北京郵電大學

计算机网络实验报告

题目：SIP 客户端的开源实现

姓　　名　程　潇

学　　院　网络空间安全学院

学　　号　2018111027

2018 年 10 月

目　录

第一章　背景知识

1.1　VOIP 技术

1.1.1　VoIP 简介

互联网协议语音（也称为 IP 语音，VoIP 或 IP 电话）是一种通过因特网协议（IP）网络传送语音通信和多媒体会话的方法和技术组。术语网络电话，宽带电话和宽带电话服务具体指的是通过公共互联网而不是通过公共交换电话网（PSTN）提供通信服务（语音，传真，SMS，语音消息）。

VoIP（Voice over Internet Protocol）简而言之就是将模拟信号（Voice）数字化，以数据封包（Data Packet）的形式在 IP 网络 (IP Network) 上做实时传递。VoIP 最大的优势是能广泛地采用 Internet 和全球 IP 互连的环境，提供比传统业务更多、更好的服务。VoIP 可以在 IP 网络上便宜的传送语音、传真、视频、和数据等业务，如统一消息业务、虚拟电话、虚拟语音/传真邮箱、查号业务、Internet 呼叫中心、Internet 呼叫管理、电话视频会议、电子商务、传真存储转发和各种信息的存储转发等。

1.1.2　VoIP 原理

发起 VoIP 电话呼叫所涉及的步骤和原理类似于传统的数字电话，涉及信号，信道设置，模拟语音信号的数字化和编码。数字信息不是通过电路交换网络传输，而是打包，并且通过分组交换网络作为 IP 分组进行传输。它们使用特殊的媒体传输协议传输媒体流，该协议使用音频编解码器和视频编解码器对音频和视频进行编码。存在各种编解码器，其基于应用要求和网络带宽来优化媒体流; 一些实现依赖于窄带和压缩语音，而其他实现则支持高保真立体声编解码器。一些流行的编解码器包括 μ-law 和 A-law 版本的 G.711，G.722，一种称为 iLBC 的开源语音编解码器，一种仅使用 8 kbit / s 的编解码器，称为 G.729，以及许多其他编解码器。

早期的 IP 语音服务提供商提供了商业模型和技术解决方案，它们反映了传统电话网络的架构。第二代提供商，如 Skype，为私人用户群建立了封闭式网络，提供免费通话和便利的好处，同时可能为访问其他通信网络（如 PSTN）收费。这限制了用户混合搭配第三方硬件和软件的自由。第三代提供商（如 Google Talk）采用了联邦 VoIP 的概念，这与传统网络的体系结构不同。当用户希望拨打电话时，这些解决方案通常允许互联网上任何两个域上的用户之间的动态互连。

除 VoIP 电话外，VoIP 还可用于许多个人计算机和其他互联网接入设备。可以通过移动数据或 Wi-Fi 发送呼叫和短信。VoIP 允许使用单一统一通信系统整合现代通信技术（包括电话，智能电话，语音和视频会议，电子邮件和存在检测）。

1.1.3 VoIP 关键技术

1.1.3.1 常用协议简介

已经使用基于开放标准的专有协议和协议以各种方式实现了 IP 语音。这些协议可以由 VoIP 电话，专用软件，移动应用程序使用或集成到网页中。VoIP 协议包括：

- **会话发起协议（SIP）**，由 IETF 开发的连接管理协议
- **H.323**，是第一个广泛实施的 VoIP 呼叫信令和控制协议之一。自从开发更新，更简单的协议（如 MGCP 和 SIP）以来，H.323 部署越来越局限于承载现有的长途网络流量。[需要引证]
- **媒体网关控制协议（MGCP）**，媒体网关的连接管理
- **H.248**，跨越融合互联网络的媒体网关的控制协议，包括传统的公共交换电话网（PSTN）和现代分组网络
- **实时传输协议（RTP）**，用于实时音频和视频数据的传输协议
- **实时传输控制协议（RTCP）**，用于 RTP 的姐妹协议，提供流统计和状态信息
- **安全实时传输协议（SRTP）**，RTP 的加密版本
- **会话描述协议（SDP）**，主要由 SIP 用于描述 VoIP 连接的文件格式
- **Inter-Asterisk eXchange（IAX）**，VoIP 服务器之间使用的协议
- **可扩展消息和状态协议（XMPP）**，即时消息，状态信息和联系人列表维护
- **Jingle**，为 XMPP 添加了点对点会话控制
- **Skype 协议**，基于点对点架构的专有 Internet 电话协议套件。

1.1.3.2 其它关键技术

- **媒体编码技术**, 主要包括的 G.711、G.723.1 和 G.729 等多媒体压缩编码技术。
- **媒体实时传输技术**, 主要采用实时传输协议 RTP。
- **业务质量保障技术**, 采用资源预留协议 RSVP 等。
- **网络传输技术**, 主要是 TCP 和 UDP。

1.2 SIP 技术

1.2.1 SIP 简介

会话发起协议（SIP）是一种信令协议，用于发起，维护和终止包括语音，视频和消息传递应用的实时会话。SIP 用于在用于语音和视频呼叫的因特网电话的应用中，在专用 IP 电话系统中，在因特网协议（IP）网络上的即时消息传送以及通过 LTE 的移动电话呼叫（VoLTE）中的信令和控制多媒体通信会话。

该协议定义了交换的消息的特定格式以及参与者合作的通信顺序。SIP 是一种基于文本的协议，包含超文本传输协议（HTTP）和简单邮件传输协议（SMTP）的许多元素。使用 SIP 建立的呼叫可以包括多个媒体流，但是在 SIP 消息中作为有效载荷交换数据的应用（例如文本消息）不需要单独的流。

