

基于 OSGi 的 IMS 客户端中 SIP Agent 的设计与实现^①

陈 亮 朱晓民 (北京邮电大学 网络与交换技术国家重点实验室 北京 100876;
东信北邮信息技术有限公司 北京 100191)

摘 要: 提出了基于 OSGi(Open Services Gateway Initiative)的满足 IMS(IP Multimedia Subsystem)规范要求的 SIP Agent 的设计与实现。该 SIP Agent 使用实现 OSGi 规范的 Equinox 为核心框架,并结合 JAIN-SIP 作为底层协议栈,能够符合绝大部分的 IMS 应用。主要包括逻辑结构说明,主要模块的结构和关键技术设计,基于 SIP Agent 搭建呼叫信令流程来论证其可用性。基于论文所提出的 SIP Agent 构建的应用程序具有低耦合、高内聚、易扩展、高性能的特性,能够方便快捷地构建 SIP 相关的应用。

关键词: OSGi; IMS; SIP

Design and Implementation of SIP Agent in IMS Client Based on OSGi

CHEN Liang, ZHU Xiao-Min

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; EB Information Technology Co. Ltd., Beijing 100191, China)

Abstract: This paper proposes the design and implementation of Sip Agent, which meet IMS(IP Multimedia Subsystem) specifications based on OSGi(Open Services Gateway Initiative). It uses the Equinox, which implements OSGi specification as the core framework, combined with JAIN-SIP, as the underlying protocol stack, and can meet most of the IMS application. This paper includes the explanation of the logical structure, the main module's structure, the introduction of key technology, and the development of a call procedure using Sip Agent, to prove its availability. The applications, which is built on Sip Agent presented in this paper, has low coupling, high cohesion, easily extensible, high-performance features, and it can build SIP-related application quickly and easily.

Keywords: OSGi; IMS; SIP

1 引言

随着移动通信业的发展以及计算机网络的成熟,传统的 IP 网络已经无法满足人们日益增长的多媒体业务需求,而 IMS^[1,2] (IP Multimedia Subsystem) 网

络无疑是目前堪称完美的解决方案;它几乎得到了国内外运营商的一致认可,其日渐火爆的根本原因在于 IMS 顺应了运营商和最终用户的需求。当前的 IMS 技术工作主要集中在探讨 IMS 网络上,而忽视了对 IMS

^① 基金项目:国家杰出青年科学基金(60525110);国家重点基础研究发展规划(973)(2007CB307103);国家自然科学基金(60902051);中央高校基本科研业务费资助(BUPT2009RC0505);电子信息产业发展基金项目
收稿时间:2009-12-27;收到修改稿时间:2010-03-12

客户端的研究,然而,IMS 客户端才是最终用户享受 IMS 技术带来的诸多成果的最直接的表现方式。只有成熟、稳定的 IMS 客户端,才能使用户完全体验下一代网络所带来的好处。

SIP^[3,4] (Session Initiation Protocol) 协议作为 IMS 网络体系结构的一个永久单元,用于建立,修改和终止包括视频,语音,即时通信,在线游戏和虚拟现实等多种多媒体元素在内的交互式用户会话。它在 IMS 网络中作为控制(信令)协议,其简单,灵活,高可靠的特性使之足以承担呼叫控制的责任,并配合实现了控制/业务与承载相分离的思想。

IMS 客户端对 SIP 协议的理解、实现和应用对其在 IMS 核心网中的使用起着至关重要的作用,本文所设计并实现的 Sip Agent 就是致力于简化在 IMS 客户端中开发基于 SIP 的应用。

2 相关背景

基于 OSGi^[5,6] (Open Services Gateway Initiative)的 IMS 客户端属于东信北邮信息技术有限公司(以下简称 EB)3G 产品系列,是 EB 独立研发的具有自主知识产权的产品。它作为面向 IMS 和 NGN 网络的接入侧通信客户端软件,提供在通信网络接入侧所需的各种重要能力,使用户能够通过该软件享受 IMS 提供的多种服务。

