

统一框架 Doubango 下的 IMS 客户端元素^①

姜懿恒^{1,2}, 孙 勇¹, 温向明¹

¹(北京邮电大学 网络构建与融合北京市重点实验室, 北京 100876)

²(北京邮电大学 研究生院, 北京 100876)

摘要: IMS 核心网平台的软件实现现在基本以 Open IMS Core 为主. 当今大部分 IMS 客户端各有所长, 但互通性较差. Doubango 的客户端元素在 Doubango 统一框架下构建, 无线客户端和固网客户端具有同样的交换信息算法并能良好互通. 阐述了当今主要的开源 IMS 客户端并进行互通测试, 并分析 Doubango 客户端元素的软件架构, 最后在其客户端基础上添加新应用功能. 本文描述的客户端架构能够作为 IMS 客户端扩展的重要基础, 具有高性能的应用功能, 可应用于不同的操作系统, 从而可扩大适用 IMS 技术的移动终端的范围.

关键词: IMS 技术; Doubango 框架; 客户端元素; 互通性能; 功能扩展

IMS Client-Side Components on the Unified Framework Doubango

JIANG Yi-Heng^{1,2}, SUN Yong¹, WEN Xiang-Ming¹

¹(Beijing Municipal Key Laboratory of the Construction and Fusion on Network, Beijing University of Posts and Telecommunications, Beijing 100876, China)

²(Graduate University, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: It is mainly Open IMS Core in the software implementation of platform for the IMS core network currently. Though most of the existing IMS clients have their own specialty, they poorly communicate between each other. The client-side components of Doubango are based on the unified framework of Doubango, wireless clients and fixed clients of which possess the same algorithm of exchange information and have good interoperability. This paper describes current main IMS clients of open source and carries out interoperability tests, then analyses the software architecture of Doubango's client-side components, finally add a new function to one of components. The client architectures presented in this paper can be important foundation of extending the IMS clients, making which take high-performance applications on, and can be applied in different operation system to expand the serviceable range of IMS clients.

Key words: IMS technology; Doubango framework; client-side component; interoperability; extensions of functions

1 引言

3GPP(Third Generation Partnership Project)在 R4 中已开始定义全 IP 网络, R5 最终将 IMS 引入 3GPP 标准. IMS 是一个独立于接入技术的基于 IP 的标准体系, 可以通过不同接入方式的客户端建立起对等的 IP 通信, 并可以获得所需的服务质量^[1]. IMS 体系除了会话管理外, 还为完成服务提供所必需的功能, 如注册、安全、计费、漫游、承载控制等.

The Open IMS Core^[2] project 由 Fraunhofer 研究所

FOKUS 创建. Open IMS Core 由 CSCFs(呼叫会话控制功能), HSS(归属用户服务器)构成. CSCFs 是 SER(SIP 快速路由)的扩展, 包含 PCSCF(代理 CSCF)、ICSCF(询问 CSCF)和 SCSCF(服务 CSCF), 实现 IMS 信号层的中心路由功能. HSS 由 FhoSS(FOKUS HSS)实现, 管理用户文件和相关路由规则等 IMS 需要在 HSS 查找的信息.

IMS 客户端则要实现与上述的 IMS 核心网的交互功能. 本文将描述对现在主流 IMS 开源客户端的早期研究, 第三部分说明这些不同团队开发的客户端在互

① 基金项目:中央高校基本科研业务费专项资金(2011RC0107);国家自然科学基金(61101106)

收稿时间:2012-02-23;收到修改稿时间:2012-03-22

相通信时的不足,第四部分分析 Doubango 各客户端元素的结构,第五部分以 Boghe 客户端为例,在其基础上添加新应用功能。

2 早期研究

本部分将描述一些现在公认表现不错的 IMS 客户端的软件。

2.1 Mercurio IMS Client^[3]

该客户端基于 3GPP 的主要 IMS 规范实现,同时也包含了部分 RCS(富通信套件)阶段 1 的标准和 OMA(开放移动联盟)的 IMS 移动服务规范. Mercurio 是现在主流的完全 3GPP IMS 软电话客户端之一,其特点有:支持 SIP/SIGCOMP 协议,适用 IPv4 和 IPv6 网络;支持 UDP, TCP, TLS 和 IPSec 传输安全协议;以 Digest MD5, AKA-v1 和 AKA-v2 算法进行 IMS 安全注册,支持 PAKC(临时响应确认)及 3GPP 的私有扩展包头等. 尽管 Mercurio 在相关特点和标准是实现得较为彻底的 IMS 客户端,但它是闭源的,开发者不能在其基础上扩展代码添加新服务,不利于 IMS 客户端的扩展开发。

