

31 | 一对多直播系统RTMP/HLS，你该选哪个？

2019-09-24 李超

从0打造音视频直播系统

[进入课程 >](#)



讲述：李超

时长 20:26 大小 14.05M



近年来，随着智能手机的普及以及移动互联网的飞速发展，音视频技术在各个行业得到了广泛的应用。尤其是娱乐直播在前几年着实火了一把，像映客、斗鱼这类以展示才艺为主的直播产品非常受大家的欢迎。

从技术角度来讲，映客、斗鱼这类娱乐直播与在线教育、音视频会议直播有着非常大的区别。在线教育、音视频会议这类直播属于**实时互动直播**，主要考虑**传输的实时性**，因此一般使用 **UDP** 作为底层传输协议；而娱乐直播对实时性要求不高，更多关注的是画面的质量、音视频是否卡顿等问题，所以一般采用 **TCP** 作为传输协议。我们称前者为实时互动直播，后者为传统直播。

本专栏的前两个模块都是介绍实时互动直播的，而从今天开始我们会讲解传统直播技术。

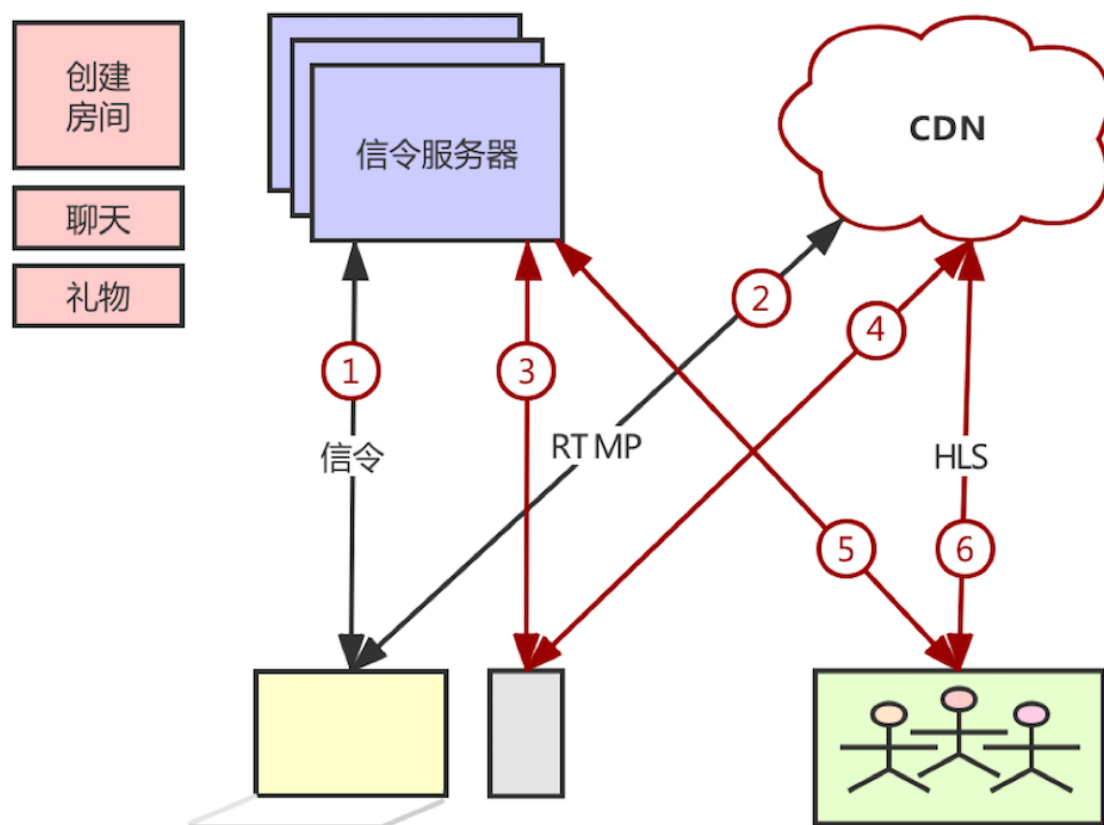
传统直播技术使用的传输协议是 RTMP 和 HLS。其中，RTMP 是由 Adobe 公司开发的，虽然不是国际标准，但也算是工业标准，在 PC 占有很大的市场；而 HLS 是由苹果公司开发的，主要用在它的 iOS 平台，不过 Android 3 以后的平台也是默认支持 HLS 协议的。

