

[home \(/\)](#)[feed \(/timeline\)](#)[javascript \(/t/javascript\)](#)[php \(/t/php\)](#)[python \(/t/python\)](#)[\(/user/  
tags\)](#)

## 文 使用WebRTC搭建前端视频聊天室——入门篇 (/a/1190000000436544)

[webrtc \(/t/webrtc/blogs\)](#)[websocket \(/t/websocket/blogs\)](#)[node.js \(/t/node.js/blogs\)](#)[javascript \(/t/javascript/blogs\)](#)[webim \(/t/webim/blogs\)](#)**天壤** (/u/lingyuCoder) 2014年03月15日发布

### 什么是WebRTC？

众所周知，浏览器本身不支持相互之间直接建立信道进行通信，都是通过服务器进行中转。比如现在有两个客户端，甲和乙，他们俩想要通信，首先需要甲和服务端、乙和服务端之间建立信道。甲给乙发送消息时，甲先将消息发送到服务器上，服务器对甲的消息进行中转，发送到乙处，反过来也是一样。这样甲与乙之间的一次消息要通过两段信道，通信的效率同时受制于这两段信道的带宽。同时这样的信道并不适合数据流的传输，如何建立浏览器之间的点对点传输，一直困扰着开发者。WebRTC应运而生

WebRTC是一个开源项目，旨在使得浏览器能为实时通信（RTC）提供简单的JavaScript接口。说的简单明了一点就是让浏览器提供JS的即时通信接口。这个接口所创立的信道并不是像WebSocket一样，打通一个浏览器与WebSocket服务器之间的通信，而是通过一系列的信令，建立一个浏览器与浏览器之间（peer-to-peer）的信道，这个信道可以发送任何数据，而不需要经过服务器。并且WebRTC通过实现MediaStream，通过浏览器调用设备的摄像头、话筒，使得浏览器之间可以传递音频和视频

### WebRTC已经在我们的浏览器中

这么好的功能，各大浏览器厂商自然不会置之不理。现在WebRTC已经可以在较新版的Chrome、Opera和Firefox中使用了，著名的浏览器兼容性查询网站caniuse上给出了一份详尽的浏览器兼容情况

# WebRTC Peer-to-peer connections - Working Draft

\*Usage stats:

Global

Support:

50.47%

Method of allowing two users to communicate directly, browser to browser using the `RTCPeerConnection` API.

Show all versions

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	IE Mobile
								2.1		
								2.2		
						3.2		2.3		
						4.0-4.1		3.0		
						4.2-4.3		4.0		
	8.0					5.0-5.1		4.1		
	9.0		31.0	webkit		6.0-6.1		4.2-4.3	7.0	
	10.0	26.0	moz	32.0	webkit					
Current	11.0	27.0	moz	33.0	webkit	7.0	19.0	webkit	7.0	5.0-7.0
Near future		28.0	moz	34.0	webkit		20.0	webkit		
Farther future		29.0	moz	35.0	webkit		21.0	webkit		
3 versions ahead		30.0	moz	36.0	webkit					

Notes

Known issues (0)

Resources (2)

Feedback

Edit on GitHub

BlackBerry 10 recognizes `RTCPeerConnection` but real support is unconfirmed.

另外根据36Kr前段时间的新闻Google推出支持WebRTC及Web Audio的Androi... 和Android版Opera开始支持WebRTC，允许用户在...，Android也开始支持WebRTC

## 三个接口

- WebRTC实现了三个API，分别是：
- \* `MediaStream`：通过`MediaStream`的API能够通过设备的摄像头及话筒获得视频、音频的同步流
  - \* `RTCPeerConnection`：`RTCPeerConnection`是WebRTC用于构建点对点之间稳定、高效的流传输的组件
  - \* `RTCDataChannel`：`RTCDataChannel`使得浏览器之间（点对点）建立一个高吞吐量、低延时的信道，用于传输任意数据

这里大致上介绍一下这三个API

## MediaStream ( getUserMedia )

`MediaStream` API为WebRTC提供了从设备的摄像头、话筒获取视频、音频流数据的功能

## W3C标准

W3C标准传送门 ([http://dev.w3.org/2011/webrtc/...](http://dev.w3.org/2011/webrtc/))

## 如何调用

- 同门可以通过调用`navigator.getUserMedia()`，这个方法接受三个参数：
1. 一个约束对象（`constraints object`），这个后面会单独讲
  2. 一个调用成功的回调函数，如果调用成功，传递给它一个流对象
  3. 一个调用失败的回调函数，如果调用失败，传递给它一个错误对象

