PDF - P-CSCF): 接口参见 QoS, 尚未完成。
- 基于 IPv4 的 IMS: “Interworking aspects and migration scenarios for IPv4-based IMS Implementations” 工作组在 2003 年 9 月成立, 相应 TR 在 2004 年 6 月被提交, 任何 Stage3 工作等待可行性研究的完成。
- Cx 和 Sh 接口增强: 定义公共标识和公共标识的共享的 CR 已经通过, 3GPP 新规范中将采用基于 Diameter 的接口。
- IMS 组管理: Stage2 已完成, Stage3 目前尚未完成。
- IMS Conferencing 尚未完成。
- IMS Messaging 尚未完成。
- 其他 SIP 能力支持: 包含各种 SIP 小的增强, 主要功能包括对于 IETF 通过的 SIP 新 RFC, 描述 3GPP 如何支持这些扩展。
- 新工作组 (WID): PS 会话多媒体应用的编、解码增强 (SA4)。
- IMS 计费, 完全依赖 IETF Diameter (参见计费)。

注意: IMS Phase2 的 Stage3 密切依赖于 IETF 的进展。

#### ◆ 其他与 IMS 相关的研究课题

- PS domain and IMS impacts for supporting IMS Emergency calls: 需求属于 TS 22.228, 已完成; SA2 预计完成时间在 2004 年 11 月, 可能推迟到 R7; Stage3 预计完成时间在 2005 年 6 月; 依赖于 IETF, T3, RAN3 的参与进度。
- Interoperability and commonality between IMS using different IP connectivity networks (IMSCOOP): 3GPP 关于 commonality 和 Interoperability 的工作在 2003 年 12 月已关闭, 期待 3GPP2 在 Interoperability 方面做一些研究。
- Push Services: TS 22.174 Stage 1 已稳定; TR 23.976 Stage 2 已完成 95 %, 只有 NRPCA 尚未确定; Stage 3 尚未开始, 目前没有确定是否需要 Stage 2 TS 和 Stage 3, 取决于 NRPCA。

- Presence: TS 22.141 Stage 1 和 TS 23.141 Stage 2 早已完成, SA4 编、解码和格式没有进展, SA4 TS 26.141 预计 2004 年 9 月份完成; CN1 将产生新 TS, 预计 2004 年 9 月完成; CN5 OSA 映射尚未开始工作; CN4 工作已完成; CN3 可能也需要参与, 但尚未开始。
- QoS 增强: 端到端 QoS 动态策略控制增强的可行性研究: TS 23.207 中增加了 Gq 接口的描述, Stage 3 (CN3) 的 TS 29.209 (Gq 接口的策略控制) 即将完成。

#### ◆ 标准进展情况

标准进展情况如表 5-1 所示。

表 5-1 标准进展情况

任务名称	开始时间	计划完成时间	现完成情况
<b>IMS2 整体情况</b>	28-08-2000	08-09-2004	55%
<b>第二阶段</b>	02-09-2002	19-09-2003	80%
<b>Cx、Sh 接口增强</b>	06-06-2003	31-08-2004	53%
<b>IMS 群组管理</b>	14-03-2002	08-09-2004	55%
第一阶段	14-03-2002	09-12-2002	100%
第二阶段	26-05-2003	31-12-2003	100%
第三阶段	13-12-2002	08-09-2004	20%
<b>IMS 会议电话</b>	04-11-2002	08-09-2004	85%
第二阶段	04-11-2002	31-12-2003	100%
第三阶段	13-12-2002	08-09-2004	75%
<b>IMS 消息</b>	14-03-2002	08-09-2004	73%
技术建议	14-03-2002	09-12-2002	100%
第一阶段	11-11-2002	11-12-2002	100%
<b>CRs to 22.140 &amp; 22.228</b>	14-03-2002	17-03-2003	100%
第二阶段	04-11-2002	23-04-2004	90%
第三阶段	13-12-2002	08-09-2004	30%
<b>IMS 本地业务</b>	01-01-2001	04-06-2004	40%
第二阶段	01-01-2001	29-03-2002	100%
第三阶段	13-12-2002	04-06-2004	0%
<b>SIP 能力增强</b>	11-11-2002	08-09-2004	65%
第二阶段	11-11-2002	20-02-2004	100%
第三阶段	13-12-2002	08-09-2004	40%
<b>能力回顾</b>	13-12-2002	12-03-2004	20%
<b>IMS 网络与分组网的互通</b>	28-08-2000	08-09-2004	42%
<b>3GPP_SIP 与 IETF_SIP 的互通</b>	28-08-2000	13-06-2003	100%
<b>Ipv6 与 Ipv4 的互通</b>	28-08-2000	12-12-2003	20%
<b>IPv6 与 Ipv4 的互通(SIP / SDP 方面)</b>	11-05-2004	08-09-2004	0%
<b>与非 IMS 的 IP 网络互通</b>	14-03-2001	04-06-2004	40%
<b>IMS 网络与电路网络的互通</b>	28-08-2000	18-03-2004	78%
<b>Mn 接口增强</b>	07-08-2001	31-08-2004	64%
<b>Mp 接口</b>	13-12-2002	31-08-2004	0%
<b>用户与运营商之间的关系以及对 ISIM 的需求</b>	15-11-2002	12-12-2002	100%
<b>R6 网络架构下的合法监听</b>	09-12-2002	18-12-2003	10%
<b>IMS 签约及接入场景</b>	16-12-2002	13-06-2003	100%

## 2) 3GPP2 IMS phase2 进展情况

3GPP2 在 2004 年 1 月决定以 MMD Rev 0 为蓝本, 采纳 3GPP R6 IMS2 所有的 CR, 起草 MMD Rev A 版本, 而非等待 3GPP R6 发布以后再进行修改。在修改过程中, 只有存在异议的情况下才对该 CR 进行技术讨论。

Rev A 版本在 Rev 0 版本的基础上主要增加 3GPP 新增特性、在线业务。Rev A 版本中增加两个部分, 其中 12 部分为在线业务安全性规范, 13 部分为在线业务规范。

## 3. 计费

### 1) 介绍

R6 前的分组域计费是基于时长、流量或者 PDP (APN 和 QoS), 随着移动互联网的应用正不断增多, 基于时长、流量和 PDP 的计费已不够用, 需要机制区分用户数据中的业务流, 并配以相关的收费。

IP Flow Based Charging 是通过 IP 过滤器来区分在用户上下行数据中的业务流, 过滤器规则由运营商定义。过滤规则一般基于 5 元组 (源 IP 地址、目的 IP 地址、源端口号、目的端口号、协议 ID)、深层过滤器 (对于用户 IP 包进行更深层的分析并能识别用户业务的状态), 在同一 PDP 中可同时实施多个业务流过滤器。

### 2) 网络架构

离线架构如图 5-2 所示, 在线架构如图 5-3 所示。

### 3) IP Flow Based Charging 对于网络结构的影响及存在问题

在 3GPP, 计费点采集的实施点正从 SGSN 改为 GGSN, 由拜访地改为归属地。与 GPRS 计费的关系 (如 identification of nodes、relation to GPRS online charging、Charging key), 在 OCS, CRF 和 TPF 中分别属于拜访地或归属地网络时, 对于网络结构的影响、以及与 IMS 系统的关系、与 WLAN 系统的关系还有待进一步研究。

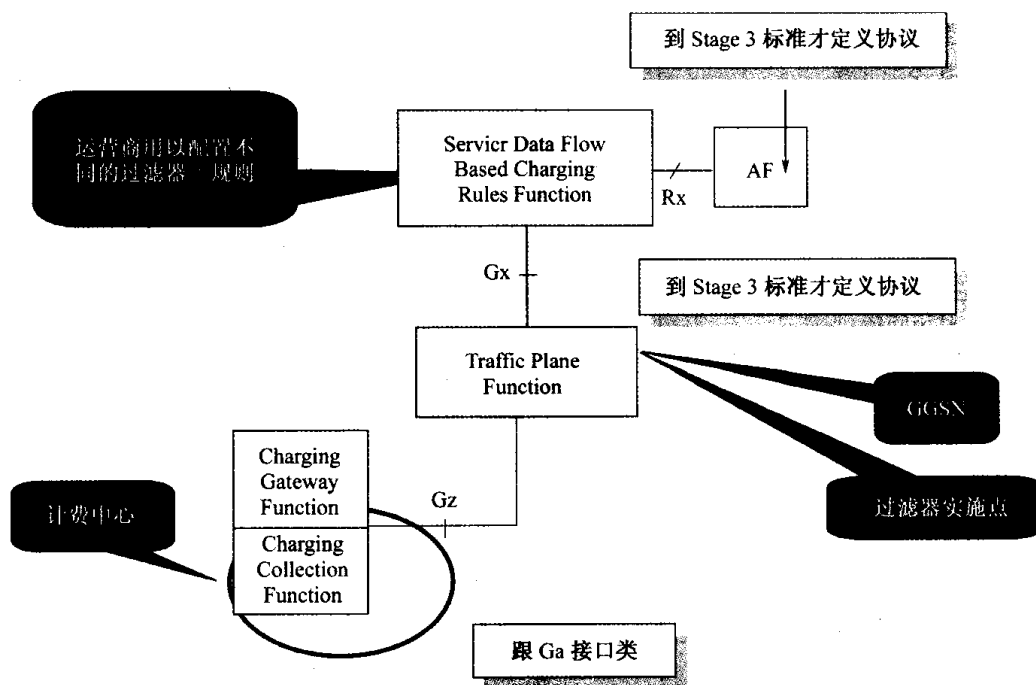


图 5-2 离线架构计费图

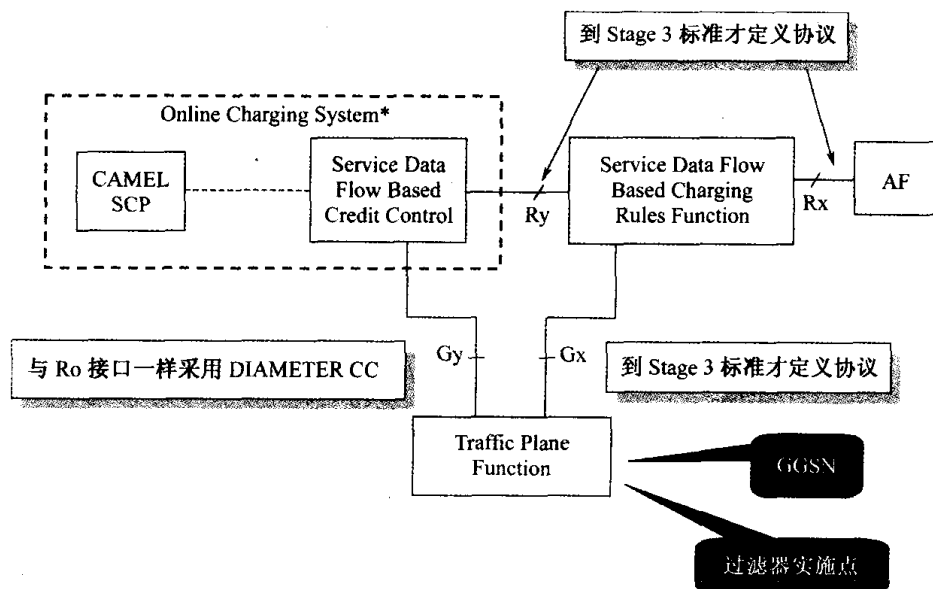


图 5-3 在线架构计费图

## 5.3 IMS 中的 SIP 协议

### 5.3.1 简介

由于 SIP 的增强方法很灵活，使其可以较容易地实施。当 3GPP 在 Release 5 中设计 IMS 时，决定用 SIP 作为会话控制协议。公用移动网有特定的需求，如低带宽、漫游、安全需求、QoS 和计费管制问题。这些对信令协议都会有特定要求。

为满足这些需求，3GPP 在 TS24.229 中定义了使用 SIP 和 SDP 的概要。概要中包括了 IMS 使用到的 IETF 规范，也指出了在 3GPP 系统中 SIP 信令和其他协议的交互。这意味着 3GPP 没有定义一个新的 SIP (有时候说成 3GPP SIP)，只是 3GPP 以某种方式使用 IETF 定义的 SIP。

然而，3GPP 有时需要特定的 SIP 协议增强。为此 3GPP 通常提交一个需求文件给 IETF，如 3GPP R5 SIP 需求就在 draft-ietf-sipping-3gpp-r5-requirements-00 文件中描述。然后 IETF 提供一个合适的协议解决方案。

IETF 和 3GPP 的标准合作在 RFC3113 中有描述，在后面的 IMS 设计阶段，3GPP 定义了自己自己的网络实体的一些行为，这些行为并不遵循 SIP 原则：像大多数 IETF 协议一样，SIP 假设终端可完成大部分智能，如中间的网络实体仅有有限的权利去修改消息。所以从 IETF 得到了意见后，3GPP 花了主要精力修改了相应的规范。当 R5 完成时，IMS 使用 SIP 时会和 IETF 运用的 SIP 协调一致起来。

在 SIP 模型中，为建立起一个会话，用户代理客户端向用户代理服务器发起请求。请求通过代理服务器在网络中路由。另外，注册服务器，因为要提供用户代理的位置信息，因而需要将 SIP 地址映射成 IP 地址。

3GPP 的 IMS 体系结构选择了这个模型，描述参见 TS 23.002 和 TS 23.228。在 IMS 中的用户代理为用户设备 (UE，即手机)。IMS 中的代理服务器和注册服务器是指名为呼叫会话控制功能 (CSCF) 的网络实体。有三种类型的 CSCF：

- 服务 CSCF (S-CSCF)，作为注册服务器并且激活基于用户数据的应用业务控制；
- 代理 CSCF (P-CSCF)，是 UE 在 IMS 网络中的第一个接触点；SIP 信令消息在 P-CSCF

和 UE 之间传送;

- 查询 CSCF (I-CSCF) 对于外部网络是第一个接触点, 特别是对于外部的 IMS 网络。

IMS 使用“归属控制”, 即会话控制信令总是由位于归属网络的 S-CSCF 负责 (这与目前的 GSM 网络不同, GSM 网络是由拜访地的 MSC 提供业务)。P-CSCF 总是和 GGSN 在同一个网络中, 可能在归属网络, 也可能在拜访网络)。

### 5.3.2 IMS 中的 SIP 扩展

3GPP 在 IMS 中既不定义新的 SIP 消息也不定义私有的 SIP 包头, 而是使用在 RFC 3261 中定义的 SIP, 并且为某些 SIP 扩展需要给予特定的支持。最重要的扩展如下所示 (它们在 IETF RFC 中都有定义)。

#### 1. 压缩

因为无线接口是稀有资源, 因而 IMS 会话有效地使用带宽是需要的。因此, 对媒体流和信令消息进行压缩很必要。在 IMS 中, 对 SIP 信令的压缩 (SigComp) 是必须支持的。UE 和 P-CSCF 完成 SIP 消息的压缩和解压缩。

#### 2. 安全

在 IMS 中使用 AKA 完成对用户的鉴权。AKA 是 3GPP 的特定鉴权机制, 它基于存储在 ISIM 和网络中的共享密钥。AKA 参数会映射给 SIP 使用的 HTTP-Digest 验证。而且, IMS 需要对经过空中接口从 UE 传来的消息进行完整性检查。因此, UE 和 P-CSCF 需支持根据 IP 安全协议 (IPSec) 规定的完整性保护。不过 IPSec 加密目前在 IMS 中不需要使用。

#### 3. 从指定的 CSCF 路由

IMS 提供业务是由归属网络运营商的控制的, 即使对于漫游用户也一样。这样, 对于发起和中止会话, 必须保证会话信令传输: (a) 归属网络中的 S-CSCF, 负责触发业务; (b) P-CSCF, 是 IMS 中 UE 的第一个和最后一个接触点。SIP 完成此需求必须要有业务路由发现、路径头机制和松散路由功能。

#### 4. 私有包头

IMS 需要在 UE 与 CSCF 之间或 CSCF 与 CSCF 之间, 将一些移动网特定信息在消息中传输。例如, Cell-ID、拜访网络名称或计费标识。这些信息在所谓的私有包头中传送。

#### 5. Precondition

IMS 重视 UE 资源管理。当媒体通道的资源被建立起来防止了媒体受阻而达到了期望的 QoS 时, 这就需要确保通知 B 方。因此, 所有必需资源的可用性是建立会话的前提。实现的解决方案是基于 SDP 提供/回答机制以及相关 SIP 和 SDP Precondition 扩展。Precondition 扩展的使用导致了特定的 SIP 呼叫流程。IMS 通过使用位于 GGSN 和 P-CSCF 之间的 Go 接口, 完成对媒体资源的策略控制。

#### 6. 网络发起的呼叫释放

在移动网中, 有时需要网络释放一个正在进行的呼叫。例如, 无线覆盖的缺乏、预付费账户空或者管理原因等。从网络侧送出一个 BYE 请求给 UE 就可以解决这个问题。但是这不符合 SIP 原则, 那就是代理服务器不允许发 BYE 消息。但是, 由于缺乏更好的解决方法, 因而 IETF 接受了 3GPP 的需求和此解决方案。

对 SIP 的一些头部进行了参数扩展, 例如, 对 WWW-authenticate 头部进行了参数扩展, 定义了一个新的 auth-param 参数字段, 用在对 REGISTER 请求的 401 (Unauthorized) 响应中, 此字段又包括 integrity-key 和 cipher-key 两个具体参数等。

对 SIP 协议中的消息体 MIME 类型增加了 “application/3gpp-ims+xml” 类型, 即 3GPP IM CN subsystem XML body, version 1, 同时约定了此类型内容不允许发送到 3GPP 的网络以外。

同时针对移动网无线接口的特点, 对 SIP 的定时器的取值进行了调整。

例如, RFC3261 中对 T1 (RTT) 定时器默认设置为 500 ms, 而在移动网络中对于涉及无线接口的部分, 如 CSCF 到 UE 以及 UE 的处理上则将 T1 的默认值设置为 2 s; 而对于核心网中不涉及无线接口处理的各功能实体之间还保留其默认值为 500 ms。对于 T2、T4 定时器的取值, 也存在同样的处理。在网络功能实体之间的默认值仍然分别为 4s 和 5s, 而在 CSCF 到 UE 以及 UE 的处理上则分别取值为 16s 和 17s。

此外, 对于有些标准, 如 RFC 3320 “Signaling Compression (SigComp)”, 在移动网络中, 在 IP Multimedia Client 与 CSCF 功能实体之间被要求采用, 以便节省有限的无线带宽。

## 5.4 IMS 中的 SIP 典型流程

### 5.4.1 IP 多媒体子系统业务交互的过程

业务点触发器 (SPT) 是那些在 SIP 信令路径上可以设置过滤准则的一些规则, 有下面的几点可以定义: SIP 方法 (如 REGISTER, INVITE, SUBSCRIBE, MESSAGE)、任何存在的报文头部、报文头部的内容或 Request-URI、请求用户的方向 (如始呼叫、终呼叫、呼叫非注册用户)。

一个过滤准则可以触发一个或多个 SPT 以便把请求消息发送给特定的应用服务器。保存在用户的服务清单 profile 中的过滤准则的集合被称为应用服务器签约信息 (Subscription Information)。为了使 S-CSCF 能够有序地处理这些不同的过滤准则, 应给每个过滤准则定义一个优先级别, 若 S-CSCF 联系不上 AS, 则 S-CSCF 应用默认的处理方式。这些默认的处理方式包括: 规则列表中具有较低优先级的过滤条件是否匹配; 不管较低优先级的触发条件, 直接释放这个会话。

过滤准则应包含下面的信息:

- 应用服务器的地址;
- 过滤准则的优先级提供应用这些准则的顺序;
- SPTs 可以使用与、或、非等逻辑表达式来构建, 满足过滤准则的那些条件成为触发点 (Trigger Points);
- 默认的处理方式;
- 在把请求消息送给 AS 前, 可以选择在消息体中增加相关的业务信息。

对于一个用户来说, 不能指定相同的优先级给两个以上的过滤准则, 也就是说每个原始过滤准则只有一个优先级。S-CSCF 向 HSS 索要注册或未注册用户的相关的 iFC 集合。若 S-CSCF 认为目前它保存的 iFC 集合是有效的话, 可以不再向 HSS 索要新的 iFC 集合。

如果 S-CSCF 从 HSS 得到多个过滤准则时, S-CSCF 将对从 Mw 接口收到一个消息, 依据这些带有优先级的过滤准则进行检查:

- 对这个请求建立一个根据其优先级排序的过滤准则列表，该过滤准则的执行顺序直到该消息通过 Mw 接口离开 S-CSCF 时才有可能被改变；
- 分析收到的请求消息，以便发现包含在消息中的 Service Point Triggers(SPT)；
- 检查最高优先级的过滤准则中的触发点 (trigger points) 是否与该请求消息中 SPT 相匹配，若不匹配，S-CSCF 则继续转向步骤 4 处理。若匹配的话，(i) S-CSCF 对该消息增加一个其入呼叫分支能识别的标识，甚至会话标识也可能被改变，例如，AS 执行第三方呼叫控制；(ii) 把该消息通过 ISC 接口送给 AS，AS 则对该消息进行相关的业务逻辑处理，AS 可以修改该消息，还可能又把该消息又重新送给 S-CSCF；(iii) 若该请求消息从 AS 通过 ISC 接口重新收到的话，则转到步骤 4 进行相关处理；
- 重复上面的步骤 2 和步骤 3，对每一个在步骤 1 中建立的过滤准则进行检查，直到最后一个过滤准则被执行；
- 基于 SIP 通常的路由方式对该请求消息进行路由转发。

如果 AS 决定结束这个请求，则 AS 会通过 ISC 接口向 S-CSCF 发送一个对前面的请求消息的最终响应消息，注意，在这个最终的响应消息中应该包含在上面步骤中的 (i) 子步骤中由 S-CSCF 添加的标识，以便 S-CSCF 能识别该响应消息。S-CSCF 会停止在规则列表中有较低优先级的过滤准则的匹配过程。

被调用的 AS/业务逻辑可以决定不参与当前调用的会话，实现的方法是在最初开始的 SIP 事务 INVITE 请求中不增加 Record-Route/Route 记录就可以了，以后所有后续的与这次会话相关的 SIP 信令不会再被路由到该 AS 上了。原始的过滤准则不能够实现让一个 SIP 应用服务器决定不再接收后续的 SIP 消息即不再参与后续 SIP 事务处理。

在设计原始的过滤准则时，应该避免类似于从一个 AS 出来的请求又被送回到同一个 AS 中来的死循环，当然，若同一个 SIP 请求又被路由回来是处于业务逻辑的处理要求而作出的选择的话，那就另当别论了。

应用触发框架图如图 5-4 所示。

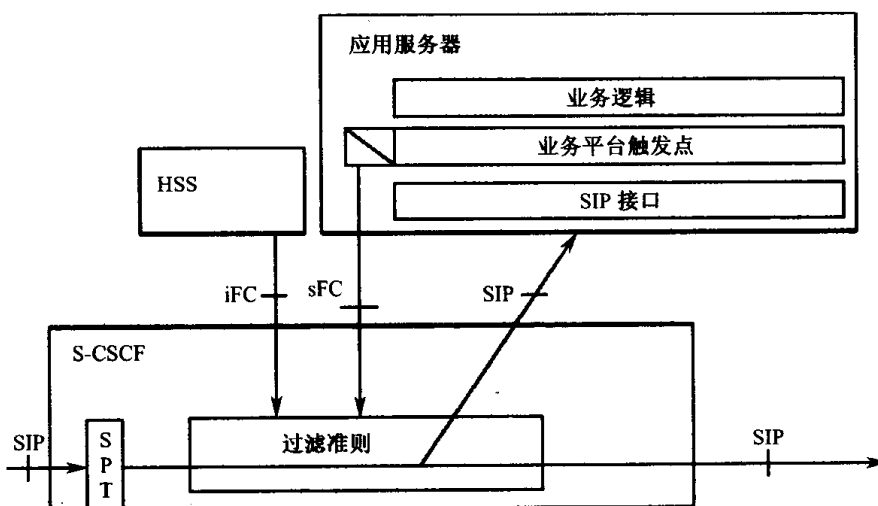


图 5-4 应用触发框架图

## 5.4.2 服务 CSCF 相关流程

服务 CSCF 相关流程包括：注册流程、订阅和通告流程、计费流程、呼叫发起流程、呼叫终结流程、释放流程等。其中注册流程如图 5-5 所示。

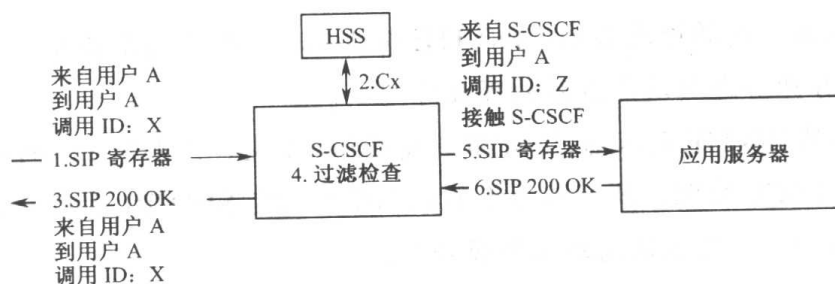


图 5-5 注册流程

### 5.4.3 应用服务器/MRFC 相关流程

这里以放音流程加以说明，放音流程如图 5-6 所示。

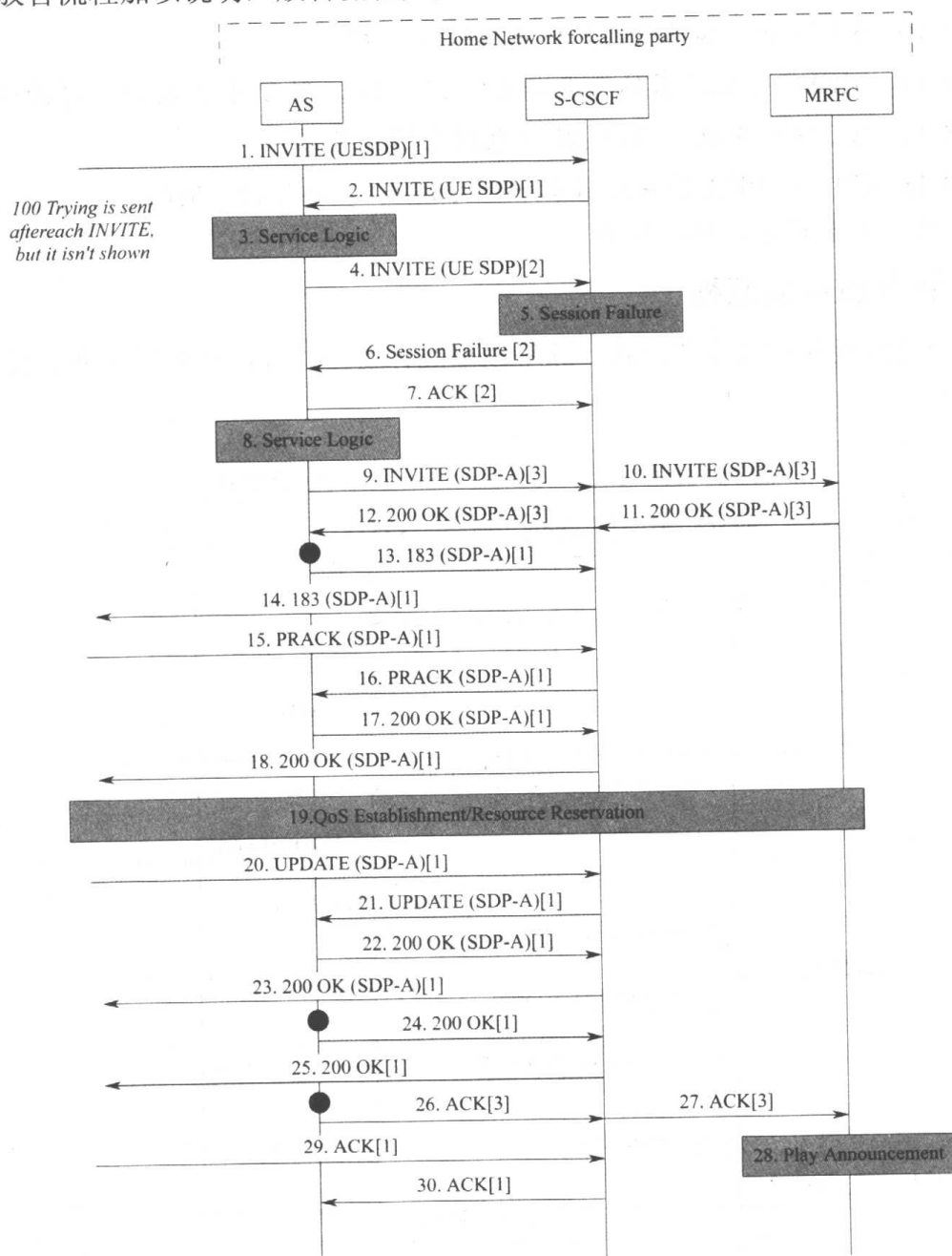


图 5-6 放音流程

在放音流程中各个步骤的含义如下：

- 步骤 1，S-CSCF 接收到 INVITE 请求[Call-ID 1]。





计费服务器流程中各步骤含义如下:

- 步骤 1, 会话初始化。
- 步骤 2, 目的方应答。
- 步骤 3, 收到最终的应答后, S-CSCF 发送 Accounting-Request 消息到 CCF, 此消息中包含指示 START\_RECORD 的 Accounting-Record-Type, 用来开始记录用户会话并产生 S-CSCF CDR。
- 步骤 4, CCF 应答, 并打开 S-CSCF CDR。
- 步骤 5, 和步骤 3 类似, 只是换成 P-CSCF。
- 步骤 6, 和步骤 4 类似, 但创建 P-CSCF CDR。

## 5.5 利用 SIP 提供 OSA 业务

### 5.5.1 OSA 简介

OSA 引入了 Internet 上的应用开发模式, 为 IT 应用与电信网的融合奠定了技术基础。OSA 规范采用面向对象的方法, 使用 UML (Unified Modeling Language) 语言进行描述, 而且 API 的实现基于中间件平台 (例如, CORBA 等)。这使得 OSA 的定义与网络技术无关, 它只为业务的实现提供统一的抽象接口, 由 OSA 开发的应用能在多种网络上运行。从根本上说, 现有网络业务不能实现跨网移植的主要原因在于: 业务的开发需要针对具体的网络, 业务逻辑一般通过专有协议直接控制网络功能实体。因此, 为特定网络开发的业务很难被移植到其他网络上去。另一方面, 跨网智能业务的开发要求业务开发人员还需要对不同网络的底层通信网络协议都有深入的了解, 这些因素抑制了业务的迅速开发, 无法满足用户对业务需求快速增长的需要。由于 OSA 的定义与具体技术无关, 因而具有网络独立性, 基于 OSA 开发的应用能在多种网络上运行。因此, 以 OSA 为基础可实现移动网络业务开放。

#### 1. OSA 的逻辑结构

OSA 规定了服务访问的几个实体。如应用 (Application)、应用服务器 (Application Server)、服务使能服务器 (Server Capability Server/SCS)、OSA 框架 (Framework) 和核心网元 (core network element)。

Applications 部署在应用服务器上, 可以基于任何标准的 IT 平台, 使用 SCS 提供的标准接口 API 来进行核心服务的访问。其中在标准接口 API 的服务端实现 SCS 功能, 在客户端实现应用 (Application)。Application 与 SCS 之间的通信, 可以使用标准的、开放性的中间件, 如通过 CORBA 技术来进行。

SCSs 实际上是负责 API 具体实现的功能实体, 即实现服务使能特性 (Service Capability Feature, SCF) 的接口类。SCS 与核心网络元素进行交互, 如 HLR, MSC, SSP 等。这样, 一个 SCS 服务伺服程序就相当于进入核心网络的一个代理或一个网关。

OSA API 规定了两大类接口: 框架接口 (framework interface) 和业务接口 (service interface), 并都用 CORBA IDL 进行了模型的定义。其中:

框架接口提供了支持利用服务接口访问网络服务的一些必需的功能, 提供业务接口安全, 管理性所必需的支持能力。如 AAA 认证、安全、服务注册、检测等功能。

业务接口, 即 Service Capability Server 部分, 提供应用访问网络能力和信息的接口。业务

接口提供途径使应用可访问传统网络能力，如呼叫管理、消息、用户交互等，业务接口也包括减轻通信应用程序的通用应用接口。

在应用和 SCS 之间的通信是使用标准的 IT 中间件机制，如 CORBA 技术来实现的。这样，SCS 是实现 API 的逻辑实体（即业务能力特性的接口类），它可能与核心网元进行交互。这些实体可能包含在归属位置寄存器（HLR）、移动交换中心（MSC）、业务交换点（SSP）中。所以 SCS 服务器可以看做是核心网络代理或者网关。一个 SCS 可以同时实现多个 SCF。由于 SCS 是逻辑实体，因此它的实现不必一定是独立的单元。例如，基于内容计费 SCF 很可能由计费（Charging）和账务（Billing）管理服务器来提供。

OSA 框架接口为应用提供了框架的能力。该实体是实现开放性的关键所在，它使得在传统 IN 范围之外实现开放、发现和集成新的业务特性成为可能。

OSA 框架提供了 SCF 的基本接入能力，它结合分布技术可以支持应用的位置信息和各种业务情景。而且，框架允许多方提供业务，甚至可以包括非标准化的 SCF，而这些业务能力是业务创新和区分的关键。框架由三大类不同的特性组成：可信度和安全管理、业务注册和检索功能以及完整性管理。其核心部分包括：信任和安全（鉴权）、SCF 注册（新的 SCF 在框架中进行注册）、SCF 工厂（创建新的 SCF 实例）、SCF 发现（发现由运营商提供的 SCF）。

## 2. OSA 的工作流程

要理解 OSA 的工作流程，可以参考 SCS 的安装过程和应用使用 SCS 中提供的能力的流程（如图 5-8 所示）。假定 SCS 实现了多方呼叫 SCF，其过程主要分为以下三步：

（1）注册新的 SCF（Registration）。首先，SCF 向框架请求注册接口（Registration interface），框架将返回该接口的引用（步骤 1~2）；接下来，多方呼叫控制 SCS 使用得到的该接口发布 SCF 的类型；而后，SCS 将通过注册接口向框架提供自己的引用（称为业务工厂 service factory）（步骤 3）。这时，框架和 SCS 就互相了解了对方。

（2）建立该业务协定（Setup of Service Agreement）。该步骤将建立允许应用使用 SCS 的环境。这里的条件之一是应用只允许使用最多有四方参与的呼叫，这些条件信息在“业务协定”（Service Agreement）中进行描述。

（3）建立通信。该步骤建立应用和 SCS 之间的通信连接。

应用接触框架后可以使用发现接口（Discovery interface）来查找可用的 SCF 实现（步骤 4~6）。假定这里应用想使用多方呼叫控制 SCF，并且想使用最多有四方参与的呼叫。因此，应用将通过发现接口（Discovery interface）向框架发送请求，要求返回所有可以满足要求的多方呼叫的 SCF 实现，然后框架将通过业务协定（Service Agreement）确认是否可以使用。如果满足条件，框架将返回这些 SCF 的列表。例如，现在有两个呼叫控制的 SCS，其中一个能够一次处理八方呼叫，一个可以一次处理六方呼叫。这样，应用可以从中选择处理六方呼叫的 SCF（步骤 7），因为比较便宜。然后，框架将通过业务工厂接口（Service Factory interface）要求具体的 SCS 创建可用的 SCF 实例；框架同时也将发送 SCF 使用条件信息，例如，这里要求每次呼叫最多四方参与（步骤 8）。这时 SCS 将创建 SCF 实例并且向应用返回框架的索引（步骤 9~10）。这样，应用就可以开始使用多方呼叫控制 SCF。

在应用使用每个 SCF 的时候都必须重复步骤 6~10。要注意的是 OSA 支持的鉴权步骤（步骤 1 以及步骤 4）在相同的域内可以忽略。例如，运营商为其框架添加新的 SCS，或者应用在与框架、SCS 相同的域中。

OSA 规范为每一个操作都定义了相应的 CORBA IDL 描述, 并通过序列图、class view 等 UML 建模方式对各操作进行了说明。

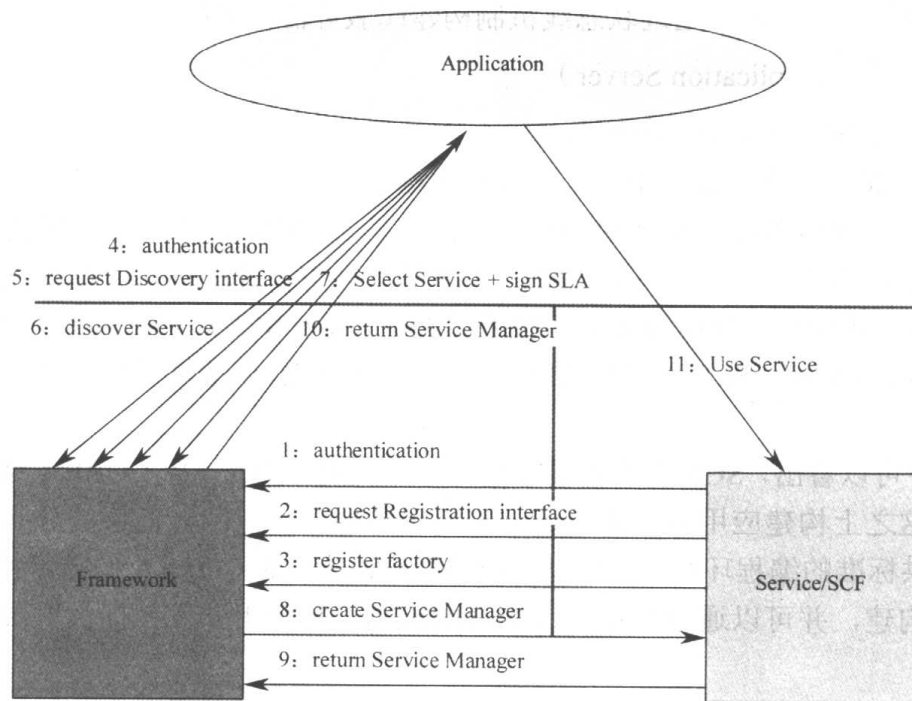


图 5-8 OSA 工作流程

### 3. 物理部署

提供 OSA API 功能的物理实体有多种类型, 例如, 交换机(switches)、IVR(Interactive Voice Response system)、HLR (Home Location Registrie)、GSSN (GPRS Support Node)、计费服务器(billing server)等。SCS 只是逻辑的实体, 并且在规范中并没有规定 SCS 是否是网络中的独立单元。原则上, SCS 可以作为网络上一个独立的节点来部署, 或者作为核心网络节点来部署。

如果 SCS 作为一个独立的节点来部署, 这种情况下, 在核心网络中支持业务的物理的子层可以明显区分。实现的方法有多种, 一种是在单个物理节点中提供所有 API 的实现, 通常被称为 OSA 的物理网关, 它具有面向所有核心网实体的协议和接口; 另一种是分布式方案, 其中 OSA 网关节点包含了框架以及一些 SCS 组件, 但是其他的 SCS 运行在不同的节点上。这就意味着 OSA 网关是一个逻辑概念的网关, 其 API 实现可以运行在分布式的节点上, 即不同的网络实体提供它们自己的 API。这种方案是 SCS 作为单独节点以及核心网络提供 SCF 实现的混合。

原则上, 如果在不同节点之间的中间件基础上开发, 则所有的开发都是可能的。然而, 有时可能不希望直接在核心网节点上开发 SCS 软件, 尤其在最终用户的触发要根据用户位置、信令或者处理负荷来动态存储的情况下。此时, 为了能够在用户启动它所感兴趣的事件时能够被触发, 则在位于网络中所有可能的节点应该能够建立和应用的通信。例如, 一个移动用户可能根据位置来附着到移动网络中的任何业务交换机上。

由此可见, 事实上可能有各种混合的实现情景。最可能的情况是: 一个网关节点提供 OAS 框架, 其中包括 SCF 的注册以及其他一些核心的 SCS; 其他的 SCS 分布在不同的节点上, 在框架中进行注册。由于框架是关键实体, 因此实现框架的节点能够提供一般电信承载的性

能，如要求达到 99.999%的可用性。而采用 CORBA 技术，对于应用部署的灵活性都是非常适合的。并且，通过 CORBA 中间件可以屏蔽低层的异构性和分布性，以一种统一的方式来进行低层的访问及控制，并且可以通过软总线机制构建起服务器之间的集群平台。

4. 应用服务器（Application Server）

基于 OSA 的体系架构如图 5-9 所示。

Application 部署在应用服务器上，可以基于标准的 IT 平台。服务使能服务器（Service Capability Server）向应用提供标准的 OSA API 接口。应用服务器使用 SCS 提供的标准接口 API 来进行核心网络的访问，如 HLR, MSC, SSP 等。其中在标准接口 API 的服务端实现 SCS 功能，在客户端实现应用。应用服务器和服务使能服务器（SCS）可以在同一个业务域或不同的业务域内。Application 与 SCS 之间的通信，可以使用标准的、开放性的中间件，如通过 CORBA 技术来进行。

从图 5-9 中可以看出，SCS 的部署很可能是分布式的。因此，像 CORBA 这样的技术是非常适合的，在这之上构建应用服务器。通过应用服务器，可以隐藏低层的分布细节，同时向应用开发者提供标准的编程环境，并可以选择熟悉的语言。而 CORBA 的技术特点，非常适合应用服务器的构建，并可以通过组件封装继承的机制，向应用开发者提供更快速有效的开发方式。

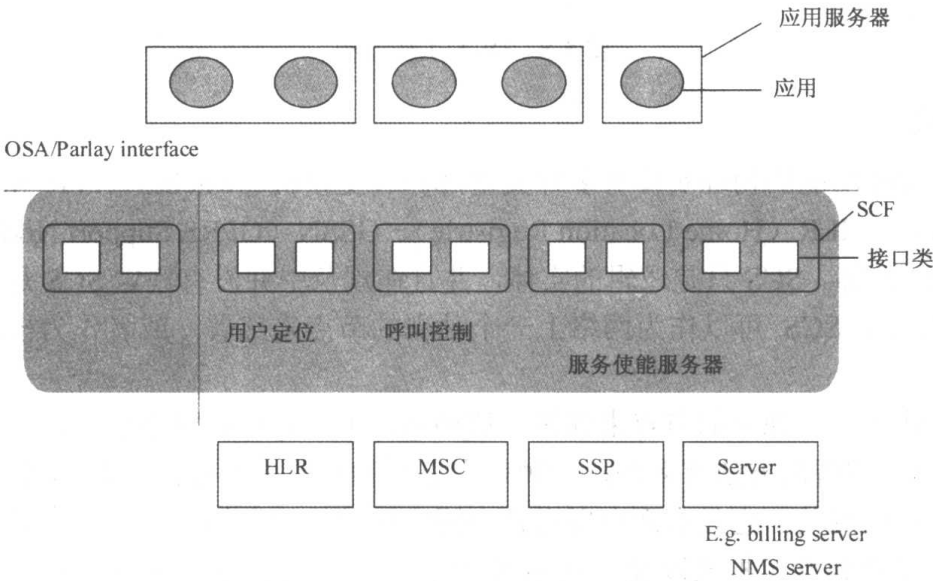


图 5-9 应用服务器结构

5.5.2 MPCC 到 ISC(SIP)的映射

1. 呼叫管理服务接口

1) CreateCall

createCall (appCall : in IpAppMultiPartyCallRef):TpMultiPartyCallIdentifier

此方法用来在 SCS 中创建新的呼叫对象。

流程如图 5-10 所示，操作及参数映射如表 5-2 和表 5-3 所示。

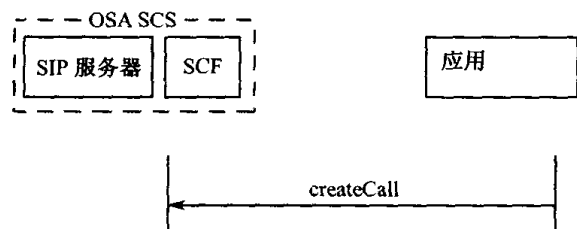


图 5-10 createCall()流程

表 5-2 正常操作

OSA SCS SIP 服务器模式	UA 模式
预先条件:	网络运营商与服务提供商已就允许应用创建呼叫对象达成一致
1	在 SCS 中创建了一个新的多方呼叫对象，并且应用得到呼叫对象的引用

表 5-3 参数映射

来自: createCall	到: SIP
appCall (IpAppMultiPartyCallRef)	N/A
Returns: TpMultiPartyCallIdentifier: - CallReference (IpMultiPartyCallRef) - CallSessionID (TpSessionID)	N/A

2) setCallLoadControl

setCallLoadControl (duration : in TpDuration, mechanism : in TpCallLoadControlMechanism, treatment : in TpCallTreatment, addressRange : in TpAddressRange) : TpAssignmentID  
此方法用来设置负载控制或者删除负载控制。  
流程如图 5-11 所示，操作及参数映射如表 5-4 和表 5-5 所示。

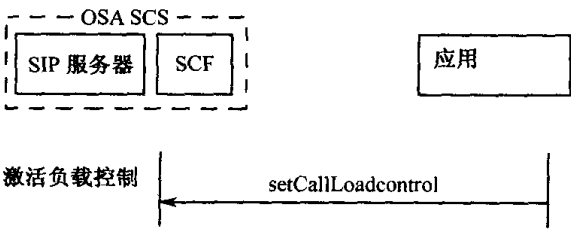


图 5-11 setCallLoadControl()流程

表 5-4 正常操作

OSA SCS SIP 服务器模式	Proxy, Redirect, UA, B2BUA, 第三方 parlay 控制器
预先条件:	网络运营商与服务提供商已就设置呼叫控制达成一致
1	应用调用 setCallLoadControl 方法来删除或者设置负载控制
2	SCS 请求 SIP 服务器来激活或者删除呼叫负载控制

表 5-5 参数映射

来自 setCallLoadControl	到 SIP
duration (TpDuration)	N/A
mechanism (TpCallLoadControlMechanism)	N/A
treatment (TpCallTreatment) TpCallTreatment sequence of: - TpCallTreatmentType, - TpReleaseCause	TpCallTreatment Type 和 TpReleaseCause
addressRange (TpAddressRange)	

2. 呼叫管理器程序接口

managerInterrupted () : void

此方法用来指示所有的事件通告和方法调用都被临时截获（由于网络资源不可用等原因）。  
流程如图 5-12 所示，操作及参数映射如表 5-6 和表 5-7 所示。

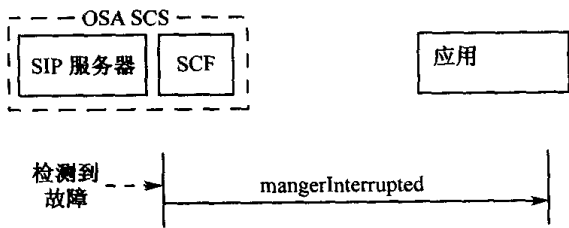


图 5-12 managerInterrupted()流程

表 5-6 正常操作

OSA SCS SIP 服务器模式	Proxy, Redirect, UA, B2BUA 或者第三方 parlay, 3rd Parlay 控制器
预先条件:	网络运营商与服务提供商就呼叫通告已达成一致，呼叫通告已经激活
1	SCS 检测到或者被通知由于故障，将不会得到更进一步的事件通告
2	SCS 调用 managerInterrupted 方法