接下来，我们先看一下传统音视频直播系统的基本架构，让你对传统直播架构的“内幕”有一个初步的了解。

传统直播基本架构

商业级直播系统的规模、结构是非常复杂的，除了最核心的音视频直播外，还包括用户管理、认证系统、直播间管理、打赏、红包、私信等很多功能，不过这些更多的是一些业务逻辑，在本文中我们不会对它们进行讲解，而是聚焦在**最核心的音视频直播技术上**。

我们先来看一下传统直播的基本架构图，如下图所示：



传统直播基本架构图

从图中可以看出，传输直播架构由**直播客户端**、**信令服务器**和**CDN 网络**这三部分组成。下面我们就来一一分析下这每个模块的功能以及它们彼此之间的联系。

直播客户端主要包括音视频数据的采集、编码、推流、拉流、解码与播放这几个功能。

但实际上，这几个功能并不是放在同一个客户端中实现的。为什么呢？因为作为主播来说，他不需要看到观众的视频或听到观众的声音；而作为观众来讲，他们与主播之间是通过文字进行互动的，因此也不需要分享自己的音视频。

所以用过快手、映客等这类产品的同学都知道，客户端按用途可分为两类，一类是主播使用的客户端，包括音视频数据采集、编码和推流功能；另一类是观众使用的客户端，包括拉流、解码与渲染（播放）功能。

对于主播客户端来说，它可以从 PC 或移动端设备的摄像头、麦克风采集数据，然后对采集到的音视频数据进行编码，最后将编码后的音视频数据按 RTMP 协议推送给 CDN 源节点（RTMP 接入服务器）。

对于观众客户端来说，它首先从直播管理系统中获取到房间的流媒体地址，然后通过 RTMP 协议从边缘节点拉取音视频数据，并对获取到的音视频数据进行解码，最后进行视频的渲染与音频的播放。

通过上面的描述，看上去主播端与观众端的开发工作量差不多。但实际上，观众端的开发工作量要小得多。其原因是，观众端要实现的就是一个播放器功能，而目前开源界有两个比较有名而又成熟的开源项目 `libplayer` 和 `VLC`，所以只要将这两个开源项目中的一个集成到你自己的项目中，基本上就完成了观众端的所有开发工作。

信令服务器，主要用于接收信令，并根据信令处理一些和业务相关的逻辑，如创建房间、加入房间、离开房间、送礼物、文字聊天等。

实际上，这部分功能并不是很复杂，但有一点需要你特别注意，那就是聊天消息的处理。我们来举个例子，在一个有 10000 人同时在线的房间里，如果其中一个用户发送了文字消息，那么服务端收到该消息之后就要给 10000 人转发。如果主播说“请能听到我声音的人回复 1”，那这时 10000 人同时发消息，服务端要转发多少条呢？要转发 $10000 * 10000 = 1$ 亿条消息。这对于任何一台服务器来说，都会产生灾难性的后果。所以，在开发直播系统的信令服务器时，**一定要关注和防止消息的洪泛**。

CDN 网络，主要用于媒体数据的分发。它内部的实现非常复杂，我们姑且先把它当作是一个黑盒子，然后只需要知道传给它的媒体数据可以很快传送给全世界每一个角落。换句话

说，你在全世界各地，只要接入了 CDN 网络，你都可以快速看到你想看到的“节目”了。

介绍完直播客户端、信令服务器和 CDN 网络之后，我们再来来看看**主播到底是如何将自己的音视频媒体流进行分享的**。

主播客户端在分享自己的音视频媒体流之前，首先要向信令服务器发送“创建房间”的信令（红圈标出的步骤 1）；信令服务器收到该信令后，给主播客户端返回一个推流地址（CDN 网络源站地址）；此时，主播客户端就可以通过音视频设备进行音视频数据的采集和编码，生成 RTMP 消息，最终将媒体流推送给 CDN 网络（红圈标出的步骤 2）。