## 浏览器兼容性

由于浏览器实现不同，他们经常会在实现标准版本之前，在方法前面加上前缀，所以一个兼容版本就像这样

```
var getUserMedia = (navigator.getUserMedia ||
                    navigator.webkitGetUserMedia ||
                    navigator.mozGetUserMedia ||
                    navigator.msGetUserMedia);
```

## 一个超级简单的例子

这里写一个超级简单的例子，用来展现getUserMedia的效果：

```
</neaa>
<body>
  <video id="video" autoplay></video>
</body>

<script type="text/javascript">
  var getUserMedia = (navigator.getUserMedia || navigator.webkitGetUserMedia || navigator.moz

  getUserMedia.call(navigator, {
    video: true,
    audio: true
  }, function(localMediaStream) {
    var video = document.getElementById('video');
    video.src = window.URL.createObjectURL(localMediaStream);
    video.onloadedmetadata = function(e) {
      console.log("Label: " + localMediaStream.label);
      console.log("AudioTracks" , localMediaStream.getAudioTracks());
      console.log("VideoTracks" , localMediaStream.getVideoTracks());
    };
  }, function(e) {
    console.log('Reeejected!', e);
  });
</script>

</html>
```

将这段内容保存在一个HTML文件中，放在服务器上。用较新版本的Opera、Firefox、Chrome打开，在浏览器弹出询问是否允许访问摄像头和话筒，选同意，浏览器上就会出现摄像头所拍摄到的画面了

注意，HTML文件要放在服务器上，否则会得到一个NavigatorUserMediaError的错误，显示PermissionDeniedError，最简单方法就是cd到HTML文件所在目录下，然后 `python -m SimpleHTTPServer`（装了python的话），然后在浏览器中输入 `http://localhost:8000/{文件名称}.html`

这里使用 `getUserMedia` 获得流之后，需要将其输出，一般是绑定到 `video` 标签上输出，需要使用 `window.URL.createObjectURL(localMediaStream)` 来创造能在 `video` 中使用 `src` 属性播放的Blob URL，注意在 `video` 上加入 `autoplay` 属性，否则只能捕获到一张图片

流创建完毕后可以通过 `label` 属性来获得其唯一的标识，还可以通过 `getAudioTracks()` 和 `getVideoTracks()` 方法来获得流的追踪对象数组（如果没有开启某种流，它的追踪对象数组将是一个空数组）

## 约束对象（Constraints）

约束对象可以被设置在 `getUserMedia()` 和 `RTCPeerConnection` 的 `addStream` 方法中，这个约束对象是WebRTC用来指定接受什么样的流的，其中可以定义如下属性：

- \* `video`: 是否接受视频流
- \* `audio`: 是否接受音频流
- \* `MinWidth`: 视频流的最小宽度
- \* `MaxWidth`: 视频流的最大宽度
- \* `MinHeight`: 视频流的最小高度
- \* `MaxHeight`: 视频流的最大高度
- \* `MinAspectRatio`: 视频流的最小宽高比
- \* `MaxAspectRatio`: 视频流的最大宽高比
- \* `MinFramerate`: 视频流的最小帧速率
- \* `MaxFramerate`: 视频流的最大帧速率

详情见Resolution Constraints in Web Real Time Comm...

## RTCPeerConnection

WebRTC使用`RTCPeerConnection`来在浏览器之间传递流数据，这个流数据通道是点对点的，不需要经过服务器进行中转。但是这并不意味着我们能抛弃服务器，我们仍然需要它来为我们传递信令（`signaling`）来建立这个信道。WebRTC没有定义用于建立信道的信令的协议：信令并不是`RTCPeerConnection` API的一部分

