

# 直播 常见问题 产品文档



腾讯云

**【版权声明】**

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

---

## 文档目录

### 常见问题

直播基础知识

视频卡顿怎么办

如何实现秒开

如何降低延迟

为何推流不成功

为何直播看不了

如何实现好的画质

如何适配苹果 ATS

如何实现连麦功能

频道模式和直播码模式ID删除问题

禁播断流的区别

关于直播域名备案和资质审查常见问题

# 常见问题

## 直播基础知识

最近更新时间：2018-07-24 16:09:24

### 1. 推流、直播和点播分别是什么？

- **推流**：主播将本地视频源和音频源推送到腾讯视频云服务器，在有些场景中也被称为“RTMP 发布”。
- **直播**：直播的视频源是实时生成的，有人推流直播才有意义，一旦主播停播，直播 URL 也就失效了，而且由于是实时直播，所以播放器在播直播视频的时候是没有进度条的。
- **点播**：点播的视频源是云端的一个文件，文件只要没有被提供方删除就随时可以播放（类似优酷土豆、爱奇艺和腾讯视频），而且由于整个视频都在服务器上，所以播放的时候是有进度条的。

### 2. 常见的直播协议有哪些？

目前常见的直播协议有三种：RTMP、FLV 和 HLS。

- **RTMP**：RTMP 协议比较全能，既可以用来推送又可以用来直播，其核心理念是将大块的视频帧和音频帧“切碎”，然后以小数据包的形式在互联网上进行传输，而且支持加密，因此隐私性相对比较理想，但拆包组包的过程比较复杂，所以在海量并发时也容易出现一些不可预期的稳定性问题。
- **FLV**：FLV 协议由 Adobe 公司主推，格式极其简单，只是在大块的视频帧和音视频头部加入一些标记头信息，由于这种极致的简洁，在延迟表现和大规模并发方面都很成熟，唯一的不足就是在手机浏览器上的支持非常有限，但是用作手机端 App 直播协议却异常合适。
- **HLS**：苹果推出的解决方案，将视频分成 5-10 秒的视频小分片，然后用 m3u8 索引表进行管理，由于客户端下载到的视频都是 5-10 秒的完整数据，故视频的流畅性很好，但也同样引入了很大的延迟（HLS 的一般延迟在 10-30s 左右）。相比于 FLV，HLS 在 iPhone 和大部分 Android 手机浏览器上的支持非常给力，所以常用于 QQ 和微信朋友圈的 URL 分享。

直播协议	优点	缺点	播放延迟
FLV	成熟度高、高并发无压力	需集成SDK才能播放	2s - 3s
RTMP	优质线路下理论延迟最低	高并发情况下表现不佳	1s - 3s
HLS(m3u8)	手机浏览器支持度高	延迟非常高	10s - 30s

### 3. 常见的点播协议有哪些？

目前常见的点播格式有三种：MP4、HLS 和 FLV。

- **MP4**：非常经典的文件格式，在移动终端和 PC 浏览器上的支持度都很好（在 iOS 和大部分 Android 设备上，都可以使用系统浏览器进行播放，在 PC 上可以使用 Flash 控件进行播放），但是 MP4 的视频文件格式比较复杂，所以处理成本高，而且由于索引表复杂度高，导致时长稍大（比如半小时）的 MP4 文件在线播放时加载速度会很慢。
- **HLS**：苹果公司力推的标准，在移动终端的浏览器上的支持度较好，但 IE 的支持情况依赖 Flash 的二次开发工作（建议使用腾讯视频云的 Flash 播放器控件），其精简的 m3u8 的索引结构可以规避 MP4 的索引慢问题，如果是用于点播，是非常不错的选择。
- **FLV**：Adobe 公司所推的标准，目前直播平台最常用的封装格式，在 PC 端有 Flash 的强力支持，但在移动终端只有 App 实现播放器才有可能支持（或者使用本播放器），大部分手机端浏览器均不支持，目前腾讯视频云的直播录制，采用的就是 FLV 视频格式。

