

基于 WebRTC 应用层网关在 iOS 端的设计与实现^①

孙建伟¹, 李超^{1,2}, 于波¹

¹(中国科学院沈阳计算技术研究所, 沈阳 110168)

²(中国科学院大学, 北京 100049)

摘要: 随着近年来新媒体技术的快速发展, 用户对实时音视频通信的质量要求越来越高. WebRTC 技术的出现, 以其强大的音视频处理引擎迅速占领市场, 对多媒体通信行业产生了巨大的影响. 然而 WebRTC 提供的 JSEP 仅仅能完成简单的媒体链接功能, 在企业级的通信中需要结合其他模块或者信令协议才能胜任完整的应用. 本文着重研究了 WebRTC 与 SIP 的互通问题, 并在 iOS 端基于 WebRTC 技术设计实现了一种应用层网关, 通过实验验证了该网关的可行性与实用性.

关键词: WebRTC; 应用层网关; iOS; 实时通信

引用格式: 孙建伟, 李超, 于波. 基于 WebRTC 应用层网关在 iOS 端的设计与实现. 计算机系统应用, 2017, 26(10): 89-94. <http://www.c-s-a.org.cn/1003-3254/5987.html>

Design and Implementation of iOS Audio and Video Transmission Based on WebRTC

SUN Jian-Wei¹, LI Chao^{1,2}, YU Bo¹

¹(Shenyang Institute of Computer Technology, Chinese Academy of Sciences, Shenyang 110168, China)

²(University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: With the development of multimedia technology, and the real-time communication technology, people's requirement for real-time audio and video communication is higher and higher. The WebRTC technology emerges and occupies the market quickly based on its powerful audio and video processing engine, which has a huge impact on multimedia communications industry. WebRTC JSEP provided, however, can only perform simple media link function. It is required to combine with other modules or signaling protocol to do a complete application in the communication of enterprise. This paper focuses on the WebRTC communication with the SIP, and in the iOS terminal it designs and implements a kind of application layer gateway based on WebRTC technology. Finally, the feasibility and practicability of the gateway is verified with experiments.

Key words: WebRTC; application layer gateway; iOS; real-time communication

随着多媒体技术、实时通信技术的发展^[1], 各种社交和办公应用 app 增长迅速, 例如国外的 facebook, skype 等. 国内的微信, QQ 等都有音视频通信的功能. 伴随着移动端用户的不断增长, 人们通过手机进行音视频通信的需求量也日益增大, 如何提高音视频传输质量成为了重点.

WebRTC 技术使浏览器成为一个实时通信应用平台^[2]. 目前全球浏览器厂商越来越多的加入到 WebRTC

技术的大潮中. 2011 年 6 月 Google 提出了开源的即时通讯项目 WebRTC, 该项目提供了视频会议的核心技术: 音视频采集、网络传输、音视频编解码、信号优化和处理等^[3]. WebRTC 给移动通讯行业带来了新突破, 这将造成深远的影响.

SIP 是 IMS 的核心技术, 对多媒体会话的建立, 中断, 结束等起着很大的作用, 并已发展成得到广泛应用. WebRTC 强大的音视频处理引擎, 吸引越来越多的

① 基金项目: 国家科技重大专项-数控系统功能安全关键技术研究 (2014ZX04009031)

收稿时间: 2017-01-15; 采用时间: 2017-02-15

移动端开发人员研究. WebRTC 提供的 JSEP 是一种弱的信令控制协议, 仅仅能完成简单的媒体链接功能, 在企业级的通信中需要结合其他模块或者信令协议才能胜任完整的应用. 本文详细对比了 SIP 与 WebRTC 的差异, 以及 JSEP 与 SIP 转换存在的问题, 最终设计和实现在 iOS 端 WebRTC 与 SIP 融合的应用层网关.

1 WebRTC 与 SIP 转换在 iOS 端的技术简介

1.1 iOS 系统背景以及架构

iOS 是苹果公司于 2007 年发布的智能终端操作系统. iOS 系统分为以下四部分: 核心系统层、核心服务层、媒体层和交互层.

1.2 WebRTC 相关技术分析

WebRTC 是基于浏览器平台的实时音视频通信技术. 如图 1 所示, WebRTC 包含三个主要部分: 语音引擎模块, 视频引擎模块和传输模块.