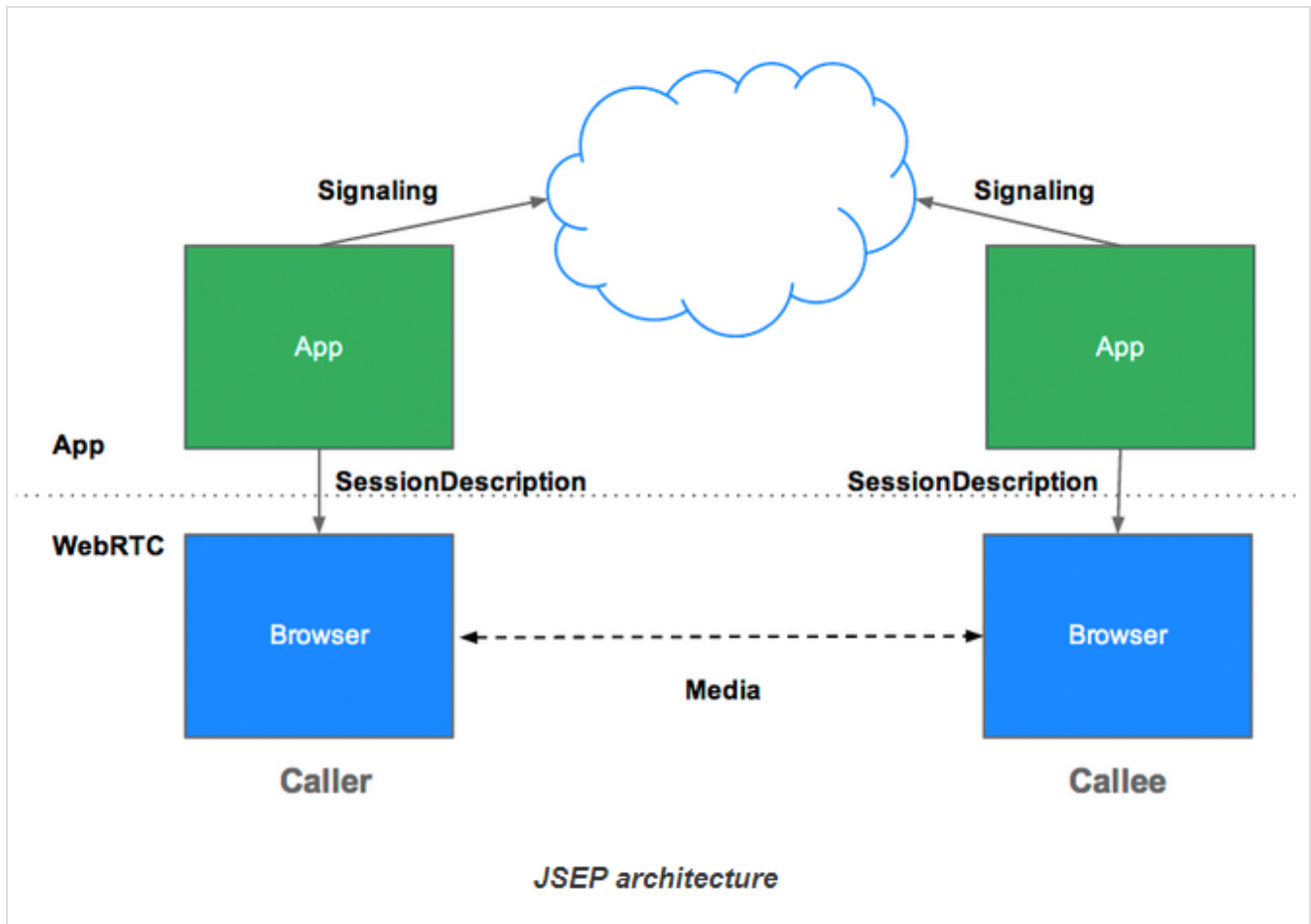
## 信令

既然没有定义具体的信令的协议，我们就可以选择任意方式（`AJAX`、`WebSocket`），采用任意的协议（`SIP`、`XMPP`）来传递信令，建立信道，比如我写的demo (<https://github.com/LingyuCoder/SkyRTC-...>，就是用的`node`的`ws`模块，在`WebSocket`上传递信令

需要信令来交换的信息有三种：

- \* `session`的信息：用来初始化通信还有报错
- \* 网络配置：比如IP地址和端口啥的
- \* 媒体适配：发送方和接收方的浏览器能够接受什么样的编码器和分辨率