SIP 与指定和携带会话媒体的其他几种协议一起使用。最常见的是，媒体类型和参数协商以及媒体设置使用会话描述协议（SDP）来执行，该协议作为 SIP 消息中的有效载荷来承载。SIP 被设计为独立于底层传输层协议，并且可以与用户数据报协议（UDP），传输控制协议（TCP）和流控制传输协议（SCTP）一起使用。对于通过不安全网络链路的 SIP 消息的安全传输，可以使用传输层安全性（TLS）来加密协议。对于媒体流（语音，视频）的传输，SIP 消息中携带的 SDP 有效载荷通常采用实时传输协议（RTP）或安全实时传输协议（SRTP）。

1.2.2 SIP 结构

1.2.2.1 协议操作

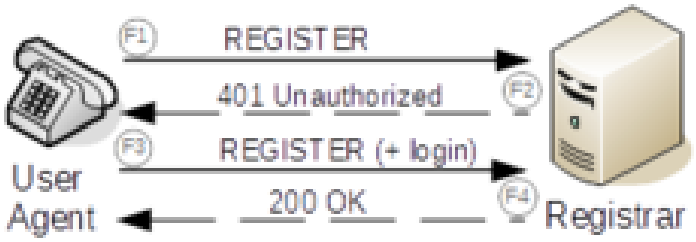


图 1-1 SIP 用户代理向 SIP 注册器注册并进行身份验证

SIP 仅涉及媒体通信会话的信令操作，主要用于建立和终止语音或视频呼叫。SIP 可用于建立双方（单播）或多方（多播）会话。它还允许修改现有的呼叫。修改可涉及更改地址或端口，邀请更多参与者，以及添加或删除媒体流。SIP 还在消息传递应用程序中找到了应用程序，例如即时消息传递，事件订阅和通知。

SIP 与其他几种协议协同工作，这些协议指定媒体格式和编码，并在呼叫建立后携带媒体。对于呼叫建立，SIP 消息的主体包含会话描述协议（SDP）数据单元，其指定媒体格式，编解码器和媒体通信协议。语音和视频媒体流通常使用实时传输协议（RTP）或安全实时传输协议（SRTP）在终端之间传输。

SIP 网络的每个资源（例如用户代理，呼叫路由器和语音邮箱）由统一资源标识符（URI）标识。URI 的语法遵循 Web 服务和电子邮件中也使用的通用标准语法。用于 SIP 的 URI 方案是 sip，典型的 SIP URI 的格式为 sip：username @ domainname 或 sip：username @ hostport，其中 domainname 需要 DNS SRV 记录来定位 SIP 域的服务器，而 hostport 可以是 IP 地址或者主机和端口的完全限定域名。如果需要安全传输，则使用方案啜饮。

SIP 采用类似于 HTTP 请求/响应事务模型的设计元素。每个事务由一个客户端请求组成，该请求调用服务器上的特定方法或函数以及至少一个响应。SIP 重用 HTTP 的大多数头字段，编码规则和状态代码，提供可读的基于文本的格式。

SIP 可以由若干传输层协议承载，包括传输控制协议（TCP），用户数据报协议（UDP）和流控制传输协议（SCTP）。SIP 客户端通常在端口号 5060 或 5061 上使用 TCP 或 UDP 来获取到服务器和其他端点的 SIP 流量。端口 5060 通常用于非加密信令流量，而端口 5061 通常用于使用传输层安全性（TLS）加密的流量。

基于 SIP 的电话网络通常实现信号系统 7（SS7）的呼叫处理功能，其中存在特殊的 SIP 协议扩展，尽管这两种协议本身是非常不同的。SS7 是一种集中式协议，其特点是复杂的中央网络架构和哑终端（传统电话手机）。SIP 是等效同伴的客户端 - 服务器协议。SIP 功能在通信端点中实现，而传统 SS7 架构仅在交换中心之间使用。

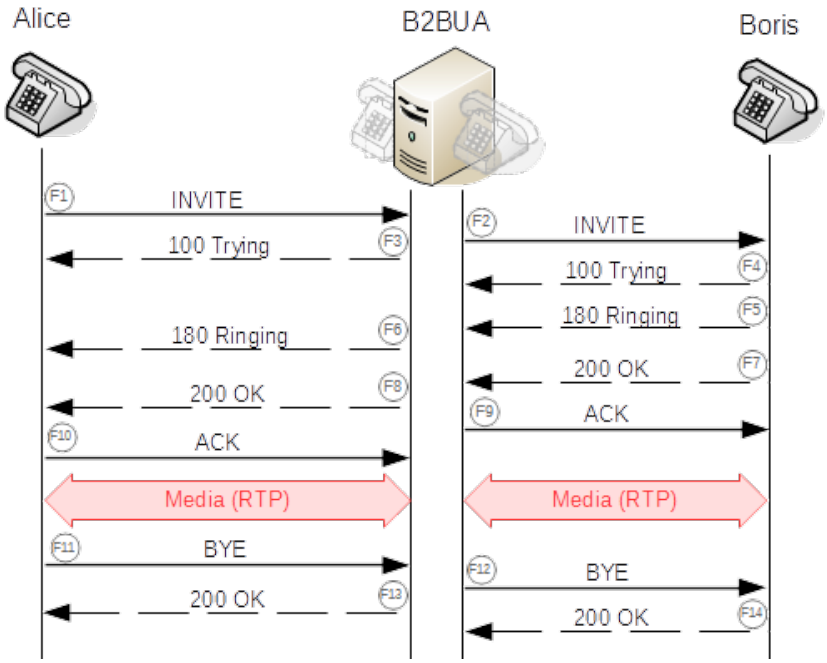


图 1-2 SIP 用户代理向 SIP 注册器注册并进行身份验证

1.2.2.2 网络元素

使用会话发起协议进行通信的网络元素称为 SIP 用户代理。每个用户代理（UA）在请求服务功能时执行用户代理客户端（UAC）的功能，在响应请求时执行用户代理服务

器（UAS）的功能。因此，任何两个 SIP 端点原则上可以在没有任何中间 SIP 基础设施的情况下操作。但是，出于网络操作原因，为了向用户提供公共服务以及为目录服务，SIP 定义了几种特定类型的网络服务器元素。这些服务元素中的每一个还在用户代理客户端和服务器中实现的客户端 - 服务器模型内通信。

