

# 基于 SIP 协议的 WebRTC 信令研究与应用<sup>①</sup>

孙建伟<sup>1</sup>, 陈立<sup>1,2</sup>, 王卫<sup>1</sup>

<sup>1</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110168)

<sup>2</sup>(中国科学院大学, 北京 100049)

通讯作者: 陈立, E-mail: [chuzhouchenli@163.com](mailto:chuzhouchenli@163.com)

**摘要:** 在近年来随着用户对音视频通话质量要求的提高, WebRTC 以其强大的多媒体处理能力得到了广泛的应用. 然而 WebRTC 提供的 JSEP 是一种弱信令, 在企业级的融合通信应用中必须将 WebRTC 与实际的信令协议相结合. SIP 是 IMS 的核心技术, 对多媒体会话的控制起着非常重要的作用. 本文介绍了 WebRTC 和 SIP 协议融合的已有方案, 研究了 WebRTC 和 SIP 协议互通需要解决的问题, 提出了一种 WebRTC 的 PeerConnection 层和 SIP 协议在客户端的融合方案, 并和其他方案对比, 得出该方案的优缺点.

**关键词:** WebRTC; PeerConnection; SIP; 音视频; 融合通信

引用格式: 孙建伟, 陈立, 王卫. 基于 SIP 协议的 WebRTC 信令研究与应用. 计算机系统应用, 2018, 27(9): 273-277. <http://www.c-s-a.org.cn/1003-3254/6519.html>

## Research and Application of WebRTC Signaling Based on SIP Protocol

SUN Jian-Wei<sup>1</sup>, CHEN Li<sup>1,2</sup>, WANG Wei<sup>1</sup>

<sup>1</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

<sup>2</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** In recent years, with the user's requirements on the audio and video communications quality being higher, WebRTC has been widely used for its powerful multimedia processing capabilities. WebRTC only provide a kind of weak signaling JSEP, but enterprise-class converged communications applications must be combination of WebRTC and the actual signaling protocol. SIP protocol is the core technology of IMS, which plays a very important role in the control of multimedia conversation. This paper introduces the existing schemes of WebRTC and SIP protocol integration, studies the problems of WebRTC and SIP protocol integration, and presents a scheme of converged communications of combining WebRTC PeerConnection and SIP protocol based on clients. This study also compares the advantages and the disadvantages of this scheme with other schemes.

**Key words:** WebRTC; PeerConnection; SIP; audio and video; converged communications

Web 实时通信 (WebRTC)<sup>[1]</sup> 是一种构建在 Web 浏览器上的实时音视频通信技术. WebRTC 由 Google 收购 Global IP Solution 公司而获得的一项技术, 并在 2011 年将其开源. WebRTC 提供了音视频采集、网络传输、音视频编解码、信号优化和处理等<sup>[2]</sup> 一整套的音视频通信解决方案. 由于 WebRTC 强大的多媒体处

理引擎, WebRTC 已经在 Chrome、Firefox、Opera、Android、iOS 等浏览器和平台上得到了支持.

会话初始协议 (Session Initiation Protocol, SIP) 是一个应用层的信令控制协议, 是 IMS 的核心的、成熟的、已经得到广泛应用的技术, 用于创建、修改和释放一个或多个参与者的会话. WebRTC 提供的 JSEP 是

① 收稿时间: 2018-01-15; 修改时间: 2018-02-09; 采用时间: 2018-02-26; csa 在线出版时间: 2018-08-16

一种仅维护信令状态机的弱信令控制协议,在企业级融合通信应用中必须将 WebRTC 和实际的信令协议相结合。

本文介绍了 WebRTC 和 SIP 融合已有的两种方案,分析了 WebRTC 和 SIP 互通需要解决的问题,提出了一种 WebRTC 的 PeerConnection 层和 SIP 协议在客户端融合的方案,最后在多种客户端实现了 WebRTC 和 SIP 融合的应用,并和其他方案对比得出了此方案的优缺点。

