

09 | 让我们揭开WebRTC建立连接的神秘面纱

2019-08-03 李超

从0打造音视频直播系统

[进入课程 >](#)



讲述：李超

时长 17:51 大小 16.35M



在上一篇[《08 | 有话好商量，论媒体协商》](#)文章中，我向你介绍了 WebRTC 进行通信时，是如何进行媒体协商的，以及媒体协商的目的是什么。

在媒体协商过程中，如果双方能达成一致，也就是商量好了使用什么编解码器，确认了使用什么传输协议，那么接下来，WebRTC 就要建立连接，开始传输音视频数据了。

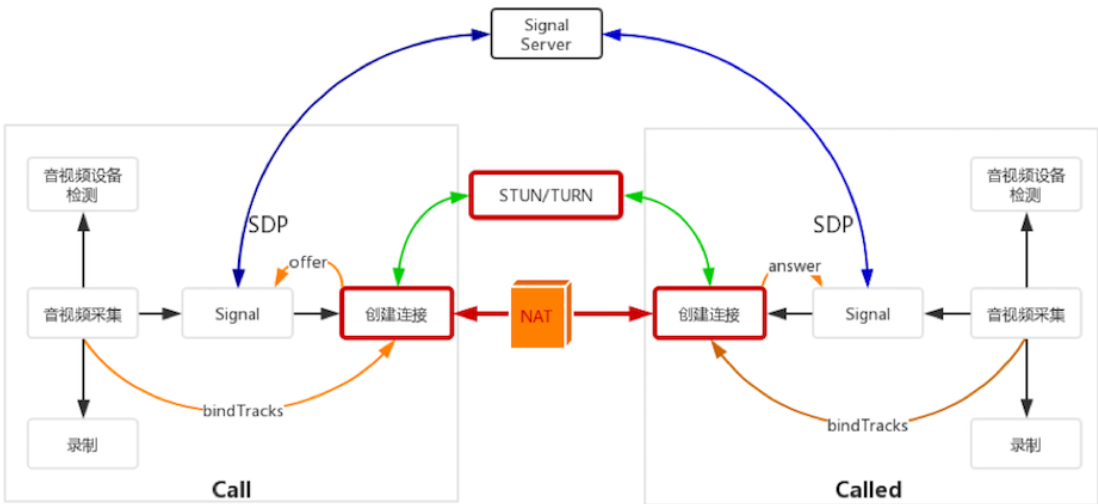
WebRTC 之间建立连接的过程是非常复杂的。之所以复杂，主要的原因在于它既要考虑传输的**高效性**，又要保证端与端之间的**连通率**。

换句话说，当同时存在多个有效连接时，它首先选择传输质量最好的线路，如能用内网连通就不用公网。另外，如果尝试了很多线路都连通不了，那么它还会使用服务端中继的方式让双方连通，总之，是“想尽办法，用尽手段”让双方连通。

对于**传输的效率与连通率**这一点，既是 WebRTC 的目标，也是 WebRTC 建立连接的基本策略。下面我们就来具体看一下 WebRTC 是如何达到这个目标的吧！

在 WebRTC 处理过程中的位置

下面这张图清晰地表达了本文所讲的内容在整个 WebRTC 处理过程中的位置。



WebRTC 处理过程图

图中的红色部分——连接的创建、STUN/TURN 以及 NAT 穿越，就是我们本文要讲的主要内容。

连接建立的基本原则

接下来，我将通过两个具体的场景，向你介绍一下 WebRTC 建立连接的基本原则。不过在讲解之前，我们先设置一些假设条件，这样会更有利于我们下面的描述：

- 通信的双方我们称为 A 和 B；
- A 为呼叫方，B 为被呼叫方；
- C 为中继服务器，也称为 relay 服务器或 TURN 服务器。

1. 场景一：双方处于同一网段内

A 与 B 进行通信，假设它们现在处于同一个办公区的同一个网段内。在这种情况下，A 与 B 有两种连通路经：

一种是双方通过内网直接进行连接；

另一种是通过公网，也就是通过公司的网关，从公网绕一圈后再进入公司实现双方的通信。

相较而言，显然第一种连接路径是最好的。A 与 B 在内网连接就好了，谁会舍近求远呢？

但现实却并非如此简单，要想让 A 与 B 直接在内网连接，首先要解决的问题是：A 与 B 如何才能知道它们是在同一个网段内呢？

这个问题还真不好回答，也正是由于这个问题不太好解决，所以，现在有很多通信类产品在双方通信时，无论是否在同一个内网，它们都统一走了公网。不过，WebRTC 很好的解决了这个问题，后面我们可以看一下它是如何解决这个问题的。