这些信息的交换应该在点对点的流传输之前就全部完成，一个大致的架构图如下：



## 通过服务器建立信道

这里再次重申，就算WebRTC提供浏览器之间的点对点信道进行数据传输，但是建立这个信道，必须有服务器的参与。WebRTC需要服务器对其进行四方面的功能支持：

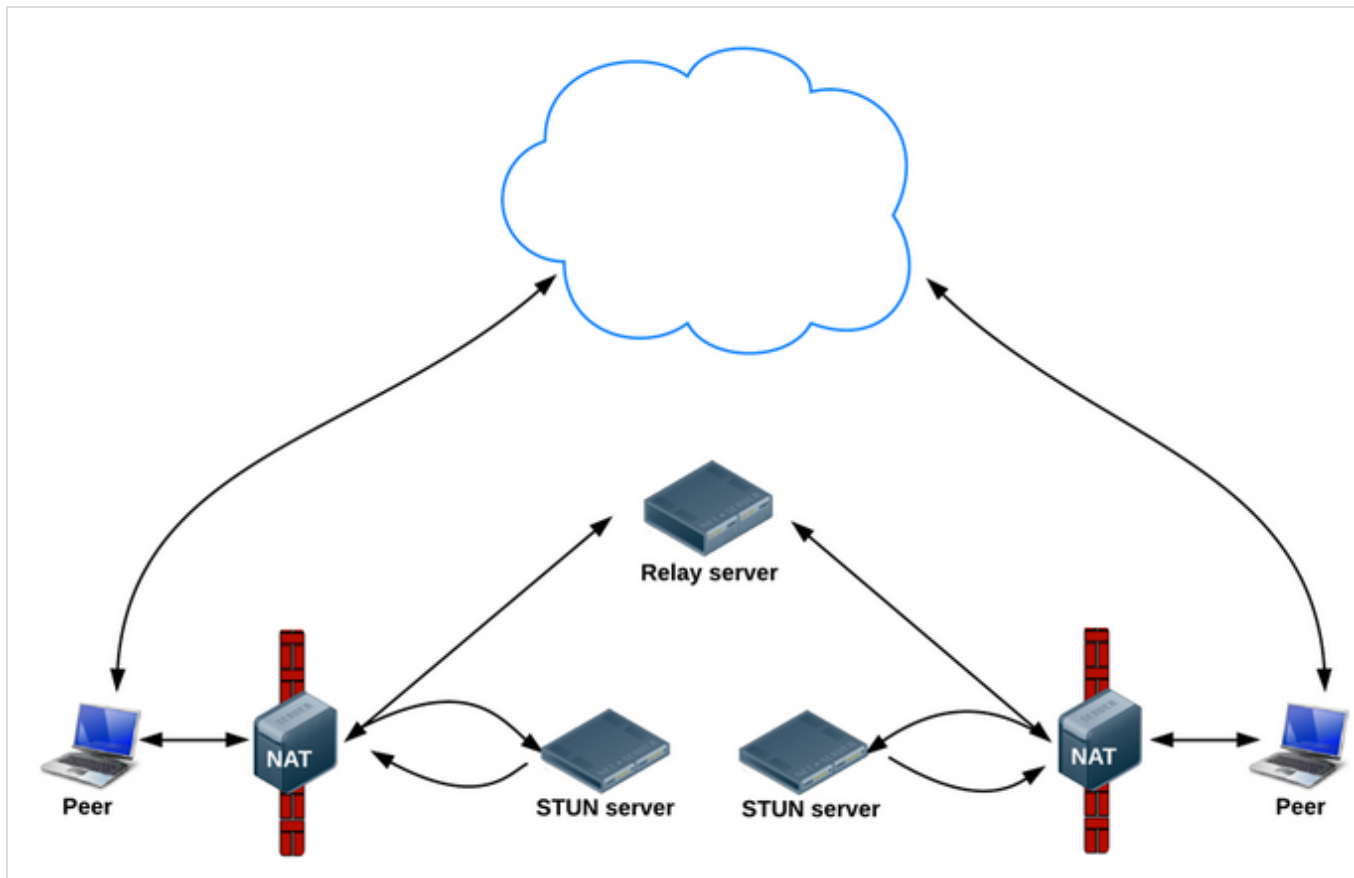
1. 用户发现以及通信
2. 信令传输
3. NAT/防火墙穿越
4. 如果点对点通信建立失败，可以作为中转服务器

## NAT/防火墙穿越技术

建立点对点信道的一个常见问题，就是NAT穿越技术。在处于使用了NAT设备的私有TCP/IP网络中的主机之间需要建立连接时需要使用NAT穿越技术。以往在VoIP领域经常会遇到这个问题。目前已经有很多NAT穿越技术，但没有一项是完美的，因为NAT的行为是非标准化的。这些技术中大多使用了一个公共服务器，这个服务使用了一个从全球任何地方都能访问得到的IP地址。在RTCPeerConnection中，使用ICE框架来保证RTCPeerConnection能实现NAT穿越

ICE，全名叫交互式连接建立（Interactive Connectivity Establishment），一种综合性的NAT穿越技术，它是一种框架，可以整合各种NAT穿越技术如STUN、TURN（Traversal Using Relay NAT 中继NAT实现的穿透）。ICE会先使用STUN，尝试建立一个基于UDP的连接，如果失败了，就会去TCP（先尝试HTTP，然后尝试HTTPS），如果依旧失败ICE就会使用一个中继的TURN服务器。

我们可以使用Google的STUN服务器：stun:stun.l.google.com:19302，于是乎，一个整合了ICE框架的架构应该长这个样子



## 浏览器兼容

还是前缀不同的问题，采用和上面类似的方法：

```
var PeerConnection = (window.PeerConnection ||  
    window.webkitPeerConnection00 ||  
    window.webkitRTCPeerConnection ||  
    window.mozRTCPeerConnection);
```