无论主播端使用的是 PC 机还是移动端，其推流步骤都是一样的，所以从上面的架构图中我们也可以看出步骤 3、步骤 4 与步骤 1、步骤 2 是一致的。

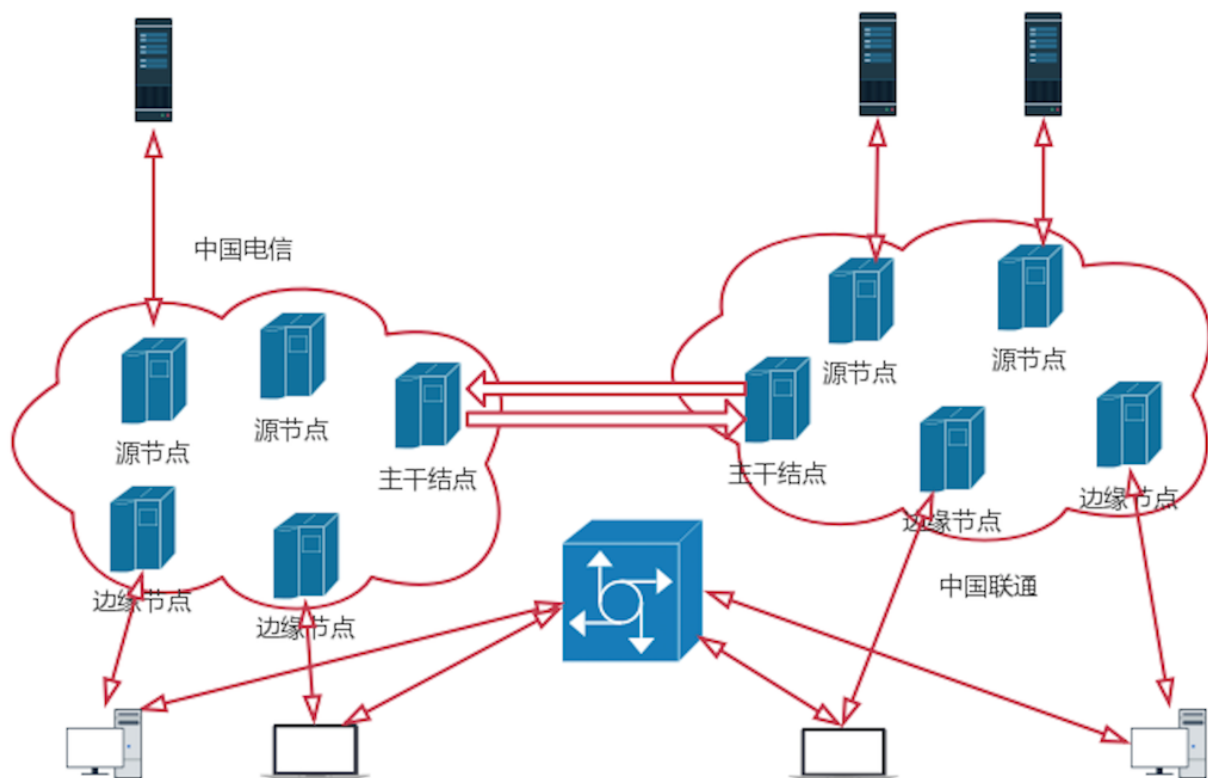
当观众端想看某个房间里的节目时，首先也要向信令服务器发消息，不过发送的可不是“创建房间”消息了，而是“加入房间”，也就是步骤 5；服务端收到该信令后，会根据用户所在地区，分配一个与它最接近的“CDN 边缘节点”；观众客户端收到该地址后，就可以从该地址拉取媒体流了，即步骤 6。

