音视频RTC模块概要设计

广立信息技术研究部

2017-05

文档变更记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 描述 |
| Draft | 2017-5-11 | 孙纲 | 初稿. |
| 1.0 | 2017-6-9 | 孙纲 | Reject中增加reason；增加notify信令 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[音视频RTC模块概要设计 1](#_Toc484772102)

[1. 总体设计 4](#_Toc484772103)

[1.1. 设计目标 4](#_Toc484772104)

[1.2. 结构示意图 4](#_Toc484772105)

[2. 各节点说明 6](#_Toc484772106)

[2.1. A端 6](#_Toc484772107)

[2.2. B端 7](#_Toc484772108)

[2.3. SPRTC 7](#_Toc484772109)

[2.4. SG 7](#_Toc484772110)

[3. 信令定义 8](#_Toc484772111)

[3.1. invite 8](#_Toc484772112)

[3.2. ringing 8](#_Toc484772113)

[3.3. ok 8](#_Toc484772114)

[3.4. reject 9](#_Toc484772115)

[3.5. modify 9](#_Toc484772116)

[3.6. report 9](#_Toc484772117)

[3.7. bye 10](#_Toc484772118)

[3.8. ack 10](#_Toc484772119)

[3.9. Offer/Answer 10](#_Toc484772120)

[3.10. ICE Candidate 12](#_Toc484772121)

[3.11. notify 12](#_Toc484772122)

# 总体设计

## 设计目标

使用webrtc音视频模块实现与PM集成会话业务流程。

## 结构示意图



说明:

1. 实线表示走TCP，虚线表示Webrtc模块内部的音视频流
2. 所有的TCP信令都是notification，对方无需应答

约定:

1. 所有的TCP信令都经由SG转发

比如：A发送invite，经由SG转发到特定的SPRTC

表达为：A发送invite给SPRTC

1. ack信令

一般ack是需要进行确认的信令的应答,包括ok和modify

而reject，bye这种信令一般无需对方的ack

# 各节点说明

## A端

一次会话的发起者，只有一个，具体描述如下：

1. 会话开始于invite，在发送invite之前，发起者需要创建webrtc的对等端（peer connection），添加本地媒体（media stream），如果失败，则直接提示失败，会话结束；如果创建成功，则发送invite给SPRTC
2. 收到ringing放回铃音
3. 收到ok表示对方同意会话请求，应该发送ack给对方，表示收到了，会话由此建立
4. 收到reject表示对方拒绝会话请求，无需发送ack，这次会话直接结束了
5. 收到ok需要与自己的发送的audio/video进行比对，取中间一致的信息。
6. 收到ok时，本地即要创建对等端的offer请求，创建成功则发送给SPRTC；创建失败则提示失败并发送bye信令给SPRTC，会话结束
7. 收到对方的answer应答，需要设置对等端answer，用来建立webrtc连接
8. ICE Candidate是对等端给的回调生成的，生成后即发送给SPRTC，ICE Candidate和offer、answer之间没有顺序要求，只要对等端回调即发送给SPRTC
9. 在invite之后任何时候都可以发送bye来结束这次会话，而且无需等待对方的ack
10. 发送bye，或者收到bye时，这次会话结束

说明:

1. 需要做超时处理，一方面是服务可能出问题，一方面是对方不反应，超时时请发送bye，走完正常流程
2. 创建对等端时需要的STUN服务和TURN服务在登录时SPRTC模块参数中获取

|  |
| --- |
| <service id="rtc">  <parameter>  <urls>  <url>turn:freemine.site:3478?transport=udp</url>  </urls>  <username>ninefingers</username>  <credential>youhavetoberealistic</credential>  </parameter>  </service> |

## B端

一次会话的接受者，只有一个，具体描述如下：

1. 会话开始于收到SPRTC发过来的invite信令
2. 发送ringing到相应的SPRTC，提示用户振铃，等待用户同意或拒绝
3. 用户同意，先要创建本地的对等端，创建本地媒体流，如果失败，则直接bye，会话结束；如果成功，发送ok给发起方，并带上用户相应的媒体收发信息
4. 用户拒绝，发送reject给发起方，会话结束
5. 如果发送ok给发起方，则需要等到对方的ack信令，等到ack后，会话正式建立
6. 收到发起方对等端的offer请求，需要设置到对等端，并且产生answer回应
7. Answer回应和ICE Candidate都由对等端的回调产生，产生后随即发送给SPRTC，Answer和ICE Candidate之间没有顺序要求
8. 建立会话后，可以发送modify来修改audio/video数据的发送接收信息
9. 发送bye，或者收到bye时，这次会话结束

说明:

1. 需要做超时处理，一方面是服务可能出问题，一方面是对方不反应，超时时请发送bye，走完正常流程
2. audio/video的接收发送需要注意，双方协商的时候需要取相对应的部分
3. 创建对等端时需要的STUN服务和TURN服务在登录时SPRTC模块参数中获取