用户代理是发送或接收 SIP 消息并管理 SIP 会话的逻辑网络端点。用户代理具有客户端和服务器组件。用户代理客户端（UAC）发送 SIP 请求。用户代理服务器（UAS）接收请求并返回 SIP 响应。与修复客户端和服务器角色的其他网络协议不同，例如在 HTTP 中，其中 Web 浏览器仅充当客户端，而从不充当服务器，SIP 要求两个对等体都实现这两种角色。UAC 和 UAS 的角色仅在 SIP 事务期间持续。

SIP 电话是一种 IP 电话，它实现 SIP 用户代理的客户端和服务器功能，并提供电话的传统呼叫功能，如拨号，接听，拒绝，呼叫保持和呼叫转移。SIP 电话可以实现为硬件设备或软电话。随着供应商越来越多地将 SIP 实施为标准电话平台，基于硬件和基于软件的 SIP 电话之间的区别是模糊的，并且 SIP 元件在许多具有 IP 功能的通信设备（例如智能电话）的基本固件功能中实现。

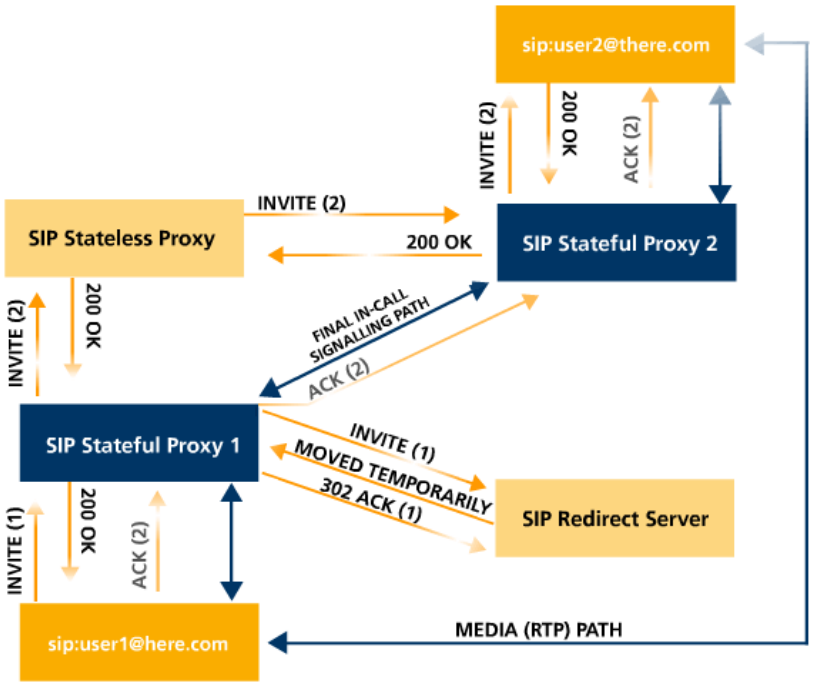


图 1-3 User1 的 UAC 使用 Invite Client Transaction 发送初始 INVITE（1）消息

示例：User1 的 UAC 使用 Invite Client Transaction 发送初始 INVITE（1）消息。如果在定时器控制的等待时段之后没有接收到响应，则 UAC 可以选择终止事务或重新发送 INVITE。收到回复后，User1 确信 INVITE 已可靠传送。然后，User1 的 UAC 必须确认响应。在交付 ACK（2）时，交易的双方都已完成。在这种情况下，可能已经建立了一个对话框。

在 SIP 中，如在 HTTP 中，用户代理可以使用消息头字段（用户代理）来标识自己，该消息头字段包含软件，硬件或产品名称的文本描述。用户代理字段在请求消息中发送，

这意味着接收 SIP 服务器可以评估此信息以执行特定于设备的配置或功能激活。SIP 网络元素的运营商有时会将此信息存储在客户帐户门户中，它可用于诊断 SIP 兼容性问题或显示服务状态。

代理服务器是具有 UAC 和 UAS 组件的网络服务器，其充当中间实体以便代表其他网络元件执行请求。代理服务器主要扮演路由的角色，这意味着它的工作是确保将请求发送到更靠近目标用户的另一个实体。代理对于实施策略也是有用的，例如用于确定是否允许用户进行呼叫。代理解释，并在必要时，在转发请求消息之前重写请求消息的特定部分。

SIP 用户代理向 SIP 注册器注册并进行身份验证。注册商是提供位置服务的 SIP 端点。它接受 REGISTER 请求，记录用户代理的地址和其他参数。对于后续请求，它提供了在网络上定位可能的通信对等体的基本手段。位置服务将一个或多个 IP 地址链接到注册代理的 SIP URI。多个用户代理可以注册相同的 URI，结果是所有注册的用户代理都接收对 URI 的调用。

1.2.3 SIP 地址和命名规则

SIP 消息的地址信息是基于 SIP 通用资源定位标记 (URL) 定义的:

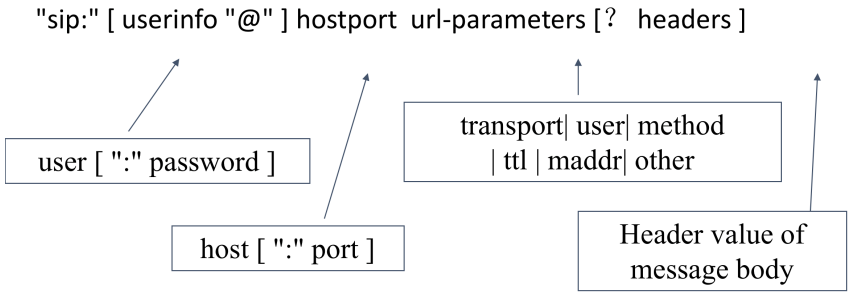


图 1-4 SIP 消息的地址信息

第二章　开源程序说明

2.1　PJSIP

2.1.1　PJSIP 简介

PJSIP 是一个用 C 语言编写的免费开源多媒体通信库，实现了基于标准的协议，如 SIP，SDP，RTP，STUN，TURN 和 ICE。它将信令协议（SIP）与丰富的多媒体框架和 NAT 遍历功能结合到高级 API 中，该 API 可移植，适用于从桌面，嵌入式系统到移动手持设备的几乎任何类型的系统。[1]

PJSIP 既紧凑又功能丰富。它支持音频，视频，状态和即时消息，并具有丰富的文档。PJSIP 非常便携。在移动设备上，它抽象出系统相关的功能，并且在许多情况下能够利用设备的本机多媒体功能。

2.1.2　PJSIP 优点

PJSIP 试图为开发人员提供构建实时多媒体通信应用所需的一切。PJSIP 已经处理了实时多媒体应用的所有三个主要组件，即信令，媒体特征和 NAT 遍历。把这些留给我们，你可以专注于应用程序逻辑。

1. 完整和整合

   PJSIP 试图为开发人员提供构建实时多媒体通信应用所需的一切。PJSIP 已经处理了实时多媒体应用的所有三个主要组件，即信令，媒体特征和 NAT 遍历。把这些留给我们，你可以专注于应用程序逻辑。

2. 非常便携

   编写您的应用程序一次，它将运行在任何类型的 Windows，Windows Mobile / CE 至 WM 6，Mac OS X PPC 和 Intel，任何处理器类型的 Linux，多种 Unix 系统，Nokia / Symbian 3rd 和 5th 版本设备，iPhone，iPad 和 iPod 上的 Apple iOS，BlackBerry 10 和 Android（计划在 v2.2 中）。PJSIP 也被用于嵌入式系统，人们报告在不同类型的处理器上成功使用嵌入式 OS / RTOS，如 uC-Linux，QNX 和 RTEMS。PJSIP 运行在 20Mhz MIPS 处理器设备上。

3. 紧凑，小而精悍

   语音呼叫应用程序使用较低级别的库从低至 150KB 开始，或使用较高级别的 PJSUA-LIB API 从几百 KB 开始，这两个 API 使用几百 KB 的堆使用。足迹当然会随着使用的功能而变化，但希望这给出了广泛的指示。

4. **完整的文档**

   我们还努力编写文档。在 API 的参考文档之上，我们在（Trac）wiki 站点上写了很多文章来帮助您进行开发。所有 PJSIP 文档都在 Trac 站点中编制索引。

5. **成熟**

   PJSIP 已经解决了许多现实问题，并且应用许多技巧来使事情发挥作用。

6. **专业开源**

   最后但同样重要的是，PJSIP 是开源软件（OSS）。开源已经使 PJSIP 能够被全球数千名开发人员使用，并且可能也被许多开发者仔细审查。寻找 PJSIP 开发人员变得越来越容易，作为开源软件，即使离开开发人员，PJSIP 代码也永远不会消失。

2.1.3 PJSIP 框架

PJSIP 包括以下几部分：

- PJSIP - Open Source SIP Stack[开源的 SIP 协议栈]
- PJMEDIA - Open Source Media Stack[开源的媒体栈]
- PJNATH - Open Source NAT Traversal Helper Library[开源的 NAT-T 辅助库]
- PJLIB-UTIL - Auxiliary Library[辅助工具库]
- PJLIB - Ultra Portable Base Framework Library[基础框架库]

PJMEDIA 是一个为 PJSIP 建立一个完整特性 SIP 用户代理应用提供的补充库，这些应用包括：softphones/hardphones，gateways or B2BUA. 使用 PJSIP 与 PJMEDIA 一起开发的应用，其具备如下的特性：

- 高度的可移殖性，与 PJSIP/PJLIB 一起，PJMEDIA 可运行在许多平台上，包括服务器、桌面、PDA 系统，定制的硬件、PDA 或移动电话。
- 多种功能：会议桥接、多种编解码器、丢包隐蔽/ PLC，音频发生器，静音探测器，声学回声消除/ AEC，RFC2833，RTP / RTCP 协议栈，speex/iLBC/GSM/G.711 编解码器等。
- 高质量：PJMEDIA 支持频率为 16KHz、32Khz 的编码和解码，事实上能支持任何音频采样率，可提供高质量的采样转换。PJMEDIA 也可以容忍一定量的网络或声音设备的不稳定和一些数据包丢失。
- 很好的支持嵌入式/DSP：占用内存小，灵活性好。该媒体组件被设计成可替换成相应功能的硬件。

PJNATH 是一个新的库，帮助应用程序进行 NAT 穿越。它实现了 NAT 穿越的最新规范：STUN、TURN 和 ICE。PJNATH 可以作为一个独立库，在您的软件中使用，也可以使用 PJSUA- LIB 库，该库很好的与 PJSIP，PJMEDIA 和 PJNATH 整合在一起，使用起来比较简单。PJNATH 有以下特点：