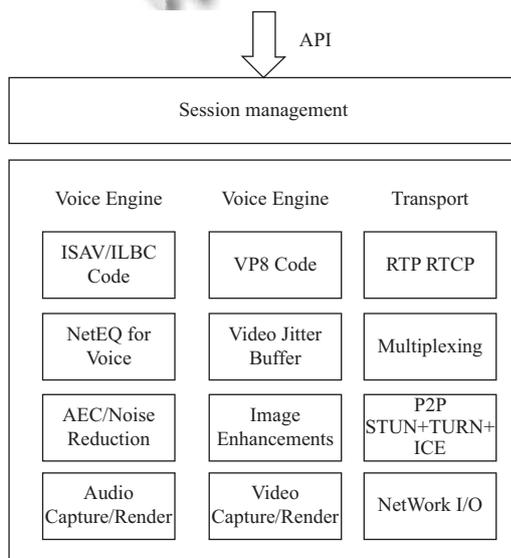


图 1 WebRTC 结构

1.3 SIP 协议简介

会话初始协议 SIP(session initiation protocol) 是基于文本的, 独立于传输层的应用层控制协议. 其用于建立, 结束以及中断多媒体会话^[4]. 是 IMS 的核心协议.

2 WebRTC 应用层网关的研究

2.1 需要解决的问题

WebRTC 在媒体流控制层面上提出了 JSEP 的连接方式. 在信令层对信令没有提出标准, 而是由开发者

自己选择. WebRTC 提供的 JSEP 是一种弱的信令控制协议, 其与强信令协议的不同在于仅仅能完成简单的媒体链接功能而不能完成整个会话过程的所有控制机制. 所以在企业级的通信中需要结合其他模块或者信令协议才能胜任完整的应用. 要实现 WebRTC 与 IMS 的核心协议 SIP 相结合, 重点在于如何解决信令互通问题.

如图 2 对二者协议栈的分析对比可得出, WebRTC 有一个封装好的基于浏览器的接口. WebRTC 通信一般使用 HTTP 或者 Socket, WebRTC 可以融合 SIP 做信令控制.

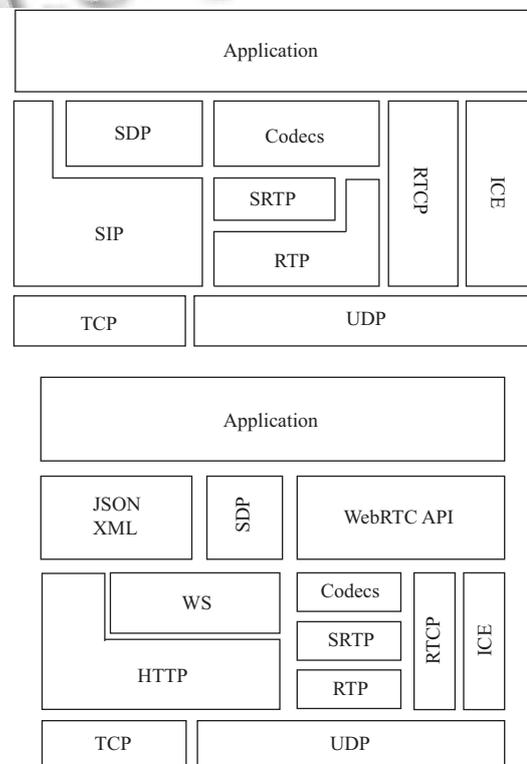


图 2 WebRTC 与 SIP 协议栈对比

如表 1 对 WebRTC 与 SIP 比较可知:

- (1) 媒体协商协议方面. 二者都使用了 SDP(会话描述协议), 而且都是 Offer/Answer 请求应答模型.
- (2) 在媒体传输方面, WebRTC 采用 SRTP, 而 SIP 采用了 RTP 也可以使用 SRTP. 没有使用 RTP 会话复用的 SIP 客户端可能无法理解 WebRTC 的 SDP.
- (3) 在传输控制方面, SIP 中 RTP 和 RTCP 是单独分开的, 在 WebRTC 中两者存在复用结构.
- (4) 在音频编解码二者均支持 G.711 标准而在视

频编解码方面 WebRTC 支持 VP8, 所以二者没有兼容的编解码。

表1 WebRTC 与 SIP 协议的不同

媒体协商	JSEP/SDP	SIP/SDP
信令传输机制	WebSocket	UDP, TCP(o)
媒体传输	SRTP	RTP, SRTP(o)
媒体传输通道	音视频使用相同窗口; SRTP和 SRTCP复用	音视频分开传输; RTP和 RTCP分离
NAT穿越	ICE	STUN(o), TURN(o), ICE(o)
音频编码	强制G.711, 其他可选	G.711, G.722等常见编码
视频编码	标准未定, 常用VP8	H.261, H.263, H.264