## 1 相关技术简介

### 1.1 SIP 协议简介

SIP 是基于文本的、独立于传输层的应用层 IMS 核心协议。其用于建立、结束以及中断多媒体会话<sup>[3]</sup>。

### 1.2 WebRTC 简介

WebRTC 语音引擎支持 iSAC、iLBC、Opus 等多种编解码器。音频 NetEQ 算法包括抖动缓冲和丢包补偿模块以延迟减至最小并提高音频质量<sup>[4]</sup>。WebRTC 视频引擎包含采集、编解码 (VP9、VP8、可添加 H264 等编解码器)、加解密、媒体文件、图像处理与显示、网络传输与媒体流控制等技术<sup>[5]</sup>。WebRTC 提供 JSEP 这样的弱信令目的就是为了 WebRTC 强大的多媒体处理能力可以和不同的实际的信令协议相结合,如 SIP、ROAP、XMPP、Jingle 等<sup>[6]</sup>。

## 2 WebRTC 的 PeerConnection 层和 SIP 协议在客户端融合方案的研究

从信令的角度来看,当下主要有两种 SIP 和 WebRTC 互通方案<sup>[7]</sup>:一种是基于服务器开发 SIP/WebRTC 转换网关。另一种是用 JavaScript 实现 SIP 协议,在此协议栈的基础上构建 WebRTC 应用。

### 2.1 基于服务器端的信令转化网关实现 WebRTC 和 IMS 网络互通

这种方案通过开发 SIP/WebRTC 转化网关实现 SIP 信令和 WebRTC 信令的转化,如 webrtc2sip<sup>[8]</sup>就是这种方案的典型代表。基于服务器端的信令转化网关实现 WebRTC 和 IMS 网络互通架构图如图 1 所示。

### 2.2 基于 JavaScript 实现的 SIP 协议开发 IMS 客户端

这种方案用 JavaScript 实现 SIP 协议并向 Web 应用开发者提供 JavaScript API。开发者调用 WebRTC

JavaScript API 开发出的 WebRTC 应用可以采用 WebSocket 传输 SIP 信令。开发者通过此类 WebRTC 应用直接注册并登录支持 WebSocket<sup>[9]</sup> 的 SIP Server,与传统 SIP 终端进行通信。使用这种解决方案的开源 SIP 项目有 Jssip 和 SipML5。

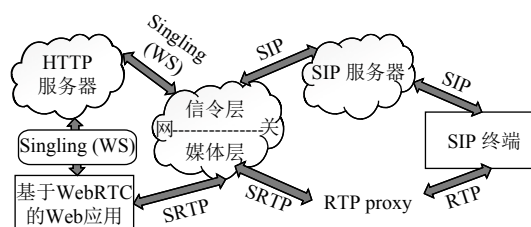


图 1 基于服务器端的信令转化网关实现 WebRTC 和 IMS 网络互通架构

### 2.3 WebRTC 的 PeerConnection 层和 SIP 协议在客户端融合方案

WebRTC C++ API (PeerConnection 层) 是面向浏览器厂商的,使的浏览器厂商容易在此基础上实现 WebRTC 标准的 Web API。本文提出了一种 WebRTC 的 PeerConnection 层和 SIP 协议在客户端实现融合互通的方案,可以适当避免上述两种方案的缺陷。该方案主要通过内嵌于客户端 SIPRTC 本地网关做 WebRTC SDP 和 SIP SDP 的转化及 SIP 信令和 WebRTC PeerConnection API 的映射。该方案基于 WebRTC PeerConnection API,但不局限于 C++ API,也包括编译等手段产生的其他语言的 PeerConnection 层 API,相应的不同终端结合的 SIP 模块会有所不同。WebRTC 的 PeerConnection 层和 SIP 协议在客户端的融合方案架构如图 2 所示。