表 5-7 参数映射

来自 managerInterrupted	到 SIP
-	N/A

3. 多方呼叫服务接口

createCallLeg (callSessionID:in TpSessionID,appCallLeg:in IpAppCallLegRef):TpCallLegIdentifier  
此方法用来在 SCS 中创建新的 CallLeg 对象。

流程如图 5-13 所示，操作及参数映射如表 5-8 和表 5-9 所示。

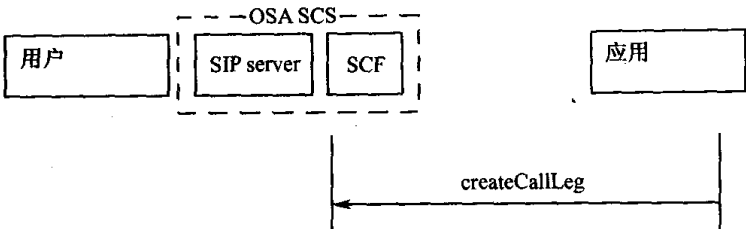


图 5-13 createCallLeg()流程

表 5-8 正常流程

OSA SCS SIP 服务器模式	Proxy, UA, B2BUA 或第三方 parlay 控制器（任意，除了重定向）
预先条件:	已经得到多方呼叫对象的引用
1	应用调用 createCallLeg 方法
2	SCS 创建请求的呼叫 leg 对象

表 5-9 参数映射

来自 callOverloadEncountered	到 SIP
callSessionID (TpSessionID)	Dialogue Correlation
appCallLeg (IpAppCallLegRef	N/A
返回 TpCallLegIdentifier: - CallLegReference (IpCallLegRef) - CallLegSessionID (TpSessionID)	Dialogue Correlation

4. 多方呼叫应用接口

callEnded (callSessionID : in TpSessionID, report : in TpCallEndedReport) : void  
当呼叫在网络中中止时，调用此方法。

流程如图 5-14 所示，操作及参数映射如表 5-10 和表 5-11 所示。

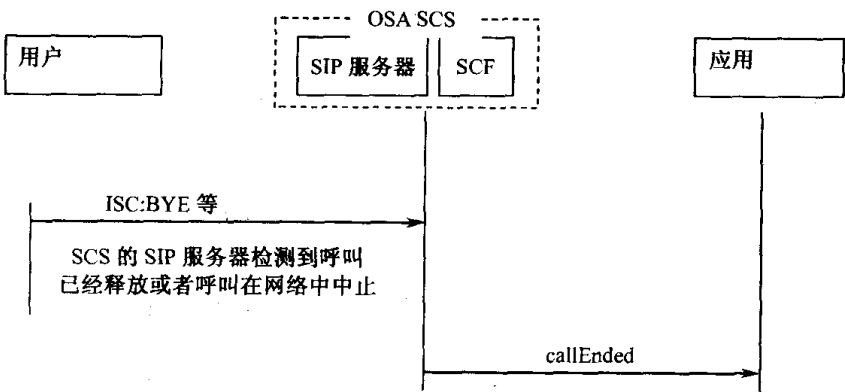


图 5-14 callEnded() 流程

表 5-10 正常操作

OSA SCS SIP 服务器模式	Proxy、UA、B2BUA 或者第三方 parlay 控制器，重定向
预先条件:	有应用监测呼叫
1	SCS 检测到没有关联到呼叫的 leg 或者呼叫已经中止，SCS 调用 callEnded 方法

表 5-11 参数映射

到 callEnded	来自 SIP: BYE, 3xx, 4xx, 5xx, 6xx
callSessionID (TpSessionID)	Dialogue Correlation
report (TpCallEndedReport) :	-
- CallLegSessionID (TpSessionID)	Dialogue Correlation
- Cause (TpReleaseCause)	TpReleaseCause

5. CallLeg 服务接口

getInfoReq (callLegSessionID: in TpSessionID, callLegInfoRequested : in TpCallLegInfoType) :  
void



它为一异步方法，用来请求与呼叫相关联的信息。

流程如图 5-15 所示，操作及参数映射如表 5-12 和表 5-13 所示。

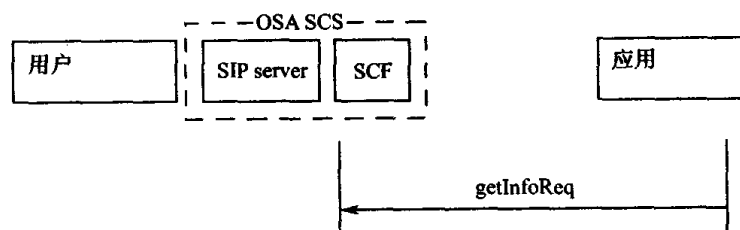


图 5-15 getInfoReq()流程

表 5-12 正常操作

OSA SCS SIP 服务器模式:	Proxy, UA, B2BUA 或者第三方 parlay 控制器(任意, 除了重定向模式)
预先条件:	应用与呼叫之间的关系存在
1	应用调用 getInfoReq 方法, SCS 监视呼叫 leg 是否能收集请求信息。
2	OSA SCS 根据从 OSA SCS SIP 服务器接收的消息发送对应的 getInfoRes()或者 getInfoErr()

表 5-13 参数映射

来自 getInfoReq	到 SIP
callLegSessionID (TpSessionID)	Dialogue Correlation
callLegInfoRequested (TpCallLegInfoType):	TpCallLegInfoType

## 6. CallLeg 应用接口

eventReportRes (callLegSessionID : in TpSessionID, eventInfo : in TpCallEventInfo) : void

它为一异步方法，用于报告出现在呼叫 leg 处的事件。

流程如图 5-16 所示，操作及参数映射如表 5-14 和表 5-15 所示。

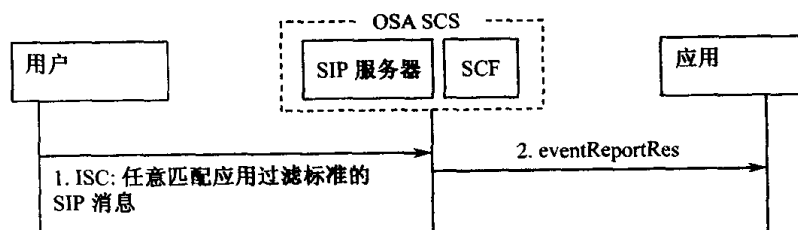


图 5-16 eventReportRes()流程

表 5-14 正常操作

OSA SCS 的 SIP 服务器模式	Proxy, UA, B2BUA 或者第三方 party 控制器
预先条件:	应用与呼叫的关系存在
1	OSA SCS 的 SIP 服务器检测到 SIP 消息 (应答或请求)
2	OSA SCS 调用 eventReportRes()方法

表 5-15 参数映射

到 eventReportRes	来自 SIP (任意 SIP 消息)
callLegSessionID (TpSessionID)	Dialogue Correlation
eventInfo (TpCallEventInfo)	TpCallEventInfo

### 5.5.3 利用 SIP 可开展的 OSA 业务

OSA 提供的业务包括大众/消费者业务和企业应用,前者包含数据消息类业务、用户位置业务和 PAM 业务,后者包含数据消息业务、呼叫控制/用户交互业务和用户位置业务。以下为典型业务:

- Enago Protal: 门户业务给订购了业务的用户提供了一个入口。
- IKV's Smart Messenger: 是针对移动用户的消息工具,提供多种方式的通信能力。
- DocAvatar: 与人的健康相关的交互多媒体内容传送业务,其中 DocAvatar 业务基于 ParlayX API。
- Broadmedia 移动门户 (pull 型业务): 许多信息集成在同一个门户上,这样用户就可以通过手机终端或者 PDA 快速得到这些信息。
- Broadmedia 提醒 (push 型业务): 允许用户定制和接收他感兴趣的信息类型和数量。
- Broadmedia 广告业务: AgilBanner 是一种工具,可以允许商家通过网页进行商业需要的广告活动 (可以通过 IP 分段、时间、频率、浏览器或网页等进行区分)。当一个网页被点击时,应用就会按照相应的标准发布相关的广告信息,同时应用会对广告用户进行计费。应用引入位置过滤器,这样不同位置的用户可能会看到不同的广告。
- GeoPlaneta iMaps 服务: 提供地图和定位服务。包括地图服务和内容服务: 欧洲和城市地图、导航、住地搜索和个性化搜索等。
- UC3M 直接网络浏览业务 (IWB): IWB 包括一个为 UMTS 终端提供的优化网络浏览器,可以通过网络技术和 Parlay 智能地将新鲜的内容预取下来。
- AtosOrigin 预付费业务: 一个符合 Parlay 规范的 Java 语言编写的预付费业务。
- TSSG Media Messenger 业务: 基于 SIP Messaging and Presence Leveraging Extensions' (SIMPLE) 协议之上的直接通信业务。它的基于 Java 的界面可以运行于任何安装了 java 虚拟机的 PC 机或 PDA 之上。Media Messenger 使用了 MIME 格式传输媒体信息,同时也采用了 MSN 的技术。这使得它可以与 MSN 进行互通。下一步,Media Messenger 还将提供基于 IP 的语音和图像传输功能。
- MailMorph 业务: 提供邮件服务。
- Customer Care 业务: 该业务可以为某个企业的消费者提供服务。提供的服务包括: 交互式的产品指南 (用户可以及时享受产品的服务,可以通过向导看到需要产品的安装和使用说明等,也可以选择前进、倒退和重放向导)。在线帮助 (通过在线的客服人员为用户提供音频或者视频的服务。如果用户有了困难,只要他通过 Customer Care 服务发出请求,那么该业务通过用户目前所在的位置会自动地为他选择一位最佳的在线服务人员为他提供服务)。
- UMS 统一消息服务: 为用户提供多种消息服务。用户可以管理自己的地址簿、设置个人信息、享受日程提醒、邮件、呼叫转移等多种服务。UMS 需要用到 OSA FW, GCCS, UI, GMS 等多个接口。
- LBS 业务: 定位业务提供用户通过短消息或者 WEB 方式发送定位请求,得到自己或者朋友的位置信息。定位业务主要采用 OSA MM 接口,向 OSA 网关发出定位请求,

OSA 网关通过和移动 GMLC 的交互得到位置信息，返回给应用。

5.6 利用 SIP 开展 CAMEL 智能业务

如图 5-17 所示，CSCF 与 IM-SSF 之间的接口为 SIP，IM-SSF 与 gsmSCF 之间的接口为 CAP，HSS-IM-SSF 之间的接口则为 MAP。相应地，在 CAMEL 签约数据类型中增加了 IP Multimedia CAMEL Subscription Information (IM-CSI)，在状态模型中增加了 IM-BCSM。由于 IM-SSF 一侧为 CAP 协议，另一侧为 SIP 协议，所以需要完成相应的映射，具体如表 5-16 所示。

可利用智能网平台对呼叫进行智能控制、对移动性管理的控制、对签约数据的查询和修改、支持定位业务、支持对 SMS 业务的控制、补充业务的调用等。运营者可根据其业务需要开展预付费业务、亲情号码业务、分时、分区计费（Cell and Time Discount）等业务。

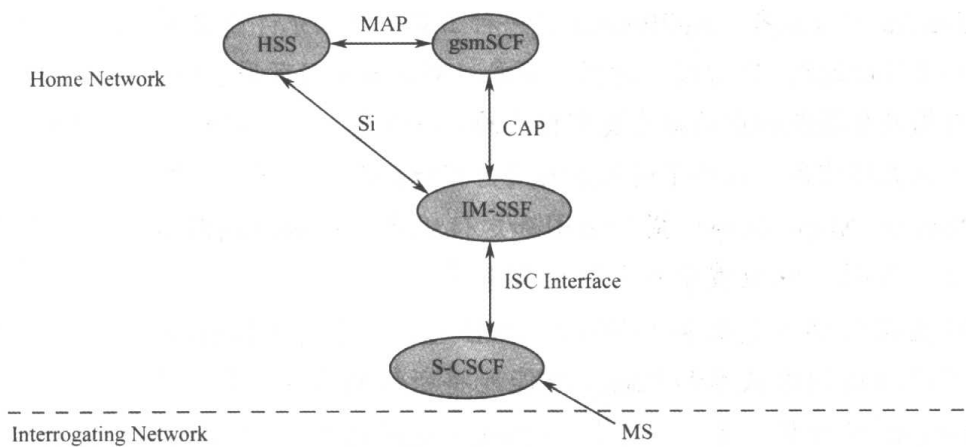


图 5-17 IMS 中的 CAP 协议

表 5-16 CAMEL DP 到 SIP 的映射

CAMEL O-IM-BCSM DP:	SIP Method/Response
DP Collected_Info	INVITE
DP Analysed_Information	N/A
DP Route_Select_Failure	4XX （不含 401, 407, 408, 480, 486）， 5xx 和 6xx （不含 600, 603）
DP O_Busy	486 此处忙、600 到处忙
DP O_No_Answer	603 拒绝、408 请求超时、480 临时不可用
DP O_Answer	200 OK
DP O_Disconnect	BYE
DP O_Abandon	CANCEL

5.7 利用 SIP 应用服务器提供业务

5.7.1 LCS 业务

利用 IMS 域支持 LCS 业务的需要解决的首要问题是，由于现有的规范定义的几种定位方

法只能通过 target UE 的 MSISDN 或是 IMSI 作为用户标识来进行定位, 而在 IMS 域中却是采用 SIP URL 作为用户标识, 因此, 当 LCS 系统收到一个以 SIP URL 为用户标识的位置信息请求时, 需要将 SIP URL 转换成用户的 MSISDN 或是 IMSI, 才能进行后续的标准 LCS 规范中定义的定位处理流程。

目前能够被大家接受的是如下几种方案: S2-034218 (Introduction of IMS to LCS Architecture) (Vodafone), S2-034171 (LCS in IMS – some thoughts on S2-033589) (Nokia), S2-034235 (Integration of IMS users in the LCS architecture) (Siemens), 最后一种方案如图 5-18 所示。

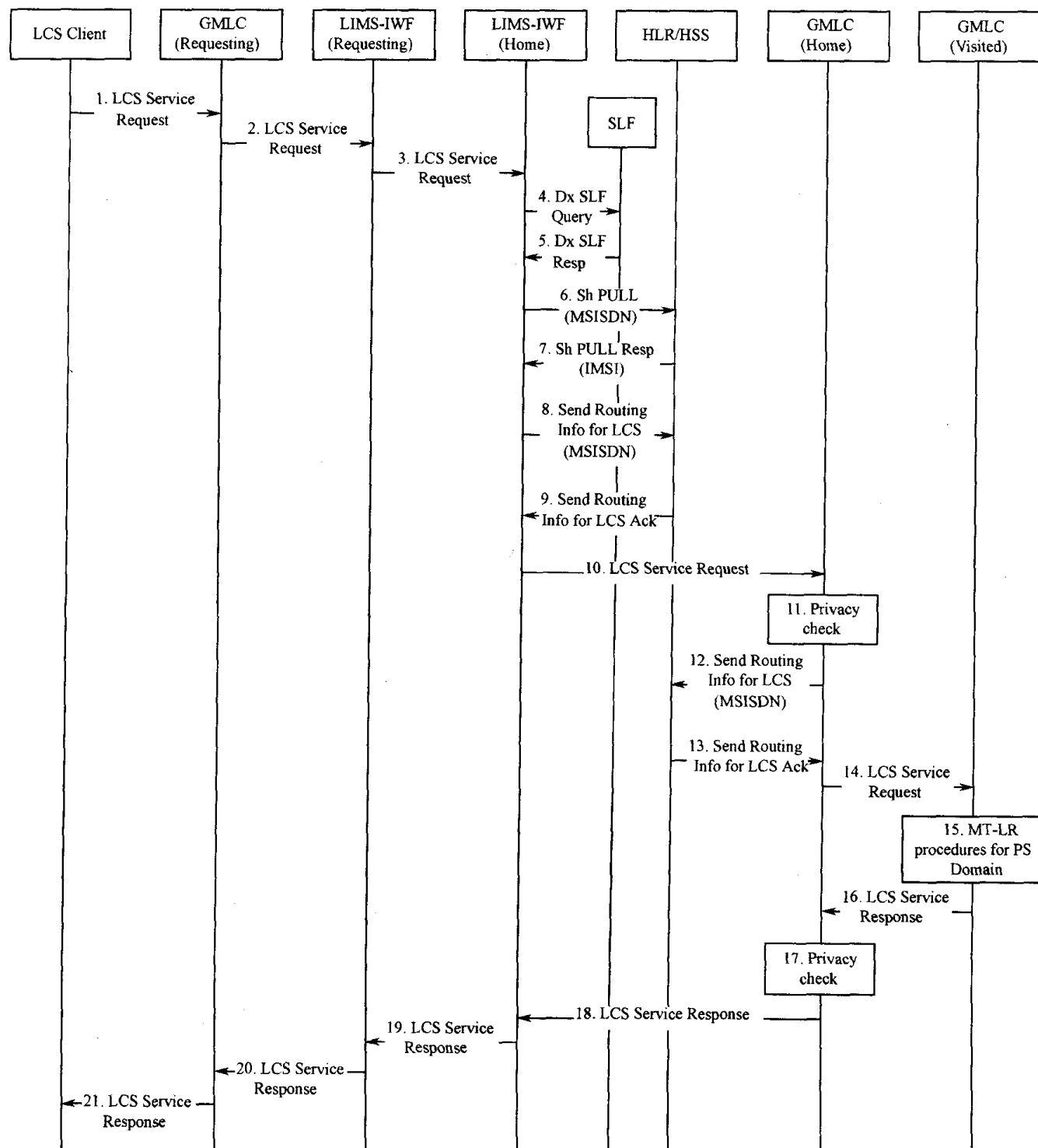


图 5-18 基于 IMS 的 LCS 结构

### 5.7.2 IM/Presence 业务

#### 1) 可提供的业务及与其他业务的关系

可提供以下基本业务:

- IM 业务用户使用点对点、点对多点的消息发送和接收, 包括内容文本、多媒体等;
- Presence 业务用户订阅其他用户/应用的 Presence, 同时能被订阅。Presence 改变后, 自动通知被授权者更改 Presence 信息;
- Group 业务用户创建、加入组等操作, 组属性更改后, 可通知相关人员;
- 好友列表用户创建、删除好友列表, 修改、制定好友列表的属性和成员等;
- 搜索业务用户根据需要, 按用户名等属性搜索相关用户, 进行匹配;
- 邀请业务用户邀请其他用户加入一个组、交换消息、共享 Presence 信息、共享内容等。
- 与其他业务的关系:
- MMS 和 SMS 通过使用 Presence 信息, 增强业务特性 (如在线状态、当前适合的通信方式等);
- 不方便通话时, 把语音转换为 MMS 或 SMS 消息传送给用户;
- 通过 SMS 接入方式使用 IM;
- IM 通过 LBS 获取用户的位置类的 Presence 信息;
- LBS 通过 IM 用户 Presence 信息决定是否推送位置信息, 同时也可以通过 IM 向用户接入 LBS 和传送位置信息;
- IM 的消息在用户不在线时可以选择转到其 E-mail 邮箱中。

#### 2) 3GPP 规范情况

3GPP R6 通过 IMS Presence 协议体系实现 IM Presence 业务。

核心网基于 SIP, 充分借鉴 SIMPLE 协议 (定义新方法 MESSAGE 来支持消息 IM 业务, 定义 Subscribe、Notify 方法来支持 Presence 业务), 需要增加 PUBLISH 事件, 只需修改相关参数。

协议参数还不是很成熟, 而支持的 Presence 属性只列举了一部分, 对终端的支持当前还不够完善。

#### 3) 3GPP IMS 消息分类

3GPP 在 TS 22.340 中将 IMS 消息分为以下几类:

##### ◆ Immediate messaging 立即消息

属于实时消息, 一般发送方知道接收方的状态, 与 Presence 业务相结合, 若接收方不可达, 消息将被丢弃或延迟。不需建立 SIP。

##### ◆ Session based messaging 基于会话的消息

近似实时传送, 通过建立 SIP 会话可进行多条消息传送, 相应机制在 IETF 中有定义。

##### ◆ Deferred delivery messaging 延迟传送的消息

属于存储转发型消息, 即 MMS。

#### 4) IMS 消息流程

IMS 消息流程如图 5-19 所示, 其中细线为 SIP 消息, 粗线为 MRSP (Message Session Relay

Protocol) 消息。

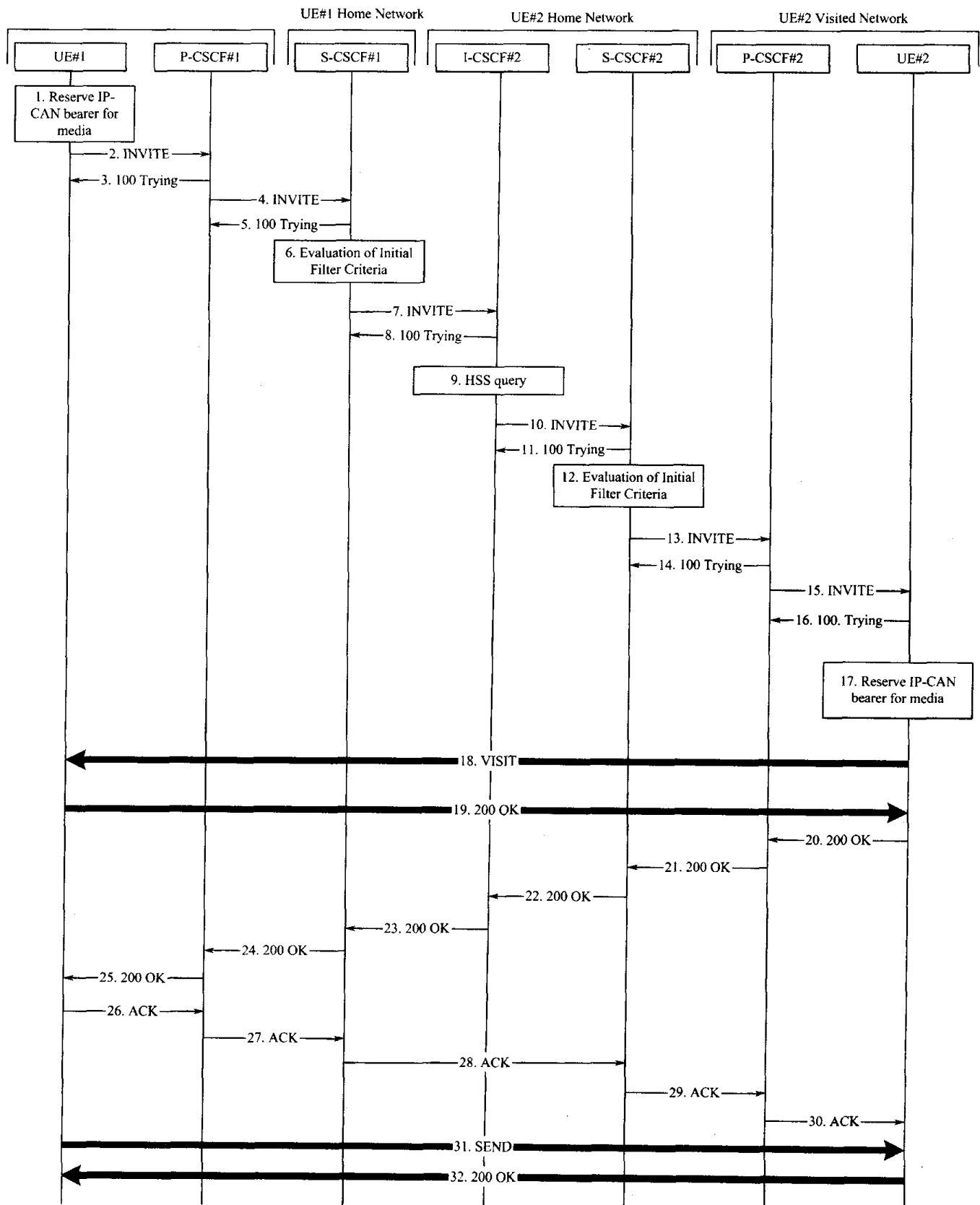


图 5-19 会话建立流程（无 preconditions 且无 SBLP）

### 5) Presence 业务架构及业务

Presence 业务架构及相关参考点如图 5-20 所示。watchers 获得事件通告的流程如图 5-21 所示，内容存储流程如图 5-22 所示。

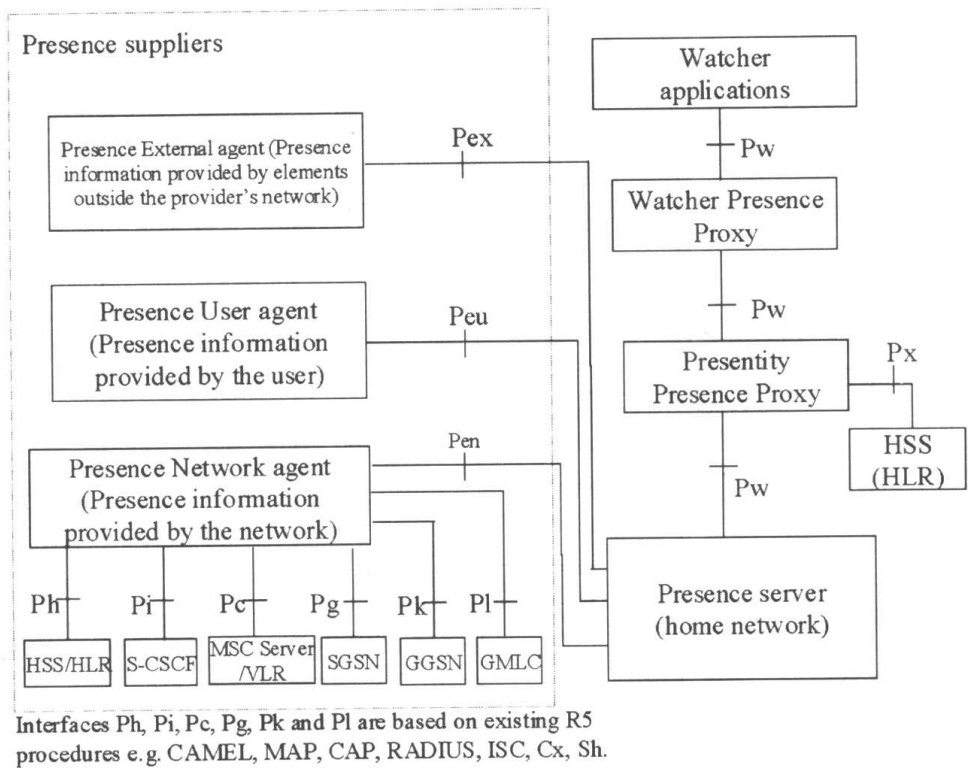


图 5-20 Presence 业务架构

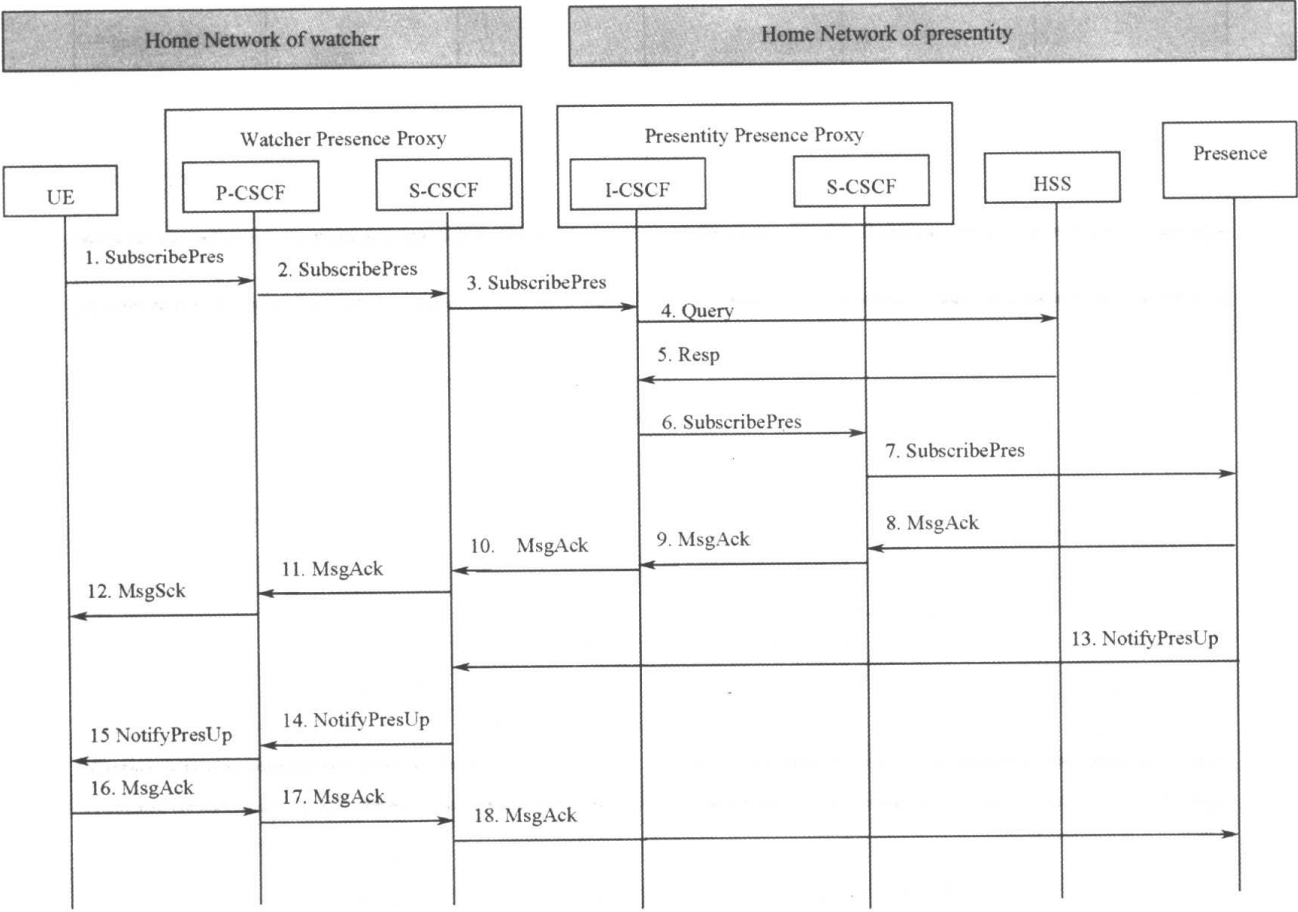


图 5-21 IMS Watcher 注册以获得事件通告流程

图 5-21 中各个步骤的含义如下：

- 步骤 (1), UE 中的 watcher agent 发送 SubscribePres 请求。
- 步骤 (2), P-CSCF 转发 SubscribePres 给 S-CSCF。
- 步骤 (3), 由于 S-CSCF 不能解析 presence 服务器地址, 继续转发 SubscribePres 给 I-CSCF。
- 步骤 (4), I-CSCF 查询 HSS 以得到与 Presentity 关联的 S-CSCF 地址。
- 步骤 (5), HSS 发送的 Query Resp 消息含有与 Presentity 关联的 S-CSCF 名字。
- 步骤 (6), I-CSCF 将 SubscribePres 消息转发给 S-CSCF。
- 步骤 (7), S-CSCF 利用必要的过滤准则, 将 SubscribePres 消息转发给 Presence 服务器。
- 步骤 (8) ~ (12), Presence 服务器进行必要的认证, 如果认证通过, 发送 MsgAck 消息, 并最终到达 UE。
- 步骤 (13), Presence 服务器发送 NotifyPresUp 消息到 S-CSCF。
- 步骤 (14) ~ (15), S-CSCF 转发 NotifyPresUp 消息, 并最终到达 UE。
- 步骤 (16) ~ (18), UE 发送 MsgAck 应答消息, 并最终到达 Presence 服务器。

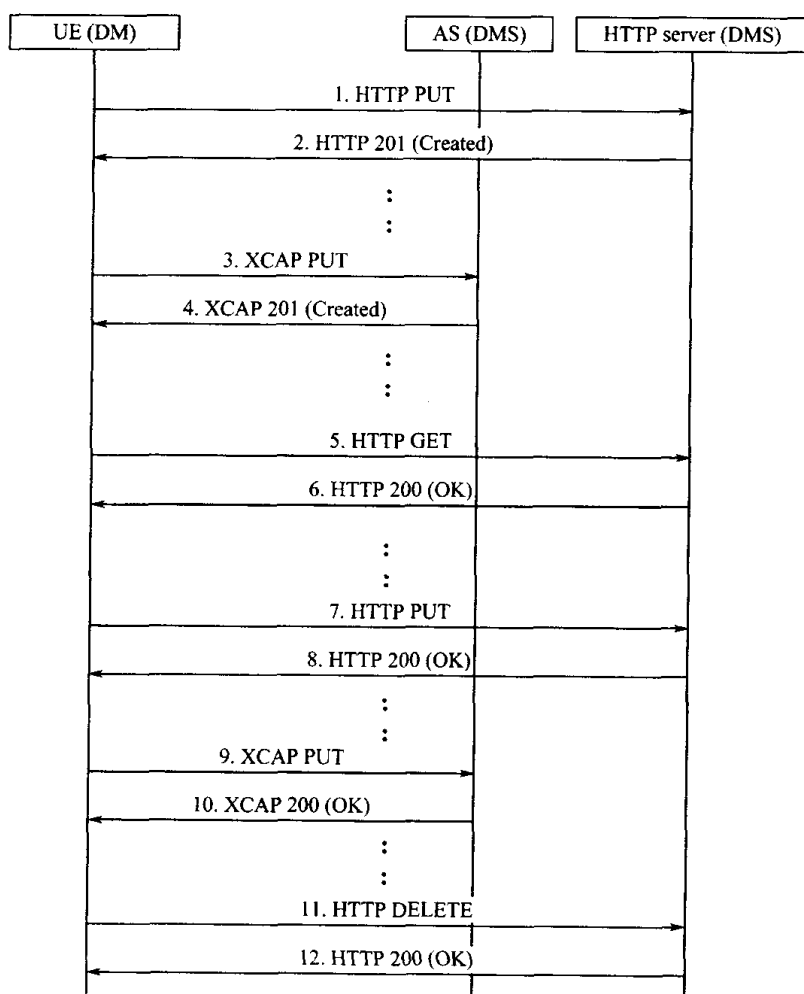


图 5-22 内容存储流程

### 5.7.3 紧急会话建立

紧急会话建立流程如图 5-23 所示, 图中各个步骤的含义如下:



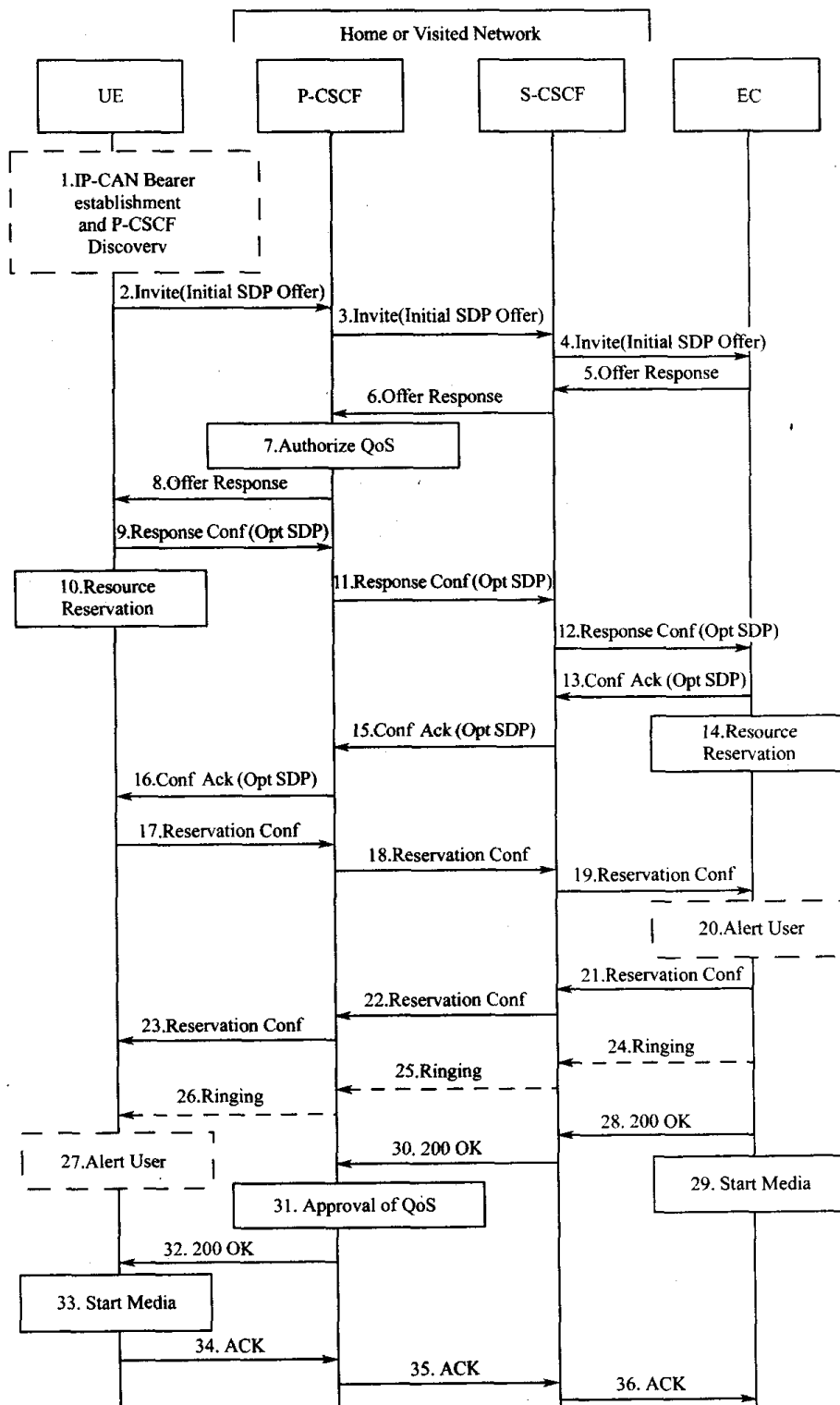


图 5-23 紧急会话建立（紧急中心支持 SIP 协议）

- 步骤（1），查找 P-CSCF。
- 步骤（2）~（4），UE 发送 SIP INVITE 请求，并最终到达 EC。
- 步骤（5）~（8），返回 Offer Response 消息，并最终到达 UE。其中，步骤 7 为检查 UE 是否有对资源的使用权限。
- 步骤（9）~（12），UE 发送 Response Confirmation 消息，并最终到达 EC。其中，步骤（10）为资源预留。
- 步骤（13）~（16），EC 发送 Response Confirmation 应答消息，并最终到达 EC。其中，

步骤 14 为资源预留。

- 步骤 (17) ~ (19), UE 发送预留 Confirmation 消息, 并最终到达 EC。
- 步骤 (20), 通告用户。
- 步骤 (21) ~ (23), EC 发送预留 Confirmation 消息, 并最终到达 UE。
- 步骤 (24) ~ (26), 为振铃消息。
- 步骤 (27), 通告用户。
- 步骤 (28) ~ (36), 媒体传输。

#### 5.7.4 利用 IMS 来支持会议

利用 IMS 来支持会议的标准目前为 24.147, 版本为 V0.1.0, 其加入会议的流程如图 5-24 所示。

图 5-24 中各个步骤的含义如下:

- 步骤 (1), INVITE 请求 UE 到 P-CSCF。
- 步骤 (2), 100 Trying 响应 P-CSCF 到 UE。
- 步骤 (3), INVITE 请求 P-CSCF 到 S-CSCF。
- 步骤 (4), 100 Trying 响应 S-CSCF 到 P-CSCF。
- 步骤 (5), 检测初始过滤条件。
- 步骤 (6), INVITE 请求 S-CSCF 到 I-CSCF。
- 步骤 (7), 100 Trying 响应 I-CSCF 到 S-CSCF。
- 步骤 (8), Public service identity (PSI)位置查询, I-CSCF 发送查询消息到 HSS, 以期获得会议创建的 MRFC/AS。
- 步骤 (9), INVITE 请求 I-CSCF 到 MRFC/AS。
- 步骤 (10), 100 Trying 响应 MRFC/AS 到 I-CSCF。
- 步骤 (11), H.248 交互来创建 UE#1 的会议资源连接。MRFC 发起的 H.248 交互, 用来为 UE#1, 在 MRFP 中创建会议连接资源。
- 步骤 (12), 183 Session Progress 响应 MRFC/AS 到 I-CSCF。
- 步骤 (13), 183 Session Progress 响应 I-CSCF 到 S-CSCF。
- 步骤 (14), 183 Session Progress 响应 S-CSCF 到 P-CSCF。
- 步骤 (15), Authorize QoS Resources。
- 步骤 (16), 183 Session Progress 响应 P-CSCF 到 UE。
- 步骤 (17), PRACK 请求 UE 到 P-CSCF。
- 步骤 (18), 资源预留。
- 步骤 (19), PRACK 请求 P-CSCF 到 S-CSCF。
- 步骤 (20), PRACK 请求 S-CSCF 到 I-CSCF。
- 步骤 (21), PRACK 请求 I-CSCF 到 MRFC/AS。
- 步骤 (22), 200 OK 响应 MRFC/AS 到 I-CSCF。
- 步骤 (23), H.248 交互来修改 UE#1 连接。
- 步骤 (24), 200 OK 响应 I-CSCF 到 S-CSCF。

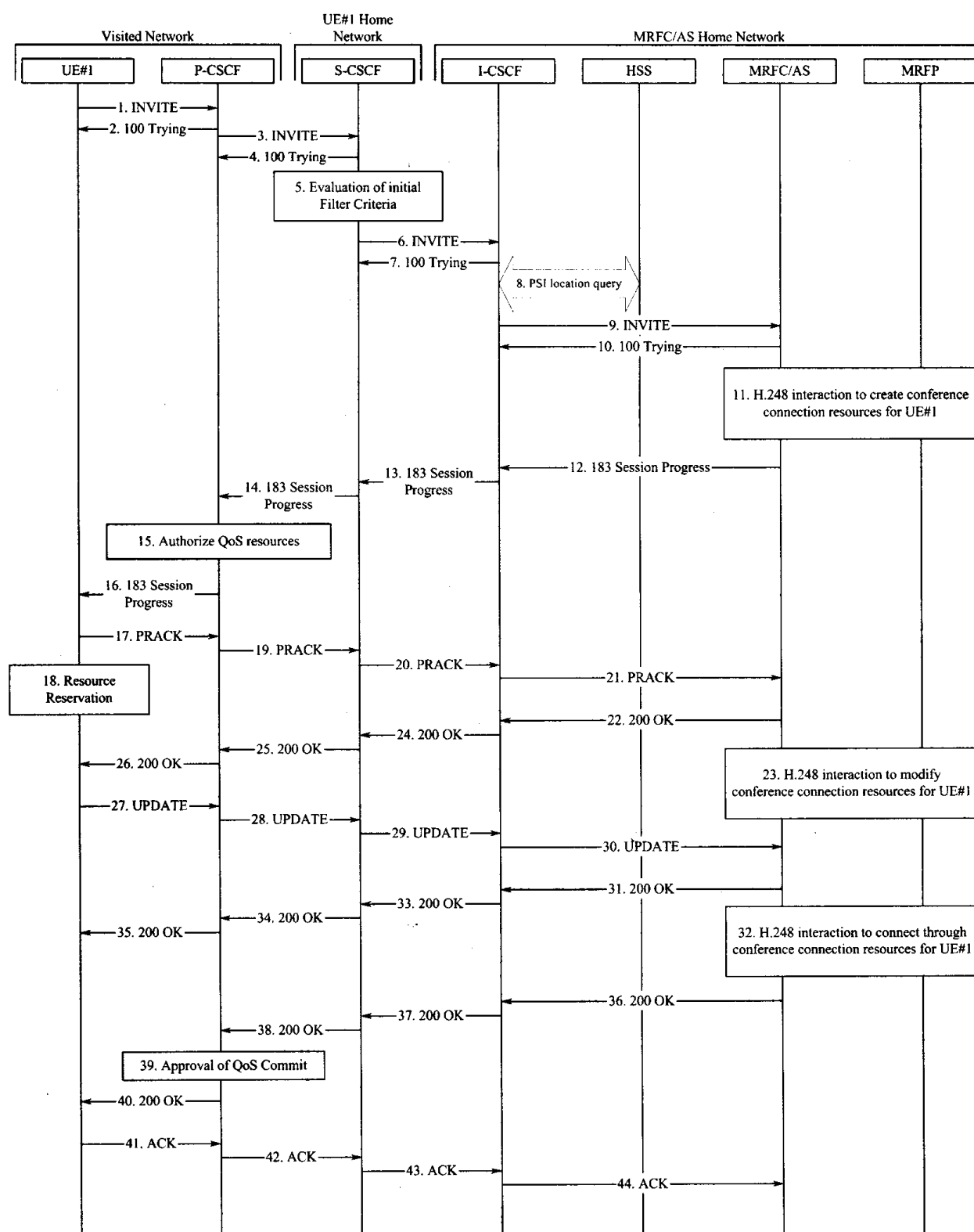


图 5-24 加入会议流程

- 步骤 (25), 200 OK 响应 S-CSCF 到 P-CSCF。
- 步骤 (26), 200 OK 响应 P-CSCF 到 UE。
- 步骤 (27), UPDATE 请求 UE 到 P-CSCF。
- 步骤 (28), UPDATE 请求 P-CSCF 到 S-CSCF。
- 步骤 (29), UPDATE 请求 S-CSCF 到 I-CSCF。
- 步骤 (30), UPDATE 请求 I-CSCF 到 MRFC/AS。

- 步骤 (31), 200 OK 响应 MRFC/AS 到 I-CSCF。
- 步骤 (32), H.248 交互来修改连接。
- 步骤 (33), 200 OK 响应 I-CSCF 到 S-CSCF。
- 步骤 (34), 200 OK 响应 S-CSCF 到 P-CSCF。
- 步骤 (35), 200 OK 响应 P-CSCF 到 UE。
- 步骤 (36), 200 OK 响应 MRFC/AS 到 I-CSCF。
- 步骤 (37), 200 OK 响应 I-CSCF 到 S-CSCF。
- 步骤 (38), 200 OK 响应 S-CSCF 到 P-CSCF。
- 步骤 (39), QoS 接纳批准。
- 步骤 (40), 200 OK 响应 P-CSCF 到 UE。
- 步骤 (41), ACK 请求 UE 到 P-CSCF。
- 步骤 (42), ACK 请求 P-CSCF 到 S-CSCF。
- 步骤 (43), ACK 请求 S-CSCF 到 I-CSCF。
- 步骤 (44), ACK 请求 I-CSCF 到 MRFC/AS。