## 创建和使用

```
        "sdp": desc
    });
    });
    //绑定本地媒体流到video标签用于输出
    myselfVideoElement.src = URL.createObjectURL(stream);
    //向PeerConnection中加入需要发送的流
    pc.addStream(stream);
    //如果是发送方则发送一个offer信令，否则发送一个answer信令
    if(isCaller){
        pc.createOffer(sendOfferFn);
    } else {
        pc.createAnswer(sendAnswerFn);
    }
}, function(error){
    //处理媒体流创建失败错误
});
//处理到来的信令
socket.onmessage = function(event){
    var json = JSON.parse(event.data);
    //如果是一个ICE的候选，则将其加入到PeerConnection中，否则设定对方的session描述为传递过来的描述
    if( json.event === "__ice_candidate" ){
        pc.addIceCandidate(new RTCIceCandidate(json.data.candidate));
    } else {
        pc.setRemoteDescription(new RTCSessionDescription(json.data.sdp));
    }
};
```

## 实例

由于涉及较为复杂灵活的信令传输，故这里不做简短的实例，可以直接移步到最后

## RTCDataChannel

既然能建立点对点的信道来传递实时的视频、音频数据流，为什么不能用这个信道传一点其他数据呢？

RTCDataChannel API就是用来干这个的，基于它我们可以在浏览器之间传输任意数据。DataChannel是建立在PeerConnection上的，不能单独使用

## 使用DataChannel

我们可以使用 `channel = pc.createDataChannel("someLabel")`；来在PeerConnection的实例上创建DataChannel，并给与它一个标签

DataChannel使用方式几乎和WebSocket一样，有几个事件：

- \* onopen
- \* onclose
- \* onmessage
- \* onerror

同时它有几个状态，可以通过 `readyState` 获取：

- \* `connecting`: 浏览器之间正在试图建立channel
- \* `open` : 建立成功，可以使用 `send` 方法发送数据了
- \* `closing` : 浏览器正在关闭channel
- \* `closed` : channel已经被关闭了

两个暴露的方法:

- \* `close()`: 用于关闭channel
- \* `send()` : 用于通过channel向对方发送数据

## 通过Data Channel发送文件大致思路

JavaScript已经提供了File API从 `input[type='file']` 的元素中提取文件，并通过FileReader来将文件的转换成DataURL，这也意味着我们可以将DataURL分成多个碎片来通过Channel来进行文件传输

## 一个综合的Demo

SkyRTC-demo (<https://github.com/LingyuCoder/...>，这是我写的一个Demo。建立一个视频聊天室，并能够广播文件，当然也支持单对单文件传输，写得还很粗糙，后期会继续完善

## 使用方式

1. 下载解压并cd到目录下
2. 运行 `npm install` 安装依赖的库 ( `express`, `ws`, `node-uuid` )
3. 运行 `node server.js` , 访问 `localhost:3000` , 允许摄像头访问
4. 打开另一台电脑，在浏览器 ( `Chrome`和`Opera` , 还未兼容`Firefox` ) 打开 `{server所在IP}:3000` , 允许摄像头和话筒访问
5. 广播文件：在左下角选定一个文件，点击“发送文件”按钮
6. 广播信息：左下角input框输入信息，点击发送
7. 可能会出错，注意F12对话框，一般F5能解决

## 功能

视频音频聊天 ( 连接了摄像头和话筒，至少要有摄像头 )，广播文件 ( 可单独传播，提供API，广播就是基于单独传播实现的，可同时传播多个，小文件还好说，大文件坐等内存吃光 )，广播聊天信息

## 参考资料

- WebRTC官方网站 (<http://www.webrtc.org/>)
- W3C-GetUserMedia (<http://dev.w3.org/2011/web...>)
- W3C-WebRTC (<http://dev.w3.org/2011/webrtc/e...>)



- Capturing Audio & Video in HTML5@html5rocks ...
- Getting Started with WebRTC@html5rocks (http:...
- caniuse (http://caniuse.com)
- ICE交互式连接建立 (http://zh.wikipedia.org/wiki/...