方案一需要在 SIP 服务器上开发为了兼容 WebRTC 信令的转换网关,开发成本较高,服务器也会因信令的转换产生一定的延迟。但是该方案会减轻客户端的开发负担。方案二需要浏览器支持 WebRTC,虽然目前主流的浏览器都已支持 WebRTC,但限于浏览器的 WebRTC JavaScript API 尚处于草案阶段,在此基础上开发应用会有所不便。但这种开发方式的开发流程较为简单,代码可以实现跨平台。这两种案都需要服务器端传输通道支持 WebSocket;且这两种方案开发的都是 Web 应用,Web 应用在性能上会低于原生应用。

本文提出的方案首先可以不必在服务器端做兼容 WebRTC 的开发,从而减小服务器的压力和信令转换造成延迟;其次不需要服务器传输通道支持 WebSocket,

从而减小服务器传输通道的限制;最后由于此时客户端由本来的 Web App 一跃变成 Native App,也一定程度上提高了客户端的性能.该方案一定程度上加大了客户端的工作量,但对比方案一的工作量,该方案的工作量完全可以接受.

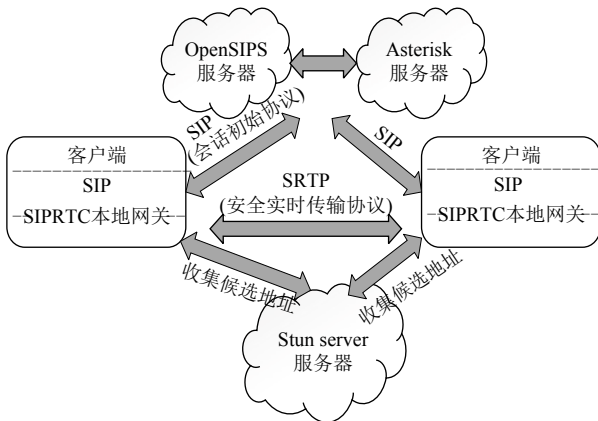


图2 WebRTC的PeerConnection层和SIP协议在客户端的融合方案架构

### 3 WebRTC的PeerConnection层和SIP协议在客户端融合系统设计与实现

本方案中最重要的一个环节就是SIPRTC本地网关的设计,其功能包括WebRTC SDP和SIP SDP转化和SIP信令和WebRTC PeerConnection API映射.

#### 3.1 WebRTC的PeerConnection层和SIP协议在客户端融合方案的本地网关设计

WebRTC使用JSEP(JavaScript Session Establish Protocol)提供的弱信令完成媒体协商.JSEP提供的信令分为主叫方的Offer信令和被叫方的Answer信令,信令中的主要信息均符合SDP(Session Description Protocol)标准.WebRTC和SIP在媒体和信令的异同点对比如表1所示.

表1 WebRTC和SIP媒体和信令的异同点对比表

类别	属性	WebRTC	SIP
信令	信令协议	尚未指定	SIP
	媒体协商	JSEP/SDP	SIP/SDP
	信令传输	WebSocket	UDP、TCP
	NAT穿越	ICE	STUN、TURN、ICE
媒体	媒体传输	SRTP	RTP、SRTP
	媒体传输通道	音视频使用相同端口	一般音视频端口分开
	音频编解码	iSAC、iLBC等	G.711、G.722等
	视频编解码	VP9、VP8等	H264、H263等

本方案中客户端通过SIPRTC本地网关做信令转化和接口映射,WebRTC PeerConnection和SIP协议的映射流程如图3所示,其中Alice是主叫,Bob是被叫.WebRTC PeerConnection和SIP映射步骤如下:

(1) Alice调用PeerConnection层createoffer API,并通过SIPRTC本地网关转化成SIP(INVITE)消息发送给Bob;

(2) Bob接收SIP消息后,调用PeerConnection层setRemoteDescription、createAnswer API,并通过SIPRTC本地网关转化成SIP(180 Ring、200 OK)消息发送给Alice;

(3) Alice接收SIP消息,调用PeerConnection层setRemoteDescription API,并发送SIP(ACK)消息给Bob,此时建立了p2p或者通过多媒体中转服务器的多媒体会话;

(4) Alice调用PeerConnection层removeRemoteStream API,并通过SIPRTC本地网关转化成SIP(BYE)消息发送给Bob;

(5) Bob接收SIP消息,调用PeerConnection层removeRemoteStream API,并发送SIP(200 OK)消息给Alice,至此完成一次音视频会话.