### 5.7.5 PoC 业务

PTT (Push-To-Talk, 按下通话) 是一种半双工的通信方式, 在支持 PoC 的用户终端上, 用户仅仅按一个键就可以和别的用户通话。PoC (Push-To-Talk over Cellular) 作为一种新型的移动通信业务, 潜在的盈利和利润增长点, 受到包括运营商和设备提供商在内的通信界的极大关注。

在这种市场背景下, 爱立信、诺基亚和西门子于 2003 年 10 月 8 日宣布, 这三家公司合作制定的基于 3GPP IMS (IP 多媒体子系统, IP Multimedia Subsystem) 的 PoC 开放规范已经正式完成。目前的版本是 1.0。该开放性规范的出现有助于降低运营、设备和终端设备市场的分裂, 保证各厂商系统间的互联互通, 并让用户更加容易地使用 Push-To-Talk 业务。爱立信、诺基亚和西门子进行 PoC 技术合作的目标有两个: 为移动运营商提供一个易于集成和全球互连的技术解决方案; 创造规范的用户终端设备的竞争环境。PoC 标准得到包括 AT&T 无线、Cingular、Sonim 科技和索尼-爱立信在内的一些工业界巨头的支持。

该 PoC 规范融合了现存的 3GPP、OMA 和 IETF 标准, 它的优点是使 PTT 业务很容易集成到现有的接入平台和包交换核心网络, 例如, GPRS, EDGE, CDMA 1x 和 802.11 等。PoC 规范由六个子规范组成: 用户需求、体系结构、信令流、用户群组/联系列表及免打扰管理和两个用户平台 (传输协议和无线接入网络)。下面简要介绍各子规范的内容。

#### 1. 用户需求子规范

用户需求子规范描述了用户和运营商对 PoC 系统和业务的要求。其包括如下内容:

- 用户可以使用的 PoC 通话方式。两种一对一通话: 即时通话 (自动应答) 和请求应答通话; 三种群组通话: 自由加入的聊天组通话、被动邀请式的即时组通话和临时建立的组通话。
- PoC 系统必须支持多种功能。两种用户识别方式 (E-mail 形式和电话号码格式)、用户列表/组管理、用户注册、用户漫游、通话时的资源控制、话音流处理、用户状态呈现、勿打扰功能和用户终端上的各种设置。

- 用户进行 PoC 通话和用户列表/组管理的各种流程。
- 正在进行一个 PoC 通话的用户处理另一个 PoC 通话或别的数据应用的方式。
- 可供选择的计费模式：包月式的固定费率、按通话时间（说话加收听时间）长度计费、按被授予的说话时间长度计费、按说话的话音媒体大小计费和按说话的次数计费。

## 2. 体系结构子规范

体系结构子规范描述了实现用户需求子规范的系统体系结构。该子规范覆盖如下内容：

- 基本的体系结构框图，如图 5-25 所示。

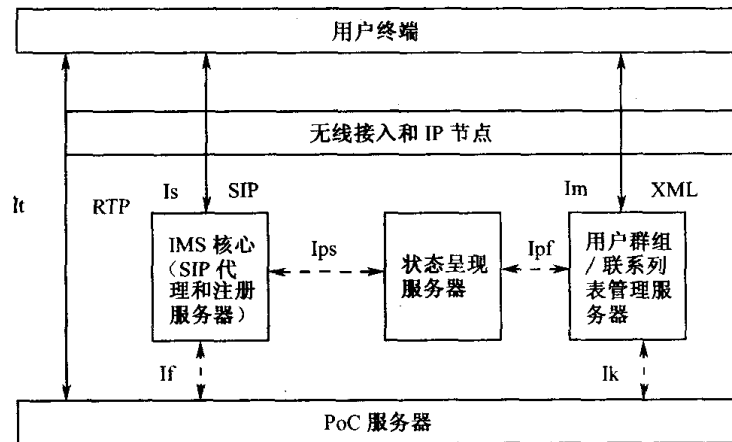


图 5-25 PoC 体系结构框图

- 体系结构中的五种通信实体：用户终端设备（运行 PoC 客户端的应用软件），IMS 核心（包含多个 SIP 代理和注册服务器，实现 SIP 信令路由，用户终端的 SIP 信令解压、授权、维护注册和 SIP 对话状态以及计费系统的接口），用户群组/联系列表管理服务器（实现用户组、联系列表和访问列表的管理），PoC 服务器（终结 SIP、RTP 和 RTCP 流，并实现 SIP 对话处理、组访问控制、组通话处理、用户访问控制、勿打扰功能、资源控制、通话用户识别、通话质量反馈、计费接口和语音的分发）和状态呈现服务器（管理和记录当前用户和网络状态）。
- 体系结构中，通信实体之间的七种接口：用户终端和 IMS 核心之间的 Is 接口（使用 UDP 上的 SIP 协议）、IMS 核心和 PoC 服务器之间的 If 接口（实现通话控制），用户终端和 PoC 服务器之间的 It 接口（支持语言流、资源控制和链路质量信息，使用 RTP 和 RTCP 协议），用户终端和列表服务器之间的 Im 接口（支持列表管理和勿打扰，使用 XML/HTTP 协议），PoC 服务器和列表服务器之间的 Ik 接口（支持群组 and 访问列表的获取）、IMS 核心和状态呈现服务器之间的 Ips 接口（支持用户注册信息的上载）、状态呈现服务器和列表服务器之间的 Ipl 接口（支持勿打扰和访问列表的上载）。其中 If, Ik, Ips 和 Ipl 接口上的协议在该规范中没有定义。
- 一些系统概念，包括：用户和群组识别、寻址方式、路由原则、安全性、资源控制、语言编码、信令压缩、用户平台自适应、计费方式、漫游、状态呈现和勿打扰。
- 高层的信令流介绍。

## 3. 信令流子规范

信令流子规范定义了 PoC 体系结构中的实体在 Is 接口上的各种信令流程。其包括如下内容：

- 公共的流程：SIP 路由、用户身份验证、信令压缩、通话媒体建立、PoC 服务器对通

话过程的控制流程和通话计时器的管理流程。

- 为使用 PoC 业务, 用户终端设备激发的各种信令流程, 包括用户注册、建立个人和各种组通话、增加用户、通话状态刷新、用户平台自适应、离开/结束通话和重新加入通话。
- 网络设备回应终端设备的信令流程和网络初始化时的信令流, 包括: 用户邀请、通话请求通知、结束通话和用户平台自适应。
- 用户终端设备上的 PoC 设置。
- 端到端的信令流。

#### 4. 用户群组/联系列表及免打扰管理子规范

用户群组/联系列表及免打扰管理子规范包含群组 and 列表服务器上的信号流, 信号参数和流程, 以及 Im 接口上的列表管理协议。该子规范分为如下部分:

- Im 接口管理的三个列表(联系列表、群组列表和通话限制列表)和对它们的操作管理。
- 免打扰标志和在 Im 接口上对它的操作。
- Im 接口上的列表管理协议。
- Im 和 Is 接口之间如何协同处理列表标识和寻址方式。
- 用户设备对列表管理的配置参数。图 5-26 描述了典型的一对一请求应答通话的信令流。这里作了一些假设, IMS 核心和 PoC 使用 SIP 通信; IMS 核心在该通话中主要作用是传发 SIP 信令。基于这些假设, 图中略去了 IMS 核心。该图描述了用户 A 请求用户 B 通话, 得到应答, 进行通话, 以及终止通话的过程。通话中说话权限的控制(双向箭头连线)可能包括多个权限控制信令。

在图 5-26 中, 实线表示 Is 和 If 接口上的 SIP 信令, 虚线和双向箭头表示 It 接口上的语音和控制流。

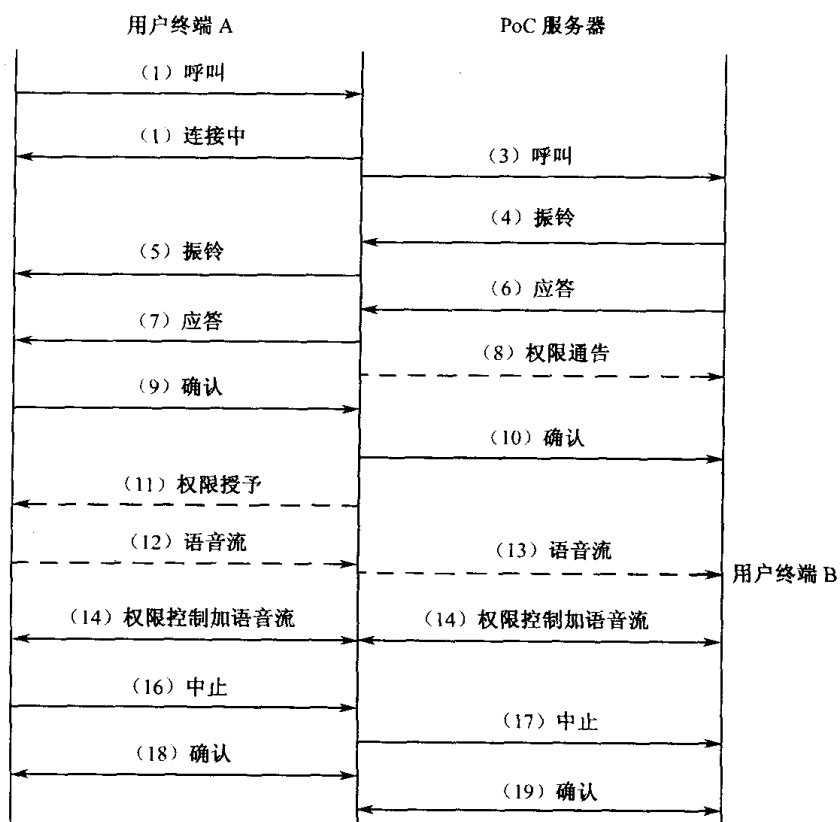


图 5-26 典型的一对一请求应答通话的信令流

## 5.8 SIP 压缩

UDVM (Universal Decompression Virtual Machine) 机制类似于 Java 虚拟机, 压缩器将字节上载到远端 UDVM, UDVM 执行字节码中的解压程序。UDVM 允许 SigComp 支持多种压缩算法。典型的字节码大小为 100~400 字节。UDVM 需要保存状态, 以使 SigComp 消息日后可使用此数据, 并且还需要有一定的安全机制。

### 5.8.1 参数扩展

#### 1. 新的参数

compression-param="comp"("sigcomp"/other-compression)

other-compression=token

via-compression="comp"EQUAL("sigcomp"/other-compression)

other-compression=token

#### 2. 实例

在图 5-27 所示的实例中, 消息(1)、(6)和(7)是经过压缩的。

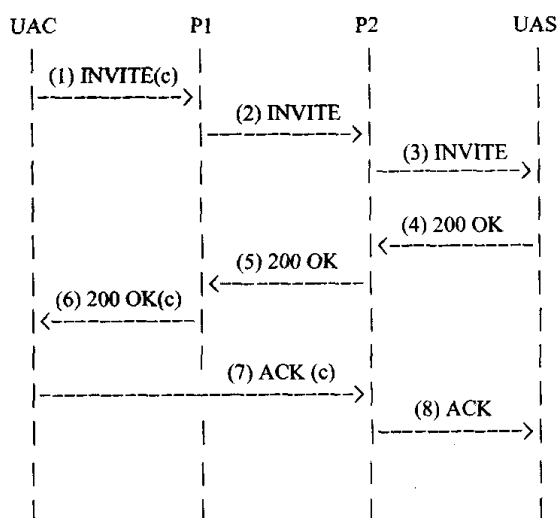


图 5-27 两个代理之间的 INVITE 事务

各消息分别如下:

#### (1) INVITE UAS

Via: UAC;comp=sigcomp

Route:P1;comp=sigcomp

Contact: UAC;comp=sigcomp

#### (2) INVITE UAS

Via: P1

Via: UAC;comp=sigcomp

Route:P2

Contact: UAC;comp=sigcomp

#### (3) INVITE UAS

Via: P2

Via: P1

Via: UAC;comp=sigcomp

Record-Route:P2

Contact: UAC;comp=sigcomp

(4) 200 OK

Via: P2

Via: P1

Via: UAC;comp=sigcomp

Record-Route:P2

Contact: UAS

(5) 200 OK

Via: P1

Via: UAC;comp=sigcomp

Record-Route:P2;comp=sigcomp

Contact: UAS

(6) 200 OK

Via: UAC;comp=sigcomp

Record-Route:P2;comp=sigcomp

Contact: UAS

(7) ACK UAS

Via: UAC;comp=sigcomp

Route:P2;comp=sigcomp

Contact: UAC;comp=sigcomp

(8) ACK UAS

Via: P2

Via: UAC;comp=sigcomp

Contact: UAC;comp=sigcomp

## 5.8.2 压缩字典

### 1. 压缩字典及其编码

0000	0d0a 5265 6a65 6374 2d43 6f6e 7461 6374	..Reject-Contact
0010	3a20 0d0a 4572 726f 722d 496e 666f 3a20	: ..Error-Info:
0020	0d0a 5469 6d65 7374 616d 703a 200d 0a43	..Timestamp: ..C
0030	616c 6c2d 496e 666f 3a20 0d0a 5265 706c	all-Info: ..Repl
0040	792d 546f 3a20 0d0a 5761 726e 696e 673a	y-To: ..Warning:
0050	200d 0a53 7562 6a65 6374 3a20 3b68 616e	..Subject: ;han
0060	646c 696e 673d 696d 6167 653b 7075 7270	dling=image;purp
0070	6f73 653d 3b63 6175 7365 3d3b 7465 7874	ose=;cause=;text
0080	3d63 6172 6433 3030 204d 756c 7469 706c	=card300 Multipl



0090	6520	4368	6f69	6365	736d	696d	6573	7361	e Choicesmimessa
00A0	6765	2f73	6970	6672	6167	3430	3720	5072	ge/sipfrag407 Pr
00B0	6f78	7920	4175	7468	656e	7469	6361	7469	oxy Authenticati
00C0	6f6e	2052	6571	7569	7265	6469	6765	7374	on Requiredigest
00D0	2d69	6e74	6567	7269	7479	3438	3420	4164	-integrity484 Ad
00E0	6472	6573	7320	496e	636f	6d70	6c65	7465	dress Incomplete
00F0	6c65	7068	6f6e	652d	6576	656e	7473	3439	lephone-events49
0100	3420	5365	6375	7269	7479	2041	6772	6565	4 Security Agree
0110	6d65	6e74	2052	6571	7569	7265	6465	6163	ment Requiredeac
0120	7469	7661	7465	6434	3831	2043	616c	6c2f	tivated481 Call/
0130	5472	616e	7361	6374	696f	6e20	446f	6573	Transaction Does
0140	204e	6f74	2045	7869	7374	616c	653d	3530	Not Existale=50
0150	3020	5365	7276	6572	2049	6e74	6572	6e61	0 Server Interna
0160	6c20	4572	726f	726f	6275	7374	2d73	6f72	1 Errorobust-sor
0170	7469	6e67	3d34	3136	2055	6e73	7570	706f	ting=416 Unsuppo
0180	7274	6564	2055	5249	2053	6368	656d	6572	rted URI Schemer
0190	6765	6e63	7934	3135	2055	6e73	7570	706f	gency415 Unsuppo
01A0	7274	6564	204d	6564	6961	2054	7970	656e	rted Media Typen
01B0	6469	6e67	3438	3820	4e6f	7420	4163	6365	ding488 Not Acce
01C0	7074	6162	6c65	2048	6572	656a	6563	7465	ptable Herejecte
01D0	6434	3233	2049	6e74	6572	7661	6c20	546f	d423 Interval To
01E0	6f20	4272	6965	6672	6f6d	2d74	6167	512e	o Brieffrom-tagQ.
01F0	3835	3035	2056	6572	7369	6f6e	204e	6f74	8505 Version Not
0200	2053	7570	706f	7274	6564	3430	3320	466f	Supported403 Fo
0210	7262	6964	6465	6e6f	6e2d	7572	6765	6e74	rbiddenon-urgent
0220	3432	3920	5072	6f76	6964	6520	5265	6665	429 Provide Refe
0230	7272	6f72	2049	6465	6e74	6974	7934	3230	rror Identity420
0240	2042	6164	2045	7874	656e	7369	6f6e	6f72	Bad Extensionor
0250	6573	6f75	7263	650d	0a61	3d6b	6579	2d6d	esource..a=key-m
0260	676d	743a	6d69	6b65	794f	5054	494f	4e53	gmt:mikeyOPTIONS
0270	204c	616e	6775	6167	653a	2035	3034	2053	Language: 504 S
0280	6572	7665	7220	5469	6d65	2d6f	7574	6f2d	erver Time-outo-
0290	7461	670d	0a41	7574	6865	6e74	6963	6174	tag..Authenticat
02A0	696f	6e2d	496e	666f	3a20	4465	6320	3338	ion-Info: Dec 38
02B0	3020	416c	7465	726e	6174	6976	6520	5365	0 Alternative Se
02C0	7276	6963	6535	3033	2053	6572	7669	6365	rvice503 Service
02D0	2055	6e61	7661	696c	6162	6c65	3432	3120	Unavailable421
02E0	4578	7465	6e73	696f	6e20	5265	7175	6972	Extension Requir
02F0	6564	3430	3520	4d65	7468	6f64	204e	6f74	ed405 Method Not
0300	2041	6c6c	6f77	6564	3438	3720	5265	7175	Allowed487 Requ

0310 6573 7420 5465 726d 696e 6174 6564 6175 est Terminatedau  
 0320 7468 2d69 6e74 6572 6c65 6176 696e 673d th-interleaving=  
 0330 0d0a 6d3d 6170 706c 6963 6174 696f 6e20 ..m=application  
 0340 4175 6720 3531 3320 4d65 7373 6167 6520 Aug 513 Message  
 0350 546f 6f20 4c61 7267 6536 3837 2044 6961 Too Large687 Dia  
 0360 6c6f 6720 5465 726d 696e 6174 6564 3330 log Terminated30  
 0370 3220 4d6f 7665 6420 5465 6d70 6f72 6172 2 Moved Temporar  
 0380 696c 7933 3031 204d 6f76 6564 2050 6572 ily301 Moved Per  
 0390 6d61 6e65 6e74 6c79 6d75 6c74 6970 6172 manentlymultipar  
 03A0 742f 7369 676e 6564 0d0a 5265 7472 792d t/signed..Retry-  
 03B0 4166 7465 723a 2047 4d54 6875 2c20 3430 After: GMThu, 40  
 03C0 3220 5061 796d 656e 7420 5265 7175 6972 2 Payment Requir  
 03D0 6564 0d0a 613d 6f72 6965 6e74 3a6c 616e ed..a=orient:lan  
 03E0 6473 6361 7065 3430 3020 4261 6420 5265 dscape400 Bad Re  
 03F0 7175 6573 7472 7565 3439 3120 5265 7175 questrue491 Requ  
 0400 6573 7420 5065 6e64 696e 6735 3031 204e est Pending501 N  
 0410 6f74 2049 6d70 6c65 6d65 6e74 6564 3430 ot Implemented40  
 0420 3620 4e6f 7420 4163 6365 7074 6162 6c65 6 Not Acceptable  
 0430 3630 3620 4e6f 7420 4163 6365 7074 6162 606 Not Acceptab  
 0440 6c65 0d0a 613d 7479 7065 3a62 726f 6164 le..a=type:broad  
 0450 6361 7374 6f6e 6534 3933 2055 6e64 6563 castone493 Undec  
 0460 6970 6865 7261 626c 650d 0a4d 494d 452d ipherable..MIME-  
 0470 5665 7273 696f 6e3a 204d 6179 2034 3832 Version: May 482  
 0480 204c 6f6f 7020 4465 7465 6374 6564 0d0a Loop Detected..  
 0490 4f72 6761 6e69 7a61 7469 6f6e 3a20 4a75 Organization: Ju  
 04A0 6e20 6d6f 6465 2d63 6861 6e67 652d 6e65 n mode-change-ne  
 04B0 6967 6862 6f72 3d63 7269 7469 6361 6c65 ighbor=criticale  
 04C0 7274 6370 2d66 6234 3839 2042 6164 2045 rtcp-fb489 Bad E  
 04D0 7665 6e74 6c73 0d0a 556e 7375 7070 6f72 ventls..Unsuppor  
 04E0 7465 643a 204a 616e 2035 3032 2042 6164 ted: Jan 502 Bad  
 04F0 2047 6174 6577 6179 6d6f 6465 2d63 6861 Gatewaymode-cha  
 0500 6e67 652d 7065 7269 6f64 3d0d 0a61 3d6f nge-period=..a=o  
 0510 7269 656e 743a 7365 6173 6361 7065 0d0a rient:seascape..  
 0520 613d 7479 7065 3a6d 6f64 6572 6174 6564 a=type:moderated  
 0530 3430 3420 4e6f 7420 466f 756e 6433 3035 404 Not Found305  
 0540 2055 7365 2050 726f 7879 0d0a 613d 7479 Use Proxy..a=ty  
 0550 7065 3a72 6563 766f 6e6c 790d 0a61 3d74 pe:recvonly..a=t  
 0560 7970 653a 6d65 6574 696e 670d 0a6b 3d70 ype:meeting..k=p  
 0570 726f 6d70 743a 0d0a 5265 6665 7272 6564 rompt:...Referred  
 0580 2d42 793a 200d 0a49 6e2d 5265 706c 792d -By: ..In-Reply-

```

0590 546f 3a20 5452 5545 6e63 6f64 696e 673a To: TRUEncoding:
05A0 2031 3832 2051 7565 7565 6441 7574 6865 182 QueuedAuthe
05B0 6e74 6963 6174 653a 200d 0a55 7365 722d nticate: ..User-
05C0 4167 656e 743a 200d 0a61 3d66 7261 6d65 Agent: ..a=frame
05D0 7261 7465 3a0d 0a41 6c65 7274 2d49 6e66 rate:...Alert-Inf
05E0 6f3a 2043 414e 4345 4c20 0d0a 613d 6d61 o: CANCEL ..a=ma
05F0 7870 7469 6d65 3a3b 7265 7472 792d 6166 xptime;;retry-af
0600 7465 723d 7561 6368 616e 6e65 6c73 3d34 ter=uachannels=4
0610 3130 2047 6f6e 650d 0a52 6566 6572 2d54 10 Gone..Refer-T
0620 6f3a 200d 0a50 7269 6f72 6974 793a 200d o: ..Priority: .
0630 0a6d 3d63 6f6e 7472 6f6c 200d 0a61 3d71 .m=control ..a=q
0640 7561 6c69 7479 3a0d 0a61 3d73 6470 6c61 uality:...a=sdpla
0650 6e67 3a0d 0a61 3d63 6861 7273 6574 3a0d ng:...a=charset:.
0660 0a52 6570 6c61 6365 733a 2052 4546 4552 .Replaces: REFER
0670 2069 7073 6563 2d69 6b65 3b74 7261 6e73 ipsec-ike;trans
0680 706f 7274 3d0d 0a61 3d6b 6579 7764 733a port=..a=keywds:
0690 0d0a 6b3d 6261 7365 3634 3a3b 7265 6672 ..k=base64;;refr
06A0 6573 6865 723d 0d0a 613d 7074 696d 653a esher=..a=ptime:
06B0 0d0a 6b3d 636c 6561 723a 3b72 6563 6569 ..k=clear;;recei
06C0 7665 643d 3b64 7572 6174 696f 6e3d 0d0a ved=;duration=..
06D0 4163 6365 7074 3a20 0d0a 613d 6772 6f75 Accept: ..a=grou
06E0 703a 4641 4c53 453a 2049 4e46 4f20 0d0a p:FALSE: INFO ..
06F0 4163 6365 7074 2d0d 0a61 3d6c 616e 673a Accept-..a=lang:
0700 0d0a 6d3d 6461 7461 206d 6f64 652d 7365 ..m=data mode-se
0710 743d 0d0a 613d 746f 6f6c 3a54 4c53 756e t=..a=tool:TLSun
0720 2c20 0d0a 4461 7465 3a20 0d0a 613d 6361 , ..Date: ..a=ca
0730 743a 0d0a 6b3d 7572 693a 0d0a 5072 6f78 t:...k=uri:...Prox
0740 792d 3b72 6561 736f 6e3d 3b6d 6574 686f y-;reason=;metho
0750 643d 0d0a 613d 6d69 643a 3b6d 6164 6472 d=..a=mid;;maddr
0760 3d6f 7061 7175 653d 0d0a 4d69 6e2d 3b61 =opaque=..Min-;a
0770 6c67 3d4d 6f6e 2c20 5475 652c 2057 6564 lg=Mon, Tue, Wed
0780 2c20 4672 692c 2053 6174 2c20 3b74 746c , Fri, Sat, ;ttl
0790 3d61 7574 733d 0d0a 723d 0d0a 7a3d 0d0a =auts=..r=..z=..
07A0 653d 3b69 643d 0d0a 693d 6372 633d 0d0a e=;id=..i=crc=..
07B0 753d 3b71 3d75 6173 3431 3420 5265 7175 u=;q=uas414 Requ
07C0 6573 742d 5552 4920 546f 6f20 4c6f 6e67 est-URI Too Long
07D0 6976 6575 7072 6976 6163 7975 6470 7265 iveupprivacyudpre
07E0 6665 7236 3030 2042 7573 7920 4576 6572 fer600 Busy Ever
07F0 7977 6865 7265 7175 6972 6564 3438 3020 ywherequired480
0800 5465 6d70 6f72 6172 696c 7920 556e 6176 Temporarily Unav

```

0810 6169 6c61 626c 650d 0a61 3d74 7970 653a aivable..a=type:  
 0820 482e 3333 3230 3220 4163 6365 7074 6564 H.33202 Accepted  
 0830 0d0a 5365 7373 696f 6e2d 4578 7069 7265 ..Session-Expire  
 0840 733a 200d 0a53 7562 7363 7269 7074 696f s: ..Subscriptio  
 0850 6e2d 5374 6174 653a 204e 6f76 200d 0a53 n-State: Nov ..S  
 0860 6572 7669 6365 2d52 6f75 7465 3a20 5365 ervice-Route: Se  
 0870 7020 0d0a 416c 6c6f 772d 4576 656e 7473 p ..Allow-Events  
 0880 3a20 4665 6220 0d0a 613d 696e 6163 7469 : Feb ..a=inacti  
 0890 7665 5254 502f 5341 5650 2052 5450 2f41 veRTP/SAVP RTP/A  
 08A0 5650 4620 416e 6f6e 796d 6f75 7369 7073 VPF Anonymousips  
 08B0 3a0d 0a61 3d74 7970 653a 7465 7374 656c :..a=type:testel  
 08C0 3a4d 4553 5341 4745 200d 0a61 3d72 6563 :MESSAGE ..a=rec  
 08D0 766f 6e6c 790d 0a61 3d73 656e 646f 6e6c vonly..a=sendonl  
 08E0 790d 0a63 3d49 4e20 4950 3420 0d0a 5265 y..c=IN IP4 ..Re  
 08F0 6173 6f6e 3a20 0d0a 416c 6c6f 773a 200d ason: ..Allow: .  
 0900 0a45 7665 6e74 3a20 0d0a 5061 7468 3a20 .Event: ..Path:  
 0910 3b75 7365 723d 0d0a 623d 4153 2043 5420 ;user=..b=AS CT  
 0920 0d0a 5757 572d 4175 7468 656e 7469 6361 ..WWW-Authentica  
 0930 7465 3a20 4469 6765 7374 200d 0a61 3d73 te: Digest ..a=s  
 0940 656e 6472 6563 7669 6465 6f63 7465 742d endrecvideoctet-  
 0950 616c 6967 6e3d 6170 706c 6963 6174 696f align=applicatio  
 0960 6e2f 7364 7061 7468 6561 6465 7273 7061 n/sdpatheaderspa  
 0970 7574 683d 0d0a 613d 6f72 6965 6e74 3a70 uth=..a=orient:p  
 0980 6f72 7472 6169 7469 6d65 6f75 7474 722d ortraitimeouttr-  
 0990 696e 7469 636f 6e63 3d34 3833 2054 6f6f inticonc=483 Too  
 09A0 204d 616e 7920 486f 7073 6c69 6e66 6f70 Many Hopslinfo  
 09B0 7469 6f6e 616c 676f 7269 7468 6d3d 3630 tionalgorithm=60  
 09C0 3420 446f 6573 204e 6f74 2045 7869 7374 4 Does Not Exist  
 09D0 2041 6e79 7768 6572 6573 706f 6e73 653d Anywheresponse=  
 09E0 0d0a 0d0a 5265 7175 6573 742d 4469 7370 ....Request-Disp  
 09F0 6f73 6974 696f 6e3a 204d 4435 3830 2050 osition: MD580 P  
 0A00 7265 636f 6e64 6974 696f 6e20 4661 696c recondition Fail  
 0A10 7572 6570 6c61 6365 7334 3232 2053 6573 ureplaces422 Ses  
 0A20 7369 6f6e 2049 6e74 6572 7661 6c20 546f sion Interval To  
 0A30 6f20 536d 616c 6c6f 6361 6c31 3831 2043 o Smallocal181 C  
 0A40 616c 6c20 4973 2042 6569 6e67 2046 6f72 all Is Being For  
 0A50 7761 7264 6564 6f6d 6169 6e3d 6661 696c wardedomain=fail  
 0A60 7572 656e 6465 7265 616c 6d3d 5355 4253 urenderealm=SUBS  
 0A70 4352 4942 4520 7072 6563 6f6e 6469 7469 CRIBE preconditi  
 0A80 6f6e 6f72 6d61 6c69 7073 6563 2d6d 616e onormalipsec-man

```

0A90 6461 746f 7279 3431 3320 5265 7175 6573 datory413 Reques
0AA0 7420 456e 7469 7479 2054 6f6f 204c 6172 t Entity Too Lar
0AB0 6765 3265 3138 3320 5365 7373 696f 6e20 ge2e183 Session
0AC0 5072 6f67 7265 7373 6374 7034 3836 2042 Progressctp486 B
0AD0 7573 7920 4865 7265 6d6f 7465 726d 696e usy Heremotermin
0AE0 6174 6564 414b 4176 312d 4d44 352d 7365 atedAKAv1-MD5-se
0AF0 7373 696f 6e6f 6e65 0d0a 4175 7468 6f72 ssionone..Author
0B00 697a 6174 696f 6e3a 2036 3033 2044 6563 ization: 603 Dec
0B10 6c69 6e65 7874 6e6f 6e63 653d 3438 3520 linextnonce=485
0B20 416d 6269 6775 6f75 7365 726e 616d 653d Ambiguousexname=
0B30 6175 6469 6f0d 0a43 6f6e 7465 6e74 2d54 audio..Content-T
0B40 7970 653a 204d 6172 200d 0a52 6563 6f72 ype: Mar ..Recor
0B50 642d 526f 7574 653a 204a 756c 2034 3031 d-Route: Jul 401
0B60 2055 6e61 7574 686f 7269 7a65 640d 0a52 Unauthorized..R
0B70 6571 7569 7265 3a20 0d0a 743d 3020 302e equire: ..t=0 0.
0B80 302e 302e 300d 0a53 6572 7665 723a 2052 0.0.0..Server: R
0B90 4547 4953 5445 5220 0d0a 633d 494e 2049 EGISTER ..c=IN I
0BA0 5036 2031 3830 2052 696e 6769 6e67 3130 P6 180 Ringing10
0BB0 3020 5472 7969 6e67 763d 300d 0a6f 3d55 0 Tryingv=0..o=U
0BC0 5044 4154 4520 4e4f 5449 4659 200d 0a53 PDATE NOTIFY ..S
0BD0 7570 706f 7274 6564 3a20 756e 6b6e 6f77 upported: unknow
0BE0 6e41 4d52 5450 2f41 5650 200d 0a50 7269 nAMRTP/AVP ..Pri
0BF0 7661 6379 3a20 0d0a 5365 6375 7269 7479 vacy: ..Security
0C00 2d0d 0a45 7870 6972 6573 3a20 0d0a 613d -..Expires: ..a=
0C10 7274 706d 6170 3a0d 0a6d 3d76 6964 656f rtpmap:...m=video
0C20 200d 0a6d 3d61 7564 696f 200d 0a73 3d20 ...m=audio ..s=
0C30 6661 6c73 650d 0a61 3d63 6f6e 663a 3b65 false..a=conf::e
0C40 7870 6972 6573 3d0d 0a52 6f75 7465 3a20 xpires=..Route:
0C50 0d0a 613d 666d 7470 3a0d 0a61 3d63 7572 ..a=fmtp:...a=cur
0C60 723a 436c 6965 6e74 3a20 5665 7269 6679 r:Client: Verify
0C70 3a20 0d0a 613d 6465 733a 0d0a 5241 636b : ..a=des...Rack
0C80 3a20 0d0a 5253 6571 3a20 4259 4520 636e : ..RSeq: BYE cn
0C90 6f6e 6365 3d31 3030 7265 6c75 7269 3d71 once=100reluri=q
0CA0 6f70 3d54 4350 5544 5071 6f73 786d 6c3b op=TCPUDPqosxml;
0CB0 6c72 0d0a 5669 613a 2053 4950 2f32 2e30 lr..Via: SIP/2.0
0CC0 2f54 4350 2034 3038 2052 6571 7565 7374 /TCP 408 Request
0CD0 2054 696d 656f 7574 696d 6572 7073 6970 Timeoutimerpsip
0CE0 3a0d 0a43 6f6e 7465 6e74 2d4c 656e 6774 :..Content-Lengt
0CF0 683a 204f 6374 200d 0a56 6961 3a20 5349 h: Oct ..Via: SI
0D00 502f 322e 302f 5544 5020 3b63 6f6d 703d P/2.0/UDP ;comp=

```

```

OD10 7369 6763 6f6d 7072 6f62 6174 696f 6e61 sigcomprobationa
OD20 636b 3b62 7261 6e63 683d 7a39 6847 3462 ck;branch=z9hG4b
OD30 4b0d 0a4d 6178 2d46 6f72 7761 7264 733a K..Max-Forwards:
OD40 2041 7072 2053 4354 5052 4143 4b20 494e Apr SCTPRACK IN
OD50 5649 5445 200d 0a43 616c 6c2d 4944 3a20 VITE ..Call-ID:
OD60 0d0a 436f 6e74 6163 743a 2032 3030 204f ..Contact: 200 0
OD70 4b0d 0a46 726f 6d3a 200d 0a43 5365 713a K..From: ..CSeq:
OD80 200d 0a54 6f3a 203b 7461 673d 0410 dd10 ..To: ;tag=...
OD90 1131 0d11 0a07 10b9 0c10 fe12 10e1 0611 .l.....
ODA0 4e07 114e 0311 4a04 114a 0710 b208 1179 N..N..J..J....y
ODB0 0611 810f 1122 0b11 5506 116b 0b11 6013 .....".U..k..`
ODC0 10b2 0811 7105 1187 1310 f709 0e8d 080d ....q.....
ODD0 ae0c 10b9 0710 8e03 0d96 0310 8a04 108a .....
ODE0 090d d70a 0f12 080f 8f09 0f8f 080d 6c06 .....l.
ODF0 0e66 090e 6c0a 0e6c 060f c607 0fc6 0511 .f..l..l.....
OE00 4806 1148 060f bf07 0fbf 070e 5506 0f16 H..H.....U...
OE10 040e f403 0eb1 0310 a609 1050 0310 a30a .....P....
OE20 0db4 050e 3606 0ed6 030d f911 0ef8 040c ....6.....
OE30 d908 0eea 0409 5303 0a4b 040e e410 0f35 .....S..K.....5
OE40 090e e408 0d3f 030f e10b 1001 0310 ac06 .....?.....
OE50 1095 0c0e 760b 0feb 0a0f ae05 102b 0410 ....v.....+.
OE60 2b08 107a 100f 4907 0fb8 0910 3e0b 100c +..z..I.....>...
OE70 070f 780b 0f6d 0910 4708 1082 0b0f f608 ..x..m..G.....
OE80 1062 080f 8708 106a 040f 780d 0fcd 080d .b.....j..x.....
OE90 ae10 0f5d 0b0f 9814 0d20 1b0d 2004 0de0 ...]. .... ..
OEAO 140e b40b 0fa3 0b07 340f 0d56 040e f403 .....4..V....
OEB0 10af 070d 3409 0f27 0410 9b04 109f 0910 ....4..'.....
OECO 5908 1072 0910 350a 1021 0a10 1708 0fe3 Y..r..5..!.....
OED0 0310 a905 0cac 040c bd07 0cc1 080c c109 .....
OEE0 0cf6 100c 720c 0c86 040d 640c 0cd5 090c ....r.....d.....
OEF0 ff1b 0bfc 110c 5d13 0c30 090c a40c 0c24 .....].0....$
OF00 0c0d 3b03 0d1a 030d 1d16 0c43 090c 9209 ..;.....C....
OF10 0c9b 0d0e cb04 0d16 060d 1005 04f2 0b0c .....
OF20 e105 0bde 0a0c ec13 0be3 070b d408 0d08 .....
OF30 0c0c c909 0c3a 040a e50c 0a23 080b 3a0e .....:.....#.:.
OF40 09ab 0f0e fa09 0f6f 0c0a 170f 0976 0c0a .....o.....v..
OF50 5f17 0de2 0f07 a80a 0f85 0f08 d60e 09b9 _.....
OF60 0b0a 7a03 0bdb 0308 c104 0ec7 0308 d302 ..z.....
OF70 048d 080b 4a05 0b8c 070b 6106 0548 0407 ....J.....a..H..
OF80 f405 1030 0407 1e08 071e 050b 9110 04ca ...0.....

```

```

0F90 090a 7109 0e87 0504 9805 0b6e 0b04 9b0f ..q.....n....
0FA0 049b 0704 9b03 04a3 0704 a310 0798 0907 .....
0FB0 9805 0b73 050b 7805 0b7d 0507 b905 0b82 ...s..x..}.....
0FC0 050b 8705 0b1d 0508 e405 0c81 050f 4405 .....D.
0FD0 1140 0508 7805 089d 050f 5805 073f 050c .@..x.....X..?..
0FE0 6d05 10f2 050c 5805 06a9 0407 b609 058c m.....X.....
0FF0 0606 1a06 0e81 0a06 160a 0ac4 070b 5a0a .....Z.
1000 0aba 030b 1b04 1145 060c 8c07 05ad 0a0e .....E.....
1010 da08 0b42 0d09 f70b 051c 0911 1608 05c9 ...B.....
1020 070d 8606 0bcf 0a06 4d04 0ba2 0606 8d08 .....M.....
1030 05e6 080e 110b 0a9b 030a 0403 0bb5 0510 .....
1040 d704 0994 050a e203 0bb2 060d 6704 0d11 .....g...
1050 0808 b71b 0e3b 0a09 a114 0485 1507 8315 .....;.....
1060 076e 0d09 3d17 06ae 0f07 e614 07be 0d06 .n..=.....
1070 0a0d 0930 1606 f212 081e 2104 aa13 10c5 ...0.....!.....
1080 080a 0f1c 0e96 180b b81a 0595 1a05 7511 .....u.
1090 063d 1606 dc1e 0e19 1605 d11d 0620 2305 .=. .... #.
10A0 2711 087d 110d 9916 04da 0d0f 1c16 0708 '...}.....
10B0 1705 b40d 08c7 1307 f812 0857 1f04 fe19 .....W....
10C0 054e 1308 0b0f 08e9 1706 c513 067b 1905 .N.....{..
10D0 f115 0744 180d fb0b 0f09 1b0d be12 0830 ...D.....0
10E0 1507 5904 0ba6 040b ae04 0b9e 040b 9604 ..Y.....
10F0 0b9a 0a0a b00b 0a90 080b 320b 096b 080b .....2..k..
1100 2a0b 0a85 090b 120a 0aa6 0d09 ea13 0d74 *. .... t
1110 1407 d213 090b 1208 4210 095b 1209 1e0d .....B..[....
1120 0cb1 0e0c 1711 094a 0c0a 530c 0a47 090a .....J..S..G..
1130 f70e 09c7 0c0a 3b07 0669 0806 6906 09e3 .....;...i..i...
1140 080b 520a 0ad8 1206 570d 0657 0709 e304 ..R....W..W....
1150 0ae9 1007 3009 0b00 0c0a 2f05 0ae9 050a ....0...../.....
1160 6b06 0a6b 0a0a ce09 0aee 030b db07 0f7e k..k.....~
1170 0a09 970a 0671 0e09 d517 0693 070e 5c07 .....q.....\.
1180 0fda 0a0f 350d 0dec 0a09 970a 0671 080b ....5.....q..
1190 220f 0985 060b 680c 0d4a 090b 0913 08f8 ".....h..J.....
11A0 1508 a204 0baa 0f05 660d 0723 090a 060b .....f..#....
11B0 0d4a 0f04 ee06 04f8 0409 2b04 0853 0708 .J.....+.S..
11C0 c003 111f 0411 1e07 0d8c 0307 3404 10db .....4...
11D0 0307 3603 0da9 0d04 200b 0451 0c04 3a04 ..6.....Q...:
11E0 0bb8 040c 2404 0595 0404 7c04 0575 0404 ....$. ....|..u..
11F0 8504 096b 0406 3d06 047b 0406 dc04 0783 ...k..=..{.....
1200 040e 1912 0400 1008 8e10 0869 0e04 120d .....i....

```

```

1210 042d 0310 b904 05d1 0407 6e04 0620 0704 .-.....n. . .
1220 7404 0bfc 0a04 5c04 0527 0409 3d04 087d t.....\..'..=.}
1230 040f ae04 0d99 0406 ae04 04da 0904 0908 .....
1240 1122 040f 1c04 07e6 040e cb05 08bd 0407 ."......
1250 0804 0fa3 0406 5704 05b4 040f 5d04 08c7 .....W.....]...
1260 080b f404 07f8 0407 3004 07be 0408 5705 .....0.....W.
1270 0d46 0404 fe04 060a 0405 4e04 0e3b 0408 .F.....N.;...
1280 0b04 0930 0408 e905 05ee 0406 c504 06f2 ...0.....
1290 0406 7b04 09a1 0405 f104 081e 0407 4404 ..{.....D.
12A0 0bdd 040d fb04 04aa 040b e307 0eee 040f .....
12B0 0904 0eb4 040d be04 10c5 0408 3005 0f30 .....0..0
12C0 0407 5904 0a0f 060e 6104 0481 040d ab04 ..Y.....a.....
12D0 0d93 0411 6b04 0e96 0504 6609 046b 0b04 ....k.....f..k..
12E0 4604 0ce1 F...