2.2 IMS Communicator^[4]

该客户端基于 Sip Communicator java 版本项目的 SIP 客户端,在 JAIN-SIP 栈和 JMF(java 媒体框架)API 的上层实现. 为了支持 IMS, JAIN-SIP 栈做了一些改变,比如新的 SIP 包头和 SDP 功能,这些改变已经捆绑在最新的 JAIN-SIP 栈里。

由 Sip Communicator 到 IMS Communicator 所做的开发主要集中在 IMS 注册和鉴权,和 IMS 会话建立. 关于 IMS 注册和鉴权, IMS Communicator 实现了 IMPI(IP 多媒体私有账户)支持,鉴权算法 AKAv1, 订阅“reg”事件及 Security Agreement 机制;关于 IMS 会话的建立,实现了 Precondition 机制,也添加了 Early Media 和 Call Transfer 功能。

2.3 UCT IMS Client^[5]

该客户端是由 University of Cape Town 开发的基于 Linux 的开源 3GPP 客户端,开始于 2006 年,到现在该客户端在稳定和功能方面都有了很大的改进. UCT IMS Client 使用 IETF 协议实现各种 IMS 功能,如 SIP(会话初始协议)、RTP(实时传输协议)、XCAP(XML 配置访问协议)、MSRP(消息会话中继协议)及 RTSP(实时流协议). 其使用几个已有的上述协议的库,包含

oSIP、eXosip、libcurl、libXML2、gStreamer、libVLC 和 libMSRP. 所有库除了 libMSRP 都是在运行时间动态链接的,因此这些库应存在于运行主机上以使客户端成功运行. libMSRP 库在主流 Linux 操作版本不包含,作为一部分包含在客户端源码中。

上述客户端的特点概括如表 1 所示。

表 1 上述客户端特点概括表

	Mercurio IMS Client	IMS Communicator	UCT IMS Client
软件许可	免费, 闭源	LGLP	GPLv3
适用平台	Windows	Windows/Linux	Linux
注册算法	AKAv1/2-MD5, MD5	AKAv1-MD5, MD5	AKAv1/2-MD5, MD5
信号层	支持 PRACK, 支持 Precondition	支持 PRACK, 支持 Precondition	支持 PRACK, 支持 Precondition
即时短信	支持 Pager mode, 支持 Session-based	不支持	支持 Pager mode, 支持 Session-based
媒体支持	语音	语音/视频	语音/视频
呈现支持	支持呈现, 支持观察者认证	不支持	支持呈现, 支持观察者认证
XCAP 支持	支持	支持	支持

3 互通测试及分析

相同客户端间的功能实现效果良好,但如果通信放在不同客户端之间效果会怎样. 互通测试的目的就是验证不同客户端间的功能通用性是否符合要求. 以 UCT IMS Client 作为测试参照客户端,用它与其他两种客户端进行通信测试,测试包含会话初始、即时短信(IM)、媒体会话(语音/视频)和呈现功能(presence)。

在会话初始测试中,基本的 SIP 信号层信息的互通是成功的,但一旦支持 PRACK 和 precondition,客户端之间不能良好地进行初始会话,尤其是在 precondition 下,各客户端不能正确交换 SIP 信息. 会话初始测试用例如图 1 所示。

在即时短信测试中,基于 page mode 的即时短信的实现标准是统一的,所以支持该模式的客户端互通成功,基于会话的即时短信有不同的实现标准,各客户端因采用不同标准编程而不能实现基于会话的即时短信互通. 即时短信测试用例如图 2 所示。

媒体会话测试包含语音测试和视频测试. UCT IMS Client 与其他两种客户端的语音测试结果良好. 视频测试中,UCT IMS Client 与其他客户端均不能视

频通话,可以看出各客户端支持的视频编解码范围是有限并不相容的.媒体会话测试用例如图 3.

呈现功能是许多 IMS 应用的基础,IMS 客户端既是呈现者又是观察者.基于 SIP 的呈现功能是相对简单的 IMS 应用,在呈现功能测试中,只要两方客户端支持呈现功能,基本的订阅状态和发布状态功能能够实现.呈现功能测试用例如图 4 所示.