#### 3.2 WebRTC的PeerConnection层和SIP协议在客户端融合方案的应用实现

图4为客户端应用设计的架构,在完成SIPRTC本地网关后,业务逻辑只需调用SIP接口,从而便于业务逻辑的实现;MQTT用于文本聊天等功能的实现.

视各客户端不同,WebRTC的PeerConnection在不同平台结合的SIP模块稍有不同,具体来说Windows平台选择pjsip<sup>[10]</sup>,Android平台选择sipdroid,iOS平台选择sofiasip;相应的WebRTC的PeerConnection API不局限于某种语言的API,包括编译手段产生的C++ API、Java API、OC API.

### 4 WebRTC的PeerConnection层和SIP协议在客户端融合方案的测试及对比

本实验在服务器端使用OpenSIPS部署SIP服务器,使用Google的Stun Server作为ICE代理,使用Asterisk服务器部署多媒体中转服务器.

在Android、iOS、Windows客户端完成了WebRTC PeerConnection和SIP协议的融合,并测试了各平台之间的音视频通信.SIPRTC本地网关转化的呼

叫和响应信令如图5和图6所示。Android端发出呼叫 并和iOS端视频通话如图7和图8所示。

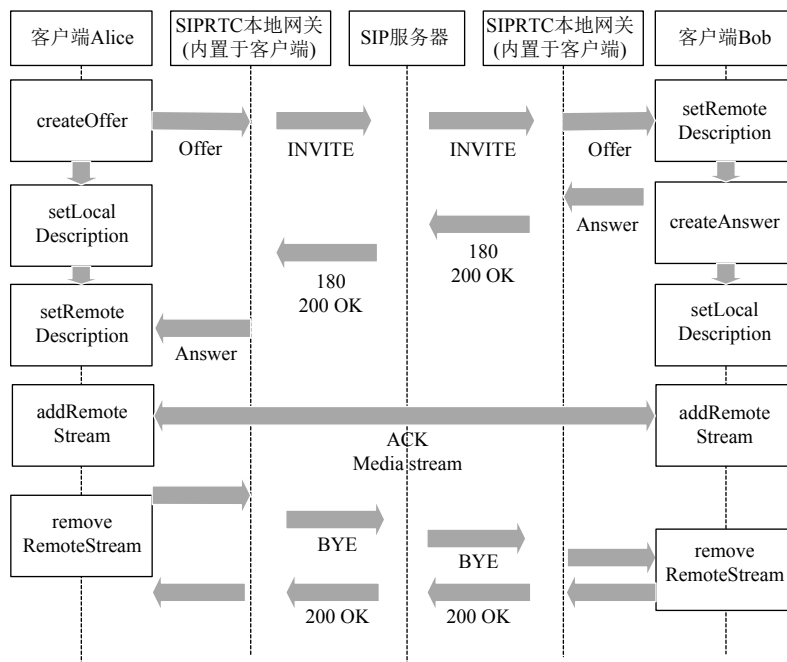


图3 WebRTC PeerConnection 和 SIP 映射流程

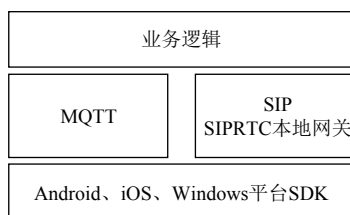


图4 客户端应用设计架构

```

> Session Initiation Protocol (SIP)
> Request-Line: INVITE sip:192.168.139.212:62944;transport=tcp SIP/2.0
> Message Header
> Message Body
  > Session Description Protocol
    > Session Description Protocol Version (v): 0
    > Owner/Creator, Session ID (o): - 2599286979667575354 2 IN IP4 0.0.0.0
    > Session Name (s):
    > Time Description, active time (t): 0 0
    > Session Attribute (a): group:BUFILE audio video
    > Session Attribute (a): mid-semantic: WMS ARDAMS
    > Media Description, name and address (m): audio 17174 RTP/SAVPF 103 111 9 102 0 8 106 105 13 126
    > Connection Information (c): IN IP4 0.0.0.0
    > Media Attribute (a): rtp:1715 IN IP4 0.0.0.0
    > Media Attribute (a): candidate:5 1 UDP 16777215 0.0.0.0 17174 typ relay
    > Media Attribute (a): candidate:5 1 UDP 16777215 0.0.0.0 14724 typ relay
    > Media Attribute (a): candidate:2395300328 1 udp 2122194687 192.168.1.110 51835 typ host generation 0
    > Media Attribute (a): candidate:2395300328 2 udp 2122194687 192.168.1.110 51835 typ host generation 0
    > Media Attribute (a): candidate:2395300328 1 udp 2122194687 192.168.1.110 51835 typ host generation 0
    > Media Attribute (a): candidate:2395300328 2 udp 2122194687 192.168.1.110 51835 typ host generation 0
    > Media Attribute (a): ice-ufrag:10Q116D1Igbn/Cc
    > Media Attribute (a): ice-pwd:TPAF1/dluVQPSZuLdphIj
    > Media Attribute (a): ice-options:google-ice
    > Media Attribute (a): mid:audio
    > Media Attribute (a): extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
    > Media Attribute (a): extmap:3 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
    > Media Attribute (a): sendrecv
    > Media Attribute (a): rtp:opus
    > Media Attribute (a): crypto:1 AES_CM_128_HMAC_SHA1_80 inline:kZEh0M61FaEMPGf433EInthR0M808PLoED4G8HtH
    > Media Attribute (a): rtptime:111 opus/48000/2
    > Media Attribute (a): fmp:111 minptime=10
    > Media Attribute (a): rtpmap:103 ISAC/16000
    > Media Attribute (a): rtpmap:9 G722/16000
    > Media Attribute (a): rtpmap:102 ILBC/8000