- STUNbis 实现，实现符合 RFC5389 标准。既提供需要使用的 STUN 网络接口，又提供基于 STUN 但更高层次的框架，既 TURN 和 ICE。
- NAT 类型检测，根据 RFC3489（STUN），在前端可以执行 NAT 类型检测。该检测方法不能对所有 NAT 类型进行穿越，但该信息可能仍然是有用，以便进行故障排除，已经被 ICE 整合，因此提供了该检测方式。
- TURN 实现，TURN 是一个中继通信协议，通过使用中继，并结合 ICE，提供了高效的最低代价的通信路径。PJNATH 中 TURN 的实现，符合 draft-ietf-behave-turn-14 草案。ICE 实现，ICE 是一个发现两个端点之间的通信路径协议。PJNATH 中 ICE 的实现符合 draft-ietf-mmusic-ice-19.txt 草案在未来，将实现更多的协议（如 UPnP IGD 和 SOCKS5）。

PJLIB-UTIL 是一个辅助库，为 PJMEDIA 和 PJSIP 提供支持。这个库中的一些功能/组件：占用内存小的 XML 解析，STUN 客户端库，异步/缓存 DNS 解析，哈希/加密功能等。占用内存小，高性能，高可移植性的抽象库和框架，被 PJSIP 和 PJMEDIA 使用。

PJLIB 是 PJLIB-UTIL、PJMEDIA 和 PJSIP 唯一依赖的库，因为它提供了完整的抽象，不仅仅是操作系统依赖的属性，还包括 LIBC 的抽象，并提供了一些有用的数据结构。PJLIB 基础框架库提供的功能：

- 内存的处理、数据的存储；数据结构的（hash 表、link 表、二叉树、等）;
- caching 和 pool；缓冲池和内存池;
- OS 抽象；
- 线程、互斥、临界区、锁对象、事件对象；
- 定时器,pj_str_t 字符串, 操作系统级别的函数抽像,socket 的抽象 (tcp/udp) , 文件的读写, 使用前的初始化，使用后的清理。

PJSIP 库框架如下图 2-1 :

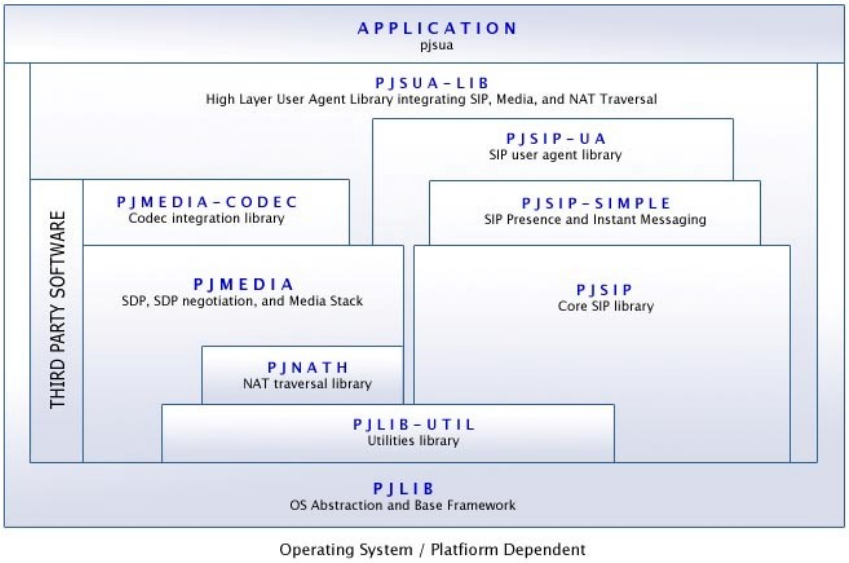


图 2-1 PJSIP 库框架结构图

第三章　实验过程和结果

3.1　下载开源代码，学习官方文档

1. 下载 pjsip 的源代码并阅读学习 pjsip 的编译说明。

2. 下载 VS2015 并熟悉编译环境。

3.2　编译开源工程文件

3.2.1　准备工作

本次实验在 VS2015 的环境下运行编译。

3.2.2　编译源程序

打开源文件，并且右键 pjsua，设置默认启动项目并编译, 项目工程目录结构如图 3-1 所示：

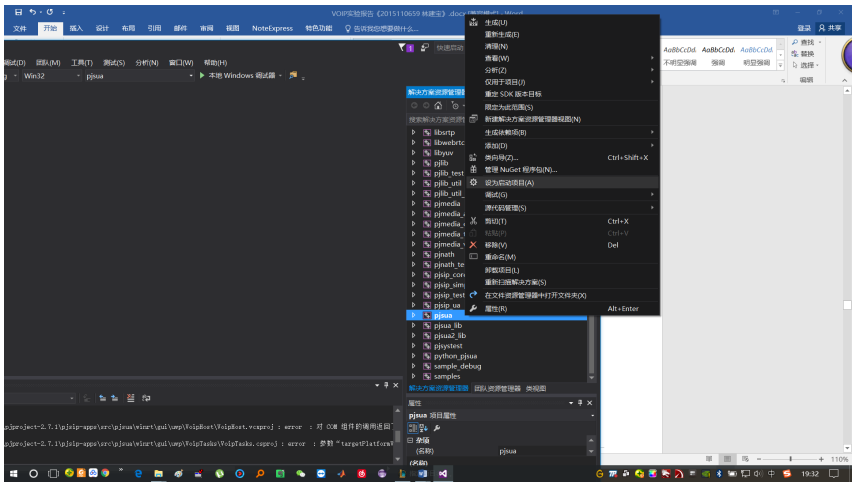


图 3-1　VS2015 工程目录结构

3.3　PJSIP 进行通话

3.3.1　注册

1. 运行 PJSIP 客户端文件，如下:

2. 输入 m，开始进行呼叫，然后输入 IVR 的 SIP 地址，命令为：sip:12345@10.105.242.72

图 3-2 软件注册界面

注册过程如下 (注意：根据语音提示，注意每一步输入数据时，都要先输入 * 键，然后才能输入有效数据。具体操作步骤如下：):

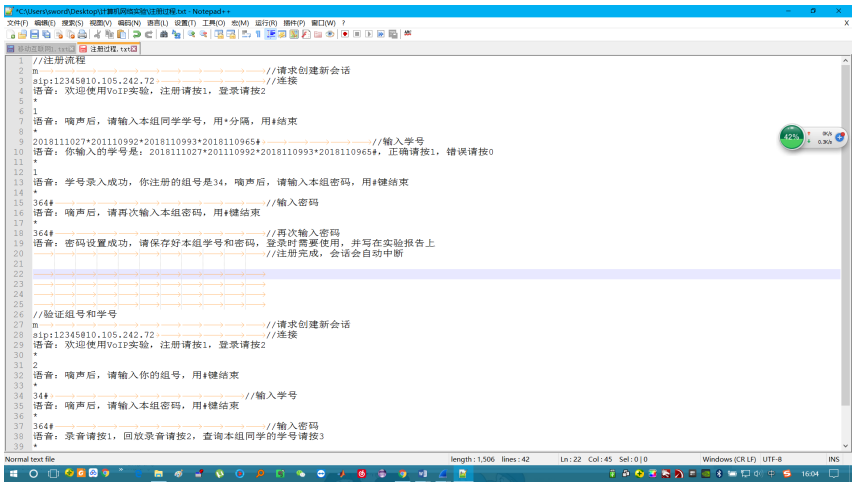


图 3-3 软件注册过程

3.3.2 验证组号和学号

步骤如下：

3.3.3 测试录音以及回放

步骤与验证学号相似，不再说明

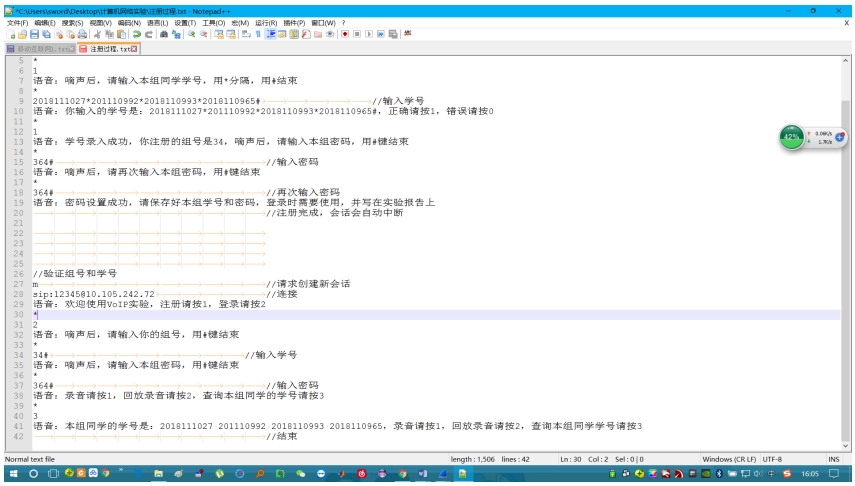


图 3-4 验证组号和学号

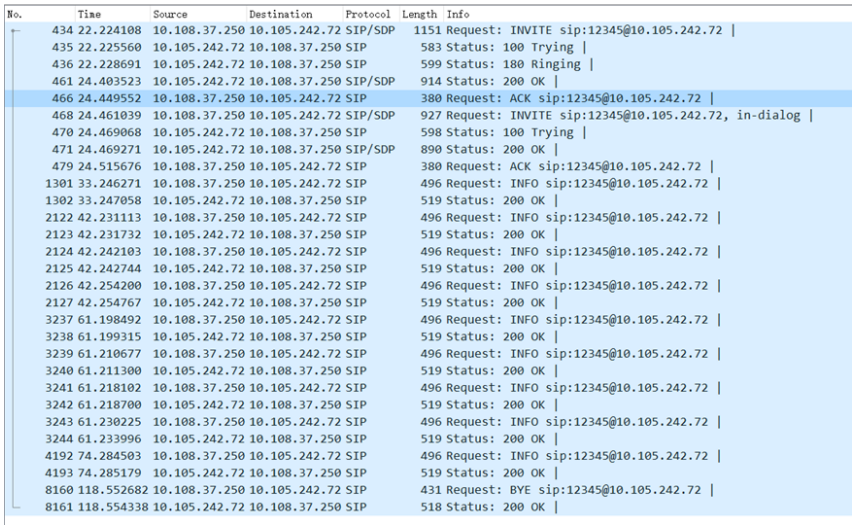
3.4 Wireshark 抓包分析

3.4.1 Wireshark 准备工作

打开程序后，需要选定抓取数据的网卡，这里只有一个本地连接，默认选择后，点击 Start 开始进行抓包。如果不进行过滤的话，会把所有数据包都抓到，这样数据量太多，不易分析。需要在 Filter 添加过滤条件 sip，以便方便查看我们需要得到的数据包，这样就过滤掉非 sip 的数据包。