```

## 2. 消息及其在压缩字典中的位置

SIP 相关消息在压缩字典中的位置如表 5-17 所示, 其中优先级值越小, 表明优先级越高, 出现机率越大。SDP 相关消息在压缩字典中的位置如表 5-18 所示。

表 5-17 SIP 消息在压缩字典中的位置

String	Pr	Off	Len
"sip:"	1	0CDD	0004
"sips:"	3	08AC	0005
"tel:"	3	08BD	0004
"SIP/2.0"	1	0CB9	0007
"SIP/2.0/UDP "	1	0CFE	000C
"SIP/2.0/TCP"	2	0CB9	000C
"INVITE"	1	0D4E	0006
"INVITE"	1	0D4E	0007
"ACK"	1	0D4A	0003
"ACK"	1	0D4A	0004
"OPTIONS"	4	0269	0007
"OPTIONS"	4	0269	0008
"BYE"	2	0C8A	0003
"BYE"	2	0C8A	0004
"CANCEL "	4	05E3	0006
"CANCEL "	4	05E3	0007
"REGISTER"	2	0B8F	0008
"REGISTER"	2	0B8F	0009
"INFO"	4	06E9	0004
"INFO"	4	06E9	0005
"SUBSCRIBE"	2	0A6C	0009
"SUBSCRIBE"	2	0A6C	000A
"NOTIFY"	2	0BC6	0006



续表

String	Pr	Off	Len
"NOTIFY"	2	0BC6	0007
"PRACK"	2	0D48	0005
"PRACK"	2	0D48	0006
"UPDATE"	2	0BBF	0006
"UPDATE"	2	0BBF	0007
"REFER"	4	066B	0005
"MESSAGE"	3	08C1	0007
"MESSAGE"	3	08C1	0008
"[CRLF]Accept:"	4	06CE	000A
"[CRLF]Accept-"	4	06EE	0009
"Contact:"	5	0009	0009
"Encoding:"	4	0597	000A
"Language:"	4	0271	000A
"[CRLF]Alert-Info:"	4	05D5	000E
"[CRLF]Allow:"	3	08F6	0009
"[CRLF]Allow-Events:"	3	0872	0010
"[CRLF]Authentication-Info:"	4	0293	0017
"[CRLF]Authorization:"	2	0AF8	0011
"[CRLF]Call-ID:"	1	0D55	000B
"[CRLF]Call-Info:"	5	002D	000D
"[CRLF]Contact:"	1	0D60	000B
"[CRLF]Content-"	4	0B35	000A
"Disposition:"	4	09EC	000D
"Encoding:"	4	0597	000A
"Language:"	4	0271	000A
"[CRLF]Content-Length:"	1	0CE1	0012
"[CRLF]Content-Type:"	2	0B35	0010
"[CRLF]CSeq:"	1	0D79	0008
"[CRLF]Date:"	4	0722	0008
"[CRLF]Error-Info:"	5	0012	000E
"[CRLF]Event:"	3	08FF	0009
"[CRLF]Expires:"	2	0C01	000B
"[CRLF]From:"	1	0D71	0008
"[CRLF]In-Reply-To:"	4	0585	000F
"[CRLF]Max-Forwards:"	1	0D31	0010
"[CRLF]Min-"	4	0768	0006
"Expires:"	4	083A	0009
"SE:"	4	06E5	0004
"[CRLF]MIME-Version:"	5	0469	0010
"[CRLF]Organization:"	5	048E	0010
"[CRLF]Path:"	3	0908	0008
"[CRLF]Priority:"	4	0623	000C
"[CRLF]Privacy:"	2	0BEB	000B
"[CRLF]Proxy-"	4	073A	0008

续表

String	Pr	Off	Len
"Authenticate:"	4	05AB	000E
"Authorization:"	4	0AFA	000F
"Require:"	4	0B6F	0009
"[CRLF]RAck:"	2	0C7A	0008
"[CRLF]Reason:"	3	08EC	000A
"[CRLF]Record-Route:"	2	0B49	0010
"[CRLF]Refer-To:"	4	0617	000C
"[CRLF]Referred-By:"	4	0576	000F
"[CRLF]Reject-Contact:"	5	0000	0012
"[CRLF]Replaces:"	4	065F	000C
"[CRLF]Reply-To:"	5	003A	000C
"[CRLF]Request-Disposition:"	4	09E2	0017
"[CRLF]Require:"	2	0B6D	000B
"[CRLF]Retry-After:"	4	03A8	000F
"[CRLF]Route:"	2	0C47	0009
"[CRLF]RSeq:"	2	0C82	0008
"[CRLF]Security-"	2	0BF6	000B
"Client:"	2	0C62	0008
"Server:"	2	0B87	0008
"Verify:"	2	0C6A	0008
"[CRLF]Server:"	4	0B85	000A
"[CRLF]Service-Route:"	3	085D	0011
"[CRLF]Session-Expires:"	3	0830	0013
"[CRLF]Subject:"	5	0051	000B
"[CRLF]Subscription-State:"	3	0843	0016
"[CRLF]Supported:"	2	0BCD	000D
"[CRLF]Timestamp:"	5	0020	000D
"[CRLF]To:"	1	0D81	0006
"[CRLF]Unsupported:"	4	04D6	000F
"[CRLF]User-Agent:"	4	05B9	000E
"[CRLF]Via:"	1	0CB2	0007
"[CRLF]Via:SIP/2.0/UDP"	1	0CF7	0013
"[CRLF]Via:SIP/2.0/TCP"	1	0CB2	0013
"[CRLF]Warning:"	5	0046	000B
"[CRLF]WWW-Authenticate:"	2	0920	0014
"[CRLF]WWW-Authenticate:Digest"	2	0920	001B
"[CRLF][CRLF]"	2	09E0	0004
";transport="	4	067A	000B
"udp"	4	07DB	0003
"tcp"	4	04C1	0003
"sctp"	4	0AC7	0004
"tls"	4	04D3	0003
";user="	3	0910	0006
"phone"	3	00F2	0005

续表

String	Pr	Off	Len
"ip"	4	008D	0002
";method="	4	074A	0008
";ttl="	4	078C	0005
";lr"	2	0CAF	0003
"Digest"	2	0934	0007
"username="	2	0B27	0009
"uri="	2	0C9B	0004
"qop="	2	0C9F	0004
"cnonce="	2	0C8E	0007
"nc="	2	0996	0003
"response="	2	09D7	0009
"nextnonce="	2	0B12	000A
"rspauth="	2	096C	0008
"realm="	2	0A66	0006
"domain="	2	0A55	0007
"nonce="	2	0B16	0006
"opaque="	4	0761	0007
"stale="	4	0148	0006
"true"	4	03F4	0004
"false"	4	0C30	0005
"algorithm="	2	09B4	000A
"MD5"	2	09F9	0003
"MD5-sess"	2	0AEA	0008
"auth"	4	031E	0004
"auth-int"	4	031E	0008
"AKAv"	2	0AE4	0004
"AKAv1-MD5"	2	0AE4	0009
"auts="	4	0791	0005
"digest-integrity"	4	00CA	0010
"ipsec-ike"	4	0671	0009
"ipsec-man"	4	0A87	0009
"smime"	4	0098	0005
";alg="	4	076E	0005
";purpose="	5	006B	0009
"icon"	5	0993	0004
"info"	5	09AB	0004
"card"	5	0081	0004
";expires="	2	0C3E	0009
"render"	5	0A61	0006
"session"	5	0AEE	0007
"alert"	5	04BD	0005
";handling="	5	005C	000A
"optional"	2	09AE	0008
"required"	5	07F4	0008

续表

String	Pr	Off	Len
"text"	5	007C	0004
"image"	5	0066	0005
"audio"	5	0B30	0005
"video"	5	0946	0005
"application"	2	0334	000B
"application/sdp"	2	0956	000F
"message/sip"	4	009B	000B
"message/sipfrag"	4	009B	000F
"message"	4	009B	0007
"sip"	4	00A3	0003
"sipfrag"	4	00A3	0007
"multipart/signed"	4	0398	0010
"multipart"	4	0398	0009
"sdp"	2	064B	0003
"xml"	2	0CAC	0003
"Mon,"	4	0773	0005
"Tue,"	4	0778	0005
"Wed,"	4	077D	0005
"Thu,"	4	03B9	0005
"Fri,"	4	0782	0005
"Sat,"	4	0787	0005
"Sun,"	4	071D	0005
"Jan"	4	04E4	0005
"Feb"	4	0881	0005
"Mar"	4	0B44	0005
"Apr"	4	0D40	0005
"May"	4	0478	0005
"Jun"	4	049D	0005
"Jul"	40B58	0005	
"Aug"	4	033F	0005
"Sep"	4	086D	0005
"Oct"	4	0CF2	0005
"Nov"	4	0858	0005
"Dec"	4	02A9	0005
"GMT"	4	03B6	0004
";tag="	1	0D87	0005
"emergency"	4	018C	0009
"urgent"	4	021A	0006
"normal"	4	0A81	0006
"non-urgent"	4	0216	000A
";duration="	4	06C4	000A
";maddr="	4	075A	0007
";received="	4	06BA	000A
";branch="	5	0D22	0008

续表

String	Pr	Off	Len
";branch=z9hG4bK"	1	0D22	000F
"SIP"	5	0CB9	0003
"UDP"	2	0CA6	0003
"TCP"	2	0CA3	0003
"TLS"	4	071B	0003
"SCTP"	4	0D45	0004
"active"	4	088C	0006
"pending"	4	01AD	0007
"terminated"	4	0ADA	000A
";reason="	4	0742	0008
";retry-after="	4	05F7	000D
"deactivated"	4	011C	000B
"probation"	4	0D16	0009
"rejected"	4	01C9	0008
"timeout"	4	0986	0007
"giveup"	4	07CF	0006
"noresource"	4	024D	000A
";id="	4	07A2	0004
"100rel"	2	0C95	0006
"precondition"	2	0A76	000C
"refer"	3	07DE	0005
"to-tag"	4	028D	0006
"from-tag"	4	01E6	0008
"replaces"	4	0A11	0008
"Q.850"	5	01EE	0005
";cause="	5	0074	0007
";text="	5	007B	0006
"path"	3	0964	0004
";refresher="	4	069B	000B
"uac"	4	0604	0003
"uas"	4	07B5	0003
"timer"	4	0CD7	0005
"pref"	5	07DD	0004
"TRUE"	4	0594	0004
"FALSE"	4	06E2	0005
";q="	4	07B2	0003
";comp=sigcomp"	1	0D0A	000D
"privacy"	3	07D4	0007
"header"	4	0967	0006
"user"	4	0911	0004
"none"	2	0AF4	0004
"critical"	4	04B7	0008
"100"	5	0BAE	0004
"100Trying"	2	0BAE	000A

续表

String	Pr	Off	Len
"180"	5	0BA3	0004
"180Ringing"	2	0BA3	000B
"181"	5	0A3B	0004
"181CallsBeingForwarded"	4	0A3B	001B
"182"	5	05A1	0004
"182Queued"	4	05A1	000A
"183"	5	0AB4	0004
"183Session Progress"	2	0AB4	0014
"200"	5	0D6B	0004
"200 OK"	1	0D6B	0006
"202"	5	0824	0004
"202 Accepted"	3	0824	000C
"300"	5	0085	0004
"300 Multiple Choices"	4	0085	0014
"301"	5	0383	0004
"301 Moved Permanently"	4	0383	0015
"302"	5	036E	0004
"302 Moved Temporarily"	4	036E	0015
"305"	5	053D	0004
"305 Use Proxy"	4	053D	000D
"380"	5	02AE	0004
"380 Alternative Service"	4	02AE	0017
"400"	5	03E6	0004
"400 Bad Request"	4	03E6	000F
"401"	5	0B5D	0004
"401 Unauthorized"	2	0B5D	0010
"402"	5	03BE	0004
"402 Payment Required"	4	03BE	0014
"403"	5	020A	0004
"403 Forbidden"	4	020A	000D
"404"	5	0530	0004
"404 Not Found"	4	0530	000D
"405"	5	02F2	0004
"405 Method Not Allowed"	4	02F2	0016
"406"	5	041E	0004
"406 Not Acceptable"	4	041E	0012
"407"	5	00AA	0004
"407 Proxy Authentication Required"	4	00AA	0021
"408"	5	0CC5	0004
"408 Request Timeout"	4	0CC5	0013
"410"	5	060F	0004
"410 Gone"	4	060F	0008
"413"	5	0A96	0004
"413 Request Entity Too Large"	4	0A96	001C

续表

String	Pr	Off	Len
"414"	5	07B8	0004
"414 Request-URI Too Long"	4	07B8	0018
"415"	5	0195	0004
"415 Unsupported Media Type"	4	0195	001A
"416"	5	0175	0004
"416 Unsupported URI Scheme"	4	0175	001A
"420"	5	023D	0004
"420 Bad Extension"	4	023D	0011
"421"	5	02DC	0004
"421 Extension Required"	4	02DC	0016
"422"	5	0A19	0004
"422 Session Interval Too Small"	4	0A19	001E
"423"	5	01D1	0004
"423 Interval Too Brief"	4	01D1	0016
"429"	5	0220	0004
"429 Provide Referrer Identity"	4	0220	001D
"480"	5	07FC	0004
"480 Temporarily Unavailable"	3	07FC	001B
"481"	5	0127	0004
"481 Call/Transaction Does Not Exist"	4	0127	0023
"482"	5	047D	0004
"482 Loop Detected"	4	047D	0011
"483"	5	0999	0004
"483 Too Many Hops"	4	0999	0011
"484"	5	00DA	0004
"484 Address Incomplete"	4	00DA	0016
"485"	5	0B1C	0004
"485 Ambiguous"	4	0B1C	000D
"486"	5	0ACB	0004
"486 Busy Here"	3	0ACB	000D
"487"	5	0308	0004
"487 Request Terminated"	4	0308	0016
"488"	5	01B4	0004
"488 Not Acceptable Here"	4	01B4	0017
"489"	5	04C7	0004
"489 Bad Event"	4	04C7	000D
"491"	5	03F8	0004
"491 Request Pending"	4	03F8	0013
"493"	5	0457	0004
"493 Undecipherable"	4	0457	0012
"494"	5	00FE	0004
"494 Security Agreement Required"	4	00FE	001F
"500"	5	014E	0004
"500 Server Internal Error"	4	014E	0019

续表

String	Pr	Off	Len
"501"	5	040B	0004
"501 Not Implemented"	4	040B	0013
"502"	5	04E9	0004
"502 Bad Gateway"	4	04E9	000F
"503"	5	02C5	0004
"503 Service Unavailable"	4	02C5	0017
"504"	5	027B	0004
"504 Server Time-out"	4	027B	0013
"505"	5	01F1	0004
"505 Version Not Supported"	4	01F1	0019
"513"	5	0344	0004
"513 Message Too Large"	4	0344	0015
"580"	5	09FB	0004
"580 Precondition Failure"	4	09FB	0018
"600"	5	07E3	0004
"600 Busy Everywhere"	3	07E3	0013
"603"	5	0B09	0004
"603 Decline"	4	0B09	000B
"604"	5	09BE	0004
"604 Does Not Exist Anywhere"	4	09BE	001B
"606"	5	0430	0004
"606 Not Acceptable"	4	0430	0012
"687"	5	0359	0004
"687 Dialog Terminated"	4	0359	0015
"Anonymous"	3	08A4	0009

表 5-18 SDP 在压缩字典中的位置

String	Pr	Off	Len
"v=0[CRLF]o="	2	0BB8	0007
"[CRLF]s="	2	0C2B	0004
"[CRLF]s="	2	0C2B	0005
"[CRLF]i="	4	07A6	0004
"[CRLF]u="	4	07AE	0004
"[CRLF]e="	4	079E	0004
"[CRLF]c=INIP4"	3	08E1	000B
"[CRLF]c=INIP6"	2	0B98	000B
"[CRLF]c="	5	08E1	0004
"[CRLF]b="	3	0916	0004
"[CRLF]t="	2	0B78	0004
"[CRLF]t=00"	2	0B78	0007
"[CRLF]r="	4	0796	0004
"[CRLF]z="	4	079A	0004
"[CRLF]k=clear:"	4	06B0	000A
"[CRLF]k=base64:"	4	0690	000B



续表

String	Pr	Off	Len
"[CRLF]k=uri:"	4	0732	0008
"[CRLF]k=prompt:"	4	056B	000B
"[CRLF]k="	5	056B	0004
"[CRLF]a=cat:"	4	072A	0008
"[CRLF]a=keywds:"	4	0685	000B
"[CRLF]a=tool:"	4	0712	0009
"[CRLF]a=ptime:"	4	06A6	000A
"[CRLF]a=maxptime:"	4	05EA	000D
"[CRLF]a=rtpmap:"	2	0C0C	000B
"[CRLF]a=recvonly"	3	08C9	000C
"[CRLF]a=sendrecv"	3	093B	000C
"[CRLF]a=sendonly"	3	08D5	000C
"[CRLF]a=inactive"	3	0886	000C
"[CRLF]a=orient:portrait"	4	0974	0013
"[CRLF]a=orient:landscape"	4	03D2	0014
"[CRLF]a=orient:seascape"	4	050B	0013
"[CRLF]a=type:broadcast"	4	0442	0012
"[CRLF]a=type:meeting"	4	055B	0010
"[CRLF]a=type:moderated"	4	051E	0012
"[CRLF]a=type:test"	4	08B1	000D
"[CRLF]a=type:H.332"	4	0817	000E
"[CRLF]a=type:recvonly"	4	054A	0011
"[CRLF]a=charset:"	4	0653	000C
"[CRLF]a=sdplang:"	4	0647	000C
"[CRLF]a=lang:"	4	06F7	0009
"[CRLF]a=framerate:"	4	05C7	000E
"[CRLF]a=quality:"	4	063B	000C
"[CRLF]a=fmtp:"	2	0C50	0009
"[CRLF]a=curr:"	2	0C59	0009
"[CRLF]a=des:"	2	0C72	0008
"[CRLF]a=conf:"	2	0C35	0009
"[CRLF]a=mid:"	4	0752	0008
"[CRLF]a=group:"	4	06D8	000A
"[CRLF]a=key-mgmt:mikey"	4	0257	0012
"[CRLF]a=key-mgmt:"	4	0257	000D
"[CRLF]a="	5	0257	0004
"[CRLF]m=audio"	2	0C21	000A
"[CRLF]m=video"	2	0C17	000A
"[CRLF]m=application"	4	0330	0010
"[CRLF]m=data"	4	0700	0009
"[CRLF]m=control"	4	062F	000C
"[CRLF]m="	5	0330	0004
"AS"	3	091A	0003
"CT"	3	091D	0003

续表

String	Pr	Off	Len
"RTP/AVP"	2	0BE3	0008
"RTP/SAVP"	3	0892	0009
"RTP/AVPF"	3	089B	0009
"udp"	4	07DB	0003
"0.0.0.0"	4	0B7E	0007
"qos"	2	0CA9	0003
"mandatory"	2	0A8D	0009
"optional"	2	09AE	0008
"none"	2	0AF4	0004
"failure"	4	0A5C	0007
"unknown"	4	0BDA	0007
"e2e"	2	0AB1	0003
"local"	2	0A36	0005
"remote"	2	0AD6	0006
"send"	2	08D9	0004
"recv"	2	0553	0004
"sendrecv"	2	093F	0008
"AMR"	2	0BE1	0003
"octet-align="	4	094A	000C
"mode-set="	4	0709	0009
"mode-change-period="	4	04F8	0013
"mode-change-neighbor="	4	04A2	0015
"crc="	4	07AA	0004
"robust-sorting="	4	0166	000F
"interleaving="	4	0323	000D
"channels="	4	0606	0009
"octet-align"	4	094A	000B
"telephone-event"	4	00EE	000F
"events"	4	00F8	0006
"rate"	4	052B	0004
"tone"	4	0453	0004
"rtcp-fb"	4	04C0	0007
"ack"	4	0D1F	0003
"nack"	4	0D1E	0004
"ttr-int"	4	098C	0007
"app"	4	0334	0003
"rpsi"	4	0CDB	0004
"pli"	4	0336	0003
"sli"	4	09A9	0003

## 5.9 小结

本章首先介绍了移动网络发展的趋势,着重介绍了 IMS 系统框架、及 IMS 系统对 SIP 协

议的新的要求及 SIP 协议新的特点，随后介绍 SIP 协议在 IMS 系统中的应用及其利用 SIP 开展的 IMS 业务。

## 5.10 习题

1. 介绍移动网络发展的趋势及 IMS 系统组成。
2. 简述 IMS 系统中，SIP 协议的新特点。
3. 简述 SIP 协议在移动网络中的应用，以及可开展的业务。

## 第6章 SIP 其他应用

知识点:

- SIP 与 ENUM 结合的应用
- SIP 与 IPv6 结合的应用
- SIP 在设备控制中的应用

第1章到第5章已经探讨了 IPv6 基本协议与基本概念,并讨论了 SIP 在软交换网络和移动领域中的应用。本章中主要关注 SIP 与 ENUM 结合、SIP 与 IPv6 结合、SIP 在控制领域中的应用。

### 6.1 SIP 与 ENUM 的结合

#### 6.1.1 ENUM 简介

随着电信事业的不断发展进步,如今每个人的名片上都要包括一长串的联系信息,如:家庭电话号码、办公室电话号码、传真号码、手机号码和电子邮件等。因此人们希望可以使用一个统一的号码,免去记忆各种烦琐信息之苦。目前在公众电话网上普遍使用的 E.164 号码,已被大众普遍接受,使用 E.164 号码作为统一的号码是比较理想的。在以上两个因素的推动下,产生了 ENUM。

ENUM (tElephone NUmber Mapping) 的全称为电话号码到 URI 的映射,它实际上是一种将 E.164 国际电话号码映射为相应的 URI 的机制。ENUM 使用了在 RFC2916 中规定的由 IETF 发展的一种协议,通过这种协议可先将 E.164 号码翻译为域名,然后再使用 Internet 的域名系统 (DNS) 接入那些可以提取出 URI 的记录。通过使用 ENUM 机制,一个电话号码对于主叫用户来说可以是各种各样的地址,包括:电话、传真和电子邮件,即主叫用户可以通过一个电话号码给被叫用户打电话、发传真或发送电子邮件等;作为被叫用户,他/她可以规定适合于自己的通过单一号码接入的方式,即规定该单一电话号码所对应的电话号码、传真号码和电子邮件地址等,而且可以通过改变 DNS 中的相应的记录容易地更改他们的联络信息,不必改变接入的号码(即前面提到的单一的号码)。

在 ENUM 机制中,通过将 E.164 号码转换为 DNS 域名,并利用已经存在的 DNS 业务,使用为支持 ENUM 在 DNS 中新引入的 NAPTR (Naming Authority Pointer) 记录,就可以找到一个给定的域名 (E.164 号码)。

使用 ENUM 的目的是在 NGN 与 3G 网络中统一号码,提高效率。

#### 6.1.2 ENUM 作用

ENUM 的作用体现在:

- ◆ 在促进网络融合以及促进市场竞争方面发挥积极的作用

ENUM 的实施将会提供一种全球性的网络基础架构服务,这种服务可以方便地实现电路交换网用户与基于 IP 网络用户的互通;允许用户使用统一的 E.164 号码接入多种应用;用户还可以方便地更改自己的联络信息而不修改接入号码。ENUM 的实现使得利用 DNS 系统来影

响呼叫的建立成为可能, 给 Internet 的业务提供商、域名注册商等提供了更多的机会, 这样可以通过 E.164 号码接入他们的业务, 而 E.164 号码在全球的应用范围要比因特网的域名或电子邮件地址广泛得多。归属于传统电信运营商的用户将有机会使用由因特网 DNS 系统参与提供的灵活的电信业务, 可以方便的为用户提供更加丰富的应用。

◆ 从长远看, ENUM 有可能成为未来网络和业务融合的一个重要机制

目前, 已能看到一些重要的应用 (如移动多媒体短信); 从长远看, ENUM 有可能成为未来网络和业务融合的一个重要机制。下一代网络将是融合的网络, 同样存在着不同网络互通时的地址转换问题; 下一代网络中, 通信手段更加多样化, 仍然存在着统一号码的需求; 因此 ENUM 在下一代网络中仍有可能发挥重要的作用。

◆ 很多国家都对 ENUM 技术表现了兴趣并在积极开展试验

目前, ENUM 已经成为很多国际标准化研究组织的研究热点。英、美、瑞典、奥地利等许多国家都在计划或正在开展 ENUM 试验。

◆ ENUM 的引入能够简化和改善现有业务的提供和管理方式, 为运营商和用户带来效益

从应用的角度, ENUM 有以下主要用途: 不同网络、不同业务、不同运营商间的互通和寻址; 用户个人业务的融合及衍生出的相关业务、业务的统一标识。

从客户的角度, 主要有两类: 企业内部应用; 公众电信业务。

### 6.1.3 利用 SIP 和 ENUM 发起呼叫

#### 1. PSTN 到 SIP 终端呼叫

利用 ENUM 建立一个由 PSTN 用户发起的到基于 IP 网络的 SIP 终端的呼叫, 其典型的流程如图 6-1 所示。

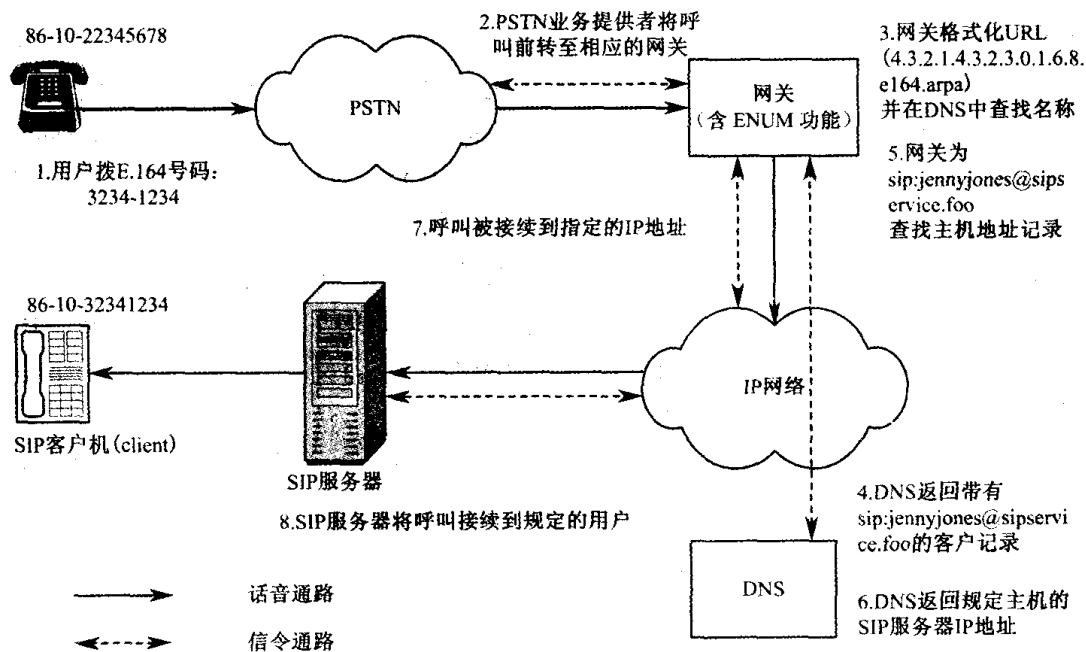


图 6-1 PSTN 用户呼叫 SIP 终端用户

图 6-1 中各个步骤的含义如下:

- 步骤 (1), 发端用户拨 E.164 号码; 拨号的方式由各国自己规定, 如长途呼叫拨前缀 “0”, 国际呼叫拨前缀 “00” 等;
- 步骤 (2), PSTN 的业务提供者将呼叫转至相应的支持 ENUM 功能的网关 (可能需要到相应的数据库查询该用户是否申请了 ENUM 功能), ENUM 网关的选择以及 ENUM

网关的物理位置需要综合考虑多方面的因素。

- 步骤 (3), 该网关必须包含 ENUM 功能, 它能够将用户所拨的数字按照规定的规则转换成 URL, 如果用户拨的号码不是全号, 则网关应能够将丢失的数据补齐, 这样才能形成一个完整有效的 URL。例如: 用户拨的号码为: 32341234, 完整的号码应是: 8610323412343。然后网关在 DNS 中查找该名称;
- 步骤 (4), DNS 返回与该 URL 相关的所有业务记录;
- 步骤 (5), 网关在 DNS 中查找规定主机的地址记录;
- 步骤 (6), DNS 返回规定主机的 SIP 服务器 IP 地址;
- 步骤 (7), 通过基于 IP 的网络将该呼叫接续到指定的 IP 地址;
- 步骤 (8), SIP 服务器将呼叫接续到规定用户的用户代理客户机上。当目的方应答时, 必须将应答监视返回给发端的本地交换机。

可能的消息流程如图 6-2 所示。

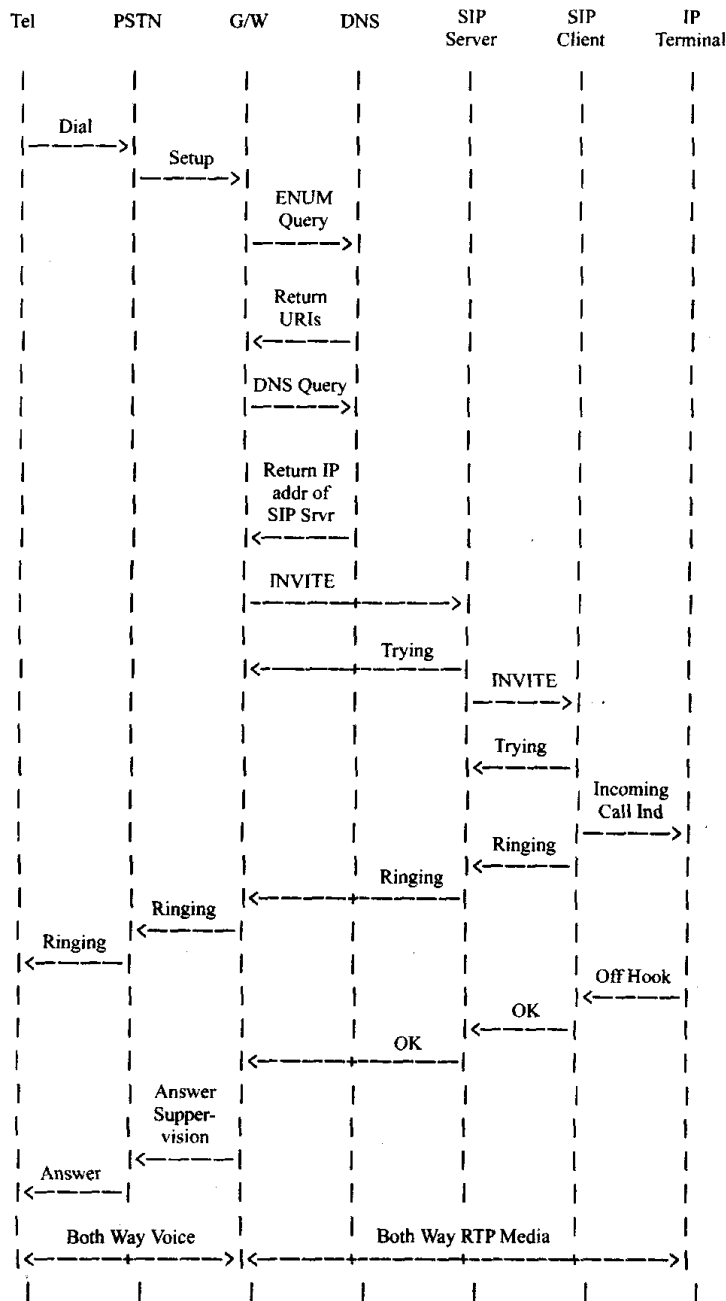


图 6-2 PSTN 用户呼叫 SIP 终端时可能的消息流程

## 2. SIP 终端到 PSTN 呼叫

利用 ENUM 建立一个由基于 IP 网络的 SIP 终端发起的到 PSTN 用户的呼叫，其典型的流程如图 6-3 所示。

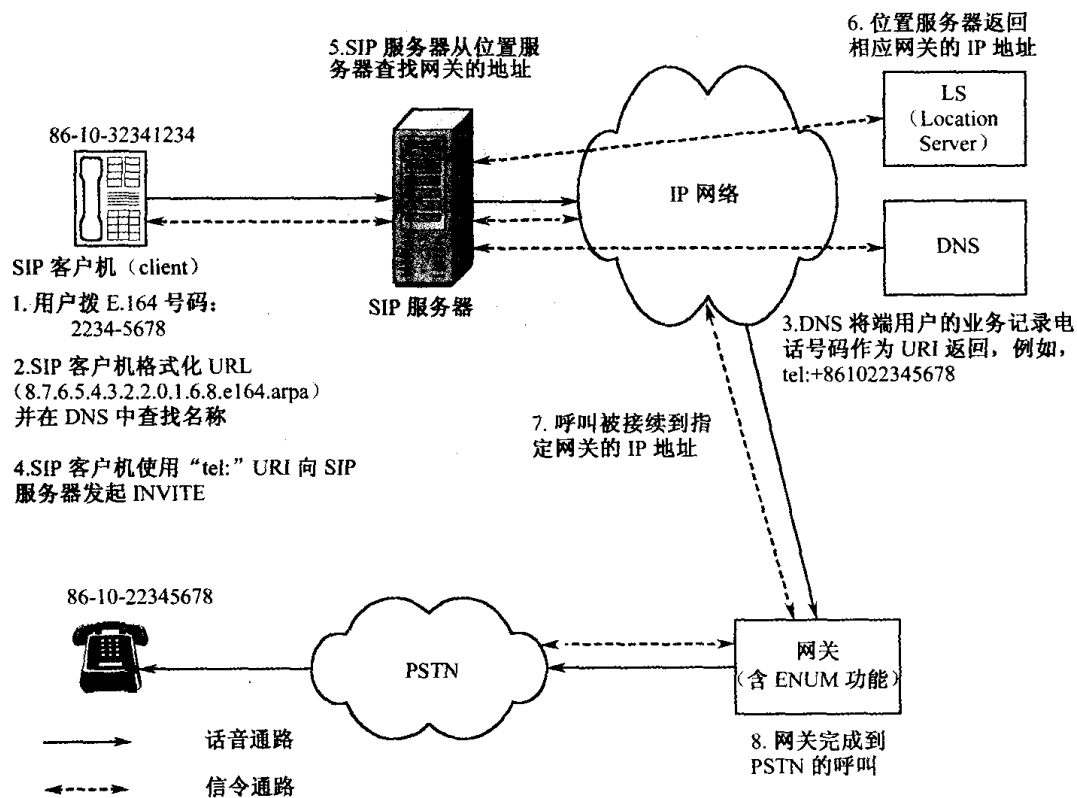


图 6-3 SIP 终端用户呼叫 PSTN 用户

图 6-3 中各个步骤的含义如下:

- 步骤 (1), 发端用户拨 E.164 号码; 拨号的方式由各国自己规定, 如长途呼叫拨前缀 "0", 国际呼叫拨前缀 "00" 等;
- 步骤 (2), SIP 客户机必须包含 ENUM 功能, 它能够将用户所拨的数字按照规定的规则转换成 URL, 如果用户拨的号码不是全号, 则 SIP 客户机应能够将丢失的数据补齐, 这样才能够形成一个完整有效的 URL。例如, 用户拨的号码为: 22345678, 完整的号码应是: 861022345678。然后 SIP 客户机在 DNS 中查找该名称;
- 步骤 (3), DNS 返回与该 URL 相关的所有 NAPTR 记录; 如果没有找到与该 URL 相关的 ENUM 记录, 则应使用用户所拨的号码继续到 PSTN 的呼叫处理;
- 步骤 (4), SIP 客户机使用 "tel:" URI 向 SIP 服务器发送 SIP INVITE;
- 步骤 (5), 如果发端和终端属于不同的 IP 电话管理域, SIP 服务器可使用一种前端协议向位置服务器查询该电话号码的网关的 IP 地址。
- 步骤 (6), 位置服务器返回与目的地号码对应的网关的 IP 地址;
- 步骤 (7), SIP 服务器将呼叫接续到指定网关的 IP 地址;
- 步骤 (8), 网关通过 PSTN 完成到目的地电话的呼叫。网关必须对来自 PSTN 的任何信令 (如回铃音、忙等) 都做出响应, 并且将相应的信息返回给呼叫的发起者。

可能的消息流程如图 6-4 所示。

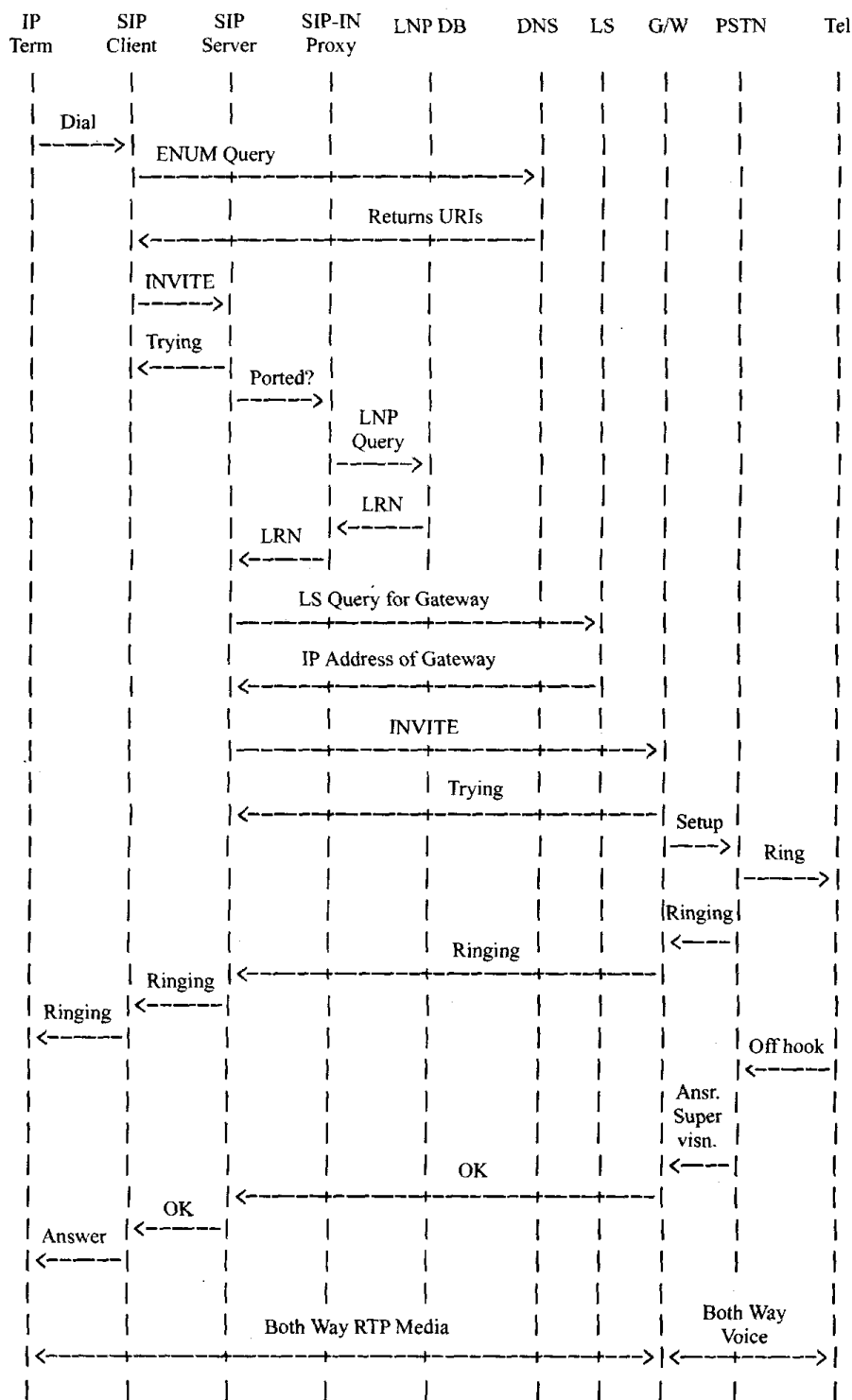


图 6-4 SIP 终端呼叫 PSTN 用户时可能的消息流程

### 3. SIP 终端间的呼叫

SIP 终端呼叫 PSTN 用户时可能的消息流程如图 6-5 所示。

图 6-5 中各个步骤的含义如下：

- 步骤 (1)，用户 A 不知道用户 B 的确切 SIP 地址，因此用户 A 在自己的 SIP 电话上输入用户 B 的 E.164 号码：+1-214-555-1234。随后 SIP 电话将该电话号码带入向预先设置的 SIP 代理服务器发出的呼叫邀请消息中。
- 步骤 (2)，此 SIP 代理服务器将呼叫发往位置服务器，得到合适的 SIP 地址。



- 步骤 (3), 位置服务器中的数据库没有特定的 E.164 号码表将号码转换成域名形式, 向 ENUM 服务器 e164.arpa 发出 ENUM 查询 (注: 只有在本图中使用 .arpa)。
- 步骤 (4), 因为 E.164 号码在 e164.arpa 域中有对应的 NAPTR 资源记录, 因此 ENUM 服务器将该记录返回给位置服务器。资源记录包括用户 B 的 SIP 地址: “sip:j.doe@acme.com”。
- 步骤 (5), 位置服务器将从 NAPTR 资源记录中的数据翻译成 300 (或者 302) “Move” 消息, 发回到 SIP 代理服务器。
- 步骤 (6), SIP 代理服务器将 300/302 消息的域翻译成 SIP URI, 放入 INVITE 消息中, 并发送到用户 B 的 IP 电话上。
- 步骤 (7), 用户 B 的 SIP 电话接到 SIP 邀请, 发 200 OK 消息回 SIP 代理服务器, 表示用户 B 已经摘机。
- 步骤 (8), 代理服务器将 200 OK 返回给用户 A 的 SIP 电话。
- 步骤 (9), 用户 A 的 SIP 电话将 ACK 确认消息发回给 SIP 代理服务器, 表示可以建立 RTP 媒体流通道。
- 步骤 (10), 代理服务器将 ACK 消息转发到用户 B 的 SIP 电话上。
- 步骤 (11), 双向的基于 UDP 的 RTP 媒体会话在两个用户间建立, 并可以开始通话。

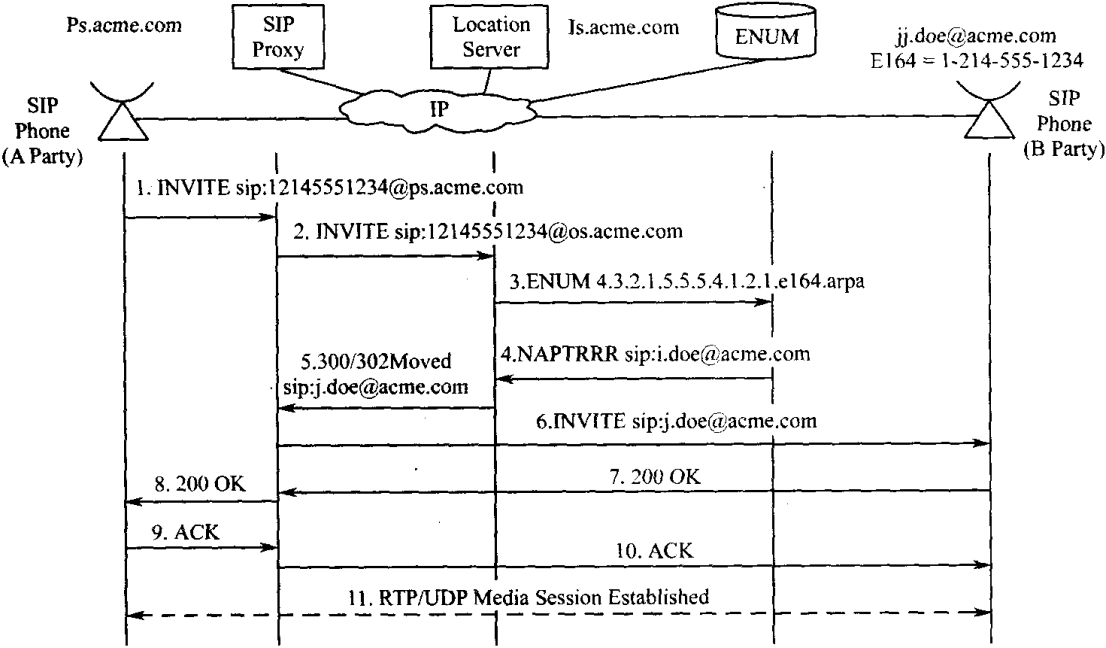


图 6-5 基于 ENUM 的 SIP 电话间通话业务流程

## 6.2 SIP 与 IPv6 的结合

### 6.2.1 兼容 IPv6 的 SDP

#### 1. 描述

RFC3266 给出了兼容 IPv6 的 SDP, 具体如下:

uri= ; 定义在RFC1630和RFC2732

multicast-address = IP4 多播/IP6 多播  
IP4-multicast = m13\*("."decimal-uchar)"/"ttl["/"整数]  
; IPv4 多播地址范围 224.0.0.0 到 239.255.255.255  
m1 = ("22" ("4"/"5"/"6"/"7"/"8"/"9"))/("23" DIGIT))  
IP6-multicast= 以 FF 开始的 IPv6 十六进制地址  
addr = FQDN/单播地址  
FQDN = 4\*(alpha-numeric/"-"/"."), 见 RFC1035  
unicast-address = IP4 地址/IP6 地址  
IP4-address = b1 3\*("."decimal-uchar)"/"0.0.0.0"  
b1 = decimal-uchar, 小于 224, 不能为 0 和 127  
IP6-address = 十六进制[":"IP4 地址] (见 RFC2373)  
hexpart = hexseq/hexseq"::"[hexseq]"/"::"[hexseq]  
hexseq = hex4\*(":"hex4)  
hex4 = 1\*4HEXDIG

2. 实例

v=0  
o=nasal 971731711378798081 0 IN IP6 2201:056D::112E:144A:1E24  
s=(几乎)是通过 Mars-II 卫星播放现场视频  
p=+1 713 555 1234  
c=IN IP6 FF1E:03AD::7F2E:172A:1E24  
t=3338481189 3370017201  
m=audio 6000 RTP/AVP 2  
a=rtpmap:2 G726-32/8000  
m=video 6024 RTP/AVP 107  
a=rtpmap:107 H263-1998/90000

6.2.2 DHCPv6 关于 SIP 的选项

该选项定义了含有可以被映射为 SIP 出口代理服务器的域名列表或者 IPv6 地址选项。具体有两个选项，一个是域名列表，另一个是 IPv6 地址选项。

1. 域名列表

其格式如图 6-6 所示。

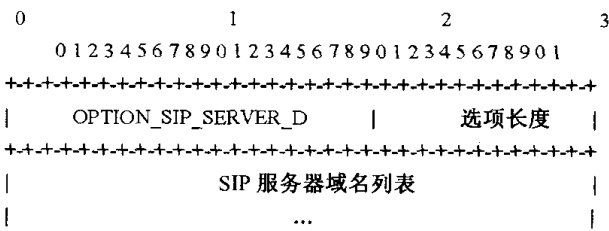


图 6-6 域名列表格式

其中，选项码为 OPTION\_SIP\_SERVER\_D(21)。选项长度为以字节为单位的 SIP 服务器域

名列表的长度。

2. SIP Servers IPv6 地址列表

其格式如图 6-7 所示。

其中，选项码为 OPTION\_SIP\_SERVER\_A(22)。选项长度为以字节为单位的选项长度。

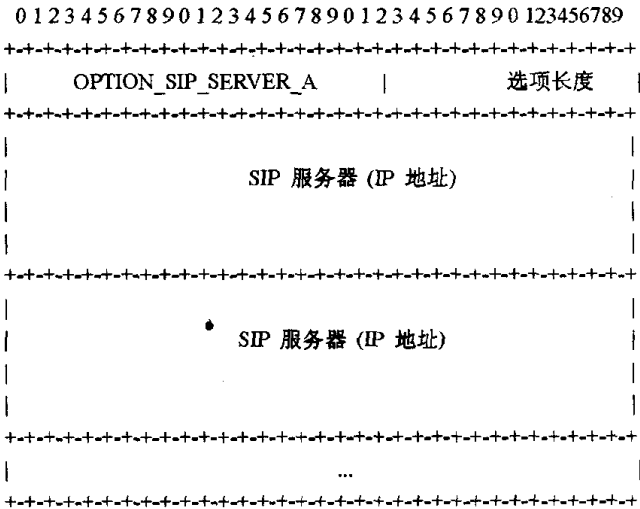


图 6-7 IP 地址列表格式

6.3 SIP 在 PINT 中的应用

6.3.1 PINT 介绍

1. 概述

ITU-T SG11 定义了 IN 与 Internet 互连的体系结构。PSTN 与 Internet 互通的业务，根据种类不同，可能涉及管理层、业务控制层及呼叫、承载层中的一个或几个层面。因此，设置了管理网关（MGF，Management Gateway Function）、业务控制网关（SCGF，Service Control Gateway Function）和呼叫/承载控制网关（C/B GF，Call/Bearer Gateway Function）3 个功能实体，分别负责管理层、业务控制层及呼叫、承载层 3 个层面的互通功能。

当然，为了互连业务的实现，IN 中原有实体的功能将相应增强，为了支持互连业务，在 Internet 侧需要新增加如下功能实体：PINT Server 及 PINT Client，并增强原有的 H.323 GKF（GateKeeper Function），使其具有 SSF 功能。PINT Client 的物理位置可以放在用户端，PINT Server 的物理位置可以放在 ISP，H.323- GKF 及 IP-SSF 可以置于 GateKeeper。

PINT Client 接收 Internet 网上的用户请求，实现 HTTP 协议与增强 SIP、SDP 协议的转化，将用户的业务请求及信息以增强的 SIP、SDP 协议传送至 PINT Server。

PINT Server 主要负责将来自于 PINT Client 的 Internet 用户的单击信号转换成业务请求消息，连同业务数据一起发送给 SCGF，并能接收经 SCGF 发来的 IN 的响应，以及将执行的结果提示给 Internet 用户。

2. PINT 基本业务

PINT 工作组定义了四种业务类型，这四种业务都属于 Web 上的 PSTN 业务。用户在浏览 Web 主页的同时也能享受 PSTN 业务。这类业务的特点是 Internet 网主要用于非语音传输，而

语音（包括传真）则是在 PSTN 网上传输。它的优势在于既可以利用 PSTN 传输语音的质量高，可根据预先设定好的标准（例如，时间、地理位置等）传输话路以及可靠性、安全性和灵活性，低成本的特点，又可以利用 Internet 网的普及性以及不受时间限制的特点。PINT 业务并不涉及语音如何在 Internet 上传输方面。

#### ◆ 单击拨号（Click To Dial）

当一个 Web 用户在浏览一个公司的主页并可以单击主页时，可以单击主页上提供的一个电话号码，通过 Web 发起一个电话呼叫，使用他的 PSTN 电话机和这家公司的一个代理人进行语音通信。这个语音通信是完全建立在 PSTN 上的，与 Internet 无关。