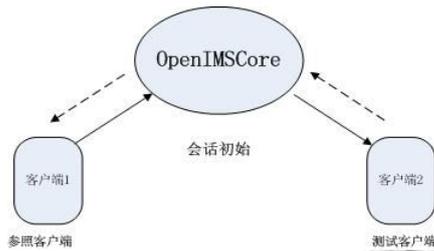


图 1 会话初始测试图

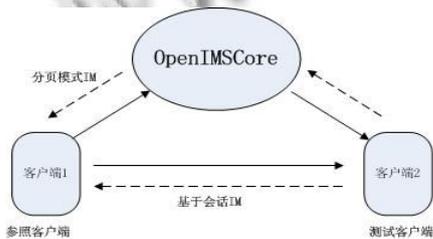


图 2 IM 测试图

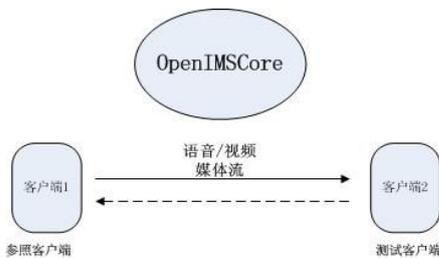


图 3 媒体会话测试图

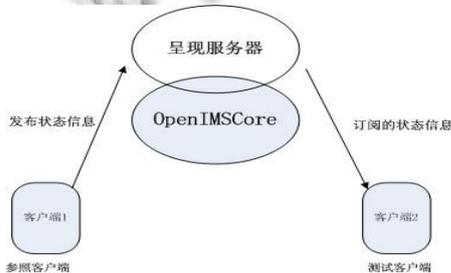


图 4 呈现功能测试图

从以上测试可以看到,基本的 IMS 应用是被广泛

支持的,如 SIP 信号层信息传输、分页模式的即时短信的互通性,但相对复杂的应用由于不同的开发人员对相关标准有不同的实现方式和 API 而不能成功实现互通.

那么就需要找一个统一框架下的客户端来作为通用 IMS 客户端,它要满足以下条件:

(1)开源.便于开发人员了解代码的实现方法,可以扩展代码来设计和融合新的服务和应用.

(2)适用于主流的操作系统的.根据 IMS 体系特点,IMS 客户端要在固定网络和无线网络都能够使用,固定网络的操作系统的非 Windows 莫属,无线网络的终端主要是手机,当下手机流行的操作系统是 Android 和 iOS(iPhone 操作系统),即同一框架下的客户端元素要有适用于 Windows 的也要有适用于 Android 和 iOS 的.

(3)具备 IMS 客户端的主要功能.起码要支持表 1 所罗列的功能.

Doubango 工程就是满足以上条件的理想工程.在第 4 部分将阐述 Doubango 框架下的客户端元素及其结构.

4 Doubango 的客户端元素

Doubango^[6]是一个兼容嵌入式和桌面系统的实验性开源的 3GPP IMS/LTE 框架.该框架由 ANSI-C 编写,这样可以提高移植性,有效地工作在嵌入式系统或有限内存及低计算能力的移动设备.它是一系列 API 的集合,实现多种协议,如信号压缩协议、IPSec、HTTP/HTTPS、XCAP、SDP、MSRP、SIP 以及 RTP.其中, tinyWRAP 是一个包装类,封装 Doubango 工程所有 API,以便开发者直接使用,简化代码工作,因此依赖其他 tiny 库.

Doubango 的客户端元素基于该框架来实现适用不同操作系统的主要 IMS 功能.客户端元素包括 Boghe^[7]、IMSDroid^[8]和 iDoubts,后两种客户端都是手机客户端,只是适用的操作系统不同,功能是相同的,在这里只描述 Boghe 和 IMSDroid.

4.1 Boghe

Boghe 是 Windows 平台下的 IMS 客户端,相对于第三部分描述的客户端, Boghe 具有更丰富的功能,比如带有多种呈现信息的好友列表并可以选择一位联系人来进行不同功能的通信的增强通讯录,可进行图片、视频、文本文件传输的内容分享功能. Boghe 客户

端的软件结构如图 5 所示。

除了 Dobango 中实现协议的 API, Boghe 包含 boghe-app、boghe-core、boghe-controls 和 boghe-xdm 四个项目。

1) boghe-app

该项目引用其余三个项目, 提供了客户端的主界面、短信窗口界面和媒体会话窗口界面, 以及不同按键下调用不同的功能函数。此外, 该项目也返回信号层出现错误的信息, 以使用户纠正错误。