2014年03月15日发布 (/a/1190000000436544)

4 推荐

收藏

## 你可能感兴趣的文章

使用WebRTC搭建前端视频聊天室——点对点通信篇 (/a/1190000000733774) 25 收藏, 5.5k 浏览

WebRTC 工作流程 (/a/1190000000608413) 1 收藏, 1.9k 浏览

使用WebRTC搭建前端视频聊天室——数据通道篇 (/a/1190000000733779) 25 收藏, 9.3k 浏览

## 讨论区

+1

另外根据35Kr前段时间的新闻

s/35/36/

用教新版本的Opera、Firefox、Chrome打开

s/教/较/

audio : 时候接受音频流

s/时候/是否/

但是这并不意味只我们能抛弃服务器

s/只/着/

(/c/1050000000438054) **weakish** (/u/weakish) · 2014年03月17日

回复 weakish (/u/weakish) :

非常感谢！我在提交之前还读了一遍都没发现还有这么多错别字

(/c/1050000000438219) **天壤** (/u/lingyucoder) · 2014年03月17日

回复 天镶 (/u/lingyucoder) :

不客气。我自己写博客也常有漏掉的错别字，猜想是自己写的东西自己读的时候大脑会偷懒，看到一些「关键字」就会唤起相应语句的记忆，然后以为已经确认过了，其实没有确认。

(/c/1050000000438220) **weakish** (/u/weakish) · 2014年03月17日

请问一下博主

“HTML文件要放在服务器上，否则会得到一个NavigatorUserMediaError的错误，显示PermissionDeniedError” 是为什么？

(/c/1050000000445357) **SylvainWEN** (/u/sylvainwen) · 2014年03月25日

回复 SylvainWEN (/u/sylvainwen) :

这个错误本来是用户禁止浏览器获取摄像头数据。但当运行本地文件时，浏览器不会询问用户是否允许，而是直接抛出错误，这是浏览器自身的设定

(/c/1050000000445375) **天镶** (/u/lingyucoder) · 2014年03月25日

回复 天镶 (/u/lingyucoder) :

感谢博主！昨天折腾了一整天在这里找到了答案。再请问一下博主是怎么发现浏览器的这个设定从而找到部署到服务器上的方法的。

(/c/1050000000445490) **SylvainWEN** (/u/sylvainwen) · 2014年03月25日

回复 SylvainWEN (/u/sylvainwen) :

我也是遇到了，报错之后查了下stackoverflow

(/c/1050000000445698) **天镶** (/u/lingyucoder) · 2014年03月26日

回复 天镶 (/u/lingyucoder) :

再次谢过楼主。受教了！

(/c/1050000000446884) **SylvainWEN** (/u/sylvainwen) · 2014年03月27日

你好，我最近也搭建了一个聊天室，在内网中，所有功能都实现，但是把它部署到服务器后，pc就不通了，显示iceconnectionstate 是failed。PS：addstream应该要在createOffer之前吧。

(/c/1050000000515254) **v\_shinc** (/u/v\_shinc) · 2014年05月21日

回复 v\_shinc (/u/v\_shinc) :

嗯，这个问题我也遇到了，不过目前在学校没有条件解决，可以考虑换一个stun服务器。addstream的话，曾经看到html5rocks上的文章说前后都行，不过一般都是放在createOffer之前，我也没尝试过放在createOffer之后

(/c/1050000000515502) **天镶** (/u/lingyucoder) · 2014年05月21日

Chrome 打开直接提示 create stream failed

(/c/1050000000625260) **kk\_625259** (/u/iakuf) · 2014年08月01日

回复 kk\_625259 (/u/iakuf) :

通常会弹出一个是否允许chrome获取摄像头，如果点不允许，就会有这个提示。没有摄像头也会有这个问题

(/c/1050000000626607) **天镶** (/u/lingyucoder) · 2014年08月03日

您好！请问这句打开另一台电脑，在浏览器（Chrome和Opera，还未兼容Firefox）打开{server所在IP}:3000，允许摄像头和话筒访问中的{server所在IP}:3000是什么意思？是我电脑的IP地址在别人电脑上打开并在后面加上：3000吗？

(/c/1050000000723669) **jixiao211** (/u/jixiao211) · 2014年10月15日

对的，因为你需要连接到那台起server的电脑

(/c/1050000000723688) **天镶** (/u/lingyucoder) · 2014年10月15日

回复 天镶 (/u/lingyucoder)：

我知道了，谢谢！

(/c/1050000000723823) **jixiao211** (/u/jixiao211) · 2014年10月15日

您好，请问我在自己电脑打开了，然后在另一台电脑浏览器打开后，但是只能显示出自己的视频窗口，而对方的视频窗口没有显示出来，请问这是什么原因？我是JavaScript初学者