#### ◆ 单击发送传真（Click To Fax）

在 Web 用户上网期间，他可以单击一个传真电话号码，将一个文件从 Web 上传到这个电话号码的传真机上。这种业务通常用于发送端是 Web 用户，但无传真机；而接收端用户仅有传真机，并无 Internet 接入的情形。例如：一个 Web 用户想向一个旅馆预定房间，而该旅馆没有开放 Web 业务或不具备 Internet 接入条件，就只能通过传真预定房间，此时 Web 用户就可以利用这个业务完成房间的预定。

#### ◆ 单击回送传真（Click To Fax Back）

Web 用户可以在上网浏览某个公司的主页期间，单击某个文件，而对这个文件，对方只能以传真的方式提供。另一种情形是，若单击的文件是一个图像信息，通过 Internet 下载可能既费时又浪费 PC 上的存储空间，而用传真机来接收更快、更方便。

#### ◆ 语音接入信息库（Voice Access To Contents）

这一业务是将 Web 主页上的一些内容以语音的方式通过 PSTN 网，送到 Web 用户的电话机上。这个业务可以在 Web 用户上网期间得到，也可以直接通过拨打 PSTN 电话得到服务。例如，一个通过 Web 浏览阅读新闻摘要的人，在他出差期间，仍然可以通过拨打某一电话号码来接收用语音传送的 Web 上的新闻摘要。

### 6.3.2 PINT 对 SIP 的增强

PINT 协议是 SIP/SDP 协议的增强，主要体现在如下几个方面。

对 SDP 的增强：

- 新的网络类型“TN”和地址类型“RFC2543”、“X - ...”；
- 新的媒体类型“text”、“image”和“application”，新的协议传输关键字“voice”、“fax”和“pager”，以及相关的格式类型与属性标签；
- 对应于被包含内容的新的格式属性；
- 用于将信息传递到电话网的新的属性标签；
- 属性标签“strict”，被客户程序用来要求服务器端必须具有某些属性。

对 SIP 的增强：

- MIME 格式的有效载荷；
- 支持“Warning:”头域；
- 增加了“SUBSCRIBE”和“NOTIFY”两个请求方法；
- “Require:”头域；
- PINT 请求中的 PINT URL 格式；

- PINT URL 中有关电话网的参数。

## 6.4 SIP 在 SPIRITS 中的应用

### 6.4.1 SPIRITS 介绍

PINT (PSTN/Internet Interworking Service) 提供了从因特网请求 PSTN/IN 业务的模型。如何解决从 PSTN/IN 请求因特网业务的问题, 是 SPIRITS 的任务。SPIRITS 是智能网和因特网互通的又一种模型。SPIRITS 的体系结构里面含有 PINT 成分, 用来完善 SPIRITS 业务, 例如, 用 PINT 来实现 SPIRTIS 服务器的注册, 也可以把 PINT 和 SPIRITS 结合起来给用户提供更完善的服务, 如用户通过 PINT 请求呼叫中心发起呼叫到该用户。随后由呼叫中心发起的呼叫到达该用户时, 可通过 SPIRTIS 对用户提示来话。

#### 1. 传送协议

传送协议本应属于服务原语具体实现的范畴, 本书放在这儿讲主要是为了便于描述业务流程, 从而更具体地阐述 SPIRTIS 服务原语。

主张采用 SIP 传送 SPIRITS 报文。首先是因为 PINT 是使用 SIP 的成功典范, 其次使用 SIP 非常容易实现 SPIRITS 服务器用 IP 电话回话。SIP 是 IP 域中实现会话的理想途径。

用 HTTP 承载 SPIRITS 报文对某些 SPIRTIS 业务实现也很简便, 但要实现 IP 回话却很牵强。还有另一种方法是重新定义传送协议, 但这显然会花费不必要的精力。

#### 2. 服务和原语分类

SPIRITS 服务实质是协调 PSTN/IN 和因特网之间的通信, 考察 SPIRITS 体系结构和三类标志性业务, 不难发现 SPIRITS 的服务可以分三类:

##### ◆ SPIRITS Sever 注册和注销

SPIRITS 服务器注册完成电话号码和 IP 地址的绑定。注册原语的实现可以使用 PINT 中的 REGISTER。通过注册, SPIRTIS 服务器作为 SIP 用户代理服务器在 SIP Proxy 或 Redirect 服务器中有了登记项后, PSTN/IN 端发来的报文才有了目的地。注册由经 PSTN 拨号上网的 PINT 客户机在因特网域发起 REGISTER 注册报文, 在某一个充当 REGISTER 服务器的 PINT 服务器中止, 第一次 REGISTER 时必须要有用户的电话号码信息。注销可以是 SPIRITS 服务器发起注销申请, 也可以是 PINT 服务器在一段时间未收到注册报文时自动注销。PINT REGISTER 过程和 SIP REGISTER 过程相似。为使 IN 端的 SCF/SDF 能准确判断用户是否注册, 可以将注册的定位信息放在 SDF。

##### ◆ 业务处理服务

业务处理服务流程可以分成两段, 一段是在 PSTN/IN 端, 在这一段上 IN 端的 SSF 检测到用户的电话忙, 并判断用户正在因特网域时, 将向用户所在的 SPIRITS 服务器发出呼叫请求, 在 SPIRITS 服务器响应后, 将响应返回给 SCF。另一段在因特网域, 业务处理是 SPIRITS 服务的实现主体。SIP INVITE 可以有效地完成业务请求, INVITE 操作在 SPIRITS 客户机发起。SPIRTIS 服务器根据用户的要求返回相应的 SIP 响应信息。

##### ◆ 业务监控服务

类似于 PINT, 一些 SPIRITS 业务可能需要监控。要达到业务监控目的, 有两个条件: 一个是 SSF 配置了相应的 DP, 这种 DP 一般是动态的事件检测点 EDP, EDP 触发后, SPIRITS

客户机发出 SUBSCRIBE 报文；另一个是 SPIRITS 服务器能返回 NOTIFY 报文。SUBSCRIBE 和 NOTIFY 报文可以完成由 EDP 触发的业务。SPIRITS 的三类标志性业务均不需要执行业务监控。

### 6.4.2 SIP 在 SPIRITS 中的应用

下面以 ICW 为例，介绍 SIP 在 SPIRITS 中的应用。Internet 的大多数用户都是用电话线通过 modem 上网，所以他们在上网的同时不能接听电话。ICW 提供了一种简单的图形界面方式通知 Internet 上的用户。当用户有电话来时，ICW 应用程序将弹出一个选择框通知用户，用户可以根据 ICW 提供的服务完成电话连接，前提是用户必须首先在 ICW 服务器上进行登记。

通过 ICW 服务，可以完成更多的呼叫连接。呼叫连接是电信部门服务质量的一个重要参数。那些终端用户忙或者没有回答的呼叫会消耗大量的 PSTN 资源。ICW 服务可以完成更多的呼叫连接，减少资源消耗。这对客户和服务提供者都有好处。

ICW 平台是 ICW 服务的基础：服务提供者还可以提供其他的业务，例如，Internet 跟我来 (Follow-me)，个性化呼叫管理 (personalized call management)，统一的信息业务 (unified messaging service)，以及其他包括语音和数据的呼叫管理。

通过 ICW 服务，服务提供者可以为用户提供以下方便：

- 通过一条简单的电话线管理语音和数据呼叫。
- 可以跟踪来话。
- 图形界面提供了一种简单的、直观的交互方式，并容易实现用户化。

如图 6-8 所示是 ICW 示意图。

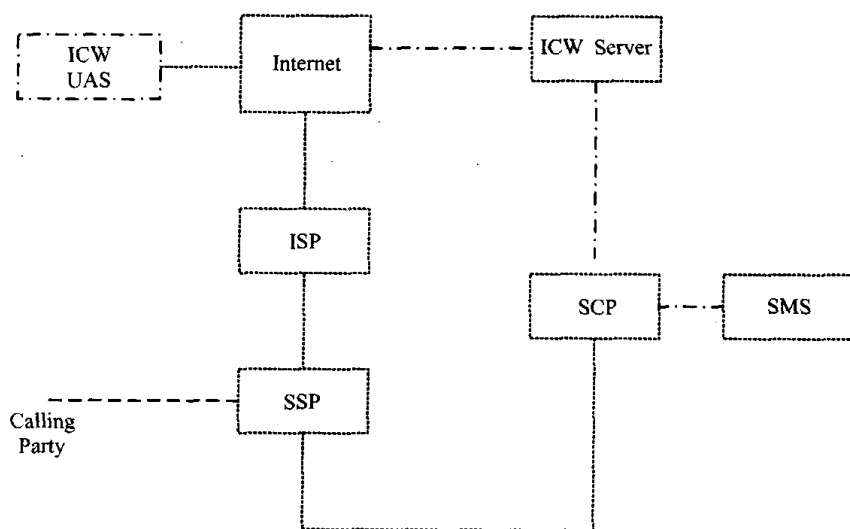


图 6-8 ICW 示意图

ICW 用户代理服务器 (UAS) 与 ICW 服务器之间的通信是建立在客户-服务器模式的基础上的。UAS 在有呼叫到来时，将弹出一个对话框通知用户呼叫方的信息，用户可以根据 UAS 提供的服务做出选择。UAS 将用户的选择消息返回给 ICW 服务器。如果用户接收来话，那么 UAS 将断开 Modem 和 Internet 的连接，将电话线接入。

ICW 服务器的功能如下：

- 监听从 SCP 应用程序发来的消息。

- 为 ICW 应用程序提供数据存储机制。
- 可以处理登记用户的图形用户界面 (GUI) 请求。

ICW 呼叫执行过程如下:

#### 1. 服务登记 (Subscription to the service)

用户在 ICW 服务器上进行登记。

用户从 ICW 服务器上下载并安装 ICW UAS 软件。

登记用户的信息存放在 SMS 和 SCP 中。

#### 2. 服务的执行 (对 ICW 用户来说是透明的)

ICW UAS 与 ICW 服务器之间建立 TCP 连接。

用户在登记前先进行身份论证, 并使用加密口令和电话号码进行登记。

ICW 服务器将用户信息存储在数据库中。

#### 3. 呼叫到来 (call arrival)

呼叫方拨打用户的电话号码。

SSP 交换机发触发请求消息给 SCP。

SCP 进行数据库查询。

SCP 确认这个呼叫是否属于已登记的 ICW 用户的 (如果不是, 则执行其他服务逻辑)。

SCP 将消息以及呼叫方的电话号码, 呼叫方名称 (可选) 和被呼叫方的电话号码发给 ICW 服务器 (并接收从 ICW 服务器发来的确认消息)。

ICW 服务器确认登记用户的 ICW 服务是否有效。如果有效, 那么它返回一个 “TRYING” 消息给 SCP。SCP 指示 SSP 播放提示音, 如振铃。ICW 服务器根据被叫方的电话号码以及 ICW UAS 的 IP 地址, 发出一个邀请消息给 ICW UAS。

如果登记用户的 ICW 服务无效, 那么 ICW 服务器返回 “NOT FOUND” 消息给 SCP。SCP 指示 SSP 按正常的呼叫处理程序执行。

#### 4. 用户返回选择消息给 SCP

ICW UAS 将用户的选择消息返回给服务器。

服务器将这个选择消息转发到 SCP。

当有呼叫来时, UAS 将弹出一个框让用户选择, 可供用户选择的项有:

- 拒绝呼叫 (refuse call): UAS 返回一个 SIP “603 refuse” 消息给 ICW 服务器。
- 等待呼叫 (hold call): 呼叫方将听到 “请等候” 的提示音, 这表示登记用户将接听来话。或者提示呼叫方, 被叫方即将给他回电话。这种情况下, SIP UAS 发一个 “182 Queue” 消息给 ICW 服务器。
- 发送语音邮件 (send to voice mail): ICW 服务器将用户的语音消息转化为数据, 发到用户的邮箱中。对 SIP UAS 来说, 它将发一个 “380 Alternative service” 给 ICW 服务器。这种重定向还可以转到其他的域。例如, 蜂窝移动电话、另一条电话线、秘书等。
- 接听电话 (accept call): UAS 断开 Modem 与 Internet 的连接, 接听来话。这时, UAS 将返回一个 “200 OK” 消息给 ICW 服务器。

## 6.5 SIP 在 DCS 中的应用

### 6.5.1 PacketCable 介绍

PacketCable 是 CableLabs 组织 1997 年制定的, CableLabs (Cable Television Laboratories, Inc.) 是 1988 年成立的有线电视行业的一个非营利研究协会, 致力于研究制定新的有线电视通信技术, 帮助有线电视运营商在现有网络中集成新技术。PacketCable 是为在双向有线电视系统上提供实时多媒体服务而制定的标准, 使用在有线电视网络中普遍采用的 DOCSIS 网络架构的网络控制信令为基础的体系。

有线电视运营商正在寻找使用有线电视基础网络 HFC 的新服务, 创造新的经济增长点, 提供数据、语音、图像综合的服务。基于有线电视系统的 IP 电话将是第一个最有可能推出的新服务。2002 年 InStat/MDR 的分析报告说, 2006 年基于有线电视的 IP 电话网将达到 75 亿美元的年收入。

PacketCable 体系结构是基于支持 DOCSIS1.1 的双向 HFC 设备、IP 骨干网、PSTN。它提供基于 HFC 设备、对 VoIP 呼叫非常重要的服务质量保证。它同时支持包头压缩以降低在 HFC 上传送的数据包长度, 这样可以提高带宽利用率, 可以承载更多的 VoIP 呼叫。

PacketCable 网络包含:

- Cable Modem 终端系统 (CMTS): 它用于与 HFC 系统用户端的 Cable Modem 通信, 同时连接到 IP 骨干网; CMTS 是设立服务质量保证的关键, 它提供动态的服务质量保证; CMTS 同时保证所有用户端 Cable Modem 同步和访问 PSTN 的优先级。
- 用户端设备 (CPE): DOCSIS1.1 标准的 Cable Modem 和 MTA。MTA 包含与用户电话机的接口, 网络接口, 编、解码器, VoIP 要求的信令和压缩功能, QoS 信令。
- 电话、信令部分: PacketCable 采用 NCS (Network-based Call Signalling) 协议, 在 NCS 协议中, 终端设备有很少的智能, 呼叫控制由 CMS (Call Management Server) 完成, CMS 也可以称为 Softswitch; PacketCable 定义的 NCS 与 MGCP 协议非常相近。CMS 是 PacketCable 的中心设备, 提供呼叫特性、呼叫建立、呼叫结束、呼叫状态; 媒体网关提供与 PSTN 的语音通道连接; 信令网关在 PacketCable 网络边缘接收和发送 TDM 网络信令。
- 媒体服务器: 提供类似智能网服务的增强型 VoIP 服务, 如录音通知服务器为主叫用户提供网络录音通知; 会议电话服务器支持多路通话; 语音信箱或统一消息服务器提供语音通信和 E-mail 通信的综合服务。
- 配置服务器: 包括 DNS、DHCP、TFTP、ToD 服务器和对 CPE 设备配置的服务器。
- RKS (通话记录保存服务器): 提供 PacketCable 网与有线电视运营商计费系统之间的接口。

### 6.5.2 基于 SIP 的基本呼叫流程

基于 SIP 的基本呼叫 DCS 流程如图 6-9 所示, 图中 ER 为边缘路由器, DP 为 DCS 代理。



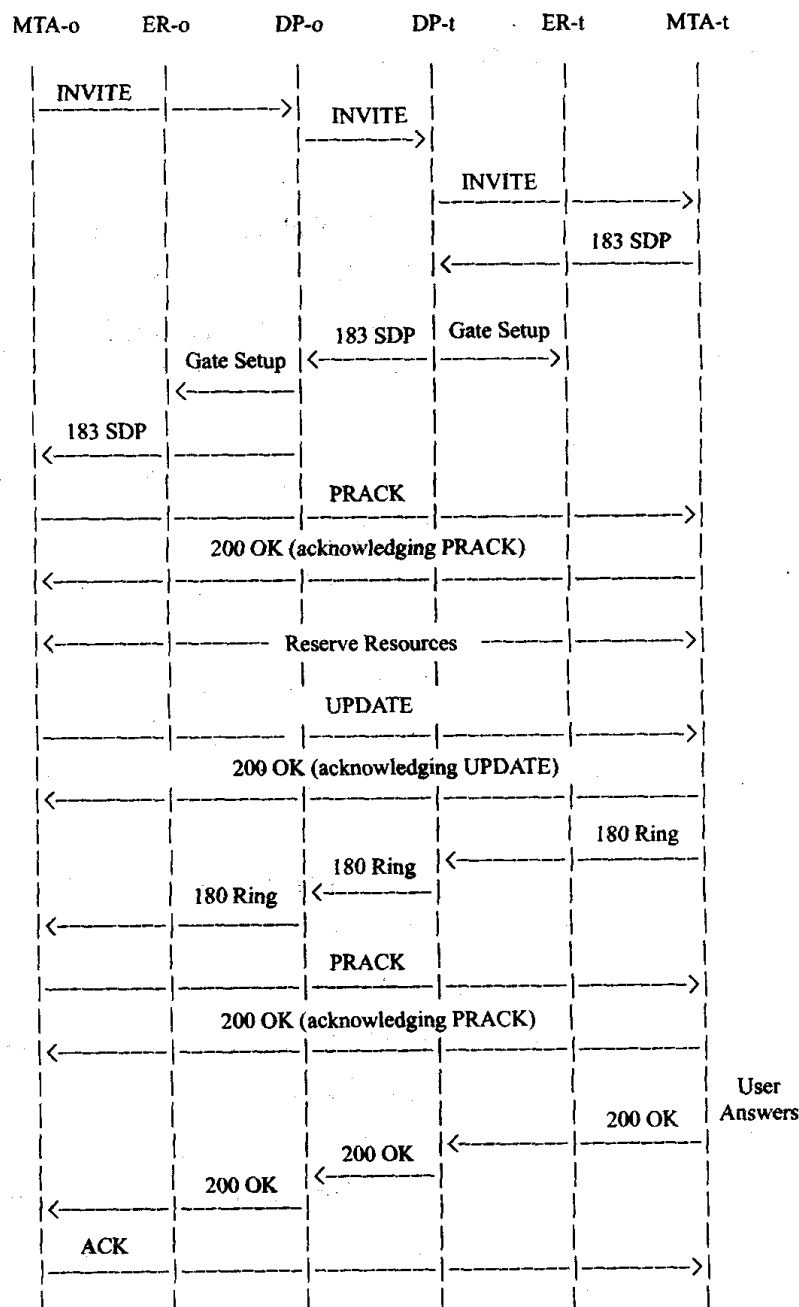


图 6-9 基于 SIP 的基本呼叫 DCS 流程

## 6.6 SIP 在设备控制中的应用

与 H.248、MGCP 不同，SIP 协议本身为一对等协议，不适合控制紧耦合的设备，但可以控制松耦合设备。

Draft-moyer-srp-appli-ances-framework 定义了一个新的方法 DO 来携带 SIP 命令，收到 DO 的实体按照消息来执行相应的动作，另外还定义了一个用于描述命令的格式——设备消息协议。

如图 6-10 所示为 SIP 在控制设备中的应用实例。

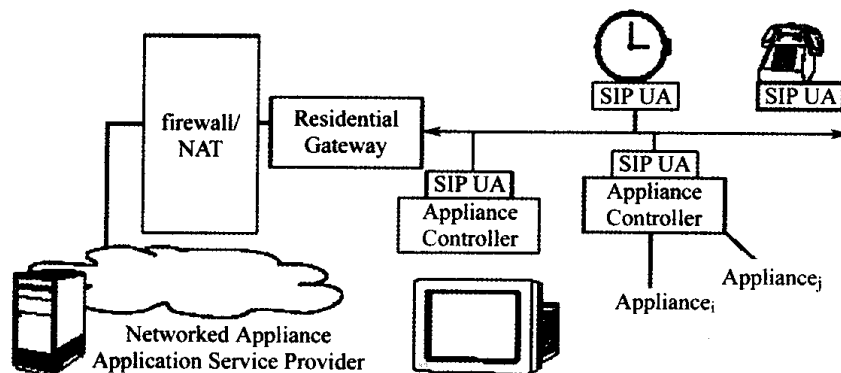


图 6-10 SIP 在控制设备中的应用

具体的消息如下：

1. REGISTER [register@home.net](mailto:register@home.net)

To: [slp:/d=alarmclock, r=bedroom, u=maguire]@ua.chips.home.net

From: [slp:/d=alarmclock, r=bedroom, u=maguire]@ua.chips.home.net

Content-type: application/ddp

[Device address]

2. INVITE sip:[slp:/d=alarmclock, r=bedroom, u=maguire]@home.net SIP/2.0

**From:** sip:announcement@alarmclock.net

To: [slp:/d=alarmclock, r=bedroom, u=maguire]@ua.chips.home.net

Via: [alarmclock.net](http://alarmclock.net)

Content-type: application/sdp

[SDP for uni-directional RTP stream]

3. INVITE sip:[slp:/d=alarmclock, r=bedroom, u=maguire]@home.net SIP/2.0

**From:** sip:announcement@alarmclock.net

To: [slp:/d=alarmclock, r=bedroom, u=maguire]@ua.chips.home.net

**Via: home.net**

Via: [alarmclock.net](http://alarmclock.net)

Content-type: application/sdp

**[SDP for uni-directional RTP stream]**

4. INVITE sip:[slp:/d=alarmclock, r=bedroom, u=maguire]@home.net SIP/2.0

**From:** sip:announcement@alarmclock.net

To: [slp:/d=alarmclock, r=bedroom, u=maguire]@ua.chips.home.net

Via: [chips.home.net](http://chips.home.net)

**Via: home.net**

Via: [alarmclock.net](http://alarmclock.net)

Content-type: application/sdp

[SDP for uni-directional RTP stream]

5. alarm 时钟以 RTP 参数响应，RTP 会话通过网络示众设备播放提示音到用户。

## 6.7 小结

前面已经探讨了 SIP 基本协议与基本概念,并讨论了 SIP 在软交换网络和移动领域中的应用。本章主要讨论了 SIP 与 IPv6 相结合的应用、SIP 与 ENUM 相结合的应用、SIP 在有线电视网络中的应用及 SIP 在控制领域中的应用。

## 6.8 习题

1. 简述 SIP 与 IPv6 结合的机制。
2. 简述 SIP 与 ENUM 相结合的应用机制。
3. 简述 SIP 其他的若干典型应用。

## 第7章 SIP 测试技术

知识点:

- 测试概述
- 相关组织
- 测试结果

本章首先介绍了协议测试的基本理论、随后介绍了 SIP 测试的相关组织及其互操作测试结果。

### 7.1 测试概述

#### 7.1.1 协议测试的基本概念

协议测试理论是协议工程学的一个重要分支。研究协议测试理论的原因在于一个标准化的协议并不能确保该协议的实现之间能够成功地进行通信。这是因为协议标准目前基本上是使用自然语言描述的,实现者对于协议的不同理解会导致不同的协议实现,甚至有时会是错误的实现。因此,需要一种有效的方法来对协议实现的正确性和有效性进行判别,这便是“协议测试”。实际上,协议测试是试图通过实验的方法找出错误的过程。在协议测试过程中既要模拟协议实现正常工作的情况,也要模拟异常使用的情况;既要模拟协议实现单独运行的情况,也要模拟协议实现之间互相通信的情况。这样才能确信在任何情况下系统的正常工作状况。由于对一个系统进行无穷尽的测试是不现实的,所以测试并不能保证一个协议实现的完全正确性,即测试只能表明“存在错误”,而不能证明“不存在错误”。这是协议测试的基本出发点。

协议测试是从软件测试的基础上发展而来。一般的软件测试可以分为功能测试和结构测试两种。结构测试又称为“白盒测试”,它是基于一个软件的内部结构进行的测试,其目的是通过执行每条语句、遍历程序的各个分支来检查整个程序代码的正确性。与此相反,功能测试只是依据软件的说明对从外部可以观察到的软件功能进行测试,所以也称之为“黑盒测试”,即被测试系统被看做是一个黑盒,仅仅通过观察外部行为来确定软件的功能实现,而不涉及程序的内部结构。功能测试的目的是根据软件的说明确定软件实现是否达到了要求,因此预先要有一个简捷明确的软件功能说明。而协议测试属于功能测试,即黑盒测试。

#### 7.1.2 协议测试的类别

目前,协议测试被人为地分成三种类型:一致性测试、互操作性测试和性能测试。其中,一致性测试是依据一个协议的描述文本对协议的某个实现进行测试,检测协议实现是否符合协议标准的要求;协议互操作性测试的主要任务是检测不同的协议实现之间是否能够可靠地进行通信交互,一个协议实现能否通过一致性测试和互操作性测试是它能否与其他实现成功

互通的重要保障；性能测试是测定协议实现的性能特性，如响应时间、时间延迟等，并据此对系统性能做出性能评价。在这三种测试中，协议一致性测试的发展相对来说最为成熟。

### 7.1.3 协议一致性测试的理论

目前，协议一致性测试的理论的几种表现体现在 ISO 制定的协议一致性测试标准 ISO/IEC-9646 中。这个标准是针对使用自然语言描述的协议的测试而制定的，最初由五个部分组成：

- ISO/IEC 9646-1: 基本原理；
- ISO/IEC 9646-2: 抽象测试集描述；
- ISO/IEC 9646-3: 树表结合表示法 TTCN；
- ISO/IEC 9646-4: 测试实现；
- ISO/IEC 9646-5: 一致性判定过程对实验室和客户的要求。

后来又补充了两个部分：

- ISO/IEC 9646-6: 侧面测试说明；
- ISO/IEC 9646-7: 实现一致性声明。

#### 1. 一致性测试的目的

ISO/IEC-9646 中对一致性定义如下：“一个一致性的实现应满足静态一致性需求和动态一致性需求，并与协议实现一致性声明（Protocol Implementation Conformance Statements，简称 PICS）中所声明的功能相符合。”一个一致性的实现应该满足所有在协议标准中显式表达的一致性需求，而一致性需求规定了一个一致性的协议实现应该做什么，不应该做什么。一致性需求一般可以分为以下三类：

- 必备要求（mandatory）：在所有例子中都可以观察到；
- 条件要求（conditional）：只有当标准中的特殊条件满足时才可以观察到；
- 可选要求（options）：由实现者来选择是否进行实现。

由于大量可选项的存在，每个协议标准实际上是描述了一类协议，因此各个协议实现的功能可能不完全一致，有些可选功能既可以实现，也可以不实现。因此，协议实现者应在 PICS 中列出所有已经实现的功能，从而使得测试人员知道可以进行何种测试。

在协议标准中定义的一致性测试需求分为两个基本组：静态一致性需求和动态一致性需求。其中，静态一致性需求规定了网络互连所规定的最小能力，对可选功能进行选择时应遵从的约束。它规定了一个协议实现所应提供的最低功能，以及不同可选功能之间的组合性与一致性。协议标准的主体由动态一致性需求构成，它定义了协议实现和外部环境进行通信时的可观察行为。它所关心的是可观察事件如协议数据单元（Protocol Data Unit，简称 PDU）或抽象服务原语（Abstract Service Primitive，简称 ASP）的发送/接收的顺序，PDU 中信息的编码以及不同 PDU 中所含信息的关系等。

对于静态一致性需求，可以通过对 IUT 的“协议实现一致性声明”PICS 进行检查来完成，这也称为“静态一致性检查”。对于动态一致性要求，则需通过一组组的测试来完成。一个测试称为一个“测试例”，测试集则是所有测试例的集合。测试集的结构是层次化的，测试例以测试组的形式组合在一起构成测试集。因此又可以对动态一致性测试进行这样的划分：

- 基本互连测试: 对测试系统与 IUT 之间的基本互连能力进行测试, 以确定是否需要进一步的行为测试;
- 能力测试: 确定 IUT 是否实现了 PICS 中所声明的功能;
- 行为测试: 检测 IUT 的动态行为是否与协议标准中的描述相一致;
- 一致性判定测试: 指在未定义标准测试集的情况下, 专门对系统的某个特定行为进行的测试, 这类测试尚未标准化。

## 2. 一致性测试过程

一致性测试过程大致可以划分为三个阶段。第一个阶段是根据某特定协议文本得到一个独立于所有协议实现的一致性抽象测试集, 称之为“一致性测试生成”。第二个阶段由实现特定一致性测试集的执行方法组成, 也称做“一致性测试实现”。第三个阶段为“一致性测试执行”, 运行已经具体化的一致性测试例, 并对 IUT 的外部行为响应进行观察, 最终可以得到测试判决, 并将测试执行的结果记录在协议一致性测试报告中。

## 3. 抽象测试方法

ISO/IEC-9646 根据不同的控制观察点定义了不同的抽象测试方法, 认为现有的端系统抽象测试方法基本上可以分为四类, 即本地测试法、协调测试法、分布测试法和远程测试法。每种端系统抽象测试方法可以用两个抽象测试功能描述, 即由测试协调过程联系起来的上测试器 (Upper Tester, 简称 UT) 和下测试器 (Lower Tester, 简称 LT)。IUT 则位于一层或多层已经过测试的协议实体之上 (以下称之为服务提供者)。端系统抽象测试方法的选择依赖于被测协议实现的上、下边界的可访问性。

本地测试法 (Local Method) 是端系统协议一致性测试中的最简单也是最基本的方法。在本地测试法中, LT、UT 以及测试协调过程是在同一个系统中实现的。这种方法的一个基本假设是在 IUT 的上、下边界都存在暴露的访问接口, 这些接口也称为控制观察点 (Point of Control and Observation, 简称 PCO), 测试系统通过这些 PCO 对 IUT 的输入激励并且观察 IUT 的输出响应, 然后根据协议描述做出测试判决。在本地测试法中的 LT 和 UT, 分别对应于 IUT 的上、下接口。LT 和 UT 的作用是在 IUT 的上、下接口处通过交换测试事件来观察 IUT 的行为, 测试协调过程用于协调 LT 和 UT 的动作, LT 同时记录所有的测试事件以作为测试判决的依据。

当端系统协议一致性测试是由独立的第三方进行, 或者测试系统与 IUT 位于不同的地点, 经常无法对 IUT 的下边界进行直接的访问。所以 ISO9646 又定义了三种端系统外部测试方法: 分布式测试法、协调式测试法和远程测试法。

在这三种外部测试方法中, 假定 IUT 的下边界不存在 PCO, 对 IUT 的激励和观察功能分成本地和外部两个部分。它们的共同特点是 LT 和 IUT 处于两个不同的系统之中, 二者之间通过提供  $(N-1)$  层服务的底层协议实现连接。在采用外部测试方法时, 为了进行测试判决, 就需要首先承认如下假设: 当测试  $(N)$  层实体时,  $(N-1)$  层实体向上提供的服务是可靠的。不承认这一假设, 一致性测试就是没有意义的, 因为这样就不能界定测试例的失败是由于 IUT 的错误行为还是  $(N-1)$  提供的服务出现错误。所以, 此时一致性测试的理想方式应当是分层的, 从最低层开始测试, 由底层测试的正确性保证底层提供服务的正确性, 从而保证上层测试判决的正确性。但是, 一致性测试的完备性是难以做到的, 所以一致性测试的判决难以证明其完全的正确性。

在分布式测试方法中, LT 和 IUT 处于两个不同的系统之中, 二者之间通过提供  $(N-1)$  层服务的底层协议实现连接, LT 是 IUT 的对等实体。UT 与 IUT 位于同一个系统内, 它要求 IUT 有暴露的上边界访问接口。和本地测试法相比较, 分布式测试法有以下几点不同:

- 必须修改基于本地测试法所设计的抽象测试集, 以包含 LT 所使用到的底层抽象服务原语, 而这些抽象服务原语和本地测试法中所用到的是互补的。
- LT 和 IUT 物理上是分离的, 因此对同一测试事件的观察变得不再同步。
- 由于 LT 和 IUT 的通信是通过  $(N-1)$  层服务实现的, 因此存在测试数据丢失、失序和出错的可能性。
- 由于 LT 和 UT 分布在不同的系统, 二者之间的同步和控制(即测试协调过程)比本地测试法要困难得多。

在分布式测试法中, 测试协调过程可以用在 PCO 处交换的 ASP 来说明, 也就是说 LT 和 UT 之间的协调是通过 IUT 实现的, 所以测试判决是基于 LT 所观察到的行为而做出的。

协调测试法是最复杂的一种端系统抽象测试方法, 它要求在 LU 和 UT 之间通过传送测试管理协议数据单元(Test Management Protocol Data Unit, 简称 TM-PDU)建立起高度的测试协调, 并在 LT 和 UT 中对所收集和观察到的信息进行报告。LT 和 UT 之间的通信可以通过被测协议实现传送 TM-PDU, 也可以使用一个足够可靠的底层协议来传送 TM-PDU。

协调测试法与分布式测试法的区别主要有两点:

- IUT 的上边界不一定要有暴露的访问接口;
- 采用标准化的测试管理协议(Test Management Protocol, 简称 TMP)和 TM-PDU 进行自动的测试管理和协调。一般采用 LT 作为主控方, UT 作为被控方的设计方法以减少 UT 的实现工作量。

到目前为止, 协调测试法还有以下不足之处:

- 未定义出独立于任何应用的标准化的测试管理协议;
- 未对使用被测协议实现还是足够可靠的底层协议传送 TM-PDU 提出建议, 在目前已有的实现方法中, 有许多基于 TCP/IP 协议实现方式的 TM-PDU 传送, 以用于高层协议的一致性测试。

远程测试法的主要特点是并不要求能访问 IUT 的上边界, 也不要求显式地测试协调过程。这种方法依赖于被测协议来实现 IUT 和 LT 间的同步。远程测试法所采用的一个假设是 IUT 的状态可由 LT 通过  $(N-1)$  层服务与之交换  $(N)$  层 PDU 来确定。测试判决则是由基于下测试器对 IUT 提供的激励以及下测试器所观察到的 IUT 的响应做出的。远程测试法在低层协议一致性测试中应用得较为广泛。

ISO/IEC-9646 针对中继系统的特点——没有开放的上下接口, 提出了两种中继系统抽象测试方法: 回绕法和穿越法。回绕法的缺点是被测中继系统只有一端的行为被直接观察到, 而另一端的行为不能被正确地评价。而穿越法则使被测中继系统在平常的操作模式下得到测试, 在两端的行为都能够被观察到。

#### 7.1.4 抽象测试集和 TTCN 简介

测试集可以说是一致性测试的核心, 一致性测试的所有活动都从测试集起步。测试集一般被划分成几个层次, 依次是测试组、测试例、测试步、测试事件, 测试例是最重要的一个

层次。每一个测试例都有自己的测试目标，并根据 IUT 的响应对其行为做出判决后得出结果。

所谓抽象测试集，就是由抽象测试例组成的测试集合，每个抽象测试例都是针对某个测试目的使用某种抽象测试方法定义的完整而且独立的测试步骤。抽象测试例是独立于任何测试系统的。ISO/IEC-9646 中建议使用 TTCN 作为测试集的描述方法。TTCN 和其他描述方法具有以下优点：

- TTCN 所采用的表示方式非常易于理解和学习；
- 形式描述语言的设计和开发目的是用于系统或过程的行为描述，所以当使用这些语言去描述测试集时，就显得有些复杂和不太灵活。而 TTCN 用于测试集的描述，非常方便和灵活；
- TTCN 是一种抽象测试描述语言，完全独立于任何特定的系统；
- 形式描述语言非常适合于类型说明，但是在协议测试集中要描述大量的测试数据，即要说明大量的值，这正是形式描述语言的弱点，而 TTCN 却非常擅长于数据值的描述；
- TTCN 的结构和语法完全是为描述一致性测试过程中各个实体间的交互事件而设计的，由静态说明部分和动态行为描述部分组成。其静态说明部分用于协议的静态信息描述，如协议数据单元和抽象服务原语的格式和参数值。为了支持标准化的一致性测试方法和框架，TTCN 还支持 PCO、定时器、测试集变量和测试例变量的描述；其动态行为描述部分提供了同消息的接收和发送相关联的操作与操作符。

TTCN 作为一种测试表示方法，完全满足了 ISO/IEC-9646 中对测试集描述语言的需求。

TTCN 可以达到以下目的：

- 提供用标准测试集表述法表述抽象测试例的方法；
- 提供独立于测试方法、层和协议的表示方法；
- 提供符合 ISO/IEC-9646 中定义的抽象测试方法的表示方法；

对应于标准的一致性测试方法和框架，使用 TTCN 描述的测试集由四部分构成，它们分别是：测试集概述部分、说明部分、约束部分和动态行为描述部分。

- 测试集概述部分定义了测试集的名称和测试集所对应的被测协议标准、PICS、协议实现用于测试的附加信息（Protocol Implementation eXtraInformation for Testing，简称 PIXIT）以及所使用的测试方法等用于理解和表示整个测试集的一般信息。
- 说明部分提供了测试集中所引用到的所有对象（如 PCO、时钟、ASP、PDU 的名称、类型、取值范围等）信息。
- 约束部分定义了协议数据单元中各个域或抽象服务原语各个参数的各种特定取值，用于在测试集的动态行为描述部分对测试事件进行约束，以简化测试事件的描述。
- 动态行为描述部分是 TTCN 抽象测试集的主体，它包含三个库即测试例库、测试步库和默认行为库，分别用来描述测试例、测试步和默认行为。在动态部分中，使用了三种类型的 TTCN 语句对测试行为进行描述：构造语句、事件语句和伪事件语句。对这些 TTCN 语句的解释和执行是进行实际测试活动的核心部分。

### 7.1.5 协议一致性测试的研究现状

协议一致性测试从 20 世纪 60 年代开始受到研究人员的重视，并投入了大量的人力物力从事这方面的研究。许多计算机厂家和实验室纷纷推出自己的测试系统和测试仪。例如，加



拿大 IDACOM 公司的 OSI 协议一致性测试仪；美国 Alcatel TITN 公司的 OSI 协议一致性测试系统 XRTLE；美国 Alcatel 公司的 OSI 协议一致性测试仪系列；美国惠普公司的测试仪系列；法国的一家实验室的跨式测试系统以及柏林的 GOR 科学研究院的计算机技术与信息协会开发的 TEKOS 测试系统等。

7.2 SIP 测试现状

7.2.1 ETSI

日前，欧洲电信标准协会（ETSI）在法国的 Mandelieu 召集了来自 60 多家公司的 130 多名工程师进行了会话初始协议互用性测试（SIPit）。这是第三次进行此类测试。

此次测试的主要内容为测试 SIP 用户主体、代理服务器、软件、电话、IP 网关、语音信箱系统、协议检测器等设备的 SIP 功能性。测试的形式多为对等测试和多方测试。有近 30% 的测试设备都是新产品。

此次测试的目的是确定在哪些情况下使设备不能够互通，从而可以改进标准的制定。

7.2.2 SIP 论坛

SIP 论坛周期性地进行 SIP 互操作性测试，称为 bake-offs，首届 bake-offs 于 1999 年在哥伦比亚大学举行，现在通常每年举行 3 次，SIP bake-off 用来找出协议的缺陷。其图标如图 7-1 和图 7-2 所示。



图 7-1 SiPit 标志



图 7-2 SIMPLEt 标志

7.2.3 SIP 协议互通测试结果

如表 7-1 所示为 SIP 协议互通测试结果。

表 7-1 SIP 协议互通测试结果

产品	基本互操作性	高级互操作性		
SIP 用户代理	UA 到 UA 的直接呼叫, 无代理服务器 UA 到 UA, 通过 SIP 代理服务器 UA 到模拟电话, 通过 SIP 代理服务器	Call transfer	会议呼叫	呼叫前转
Cisco ATA-186 telephony adapter	全部成功互操作	成功互操作	成功互操作	尚未测试
Cisco 7960 SIP phone	全部成功互操作	成功互操作	成功互操作	尚未测试
Difinium Mercury	全部成功互操作	成功互操作	成功互操作	成功互操作
Indigo Software SIP User Agent	全部成功互操作	尚未测试	尚未测试	成功互操作
Mediatix APA III-4FXS	全部成功互操作	成功互操作	成功互操作	成功互操作
CyberTel CyberCom Server Class	全部成功互操作	尚未测试	尚未测试	尚未测试

续表

产品	基本互操作性	高级互操作性			
SIP 代理服务器	UA 到 UA (通过 SIP 代理服务器) UA 到 UA (通过多个 SIP 代理服务器) UA 到模拟电话(通过基于 SIP 的 PSTN 网关)	传输协商	顺序分叉	并行分叉	回路检测
Indigo Proxy Server	全部成功互操作	成功互操作	成功互操作	成功互操作	成功互操作
Mockingbird SIP server	全部成功互操作	成功互操作	成功互操作	尚未测试	成功互操作
NetCentrix call control server	全部成功互操作	尚未测试	成功互操作	尚未测试	尚未测试
SS8 Signaling Switch	全部成功互操作	成功互操作	尚未测试	成功互操作	成功互操作
Vovida.org Vocal	全部成功互操作	成功互操作	成功互操作	尚未测试	尚未测试
SIP-to-PSTN gateways	UA 到模拟电话 (通过 SIP 代理服务器) SIP/IP trunking, analog phone (通过 PSTN 网关) PRI trunking, UA 到 UA (通过 PSTN 网关)	编码协商	传输协商 (TCP-UDP)	传真	Reinvite 方法
Cisco AS5350 Universal 网关	全部成功互操作	成功互操作	成功互操作	成功互操作	成功互操作
Mediatix APA III-4FXO	第三种尚未支持, 其他成功互操作	成功互操作	尚未支持	尚未测试	成功互操作
Mockingbird Nuvostream multiprotocol server	全部成功互操作	尚未测试	尚未测试	尚未测试	尚未测试
Nuera ORCA GX-8	全部成功互操作	成功互操作	尚未支持	成功互操作	成功互操作

注: 测试中使用 Pingtel xpressa SIP 电话和 dynamicsoft Session Management Suite SIP 代理服务器。

## 7.2.4 可用于测试的公开 SIP 服务器

表 7-2 为公开的, 可用于测试的 SIP 服务器。

表 7-2 SIP 服务器

组 织	SIP URL 地址	传 输	模 式
hotfoon	sip:hotfoon.com	UDP	redirect
iptel.org	sip:yourname@iptel.org	UDP	redirect 和代理
Lucent Technologies, Inc.	sip:www-db.research.bell-labs.com	UDP, TCP	redirect, proxy
Centre of Network Research, Computer Engineering Department, Prince of Songkhla University, Thailand	sip:cnrsipserver.coe.psu.ac.th	UDP, TCP	redirect, proxy, default is redirect
Vienna University of Technology, Institute of Communication Networks	sip:kalypso.iikn.tuwien.ac.at	UDP, TCP	proxy
zdots (Microappliances)	sip:zdots.com	UDP, TCP	proxy
Nortel sipfx	sip:sipfx.com	UDP, TCP	redirect 和代理
sipcenter (Ubiquity)	sip:sipcenter.com	UDP, TCP	Redirect 和代理
3Com siphappens	sip:siphappens.com	UDP only	Redirect 和代理
MCI SIP interoperability test web	sip:sipaccount.mci.com	UDP, TCP	Redirect 和代理

## 7.3 小结

本章首先介绍了协议测试的基本理论，然后介绍了 SIP 测试的相关组织及互操作测试结果，最后介绍了一些可以公开的用于测试的 SIP 服务器。目前关于协议测试及 SIP 协议测试，仍然有很多理论在进展之中。

## 7.4 习题

1. 简述协议测试的基本理论。
2. 简述 SIP 协议测试的相关组织。
3. 简述 SIP 协议测试的现状。

## 第 8 章 SIP 业务开发及部署

知识点:

- SIP CGI
- SIP Servlet
- CPL
- 开发实例
- SIP 业务部署

传统的电话网络使用智能网技术 (IN, Intelligence Network) 以方便网络运营商添加业务而无须变动承载业务的电话网络。与之相对应的 Internet 必须有相应的机制来保证新的业务能够被很快地生成和部署。本章将介绍在 SIP 协议环境下业务创建的相关机制: 呼叫控制语言 (CPL: Call Processing Language)、SIP CGI (公共网关接口, Common Gateway Interface)、SIP Servlet, 另外也介绍了和 SIP 应用相关的 J2EE 技术和 Web 服务技术, 最后还阐述了 SIP 业务部署。

### 8.1 SIP CGI 技术

众所周知, CGI 并不是一项新的技术。它最初的出现是和 HTTP 协议一起, 使静态网页具有了与浏览者交互的功能, 极大地推动了 Web 技术的发展。CGI 技术提供了一种在 Web 环境下创建新业务的机制。从 Web 的发展来看, 无疑 CGI 技术又是非常成功的。它具有以下一些特点:

- 语言的独立性; CGI 技术定义的是接口而非具体的编程语言, 它可以与 perl、C、VisualBasic、tcl 等语言一起工作使用;
- CGI 技术允许 CGI 的应用程序对消息的头部 (header) 具有访问权限;
- 产生应答的方式与其他的一些技术, 如 Java servlets 不同。CGI 处理应答的所有部分, 包括头部、应答码、消息体。

SIP 协议在发展的过程中很大程度上借鉴了 HTTP 协议, 例如, 消息的格式、消息请求/应答的交互方式, 用文本格式对消息进行编码等。所以 CGI 规范的特点也特别适合 SIP 协议。因此可以采用 CGI 作为 SIP 的一种业务创建 (Service Creation) 的机制。

#### 8.1.1 SIP CGI 与 HTTP CGI 的区别

尽管 SIP 与 HTTP 具有相似的语法和请求/应答的模型, 但是它们还是有一些关键的不同点。如代理服务器在 SIP 协议中起着重要的作用, 而在 HTTP 协议中则不是很重要; SIP 协议提供了注册的机制, 而 HTTP 协议没有。这些不同点也反映了 SIP CGI 和 HTTP CGI

的区别。

- SIP CGI 主要运行在代理服务器、重定向服务器、注册服务器上；而 HTTP CGI 则主要运行在用户代理 (user agent) 上；
- SIP CGI 允许脚本执行代理转发、重定向等操作；而 HTTP CGI 则不支持；
- SIP CGI 支持持续性模型 (persistence model)，通过一些机制使得允许脚本在 SIP 事务所包含的所有消息中保持状态；而 HTTP CGI 并不支持该模型。

### 8.1.2 SIP CGI 规范

与 HTTP CGI 类似，SIP CGI 规范主要定义了一些机制来使用户能够运行服务器上的应用程序。服务提供商可以使用 C、visual Basic、perl 等语言来开发应用程序，实现不同的业务。

#### 1. SIP CGI 脚本如何从服务器获取数据

SIP CGI 对于 SIP 请求或者应答消息的头部和消息体采用了不同的方式。服务器采用操作系统环境变量的机制来传递消息的头部信息。RFC3050 定义了超过 20 个的环境变量，例如，AUTH\_TYPE、REMOTE\_USER 等。而对于消息体，服务器则采用标准输入文件描述符 stdin 来传递。

