

昵称：孤竹君
园龄：6年11个月
粉丝：35
关注：0
+加关注

2016年3月						
日	一	二	三	四	五	六
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看

- 常用链接
- 我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

- 随笔分类(106)
- ACE(11)

ATL/WTL(6)

C++(14)

chromium(10)

COM(2)

IE Program(7)

Miscellaneous(4)

Net Program(2)

Shell(2)

SQL Server(2)

VC++(29)

WebRTC(6)

代码封装(5)

设计模式(2)

心情日记(4)

- 随笔档案(85)
- 2015年4月 (2)

2015年3月 (4)

2014年9月 (1)

2014年8月 (1)

2014年6月 (4)

2013年11月 (1)

2013年5月 (1)

2012年11月 (1)

2010年11月 (1)

2010年10月 (1)

2010年5月 (1)

2010年4月 (1)

2010年3月 (1)

2009年12月 (2)

2009年11月 (2)

2009年10月 (2)

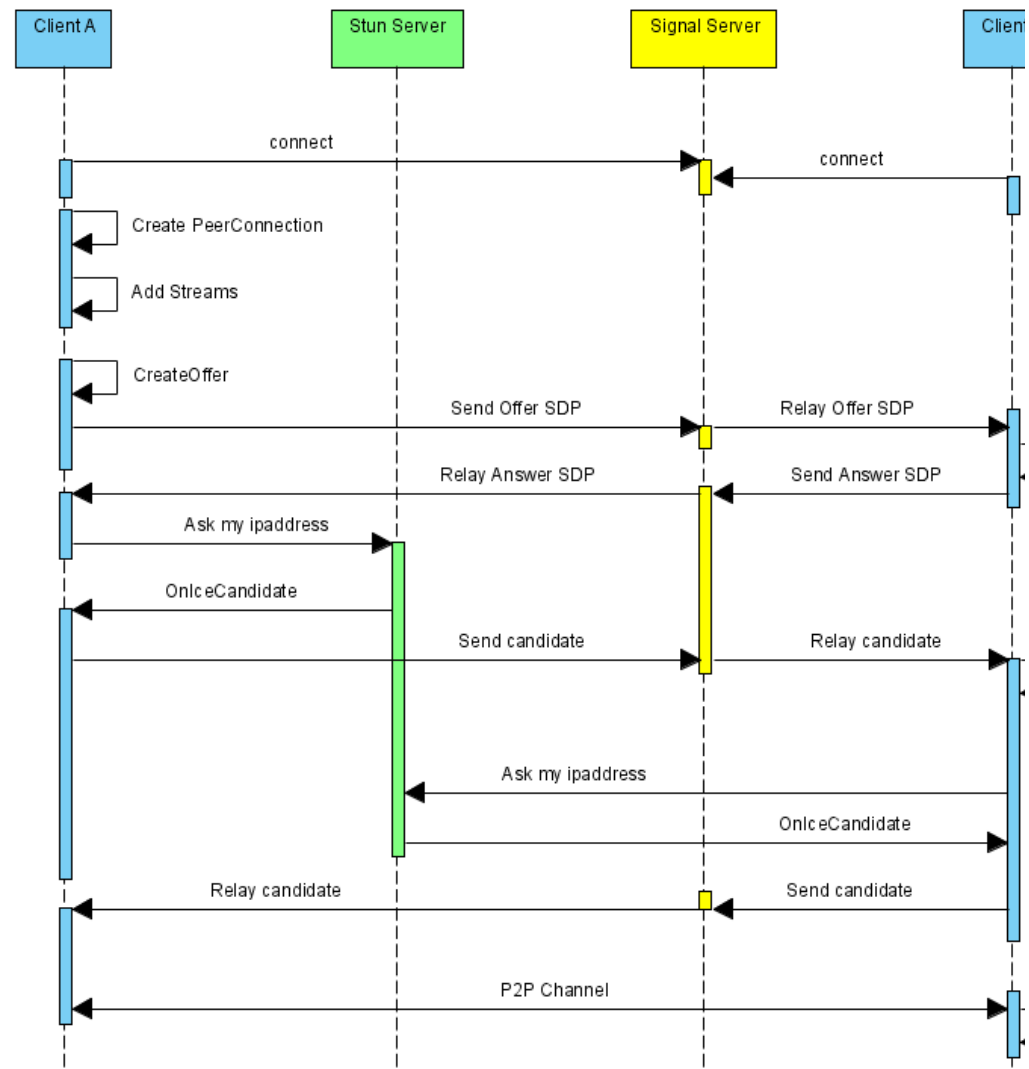
2009年8月 (6)

2009年7月 (8)

WebRTC手记之初探

转载请注明出处:<http://www.cnblogs.com/fangkm/p/4364553.html>

WebRTC是HTML5支持的重要特性之一，有了它，不再需要借助音视频相关的客户端，直接通过浏览器的Web页面就可以实现音视频对聊功能。而且WebRTC项目是开源的，我们可以借助WebRTC源码快速构建自己的音视频对聊功能。无论是使用前端JS的WebRTC API接口，还是在WebRTC源码上构建自己的对聊框架，都需要遵循以下执行流程：