(/c/1050000001801409) **Keep\_Fighting\_** (/u/keep\_fighting\_) · 2014年11月18日

回复 Keep\_Fighting\_ (/u/keep\_fighting\_)：

出现错误可能的原因很多，可以F12看一下控制台报什么错

(/c/1050000001815656) **天镶** (/u/lingyucoder) · 2014年11月18日

回复 天镶 (/u/lingyucoder)：

恩恩，找到了，，，谢了！

(/c/1050000001837446) **Keep\_Fighting\_** (/u/keep\_fighting\_) · 2014年11月18日

回复 天镶 (/u/lingyucoder)：

有时正常，有时出现ICE failed, see about:webrtc for more details这个错误，请问这是什么原因？

(/c/1050000001842229) **Keep\_Fighting\_** (/u/keep\_fighting\_) · 2014年11月18日

您好，我是在校学生正在学习WebRTC，但几乎不懂JavaScript，想先跑通试试效果。

我下载了SkyRTC-demo数据包，但不太会用。

我理解的是把index.html和server.js放在服务器端（服务器端先安装Node.js及npm环境），然后在DOS界面node server.js（服务器开始监听），再打开Chrome浏览器，

但“访问localhost:3000”不懂。求解？？？？？

SkyRTC-client.js放在客户端。然后再客户端打开Chrome浏览器，“打开{server所在IP}:3000”

也不理解。望解答，非常感谢！！！！

(/c/1050000002416222) **keny** (/u/keny) · 2014年12月10日

回复 Keep\_Fighting\_ (/u/keep\_fighting\_)：

这个应该是向ICE服务器请求无效了，换个靠谱的ICE服务器试试

(/c/1050000002416833) **天镶** (/u/lingyucoder) · 2014年12月10日

回复 kenny (/u/keny)：

起SkyRTC-demo的话，只需要在文件夹下npm install一下，然后node server.js，访问http://localhost:3000 (http://localhost:3000)就可以了。由于需要两台机子，所以另一台也需要访问，访问的方式就是通过将localhost换成你启动服务的IP

(/c/1050000002416837) **天壤** (/u/lingyucoder) · 2014年12月10日

回复 天壤 (/u/lingyucoder) :

非常感谢，我配置过node.js和npm环境，进入SkyRTC-demo-master文件夹后，npm install后显示npm WARN package.json SkyRTC-demo@ No repository field.

然后node server.js 显示module.js:340 throw error.....

请问博主能解惑么？

(/c/1050000002417929) **keny** (/u/keny) · 2014年12月10日

很好的文章，必须赞。

(/c/1050000002488073) **lijp** (/u/lijp) · 2015年01月15日

你好能加一下你的Q吗，我还是学生 有写问题先请教一下。 我的Q346357987

(/c/1050000002602648) **TangYee** (/u/tangye) · 2015年03月17日

我们团队也正在研究web RTC，目前已经实现web端的多人视频接入，下一步往移动端开发，如有感兴趣的个人或者团队，愿意的话 跟我联系38443950 QQ

(/c/1050000002715421) **stone0905** (/u/stone0905) · 2015年04月28日

您好，非常感谢您的文章。部署好环境后，电脑进入http://localhost:3000 (http://localhost:3000)后摄像头可以开启，通过另一台电脑进入后，两个聊天室好像独立的一般，发送的信息互相看不到。而且发送文件功能好像无法使用。请问是什么原因？谢谢

(/c/1050000002882359) **第一最寂寞** (/u/diyizuijimo) · 2015年06月06日

回复 v\_shinc (/u/v\_shinc) :

您好，我们最近刚刚开始评估使用WebRTC实现多人聊天，只在内网使用，您基于内网的例子能发给我看看么？多谢

(/c/1050000003042391) **songofhawk** (/u/songofhawk) · 2015年08月01日

楼主的文章写得真好，赞一个

(/c/1050000003042392) **songofhawk** (/u/songofhawk) · 2015年08月01日

请问可以支持手机么，我用手机测试一下不行，是方式不对呢还是目前不支持呢。

(/c/1050000003045889) **coderz** (/u/coderz) · 2015年08月03日

回复 coderz (/u/coderz) :

http://caniuse.com/#search=webrtc (http://caniuse.c... 这里有兼容性，移动端兼容性比较次

(/c/1050000003054310) **天壤** (/u/lingyucoder) · 2015年08月05日

楼主请教下：

如果想要获得getUserMedia取得的视频内容，服务端是要用类似PeerConnection的chrome webrtc native代码还是可以直接用RTP/RTCP协议监听某个端口，读取音视频信息？谢谢

(/c/1050000003490880) **dzm** (/u/dzm) · 2015年08月25日

回复 dzm (/u/dzm)：

服务端？webrtc的数据特点就是数据不走服务端，你这服务端是个啥？

(/c/1050000003492540) **天镶** (/u/lingyucoder) · 2015年08月25日

回复 天镶 (/u/lingyucoder)：

就是想把数据传回服务器保存

(/c/1050000003493110) **dzm** (/u/dzm) · 2015年08月25日

楼主，你好，刚开始跑你写的skyrtp-demo，局域网内两台电脑跑通了，可以视频语音发文件，但是手机端ios浏览器能够进入房间，出现creat stream failed错误，Android浏览器连房间都进不去，请问这是什么原因呢？