需要注意的是，在传统直播系统中，一般推流都使用的 RTMP 协议，而拉流可以选择使用 RTMP 协议或者 HLS 协议。

以上就是传统的直播架构，下面我们再来简要介绍一下 CDN 网络。

CDN 网络的实现

CDN 网络的构造十分复杂（如下图所示），一般情况下，它先在各运营商内构建云服务，然后再将不同运营商的云服务通过光纤连接起来，从而实现跨运营商的全网 CDN 云服务。



CDN 网络

而每个运营商云服务内部包括了多个节点，按功能分为 3 类。

源节点，用于接收用户推送的媒体流。

主干结点，起到媒体数据快速传递的作用，比如与其他运营商传送媒体流。

边缘节点，用于用户来主动接流。一般边缘节点的数量众多，但机子的性能比较低，它会被布署到各地级市，主要解决网络最后一公里的问题。

接下来，我们简要描述一下 CDN 网络的处理流程。

当一个主播想将自己的音视频共享出去的时候，首先通过直播系统的信令服务器获取到可以推送媒体流的 CDN 源节点。CDN 网络从源节点接收到媒体数据后，会主动向各个主干结点传送流媒体数据，这样主干结点就将媒体数据缓存起来了。当然这个缓冲区的大小是有限的，随着时间流逝，缓冲区中的数据也在不断更替中。

当有观众想看某个主播的节目时，会从直播系统的信令服务器获取离自己最近的 CDN 边缘节点，然后到这个边缘节点去拉流。由于他是第一个在该节点拉流的用户，因此该 CDN 边

缘节点还没有用户想到的媒体流，怎么办呢？那就向主干结点发送请求。主干结点收到请求后，从自己的缓冲区中取出数据流源源不断地发给边缘节点，这时边缘节点再将媒体数据发给观众。

当第二个观众再次到该 CDN 边缘节点接流时，该节点发现该流已经在自己的缓存里了，就不再向主干结点请求，直接将媒体流下发下去了。因此，观众在使用 CDN 网络时会发现，第一个观众在接流时需要花很长时间才能将流拉下来，可是后来的用户很快就将流拉下来进行播放了。

以上就是 CDN 网络的基本原理，接下来我们再来看看 RTMP 协议与 HLS 协议的比较。

RTMP 介绍

RTMP，全称 Real Time Messaging Protocol，即**实时消息协议**。但它实际上并不能做到**真正的实时**，一般情况最少都会有几秒到几十秒的延迟，底层是基于 TCP 协议的。

RTMP 的传输格式为 **RTMP Chunk Format**，媒体流数据的传输和 RTMP 控制消息的传输都是基于此格式的。

需要注意的是，在使用 RTMP 协议传输数据之前，RTMP 也像 TCP 协议一样，先进行三次握手才能将连接建立起来。当 RTMP 连接建立起来后，你可以通过 RTMP 协议的控制消息为通信的双方设置传输窗口的大小（缓冲区大小）、传输数据块的大小等。具体细节可以参考[文档 rtmp specification 1.0](#)，或者参考下一篇专门介绍 RTMP 协议细节的文章，本文就不再详述了。

1. 优势

RTMP 协议在苹果公司宣布其产品不支持 RTMP 协议，且推出 HLS 技术来替代 RTMP 协议的“打压”下，已停止更新。但协议停止更新后，这么多年仍然屹立不倒，说明该协议肯定有它独特的优势。那有哪些呢？

RTMP 协议底层依赖于 TCP 协议，不会出现丢包、乱序等问题，因此音视频业务质量有很好的保障。

使用简单，技术成熟。有现成的 RTMP 协议库实现，如 FFmpeg 项目中的 librtmp 库，用户使用起来非常方便。而且 RTMP 协议在直播领域应用多年，技术已经相当成熟。