上述序列中，WebRTC并不提供Stun服务器和Signal服务器，服务器端需要自己实现。Stun服务器可以用google提供的实现stun协议的测试服务器（<stun:stun.l.google.com:19302>），Signal服务器则完全需要自己实现了，它需要在ClientA和ClientB之间传送彼此的SDP信息和candidate信息，ClientA和ClientB通过这些信息建立P2P连接来传送音视频数据。由于网络环境的复杂性，并不是所有的客户端之间都能够建立P2P连接，这种情况下就需要有个relay服务器做音视频数据的中转，本文本着源码剖析的态度，这种情况就不考虑了。这里说明一下，stun/turn、relay服务器的实现在WebRTC源码中都有示例，真是名副其实的大宝库。

上述序列中，标注的场景是ClientA向ClientB发起对聊请求，调用描述如下：

2009年6月 (6)
2009年5月 (11)
2009年4月 (16)
2009年3月 (12)

文章分类(2)

收藏文章(2)

文章档案(2)

2009年4月 (2)

最新评论

1. Re:WebRTC手记之WebRtcVideoEngine2模块

请教一下看源码的方法，源码没有工程，我不知道怎么才能运行起来

--于光远

2. Re:WebRTC手记之框架与接口

请教一下为什么这个client代码工程有多重定义，我给相关的头文件加了

#pragma once
还是不能解决问题，编译peerconnection_server,没有问题

--于光远

3. Re:WebRTC手记之框架与接口

--于光远

4. Re:WebRTC手记之本地视频采集

首先感谢楼主分享的这几篇博文，写得真的很好！@Roisterr我也遇到过和你一样的错误提示信息。不过，我的情况是摄像头正常打开并采集视频信号，然后正常播放。只是在退出的时候，因为没有clean up.....

--johnson04

5. Re:WebRTC手记之框架与接口

写的不错，这个图你用什么工具绘制的？

--oldmtn

- ClientA首先创建PeerConnection对象，然后打开本地音视频设备，将音视频数据封装成MediaStream添加到PeerConnection中。
- ClientA调用PeerConnection的CreateOffer方法创建一个用于offer的SDP对象，SDP对象中保存当前音视频的相关参数。ClientA通过PeerConnection的SetLocalDescription方法将该SDP对象保存起来，并通过Signal服务器发送给ClientB。
- ClientB接收到ClientA发送过的offer SDP对象，通过PeerConnection的SetRemoteDescription方法将其保存起来，并调用PeerConnection的CreateAnswer方法创建一个应答的SDP对象，通过PeerConnection的SetLocalDescription的方法保存该应答SDP对象并将它通过Signal服务器发送给ClientA。
- ClientA接收到ClientB发送过来的应答SDP对象，将其通过PeerConnection的SetRemoteDescription方法保存起来。
- 在SDP信息的offer/answer流程中，ClientA和ClientB已经根据SDP信息创建好相应的音频Channel和视频Channel并开启Candidate数据的收集，Candidate数据可以简单地理解成Client端的IP地址信息（本地IP地址、公网IP地址、Relay服务端分配的地址）。
- 当ClientA收集到Candidate信息后，PeerConnection会通过OnIceCandidate接口给ClientA发送通知，ClientA将收到的Candidate信息通过Signal服务器发送给ClientB，ClientB通过PeerConnection的AddIceCandidate方法保存起来。同样的操作ClientB对ClientA再来一次。
- 这样ClientA和ClientB就已经建立了音视频传输的P2P通道，ClientB接收到ClientA传过来的音视频流，会通过PeerConnection的OnAddStream回调接口返回一个标识ClientA端音视频流的MediaStream对象，在ClientB端渲染出来即可。同样操作也适应ClientB到ClientA的音视频流的传输。

这里的流程仅仅是从使用层面上描述了一下，具体内部都做了什么、怎么做的，以后的文章中会慢慢细扒，万事开头难，自我鼓励一下。

分类: [WebRTC](#)

好文要顶

关注我

收藏该文



孤竹君
关注 - 0
粉丝 - 35

[+加关注](#)

3

0

(请您对文章做出评价)

« 上一篇: [Chromium的GPU进程启动流程](#)

» 下一篇: [WebRTC手记之框架与接口](#)

posted @ 2015-03-25 00:53 孤竹君 阅读(1942) 评论(11) 编辑 收藏

评论列表

#1楼 2015-03-25 08:07 sunlovesea

持续关注中....

支持(0) 反对(0)

#2楼 2015-03-25 08:07 睿德泽林

加油，希望能写成系列文章

支持(0) 反对(0)

#3楼 2015-03-25 08:24 KIUU

WebRTC 现在下载不下来，楼主下载下来了吗？完整下载的大小有8G多吧？

支持(0) 反对(0)