综上所述, 要解决 WebRTC 与 SIP 的互通问题, 需要进行信令转换, ICE 兼容以及编解码等方面的研究。本文重点是设计 WebRTC 的应用层网关, 对于媒体网关解决音视频编解码兼容性问题以及 ICE 兼容性就不做详细介绍了。

2.2 已有的解决方案

主流的解决方案有以下几种^[5]: 第一种是基于 JavaScript 实现 SIP 协议, 在其上构建 WebRTC, 第二种开发 SIP/WebRTC 转换网关内置到 WebRTC 服务器中, 第三种是开发 SIP/WebRTC 转换网关单独作为服务器。

第一种用 JavaScript 实现 SIP 协议并给 Web 应用程序开发者提供 JS API。这种 WebRTC 应用可以支持 SIP Server, 可与传统 SIP 客户端进行通信。

第二种和第三种方案是开发 SIP/WebRTC 网关实现 SIP 与 WebRTC 的互通。采用这种解决方案的开源项目 webrtc2sip^[6](由爱立信公司提出)。

上述三种解决方案都存在缺陷, 第一种方案是对 Web 应用程序开发很便利的解决方案, 对 iOS 客户端来说是不适用的。而第二种第三种是开发一个转换网关, 不管是嵌入到 WebRTC 服务器还是单独出一个服务器, 都增加了服务器的成本。如果用户量足够大很可能会使服务器性能大大降低。针对以上问题, 本文提出在客户端实现 JSEP-TO-SIP 的应用层网关结构。

3 系统设计

3.1 系统框架

如图3所示的设计中包含了 iOS 客户端(内嵌 JSEP-TO-SIP 的信令网关), 媒体网关, SIP 服务器。本文主要思

想是将信令网关与媒体网关分离, 将 JSEP-TO-SIP 网关嵌入到 iOS 端, 进而减小服务器的压力和开销成本。

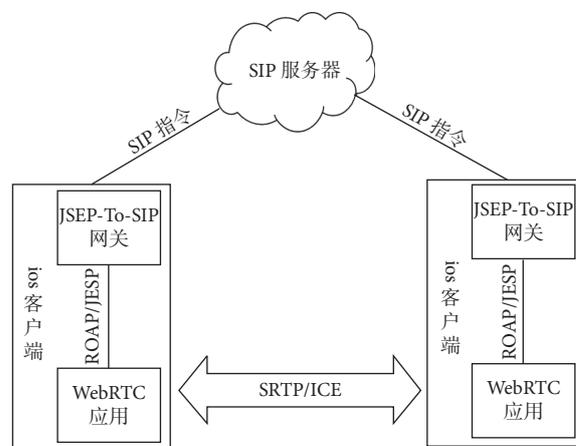


图3 iOS 客户端到 iOS 客户端的信令通道

iOS 客户端通过 JSEP-TO-SIP 网关对 ROAP/JESP 到 SIP 的协议转换和对等的 iOS 客户端连接。iOS 客户端应用程序主要实现用户登录, 拨号, 视频显示界面等。JSEP-TO-SIP 网关负责把 JSEP 信令转换成 SIP 信令, 并发送给 SIP 服务器, 以及将 SIP 信令转换成 JSEP 信令。通过信令交换媒体信息后, 两端建立起 P2P 的 SRTP 媒体通道。

如表2是 JSEP 与 SIP 的映射关系。

表2 JSEP 到 SIP 映射

JSEP API	SIP Message
createOffer(), setLocalDescription()	SIP:Invite
createAnswer()	SIP:200 OK
setRemoteDescription()	SIP:ACK
close()	SIP:BYE
addStream(), removeStream()	SIP:Re-Invite
PeerConnectionErrorCallBack()	SIP:4xx, 5xx, 6xx

JSEP-TO-SIP 网关负责 JSEP 到 SIP 的转换以及 SIP 到 JSEP 的转换。重点解决的就是 WebRTC 与 SIP 协议栈的区别而产生的互通的问题。

3.2 JSEP-TO-SIP 网关工作流程举例

如图4所示, JSEP 与 SIP 的交互过程为:

(1) ClientA 和 ClientB 分别创建自己 RTCPeerConnection 实例。

(2) ClientA, ClientB 通过 RTCPeerConnection 所提供的 setLocalDescription() 方法, 将自己的 SDP 描述符等其它参数信息交给 RTCPeerConnection 实例^[7]。

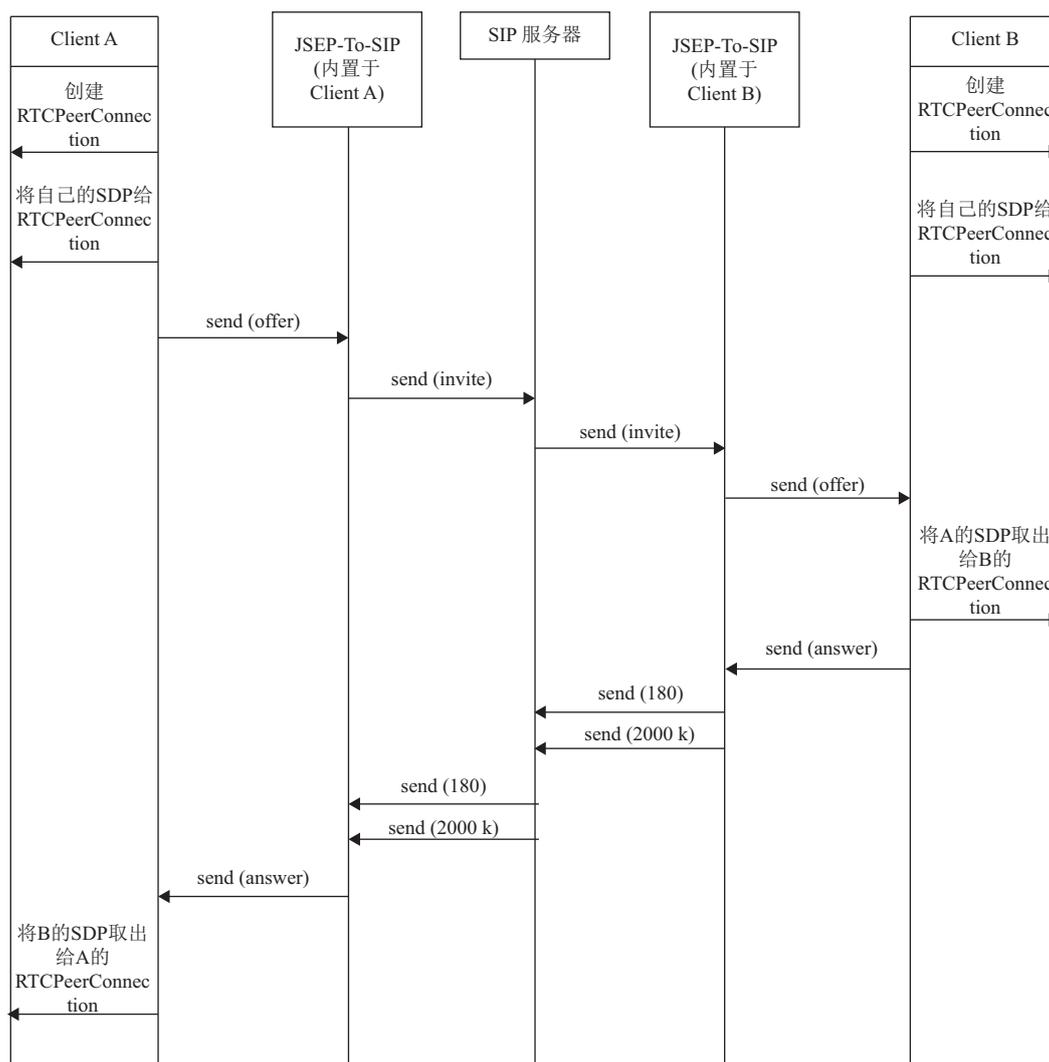


图4 JSEP 与 SIP 信令交互过程

(3) ClientA 通过 RTCPeerConnection 所提供的 createOffer() 方法建立一个包含 ClientA 的 SDP 描述符等其它参数信息的 offer 信令. ClientA 将 Offer 信令发送给 JSEP-TO-SIP 网关.