    > Media Attribute (a): rtpmap:8 PCMA/8000
    > Media Attribute (a): rtpmap:186 CN/32000
    > Media Attribute (a): rtpmap:105 CN/16000
    > Media Attribute (a): rtpmap:13 CN/8000
    > Media Attribute (a): rtpmap:127 red/8000
  
```

图5 SIPRTC 本地网关转化的呼叫信令

```

> Session Initiation Protocol (SIP)
> Status-Line: SIP/2.0 200 OK
> Message Header
> Message Body
  > Session Description Protocol
    > Session Description Protocol Version (v): 0
    > Owner/Creator, Session ID (o): - 763538463993773815 2 IN IP4 127.0.0.1
    > Session Name (s):
    > Time Description, active time (t): 0 0
    > Session Attribute (a): group:BUFILE audio video
    > Session Attribute (a): mid-semantic: WMS ARDAMS
    > Media Description, name and address (m): audio 68525 RTP/SAVPF 103 111 102 0 8 106 105 13 126
    > Connection Information (c): IN IP4 192.168.1.112
    > Media Attribute (a): rtp:9 IN IP4 0.0.0.0
    > Media Attribute (a): candidate:617392483 1 udp 2122268223 192.168.1.112 68525 typ host generation 0 network-id 1 network-cost 10
    > Media Attribute (a): candidate:1783984547 1 tcp 1518288447 192.168.1.112 62950 typ host tcp-type passive generation 0 network-id 1 network-cost 10
    > Media Attribute (a): ice-ufrag:VtUu
    > Media Attribute (a): ice-pwd:cu7M08JEJt0brglxq3oY340
    > Media Attribute (a): ice-options:renomination
    > Media Attribute (a): mid:audio
    > Media Attribute (a): extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
    > Media Attribute (a): extmap:3 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
    > Media Attribute (a): sendrecv
    > Media Attribute (a): rtp:opus
    > Media Attribute (a): crypto:1 AES_CM_128_HMAC_SHA1_80 inline:R0wz2o5wEIQ3k4P/c/Eghs/39CjJxK/8562
    > Media Attribute (a): rtpmap:103 ISAC/16000
    > Media Attribute (a): rtpmap:105 CN/16000
    > Media Attribute (a): rtpmap:111 opus/48000/2
    > Media Attribute (a): fmp:111 minptime=10;useinbandfec=1
    > Media Attribute (a): rtpmap:102 ILBC/8000
    > Media Attribute (a): rtpmap:8 PCMA/8000
    > Media Attribute (a): rtpmap:186 CN/32000
    > Media Attribute (a): rtpmap:105 CN/16000
    > Media Attribute (a): rtpmap:13 CN/8000
    > Media Attribute (a): rtpmap:126 telephone-event/8000
    > Media Attribute (a): ssrc:3399186748 cname:p3jM0x77Vb0LLta
    > Media Attribute (a): ssrc:3399186748 msid:8020P5 dcd17523-6324-47de-a708-b65432938697
    > Media Attribute (a): ssrc:3399186748 mslabel:8020P5
    > Media Attribute (a): ssrc:3399186748 label:dcd17523-6324-47de-a708-b65432938697
    > Media Description, name and address (m): video 9 RTP/SAVPF 100 116 117 96
    > Connection Information (c): IN IP4 0.0.0.0
    > Media Attribute (a): rtp:9 IN IP4 0.0.0.0
    > Media Attribute (a): ice-ufrag:VtUu
  