点播协议	优点	缺点
HLS(m3u8)	手机浏览器支持度高	大量小分片的文件组织形式，错误率和维护成本均高于单一文件
MP4	手机浏览器支持度高	格式过于复杂和娇贵，容错性很差，对播放器的要求很高
FLV	格式简单问题少，适合直播转录制场景	手机浏览器支持差，需集成SDK才能播放

#### 4. 常见的推流协议有哪些？

虽然 RTMP 在直播领域不是特别流行，但是在推流服务，也就是【主播】>【服务器】这个方向上 RTMP 居于主导地位，目前国内的视频云服务都是以 RTMP 为主要推流协议，由于腾讯视频云 SDK 第一个功能模块就是主播推流，所以也被称为是 RTMP SDK。

#### 5. 腾讯 RTMP SDK 支持哪些功能和协议？

腾讯视频云 RTMP SDK 支持推流、直播和点播三个功能：

- **推流**：支持 RTMP 发布协议，并包含硬件加速、美颜滤镜、带宽适应、清晰度调整等强大功能。
- **直播**：支持 FLV 协议和 RTMP 协议，推荐使用 FLV，具有秒开优化，延迟自动控制技术以及适应性良好的硬件解码能力。
- **点播**：支持 MP4\HLS\FLV 文件在线点播服务，注意老版本 SDK 是只支持 FLV 点播的。

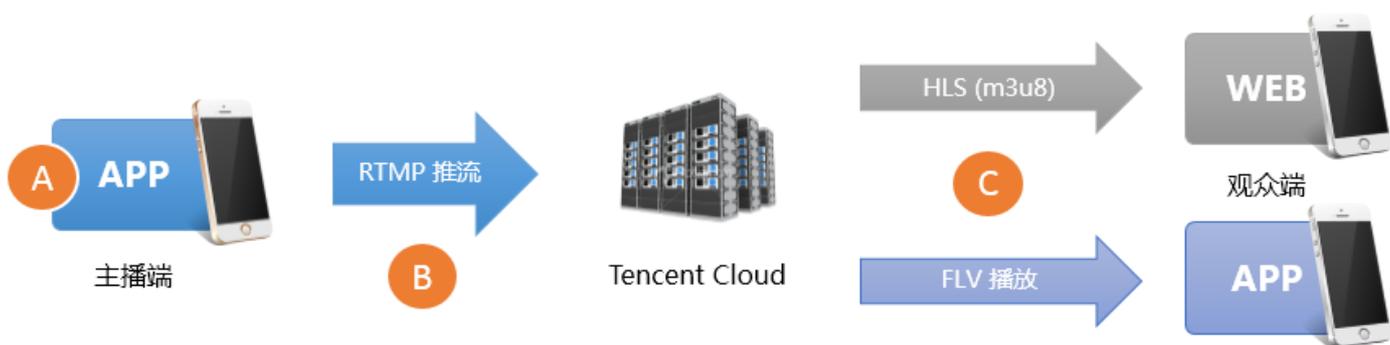
#### 6. 直播需要网络视听许可证吗？

通过互联网向用户提供直播等音视频服务的应用程序，需要取得相关部门颁发的《信息网络传播视听节目许可证》。

# 视频卡顿怎么办

最近更新时间：2018-07-24 16:19:42

## 1. 为什么会卡顿



卡顿的原因无外乎三种情况：

### • 原因 1：帧率太低

如果主播端手机性能较差，或者有很占 CPU 的后台程序在运行，可能导致视频的帧率太低。正常情况下每秒 15FPS 以上的视频流才能保证观看的流畅度，如果 FPS 低于 10 帧，可以判定为**帧率太低**，这会导致**全部观众**的观看体验都很卡顿。

### • 原因 2：上传阻塞

主播的手机在推流时会源源不断地产生音视频数据，但如果手机的上传网速太小，那么产生的音视频数据都会被堆积在主播的手机里传不出去，上传阻塞会导致**全部观众**的观看体验都很卡顿。