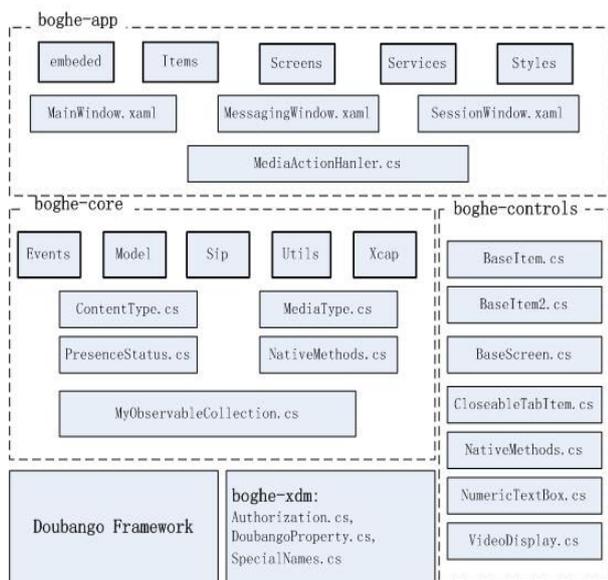


图 5 Boghe 客户端软件框架

(2) boghe-core

该项目引用 boghe-xdm, 实现客户端核心程序, 包含五个文件 Events、Model、Sip、Utils 和 Xcap。Events 定义一些事件所需的参数和类型, 如配置事件、联系人事件、通信记录事件、状态监视事件等。Model 定义客户端的默认设置及事件和服务的实现函数, 如对应于 Events 中的事件, 也有注册信息事件和监听者事件。Sip 定义 SIP 事件的参数和服务的建立, 这些事件包括邀请事件, 短消息事件, MSRP 协议事件, 注册事件, 订阅发布事件以及视频、音频的媒体会话事件。Utils 描述了一些实用工具的定义, 包含声音播放器、XML 文件读写器、日期时间显示和 SIP 地址替换工具。Xcap 则提供 XCAP 协议的功能, 使客户端读取 XCAP 服务器的订阅列表, 更新好友呈现状态。

(3) boghe-controls

该项目提供一些窗口的控制程序, 包括时间标签,

窗口标题处理, 窗口关闭处理, 文本框输入以及窗口的视频呈现。

(4) boghe-xdm

该项目解决 XDM 的实现问题, 做到有效存储并管理网络中的 XML 文档, 包含在线授权规则和资源列表管理。在线授权规则在该项目中含有 unknown、allowed、pending、blocked 及 revoked。资源列表管理的实现则是从服务器获取在线列表内容, 处理在线列表的订阅关系。

4.2 IMSDroid

IMSDroid 是适用手机操作系统 Android 的客户端软件, 用 Java 程序编写。

IMSDroid 同样引用 Doubango 框架, 最终开发的程序根据功能封装在不同的程序包里, 主要包含 sip、media、Model、Services、Screens 程序包。其软件架构如图 6 所示。

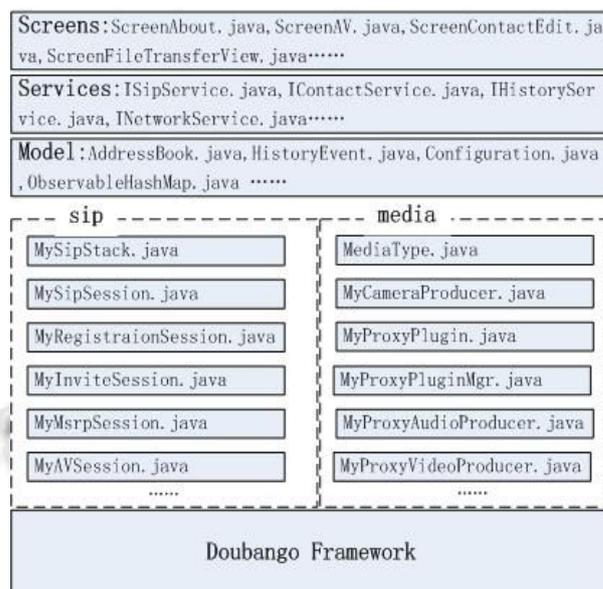


图 6 IMSDroid 客户端软件框架

(1) sip

sip 下的文件实现 SIP 会话的功能模块, 对于相关的 SIP 功能进行了定义, 包含定义 SIP 栈、用户注册、SIP 会话的建立、基于 MSRP 的信息会话建立、基于视频/音频的会话建立、客户端发布/订阅呈现状态以及呈现状态枚举声明。