(/c/1050000003699899) **嘿巨蟹** (/u/heijuxie) · 2015年09月02日

楼主，请问下，把demo部署到服务器上后，同一个局域网里访问的话有视频，可如果是不同的网络去访问就看不到远程的视频了，这是不是stun服务器的问题（用的默认的stun.l.google.com:19302）。如果是的话，解决方案是自行搭建吗

(/c/1050000003720158) **三国lz** (/u/sanguolz) · 2015年09月08日

回复 嘿巨蟹 (/u/heijuxie)：

上caniuse看看webrtc在移动端的兼容性你就知道了

(/c/1050000003726035) **天镶** (/u/lingyucoder) · 2015年09月09日

回复 三国lz (/u/sanguolz)：

可以考虑自己搭建一个stun服务器

(/c/1050000003726037) **天镶** (/u/lingyucoder) · 2015年09月09日

回复 天镶 (/u/lingyucoder)：

谢楼主回复，ios试过chrome、Firefox、Safari、搜狗、遨游等浏览器依旧都是显示creat stream failed错误，Android在chrome和Firefox可以了

(/c/1050000003726151) **嘿巨蟹** (/u/heijuxie) · 2015年09月09日

回复 天镶 (/u/lingyucoder)：

并且Android在跑chrome、Firefox时，麦克有强烈刺耳噪音和回声，请问楼主怎么调整可以解决这些问题？

(/c/1050000003726163) **嘿巨蟹** (/u/heijuxie) · 2015年09月09日

回复 天镶 (/u/lingyucoder)：

楼主，谢谢回复。请问可以给一个QQ、微信之类的联系方式吗，我们现在自己摸索webrtc一段时间了，有些问题想请教一下

(/c/1050000003726380) **三国lz** (/u/sanguolz) · 2015年09月09日

楼主 在吗 多人聊天室已经实现了 在局域网上面可以用 部署到网络上 然后观看是黑屏的 视频的采集是正常的  
(/c/1050000003783826) **674897883** (/u/674897883) · 2015年09月23日

江湖救急 @天镶 (/u/skyinlayer)

(/c/1050000003783829) **674897883** (/u/674897883) · 2015年09月23日

回复 三国lz (/u/sanguolz) :

你解决问题了吗 你问楼主的问题

(/c/1050000003784040) **674897883** (/u/674897883) · 2015年09月23日

回复 674897883 (/u/674897883) :

应该是谷歌提供的stun服务器不行，换别的吧，或自己搭

(/c/1050000003784695) **三国lz** (/u/sanguolz) · 2015年09月23日

回复 三国lz (/u/sanguolz) :

你部署网络上实现效果了吗

(/c/1050000003789077) **674897883** (/u/674897883) · 2015年09月24日

回复 674897883 (/u/674897883) :

现在用的“实时猫”的sdk在开发，你可以试试看

(/c/1050000003790450) **三国lz** (/u/sanguolz) · 2015年09月24日

回复 distance (/u/distance) :

同求解 您弄明白了这是咋回事了吗？？

(/c/1050000003939232) **铁臂大龙虾** (/u/tiebidalongxia) · 2015年11月02日

回复 铁臂大龙虾 (/u/tiebidalongxia) :

暂时还没有，如果你有发现的话也请告知一下噢~

(/c/1050000003942269) **distance** (/u/distance) · 2015年11月02日

@天镶 (/u/skyinlayer)

skyRTC中的/public/SkyRTC-client.js，有很多监听服务器的事件，如：

```
this.on('_offer', function(data) {
```

```
    that.receiveOffer(data.socketId, data.sdp);
    that.emit("get_offer", data);
});
```

想问该事件后面“that.emit(“get\_offer”, data);”触发的是什么事？在代码中都没有找到，也没找到这个触发后会调用什么代码。

(/c/1050000003957720) **distance** (/u/distance) · 2015年11月06日


写的真好，正在找这方面的资料呢

(/c/1050000004292602) **kinglisky** (/u/kinglisky) · 1月12日

请先 登录 后评论

本文隶属于专栏

说学逗唱 (/blog/skyinlayer)

 天壤 (/u/lingyucoder)  
作者

关注专栏

系列文章

使用WebRTC搭建前端视频聊天室——点对点通信篇 (/a/1190000000733774) 25 收藏， 5.5k 浏览

使用WebRTC搭建前端视频聊天室——数据通道篇 (/a/1190000000733779) 25 收藏， 9.3k 浏览

分享扩散：

...