```

图6 SIPRTC 本地网关转化的响应信令

经过多次测试和抓包分析,验证了 WebRTC 的 PeerConnection 层在客户端融合 SIP 协议方案的可行性,对比前两种方案,发现本方案的客户端流畅度要高于前两种方案的客户端,且本方案服务器造成的延迟要低于方案一的服务器,特别是服务器负载较大时,本方案服务器造成的延迟会明显低于方案一的服务器。

### 5 结论

本文介绍了已有的 WebRTC 和 SIP 协议融合的方案

案,研究了 WebRTC 和 SIP 协议互通需要解决的问题,提出了一种 WebRTC 的 PeerConnection 层和 SIP 协议在客户端融合的方案,通过查阅大量资料、搭建试验环境、编码、测试和对比,验证了该方案的可行性和优越性.但是由于使用 Google 提供的 Stun Server 做 ICE 代理,有时会有明显的延迟,后期的工作将集中研究、开发及搭建自己的 ICE 代理;另外后期我们也将不同应用场景下对各客户端的稳定性做测试,尽早将该方案产品从实验室推向市场.

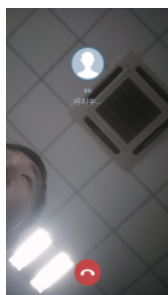


图7 Android 端发出呼叫

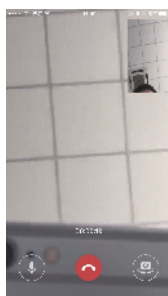


图8 iOS 端视频会话

## 参考文献

- 1 梁艳. 基于 HTML5 的 WebRTC 技术浅析. 信息技术, 2014, 8(2): 52-56.
- 2 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.
- 3 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.
- 4 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.
- 5 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, (1): 141-143, 150. [doi: [10.3969/j.issn.1002-8692.2013.01.040](https://doi.org/10.3969/j.issn.1002-8692.2013.01.040)]
- 6 刘明栓. WebRTC 系统中信令子系统的设计与实现[硕士学位论文]. 北京: 北京邮电大学, 2014.
- 7 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](https://doi.org/10.1109/MCOM.2013.6495759)]
- 8 Mamadou DIOP. webrtc2sip-smart SIP and media gateway for WebRTC endpoints technical guide. <http://www.webrtc2sip.org/technical-guide-1.0.pdf>. [2013].
- 9 draft-ietf-sipcore-sip-Websocket-04. <http://tools.ietf.org/html/draft-ietf-sipcore-sip-Websocket-04>. [2013-08-22].
- 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.