#### 2. CGI 应用程序处理结果的输出格式

CGI 脚本应用程序输出有一些消息组成，这些消息分别对应了脚本要求服务器将要执行的动作。每条消息由一个动作 (action)、多个 CGI 头部和多个 SIP 头部组成。RFC 3050 中定义了 Status, CGI-PROXY-REQUEST, CGI-FORWARD-RESPONSE, CGI-SET-COOKIE, CGI-AGAIN 5 种动作，同时还定义了 CGI-Request-Token 和 CGI-Remove 两种 CGI 头部。消息中所涉及的 SIP 头部语法同 RFC2543。

## 8.2 CPL

CPL 是 IETF 的 IPTEL 工作组提出的一种用于描述和控制 IP 电话呼叫过程的语言，它用一个 CPL 脚本文件来描述业务逻辑。脚本文件可以通过不同方式创建，如使用文本编辑器、图形工具等。脚本描述的业务逻辑可以对应于一个或一组地址，此指定地址被称为脚本拥有者，它可以是主叫方，也可以是被叫方，但一个脚本中只能描述该地址作为其中一方的行为。

一个典型的 CPL 脚本如图 8-1 所示。

该脚本描述了一个遇忙/无应答呼叫前转业务逻辑，该业务逻辑属于某个地址，当呼叫此地址时，执行该业务逻辑。可以看到，CPL 脚本由一系列标记组成，标记分为两种：节点（如 cpl, incoming, location, proxy）和节点的输出（如 busy, noanswer 是 proxy 节点的输出），所有标记组成一棵业务逻辑树，如图 8-2 所示。脚本逻辑由一个共同的根节点 (cpl) 开始。当 CPL 脚本运行时，先执行树中的根节点，根据节点执行的结果，服务器判断节点的输出是什么，然后执行此输出指向的下一节点，直至到达叶节点为止。

```

<cpl>
  <incoming>
    <location url="01012345678">
      <proxy timeout="8">
        <busy>
          <location url="01087654321">
            <proxy />
          </location>
        </busy>
        <noanswer>
          <location url="01087654321">
            <proxy />
          </location>
        </noanswer>
      </proxy>
    </location>
  </incoming>
</cpl>

```

图 8-1 CPL 脚本：遇忙/无应答呼叫前转

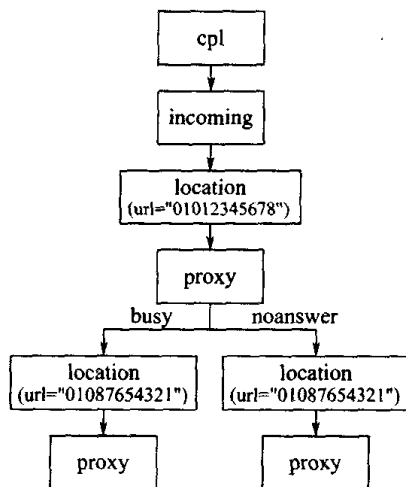


图 8-2 遇忙/无应答呼叫前转业务逻辑树

下面将详细介绍 CPL 语言的结构以及如何部署，运行 CPL 脚本。

### 8.2.1 CPL 脚本结构

CPL 语言由一系列标记（tag）组成，各个标记按树形结构排列。按照标记包含语义的不同，CPL 标记可分为两类：节点和输出；节点类标记规定了某种行为或做出某种判断（如前文所述的 cpl、incoming、location、proxy 标记），输出类标记则指规定行为的后果或判断标记的条件分支（如 busy、noanswer）。一个节点可以有零个、一个或多个输出，无输出的节点后直接跟子节点，而有输出的节点的子节点包含在各个输出中，每个输出只能有一个子节点（如 proxy 节点可以有 busy、noanswer 等输出，每个输出后包含了一个 proxy 的子节点 location。而 cpl 节点没有输出，直接跟 incoming 子节点）。节点可具有一些属性，用以对此节点的动作或判断行为作一定的条件规定。

CPL 脚本包含两种信息：呼叫处理行为（call processing action）和辅助信息（ancillary information），呼叫处理行为用以描述信令服务器对一个呼叫建立事件的判断和操作；辅助信息是服务器正确处理脚本时必须的一些信息，CPL 至今未定义任何辅助信息。

呼叫处理行为包括顶级行为和次级行为两种，顶级行为由到达服务器的信令事件触发。CPL 定义了两种顶级行为：incoming 和 outgoing，当脚本拥有者为呼叫被叫方时触发 incoming 行为；当脚本拥有者为呼叫发起方时触发 outgoing 行为。次级行为是被其他行为调用的行为，CPL 禁止递归调用次级行为，次级行为用标记<subaction>表示。

呼叫处理行为是以一系列节点来描述的，这些树状结构的节点描述了业务逻辑中的各个分枝及各种行为。节点可以有多个属性参数，精确规定节点的含义。一个节点可以有多个分支节点，具体执行那个分支节点由行为的结果或判断条件而定。所有 CPL 脚本的根节点都是“cpl”，cpl 下面可以跟 4 种节点：incoming、outgoing、subaction 以及 ancillary，其中 incoming、outgoing 只能出现一种；由于 ancillary 事实上还未定义，因此实际的脚本中 cpl 的子节点只可能为前三种。无论是顶级节点 incoming、outgoing 还是次级节点 subaction，其子节点都由转换节点、地址信息节点、信令节点和非信令节点组成。

#### 1. 转换节点

转换节点（switches）的含义是指可以根据呼叫的某种属性或其他与呼叫无关的某种条件，

与该节点的各种输出分支进行匹配, 选择符合条件的子节点。匹配按分支在脚本中的顺序进行, 当遇上第一个匹配的条件分支时停止继续匹配, 开始执行该分支指向的下一个节点。每个 `switches` 节点都有两种特殊的输出分支: `not-present` 和 `otherwise`。`not-present` 表示呼叫请求中未出现脚本规定的匹配变量信息。`otherwise` 的含义是: 如果没有可以匹配的分支, 则匹配该输出; 由于匹配按每个分支在脚本中的顺序进行, 则 `otherwise` 只能作为最后一个分支。如果脚本中未找到匹配的条件选项, 且未规定 `otherwise` 条件选项, 则脚本应执行某种默认的行为。

`switches` 节点有以下几种:

- 地址选择 (`address-switch`): `address-switch` 允许 CPL 脚本根据呼叫请求中的某个地址信息做出选择。
- 字符串选择 (`string-switch`): `string-switch` 允许 CPL 脚本根据呼叫请求信息中出现的字符串做出选择。
- 时间选择 (`time-switch`): `time-switch` 允许 CPL 脚本根据脚本执行时的时间做出选择。
- 优先级选择 (`priority-switch`): `priority-switch` 允许脚本根据规定的呼叫优先级进行选择。
- 语言选择 (`language-switch`): `language-switch` 允许脚本根据呼叫发起方期望用以交流的语言进行选择。

## 2. 地址集修改节点

CPL 脚本执行时必不可少的一类信息是呼叫地址, 在整个呼叫过程 (包括次级行为) 中将可能涉及的地址存放在同一个地址集中。地址集中的地址信息可以在脚本中被添加、删除, 也可从外部资源中查找得到。为了对地址集的地址信息进行添加、删除, CPL 定义了以下三种地址集修改节点 (`location modifier`):

- 显式添加地址节点 (`location`): `location` 节点在地址集中加入一个或多个指定的地址。
- 地址查找节点 (`lookup`): `lookup` 节点从外部资源中获得地址信息。
- 地址过滤节点 (`remove-location`): `remove-location` 节点按照一定条件从地址集中删除一个或多个地址。

## 3. 信令节点

信令节点引起底层协议的信令事件, CPL 定义了 3 种信令操作: `proxy`, `redirect` 及 `reject`。

`proxy` 节点的作用是将呼叫目的地指向地址集中的某一个或多个地址, 将地址集中的那个地址作为指向地址由 `proxy` 节点的属性参数决定。执行 `proxy` 节点将向服务器发起呼叫连接请求的信令消息, 在网络对此次呼叫连接请求的响应到达之后, 按照响应类型决定下一步行为。由于到达服务器的响应可能有多个, 例如, 连接成功后网络会向服务器发出一个连接成功信息, 用户应答时又会发出一个用户已应答信息, 因此, 在一次呼叫请求的多个响应中, 服务器选择其中“最好”的一个响应 (什么是“最好”的响应由底层协议或服务器管理配置规则规定) 作为本次连接的响应。如果呼叫请求成功, 则 CPL 脚本停止执行, 服务器进行默认的后继处理, 通常是允许呼叫建立。否则, 按照响应类型从 `proxy` 节点的输出中选择一个, 执行该输出指向的节点。

`proxy` 的输出可以为 `busy`, `noanswer`, `redirection`, `failure` 及 `default`。如果呼叫响应为忙, 执

行 busy 输出指向的节点；如果呼叫一定时间后仍无应答，则执行 noanswer 输出指向的节点；如果呼叫被重定向，则执行 redirect 输出指向的节点；如果呼叫连接因为任何原因失败，则执行 failure 输出指向的节点；如果呼叫响应为以上输出中的一种，但脚本中未规定此输出，则执行 default 输出对应的节点；如果脚本未规定 default 输出，则 CPL 停止执行，服务器执行默认行为；如果执行 proxy 行为时地址集中没有地址，或者其中的地址不能被代理，即不能按信令请求建立连接（如“http”地址），则执行 failure 输出。

一旦 proxy 操作完成，控制权转移到其他节点，所有被尝试建立呼叫连接请求的地址都应从地址集中清除。在 proxy 输出 redirection 的情况下，被重定向的新地址必须加入到地址集中。

redirect 节点使服务器将呼叫发起方的呼叫重定向到地址集中的地址。redirect 节点后无输出，也无子节点，重定向操作后整个 CPL 脚本结束执行。

reject 操作使服务器拒绝呼叫尝试并立即结束整个 CPL 脚本的执行，reject 节点后无输出，也无子节点。

信令操作节点引起底层信令协议的信令事件，虽然对任何协议而言，proxy、redirect、reject 节点的含义是相同的，但对不同的信令协议服务器，具体的操作是不同的。

#### 4. 非信令节点

非信令节点 (Non-singalling operations) 将触发一些与底层协议无关的行为，主要包括 mail 节点、log 节点及次级行为定义节点 sub-action、次级行为引用节点 sub。

mail 节点的作用是用电子邮件通知用户 CPL 脚本的状况。由于电子邮件发送是否成功不可能实时得知，因此 mail 节点最多只能有一个可能的子节点，不管 mail 节点执行的结果是什么，业务逻辑都将直接执行下一个节点。

log 节点将呼叫信息写入日志，服务器不能让 log 操作失败，因此，log 节点也最多只能有一个子节点。

为了能用 CPL 更好地描述呼叫流程，同时也为了脚本的重用性和模块化，CPL 中定义了次级行为。定义和引用次级行为需要使用两种节点：sub-action 及 sub。sub-action 节点定义次级行为，此行为将被位于顶级行为中的 sub 节点调用。脚本执行到 sub-action 时不进行任何操作，而是转而执行顶级行为中的下一个子节点，sub-action 中规定的呼叫行为在被引用时才真正加以执行。sub 节点的功能是引用一个已定义过的次级行为，sub 节点不可引用顶级行为、自身作用域中定义的行为或任何在其后定义的行为，也不可引用其他脚本中定义的次级行为。脚本验证时必须检查是否有不合法的 sub 节点引用。

### 8.2.2 CPL 框架结构

在本小节将主要介绍 CPL 的网络模型，CPL 脚本如何在网络上部署、运行，CPL 脚本能描述的业务特征是什么，最后简单介绍 CPL 业务的特征交互问题。

#### 1. CPL 的网络模型

CPL 脚本运行在 Internet 电话网中，CPL 的网络模型包含两种网络组件：终端系统和信令服务器。终端系统是指可以发起和/或接收信令消息及媒体流的设备，如普通电话、SIP 终端、PC 电话客户端等。终端系统可以发起呼叫，并可以接收、拒绝、前转来电。信令服务器是指可以中转或控制信令信息的设备。在 SIP 中，信令服务器指代理服务器 (proxy server)、重定向服务器 (redirect server) 及注册服务器 (registrar)；在 H.323 中，信令服务器指网守



(gatekeeper)。信令服务器可以对呼叫建立信息做三种处理：代理、重定向和拒绝。代理是指将呼叫转移到一个或多个其他网络或终端，并返回一个收到的响应信息。重定向是指向发起请求的系统返回一个新的地址信息，指示系统应该按向地址发出请求。拒绝是指向发起请求的系统返回消息，指示呼叫建立请求被拒绝，不可完成。信令服务器通常包含用户位置信息。不管是通过注册、静态配置、动态搜索方式，信令服务器可以知道用户的位置，以决定对代理或重定向行为做出智能化选择。

当终端发起呼叫时，呼叫建立请求可能选择不同路由，经过网络上的不同组件。因此，一个呼叫建立过程可以被多个信令服务器控制。

## 2. CPL 业务的运行

成功地创建一个 CPL 业务需要经过创建脚本、验证脚本、上载脚本及运行脚本等步骤。脚本的创建及语法验证由脚本开发者完成，创建 CPL 脚本的设备与 CPL 业务的呼叫处理设备无任何关系。CPL 脚本在创建后需要传输到服务器的脚本库中，脚本上传方式包括 Web 文件上载、SIP 注册服务器信息上载、SNMP 等，上载到脚本库的脚本会一直驻留，除非脚本创建者、有权限的管理员对脚本进行修改、删除，或者脚本超出了有效期。脚本拥有者可以通过网络查找到自己的脚本，进行修改并重新上载。

CPL 脚本运行在信令服务器上。通常，信令服务器负责控制呼叫建立过程，它维持一个用户位置信息的数据库，根据呼叫请求及数据库中的信息来判断应该对呼叫做出 proxy、redirect 或 rejection 响应。但运行 CPL 脚本时，脚本代替了信令服务器中的用户定位功能，服务器不需查询数据库，而是按脚本执行相应行为，如果脚本没有为某呼叫请求规定相应的行为，则信令服务器按正常情况处理该呼叫请求。

一个 CPL 脚本一般与一个特定的地址相关，脚本可以运行在对此地址的呼叫建立请求必须经过的任一网络服务器上。CPL 脚本也可以与一个地址集合对应，这样的脚本包含的业务逻辑对地址集中所有的地址生效，但此时用户仍可为地址集中某个地址定义特定的脚本并执行，在这种情况下，两个脚本之间可能产生业务特征冲突问题。

一个呼叫建立到达信令服务器时，按照请求中的源地址和目的地址，服务器从 CPL 脚本数据库中寻找出匹配的脚本执行，如果在执行过程脚本过程中改变了呼叫目的地址从而触发了其他脚本，则服务器按新的目的地址查找与新地址匹配的脚本来执行，在此过程中也可能产生业务特征冲突的问题。

## 3. CPL 的业务特征交互

业务特征交互是电信常用术语之一，它是指两个或两个以上的特征行为导致产生了二义性或行为冲突。CPL 中的特征冲突大致可以分为三种：业务特征与业务特征间的冲突、脚本与脚本间的冲突、服务器与服务器间的冲突。

### 1) 业务特征与业务特征间的冲突

特征与特征间的冲突是指同一个 CPL 脚本中实现不同业务特征而产生的冲突，在 CPL 脚本中，业务特征被描述为一系列的条件——行为组合，在一定条件下，只能执行其中的一个行为。如传统电话业务中呼叫遇忙等待和呼叫遇忙前转是两个冲突的业务特征，但在 CPL 脚本中，对于呼叫遇忙事件只能为它定义一个相应行为，因此，一个 CPL 脚本不存在业务特征之间的冲突。

## 2) 脚本与脚本间的冲突

由于一个呼叫可能会触发多个脚本的执行, 因此产生了脚本与脚本间的冲突。一个呼叫触发多个脚本的情况有几种:

- 在一个脚本的执行中由于发生呼叫前转而触发另一个脚本。
- 一个呼叫的源地址和目的地址分别对应了一个 CPL 脚本。
- 个人用户对某个地址定义了个人脚本, 管理员又对包含此地址的地址集定义了一个公共的脚本。

这些情况下都可能发生脚本间的冲突。脚本间冲突包括脚本执行顺序的冲突及脚本中定义的业务特征上的冲突两种情况。对于执行顺序上的冲突, 解决冲突的关键在于决定脚本执行的顺序。对于呼叫前转触发的多个脚本, 执行顺序是显而易见的: 先执行第一个触发的脚本, 当目的地被转移到另一地址并触发第二个脚本时, 则执行第二个脚本, 并将执行的最终结果作为呼叫响应报告给第一个脚本, 这个响应被看成与网络发出的普通呼叫响应相同, 第一个脚本按此响应对应的行为继续执行。

对于同一个呼叫用户分别为源地址和目的地址制定了 CPL 脚本的情况, IETF 的 CPL 框架结构标准中规定, 先执行源地址用户的脚本。如果源地址用户在对地址进行了号码翻译、呼叫过滤等功能后呼叫目的地址被改变, 则按新的目的地址发起呼叫请求并执行与新目的地址相应的脚本。如果呼叫目的地址未被改变, 服务器继续执行目的地址用户指定的脚本。

在脚本间冲突的三种情况中, 最复杂的是第三种情况。管理员脚本与个人脚本之间的执行顺序与脚本中包含的业务逻辑有关, 例如, 管理员脚本中对呼叫源地址及目的地址做了某种限定, 则它应该在源地址的个人脚本执行之后, 目的地址的个人脚本执行之前执行, 以避免用户通过适当的呼叫前转来躲过管理员脚本的限定。对于允许管理者为多个地址制定脚本的服务器来说, 服务器应该允许管理者来配置脚本执行顺序, 或配置为在某些情况下不执行管理者脚本。

对与多个脚本间业务特征的冲突, 由于解决脚本执行顺序后, 可以看成多个服务器上脚本间的冲突, 因此也可看成为服务器与服务器间的冲突。

## 3) 服务器与服务器间的冲突

服务器之间的冲突是几种冲突中最难解决的, 一个典型的业务冲突例子是呼叫前转和呼叫屏蔽, 呼叫屏蔽需要对来电的源地址进行检查, 拒绝某些源地址的呼叫, 但如果呼叫发起者采用了呼叫前转, 则通过一个前转地址将呼叫接续到要求呼叫屏蔽的目的地, 此时服务器不能正确地判断是否应该拒绝此次呼叫, 呼叫屏蔽可能未失败。这种业务冲突一直未能找到一个很好的解决方案, 即使在传统的电路交换网中, 这个问题也未能解决, CPL 业务中也存在这个问题。

对于另一种服务器间的冲突, 如用户 A 将呼叫前转到用户 B, 用户 B 又将呼叫前转到用户 A, 不管是否是通过 CPL 脚本来控制服务器的信令操作都会导致这种冲突的产生, 因此最好由网络协议来解决这个问题, SIP 协议就能检测出这样的循环并采用一定机制加以处理。虽然支持 CPL 的服务器不需要为此制定专门的机理, 但是, 一旦类似的循环被检测出, 服务器就应不再继续执行脚本, 并可通过某种出错报告方式向脚本拥有者报告错误。

## 4. CPL 服务器的默认行为

在某些情况下, CPL 脚本可能并未对出现的条件指定响应行为, 此时运行 CPL 的服务器

应该按脚本当前的执行状况来对此事件做出默认反映。IETF 的 CPL 语言规范中指定的服务器默认行为如表 8-1 所示。

表 8-1 CPL 服务器默认行为

非正常情况	服务器操作
脚本中无对地址集的修改操作或无信令操作 地址集为空	按服务器自身定义的机理确定被叫用户位置，然后按自身定义的机理对此次呼叫做出呼叫代理、前转或拒绝此次呼叫的处理
无改变地址操作或信令操作 地址集非空	只可能在主叫方为脚本拥有者的情况下出现（outgoing 类），服务器将地址集中的地址作为目的地接续呼叫
已执行改变地址操作 无信令操作	将地址集中的地址作为目的地接续此次呼叫，或将此地址作为重定向地址重定向此次呼叫，具体采用那种方式由服务器定义的策略决定。如果地址集为空，拒绝此次呼叫
proxy 节点未规定无应答输出 未规定超时时间	呼叫持续振铃，直到超过服务器允许的时间或超过最长响应时间

### 8.2.3 CPL 的应用

CPL 可以用来描述一些简单而个性化强的业务，非常适合终端用户定制一些个人化的业务，也可用于若干地址的共同呼叫策略管理，例如，对一个网内的用户进行呼叫屏蔽或呼出筛选。用 CPL 语言描述的具体业务如下：

#### ◆ 呼叫遇忙/无应答前转

当某个目的地址为忙或无应答时，应用 CPL 脚本可将呼叫转移到用户预定义的某个地址上。

#### ◆ 信息地址

公司可以给用户提供一个通用的信息地址，当用户呼叫此地址时，如果是在工作时间，用户可以得到一个可以接听此电话的人的名单，如果不在工作时间，可以向用户终端发送 Web 页面，指明在什么时间可以呼叫这个地址。

#### ◆ 智能化用户定位

当呼叫到达时，按照呼叫的类型和用户注册信息，选择恰当的注册位置接续此呼叫。当呼叫目的地用户注册了多个符合条件的终端时，可接通多个终端，每个终端的摘机信息分别报告给主叫方。

#### ◆ 与媒体信息结合的智能化用户定位

当呼叫到达时，按媒体能力选择用户注册的终端中最符合此呼叫请求的终端，如果该终端在一定时间段内无应答，则按媒体能力匹配程度，依次选择下一个终端进行接续。

#### ◆ 客户账单分配

此业务适合用做律师事物所等一些有大量固定客户，并需对客户电话咨询收取费用的办公室。当客户呼叫一个办公室电话时，可以按呼叫地址确定客户是谁并将此客户的姓名、地址、呼叫时间记录下来。如果无与此地址对应的客户，该呼叫被转移到秘书那里，这样有利于处理客户的账单。

在下一代网络中应用 CPL 开发新业务具有以下优点：

- 简单、效率高、容易实现；

- 容易验证正确性;
- 运行安全;
- 容易编写和解析;
- 可扩展, 服务器可以在不影响正在执行的脚本的情况下增加对新的语言特征的支持, 服务器也可以很容易地发现未实现的语言特征并安全地通知用户。

但 CPL 语言也存在一些不足之处:

- 只允许创建与呼叫控制有关的业务, 很难开发使用多种网络能力的业务;
- 不提供用户交互特征。

CPL 的一些缺陷可以通过对 CPL 语言的扩展加以解决, 如增加标记以支持提供用户交互。另一些不足则很难通过对 CPL 本身的改进来弥补, 但一种技术不可能解决所有的问题, 可以采用其他技术手段实现 CPL 不能支持的功能。

## 8.3 JAIN SIP Servlet

### 8.3.1 Servlet 基本知识

Servlet 是 Java 2.0 中新增的一个全新功能。Java Servlet 是运行在请求/面向请求服务器上的模块, 如一个 Java-enabled Web 服务器和类似这样的延伸。例如, 一个 Servlet 可以从一个 HTML 订单表中获取数据然后用一些商业上的算法来更新公司相应的订单数据库。

也就是说, Servlet 能够像 CGI 脚本一样扩展 Web 服务器功能, 但是 Servlet 占用很少的密集资源。有很多用 CGI 脚本编制的一些站点由于访问量剧增, 性能迅速下降, 这是 CGI 脚本一个缺点。同时由于 Servlet 是用 Java 编写的, 因此是跨平台的。实际 Servlet 是电子商务真正的开始。

Servlet API 用来编写 Servlet, 编写 Servlet 可以不关心一个 Servlet 是怎样被装载, Servlet 运行的服务器环境是什么, 或者用来传输数据的协议是什么等, 这样 Servlet 就可以融合在不同的 Web 服务器中。Servlet 可以有效地替代 CGI 脚本, 它可以方便地产生容易编写而且运行快的动态文本, 可以很方便地调试并寻找出程序问题所在。

Servlet 执行 Javax.Servlet.Servlet 接口, Servlet 编写者可以通过直接实现此接口开发 Servlet, 但这样通常没有必要, 因为大多数 Servlet 是针对用 HTTP 协议的 Web 服务器, 这样最常用开发 Servlet 的办法是用 Javax.Servlet.http。HttpServlet 内的 HttpServlet 类通过扩展 GenericServlet 基类执行 Servlet 接口, 提供了处理 HTTP 协议的功能。一般地, 用 HttpServlet 指定的类编写的 Servlets 可以多线程地并发运行 service 方法。

#### 1. 一个 HTTP Servlet 处理 GET 和 HEAD 方法的例子

```
public class SimpleServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        // 首先设置头部
        res.setContentType("text/html");
        // 用 writer 方法返回响应数据
```

```

    PrintWriter out = res.getWriter();
    out.println("<HEAD><TITLE> SimpleServlet Output</TITLE></HEAD><BODY>");
    out.println("<h1> SimpleServlet Output </h1>");
    out.println("<P>This is output is from SimpleServlet.");
    out.println("</BODY>");
    out.close();
    }

    public String getServletInfo() {
    return "A simple Servlet";
    }
}

```

这个例子完整地实现了一个 Servlet。

## 2. 一个 HTTP Servlet 处理 POST 方式的例子

这里是个用 HTML 带 POST 表单的例子：

```

<HTML>
<HEAD><TITLE>JdcSurvey</TITLE></HEAD>
<BODY>
<FORM action=http: //demo: 8080/Servlet/survey method=POST>
<INPUT type=hidden name=survey value=Survey01Results>
<BR><BR>How Many Employees in your Company?<BR>
<BR>1-100<INPUT type=radio name=employee value=1-100>
<BR>100-200<INPUT type=radio name=employee value=100-200>
<BR>200-300<INPUT type=radio name=employee value=200-300>
<BR>300-400<INPUT type=radio name=employee value=300-400>
<BR>500-more<INPUT type=radio name=employee value=500-more>
<BR><BR>General Comments?<BR>
<BR><INPUT type=text name=comment>
<BR><BR>What IDEs do you use?<BR>
<BR>JavaWorkShop<INPUT type=checkbox name=ide value=JavaWorkShop>
<BR>J++<INPUT type=checkbox name=ide value=J++>
<BR>Cafe'<INPUT type=checkbox name=ide value=Cafe'>
<BR><BR><INPUT type=submit><INPUT type=reset>
</FORM>
</BODY>
</HTML>

```

这里的 Servlet 将表单数据写入一个文件，并且用一个 thank you 信息响应用户。具体的 Servlet 方法，如下例：

```

public void doPost(HttpServletRequest req, HttpServletResponse res)

```

```
throws ServletException, IOException
{
// 首先设置响应的 content type 头部
res.setContentType("text/html");
// 得到响应的 PrintWriter 以返回文本给客户端
PrintWriter toClient = res.getWriter();
try {
// 打开一个文件写入结果
String surveyName = req.getParameterValues("survey")[0];
FileWriter resultsFile = new FileWriter(resultsDir + System.getProperty("file.separator")
+ surveyName + ".txt", true);
PrintWriter toFile = new PrintWriter(resultsFile);
// 从客户端得到表单数据并存储在这个文件中
toFile.println("");
Enumeration values = req.getParameterNames();
while(values.hasMoreElements()) {
String name = (String)values.nextElement();
String value = req.getParameterValues(name)[0];
if(name.compareTo("submit") != 0) {
toFile.println(name + ": " + value);
}
}
toFile.println("");
// 关闭文件
resultsFile.close();
// 把 “thank you for participating”返回给客户端
toClient.println("");
toClient.println("");
toClient.println("Thank you for participating");
toClient.println("");
} catch(IOException e) {
e.printStackTrace();
toClient.println( "A problem occurred while recording your answers. "
+ "Please try again.");
}
// 关闭 writer; 响应完成
toClient.close();
}
```

这个 doPost 方法是用 getParameterNames 和 getParameterValues 方法来从表单中获取数据。因为它返回文本给客户端，doPost 将调用 getWriter 方法，在写入响应主体部分之前，它对响

应头部字段进行了设置，响应完成后，关闭 writer。

### 3. Servletrunner

如果要在 Web 服务器上运行 Servlet，请参看相应服务器的说明书。这里只介绍 Servletrunner，它是多线程的，可以运行多个 Servlet，但它在服务器启动时不会自动启动。Servletrunner 在/bin 目录中。用-help 调用它会有下列信息出现：

```
% ./bin/Servletrunner -help
Usage: Servletrunner [options]
Options:
-p port      the port number to listen on
-b backlog   the listen backlog
-m max       maximum number of connection handlers
-t timeout   connection timeout in milliseconds
-d dir       Servlet directory
-r root      document root directory
-s filename  Servlet property file name
-v          verbose output
%
```

为了看见这些选项的默认值，可以用-v 开关调用它。

```
% ./bin/Servletrunner -v
Server settings:
port = 8080
backlog = 50
max handlers = 100
timeout = 5000
Servlet dir = .
document dir = .
Servlet propfile = .: Servlet.properties
```

一旦执行 Servletrunner，便能通过浏览器直接调用它们，如 `http://machine-name:port/Servlet/Servlet-name`，这里 Servlet-name 对应于一个 Servlet 的名字。

### 8.3.2 SIP Servlet

SIP Servlets API 使用与 HTTP Servlet 处理 HTTP 报文相同的方式，扩展了 HTTP Servlet 定义，以包含对 SIP 报文的处理。与 HTTP Servlet 相同，用于 SIP Servlet 的目标平台为代理服务器或应用服务器。SIP Servlet 可以利用 JAIN SIP API 的结构。该 API 的结构与 HTTP Servlet 的结构相似。

由于 SIP API 涵盖的内容远比用户代理或基于客户的软件所需要的内容丰富，所以就又开发出了 SIPLite API，作为 SIP API 和基本呼叫控制处理的一个功能子集。SIPLite API 面向无需全套 SIP API 类定义的用户代理或覆盖区较小的客户。

#### 1. 模型

SIP Servlet 模型如图 8-3 所示。

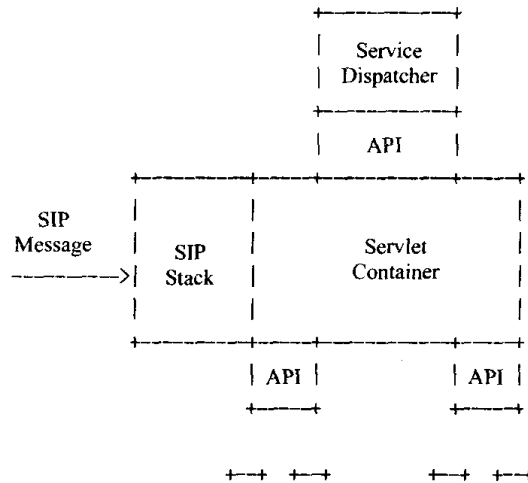


图 8-3 SIP Servlet 模型

## 2. 接口

### 1) ContactDatabase

```

public java.util.List getContacts(SipAddress sipAddr)
public void setContacts(SipAddress sipAddr, java.util.List contacts)
public void addContact(SipAddress sipAddr, Contact contact)
public void removeContact(SipAddress sipAddr, Contact contact)

```

### 2) Contact

Contact 地址是 SIP 联系地址。

```

public interface Contact extends SipAddress
public Contact getWildcard()
public float getQ()
public void setQ(float q)
public String getAction()
public void setAction(String action)
public String getExpires()
public void setExpires(String expires)
public interface ServletConfig
public String getInitParameter(String name)
public SipServletContext getServletContext()

```

### 3) ServletConfig

ServletConfig 地址是 Servlet 的配置信息。

### 4) SingleThreadModel

此接口申明如下：

```

public interface SingleThreadModel

```

### 5) SipAddress

此类方法如下：



```

public String getDisplayName()
public SipURL getSipURL()
public String getURI()
public String getParameter(String name)
public java.util.Iterator getParameterNames()

```

## 6) SipConstants

SipConstants 地址是 SIP 常量。

```

public static final String INVITE    = "INVITE";
public static final String ACK       = "ACK";
public static final String OPTIONS   = "OPTIONS";
public static final String BYE       = "BYE";
public static final String CANCEL    = "CANCEL";
public static final String REGISTER = "REGISTER";
public static final int SC_TRYING    = 100;
public static final int SC_RINGING   = 180;
public static final int SC_CALL_FORWARDING = 181;
public static final int SC_QUEUED    = 182;
public static final int SC_OK        = 200;
public static final int SC_MULTIPLE_CHOICES = 300;
public static final int SC_MOVED_PERMANENTLY = 301;
public static final int SC_MOVED_TEMPORARILY = 302;
public static final int SC_SEE_OTHER = 303;
public static final int SC_USE_PROXY = 305;
public static final int SC_ALTERNATIVE_SERVICE = 380;
public static final int SC_BAD_REQUEST = 400;
public static final int SC_UNAUTHORIZED = 401;
public static final int SC_PAYMENT_REQUIRED = 402;
public static final int SC_FORBIDDEN = 403;
public static final int SC_NOT_FOUND = 404;
public static final int SC_METHOD_NOT_ALLOWED = 405;
public static final int SC_PROXY_AUTHENTICATION_REQUIRED = 407;
public static final int SC_REQUEST_TIMEOUT = 408;
public static final int SC_CONFLICT = 409;
public static final int SC_GONE = 410;
public static final int SC_LENGTH_REQUIRED = 411;
public static final int SC_PRECONDITION_FAILED = 412;
public static final int SC_REQUEST_BODY_TOO_LARGE = 413;
public static final int SC_REQUEST_URI_TOO_LARGE = 414;
public static final int SC_UNSUPPORTED_MEDIA_TYPE = 415;
public static final int SC_BAD_EXTENSION = 420;

```

```
public static final int SC_TEMPORARILY_NOT_AVAILABLE = 480;
public static final int SC_INVALID_CALL_ID           = 481;
public static final int SC_LOOP_DETECTED             = 482;
public static final int SC_TOO_MANY_HOPS             = 483;
public static final int SC_INTERNAL_SERVER_ERROR     = 500;
public static final int SC_NOT_IMPLEMENTED           = 501;
public static final int SC_BAD_GATEWAY               = 502;
public static final int SC_SERVICE_UNAVAILABLE       = 503;
public static final int SC_GATEWAY_TIMEOUT           = 504;
public static final int SC_VERSION_NOT_SUPPORTED     = 505;
public static final int SC_BUSY                      = 600;
public static final int SC_DECLINE                   = 603;
public static final int SC_DOES_NOT_EXIST            = 604;
public static final int SC_NOT_ACCEPTABLE            = 606;
```

### 7) SipFactory

此类方法如下：

```
public SipTransaction createSipTransaction(SipAddress from, SipAddress to)
public SipURL createSipURL(String sipURL) throws ParseException
public SipURL createSipURL(String user, String host)
public SipURL createSipURL(String user, String host, int port)
public SipAddress createSipAddress(String sipAddress) throws ParseException
public SipAddress createSipAddress(SipURL sipURL)
public SipAddress createSipAddress(SipURL sipURL, String displayName)
public Contact createContact(String contact) throws ParseException
public Contact createContact(SipURL url, String displayName)
public Contact createContact(String uri, String displayName)
public Contact createContact(SipAddress addr)
```

### 8) SipMessage

此类方法如下：

```
public String getMethod()
public String getProtocol()
public String getRemoteAddr()
public String getRemoteHost()
public String getServerName()
public int getServerPort()
public String getServerProtocol()
public SipTransaction getTransaction()
public String getHeader(String name)
public java.util.Iterator getHeaderNames()
public java.util.Iterator getHeaders(String name)
```

```
public void setHeader(String name, String value)
public void addHeader(String name, String value)
public SipAddress getFrom()
public SipAddress getTo()
public java.util.List getContacts()
public void setContacts(java.util.List contacts)
public int getContentLength()
public String getContentType()
public void setContentType(String type)
public byte[] getContent()
public Object getParsedContent() throws ParseException
public Object getMediaDescription() throws ParseException
public void setContent(byte[] buf)
```

### 9) SipRequest

此类方法如下：

```
public interface SipRequest extends SipMessage
public SipURL getRequestURI()
public String getAuthType()
public String getRemoteUser()
public boolean recordRoute(boolean recordIt)
public SipResponse createResponse()
public Object send(SipURL url)
```

### 10) SipResponse

SipResponse 地址 SIP 响应。

此类方法如下：

```
public interface SipResponse extends SipMessage
public int getStatus()
public String getReasonPhrase()
public Object getRequestToken()
public void setStatus(int code, String reason)
public void setStatus(int code)
public void send()
```

### 11) SipServlet

此类方法如下：

```
public void init(ServletConfig config)
public boolean gotRequest(SipRequest req) throws ServletException
public boolean gotResponse(SipResponse res) throws ServletException
public void destroy()
public ServletConfig getServletConfig()
```

```
public String getServletInfo()
public void log(String msg)
public void log(String msg, Throwable t)
```

## 12) SipServletContext

此类方法如下：

```
public int getMajorVersion()
public int getMinorVersion()
public String getServerInfo()
public Object getAttribute(String name)
public java.util.Iterator getAttributeNames()
public void setAttribute(String name, Object value)
public void removeAttribute(String name)
public void log(String msg)
public void log(String msg, Throwable t)
```

## 13) SipTransaction

SipTransaction 地址是 SIP 事务。

此类方法如下：

```
public String getCallID()
public SipAddress getFrom()
public SipAddress getTo()
public String getCSeq()
public SipRequest createRequest(String method)
public Object getAttribute(String name)
public java.util.Iterator getAttributeNames()
public void setAttribute(String name, Object value)
public void removeAttribute(String name)
public void addListener(SipServlet servlet)
public void removeListener(SipServlet servlet)
```

## 14) SipURL

此类方法如下：

```
public String getUser()
public void setUser(String user)
public String getPassword()
public void setPassword(String password)
public String getHost()
public void setHost(String host)
public int getPort()
public void setPort(int port)
public String getParameter(String name)
```

```
public void setParameter(String name)
public void setParameter(String name, String value)
public void removeParameter(String name)
public java.util.Iterator getParameterNames()
public String toString()
```

### 3. 类

#### 1) SipUtils

此类方法如下：

```
public abstract class SipUtils
public static long getJavaTime(String time)
```

#### 2) SipServletAdapter

此类方法如下：

```
public void init(ServletConfig config)
public boolean gotRequest(SipRequest req) throws ServletException
public boolean doInvite(SipRequest req) throws ServletException
public boolean doAck(SipRequest req) throws ServletException
public boolean doOptions(SipRequest req) throws ServletException
public boolean doBye(SipRequest req) throws ServletException
public boolean doCancel(SipRequest req) throws ServletException
public boolean doRegister(SipRequest req) throws ServletException
public boolean gotResponse(SipResponse res) throws ServletException
public void destroy()
public ServletConfig getServletConfig()
public String getInitParameter(String name)
public java.util.Iterator getInitParameterNames()
public SipServletContext getServletContext()
public String getServletInfo()
```

### 4. 异常

#### 1) ParseException

此类方法如下：

```
public ParseException()
public ParseException(java.lang.String msg)
```

#### 2) ServletException

此类方法如下：

```
public ServletException ()
public ServletException (java.lang.String msg)
```

### 5. 用法

#### 1) 回调注册

所用到的方法如下：

```
public void addListener(SipServlet servlet)
public void removeListener(SipServlet servlet)
```

## 2) 消息处理

```
public Object SipRequest.send(SipURL nextHop)
public void SipResponse.send()
```

创建消息:

```
public boolean doInvite(SipRequest req) {
    SipResponse res = req.createResponse();
    res.setStatus(486);
    res.send();
    return true;
}
```

## 3) 访问消息内容

所用到的方法如下:

```
public byte[] getContent()
public Object getParsedContent() throws ParseException
public Object getMediaDescription() throws ParseException
public void setContent(byte[] buf)
```

# 8.4 J2EE 开发环境

由于 Parlay 大多面向计算机行业的第三方, 这些厂商对通信知识了解不多, 而且电子商务、网站大多运行于 J2EE 环境, 为此需要 Parlay 也运行于此环境, 如 IBM 的 Websphere 和 BEA 的 Weblogic 等。为了叙述好 Parlay 的开发, 本节将介绍 J2EE 的相关知识。

## 8.4.1 J2EE 引言

在软件产业中, 基于构件的技术是当前的热点, 在面向对象的技术发展的今天, 构件技术在软件系统的开发上, 解决了重复开发的问题, 提高了软件开发的效率。在软件构件的技术上, 众多厂商形成了几大阵营:

- 以微软为首的 DCOM/COM+阵营, 从 DDE, OLE 到 ActiveX 等, 提供了构件开发的基础, 如 VC, VB, DELPHI 等都支持 DCOM, 但是脱离不了桌面计算的底子。
- 以 Apple 和 IBM 为首的 OpenDoc 阵营, 现在没有得到更多的厂商支持, 加之 Apple 和 IBM 的硬件平台在 C/S 应用中没有成为主流, 所以发展并不乐观。
- OMG (对象管理组织), 它推出了跨平台、跨语言的 CORBA 规范, 并已逐渐成为业界的标准。CORBA 的核心是 ORB (对象请求代理), 它提供了分布对象的通信手段, 然而, 实践已经证明: 在大事务环境中, 光有 ORB 是不够的。它缺乏健壮的体系结构来支持关键任务, 而且 ORB 把事务、并发性、持久性等底层处理的任务交给了应用程序开发者, 加大了开发者的负担, 而这正是 TP monitor 所擅长的; 但 TP monitor 不是面向对象的, 它们的灵活性、扩展性和重用性较差。为了解决这些问题, Anne

Thomas 提出了 CTM (Component transaction monitor) 的概念 (也称 OTM, 具有组件执行环境), 它把 TP monitor 和 ORB 技术有机地结合起来。CTM 提供了自动处理事务、并发性、安全、持久性和资源管理, 提供了处理关键任务的机制。但问题又随之出现, 由于 CORBA CTM 自身的缺陷, 用户在某个厂商 CTM 中开发的组件很难在另外一个厂商的 CTM 中重用; CORBA 规范对底层服务也进行标准化, 厂商难以忍受重写像 TUXEDO, CICS 之类的产品来适应它。所以 CORBA CTM 的市场非常零星。

作为 OMG 成员之一的 Sun 挟 Java 利器, 推出了 EJB CTM, 它允许厂商灵活地实现事务、安全等底层服务; 它不但平台独立, 而且用户在 EJB 兼容的 CTM 中开发的组件, 可以迁移到另外一些 EJB 兼容的 CTM 中; 另外, 值得一提的是, 它给第三方厂商提供了良好的发展空间。它们可以开发组件库或者插件。ERP、事务对象框架 (business object framework) 也逐渐向 EJB 靠拢 (IBM San Francisco business object framework 已经移植到 EJB 中)。

由于 EJB 具有这些优点, 众多厂商 (如 BEA, IBM, Inprise, IONA, Oracle, Sun, Sybase) 在第一时间推出了对 EJB 支持的产品。EJB 的出现, 打破了 CORBA 一枝独秀的局面。EJB 汲取了 CORBA 的精华, 吸收了 CORBA 的经验和教训并加以改进, 后来居上, 并与 Servlet, JSP, JNDI, JDBC 等, 构成了分布计算新的风景线。

EJB 是一种基于构件的开发模型, 也是 Sun 公司 J2EE 规范的核心, 用于创建可伸缩、跨平台、分布式应用, 并且可创建具有动态扩展性的服务器应用。EJB 构件模型使得构件可包含应用系统逻辑, 并且构件可重用, 另外还提供资源管理、事务支持和并发性管理。EJB 将带给分布式、面向对象开发一种新的模式、新的尝试。

### 8.4.2 EJB 模型

EJB 是一种 C/S 结构, 提供服务的方法运行在服务器端, 根据客户端的请求调用相应的方法, 其基本结构如图 8-4 所示。从图中可以看出 EJB 的主要组成部分包括: EJB 构件、容器、服务器、EJB 对象 (Object) 和 EJB Home。

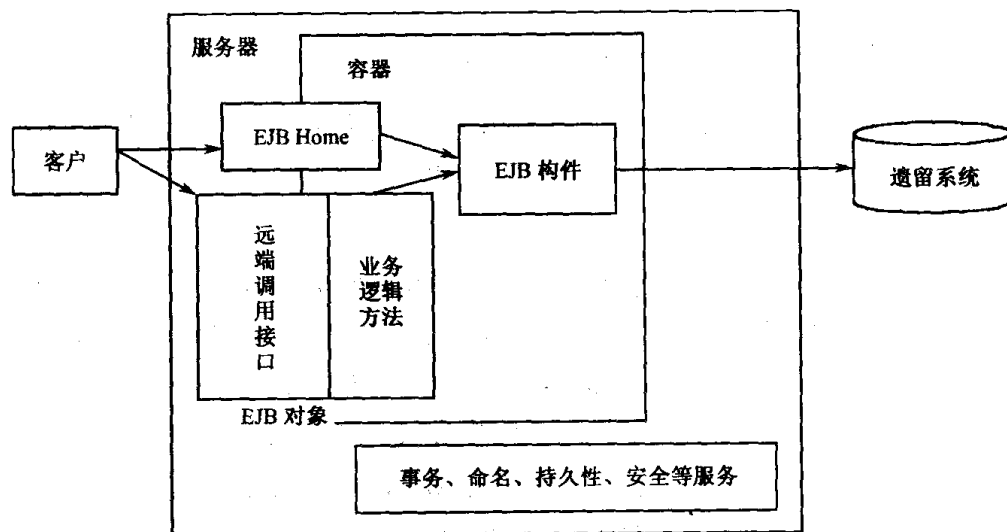


图 8-4 EJB 模型图

#### 1. 客户

有两种类型的客户机可以使用 EJB。

- EJB 客户机: 一个使用 EJB API 的 Java 客户机。客户机利用 JNDI 确定对象, 利用 IIOP 或 JRMP 协议上的 Java RMI 来调用远程方法 (RMI-JRMP 可以使得远端对象具有多态性, 而 RMI-IIOP 对安全及事务服务支持得更好)。如果使用 RMI-IIOP, CORBA IDL 的使用是隐含的, 也就是说, 开发人员只使用 Java 代码, 开发客户机程序时可以不了解 CORBA 及其 IDL 知识。
- 纯 CORBA 客户机: 用 CORBA IDL 支持的任何语言写的客户机。客户机用 COS 命名服务来确定对象, 用 CORBA IDL 来调用远程方法, 用对象事务服务 OTS (Object Transaction Service) 来执行事务, 其中程序设计人员要创建一个 IDL 文件, 即 CORBA IDL 的使用是显式的。

当然, 基于 DCOM 的客户端, 只要和 EJB 建立有效的映射, 也是可行的。目前的 J2EE Client Access Services (CAS) Table Bridge 可以让基于 DCOM 的程序访问 EJB 的实体 bean。

## 2. EJB 构件

EJB 中的 bean 可以分为会话 bean (维护会话)、实体 bean (处理事务) 和消息 bean (提供异步消息机制)。

### ◆ 会话 bean (维护会话)

会话 bean 是短暂的对象, 运行在服务器端, 并执行一些应用逻辑处理, 它由客户端应用程序建立, 其数据需要自己来管理。由于会话 bean 的粒度较大, 可以包含很多业务逻辑, 不需要反复从网络上装载, 所以可以提高网络性能 (此思想类似移动 agent 的思想)。