市场占有率高。在日常的工作或生活中，我们或多或少都会用到 RTMP 协议。如常用的 FLV 文件，实际上就是在 RTMP 消息数据的最前面加了 FLV 文件头。

相较于 HLS 协议，它的实时性要高很多。

2. 劣势

RTMP 有优势，也有劣势。在 RTMP 的众多劣势中，我认为最为关键的有两条。

苹果公司的 iOS 不支持 RTMP 协议，按苹果官方的说法，RTMP 协议在安全方面有重要缺陷。

在苹果的公司的压力下，Adobe 已经停止对 RTMP 协议的更新了。

可以看出 RTMP 协议已经失去了未来，只是由于目前没有更好的协议可以直接代替它，所以它还能“苟延残喘”存活几年。但它最终一定会消亡，这是毋庸置疑的。

HLS 介绍

HLS，全称 HTTP Live Streaming，是苹果公司实现的基于 HTTP 的流媒体传输协议。它可以支持流媒体的直播和点播，主要应用在 iOS 系统和 HTML5 网页播放器中。

HLS 的基本原理非常简单，它是将多媒体文件或直接流进行切片，形成一堆的 ts 文件和 m3u8 索引文件并保存到磁盘。

当播放器获取 HLS 流时，它首先根据时间戳，通过 HTTP 服务，从 m3u8 索引文件获取最新的 ts 视频文件切片地址，然后再通过 HTTP 协议将它们下载并缓存起来。当播放器播放 HLS 流时，播放线程会从缓冲区中读出数据并进行播放。

通过上面的描述我们可以知道，**HLS 协议的本质就是通过 HTTP 下载文件，然后将下载的切片缓存起来。**由于切片文件都非常小，所以可以实现边下载边播的效果。HLS 规范规定，播放器至少下载一个 ts 切片才能播放，所以 HLS 理论上至少会有一个切片的延迟。

1. 优势

HLS 是为了解决 RTMP 协议中存在的一些问题而设计的，所以，它自然有自己的优势。主要体现在以下几方面：

RTMP 协议没有使用标准的 HTTP 接口传输数据，在一些有访问限制的网络环境下，比如企业网防火墙，是没法访问外网的，因为企业内部一般只允许 80/443 端口可以访问外网。而 HLS 使用的是 HTTP 协议传输数据，所以 HLS 协议天然就解决了这个问题。

HLS 协议本身实现了码率自适应，不同带宽的设备可以自动切换到最适合自己的码率的视频进行播放。

浏览器天然支持 HLS 协议，而 RTMP 协议需要安装 Flash 插件才能播放 RTMP 流。

2. 不足

HLS 最主要的问题就是**实时性差**。由于 HLS 往往采用 10s 的切片，所以最小也要有 10s 的延迟，一般是 20 ~ 30s 的延迟，有时甚至更差。

HLS 之所以能达到 20 ~ 30s 的延迟，主要是由于 HLS 的实现机制造成的。HLS 使用的是 HTTP 短连接，且 HTTP 是基于 TCP 的，所以这就意味着 HLS 需要不断地与服务器建立连接。TCP 每次建立连接时都要进行三次握手，而断开连接时，也要进行四次挥手，基于以上这些复杂的原因，就造成了 HLS 延迟比较久的局面。

如何选择 RTMP 和 HLS

分析完 RTMP 和 HLS 各自的优势和劣势后，接下来我们就结合项目实践，对这二者的使用场景做一些建议。

流媒体接入，也就是推流，应该使用 RTMP 协议。

流媒体系统内部分发使用 RTMP 协议。因为内网系统网络状况好，使用 RTMP 更能发挥它的高效本领。

在 PC 上，尽量使用 RTMP 协议，因为 PC 基本都安装了 Flash 播放器，直播效果要好很多。

移动端的网页播放器最好使用 HLS 协议。

iOS 要使用 HLS 协议，因为不支持 RTMP 协议。

点播系统最好使用 HLS 协议。因为点播没有实时互动需求，延迟大一些是可以接受的，并且可以在浏览器上直接观看。

小结

本文我们首先详细介绍了传统直播系统的架构，并展示了主播端与观众端是如何通过该系统进行直播的。

然后我们又对 RTMP 协议和 HLS 协议进行了比较，给出了一些使用建议，这些只能说是一些经验，具体实践时还得根据产品实际需求来考虑。比如，有些产品只需要网页播放，那就考虑 HLS 协议；有些产品很简单，是私有部署，不需要考虑互联网需求，那直接用 RTMP 协议即可。所以说，如何选择还是得根据产品需求来定。