2. 场景二：双方处于不同点

A 与 B 进行通信，它们分别在不同的地点，比如一个在北京，一个在上海，此时 A 与 B 通信必须走公网。但走公网也有两条路径：

一是通过 P2P 的方式双方直接建立连接；

二是通过中继服务器进行中转，即 A 与 B 都先与 C 建立连接，当 A 向 B 发消息时，A 先将数据发给 C，然后 C 再转发给 B；同理，B 向 A 发消息时，B 先将消息发给 C，然后 C 再转给 A。

对于这两条路径你该如何选择呢？对于 WebRTC 来讲，它认为**通过中继的方式会增加 A 与 B 之间传输的时长，所以它优先使用 P2P 方式；如果 P2P 方式不通，才会使用中继的方式。**

通过上面两个场景的描述，我想你应该已经了解到 WebRTC 为了实现端与端之间连接的建立，做了非常多的工作。下面我们就来一起看看 WebRTC 建立连接的具体过程吧！

什么是 Candidate

在讲解 WebRTC 建立连接的过程之前，你有必要先了解一个基本概念，即 **ICE Candidate（ICE 候选者）**。它表示 WebRTC 与远端通信时使用的协议、IP 地址和端口，一般由以下字段组成：

本地 IP 地址

本地端口号

候选者类型，包括 host、srflx 和 relay

优先级

传输协议

访问服务的用户名

.....

如果用一个结构表示，那么它就如下面所示的样子：

 复制代码

```
1 {  
2     IP: xxx.xxx.xxx.xxx,  
3     port: number,  
4     type: host/srflx/relay,  
5     priority: number,  
6     protocol: UDP/TCP,  
7     usernameFragment: string  
8     ...  
9 }
```

其中，候选者类型中的 host 表示**本机候选者**，srflx 表示内网主机映射的**外网的地址和端口**，relay 表示**中继候选者**。

当 WebRTC 通信双方彼此要进行连接时，每一端都会提供许多候选者，比如你的主机有两块网卡，那么每块网卡的不同端口都是一个候选者。

WebRTC 会按照上面描述的格式对候选者进行排序，然后按优先级从高到低的顺序进行连通性测试，当连通性测试成功后，通信的双方就建立起了连接。

在众多候选者中，**host 类型的候选者优先级是最高的**。在 WebRTC 中，首先对 host 类型的候选者进行连通性检测，如果它们之间可以互通，则直接建立连接。其实，**host 类型之间的连通性检测就是内网之间的连通性检测**。WebRTC 就是通过这种方式巧妙地解决了大家认为很困难的问题。

同样的道理，如果 host 类型候选者之间无法建立连接，那么 WebRTC 则会尝试**次优先级的候选者，即 srflx 类型的候选者**。也就是尝试让通信双方直接通过 P2P 进行连接，如果连接成功就使用 **P2P 传输数据**；如果失败，就最后尝试使用 relay 方式建立连接。

通过上面的描述，你是不是觉得 WebRTC 在这里的设计相当精妙呢？当然在 WebRTC 看来，以上这些只不过是一些“皮毛”，在下一篇关于 NAT 穿越原理一文中，你还会看 WebRTC 在 NAT 穿越上的精彩处理。