**国内运营商**提供的宽带上网套餐中，下载网速虽然已经达到了 10Mbps，20Mbps 甚至是 100Mbps，但上传网速却还一直限制的比较小，很多小城市的上行网速最快是 512Kbps（也就是每秒最多上传 64KB 的数据）。

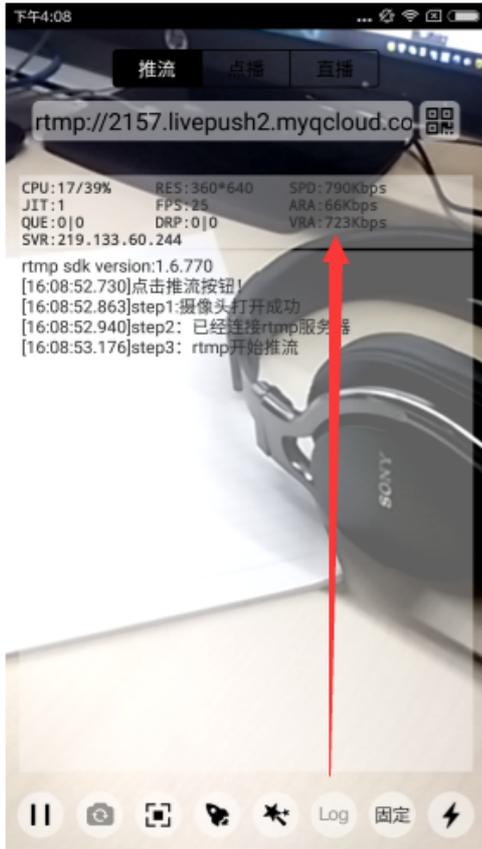
**Wi-Fi 上网**遵循 IEEE 802.11 规定的载波多路侦听和冲突避免标准，简言之就是一个 Wi-Fi 热点同时只能跟一个手机通讯，其它手机在跟热点通讯前都要先探测或询问自己是否能够通讯，所以一个 Wi-Fi 热点使用的人越多就越慢。同时 Wi-Fi 信号受建筑墙体的屏蔽干扰很严重，而一般的中国普通家庭很少在装修时考虑好 Wi-Fi 路由器和各个房间的信号衰减问题，可能主播本人也不清楚自己做直播的房间离家里的路由器究竟穿了几堵墙。

### • 原因 3：下行不佳

就是观众的下载带宽跟不上或者网络很波动，比如直播流的码率是 1Mbps 的，也就是每秒钟有 1M 比特的数据流要下载下来，但如果观众端的带宽不够，就会导致观众端体验非常卡顿。下行不佳只会影响当前网络环境下的观众。

## 2. 发现问题的“眼睛”

推流 SDK 提供了一种状态反馈机制，每隔 1-2 秒就会将内部各种状态参数反馈出来，我们可以通过注册 `TXLivePushListener` 监听器来获取这些状态。



RTMP SDK 状态参数表

推流状态	参数名称	含义说明
CPU	CPU使用率	App: 17% Sys: 39%
RES	推流分辨率	360 * 640
SPD	网络上行速度	790kbps
JIT	网络抖动	不推荐参考
FPS	视频帧率	25帧/秒
ARA	音频码率	66kbps
QUE	缓冲积压	0帧
DRP	主动丢包	推流以来尚未丢过
VRA	视频码率	723kbps

推流状态	含义说明
NET_STATUS_CPU_USAGE	当前进程的 CPU 使用率和本机总体的 CPU 使用率
NET_STATUS_VIDEO_FPS	当前视频帧率，也就是视频编码器每条生产了多少帧画面
NET_STATUS_NET_SPEED	当前的发送速度（单位：kbps）
NET_STATUS_VIDEO_BITRATE	当前视频编码器输出的比特率，也就是编码器每秒生产了多少视频数据，单位：kbps
NET_STATUS_AUDIO_BITRATE	当前音频编码器输出的比特率，也就是编码器每秒生产了多少音频数据，单位：kbps
NET_STATUS_CACHE_SIZE	音视频数据堆积情况，这个数字超过个位数，即说明当前上行带宽不足以消费掉已经生产的音视频数据
NET_STATUS_CODEC_DROP_CNT	全局丢包次数，为了避免延迟持续恶性堆积，SDK 在数据积压超过警戒线以后会主动丢包，丢包次数越多，说明网络问题越严重。
NET_STATUS_SERVER_IP	连接的推流服务器的 IP，一般应该是离客户端跳数比较少的就近服务器