会话 bean 可进一步分为无状态 (不维护客户端状态) 和有状态 (维护客户端状态) 两种。会话 bean 的配置描述符 (Deployment Descriptor) 必须声明该 bean 是有状态或无状态的。一个无状态 bean 是在方法调用间不维护任何状态信息的 bean。通常, 会话 bean 的优点是代替客户端维护状态。然而, 让会话 bean 无状态也有一个好处。无状态 bean 不维护状态, 所以没有需要保存的信息, 容器可以删除 bean 的实例, 节省了主存。有状态和无状态的会话 bean 都可以访问数据库, 并且参与一个事务。为了让 bean 在事务中执行它的任务, bean 开发者可以在 bean 中实现 `javax.ejb.SessionSynchronization` 接口, 容器能自动检测这个接口, 容器会使用这个接口中的方法以使 bean 得到事务的状态信息。

容器开发商可以实现把会话 bean 的实例从主存移到二级存储中的交换机制 (钝化), 这可以增加一段时间内实例化的会话 bean 的总数。容器维护一个 bean 的时间期限, 当某个 bean 的不活动状态时间达到这个阈值, 容器就把这个 bean 拷贝到二级存储中并从主存中删除; 而当它再次活动时, 就从二级存储中恢复到主存 (活化)。

### ◆ 实体 bean (处理事务)

实体 bean 是持久的对象, 可以被其他对象调用。与会话 bean 不同, 实体 bean 不维护客户端状态, 且必须在建立时指定一个惟一的标识, 并提供相应的机制允许客户应用程序根据实体 bean 标识来定位 beans 实例。多个用户可以并发访问实体 bean, 事务间的协调由容器来完成。实体 bean 支持事务处理, 当系统停机时, 也可恢复。

EJB 规范中定义了两种处理实体 bean 的持久性模型, 即自管理的持久性 (Beans Managed Persistence) 及容器管理的持久性 (Container Managed Persistence)。自管理持久性是由 EJB 自己来管理持久性, 它需要 EJB 开发者来编写数据库或应用程序的处理逻辑, 并加入到类的 `ejbcreate()`、`ejbremove()`、`ejbfind()`、`ejbload()` 和 `ejbstore()` 等方法中。容器管理的持久性是将 EJB



持久性管理交给容器来完成, 开发者一般要在配置描述符中的 `ContainerManagedFields` 属性中指定 EJB 实例持久性域。使用容器管理持久性, 用户无须知道实体存储的数据源, 也不需要参与复杂、烦琐的编码工作。与会话 bean 一样, 实体 bean 也支持活化和钝化的过程。

通常会话 bean 和实体 bean 是一起使用的。会话 bean 允许保存客户端的状态信息, 客户端和会话 bean 实例间是一一对应的。实体 bean 允许保存记录的信息, 实体 bean 实例和记录间则是一一对应的。一个理想的情况是客户端通过会话 bean 连接服务器, 然后会话 bean 通过实体 bean 访问数据库, 这使得既可以保存客户端的信息又可以保存数据库记录的信息。

#### ◆ 消息 bean (提供异步消息机制)

消息 bean 是 EJB2.0 对 EJB1.1 规范的一个基础性更改, 专门设计用来处理 JMS (Java Message System) 消息。JMS 是一种与厂商无关的 API, 用来访问消息收发系统, 并提供了与厂商无关的访问方法, 以访问消息收发服务。许多厂商目前都支持 JMS, 包括 IBM 的 MQSeries、BEA 的 WebLogic JMS service 和 Progress 的 SonicMQ 等。JMS 中消息收发是异步的, 也就是说, JMS 客户机可以发送消息而不必等待回应, 这完全不同于基于 RPC 的系统, 如 CORBA2.3 及以前版本和 JavaRMI。

通常 EJB 系统中有两种方式使用 JMS, EJB1.1 及其以前的版本把 JMS 作为一种 bean 可用的资源, 使用 JMS API 的 bean 就是消息的产生者或发送者。在这种情况下, bean 将消息发送给队列的虚拟通道; 而在 EJB2.0 中则存在消息 bean, 它是消息的使用者或接收者, 它监听特定的虚拟通道, 并处理发送给该通道的消息。

和会话 bean 以及实体 bean 一样, 消息 bean 也是一种完备的企业级 bean, 但其间仍存在一些重要的区别。消息 bean 没有远程接口或本地接口。这是因为消息 bean 不是 RPC 构件, 它没有供 EJB 客户机调用的业务方法。同时消息 bean 类似于无状态的会话 bean, 即这两种 bean 在两次请求之间都不保持任何状态, 因此消息 bean 是无状态的。

### 3. EJB 容器

EJB 容器是 EJB 构件运行的环境, 是一层代替 bean 执行相应服务的接口。EJB 容器负责提供协调管理、资源管理、版本控制、动态性、一致性、安全、事务处理和 RMI 等功能。另外容器建立上下文环境, 负责切换、协调不同 EJB 对象。会话 bean 中的与二级存储中交换以及实体 bean 中的持久性管理都是由容器完成的。

客户应用程序通常不和 EJB 直接打交道, 而是要通过容器提供的接口, 该接口提供了 EJB 的客户视角。

一类接口称为 Home 接口, 由 EJB Home 提供。Home 接口不是由 bean 来实现, 而是通过称为 Home 对象的类来实现。一个 Home 对象的实例在服务器中实例化, 使得客户端可以访问它们。Home 接口允许客户建立或删除 EJB。对实体 EJB 来说, Home 接口还提供定位特定 EJB 实例的功能。当用户请求 EJB 服务时, 它首先要通过 JNDI 来定位对象的 Home 接口, EJB 类及其容器对用户来说是透明的。Home 接口必须提供建立 EJB 对象的方法, 一旦用户找到所需的 EJB 类后, 它就可以通过调用 Home 接口中的生成方法建立 EJB 对象。

另一类接口称为远端调用接口 (remote interface)。EJB 对象是容器提供的 EJB 类的一个实例, 它用来实现 EJB 的远端调用接口。客户总是通过调用 EJB 远端调用方法来执行 EJB 服务。用户调用 EJB 对象时, 由 EJB 容器接受请求, 并将任务交给 EJB 对象。这种机制保证为用户及 EJB 提供透明的状态管理、事务控制及安全性服务。

#### 4. 遗留系统

EJB 的发展方向是要支持多种环境, 包括:

- TP Monitor, 如 IBM TXSeries(CICS&Encina), BEA Tuxedo (通过 jolt 连接)。
- Componet Transaction Server, 如 Sybase Jaguar CTS, Microsoft Transaction Server。
- CORBA 平台, 如 Inprise VisiBroker, Iona Orbix。
- 数据库管理系统 (通过 jdbc), 如 Informix, Oracle, Sybase, CloudScape。
- Web 服务器: 如 Java Web Server, Apache, Oracle Application Server。

### 8.4.3 EJB 角色

如图 8-5 所示, EJB 体系结构通过把编程的过程分为七个不同的角色而使开发复杂的应用系统变得简单。七个不同的角色, 每个都有特定的任务。

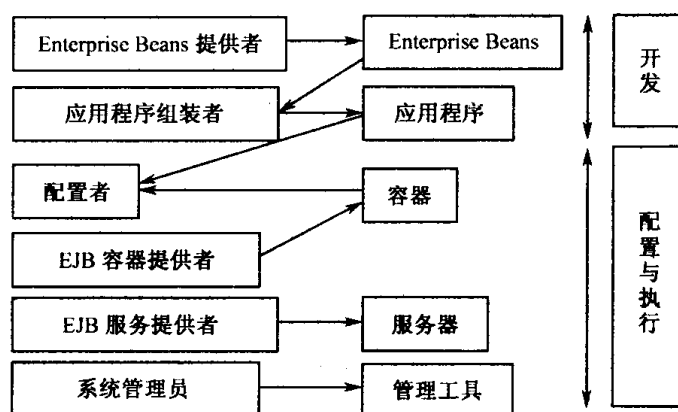


图 8-5 角色及生命周期图

- EJB 服务提供者 (EJB Server Provider): 是典型的提供分布式底层服务的代理。它提供了一个分布式应用程序开发者需要的平台和设施, 也提供了分布式程序的运行环境。
- EJB 容器提供者 (EJB Container Provider): 是分布式系统, 是事务处理、安全方面的专家容器提供者, 提供了一个 EJB 的配置工具 (如 GenIC) 和这些配置实例的运行时支持。
- 企业 beans 提供者 (Enterprise beans Provider): 是金融或电信等应用领域的专家。它提供了业务方法, 不仅定义了 beans 的远端调用接口和 Home 接口, 还定义了 beans 的配置描述符。因为容器管理着系统级的任务, 所以它不需要关心分布处理、事务处理、安全性能等方面。
- 应用程序组装者 (Application Assembler): 是一个将定制的企业 beans 和其他的构件 (如 GUI 客户、Applet、Servlet 等) 组装成一个完整的应用程序的行家。他接受上述企业 Beans 提供者输出的 EJB-jar 文件作为自己的输入, 在配置描述符中插入应用程序组装指令。组装器只关心 beans 的接口, 包括远端调用接口和 Home 接口, 而不关心这些接口是怎么实现的。
- 配置者 (Deployer): 为特定的企业 beans 配置特殊的操作环境。配置者为了适应应用程序, 将多个企业 beans 组合起来, 通过修改企业 beans 的属性来达到配置相应的操

作环境。例如，配置者通过设置配置描述符的相应的属性来设置事务和安全。配置者的另一个任务是将应用程序与现有的企业管理软件结合起来。常见的配置工具有 GenDD、DumpDD 等。

- 系统管理员 (System Administrator): 是与配置应用有关的。管理员配置、管理企业计算和网络服务的底层，包括 EJB 服务和容器。管理员监视着应用服务器的运行状况，当应用服务器不正常运行时就采取相应的措施。
- 持久性管理器提供者 (Persistence Manager Provider): 是 EJB2.0 版本中才出现的角色，他负责容器管理的持久性模型中、实体 beans 持久性在容器中的安装。

传统的应用程序开发者现在是企业 beans 的提供者，也可能是应用程序的装配者。这样能使他们将主要精力放在业务逻辑上面。虽然分布式的应用程序还是比较复杂，可是应用程序设计者的工作却变得简单了，因为所有复杂的工作都交给了 EJB 服务器和容器提供者了。

#### 8.4.4 EJB 和其他技术的关系

EJB 技术不是凭空产生的，它是在 Sun JavaBeans 技术的基础上，汲取 CORBA 的精华，逐渐得到发展的，但是它们也有一定的区别。

##### 1. EJB 和 JavaBeans 的关系

很多人往往把 JavaBeans 和 EJB 混淆，JavaBeans 提供了基于构件的开发机制，一般 JavaBeans 是可视化的构件，也有一些 JavaBeans 是非可视化的，JavaBeans 可以在多个应用系统中重用。一个标准的 JavaBean 是一个客户端构件，在运行时不能被其他客户机程序存取或操作。但客户端的 JavaBeans 容器可以根据 JavaBeans 的属性、方法、事件的定义在设计时或运行时对 JavaBeans 进行操作，JavaBeans 不一定要用于 C/S 系统。EJB 没有用户界面，并完全位于服务器端，EJB 可以由多个 JavaBean 组成，规范详细说明了 EJB 容器需要满足的需求以及如何与 EJB 构件相互协作。EJB 可以和远程的客户端程序通信，并提供一定的功能。根据规范说明，EJB 是 C/S 系统的一部分，如果不和客户端程序交互，EJB 一般不执行具体的功能，EJB 和 JavaBeans 的一个重要区别是 EJB 提供了网络功能。

##### 2. EJB 和 CORBA 的关系

由于 CORBA 规范的广泛流行，以后将会有很多基于 CORBA/IIOP 的 EJB 产品出现，为了保证多个开发商之间的基于 CORBA 的 EJB 产品之间的互操作性，规范定义了 EJB 到 CORBA 的映射，它分为四个部分：

- 分布映射：定义了 EJB 和 CORBA 对象之间的关系，以及 EJB 规范中定义的 Java RMI 远程接口到 OMG IDL 的映射。
- 命名映射：说明了如何利用 COS 命名服务来确定 EJB Home 对象。
- 事务映射：定义了 EJB 的事务支持 (TX-NOT-SUPPORTED, TX-REQUIRES, TX-SUPPORT, TX-REQUIRES-NEW, TX-MANDATORY, TX-BEAN-MANAGED) 到 OMG OTS 的映射。
- 安全性映射：定义了 EJB 中的安全性特征到 CORBA 安全性的映射。

#### 8.4.5 常见 EJB 系统

EJB 规范是 Sun 公司制定的，EJB 1.1 推出后立即引起了计算机界的广泛关注和产业界的

支持,并推出成熟的产品,如 IBM 的 WebSphere、BEA 公司的 WebLogic、Inprise 公司的 IAS(Inprise Application Server)。在国内,东方通科技公司推出了 TongWeb,金蝶软件公司旗下的 APUSIC 软件公司推出了 Apusic Application Server1.0。另外,还有一些比较小的、可以用来做学习和研究的、提供源代码的产品,如 OpenEjb, EJBoss, JonAS, Enhydra。

### 1. WebSphere

IBM WebSphere 是一个完善的、开放的 Web 应用服务器,它是 IBM 电子商务应用架构的核心,在高级版中开始支持 EJB 的编程模型。EJB 是 WebSphere 最核心的对象技术之一,它提供了具有交易功能的服务器端的 Java 构件,同时又是一种新型的对象分布技术编程模型;使用 EJB 构件完成的应用会支持更高级的基于数据库的交易处理功能,如多个数据库之间的更新、两阶段提交等。最关键的是,WebSphere 把 EJB 和 IBM 其他优秀的产品(如 TXSeries)结合在了一起。在具体的应用中,基于安全的考虑,性能考虑和系统管理考虑,还可以增加 Firewall, WebSphere Performance Pack 和 Tivoli。所以,IBM 的 EJB 容器不一定是最好的,但是 WebSphere 中的 EJB 和 IBM 的其他产品的结合却是做得最好的。通过 Visual Age,可以很快地开发出 EJB 构件,然后部署在 WebSphere 应用服务器中,快速地开发一个电子商务应用。

### 2. WebLogic

作为服务器端商务逻辑的行业标准和 Java2 平台的基石,EJB 是 BEA WebLogic 应用服务器的主要技术。BEA 是通过为 BEA WebLogic Server 5.1 提供一个附加的软件包,实现了对 EJB2.0 的支持。凭借 BEA WebLogic Server 对 EJB2.0 的支持,BEA 可帮助开发人员更快地推出电子商务应用系统,从而让开发人员极大地受益于这一业内的最新标准。

### 3. IAS

Inprise 的 IAS 也提供了 EJB 的运行环境,不过没有和别的产品结合在一起,技术性比较浓一些。VisiBroker 是 IAS 的基础,所以在启动 IAS 之前,网络中至少要有一台机器运行了 Smart Agent,默认的端口号是:14000。IAS 是系列服务和工具的集合,通过这些服务和工具,可以编译、配置、管理基于 Web 的多层体系结构的应用程序。IAS 中的 Visibroker 有效地实现了连接和线程的管理,如果有多个客户同时请求同一个服务器。IAS 通过连接池和线程池来有效地管理这些请求。在底层,IAS 是通过 RMI/IIOP 来处理多台机器之间的通信。IAS 提供了一个完整的 EJB 容器服务,提供了一个标准的 EJB 容器工具包,这个工具包既可以独立运行,也可以结合到自己的应用中去。IAS 的 EJB 容器服务可以自己创建、管理一个 EJB 容器,可以用来配置、运行 EJB,还可以监视 EJB 的运行状态。另外,IAS 也提供了自己的 JNDI 名字空间,通过 JNDI 管理 bean 的注册,资源的获取。可以说 IAS 是一个纯粹的 EJB 运行环境。

值得一提的是,在现有的 EJB 开发工具中,Jbuilder4 把 WebLogic 和 IAS 集成在它的 IDE 中,需要另外安装 WebLogic 或者 IAS。而 IBM 的 Visual Age 当然就把自己的 WebSphere 优先考虑了。

## 8.4.6 开发实例

实体 bean 在 EJB 1.0 规范中是可选的,而会话 bean 则是必须包含的,本例以有状态的会话 bean 来开发简单的网上购物车 cart(会话 bean 实例一般不与其他客户端共享,这允许会话 bean 维护客户端的状态,正好满足购货车的要求。因为众多顾客可以同时购货,向他们自己的购货车中加东西,而不是向一个公共的购货车中加入私人的货物)。开发工具涉及

Jbuilder4.0、IAS。

### 1. 创建 Remote Interface

创建 Remote Interface 的语句如下：

```
public interface Cart extends javax.ejb.EJBObject {
    void addItem(Item item) throws java.rmi.RemoteException; //以下为业务逻辑方法的定义
    void removeItem(Item item) throws ItemNotFoundException, java.rmi.RemoteException;
    float getTotalPrice() throws java.rmi.RemoteException;
    java.util.Enumeration getContents() throws java.rmi.RemoteException;
    void purchase() throws PurchaseProblemException, java.rmi.RemoteException;
}
```

### 2. 在定义好接口后，实现 Bean 类

在定义好接口后，实现 Bean 类的语句如下：

```
public class CartBean implements javax.ejb.SessionBean { //会话 bean 是 javax.ejb.SessionBean 接口
    public void setSessionContext(javax.ejb.SessionContext sessionContext) {...}
    public void ejbCreate(String cardHolderName, String creditCardNumber, Date expirationDate) {...}
    public void ejbRemove() {...} //以下为业务逻辑的具体实现
    public void addItem(Item item) {...}
    public void removeItem(Item item) throws ItemNotFoundException {...}
    public float getTotalPrice() {...}
    public java.util.Enumeration getContents() {...}
    public void purchase(){...}
    public void ejbActivate() {...} //把 bean 恢复到主存中去
    public void ejbPassivate() {...} //把 bean 从主存转移到二级存储，这种交换机制可以增加一段时间内实例化的会话 bean 的总数
}
```

### 3. Home Interface

Home Interface 语句如下：

```
public interface CartHome extends javax.ejb.EJBHome {
    Cart create(String cardHolderName, String creditCardNumber, java.util.Date expirationDate) //生成一个新的 bean 对象，此方法实际由 EJBObject 实现，调用 bean 类的 ejbCreate 方法
    throws java.rmi.RemoteException, javax.ejb.CreateException;
}
```

### 4. Client 的实现

```
public class CartClient {...}
    public static void main(String[] args) throws Exception {
        javax.naming.Context context = new javax.naming.InitialContext(); //得到 JNDI 结构的根目录
        Object ref = context.lookup("cart"); //在目录树中查找 cart
        CartHome home = (CartHome) javax.rmi.PortableRemoteObject.narrow(ref, CartHome.class);
        Cart cart;
```

```
{ String cardHolderName = "Jack B. Quick";  
    String creditCardNumber = "1234-5678-9012-3456";  
    Date expirationDate = new GregorianCalendar(2001, Calendar.JULY, 1).getTime();//具体的信息  
    cart = home.create(cardHolderName, creditCardNumber, expirationDate); }//创建一个新的 bean, 即 cart,  
此变量是远端调用接口的引用, 通过 cart, 可调用 EJB bean 的方法, 执行购买等操作  
}
```

经过编译, 会将所有的 bean 打包放在一个 jar 文件中, 与普通的 jar 文件相比, 此文件中有一个 manifest 文件的入口以声明一个配置描述符类的实例 (EJB1.1 以后用 XML 描述), 用于配置所需的分布式特性, 如安全、事务等。然后根据服务器端提供的工具来配置 bean 到服务器端, 此时将使用配置描述符实例中的属性, 一旦配置完毕, 用户就可调用 bean 中的方法, 以完成所需的服务。

### 8.4.7 J2EE 小结

采用 EJB 可以提高开发人员的效率, 构件开发人员和应用开发人员不需要实现系统中的一些复杂的逻辑结构, EJB 模型保证开发出来的构件可以被部署在任何支持 EJB 规范的系统中, 即使该系统是由不同的开发商提供的。

- 从企业应用多层结构的角度来看, EJB 是商业逻辑层的构件技术, 与 JavaBeans 不同, 它提供了事务处理的能力;
- 从分布式计算的角度来看, EJB 像 CORBA 一样, 提供了分布式技术的基础, 提供了对象之间的通信手段;
- 从 Internet 技术应用的角度来看, EJB 和 Servlet, JSP 一起成为新一代应用服务器的技术标准。Servlet 负责与客户端通信, 访问 EJB, 并把结果通过 JSP 产生页面传回客户端, 成为开发的新潮流;
- 从发展的角度看, EJB 还会进一步地完善与发展。EJB 2.0 中, 新的容器管理的持久性模型比以前的模型要灵活得多, 它提供持久性管理器, 允许各种实体为复杂的对象图建立模型; 为了提供跨容器的可移植性, 推出了 EJB Query Language 作为查找和选择操作的一种通用的查询语言; 还在 Home 接口中增加了一些方法; 改进了安全机制。另外 CCM (CORBA Container Model) 规范可以使得 CORBA 构件和 EJB 有效地集成起来。CCM 采用支持 EJB 规范中的对象封装方案, 使 EJB 可作为 CORBA 构件重新使用, EJB 提供业务逻辑的构造能力, CORBA 则提供跨平台的分布通信能力。当然, CORBA 构件也可以在 EJB 中执行。

随着 EJB 中的进程管理、线程池、并发控制和资源管理的增强, EJB 的威力正越来越大地展现出来, 从而构成企业计算的基础。总而言之, 在日新月异的技术发展和更新中, EJB 及其后继者, 将在 Java 技术的大旗下不断攻城克地, 占领企业计算的大好江山。

## 8.5 Web 服务

### 8.5.1 Web 服务简介

Web Service 是在 Internet 上进行分布式计算的基本构造块。开放的标准以及对用户和应用

程序之间的通信和协作的关注产生了这样一种环境,在这种环境下,Web 服务成为应用程序集成的平台。应用程序是通过使用多个不同来源的 Web 服务构造而成的,这些服务相互协同工作,而不管它们位于何处或者如何实现。

有多少个构建 Web 服务的公司,就可能有多少种 Web 服务定义。不过几乎所有定义都具有以下共同点:

- Web 服务通过标准的 Web 协议向 Web 用户提供有用的功能。多数情况下使用 SOAP 协议。
- Web 服务可以非常详细地说明其接口,这使用户能够创建客户端应用程序与它们进行通信。这种说明通常包含在称为 Web 服务说明语言 (WSDL) 文档的 XML 文档中。
- Web 服务已经过注册,以便潜在用户能够轻易地找到这些服务,这是通过通用发现、说明和集成 (UDDI) 来完成的。

Web 服务体系结构的主要优点之一是:允许在不同平台上、以不同语言编写的各种程序以基于标准的方式相互通信。SOAP 比以前的方法要简单得多,因此要实现与标准兼容的 SOAP,障碍也要少得多。引入 Web 服务与引入以前的技术相比,其成本要低得多。

### 8.5.2 SOAP

SOAP 是 Web 服务的通信协议。SOAP 用来定义消息的 XML 格式,包含在一对 SOAP 元素中的、结构正确的 XML 段就是 SOAP 消息。

SOAP 规范的其他部分介绍如何将程序数据表示为 XML,以及如何使用 SOAP 进行远程过程调用 (RPC)。这些可选的规范部分用于实现 RPC 形式的应用程序,其中客户端将发出一条 SOAP 消息 (包含可调用函数,以及要传送到该函数的参数),然后服务器将返回包含函数执行结果的消息。目前,多数 SOAP 实现方案都支持 RPC 应用程序,这是因为习惯于开发 COM 或 CORBA 应用程序的编程人员熟悉 RPC 形式。SOAP 还支持文档形式的应用程序,在这类应用程序中,SOAP 消息只是 XML 文档的一个包装。文档形式的 SOAP 应用程序非常灵活,许多新的 Web 服务都利用这一特点来构建使用 RPC 难以实现的服务。

SOAP 规范的最后一个可选部分定义了包含 SOAP 消息的 HTTP 消息的样式。此 HTTP 绑定非常重要,HTTP 绑定虽然是可选的,但几乎所有 SOAP 实现方案都支持 HTTP 绑定。由于这一原因,人们通常误认为 SOAP 必须使用 HTTP。其实,有些实现方案也支持 MSMQ、MQ 系列、SMTP 或 TCP/IP 传输。

开始使用 SOAP 时,最容易混淆的是 SOAP 规范及其许多实现方案之间的差异。多数使用 SOAP 的用户并不直接编写 SOAP 消息,而是使用 SOAP 工具包来创建和分析 SOAP 消息。这些工具包通常将函数调用从某种语言转换为 SOAP 消息。例如,Microsoft SOAP Toolkit 2.0 将 COM 函数调用转换为 SOAP,而 Apache Toolkit 将 Java 函数调用转换为 SOAP。函数调用的类型和支持的参数数据类型随每个 SOAP 实现方案的不同而不同,因此适用于一个工具包的函数可能并不适用于另一个工具包。

HTTP 的普及和 SOAP 的简单性使用户几乎可以从任何环境中调用它们,因此成为 Web 服务的理想基础。

### 8.5.3 WSDL

WSDL (Web Services Description Language) 为 Web 服务说明语言,可以简单地认为 WSDL

文件是一个 XML 文档，用于说明一组 SOAP 消息以及如何交换这些消息。WSDL 对于 SOAP 的作用就像 IDL 对于 CORBA 或 COM 的作用。由于 WSDL 是 XML 文档，因此很容易进行阅读和编辑。

WSDL 文件用于说明消息格式的表示法以 XML 架构标准为基础，这意味着它与编程语言无关，因此适用于说明可从不同平台、以不同编程语言访问的 Web 服务接口。除说明消息内容外，WSDL 还定义了服务的位置，以及使用什么通信协议与服务进行通信。有几种工具可以读取 WSDL 文件，并生成与 Web 服务通信所需的代码。

#### 8.5.4 UDDI

通用发现、说明和集成 (UDDI) 是 Web 服务的黄页。与传统黄页一样，可以搜索提供所需服务的公司，阅读以了解所提供的服务，然后与某人联系以获得更多信息。

UDDI 目录条目是介绍所提供的业务和服务的 XML 文件。UDDI 目录条目包括三个部分。“白页”介绍提供服务的公司：名称、地址、联系方式等；“黄页”包括基于标准分类法的行业类别；“绿页”详细介绍了访问服务的接口，以使用户能够编写应用程序以使用 Web 服务。服务的定义是通过一个称为类型模型 (或 tModel) 的 UDDI 文档来完成的。多数情况下，tModel 包含一个 WSDL 文件，用于说明访问 Web 服务的 SOAP 接口，但是 tModel 非常灵活，可以说明几乎所有类型的服务。

UDDI 允许用户查找提供所需的 Web 服务的公司。如果用户已经知道要与谁进行业务合作，但尚不了解它还能提供哪些服务，这时该如何处理呢？WS-Inspection 规范允许用户浏览特定服务器上提供的 Web 服务的集合，从中查找所需的服务。

#### 8.5.5 Web 服务安全

通常，刚接触 SOAP 的用户提出的第一个问题就是 SOAP 如何解决安全性问题。在其早期开发阶段，SOAP 被看做是基于 HTTP 的协议，所以认为 HTTP 的安全性对于 SOAP 已经足够了。毕竟目前有数以千计的 Web 应用程序都在使用 HTTP 安全性，所以这对于 SOAP 确实已经足够。因此，当前的 SOAP 标准假定安全性属于传输问题，而并不作为安全性问题处理。

当 SOAP 扩展至更为通用的协议，并运行于众多传输之上时，安全性问题就变得突出了。例如，HTTP 提供若干种方法对进行 SOAP 调用的用户进行身份验证，但是当消息从 HTTP 路由到 SMTP 传输时，怎样传播该身份标识呢？SOAP 是作为构造块协议进行设计的，所以幸运的是，已经有了相应的规范以基于 SOAP 为 Web 服务提供额外的安全保护功能。WS-Security 规范定义了一套完整的加密系统，而 WS-License 规范定义了相应的技术，以保证调用者的身份标识，并确保只有授权用户才可以使用 Web 服务。

#### 8.5.6 WS-Routing

Web 服务路由规范 (WS-Routing) 定义了路由 SOAP 消息的机制。SOAP 是一个轻量级的有线传输协议，定义了一系列传输交换机制，用来传输在应用层协议上使用的方法调用。SOAP 实际上没有定义从一点发送消息到另一点的机制，即使在它的规范中引用了一个虚拟的消息路径机制。WS-Routing (以前被称做 SOAP 路由协议) 是一个无状态协议，它扩展了 SOAP 协议，WS-Routing 通过定义一个方法来说明一个预先设计好的路由或传输路径，这个路径将



从消息源开始, 经过若干中介, 最后到达消息的最终接受者。

通过使用 SOAP 扩展模型, 这个基于 SOAP 的规范被设计用于和其他一些机制进行组合从而提供一个完善的消息环境。WS-Routing 没有定义如何实施可靠的和安全的通信方法。不过可以期望其他的 SOAP 扩展规范来提供这些能力。同时, WS-Routing 也不会尝试去超越 SOAP, 定义新的传输方式。

WS-Routing 的设计目标是尽量采用最小的额外代价, 将一个消息路径封装在 SOAP 消息里, 以使这个消息包含足够的信息, 用于在 Internet 上使用诸如 TCP 和 UDP 这样的传输协议来实施消息传输。在 WS-Routing 中, 要支持这样一些消息交换机制:

- SOAP message path model, SOAP 消息路径模型;
- Full-duplex, one-way message patterns, 全双工单向的消息模式;
- Full-duplex, request-response message patterns, 全双工请求/响应消息模式;
- Message correlation, 消息相关。

WS-Routing 没有定义如何实施可靠的和安全的通信方法。不过可以期望其他的 SOAP 扩展规范来提供这些能力。同时, WS-Routing 也不会尝试去超越 SOAP, 定义新的传输方式。

### 8.5.7 WS-Referral

Web 服务指引规范 (WS-Referral) 是一个基于 SOAP 的协议, 用来配置用于转发消息的 SOAP 节点 (SOAP 路由器) 中关于消息路径 (路由条目) 的指令。WS-Referral 是与 Web 服务路由协议 (WS-Routing) 相正交的协议, 它提供了如何配置 SOAP 路由器以建立一个消息路径的方法, 而 WS-Routing 则提供了描述消息实际路径的机制。

WS-Referral 通过提供多种服务, 来协助配置消息路径。SOAP 路由器除了可以提供诸如高性能负载消息传输、集团防火墙服务等转发服务 (relay service) 外, 还会提供诸如负载平衡、镜像、高速缓存以及客户端认证等 Web 服务。举例来说, 一个 Web 服务可以将它的部分职能委派给第三方服务, 而对于用户而言并不知道这些职能是由第三方负载的。

### 8.5.8 DIME 和 WS-Attachments

尽管 XML 的功能十分强大, 但在广阔的应用程序设计领域中, 在很多情况下将非 XML 数据发送到 Web 服务可能更为重要。例如, 将二进制数据 (如 JPEG 图像) 连同 SOAP 消息一并发送出去可能更实用。将数据简单打包到一起同样很有必要。例如, 一个 SOAP 消息可能包含其他 SOAP 消息, 就像人们常常看到的电子邮件中附加了电子邮件一样。直接 Internet 消息封装 (DIME) 规范提供了数据块打包机制。WS-Attachments 定义了如何使用 DIME 在 SOAP 消息中包含附件以及如何在 DIME 数据包中引用这些附件。

## 8.6 SIP 开发实例

### 1. 程序分析

程序分析如下:

```
package org.mitre.jsip;//包
import java.io.*;//引入Java包
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import org.mitre.jsip.event.*; //引入SIP包
public class SipTester extends JFrame implements Runnable { //类
    private SipClient client; //申明一些变量
    private SipUser localUser;
    private SipRegister regServer;
    private static final boolean DEBUG = true;
    private JRadioButton regButton;
    private JRadioButton invButton;
    private JTextArea scrollback;
    private JTextField entry;
    private final String remoteUri = "sip:srjones@blabla.mitre.org"; //地址
    private final String regServerUri = "sip:sturgis.mitre.org";
    private final int [] payloadTypes = {0, 3};
    private final String mcastAddr = "225.5.5.5";
    private final int mcastPort = 23322;
    private static String imUri;
    private SipCall call;
    private SipCallMember call_leg;
    public SipTester (String contact) {
        SipUri uri = new SipUri( contact );
        client = SipClient.getSipClient();
        localUser = null;
        client.setDefaultUserMode( true );
        client.setExplicitProxyMode( false );
        String laddr = Sip.getLocalAddress();
        client.setCallForward( false );
        client.setCallForwardUri( null );
        client.setCallForwardMessage( null );
        client.setMaxForwards( 0 );
        client.setBusyMessage( null );
        client.setHideViaMode( SipClient.HideHop );
        localUser = new SipUser( client, uri );
        client.setDefaultUser( localUser );
        SipUri remoteuri = new SipUri( imUri );
        call = new SipCall( localUser, null );
        call_leg = new SipCallMember( call, remoteuri );
        initSelectGui();
        // 侦听消息
```

```

CallListener cl = new CallListener() {
    public void responseReceived(CallEvent ce) {
        System.out.println("Received response message event");
    }
    public void requestReceived(CallEvent ce) {
        // same with this
        System.out.println("ST: Received request message event");
        SipRequestMessage requestMessage = (SipRequestMessage) ce.getMessage();
        handleRequestReceived( requestMessage );
    }
};
client.addCallListener(cl);
if ( DEBUG ) System.out.println( "SipClient: initialization complete" );
}

protected void initGui() {
    Container cp = this.getContentPane();
    cp.setLayout( new BorderLayout() );
    scrollback = new JTextArea(10, 40); // 10 rows and 40 columns
    cp.add( new JScrollPane( scrollback ), BorderLayout.CENTER );
    JPanel entryPanel = new JPanel();
    entry = new JTextField( 40 );
    JButton sendButton = new JButton( "Send" );
    sendButton.addActionListener( new ActionListener() {
        public void actionPerformed( ActionEvent ae ) {
            String message = entry.getText();
            entry.setText("");
            scrollback.append( localUser.getUri().user() + ": " + message + "\n" );
            sendMessage( message );
        }
    });
    entryPanel.add( entry );
    entryPanel.add( sendButton );
    cp.add( entryPanel, BorderLayout.SOUTH );
}

protected void initSelectGui() {
    Container cp = this.getContentPane();
    cp.setLayout( new BorderLayout() );
    JPanel centerPanel = new JPanel( new BorderLayout() );
    ButtonGroup bg = new ButtonGroup();
    regButton = new JRadioButton();
    regButton.setText( "Send REGISTER" );

```

```

regButton.setSelected( true );
    invButton = new JRadioButton();
invButton.setText( "Send INVITE" );
bg.add(regButton);
bg.add(invButton);
centerPanel.add(regButton, BorderLayout.NORTH);
centerPanel.add(invButton, BorderLayout.CENTER);
JPanel buttonPanel = new JPanel();
JButton okButton = new JButton();
JButton cancelButton = new JButton();
okButton.setLabel( "Execute" );
okButton.addActionListener( new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        executeCommand();
    }
});
cancelButton.setLabel( "Exit" );
cancelButton.addActionListener( new ActionListener() {
    public void actionPerformed( ActionEvent evt ) {
        exitCommand();
    }
});
buttonPanel.add(okButton);
buttonPanel.add(cancelButton);
cp.add( centerPanel, BorderLayout.CENTER );
cp.add( buttonPanel, BorderLayout.SOUTH );
setSize( 240, 150 );
}.

protected void handleRequestReceived( SipRequestMessage requestMessage ) {
SipUri sender = new SipUri( requestMessage.getHeaderData( SipHeader.From ) );
if ( sender == null ) {
    System.err.println( "Don't know where this came from... tossing" );
    return;
}
// 测试即时消息
if ( requestMessage.getMethod() == Sip.MESSAGE ) {
    // put it up on the text area
    String message = requestMessage.messageBody();
    System.err.println("Message received is: [" + message + "]");
    String newText = sender.user() + ": " + message + "\n";
    scrollbar.append( newText );
}

```

```

    }
    }
    public void run() {
// 处理消息
for (;;) {
    while (!client.haveMessages() ) {
    try {
        Thread.currentThread().sleep(3000);
        //          System.out.print(".");
    } catch (InterruptedException ie) {break;}
    }

    System.out.println("Message in queue, forwarding...");
    client.incomingMessage();
}
}

protected void sendRegister(String username, String password) {
// 创建REGISTER响应的侦听器
CallListener regCallListener = new CallListener() {
    public void responseReceived(CallEvent ce) {
        System.out.println( "REGCALLLISTENER activated" );
        // 判断是否为响应消息
        SipResponseMessage respMessage = (SipResponseMessage)ce.getMessage();
        if ( respMessage.getStatus().getCode() == 401 ) {
            System.out.println("REGCALLLISTENER - updating status");
            try {
                regServer.setAuthentication(respMessage);
            } catch ( RegisterException re ) {
                System.err.println( re.toString() );
                return;
            }
            try {
                regServer.requestRegister( false );    // 尝试再次注册
            } catch ( RegisterException re ) {
                System.err.println("Couldn't register: " + re );
                return;
            }
        }
    }
}

public void requestReceived(CallEvent ce) {
    // 忽略

```

```

    }
    };
    SipClient.getSipClient().addCallListener( regCallListener );
    //创建REGISTER消息给服务器
    regServer = localUser.addServer( regServerUri );
    try {
        regServer.requestRegister( password );
    } catch ( RegisterException re ) {
        System.err.println( re );
    }
    }
    protected void sendMessage( String message ) {
        call_leg.requestMessage( message );
    }
    protected void sendInvite() {
        // 创建并发送INVITE消息
        SdpMessage sdpMessage = new SdpMessage();
        sdpMessage.setName("jSIP");
        sdpMessage.setUsername("steve");
        sdpMessage.addMedia( SdpMessage.AUDIO, "0", "RTP/AVP", mcastPort);
        sdpMessage.setIpAddress(mcastAddr);
        sdpMessage.setPort(mcastPort);
        SipUri remoteuri = new SipUri(remoteUri);
        SipCall call = new SipCall( localUser, null );    // this is for creating a brand new call
        SipCallMember member = new SipCallMember( call, remoteuri );
        member.requestInvite( sdpMessage.message(), new MimeType( "application/sdp" ) );
    }
    protected void executeCommand() {
        if ( regButton.isSelected() ) {
            sendRegister("user1", "none"); // junk
        } else if ( invButton.isSelected() ) {
            sendInvite();
        } else {
            System.out.println("Don't know which command to execute");
        }
    }
    public void exitCommand() {
        // try to unregister from the registry server
        try {
            regServer.requestRegister( true );
        } catch ( RegisterException re ) {}
    }

```

```
System.exit( 0 );
}

public static void main(String[] args) {
// 测试IM
String reginfo = args[0];    // this consists of username:password
imUri = args[1];
int colon = reginfo.indexOf( ':' );
if ( colon < 0 ) {
    System.err.println( "First argument should be in form username:password" );
    System.exit( 1 );
}
String contact = reginfo.substring( 0, colon );
String password = reginfo.substring( colon + 1 );
final SipTester tester = new SipTester( contact );
tester.setTitle( "Chat with " + imUri );
// 启动消息队列线程
Thread messageThread = new Thread( tester );
messageThread.start();
tester.pack();
tester.setVisible( true );
// 当窗口关闭时, 退出程序
tester.addWindowListener( new WindowAdapter() {
    public void windowClosed( WindowEvent e ) {
        tester.exitCommand();
    }
} );
//尝试注册
tester.sendRegister( contact, password );
}
}
```

## 2. 界面

图 8-6 为登录界面, 图 8-7 为选择同伴地址界面, 图 8-8 为消息对话界面。

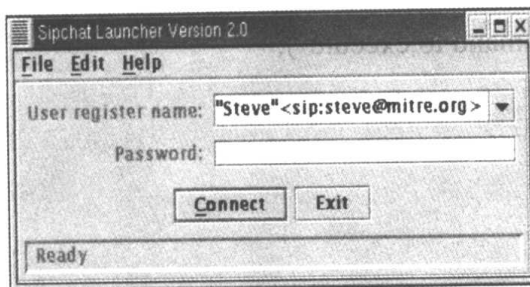


图 8-6 登录界面

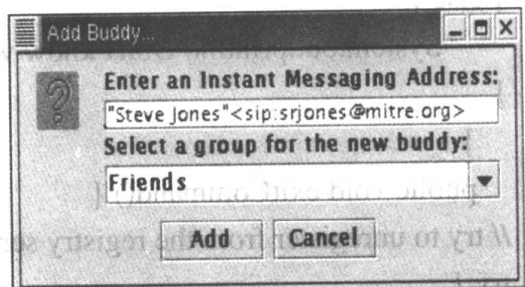


图 8-7 选择同伴地址界面

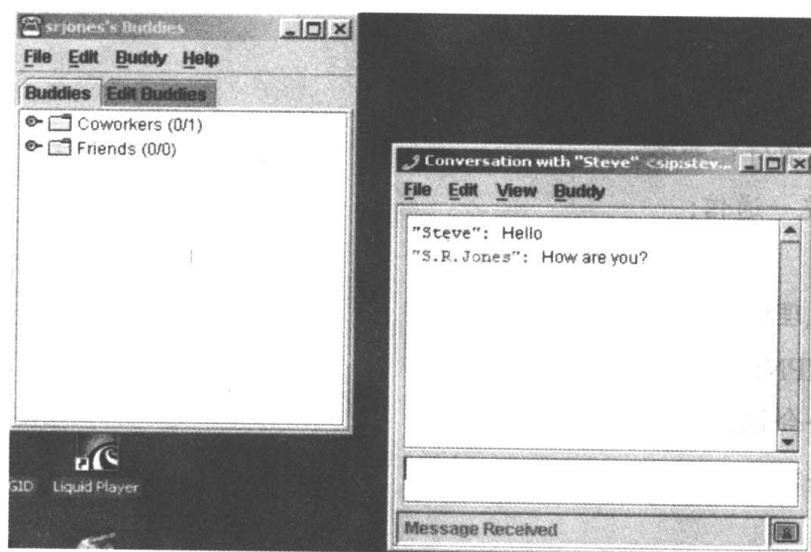


图 8-8 消息对话界面

## 8.7 SIP 业务部署

### 8.7.1 SIP 业务部署模型

SIP 业务部署模型如图 8-9 所示。

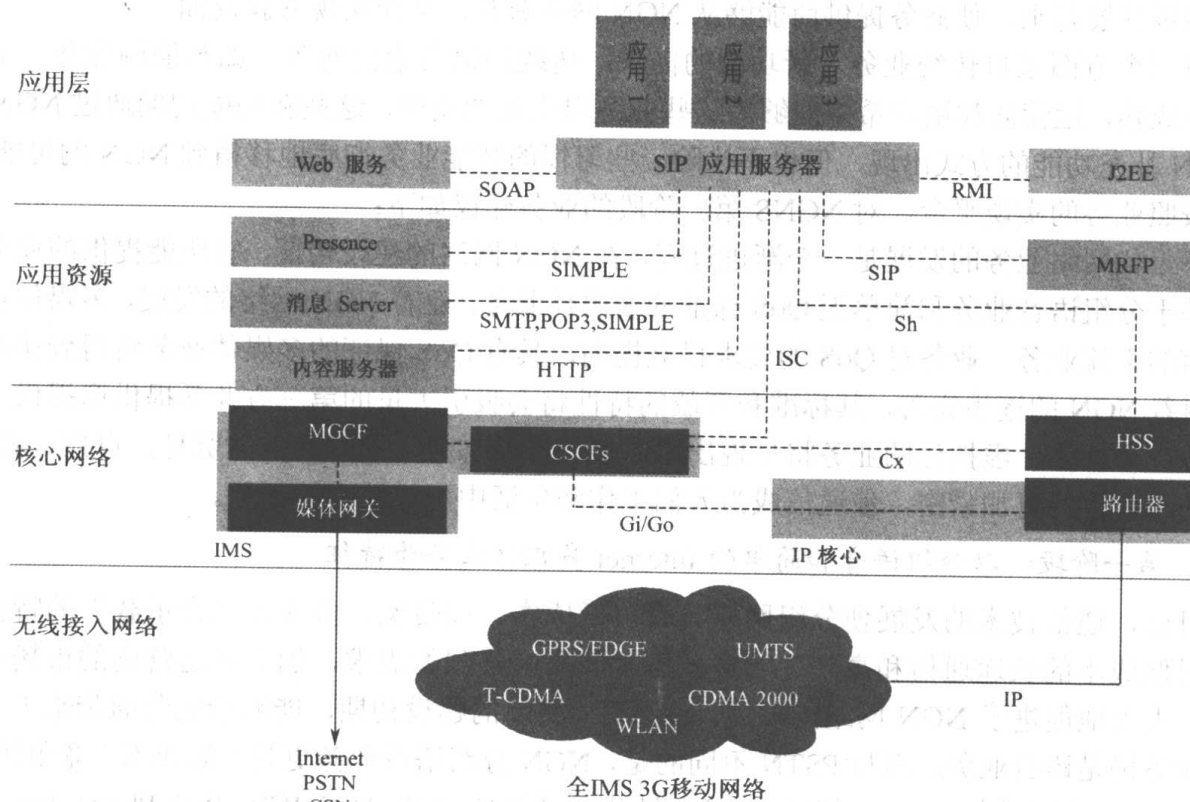


图 8-9 SIP 业务部署模型

### 8.7.2 SIP 业务

可部署的业务如下：



- 呈现业务;
- Web 800;
- 基于 SIP 的预付费业务;
- 统一消息/统一通信;
- 即时消息;
- 个人通信代理;
- 综合通信 VPN;
- 多媒体综合会议;
- 多媒体消息;
- Web 会议。

### 8.7.3 SIP 业务发展步骤

在 NGN 业务发展的第一阶段, 业务开展的推动力来自三个方面:

第一个方面来自运营商: 为了吸引用户尝试 NGNS, 运营商开始提供语音和 Internet 结合的业务, 体现 NGN 分组业务的承载优势。

第二个方面来自业务提供商: 由于 NGN 网络提供开放的 API 接口, 因此吸引着 ICP 向 NGNS Developer/Provider 转变。基于共建价值链的需要, 那些充满个性和能带来收益的新型业务逐步开展起来, 使业务提供商能够从 NGN 网络获利, 从而实现双赢局面。

第三个方面来自传统业务对新功能的需求: 传统的语音增值业务 (如智能网业务) 的发展已经成熟, 运营商和用户希望能够为这些业务提供新的功能, 这些新功能初期通过 NGN 提供的 IN 补充功能的方式出现, 但也不排除一些习惯的智能业务彻底地移植到 NGN 的可能。

按照业务的实现平台, 对 NGNS 第一阶段的业务建议如下:

下一代网络业务的发展是一个渐进过程, 在 NGN 网络的建设初期, 它所能提供的业务主要是基于分组语音业务和简单无 QoS 保证的多媒体业务; 随着 NGN 网络的发展, 多媒体将成为通信的主流业务, 业务对 QoS 的要求日益提高, 具有 QoS 保证的多媒体业务将得到快速发展; 随着 NGN 的逐步完善, 其标准和开放的特性将会吸引大量的第三方业务提供商提供丰富的业务, 个性化、虚拟化的业务将呈现出爆炸性的增长态势, 同时通信与贸易、自控、教学、医疗的结合也将更加紧密, 使通信成为人们工作和生活中不可缺少的部分。