(2) media

media 负责消息、音频、视频、文件传输的媒体

流管理和控制。该程序包枚举媒体类型，定义视频会议时设置本地视频数据缓存区并打开和切换摄像头，音频/视频通话插件的生成及流的呈现。

(3)Model

Model 包实现通讯录，通信记录及设置默认的软件各功能参数。通讯录功能含有组的呈现、添加和删除，在组内添加、删除联系人以及将通讯录内联系人设为状态观察者；通信记录功能记录各种通信记录，包含音频/视频通话、短信息和文件共享的记录，这些记录事件也具有状态类型的标识，如发出、接收、未接；配置类用参数区对软件参数进行默认配置。

(4)Services

Services 包定义了有关服务的类及其接口，包括参数设置、通讯录、通话记录、网络服务、屏幕显示、SIP 会话、调用提示音和 XCAP 服务等。

(5)Screens

该包的文件对应用程序的界面进行设置，包括界面显示，界面控制和界面程序响应。

Boghe 和 IMSDroid 具有一致的协议框架 Doubango Framework 作为基础协议栈，实现算法基本一致，因而这些不同平台上的客户端元素互通良好，并涵盖了固网和移动网的主流操作系统，它们成为 IMS 通用客户端具有相当大的优势。

5 功能扩展

Doubango 框架的客户端元素是开源的，并且开发者也留下了诸多接口，供后来的开发者添加或修改功能。Doubango 的客户端元素具有一对一的短信功能，但都不具有一对多的短信功能，即会议聊天功能。下面本文以 Boghe 为例来进行会议聊天功能扩展，实现多方用户同时进行文本类型的通信。

Boghe 以 C#编写，具有面向对象语言的所有特点，本文继承 ScreenGroupEdit 类和 ScreenContactEdit 类建立新的分组并在新分组里添加联系人，构建会议组，然后继承 MessagingWindow 类建立会议窗口界面，每个发言的组成员名字及其发言都会显示在对话框内。

在发送会议信息时，客户端要对会议信息做出标记以区别个人信息，同时在会议信息包头加上会议组名称以区别不同的会议组，远方客户端在收到信息时可以准确显示在相应的会议组内。

客户端的短信会议功能采用松耦合会议类型，即

会议中每个参加者间没有信令连接关系，没有会议中心，利用 SIP 的组播地址支持，具体流程如图 7 所示。

本文利用 pager(分页模式)用信令传递会议信息，MESSAGE^[9]是在 body 里携带会议消息内容的方法，不创建对话，body 里的消息格式是 plain/text。尽管每次传递消息需要通过核心网，但由于消息之间不关联，不需要走相同的路由，减小网络拥塞率，提高网络资源的利用率。

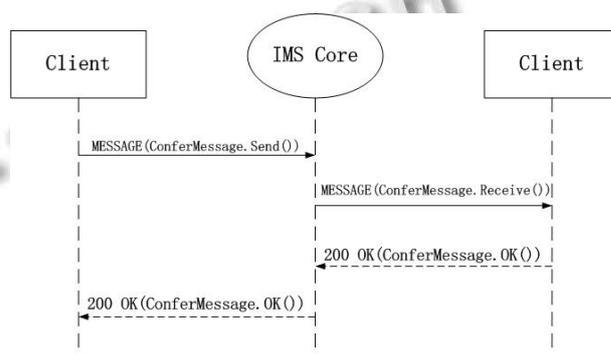


图 7 短信会议信令流程

Boghe 调用 ConferMessage.Send()方法发送 MESSAGE 消息，消息中携带组播地址及发言内容，由 IMS 核心网查找组播地址然后发送至各个接收方，各接收方触发 ConferMessage.Receive()事件检查 MESSAGE 包头的会议组名称，将接收的发言内容显示在相应会议组内。同时接收方触发 ConferMessage.OK()事件发送 200OK，发送方收到至少一个 200OK 触发 ConferMessage.OK()事件后可以发送下一个 MESSAGE。

6 结语

鉴于不同研究组织发布的 IMS 客户端互通性较差的缺点，本文引入了基于统一框架 Doubango 的客户端元素，分析了不同客户元素的软件架构，并在此基础上添加新的服务应用。这些客户端元素可应用于不同的操作系统，扩大了适用 IMS 技术的移动终端的范围，具有良好的互通性。它们具有的开源特性使全世界感兴趣的研究人员可以在本文描述的架构基础上更好地发展和改进它们的功能和服务，为用户带来更好的 IMS 体验。

参考文献

1 Poikselka M, Mayer G, Khartabil H. 赵鹏,周胜,望玉梅译,

(下转第 113 页)