收集 Candidate

了解了什么是 Candidate 之后，接下来，我们再来看一下端对端的连接是如何建立的吧。

实际上，端对端的建立更主要的工作是**Candidate 的收集**。WebRTC 将 Candidate 分为三种类型：

host 类型，即本机内网的 IP 和端口；

srflx 类型，即本机 NAT 映射后的外网的 IP 和端口；

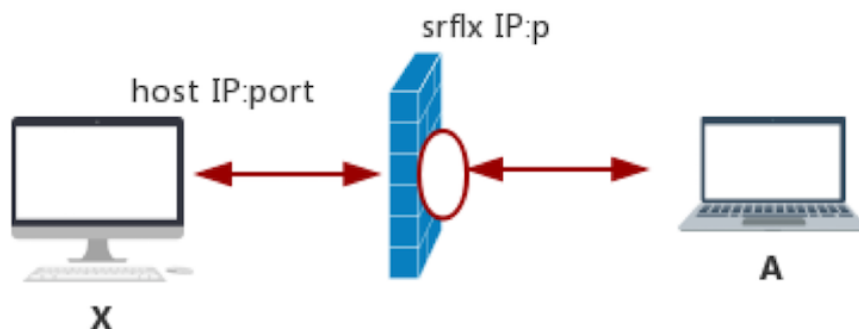
relay 类型，即中继服务器的 IP 和端口。

其中，host 类型优先级最高，srflx 次之，relay 最低（前面我们已经说明过了）。

在以上三种 Candidate 类型中，**host 类型的 Candidate 是最容易收集的**，因为它们都是本机的 IP 地址和端口。对于 host 类型的 Candidate 这里就不做过多讲解了，下面我们主要讲解一下 srflx 和 relay 这两种类型的 Candidate 的收集。

1. STUN 协议

srflx 类型的 Candidate 实际上就是内网地址和端口**经 NAT 映射**后的外网地址和端口。如下图所示：



NAT 地址映射图

你应该知道，如果主机没有公网地址，是无论如何都无法访问公网上的资源的。例如你要通过百度搜索一些信息，如果你的主机没有公网地址的话，百度搜索到的结果怎么传给你呢？

而一般情况下，主机都只有内网 IP 和端口，那它是如何访问外网资源的呢？实际上，在内网的网关上都有 NAT (Net Address Transport) 功能，**NAT 的作用就是进行内外网的地址转换**。这样当你要访问公网上的资源时，NAT 首先会将该主机的内网地址转换成外网地址，然后才会将请求发送给要访问的服务器；服务器处理好后将结果返回给主机的公网地址和端口，再通过 NAT 最终中转给内网的主机。

知道了上面的原理，你要想让内网主机获得它的外网 IP 地址也就好办了，只需要在公网上架设一台服务器，并向这台服务器发个请求说：“Hi！伙计，你看我是谁？”对方回：“你不是那 xxxx 吗？”这样你就可以知道自己的公网 IP 了，是不是很简单？

实际上，上面的描述已经被定义成了一套规范，即 RFC5389，也就是 **STUN 协议**，我们只要**遵守这个协议就可以拿到自己的公网 IP 了**。

这里我们举个例子，看看通过 STUN 协议，主机是如何获取到自己的外网 IP 地址和端口的。

首先在外网搭建一个 STUN 服务器，现在比较流行的 STUN 服务器是 CoTURN，你可以到 GitHub 上自己下载源码编译安装。

当 STUN 服务器安装好后，从内网主机发送一个 binding request 的 STUN 消息到 STUN 服务器。

STUN 服务器收到该请求后，会将请求的 IP 地址和端口填充到 binding response 消息中，然后顺原路将该消息返回给内网主机。此时，收到 binding response 消息的内网主机就可以解析 binding response 消息了，并可以从中得到自己的外网 IP 和端口。

2. TURN 协议

这里需要说明一点，relay 服务是通过 TURN 协议实现的。所以我们经常说的 relay 服务器或 TURN 服务器它们是同一个意思，都是指中继服务器。

咱们言归正转，知道了内网主机如何通过 STUN 协议获取到 srflx 类型的候选者后，那么中继类型候选者，即 relay 型的 Candidate 又是如何获取的呢？下面我们就来看一下。

首先你要清楚，**relay 型候选者的优先级与其他类型相比是最低的**，但在其他候选者都无法连通的情况下，relay 候选者就成了最好的选择。因为**它的连通率是所有候选者中连通率最高的**。

其实，relay 型候选者的获取也是通过 STUN 协议完成的，只不过它使用的 STUN 消息类型与获取 srflx 型候选者的 STUN 消息的类型不一样而已。

RFC5766 的 TURN 协议描述了如何获取 relay 服务器（即 TURN 服务器）的 Candidate 过程。其中最主要的是 Allocation 指令。通过向 TURN 服务器发送 Allocation 指令，relay 服务就会在服务器端分配一个新的 relay 端口，用于中转 UDP 数据报。

不过这里我只是简要描述了下，如果你对这块感兴趣的话，可以直接查看 RFC5766 以了解更多的细节。

NAT 打洞 / P2P 穿越

当收集到 Candidate 后，WebRTC 就开始按优先级顺序进行连通性检测了。它首先会判断两台主机是否处于同一个局域网内，如果双方确实是在同一局域网内，那么就直接在它们之间建立一条连接。