## SPRTC

提供会话的服务节点

1. 一次会话开始于收到一个A发送过来的invite信令
2. 收到PSC发送bye消息，进行转发，该会话结束
3. 转发所有的TCP信令
4. 做超时处理，如果没有TCP信令，则该会话结束，并给双方发送bye信令

## SG

会话中所有的TCP信令的转发者，转发时，会明确告诉信息接收者，信息的源是谁，只分析修改message元素的一些属性，不修改，也不检查内部子元素

1. 在login应答信令中，SG就会告知绑定SPRTC的param(STUN和TURN服务器列表)
2. 在收到A的invite请求时，知道服务名RTC，然后根据此服务名，到这个A的服务绑定表里面找到绑定的服务节点，把该信令发给它；如果这个服务节点不在了，那就什么都不做，并不通知A
3. SG在检测到某一个SPRTC不在的时候，不会主动更新原先被分配与此SPRTC绑定的绑定信息

# 信令定义

信令中各个节点的解释同之前SESSION协议中的定义，下面只列出各个信令格式，具体SG和SPRTC的处理参考SESSION协议文档。

具体A的id为1111，B的id为2222。

## invite

|  |
| --- |
| <message type='notification' module='MID\_RTC'>  <invite id='guid' from='1111' to='2222'>  <audio recv='1' send='1'/>  <video recv='1' send='1'/>  </invite>  </message> |

1. invite元素

id属性唯一指明一次会话的id，A生成，需要保证全局唯一

from属性指明A的id

to属性指明B的id

1. audio元素(可选)

recv属性为1指明接收对方audio数据，为0指明不接收对方audio数据，send属性为1表示发送audio数据给对方，为0表示不发送audio数据给对方

1. video元素(可选)

recv属性为1指明接收对方video数据，为0指明不接收对方video数据，send属性为1表示发送video数据给对方，为0表示不发送video数据给对方

## ringing

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <ringing id='guid' from='1111' to='2222'>  </ringing>  </message> |

## ok

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <ok id='guid' from='1111' to='2222'>  <audio recv='1' send='1'/>  <video recv='1' send='1'/>  </ok>  </message> |

1. ok元素

id，from，to，属性同invite信令

1. audio元素

audio元素的属性参考invite信令，意义与之相同，针对的是B

begintime，开始时间

1. video元素

video元素的属性参考invite信令，意义与之相同，针对的是B

## reject

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <reject id='guid' from='1111' to='2222'>  <reason type=’normal/error/busy/unsupport/other’ desc=’just close’/>  </reject>  </message> |

1. reject元素

id，from，to，属性同invite信令

1. reason元素

reject的原因，type表示关闭的类型

normal：正常关闭

error：出错关闭

busy：忙线中

unsupport：不支持音视频关闭

other：其他关闭原因

desc为reject的描述

## modify

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <modify id='guid' from='1111' to='2222'>  <audio recv='0' send='1'/>  <video recv='0' send='1'/>  </modify>  </message> |

1. modify元素

id，from，to，属性同invite信令

## report

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <report id='guid' from='1111' to='2222'>  <undefined/>  </report>  </message> |

1. report元素

id，from，to，属性同invite信令

1. report子元素，该文档不做描述，由终端自行定义

## bye

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <bye id='guid' from='1111' to='2222'>  <reason type=’normal/error/busy/unsupport/other’ desc=’just close’/>  </bye>  </message> |

1. bye元素

id，from，to，属性同invite信令

1. reason元素

bye的原因，type表示关闭的类型

normal：正常关闭

error：出错关闭

busy：忙线中

unsupport：不支持音视频关闭

other：其他关闭原因

desc为bye的描述

## ack

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <ack id='guid' from='1111' to='2222'/>  </message> |