### 3. 帧率太低

#### 3.1 帧率太低的评判

通过 TXLivePushListener 的 VIDEO\_FPS 的状态数据，我们可以获得当前推流的视频帧率。正常来说每秒 15FPS 以上的视频流才能保证观看的流畅度，如果 FPS 在 10 帧以下，观众就会明显的感到画面卡顿。

## 3.2 针对性优化方案

### 3.2.1 观察 CPU\_USAGE 的大小

通过 TXLivePushListener 的 CPU\_USAGE 的状态数据，我们可以获得**当前推流 SDK 的 CPU 占用情况**和**当前系统的 CPU 占用情况**。如果当前系统的整体 CPU 使用率超过 80%，那么视频的采集和编码都会受到影响，无法正常发挥作用；如果 CPU 使用率达到 100%，那么主播端本身就已经很卡，观众端要有流畅的观看体验显然是不可能的。

### 3.2.2 确认谁在消耗 CPU

一款直播 App 中使用 CPU 的不可能只有 RTMP SDK，弹幕、飘星、文本消息互动等都有可能消耗一定的 CPU，这些都是不可避免的。如果单纯要测试推流 SDK 的 CPU 占用情况，可以使用我们的[简单版 DEMO](#)来观察和评估。

### 3.2.3 不盲目追高分辨率

过高的视频分辨率并不一定能带来清晰的画质：首先，较高的分辨率要配合较高的码率才能发挥效果，低码率高分辨的清晰度很多时候比不上高码率低分辨率。其次，像 1280 x 720 这样的分辨率在平均 5 寸左右的手机屏幕上并不能看出优势，要像跟 960 x 540 的分辨率拉开差距，只有在 PC 上全屏观看才能有明显的感官差异。但较高的分辨率会显著提升 SDK 的 CPU 使用率，因此常规情况下推荐使用 TXLivePush 的 setVideoQuality 设置**高清**档即可，盲目追高分辨率有可能达不到预期的目标。

### 3.3.4 适当使用硬件加速

现在的智能手机都支持硬件编码来降低视频编码对 CPU 的依赖，如果您发现您的 App 的 CPU 使用率过高，可以开启硬件编码来降低 CPU 使用率。TXLivePush 的 setVideoQuality 的**高清档**默认使用的是软件编码（硬件编码在部分 Android 手机上的编码效果不佳，马赛克感很强是个硬伤），如果要使用硬件编码，可以使用 TXLivePushConfig 的 enableHWAcceleration 选项开启。