(4) JSEP-TO-SIP 网关将 Offer 信令转换成 SIP 信令的 Invite 并发送给 SIP 服务器, SIP 服务器根据被叫标识, 找到 ClientB 并发送给 ClientB 的 JSEP-TO-SIP 网关.

(5) ClientB 将信令中所包含的 SDP 描述符等其它参数信息提取出来, 通过 RTCPeerConnection 所提供的 setRemoteDescription() 方法交给 ClientB 的 RTCPeerConnection 实例^[7].

(6) ClientB 通过 RTCPeerConnection 所提供的

createAnswer() 方法建立一个包含 ClientB 的 SDP 描述符等其它参数信息的 Answer 信令. ClientB 将 Answer 信令发送给 JSEP-TO-SIP 网关.

(7) JSEP-TO-SIP 网关将 Answer 转换成 SIP 的 200 ok 发送给 SIP 服务器, SIP 服务器发送给 ClientA 的 JSEP-TO-SIP 网关.

(8) ClientA 将其中 ClientB 的 SDP 描述符等其它参数信息提取出来, 用 setRemoteDescription() 方法交给 ClientA 自己的 RTCPeerConnection 实例^[7].

经过以上步骤, ClientA 和 ClientB 所建立的 RTCPeerConnection 实例包含了 ClientA 和 ClientB 的 SDP 描述符等其它参数信息^[7]. 建立 P2P 的 SRTP 媒体通道, 客户端可以进行媒体传输.

3.3 WebRTC 在 iOS 端的应用框架

从谷歌官网下载 WebRTC 源码, 得到源码后, 将其编译, 然后与 PJSIP 封装成在 iOS 平台可用的 SDK 如图 5 所示为 WebRTC 与 iOS 结合架构图。

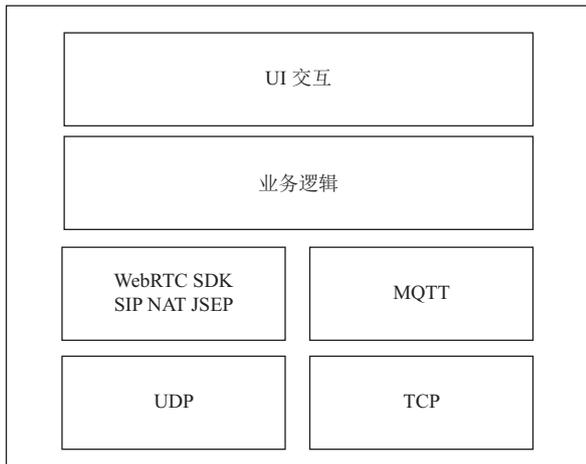


图 5 iOS 与 WebRTC 结合架构图

采用 MVC 模型对 UI 设计, 其对项目文件易于管理, 并且用户只关心界面的视图效果。

WebRTC 语音引擎支持 iSAC、iLBC、Opus 3 种编解码器。音频 NetEQ 算法包括抖动缓冲及丢包补偿模块以延迟减至最小并提高音频质量^[8]。

视频部分包含采集、编解码 (VP8)、加密、媒体文件、图像处理与显示、网络传输与媒体流控制等技术^[9]。

4 测试及比较结果

本文完成了在 iOS 端 WebRTC 与 SIP 融合的应用层网关, 并实现了在 iOS 客户端的 WebRTC 应用的音视频实时通信。应用层网关主要是基于 Node.js 实现的, WebRTC 提供的 JSEP 是基于 RTCPeerconnection 开发的, RTCPeerconnection 是 Node.js 向开发者提供的一个模块, 用于支持 WebRTC 服务。SIP 模块基于 pjsip 库开发^[10]。

由于 WebRTC 采用 JSEP 作为控制协议, 与 SIP 服务器交互时需要转换为 SIP 信令。本文搭建的 SIP/WebRTC 互通实验环境用 Asterisk 做 SIP 服务器, 由于 Asterisk 具备媒体编解码能力以及 SRTP/RTP 转换功能, 所以该实验环境中, Asterisk 也用作媒体网关。

本次实验用 15 台 iPhone 进行测试, 建立音视频连

接, 相互通信共计 20 次。测试中通过 Wireshark 抓包, 获得 Offer 信令经应用层网关生成的 INVITE 消息和 Answer 信令经应用层网关生成的 200 OK 应答消息, 分别如图 6、图 7 所示。分析可知, 信令网关在 IMS 网络中表现的和传统 SIP 终端一致。20 次实验中, 在同一局域网内 WebRTC 应用进行通信, JSEP-TO-SIP 网关均可正常工作, 达到预期效果, 说明本文所提出的在 iOS 端实现 JSEP-TO-SIP 网关是可行的。