1. ack元素

id，from，to，属性同invite信令

## Offer/Answer

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <sdp id='guid' from='1111' to='2222'>  { "sdp" : "v=0\r\no=- 3467940624196228792 2 IN IP4 127.0.0.1\r\ns=-\r\nt=0 0\r\na=group:BUNDLE audio video\r\na=msid-semantic: WMS stream\_label\r\nm=audio 9 UDP/TLS/RTP/SAVPF 111 103 104 9 102 0 8 106 105 13 110 112 113 126\r\nc=IN IP4 0.0.0.0\r\na=rtcp:9 IN IP4 0.0.0.0\r\na=ice-ufrag:QEmK\r\na=ice-pwd:42hpjNDqaRjdDokT9q2ckACO\r\na=fingerprint:sha-256 F1:A9:7F:12:07:66:31:60:BA:EF:AC:84:9C:1E:FD:DD:5A:B7:72:42:56:B4:27:17:05:61:CD:89:21:B9:7E:B9\r\na=setup:actpass\r\na=mid:audio\r\na=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level\r\na=sendrecv\r\na=rtcp-mux\r\na=rtpmap:111 opus/48000/2\r\na=rtcp-fb:111 transport-cc\r\na=fmtp:111 minptime=10;useinbandfec=1\r\na=rtpmap:103 ISAC/16000\r\na=rtpmap:104 ISAC/32000\r\na=rtpmap:9 G722/8000\r\na=rtpmap:102 ILBC/8000\r\na=rtpmap:0 PCMU/8000\r\na=rtpmap:8 PCMA/8000\r\na=rtpmap:106 CN/32000\r\na=rtpmap:105 CN/16000\r\na=rtpmap:13 CN/8000\r\na=rtpmap:110 telephone-event/48000\r\na=rtpmap:112 telephone-event/32000\r\na=rtpmap:113 telephone-event/16000\r\na=rtpmap:126 telephone-event/8000\r\na=ssrc:1815346221 cname:dtS23U8Cznnwfmej\r\na=ssrc:1815346221 msid:stream\_label audio\_label\r\na=ssrc:1815346221 mslabel:stream\_label\r\na=ssrc:1815346221 label:audio\_label\r\nm=video 9 UDP/TLS/RTP/SAVPF 96 98 100 127 125 97 99 101 124\r\nc=IN IP4 0.0.0.0\r\na=rtcp:9 IN IP4 0.0.0.0\r\na=ice-ufrag:QEmK\r\na=ice-pwd:42hpjNDqaRjdDokT9q2ckACO\r\na=fingerprint:sha-256 F1:A9:7F:12:07:66:31:60:BA:EF:AC:84:9C:1E:FD:DD:5A:B7:72:42:56:B4:27:17:05:61:CD:89:21:B9:7E:B9\r\na=setup:actpass\r\na=mid:video\r\na=extmap:2 urn:ietf:params:rtp-hdrext:toffset\r\na=extmap:3 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time\r\na=extmap:4 urn:3gpp:video-orientation\r\na=extmap:5 http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01\r\na=extmap:6 http://www.webrtc.org/experiments/rtp-hdrext/playout-delay\r\na=sendrecv\r\na=rtcp-mux\r\na=rtcp-rsize\r\na=rtpmap:96 H264/90000\r\na=rtcp-fb:96 ccm fir\r\na=rtcp-fb:96 nack\r\na=rtcp-fb:96 nack pli\r\na=rtcp-fb:96 goog-remb\r\na=rtcp-fb:96 transport-cc\r\na=fmtp:96 level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=42e01f\r\na=rtpmap:98 VP8/90000\r\na=rtcp-fb:98 ccm fir\r\na=rtcp-fb:98 nack\r\na=rtcp-fb:98 nack pli\r\na=rtcp-fb:98 goog-remb\r\na=rtcp-fb:98 transport-cc\r\na=rtpmap:100 VP9/90000\r\na=rtcp-fb:100 ccm fir\r\na=rtcp-fb:100 nack\r\na=rtcp-fb:100 nack pli\r\na=rtcp-fb:100 goog-remb\r\na=rtcp-fb:100 transport-cc\r\na=rtpmap:127 red/90000\r\na=rtpmap:125 ulpfec/90000\r\na=rtpmap:97 rtx/90000\r\na=fmtp:97 apt=96\r\na=rtpmap:99 rtx/90000\r\na=fmtp:99 apt=98\r\na=rtpmap:101 rtx/90000\r\na=fmtp:101 apt=100\r\na=rtpmap:124 rtx/90000\r\na=fmtp:124 apt=127\r\na=ssrc-group:FID 3010146001 1841924587\r\na=ssrc:3010146001 cname:dtS23U8Cznnwfmej\r\na=ssrc:3010146001 msid:stream\_label video\_label\r\na=ssrc:3010146001 mslabel:stream\_label\r\na=ssrc:3010146001 label:video\_label\r\na=ssrc:1841924587 cname:dtS23U8Cznnwfmej\r\na=ssrc:1841924587 msid:stream\_label video\_label\r\na=ssrc:1841924587 mslabel:stream\_label\r\na=ssrc:1841924587 label:video\_label\r\n", "type" : "offer" }  </sdp>  </message> |

1. sdp元素

id，from，to，属性同invite信令

1. sdp的内容为json数据，在type属性中指明了是offer还是answer

## ICE Candidate

|  |
| --- |
| <message type='notification' module='MID\_RTC' to='sps1'>  <ice\_candidate id='guid' from='1111' to='2222'>  { "candidate" : "candidate:3303326613 1 udp 2122260223 192.168.10.70 63209 typ host generation 0 ufrag QEmK network-id 1 network-cost 50", "id" : "audio", "label" : 0, "type" : "candidate" }  </ice\_candidate>  </message> |

1. ice\_candidate元素

id，from，to，属性同invite信令

1. ice\_candidate的内容为json数据

## notify

其他端对音视频请求做了操作，会收到音视频的推送notify信令。fullto表示已在哪个端做了处理。

|  |
| --- |
| <message module="MID\_RTC" from="rtc:sprtc\_82\_1/" type="notification" to="1111">  <notify from="2222" id="3f59e1e9-5bd0-491b-a0e6-4d277abd1516" to="1111" fullto="1111/phone" action="accept"/>  </message> |