## 4. 上传阻塞

据统计，视频云客户群 80% 以上的直播间卡顿问题，均是由于主播端上传阻塞所致。

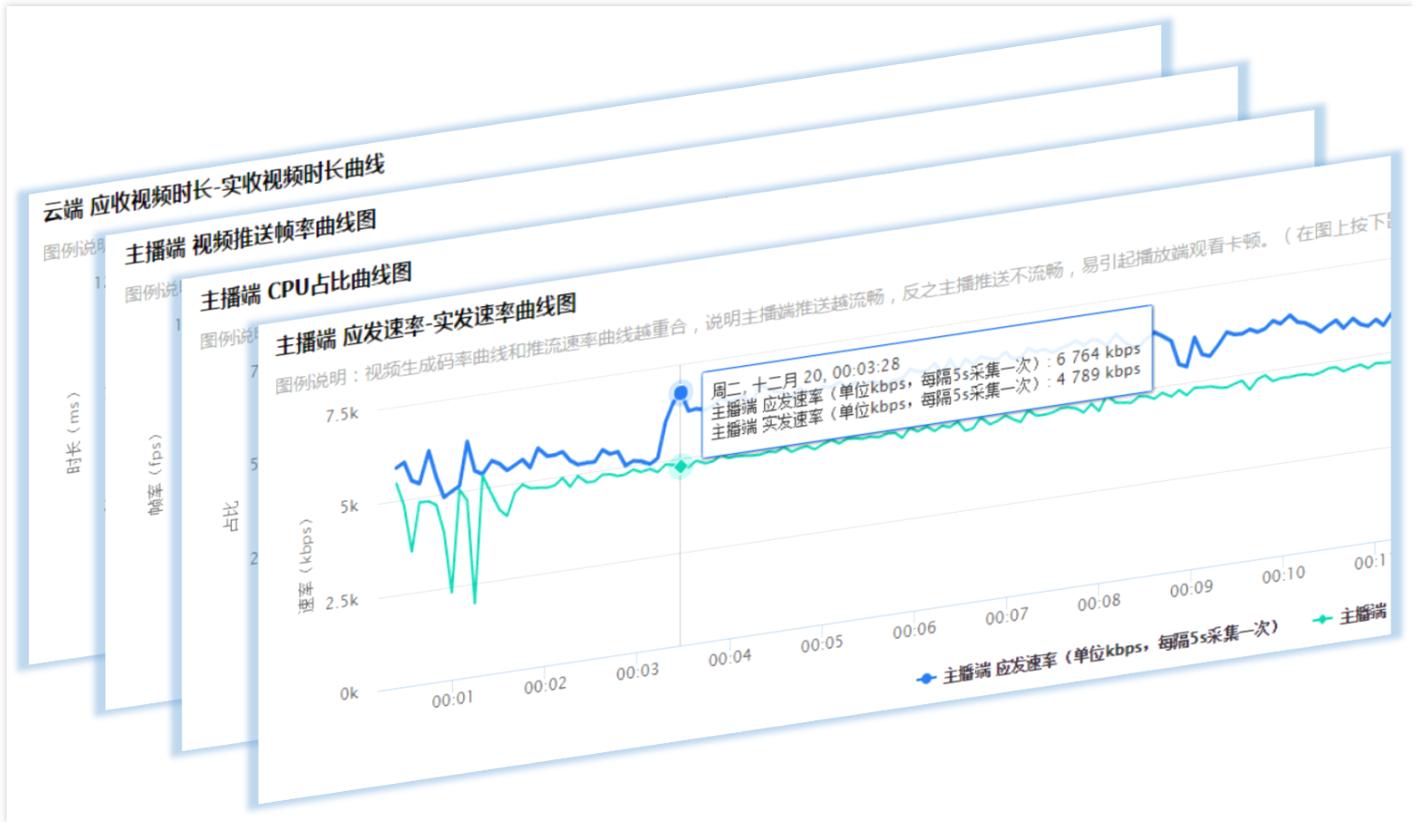
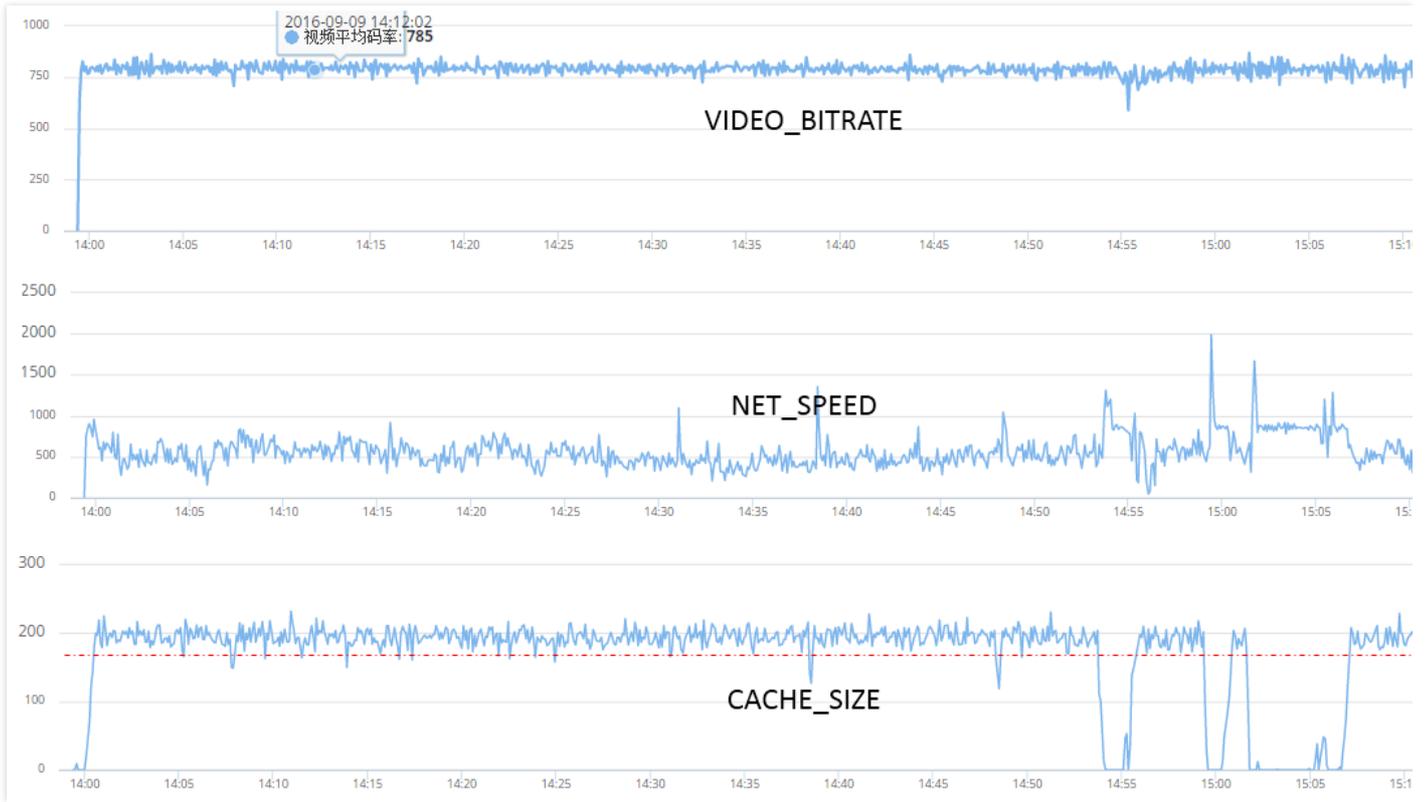
### 4.1 上传阻塞的评判