作为 IMS 网络中的软终端,除了为用户提供基本的 VOIP 功能和 IM 功能,还为用户提供通过 IMS 网络获得基本的服务,如 presence 服务、group 服务、多媒体服务(彩铃彩像服务)等等。同时,还为用户提供多种业务定制功能,如个人状态信息的管理和发布、好友状态信息的订阅、个人信息管理、联系人列表管理、群组管理、早期媒体的接收和显示等。

3 设计与实现

3.1 Sip Agent 总体设计

Sip Agent 的搭建平台是基于 OSGi 的,其模块均以插件的形式存在于 IMS 客户端的架构中,这就使得 Sip Agent 具有 OSGi 所带来的灵活,动态,高效,鲁棒等特性,使得 Sip Agent 具有很好的可定制性,松散的耦合度。Sip Agent 的整体逻辑结构如图 1 所示,主要分为两层:

(1) SIP Stack Adapter 层: 该层将协议栈的接口

进行适配封装,负责信令的发送和接收;同时开辟一个线程池来提升信令收发的性能,当接收到消息后上报事件给上层^[7]。

(2) SIP Agent Service 层: 该层抽象了 SIP 协议中常用的请求和响应,将每类请求/响应及相关事务封装成对象,按协议进行消息自动处理。

① CoreService 模块: 负责对外的所有接口,上层只有通过调用 CoreService 的方法才能创建需要的信令对象;同时 CoreService 模块还负责统一保存和管理信令的事务状态。

②Session 模块: 负责发起呼叫、终止呼叫等的基本呼叫流程的信令处理,根据相关的 IETF 规范,内部维持一个状态机,自动处理相关的信令交互。同时基于 OSGi 良好的扩展性允许上层程序加入定制的处理逻辑。

③Reference 模块: 负责所有关于 REFER 请求相关的事务处理。

④Publication 模块: 负责所有关于 PUBLISH 请求相关的事务处理。

⑤Subscription 模块: 负责所有关于 SUBSCRIBE 和 NOTIFY 相关的事务处理,支持 IETF RFC3265 规范。

⑥PageMessage 模块: 负责所有关于 MESSAGE 消息相关的事务处理,通过该模块能够实现消息的及时交互。

⑦EventFramework 模块: 负责提供事件通知机制。

⑧Registration/Auth 模块: 负责注册接入 IMS 核心网相关的事务处理,支持可配置主动刷新注册及被动鉴权认证,鉴权认证支持 IMS-AKA 鉴权^[8]和 HTTP Digest 鉴权方式^[9]。

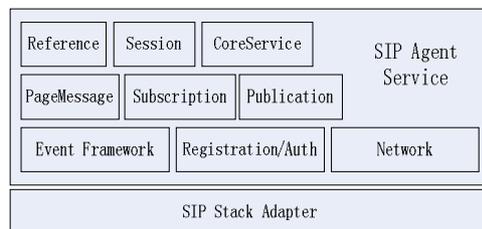


图 1 Sip Agent 逻辑结构图

Sip Agent 的静态结构图如图 2 所示,所有封装的信令结构均继承自 ServiceMethod 接口的实现类,

该类负责提供状态机、操作即将发送的请求消息、缓存之前发送过的请求和响应消息、自定义设置继承该类的消息的头部信息及消息体。每一类信令均有对应的监听器实现信令间事务交互，并能通知上层触发相关操作。**CoreService** 是一个独立的对外接口，该接口的实现类采用简单工厂模式来创建需要的信令模型。