3.4.2 Wireshark 抓取并分析数据包

做一个登录查询本组学号的的抓包实验，然后进行下列分析。当完成操作后，抓取到相应的数据包如下图：



No.	Time	Source	Destination	Protocol	Length	Info
434	22.224108	10.108.37.250	10.105.242.72	SIP/SDP	1151	Request: INVITE sip:12345@10.105.242.72 |
435	22.225560	10.105.242.72	10.108.37.250	SIP	583	Status: 100 Trying |
436	22.228691	10.105.242.72	10.108.37.250	SIP	599	Status: 180 Ringing |
461	24.403523	10.105.242.72	10.108.37.250	SIP/SDP	914	Status: 200 OK |
466	24.449552	10.108.37.250	10.105.242.72	SIP	380	Request: ACK sip:12345@10.105.242.72 |
468	24.461039	10.108.37.250	10.105.242.72	SIP/SDP	927	Request: INVITE sip:12345@10.105.242.72, in-dialog |
470	24.469068	10.105.242.72	10.108.37.250	SIP	598	Status: 100 Trying |
471	24.469271	10.105.242.72	10.108.37.250	SIP/SDP	890	Status: 200 OK |
479	24.515676	10.108.37.250	10.105.242.72	SIP	380	Request: ACK sip:12345@10.105.242.72 |
1301	33.246271	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
1302	33.247058	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
2122	42.231113	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
2123	42.231732	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
2124	42.242103	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
2125	42.242744	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
2126	42.254200	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
2127	42.254767	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
3237	61.198492	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
3238	61.199315	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
3239	61.210677	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
3240	61.211300	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
3241	61.218102	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
3242	61.218700	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
3243	61.230225	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
3244	61.233996	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
4192	74.284503	10.108.37.250	10.105.242.72	SIP	496	Request: INFO sip:12345@10.105.242.72 |
4193	74.285179	10.105.242.72	10.108.37.250	SIP	519	Status: 200 OK |
8160	118.552682	10.108.37.250	10.105.242.72	SIP	431	Request: BYE sip:12345@10.105.242.72 |
8161	118.554338	10.105.242.72	10.108.37.250	SIP	518	Status: 200 OK |