#### 4.1.1 : BITRATE 与 NET\_SPEED 的关系

BITRATE( = VIDEO\_BITRATE + AUDIO\_BITRATE ) 指的是编码器每秒产生了多少音视频数据要推出去，NET\_SPEED 指的是每秒实际推出了多少数据，所以如果 BITRATE == NET\_SPEED 的情况是常态，则推流质量会非常良好；而如果 BITRATE >= NET\_SPEED 这种情况的持续时间比较长，推流质量就很难有什么保障。

#### 4.1.2 : CACHE\_SIZE 和 DROP\_CNT 的数值

BITRATE >= NET\_SPEED 的情况一旦出现，编码器产生的音视频数据就会在主播的手机上积压起来，积压的严重程度以 CACHE\_SIZE 这个状态值展示出来，如果 CACHE\_SIZE 超过警戒线，SDK 会主动丢弃一些音视频数据，从而触发 DROP\_CNT 的增长。下图所示就是一个典型的上行阻塞，途中 CACHE\_SIZE 始终在**红色警戒线**以上，说明上行网络不足以满足数据的传输需求，也就是上行阻塞严重：



注意：

您可以在【直播控制台】>【质量监控】里看到类似上图的图表。

## 4.2 针对性优化方案

### 4.2.1 主动提示主播

对于注重清晰度的场景下，通过合适的 UI 交互提示主播“当前网络质量很糟糕，建议您拉近离路由器的距离，避免 Wi-Fi 穿墙”是最好的选择。

RTMP SDK 的推流功能文档中有涉及 **事件处理** 的介绍，您可以利用它来做到这一点，推荐的做法是：如果 App 在短时间内连续收到 RTMP SDK 的多个 **PUSH\_WARNING\_NET\_BUSY** 事件，则提示主播网络关注一下当前网络质量，因为对于上行阻塞这种情况而言，主播本人是没办法通过视频的表现感知到的，只能通过观众的提醒或者 App 的提醒来了解。

### 4.2.2 合理的编码设置

如下是我们推荐的编码设置（适合美女秀场，更多信息请参考 [如何实现更好的画质？](#)），可以通过 TXLivePush 里的 setVideoQuality 接口进行相应档位的设置：