但如果两台主机不在同一个内网，WebRTC 将尝试**NAT 打洞，即 P2P 穿越**。在 WebRTC 中，NAT 打洞是极其复杂的过程，它首先需要对 NAT 类型做判断，检测出其类型后，才

能判断出是否可以打洞成功，只有存在打洞成功的可能性时才会真正尝试打洞。

WebRTC 将 NAT 分类为 4 种类型，分别是：

完全锥型 NAT

IP 限制型 NAT

端口限制型 NAT

对称型 NAT

而每种不同类型的 NAT 的详细介绍我们将在下一篇关于 NAT 穿越原理一文中进行讲解，现在你只要知道 NAT 分这 4 种类型就好了。另外，需要记住的是，对称型 NAT 与对称型 NAT 是无法进行 P2P 穿越的；而对称型 NAT 与端口限制型 NAT 也是无法进行 P2P 连接的。

ICE

了解了上面的知识后，你再来看 ICE 就比较简单了。其实 ICE 就是上面所讲的获取各种类型 Candidate 的过程，也就是：**在本机收集所有的 host 类型的 Candidate，通过 STUN 协议收集 srflx 类型的 Candidate，使用 TURN 协议收集 relay 类型的 Candidate。**

因此，有人说 ICE 就是包括了 STUN、TURN 协议的一套框架，从某种意义上来说，这样描述也并不无道理。

小结

通过上面的讲解，我想你现在已经基本了解 WebRTC 端对端建立连接的基本过程。在 WebRTC 中，它首先会尝试 NAT 穿越，即尝试端到端直连。如果能够穿越成功，那双方就通过直连的方式传输数据，这是最高效的。但如果 NAT 穿越失败，为了保障通信双方的连通性，WebRTC 会使用中继方式，当然使用这种方式传输效率会低一些。

在整个过程中，WebRTC 使用**优先级**的方法去建立连接，即局域网内的优先级最高，其次是 NAT 穿越，再次是通过中继服务器进行中转，这样就巧妙地实现了“既要高效传输，又能保证连通率”这个目标。

当然，即使 WebRTC 处理得这样好，但还有不够完美的地方。举个例子，对于同一级别多个 Candidate 的情况，WebRTC 就无法从中选出哪个 Candidate 更优了，它现在的做法是，在同一级别的 Candidate 中，谁排在前面就先用谁进行连接。

思考时间

若你查阅相关资料，一定会发现 Candidate 的类型是四种，而不是三种，多了一种 prflx 类型，那么 prflx 类型与 srflx 类型的区别是什么呢？

欢迎在留言区与我分享你的想法，也欢迎你在留言区记录你的思考过程。感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给更多的朋友。



从 0 打造音视频直播系统

手把手教你打造实时互动音视频直播系统

李超

新东方音视频直播技术专家
前沪江音视频架构师



新版升级：点击「👤请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 08 | 有话好商量，论媒体协商

精选留言 (6)

写留言



Jason

2019-08-05

是不是这样的？srflx：内网地址被NAT映射后的地址，对称型 NAT 与对称型 NAT、对称型 NAT 与端口限制型 NAT是无法进行 P2P 穿越的；prflx：TUN Server上为客户端分配的中继地址，与各种NAT类型地址都可以进行P2P连接；

展开 ▾



刘丹

2019-08-04

请问ICE是哪3个英文单词的缩写？是Internet Communication Engine吗？能否创建一个术语表章节？

作者回复: Interactive Connectivity Establishment (ICE)，可以参考
<https://tools.ietf.org/html/rfc5245>



1



山石尹口

2019-08-03

连通性检测时的超时设置比较重要，设置短了，会把可以连通的判断为不能连通，设置长了，就会在不能连通的配对上浪费时间

展开 ▾

作者回复: 没错！



许童童

2019-08-03

prflx candidate (prflx候选者)：是一个候选地址，通过从主机候选地址发送一个STUN请求到运行在Peer候选地址上的STUN服务器而获取的候选地址。

展开 ▾

作者回复: 它与 srflx有什么不同呢？



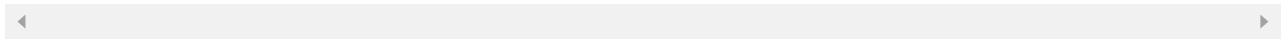
许童童

2019-08-03

老师你好，可以解释一下为什么需要 NAT 穿越吗？

展开 ▾

作者回复: 因为两端直接建立连接效率高，占用资源少



💬 1



Beast-Of-Prey

2019-08-03

打卡

展开 ▾