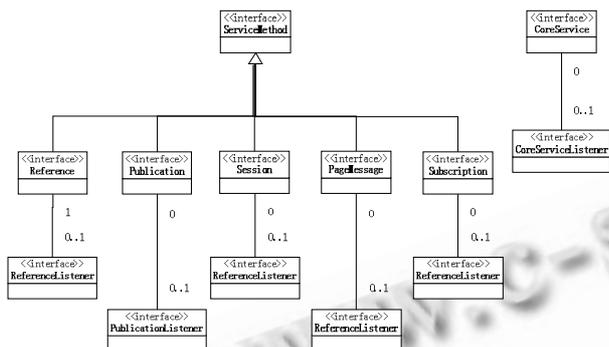


图 2 静态结构图

3.2 关键技术

3.2.1 事件框架的设计

为了减少模块之间的相互依赖，**Sip Agent** 设计了专门的事件框架来实现上下层之间的通信。此事件框架采用观察者模式 (**OBSERVER**) 来实现事件通知机制。**CoreService** 作为底层协议栈适配层的观察者，监听消息的接收，同时 **CoreService** 本身将采用装饰者模式 (**FACADE**) 来简化各种信令的创建，处理等操作。

事件框架可以很好地解耦，使得上下层模块之间不会存在相互的调用的现象。另一方面，采用事件通知机制使得程序能够异步地进行操作，能够使得整个框架的性能有一定的保障。

3.2.2 分发路由的设计

由于信令的收发具有突发性，这就需要 **Sip Agent** 具有较高的性能来处理网络数据。图 3 显示路由分发的总体设计，总体思路是采用消息驱动，以异步的方式接收和处理消息，并从线程池中分配可用的线程来分发消息。正是由于采用消息驱动的方式，使得消息的路由分发不会被堵塞，同时加上多线程处理使得消息分发模块的效率保持一个比较高的水准。

该设计中之所以采用线程池技术，主要是考虑到突发的网络消息到来的速度很快，如果对每个网络消

息都创建单独的线程去处理，开销太大；如果不使用单独线程去处理网络消息，整个消息处理都会处于阻塞的状态，当一个消息未处理完就无法处理新消息。综合考虑之下，采取了线程池的方式来处理。系统启动时，在线程池内分配好实例，当从网络中接收到消息后，就将该消息委托给线程池的实例去处理，这样既可以减少频繁创建线程的开销，又可以提高网络数据的处理速度。

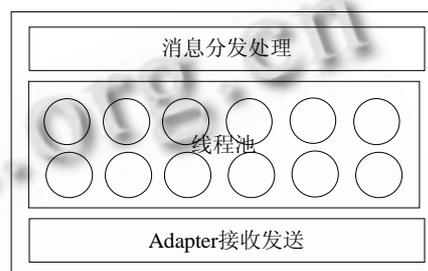


图 3 消息分发总体结构图

消息分发处理将 **SIP** 消息进行解析，根据相关关键字段信息将消息分发到对应的处理模块中去，同时对消息进行缓存，对其事务进行管理，以便进行后续的逻辑处理。

3.3 应用实例

下面以基本呼叫为例，详细介绍客户端如何使用 **Sip Agent** 快速地构建呼叫连接，挂断及媒体交互协商。具体流程图如图 4 所示：

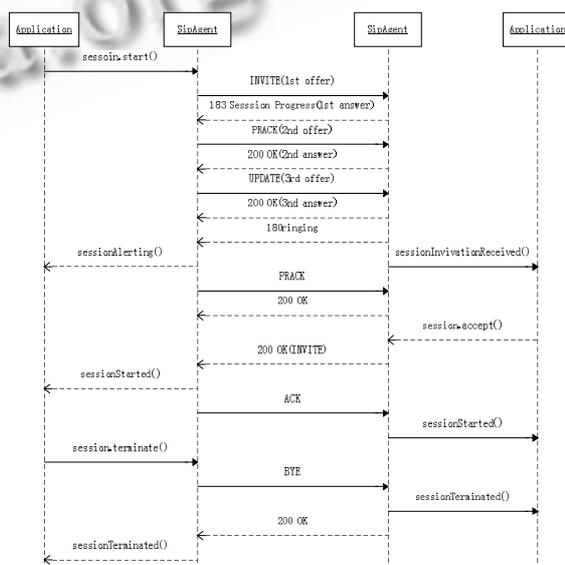


图 4 典型呼叫流程

(1) 建立呼叫连接:

上层应用程序调用 `session.start()` 方法来发起呼叫, **Sip Agent** 引擎将会根据初始配置或用户设置的媒体交互信息构建 **SDP** 消息, 同时发送 **INVITE** 请求到远端。**SDP** 消息是用来描述此次会话过程中双方支持的媒体能力信息。远端的 **Sip Agent** 通过协议栈及适配层所收到呼叫请求进行信令协商, 交换媒体信息^[10], 确定双方的媒体能力, 这部分均由 **Sip Agent** 根据协议进行自动交互协商, 简化了开发流程; 当会话协商完成之后, 远端的 **Sip Agent** 将会通过注册的监听器触发 `sessionInvitationReceived()` 事件, 同时发起方将会收到 **180Ringing** 响应, 并触发 `sessionAlerting()` 事件通知发起方远端已振铃, 呼叫协商完成; 远端用户摘机, 只需调用 `session.accept()` 方法来完成呼叫建立, 同时发起方收到 **200OK** 之后 **Sip Agent** 将会上报 `sessionStart()` 方法来完成呼叫建立。

(2) 结束呼叫连接:

上层应用程序可以调用 `session.terminate()` 方法来终止呼叫, 此时 **Sip Agent** 引擎将会发送一个 **BYE/CANCEL** 请求至远端来请求终止会话。在本例中, 由于呼叫已经建立, **Sip Agent** 根据状态机当前的状态自动构建 **BYE** 请求来终止此时会话, 远端收到后立即回复 **200OK**, 同时通过 `sessionTerminated()` 上报通知对端请求结束会话; 主动挂断方在收到远端返回的 **200OK** 后也会通过 `sessionTerminated()` 上报通知会话已经结束。

4 总结

近来, **IMS** 网络受到越来越多的国内外运营商的追捧, 不少国家的 **IMS** 网络运营已经处于商用阶段, 我国的 **IMS** 网络也已经处于测试阶段尾声, 接近商用。**IMS** 客户端的研发已经逐渐成熟, 本文所设计并实现的支持 **IMS** 规范的 **Sip Agent**, 依托于 **OSGi** 框架,

并抽象出常用的信令事务处理, 满足基本的 **SIP** 应用和绝大部分的 **SIP** 扩展应用, 具有高效, 稳定, 灵活和易扩展的特性。

参考文献

- 1 Mikka Poikselka, Georg Mayer, Hisham Khartabil 著, 赵鹏, 周胜, 望玉梅译, **IMS: 移动领域的 IP 多媒体概念和服务**. 北京: 机械工业出版社, 2005.
- 2 胡乐明, 曹磊, 陈洁. **IMS 技术原理及应用**. 北京: 电子工业出版社, 2006.
- 3 张智江, 张云勇, 刘韵洁著. **SIP 协议及其应用**. 北京: 电子工业出版社, 2005.
- 4 Rosenberg J, Schulzrinne H, Camarillo G. IETF. RFC 3261, **SIP: Session Initiation Protocol**, Jun 2002.
- 5 OSGi Alliance. **OSGi Service Platform Core Specification Release [2009-6-13]**. <http://www.osgi.org>.
- 6 OSGi Alliance. **OSGi Service Platform Service Compendium Release[2009-6-13]**. <http://www.osgi.org>.
- 7 Cao YF, Liao JX, Qi Q, Zhu XM. A cache based session setup mechanism for IMS, **IEEE International Conference on Communications Work shops**, 19-23 May 2008. 261 - 265.
- 8 Niemi A, Arkko J, Torvinen V. IETF, RFC 3310, **Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)**, September 2002.
- 9 Franks J, Hallam-Baker P, Hostetler J. RFC2617. **HTTP Authentication: Basic and Digest Access Authentication**, June 1999.
- 10 Rosenberg J, Schulzrinne H. IETF. RFC3264. **An Offer/Answer Model with the Session Description Protocol(SDP)**, June 2002.