图 3-5 Wireshark 抓取到的数据包

分析确认请求的数据包，能看到一些信息如下图：

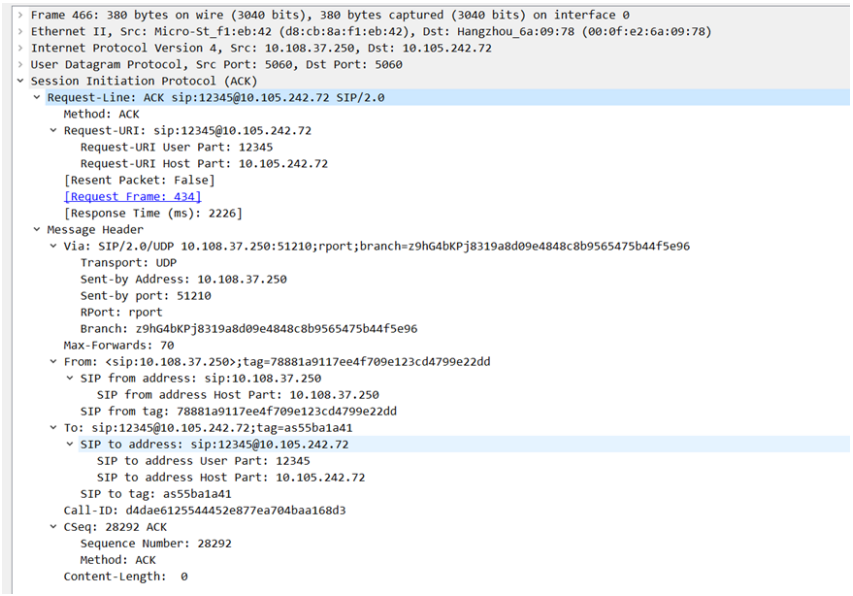


图 3-6 Wireshark 抓取到的数据包信息

继续分析：点击 Wireshark 菜单栏中的”Telephony”工具，选择“VoIP Calls”工具，如下图：

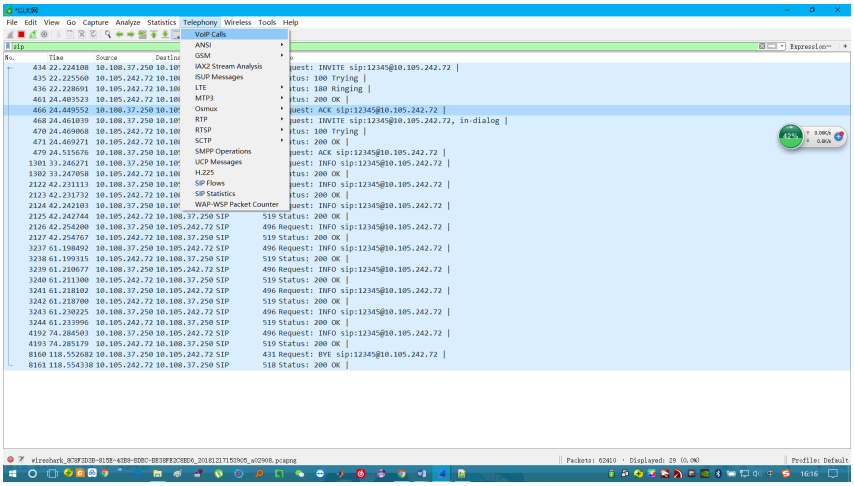


图 3-7 Wireshark 抓取到的数据包 Telephony 分析结果

然后，选中该 sip 的通讯过程，点击“Flow Sequence”按钮，便可以查看双方的通话过程：

结果如下两图：

上述是登录查询学号时的抓包实验，然后又进行了小组录音和回放等实验，数据包分析和上述大体相似，在此不一一截图描述了。

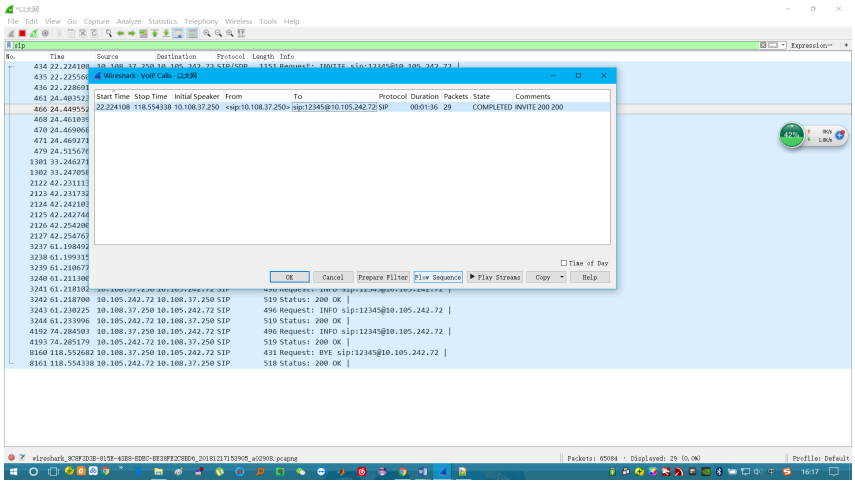
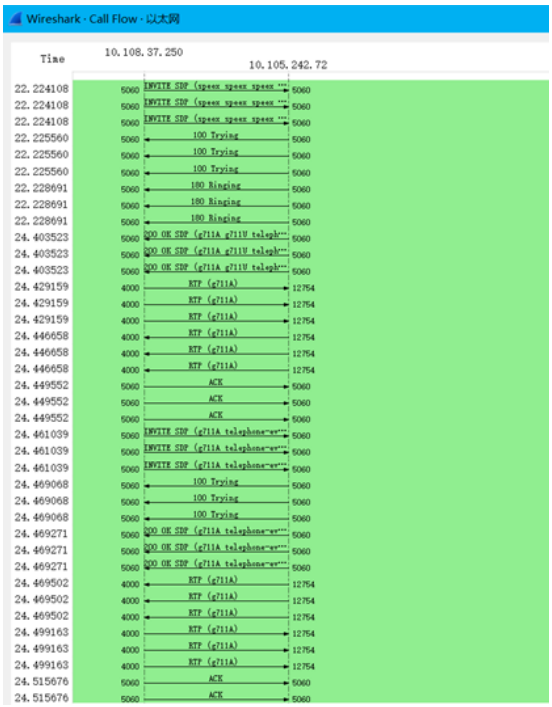


图 3-8 通话过程

(a)



(b)

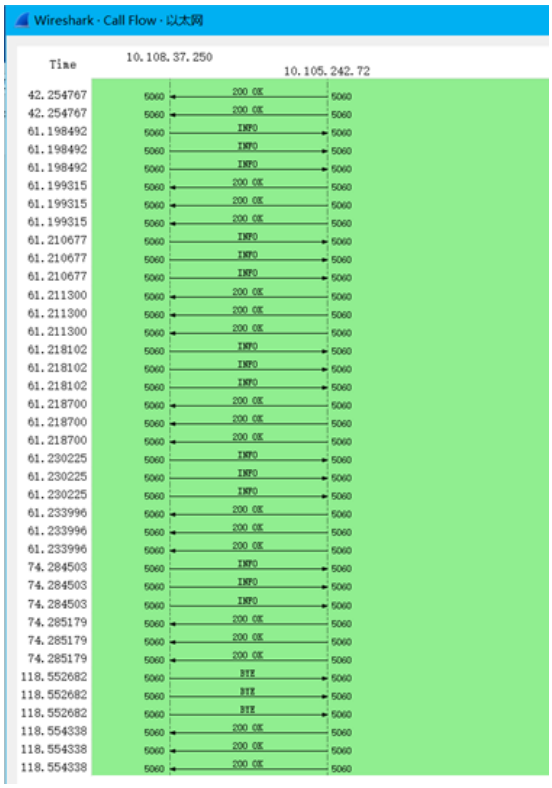


图 3-9 SIP 通信过程：(a)第一部分，(b)第二部分

3.5 问题及解决方法

3.5.1 输入超时

在刚开始直接输入数字及 # 号键时，会出现无法识别输入，应该在输入前先输入 * 号键再回车得到有效输入，如下图:

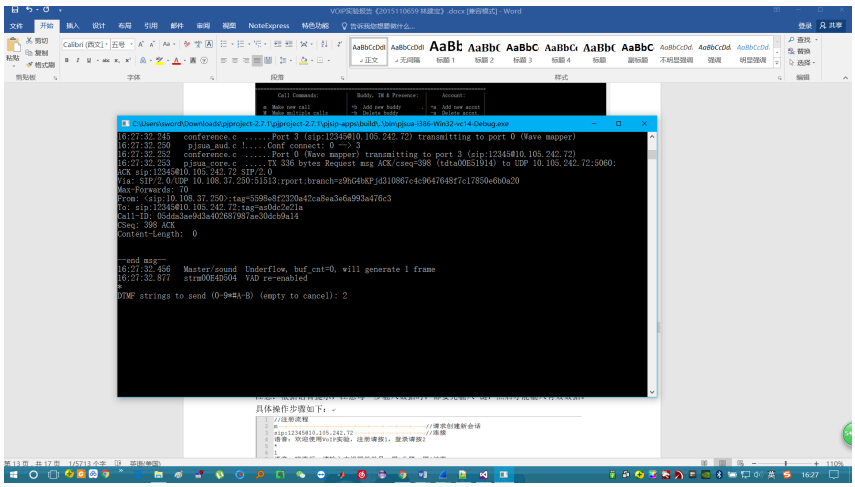


图 3-10 输入超时

参考文献

[1] Teluu Ltd. PJSIP [EB/OL]. 2018 [2018-09-05]. https://www.pjsip.org/.