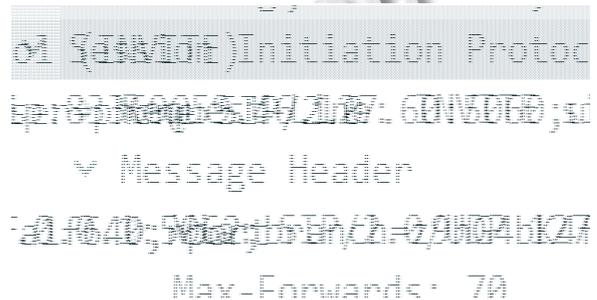


图 6 JSEP-TO-SIP 网关生成的 SIP 请求消息

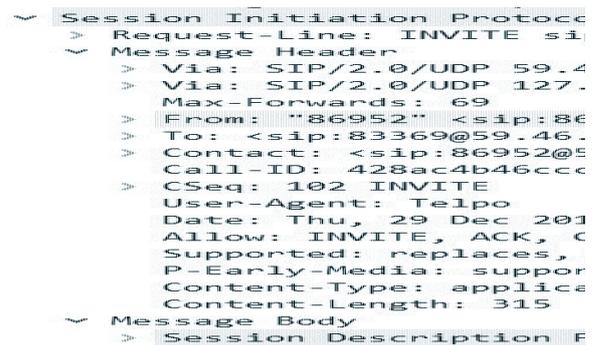


图 7 JSEP-TO-SIP 网关生成的 SIP 应答消息

实验过程中, WebRTC 应用实时音视频通信流畅, 不过从双方视频画面来看, 存在 170 ms-270 ms 的延迟。JSEP-TO-SIP 网关对信令的转换带来一定的时间延迟, 但比在服务器端实现信令转换的延迟有所减少, 放到服务器处理网关转换需要跟服务器交互以及服务器转换网关需要做两次转换, 而 iOS 端延迟的 170 ms-270 ms 是可以接受的。

5 结论

本文分析了 SIP 和 WebRTC 控制协议转换问题, 介绍了现存的解决方案, 并提出了在 iOS 端实现 JSEP-TO-SIP 网关。通过搭建实验环境, 验证了该方案的可行性以及优越性。当然还需要考虑到 JSEP-TO-SIP 网

关的负载能力, 时间延迟以及处理协议转换对 iOS 端性能的影响和耗电量等问题.

参考文献

- 1 陈建明, 张翼翔. 用于移动学习的流媒体服务器研究. 微型机与应用, 2013, 32(3): 10–12.
- 2 竹洪涛. 一种基于 SIP 和 WebRTC 的实时可视对讲方案设计[硕士学位论文]. 成都: 西南交通大学, 2013.
- 3 王桐. 基于 Android 智能移动终端的无线远程管控系统研究与应用[硕士学位论文]. 成都: 西南交通大学, 2013.
- 4 邓敦望. SIP 视频会议服务器设计与实现[硕士学位论文]. 北京: 北京邮电大学, 2010.
- 5 Amirante A, Castaldi T, Miniero L, *et al.* On the seamless interaction between WebRTC browsers and SIP-based conferencing systems. IEEE Communications Magazine, 2013, 51(4): 42–47. [doi: 10.1109/MCOM.2013.6495759]
- 6 Mamadou DIOP. Webrtc2sip-smart SIP and media gateway for WebRTC endpoints technical guide. <http://www.webrtc2sip.org/technical-guide-1.0.pdf>. [2013].
- 7 胡勋, 周渊平. 基于 WebRTC 的 Android 移动端无线视频传输. 微型机与应用, 2015, 34(19): 83–85. [doi: 10.3969/j.issn.1674-7720.2015.19.024]
- 8 吴江锐. WebRTC 语音引擎中 NetEQ 技术的研究[硕士学位论文]. 西安: 西安电子科技大学, 2013.
- 9 胡敏, 刘六程, 刘鹏. 基于 WebRTC 的视频会议系统的设计与实现. 电视技术, 2013, 37(1): 141–143, 150.
- 10 黄月祥. 基于 PJSIP 的嵌入式 VoIP 终端的研究与实现[硕士学位论文]. 昆明: 昆明理工大学, 2011.