总体来看，随着人们对实时性、互动性的要求越来越高，人们对传统直播技术的需求越来越少。本专栏前两个模块中介绍的 WebRTC 技术正是为了解决人们对实时性、互动性需求而提出的新技术。所以，以 WebRTC 为代表的实时直播系统必然会成为未来的主流。另一方面，随着 5G 的落地与推广，带宽问题会得到极大的解决，因此人们在享受实时性、互动性便利的同时，音视频服务质量也会得到极大的改善，这将加速传统的基于 RTMP /HLS 的直播技术的淘汰。

当然，基于 RTMP/HLS 的直播技术的淘汰不是一蹴而就的，因为在大负载方面，使用 CDN 技术的传统直播目前还有一定优势，所以短期内可以考虑 WebRTC 的实时直播技术与使用 RTMP 协议为代表的传统直播技术相结合的方案，既可以满足用户对于实时性、互动性的需求，又可以满足大负载的需求，这也是各大厂积极推动的一套方案。

思考时间

我们前面讲过 WebRTC 采用 UDP/RTP 进行媒体数据传输，而 RTMP 却使用了 TCP 协议，那它为什么不使用 UDP 协议呢？

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 30 | 实战演练：通过WebRTC实现多人音视频实时互动直播系统

下一篇 32 | HLS：实现一对多直播系统的必备协议

精选留言 (5)

写留言

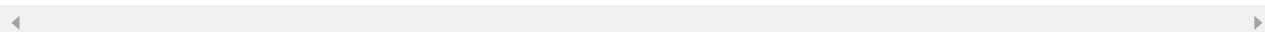


刘丹

2019-09-26

看这门课的目录，好像WebRTC只支持几个人、几十个人的视频会议，不支持上万人在线的直播？

作者回复：后面倒数第4篇就是万人直播的方案，通过CDN网络，都是现成的技术



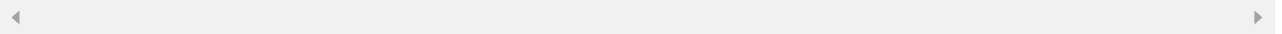
Geek_701918

2019-09-26

还有一个RTMP不支持h265

展开 ▾

作者回复: 赞，确实是这样的



Keep-Moving

2019-09-24

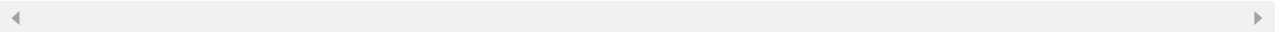
RTMP协议

* 优势:

1. 底层依赖于TCP
2. 使用简单，技术成熟
3. 市场占有率高...

展开 ∨

作者回复: 赞!

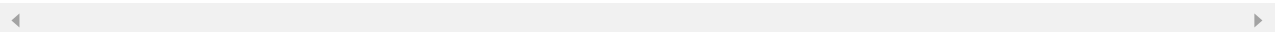


John

2019-09-24

一直想不明白很多电脑上的Chrome是怎么播放m3u8的 老师能不能讲解一下

作者回复: 后面就会讲到，耐心等待!



frank

2019-09-24

技术解决的问题域决定其实现，rtmp主要的应用场景是单向直播，并不是双向互动（虽然现在小程序也可以rtmp做双向视频通讯，毕竟做了大量优化的），所以使用TCP带来的一秒左右的延时完全可以接收，而且可以简化协议设计、实现和大规模商用

展开 ∨

作者回复: 同意你看面的说法。但如果想做实时互动的話，使用 TCP 时，如果在网络不错的情况下是没有问题的，但在极端网络情况下，TCP 就无能为力了。