#### 1. 第一阶段: 以分组语音和简单的 Internet 多媒体业务为特征

目前, 通信技术的发展使分组网络具有“低成本、高带宽、多业务综合承载”的特性, 人们对低成本的长途通信和高带宽的多业务承载具有强烈的需求, 加上多运营商的市场竞争环境, 大大地促进了 NGN 网络的发展。在 NGN 网络的建设初期, 能够给运营商带来丰厚利润的业务仍是语音业务, 而与 PSTN 不同的是, NGN 分组语音业务更具“低成本”竞争优势。在这一阶段, 将语音与 Internet 相结合, 可大量开展诸如 Click To Dial, ICW, Web Phone, Presence, Personal Communication Assistant 等业务, 而在多媒体业务方面, 则可提供廉价的无 QoS 保证的可视电话、会议电视等, 造成 QoS 无法保证的主要原因是因为该阶段的 IP 网络自身无法提供理想的 QoS 保证。

此外, 在这一阶段, PSTN 与分组网络形成互补局势, PSTN 不具有高带宽、低成本特性, 而分组网络则不具有广覆盖、QoS 保证的特性, 两者具有互补能力。例如, 在分组语音承载

过程中,可在本地选择分组网、长途选择 PSTN,或者在发端选择分组网、在收端选择 PSTN。

在这一阶段,最具有吸引力的业务是那些语音与 Internet 相结合的业务。

### 2. 第二阶段:以有 QoS 保证的多媒体业务为特征

随着分组网络的逐渐完善,网络开始能够以较为低廉的成本提供具有 QoS 保证的通信业务,人们对通信的要求也开始从“成本为主”转变为“成本、质量并重”的阶段。在这一阶段,由于 QoS 取得了革命性突破,促使电子商务、远程医疗、远程教学、远程控制、高质量会议电视等业务飞速发展,通信给许多行业带来革命性的变化,真正意义上的电子商务开始普及,运营商的通信收入将由以语音为主转变为以多媒体为主。

由于电子商务除了涉及网络运营商之外,还涉及业务、接入和内容等诸多运营商,从而导致两个巨大变化:

- 网络的开放性大大提高,以 Parlay 为代表的开放 API 将得到广泛应用;
- 新型通信网络价值链将逐步替代目前的简单通信网络价值链。

### 3. 第三阶段:以个性化、虚拟化业务为特征

高利润电子商务和远程医疗等宽带应用将吸引众多的业务提供商设计出更具个性化和专业化的业务,同时虚拟运营商的出现使客户不必面对多个网络运营商或业务提供商,效率大大提高。

随着业务的进一步发展,即使人们不在同一办公室也可方便地进行沟通,于是出现了虚拟办公室,业务供应商将根据虚拟办公的需求开发出更为适合的软件,同样虚拟社区、虚拟校园等业务也将逐步展开。

在这一时期,通信行业的含义开始发生变化,远比今天的含义广泛得多。

## 8.8 小结

本章主要介绍了利用 SIP 进行业务开发的几种典型方法,如呼叫控制语言(CPL: 呼叫处理语言)、SIP CGI(公共网关接口: Common Gateway Interface)、SIP Servlet。另外也介绍了和 SIP 应用相关的 J2EE 技术和 Web 服务技术,最后还阐述了 SIP 业务部署步骤。

## 8.9 习题

1. 简述 SIP 业务开发的几种典型方法。
2. 简述 CPL 组成。
3. 简述 SIP Servlet 开发步骤。

## 第9章 SIP 产品现状及未来发展

知识点:

- SIP 产品现状
- SIP 不足
- SIP 未来发展

本章将主要介绍目前 SIP 相关产品的现状、SIP 不足以及对 SIP 的未来加以展望。

### 9.1 SIP 产品现状

法国电信公司 Cegetel 在 SuperComm 展示会上演示了一套 SIP 多媒体消息系统, 采用这套系统, PDA、笔记本和移动电话都可以非常方便地通过公网通信。该公司认为, 这套系统不能通过 H.323 实现, 因为该协议占用了 PDA 上太多的资源。

#### 9.1.1 具有代表性的 SIP 产品

目前值得关注的国外比较具有代表性的 SIP 产品包括如下几个方面。

##### 1. Dynamicsoft

SIP 产品开发商 Dynamicsoft 是 Cegetel 的合作者, 现在成为了 3Com 的产品开发商。该公司致力于为 SIP 产品设计和制造开发平台供其他厂商在此基础上开发市场需要的新兴语音服务。Dynamicsoft 认为 SIP 更适合移动设备市场。它的观点如下:

- 语音和多媒体数据将不可避免地由 IP 承载。
- 新型服务包括即时消息、语音会议和统一消息等。
- SIP 协议比 H.323 协议更适合用于呼叫控制。
- SIP 可以作为第 3 代通信服务的呼叫控制协议。

##### 1. 3Com

3Com 公司在 2004 年 4 月 25 日宣称已成功完成其 SIP 产品的第 4 次测试, 这次测试的目的是为了检验软件和硬件的 SIP 互操作性。这是第一次整个电话系统都建立在 SIP 基础上的测试, 如通过一台 SIP 网关访问 PSTN 等。由于 SIP 已经获得了各个电信设备开发商、电信运营商和语音服务商的青睐和支持。相关产品层出不穷, 使得此类产品的互操作性成为产品遵从 IETF 标准的关键。

SIP 是一种开放的 IP 电话信令协议。SIP 实现了 CLASS 服务 (Custom Local Area Signaling Service), 如呼叫阻塞、呼叫方 ID 和呼叫转移等。这些服务通过 SIP 技术实现比在 PSTN 上实现同样的服务快捷得多。

这次测试的厂商有: 3Com, Audio Talk Networks, BroadSoft, Catapult Communications,

Cisco, Columbia University, Delta Information Systems, Dialogic, dynamicsoft, Ericsson, Hewlett-Packard, Hughes Software, Indigo Software, Iwatsu Electric, Komodo, Lucent, MCI, Mediatrix, Microappliances, Mitel, Motorola, NetSpeak, Network Associates, Nortel, Nuera, Oy LM Ericsson Ab, OZ, Pingtel, Siemens, SS8, Startek, Telogy, Ubiquity, VegaStream 和 Vovida。

### 3. Siemens

Siemens 集团下的 Siemens Information and Communication Networks 公司于今年 5 月 9 日正式向业界推出了 OptiPoint 100 Advance IP 电话, 这是一款采用 SIP 协议的 IP 电话。这套产品完全符合 SIP 的要求, 呼叫可以从 PSTN、Intranet 或者 Internet 发起。此类电话产品, 包括电话机、客户程序和通信服务器、网关都可应用于企业通信系统和电信规模的 IP 电话系统。

OptiPoint 100 Advance phone 内置适配器直接连接到 IP 网络。系统具有的 OptiGuide 界面, 允许用户方便地从电话的 24 字符 LCD 上显示的菜单中选择电话特性。

OptiPoint 100 Advance phone 的 SIP 协议栈由 Mediatrix Telecom 公司研制。这是一家设计和部署低密度 IP 网关和终端适配器的公司。

### 4. ComGates

ComGates 推出的 CMG/CSS 2000 是一种具有多协议能力的软交换产品, 它采用了专有的一套独特处理过程, 实时地、自动地交换已有全部标准下的通信数据流。

CMG/CSS 2000 对具有不同版本的 H.323、SIP、MGCP 协议的电话网络具有互操作性。所以这些协议网络之间可以交换数据流。

### 5. Cisco

Cisco 将 SIP 功能嵌入了其 AS5300 接入服务器、Cisco 2600 和 Cisco 3600 系列路由器, 这些嵌入功能负责 IP 网络上语音和多媒体呼叫的信令控制, SIP 特性则由 Cisco IOS Release 12.1(1)T 引入以作为 VoIP 网络软件。Cisco 的 SIP 软件具备 SIP 协议的固有优势, 在个人移动服务上功能强大。

Cisco 的 SIP 特性已经接受了与其他开发商产品的互操作测试, 这些特性包括以下功能: 可配置带内警告, 最大 SIP 重定向数目设置, 可选 SIP 或者 H.323, 支持 SIP 消息的 UDP 和 TCP 传输, 强大的调试功能, 支持 DNS SRV 记录解析 SIP 服务器主机名, 可配置 SIP 消息计时和重试。这套 SIP 功能使得 Cisco 语音平台可以支持电信级语音质量的编码能力。虽然 SIP 比 H.323 简单, 但是 SIP 提供了类似的呼叫控制能力。

### 6. Lucent

Lucent 实现了一套供第三方开发者采用 SIP 的可编程软件平台: Elemedia SIP Server, 它是基于 Lucent 软交换功能的新产品, Elemedia Server 支持 SIP 全部标准服务和组成元素。Elemedia 是 Lucent 集团下的一家为服务供应商开发基于标准的、电信级多媒体通信软件平台的公司。Elemedia 产品都建立在 Bell 实验室的研究基础之上。

除了用户认证、重定向和注册服务之外, Elemedia SIP Server 还支持传统的电话服务, 例如, 个人移动、定时路由和基于被叫人员地理位置的呼叫转移。Elemedia SIP Server 可以让服务供应商将基本的 IP 电话服务集成到 Web、E-mail 和聊天服务中。

Elemedia SIP Server 改进了 Lucent 的 softswitch API, 这些 API 是 Lucent Full Circle 项目的一部分, 而且可以控制 Elemedia Media Server 以提供丰富的媒体服务。

## 7. Hughes

Hughes 公司位于印度 Bangalore 的开发中心关注 Internet 领域的产品和服务开发, 不过他们现在对电信产品的开发也越来越有兴趣了。近来推出的产品 EZ Phone 就是一款基于 SIP 的网络电话产品。

EZ Phone 是使用 SIP 的客户-服务器方式的 Internet 电话。电话服务器可以驻留在机构内已经存在的其他服务器上, 由用户建立和取消呼叫。服务器软件与 Intranet Web 服务器接口以提供基于 Web 的用户注册。客户程序则安装在用户的工作站上。

## 8. SS8 Networks

SS8 已经率先向其可能的合作伙伴和客户发布了免费的 SIP 客户和服务器协议栈。该协议栈在 Chicago 的第 4 次 SIP 产品互操作特性测试中已经成功地经过了与其他 SIP 产品生产商互连的测试。

SIP Bake-off 测试是检验采用了 SIP 协议的 IP 电话产品 (包括软件和硬件) 的互操作性的工业测试。

SS8 Networks 将 SIP 协议完全地整合进它的新一代 IP 电话信令产品中。SS8 的解决方案为 IP AIN 服务提供了新结构。其产品已经具备了与其他相关信令协议的无缝接口。这些协议包括 SS7、SIP 和 H.323 等。

## 9. 微软

在 Voice on the Net 大会的基调发言中, 微软公司嵌入式系统集团总经理 Peyton Smith 描述了企业的电话系统和计算机系统在一个通用网络环境下相互操作的具体方法。这一趋势的中心内容将是 Session Initiation Protocol (SIP) 协议的应用。SIP 协议的设计目的是使企业能够在一个单一系统中实现语音、数据和消息的交换。Smith 称, SIP 将改变一切。

Smith 是在微软公司发布 Greenwich 服务器软件数周后发表上述评论的。Greenwich 服务器软件将使企业具有支持使用 SIP 协议的保密即时消息的能力。

Greenwich 软件的未来版本将整合由这种标准支持的其他通用企业应用功能, 如基于 Internet 协议 (IP) 的语音通话、视频会议等。微软公司和其他诸如 IBM 的 Lotus 等企业软件开发公司将 SIP 协议, 称为即时消息互操作性的主要推动力量和单一服务器系统环境下将各种不同形式的通信联系在一起的基础。

### 9.1.2 SIP 防火墙及 NAT

Acme Packet

Aravox Technologies, Inc.

Cosystems Technologies Pvt. Ltd.

Ingate Systems

Intertex Data AB

Kagoor Networks

Netrake

Nexcell Telecom Co., Ltd.

Ridgeway Systems & Software

### 9.1.3 SIP 网关

AddPac Technology

Adtech

Antepo, Inc.

Arrayvox Communications

AudiCodes Ltd.

Azatel Communications Inc

Bitwise Communication

Blue Sky Labs

CIRPACK

Cisco Systems

Columbia University Computer Science

Comgates

Cyber Leap Data Solutions Limited

Cybersys Computer Ltd.

Dataflex

Dialogic

Digital Talk, llc

e-horizon Streaming Technologies

Elesign Inc.

empowerTel Networks

Intel Corp

Interactive

Komodo Technology (Cisco Systems)

Lucent Technologies

Mediatrix

MIP Telecom

Mockingbird Networks

net.com

Nortel Networks

Nuera

Quintum Technologies, Inc.

sentitO Networks

Siemens

SyncSys, Inc.

Syndeo

Telecom Technologies

VegaStream

VIVE Synergies Inc.

VocalData

Voyant Technologies

#### **9.1.4 SIP 服务器**

3Com

Acme Packet

Aspect Communications

Asterisk

AT&T Labs

aTelo

AudioCodes

Broadsoft

Carnegie Mellon University

Centile Inc.

Cisco Systems

Cognitronics Network Media Servers

Columbia University Computer Science

Convedia Corporation

Dataflex

Dialexia Communications, Inc.

dynamicsoft

e-horizon Streaming Technologies

Ericsson

eStara

EyeP Media S.A.

HCL Technologies Ltd

Hewlett-Packard Labs

HotSip

Hughes Software Systems

Indigo Software

Infoglyptic Software Pvt Ltd

inPACT technologies

Interactive

IP Unity

Iperia

iptel.org

LongBoard

Lucent Technologies

Mailvision

MasterMind Technologies

Microappliances

MIP Telecom  
Mistral Software  
Mockingbird Networks  
NetCentrex, Inc.  
NexTone Communications  
Nortel Networks  
Nuance  
Object Software  
Pactolus  
Porta Software  
RADVision  
serVonic GmbH  
SIP Servers  
ss8 networks  
Sylantro  
Syndeo  
Tekelec  
Ubiquity  
Vovida  
Voyant Technologies  
Webkonferenzen

### 9.1.5 SIP 服务

3GSOFTPHONE SERVICES  
A.N Telecom (UK) LTD  
aTeloBroadsoft  
Centile Inc.  
Comverse  
Dataflex  
deltathree  
Digital TALK  
Digital Talk, llc  
eDial  
eStara  
Level3 Communications  
Mistral Software  
Pactolus  
Siemens Switzerland Ltd.  
SipTelecom  
ss8 networks



Tario.Net

Telia

Ubiquity

Webley Systems

Webley Systems, Inc.

WorldCom, Inc.

### **9.1.6 SIP 组件**

Aspect Communications

aTelo

Blue Sky Labs

Broadsoft

Circle24 Technologies

Cisco Systems

Columbia University Computer Science

Data Connection

Dataflex

Dexcel Electronics Designs

dynamicsoft

ECI IP Inc.

empowerTel Networks

Gallery IP Telephony, Inc.

HCL Technologies Ltd

HelloSoft

Hughes Software Systems

Indigo Software

Interactive

LongBoard

Lucent Technologies

Mailvision

MasterMind Technologies

MCK Advance Technologies Ltd

Mistral Software

National Institute of Standards and Technology

Net Brahma Technologies Pvt. Ltd.

Netergy Networks

Nortel Networks

Nuera

Ortena Networks

Pactolus

RADVision

SASKEN Communication Technologies

ss8 networks

Sylantro Systems

Syndeo

T!Semantics

Telecom Technologies

Telogy Networks

Trillium Digital Systems

VocalData

Vovida

Vovida (Cisco Systems)

Wind River Systems

Wipro Technologies

### 9.1.7 SIP 软件工具

Agilent Technologies

Catapult Communications

Delta Protocol Test Solutions

Empirix Inc.

EyeP Media S.A.

Hotfoon

Inet Technologies, Inc.

RADCOM

RADVision

SIP Communications, Inc.

T!Semantics

### 9.1.8 SIP 用户代理

3Com

Adtech

Alcatel

AT&T Labs

Avaz Networks

BCM

Cisco Systems

Columbia University Computer Science

Dexcel Electronics Design

dynamicsoft

e-horizon Streaming Technologies

Encore Software Ltd.  
Ericsson  
eStara  
EyeP Media S.A.  
GMD Fokus  
Grandstream Networks, Inc.  
HCL Technologies Ltd.  
HotSip  
Hughes Software Systems  
Indigo Software  
Interactive  
Komodo Technology (Cisco Systems)  
Lucent Technologies  
Mediatrix  
Microappliances  
MIP Telecom  
Mitel Networks  
Nokia Research  
Nortel Networks  
PingTel  
Polycom  
Siemens Switzerland Ltd.  
Signal Communication, Inc.  
Snom Technology AG  
SyncSys, Inc.  
theKompany.com  
Ubiquity  
University College London Computer Science  
VocalData  
Vovida  
Xten Networks Inc.

## 9.2 SIP 未来发展

目前 SIP 协议并没有在电信领域大规模商用的先例，主要是受以下几个方面的影响：

- SIP 终端的智能性过强，可能会给网络带来很多安全问题；
- 用户终端可以绕开运营商的管理直接向被叫用户发起呼叫；
- 其网络结构采用平面化形式，运营商不便进行管理和计费；
- 与 H.323 相比，SIP 不具备点到多点的会话控制能力，如控制一方成为主席，或点名一方发言等。

## 9.3 小结

本章介绍了 SIP 产品种类及 SIP 相关厂商。分析了 SIP 的现状和不足，如终端强智能性、终端旁路网络、网络结构及会话控制能力。目前 SIP 规范仍在不断地得到完善，SIP 协议、产品、系统也不断地推出并在实际网络中得到应用与检验。

## 9.4 习题

1. 简述 SIP 产品种类及现状。
2. 分析 SIP 的一些不足。

## 附录 A 常见缩略语

AAA Authorization, Authentication, and Accounting	授权、认证与记账
AIN Advanced Intelligent Network	高级智能网络
AM Account Management	账号管理
API Application Programming Interface	应用程序接口
APP Application	应用程序
AS Access Session	访问会话
ATM Abstract Test Method	抽象测试方法
ATS Abstract Test Suite	抽象测试集
CAMEL Customized Application for Mobile Network	移动网络增强逻辑的客户化应用
CC Call Control	呼叫控制
CORBA Common Object Request Broker Architecture	公共对象请求代理结构
CSE Camel Service Environment CAMEL1	业务环境
DIME Direct Internet Message Encapsulation	直接 Internet 消息封装
EMS Enhanced Messaging Service	增强消息服务
ES ETSI Standard	ETSI 标准
ETSI European Telecommunications Standards Institute	欧洲电信标准化研究所
FW FrameWork	框架
GPRS General Packet Radio System	通用分组无线系统
GSM Global System for Mobile Communications	全球移动通信
HE Home Environment	归属环境
HE-VASP Home Environment Value Added Service Provider	归属环境增值业务提供者
HLR Home Location Register	归属位置寄存器
ICS Implementation Conformance Statement	实现一致性申明
IDL Interface Description Language	接口描述语言
IETF Internet Engineering Task Force	Internet 工程任务组
IIOP Internet Inter-ORB Protocol	Internet ORB 之间协议
IM Instant Messaging	即时消息
IN Intelligent Network	智能网络
INAP Intelligent Networks Application Part	智能网络应用规程
IP Internet Protocol	Internet 协议
ISP Internet Service Provider	Internet 服务提供者
ISUP ISDN User Part	ISDN 用户规程
IT Information Technology	信息技术
IUT Implementation Under Test	实现测试

IVR Interactive Voice Response	交互语音应答
IXIT Implementation eXtra Information for Testing	额外实现测试
JAIN Integrated Network APIs for the Java platform	Java 平台集成网络 API
JCP Java Community Process	Java 通信过程
LT Lower Tester	底层测试
MAP Mobile Application Part	移动应用规程
ME Mobile Equipment	移动设备
MExE Mobile Station (Application) Execution Environment	移动站 (应用) 执行环境
MIME Multipurpose Internet Mail Extensions	多用途 Internet 邮件扩展
MM7 Communication protocol between MMS Relay/Server and MMS Application Server MMS	中继/服务器与 MMS 应用服务器 之间通信的协议
MMS Multimedia Message Service	多媒体消息服务
MMS-C Multimedia Message Service Center	MMS 中心
MPS Mobile Positioning System	移动定位服务
MS Mobile Station	移动台
MSC Mobile Switching Center	移动交换中心
MSISDN Mobile Station ISDN Number	移动站 ISDN 号码
OASIS Organization for the Advancement of Structured Information Standards	高级结构化信息标准组织
OMA Open Mobile Alliance	开放移动联盟
OSA Open Service Architecture	开放业务体系
PICS Protocol Implementation Conformance Statement	协议无关申明
PIN Personal Identification Number	个人标识号码
PINT PSTN and Internet inter-networking	PSTN 与 Internet 互连
PLMN Public Land Mobile Network	公众陆地移动网
PSE Personal Service Environment	个人业务环境
PSTN Public Switched Telephone Network	公共交换电话网络
SAML Security Assertion Markup Language	SSTC 开发的安全标记语言
SAT SIM Application Tool-Kit	SIM 应用工具包
SCF Service Capability Feature	业务能力特征
SCP Service Control Point	业务控制点
SCS System Conformance Statement	系统遵循申明
SCS Service Capability Server	业务能力服务器
SIM Subscriber Identity Module	用户标识组件
SIP Session Initiation Protocol	会话发起协议
SIP-T Session Initiation Protocol for Telephone	SIP 电话协议
SLA Service Level Agreement	服务级别
SMS Short Message Service	短信业务
SMS-C Short Message Service Center	SMS 中心

SMTP Simple Mail Transfer Protocol	简单的邮件传送协议
SOAP Simple Object Access Protocol	简单对象访问协议
SPAN Services and Protocols for Advanced Networks	高级网络业务协议
SSTC Security Services Technical Committee (OASIS)	OASIS 安全业务技术委员会
SUT System Under Test	系统测试
TDTF Telecom Domain Task Force	电信任务组
TP Test Purpose	测试目的
TSM Trust and Security Management	可信与安全管理
TSS Test Suite Structure	测试集结构
UCP Universal Computer Protocol	通用计算机协议
UDDI Universal Description, Discovery and Integration	通用描述、发现与集成
URI Uniform Resource Identifier	通用资源定位
USIM User Service Identity Module	用户业务标识组件
VASP Value Added Service Provider	增值业务提供者
VHE Virtual Home Environment	虚拟归属环境
VLR Visited Location Register	拜访位置寄存器
WAP Wireless Application Protocol	无线应用协议
WG Working Group	工作组
WGP Wireless Gateway Proxy	无线网关代理
WGS 84 World Geodetic System 1984	世界地理系统 1984
WPP Wireless Push Proxy	无线 Push 代理
WS Web Service	Web 服务
WSDL Web Service Definition Language	Web 服务描述语言
WS-I Web Services-Interoperability Organization	Web 服务互操作性组织
W3C World Wide Web Consortium	全球 Web 联盟
XML Extensible Markup Language	可扩展标记语言
3GPP 3rd Generation Partnership Project	第三代移动通信合作组织工程

## 附录 B SIP 相关网址

### 1. SIP 标准组织地址

IPCC (ISC) 论坛: <http://www.softswitch.org>

Jonathan Rosenberg's Home Page: // <http://www.jdrosen.net/>

The SIP Center: <http://www.sipcenter.com/>

SIP: Session Initiation Protocol page: <http://www.cs.columbia.edu/sip>

SIP 论坛: <http://www.sipforum.org/>

SUN JAIN: <http://java.sun.com/products/jain/>

IETF 组织: <http://www.ietf.org>

ITU 组织: <http://www.itu.org>

IETF PINT working group: <http://www.ietf.org/html.charters/pint-charter.html>

3GPP for third-generation wireless networks: <http://www.3gpp.org/>

International Multimedia Telecommunications Consortium (IMTC): <http://www.imtc.org/>

DTS/TIPHON-03018 - Interface Protocol Requirements Definition; Implementation of TIPHON architecture using SIP : <http://docbox.etsi.org/tech-org/tiphon/Document/tiphon/07-drafts/wg3/DTS03018/>

PacketCable: <http://www.packetcable.com/>

IETF MMUSIC (Multiparty Multimedia Session Control) working group: <http://www.ietf.org/html.charters/mmusic-charter.html>

IETF SIP working group: <http://www.ietf.org/html.charters/sip-charter.html>

impp (Instant Messaging and Presence Protocol)WG: <http://www.ietf.org/html.charters/impp-charter.html>

iptel WG: <http://www.ietf.org/html.charters/iptel-charter.html>

Dean Willis 主页: <http://www.softarmor.com/sipwg>

HotSip, products for SIP, -presenceservices and -infrastructure, [www.hotsip.com](http://www.hotsip.com)

VOIP FORUM, [www.voip-forum.com](http://www.voip-forum.com)

GecKo's IP xStream - IP telephony news and market research, [www.iptelephony.org](http://www.iptelephony.org)

Vocal, [www.vovida.org/applications/downloads/vocal](http://www.vovida.org/applications/downloads/vocal)

### 2. SIP 会议及其网站

SIPit 12 , [www.hotsip.com/sipit12/](http://www.hotsip.com/sipit12/)

SIPit 13, 2003-08-18 to 22, Ottawa, Canada, [www.mitel.com/sipit/](http://www.mitel.com/sipit/) \*

SIPit 14, 2004-02-08 to 13, Cannes, France\*

SIPit 15, 2004-09, Taiwan \*

Fall 2003 VON, Boston, MA, September 22-25, [www.pulver.com/von/](http://www.pulver.com/von/) \*

SIPop! 2003, colocated with Fall 2003 VON, [www.pulver.com/sipop/](http://www.pulver.com/sipop/)



Spring 2004 VON, Santa Clara, CA, March 29 – April 1

NYSSA Telecom/VoIP Industry Conference, 2003-12-03, NYC, [www.nyssa.org/events/120303-telecom-voipindustry-conference.html](http://www.nyssa.org/events/120303-telecom-voipindustry-conference.html)

Internet Telephony Conference and Expo, Long Beach, CA, October 14-16, 2003, [www.tmcnet.com/itexpo/](http://www.tmcnet.com/itexpo/)

The pulver conference calendar, [www.pulver.com/conference/](http://www.pulver.com/conference/)

### 3. SIP 产品及其网站

Bonephone IPv6 SIP phone, [www.iptel.org/products/bonephone/](http://www.iptel.org/products/bonephone/) \*

Ingate, [www.ingate.com](http://www.ingate.com)

Intertex, [www.intertex.se](http://www.intertex.se)

KPhone IPv6 SIP phone, [www.iptel.org/products/kphone/](http://www.iptel.org/products/kphone/) \*

linphone, [www.linphone.org/](http://www.linphone.org/) or [simon.morlat.free.fr/](http://simon.morlat.free.fr/) \*

more IP telephony products, [www.iptel.org/info/products/](http://www.iptel.org/info/products/) \*

Simple Traversal of UDP through NATs (STUN server), [www.vovida.org/applications/downloads/stun/](http://www.vovida.org/applications/downloads/stun/)

SIP Express Router (SER), [www.iptel.org/ser](http://www.iptel.org/ser) \*

SIP hard phones, [www.iptel.org/info/products/index.php?category=hardphone&name=Hardphones&siponly=1](http://www.iptel.org/info/products/index.php?category=hardphone&name=Hardphones&siponly=1)

SIP Residential Gateway (SIPRG), [www.vovida.org/applications/downloads/siprg/](http://www.vovida.org/applications/downloads/siprg/)

SIP soft phones,

[www.iptel.org/info/products/index.php?category=softphone&name=Softphones&siponly=1](http://www.iptel.org/info/products/index.php?category=softphone&name=Softphones&siponly=1)

SIP software products, [www.iptel.org/products/](http://www.iptel.org/products/) \*

SIP User Agent (SIPSet), [www.vovida.org/applications/downloads/sipset/](http://www.vovida.org/applications/downloads/sipset/)

SIPTiger, [www.vovida.org/applications/downloads/siptiger/](http://www.vovida.org/applications/downloads/siptiger/)

SJPhone, [www.sjlabs.com/](http://www.sjlabs.com/)

Snom, [www.snom.com](http://www.snom.com)

SOFTFRONT-VoIP technology, [www.softfront.co.jp/en/tech/ipv6.html](http://www.softfront.co.jp/en/tech/ipv6.html)

The Vovida Open Communication Application Library (VOCAL), [www.vovida.org/applications/downloads/vocal/](http://www.vovida.org/applications/downloads/vocal/) \*

6VOICE IPV6 SIP phone, [www.telscom.ch/6voice/6voice\\_version.htm](http://www.telscom.ch/6voice/6voice_version.htm) \*

VegaStream, [www.vegastream.com](http://www.vegastream.com)

VON's SIP product list, [www.pulver.com/products/sip/](http://www.pulver.com/products/sip/)

## 附录 C SIP servlet 类层次及接口层次

### 1. 类层次

```
class java.lang.Object
    class java.util.EventObject (实现 java.io.Serializable)
        class javax.servlet.sip.SipApplicationSessionEvent
        class javax.servlet.sip.SipErrorEvent
        class javax.servlet.sip.SipSessionBindingEvent
        class javax.servlet.sip.SipSessionEvent
    class javax.servlet.GenericServlet (实现 java.io.Serializable, javax.servlet.Servlet,
        javax.servlet.ServletConfig)
        class javax.servlet.sip.SipServlet
class java.lang.Throwable (实现 java.io.Serializable)
    class java.lang.Exception
        class javax.servlet.ServletException
            class javax.servlet.sip.Rel100Exception
            class javax.servlet.sip.ServletParseException
class javax.servlet.sip.TooManyHopsException
```

### 2. 接口层次

```
interface java.lang.Cloneable
    interface javax.servlet.sip.Address
    interface javax.servlet.sip.URI
        interface javax.servlet.sip.SipURI
        interface javax.servlet.sip.TelURL
interface java.util.EventListener
    interface javax.servlet.sip.SipApplicationSessionListener
    interface javax.servlet.sip.SipErrorListener
    interface javax.servlet.sip.SipSessionActivationListener
    interface javax.servlet.sip.SipSessionAttributeListener
    interface javax.servlet.sip.SipSessionBindingListener
    interface javax.servlet.sip.SipSessionListener
    interface javax.servlet.sip.TimerListener
interface javax.servlet.sip.Proxy
interface java.io.Serializable
    interface javax.servlet.sip.SipSession
interface javax.servlet.ServletRequest
```

---

```
interface javax.servlet.sip.SipServletRequest (同时扩展 javax.servlet.sip.SipServletMessage)
interface javax.servlet.ServletResponse
interface javax.servlet.sip.SipServletResponse (同时扩展 javax.servlet.sip.SipServletMessage)
interface javax.servlet.sip.ServletTimer
interface javax.servlet.sip.SipApplicationSession
interface javax.servlet.sip.SipFactory
interface javax.servlet.sip.SipServletMessage
interface javax.servlet.sip.SipServletRequest (同时扩展 javax.servlet.ServletRequest)
interface javax.servlet.sip.SipServletResponse (同时扩展 javax.servlet.ServletResponse)
interface javax.servlet.sip.TimerService
```

## 附录 D jsip 类与接口

### 1. 类结构图

```
class java.lang.Object
class java.util.AbstractCollection (实现 java.util.Collection)
class java.util.AbstractList (实现 java.util.List)
class java.util.ArrayList (实现 java.lang.Cloneable, java.util.List, java.io.Serializable)
class org.mitre.jsip.SipUriList
class org.mitre.jsip.SipViaList
class org.mitre.jsip.util.Base64
class java.awt.Component (实现 java.awt.image.ImageObserver, java.awt.MenuContainer,
java.io.Serializable)
class java.awt.Container
class java.awt.Window (实现 javax.accessibility.Accessible)
class java.awt.Frame (实现 java.awt.MenuContainer)
class javax.swing.JFrame (实现 javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants)
class org.mitre.jsip.SipTester (实现 java.lang.Runnable)
class org.mitre.jsip.util.DigestAuthentication
class org.mitre.jsip.proxy.DNSProxyLookup (实现 org.mitre.jsip.proxy.ProxyLookup)
class java.util.EventObject (实现 java.io.Serializable)
class java.awt.AWTEvent
class org.mitre.jsip.event.CallEvent
class org.mitre.jsip.proxy.FileProxyLookup (实现 org.mitre.jsip.proxy.ProxyLookup)
class org.mitre.jsip.GeneralUri
class org.mitre.jsip.SipUri (实现 java.io.Serializable)
class org.mitre.jsip.HeaderParser
class org.mitre.jsip.MessageSocket
class org.mitre.jsip.MulticastMessageSocket
class org.mitre.jsip.TCPMessageSocket
class org.mitre.jsip.UDPMessageSocket
class org.mitre.jsip.MessageSocketFactory
class org.mitre.jsip.MimeContentType
class org.mitre.jsip.util.Poller
class org.mitre.jsip.proxy.ProxyServer
```

```
class org.mitre.jsip.SdpMessage
class org.mitre.jsip.Sip
class org.mitre.jsip.SipCall
class org.mitre.jsip.SipCallMember
class org.mitre.jsip.SipHeader
class org.mitre.jsip.SipMessage
class org.mitre.jsip.SipRequestMessage
class org.mitre.jsip.messages.AckMessage
class org.mitre.jsip.messages.ByeMessage
class org.mitre.jsip.messages.CancelMessage
class org.mitre.jsip.messages.InviteMessage
class org.mitre.jsip.messages.MessageMessage
class org.mitre.jsip.messages.NotifyMessage
class org.mitre.jsip.messages.OptionsMessage
class org.mitre.jsip.messages.RegisterMessage
class org.mitre.jsip.messages.SubscribeMessage
class org.mitre.jsip.messages.UnsupportedMessage
class org.mitre.jsip.SipResponseMessage
class org.mitre.jsip.SipRegister (实现 org.mitre.jsip.util.PollTask)
class org.mitre.jsip.SipStatus
class org.mitre.jsip.SipTransaction
class org.mitre.jsip.SipUser
class org.mitre.jsip.SipUtil
class org.mitre.jsip.SipVia
class JACE.ASX.Task (实现 JACE.Reactor.EventHandler, java.lang.Runnable)
class JACE.tests.ASX.ProducerConsumer
class org.mitre.jsip.event.CallEventMulticaster (实现 org.mitre.jsip.event.CallListener)
class java.lang.Thread (实现 java.lang.Runnable)
class org.mitre.jsip.MessageThread
class org.mitre.jsip.SipClient
class java.lang.Throwable (实现 java.io.Serializable)
class java.lang.Exception
class org.mitre.jsip.BadHeaderException
class org.mitre.jsip.util.DigestAuthenticationException
class org.mitre.jsip.RegisterException
class org.mitre.jsip.proxy.SipProxyException
class java.util.TimerTask (实现 java.lang.Runnable)
class org.mitre.jsip.util.PollAgent
```

## 2. 接口结构图

`interface java.util.EventListener`

`interface org.mitre.jsip.event.CallListener`

`interface org.mitre.jsip.util.PollTask`

`interface org.mitre.jsip.proxy.ProxyLookup`

## 附录 E SIP RFC 与草案

### 1. SIP RFC

SIP RFC 如附表 5-1 所示。

附表 5-1 SIP RFC

RFC 序号	RFC 标题	备 注
RFC 3725	Best Current Practices for Third Party Call Control in SIP	
RFC 3680	A Session Initiation Protocol (SIP) Event Package for Registrations	
RFC 3666	PSTN SIP Call Flow Examples	
RFC 3665	Basic SIP Call Flow Examples	
RFC 3581	An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing	
RFC 3524	Mapping of Media Streams to Resource Reservation Flows	单个流的预留 (SRF)
RFC 3515	The Session Initiation Protocol (SIP) Refer Method	Refer 方法
RFC 3487	Requirements for Resource Priority Mechanisms for the Session Initiation Protocol (SIP)	
RFC 3486	Compressing the Session Initiation Protocol (SIP)	
RFC 3485	The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp)	
RFC 3455	Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)	
RFC 3428	Session Initiation Protocol (SIP) Extension for Instant Messaging	
RFC 3427	Change Process for the Session Initiation Protocol (SIP)	
RFC 3420	Internet Media Type message/sipfrag	
RFC 3407	Session Description Protocol (SDP) Simple Capability Declaration	
RFC 3398	Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping	
RFC 3388	Grouping of Media Lines in Session Description Protocol (SDP)	
RFC 3371	Session Initiation Protocol for Telephones (SIP-T): Context and Architectures	
RFC 3361	Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers	
RFC 3351	User Requirements for SIP in Support of Deaf, Hard of Hearing, and Speech Impaired Individuals	
RFC 3319	Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers	
RFC 3266	Support for IPv6 in Session Description Protocol (SDP)	
RFC 3361	Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers	
RFC 3327	Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts	路径头

续表

RFC 序列号	RFC 标题	备 注
RFC 3326	The Reason Header Field for the Session Initiation Protocol (SIP)	原因
RFC 3325	Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks	P-Asserted-Identity 和 P-Preferred-Identity
RFC 3324	Short Term Requirements for Network Asserted Identity	
RFC 3323	A Privacy Mechanism for the Session Initiation Protocol (SIP)	Privacy 头
RFC 3329	Security Mechanism Agreement for the Session Initiation Protocol (SIP)	
RFC 3321	Signaling Compression (SigComp) – Extended Operations	
RFC 3322	Signaling Compression (SigComp) Requirements and Assumptions	
RFC 3320	Signaling Compression (SigComp)	
RFC 3313	Private Session Initiation Protocol (SIP) Extensions for Media Authorization	
RFC 3312	Integration of Resource Management and SIP	Preconditions 框架
RFC 3311	The Session Initiation Protocol (SIP) UPDATE Method	
RFC 3261 (Bookmarks kindly provided by Alexandre Gilles)	SIP: Session Initiation Protocol	
RFC 3262	Reliability of Provisional Responses in the Session Initiation Protocol (SIP)	PRACK 方法
RFC 3263	Session Initiation Protocol (SIP): Locating SIP Servers	
RFC 3264	An Offer/Answer Model with the Session Description Protocol (SDP)	会话协商
RFC 3265	Session Initiation Protocol (SIP)-Specific Event Notification	SUBSCRIBE 和 NOTIFY
RFC 3204	MIME Media Types for ISUP and QSIG Objects	
RFC 3087	Control of Service Context using SIP Request-URI	
RFC 3050	Common Gateway Interface for SIP	
RFC 2976	The SIP INFO Method	
RFC 2848	The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services	
RFC 2327	The Session Description Protocol (SDP)	

## 2. 草案

SIP Forked Media

A SIP Extension: Informational Responses to the REFER method

Models for Multi Party Conferencing in SIP

SIP Requirements for support of Multimedia and Video

Third Party Call Control for Resource Managemen

Using SIP for Peer-to-Peer Third-Party Call Control

SIP Telephony Service Examples

SIP Extensions for supporting Distributed Call State

SIP Record-Route/Route Hiding

Diversion Indication in SIP

SIP extension for tracking locations attempted

SIP for the establishment of xcast-based multiparty conferences



SIP Enabled Services to Support the Hearing Impaired  
SCTP as a Transport for SIP  
Guidelines for Authors of SIP Extensions  
Framework for SIP Call Control Extensions  
Third party call control with SDP preconditions  
Mandating SIP Extension Support by Servers  
Control of Service Context using SIP Request-URI  
SIP Telephony Call Flow Examples  
SIP Call Control: Transfer  
SIP 183 Session Progress Message  
Distributed Multipoint Conferences using SIP  
Reliability of Provisional Responses in SIP  
Providing for Multiple-Proxy Authentication of a SIP Request  
SIP Session Timer  
The SIP PROPOSE Method  
Protocol Complications with the IP Network Address Translator  
User Agent Configuration  
SIP SNMP MIB  
Supported Header  
Service Enablers  
Caller Preferences  
PSTN Interconnection  
Quality of Service  
Presence, Events and Instant Messaging  
Mobile Hosts  
Alignment with the PacketCable DCS specifications;  
Exploration of applications to SIP in mobile environments;  
Interworking of SIP state machines and IN call models;  
Interworking with the MLPP multi-level priority specifications of military ISUP;  
Integration of QoS, policy, and signaling;  
Relationship between SIP, QoS and OSP as well as AAA architecture;  
SIP for selecting MPLS routes.  
SIP extensions for device control;  
Using DHCP and SLP to configure SIP servers and clients  
Distributed Call Signaling, a set of PacketCable proposals;  
Mobility-related;  
SIP APIs such as sip-cgi and servlets;  
Caller preferences allow callers to guide the behavior of proxies and user agents;  
QoS negotiation  
H.323 interworking;  
Security (authentication, encryption) issues  
SIPstone: benchmarking SIP servers

An XML Representation for Expression of Presence Policy Capabilities  
 An XML Representation for Expression Policy Capabilities  
 Extensions to PIDF for Conveying Phone State  
 An XML Configuration Access Protocol  
 A SIP Event Package for Modification Events in XCAP  
 An XML Document Format for Representing Resource Lists  
 Presence Authorization Rules  
 Requirements for Manipulation of Data Elements in SIMPLE  
 Advanced Instant Messaging Requirements for the Session Initiation Protocol  
 A Component Model for SIMPLE  
 A SIP Event Notification Extension for Resource Lists (formerly Buddy List Presence)  
 An SIP Event Sub-Package for Watcher Information  
 An XML Based Format for Watcher Information  
 SIP Instant Message Sessions  
 SIP for Presence

### 3. 3GPP/3GPP2 关于 SIP 及 IMS 的标准

3GPP/3GPP2 关于 SIP 及 IMS 的标准如附表 5-2 所示。

附表 5-2 3GPP/3GPP2 关于 SIP 及 IMS 的标准

TIA(3GPP2)规范	等效的 3GPP R5 规范	标题
S.R0037		IP Network Architecture Model for cdma2000 Spread Spectrum Systems
TIA-872.000 (X.S0013.000)		IP Multimedia Subsystem-Overview
TIA-872.002 (X.S0013.002)	23.228 V5.	IP Multimedia Subsystem-Stage-2
TIA-872.003 (X.S0013.003)	23.218 V5.	IP Multimedia Subsystem-IP Multimedia Call Model Stage-2
TIA-872.005 (X.S0013.005)	29.228 V5.	IP Multimedia Subsystem Cx Interface. Signaling Flows and Message Contents
TIA-872.007 (X.S0013.007)	32.200 V5.	IP Multimedia Subsystem-Charging Architecture
TIA-872.010 (X.S0013.010)	29.238 V5.	IP Multimedia Subsystem Sh Interface;Signaling Flows and Message Contents-Stage-2
TIA-872.004 (X.S0013.004)	24.229 V5.	IP Multimedia Call Control Protocol-Based on SIP and SDP Stage-3
TIA-872.006 (X.S0013.006)	29.229 V5.	Cx Interface Based on the Diameter Protocols; Protocol Details
TIA-87.2008 (X.S0013.008)	32.225 V5.	IP Multimedia Subsystem-Accounting Information Flows and protocol
TIA-872.001 (X.S0013.001)	29.239 V5.	Sh Interface Based on the Diameter Protocol Details-Stage-3
S.P0086-0	33.203 V5.	Multimedia Domain Security Framework

## 参 考 文 献

- 1 IETF Internet Protocol, RFC 2543, SIP: Session Initiation Protocol, 1999
- 2 IETF Internet Protocol, RFC 3372—2002, Session Initiation Protocol for Telephones (SIP-T): Context and Architectures
- 3 SUN, SIP Servlets 1.0
- 4 SUN, JAIN SIP Lite
- 5 3GPP TS 22.127: “Service Requirement for the Open Service Access (OSA); Stage 1” .
- 6 3GPP TS 23.228: “IP Multimedia Subsystem (IMS); Stage 2” .
- 7 3GPP TS 24.229: “IP Multimedia Call Control Protocol based on SIP and SDP; Stage 3” .
- 8 3GPP TS 24.228: “Signalling flows for the IP multimedia call control based on SIP and SDP; Stage 3” .
- 9 3GPP TS 29.328: “IP Multimedia Subsystem (IMS) Sh Interface signalling flows and message contents” .
- 10 IETF Third party call control draft: “Third Party Call Control in SIP”(draft-rosenberg-sip-3pcc-03.txt).
- 11 Object Management Group. The Common Object Request Broker: Architecture and Specification. Revision 2.0, July 1995.
- 12 Parlay X White Paper, Version 1.0: <http://parlay.org/specs/library/index.asp>.
- 13 Web Services Description Language (WSDL) 1.1.
- 14 RFC 3455 “Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)”.
- 15 3GPP TS 23.228 “IP Multimedia Subsystem (IMS), Stage 2”.
- 16 3GPP TS 24.229 “IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3”.
- 17 San Ramon. International Softswitch Consortium: Reference Architecture, <http://www.softswitch.org/>, June 2002
- 18 Next Generation Networks: the service offering standpoint, Eurescom P1109, Nov, 2001.
- 19 Handley, M. and V. Jacobson, “SDP: Session Description Protocol” ,
- 20 Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997.
- 21 Rosenberg, J. and H. Schulzrinne, “An Offer/Answer Model with Session Description Protocol (SDP)” , RFC 3264, June 2002.
- 22 Rosenberg, J., “The Session Initiation Protocol (SIP) UPDATE Method,” RFC 3311, September 2002.
- 23 Schulzrinne, S., Casner, S., Frederick, R. and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications” , RFC 1889, January 1996.
- 24 Rosenberg, J. and H. Schulzrinne, “Reliability of Provisional Responses in Session Initiation Protocol (SIP)” , RFC 3262, June 2002.
- 25 C. Kalmanek, W. Marshall, P. Mishra, D. Nortz, and K. K. Ramakrishnan, “DOSAs: an architecture for providing robust IP telephony service,” in Proceedings of the Conference on Computer Communications (IEEE Infocom), (Tel Aviv, Israel), Mar. 2000.

Images have been losslessly embedded. Information about the original file can be found in PDF attachments. Some stats (more in the PDF attachments):

```
{
  "filename": "XzExMzQwMjY0LnppcA==",
  "filename_decoded": "_11340264.zip",
  "filesize": 22977136,
  "md5": "ecf39dcae2c9dbdacf755992e52197f8",
  "header_md5": "773073e6381269abb26ab69e2650fed3",
  "sha1": "8ae9a66ef792f3ad541ba6403079ed81b665d6d2",
  "sha256": "0bd7d081a580e671630b532b8099dfbe0403ea94f35c2d95102cb55baed16f42",
  "crc32": 473354824,
  "zip_password": "wcpfxk&^TDwcpfxk",
  "uncompressed_size": 23813854,
  "pdg_dir_name": "",
  "pdg_main_pages_found": 240,
  "pdg_main_pages_max": 240,
  "total_pages": 255,
  "total_pixels": 1652280960,
  "pdf_generation_missing_pages": false
}
```