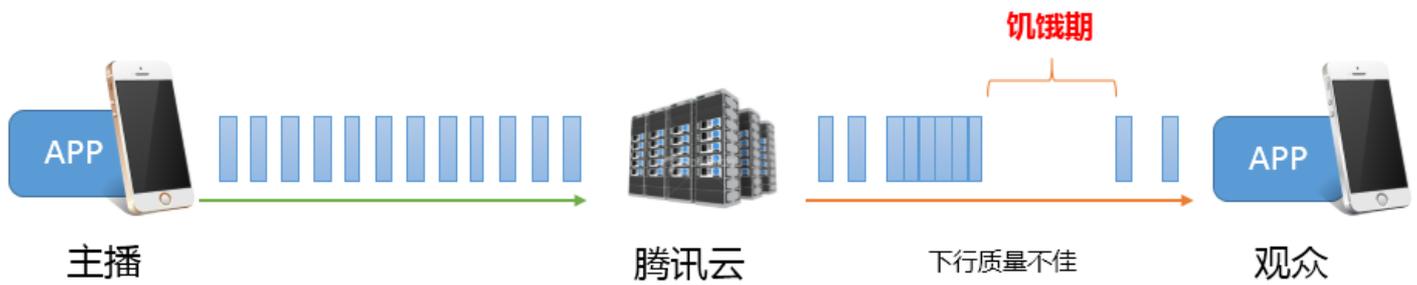
档位	分辨率	FPS	码率	使用场景
标清	360*640	15	400kbps - 800kbps	如果您比较关注带宽成本，推荐选择该档位，画质会偏模糊，但带宽费用较高清档要低 60%。
高清 (推荐)	540*960	15	1200kbps	如果您比较关注画质，推荐选择该档位，能确保绝大多数主流手机都能推出很清晰的画面。
超清	720*1280	15	1800kbps	慎用：如果您的场景多是小屏观看不推荐使用，如果是大屏幕观看且主播网络质量很好可以考虑。

### 4.2.3 启用流控辅助

有些客户会说：“我们的 App 任何用户都有可能使用，让我去决定他们的网速不现实。”如果主播的上传网速的不确定性确实很大，推荐开启网络自适应，参考文档见 [iOS 平台](#) & [Android 平台](#)。不过我们依然建议优先参考 **4.2.1 主动提示主播** 中的方案，毕竟既要高清流畅，又不能保证上传网速，本身就是 **既要马儿跑得快，又要马儿不吃草** 的事情。

设置项	中文含义	建议值
videoBitrateMin	最小码率	400
videoBitrateMax	最大码率	1000（推荐根据具体分辨率而定）
enableAutoBitrate	码率自适应	开启
autoAdjustStrategy	码率调整策略	AUTO_ADJUST_BITRATE_STRATEGY_2

## 5. 播放端的优化



## 5.1 卡顿 & 延迟

如上图，下行网络的波动或者下行带宽不沟通，都会导致在播放过程中出现一段段的**饥饿期**——App 这段时间内拿不到可以播放的音视频数据。如果想要让观众端的视频卡顿尽量少，就要尽可能地让 App 缓存足够多的视频数据，以保证它能平安度过这些“饥饿期”，但是 App 缓存太多的音视频数据会引入一个新的问题 —— **高延迟**，这对互动性要求高的场景是很坏的消息，同时如果不做延迟修正和控制，卡顿引起的延迟会有**累积效应**，就是播放时间越久，延迟越高，延迟修正做得好不好是衡量一款播放器是否足够优秀的关键指标。所以**延迟和流畅是一架天平的两端**，如果过分强调低延迟，就会导致轻微的网络波动即产生明显的播放端卡顿。反之，如果过分强调流畅，就意味着引入大量的延迟（典型的案例就是 HLS(m3u8)通过引入 10-30 秒的延迟来实现流畅的播放体验）。

## 5.2 针对性优化方案

为了能够让您无需了解过多流控处理知识就能优化出较好的播放体验，腾讯云 RTMP SDK 经过多个版本的改进，优化出一套自动调节技术，并在其基础上推出了三种比较优秀的 [延迟控制方案](#)：

- **自动模式**：如果您不太确定您的主要场景是什么，可以直接选择这个模式。

把 TXLivePlayConfig 中的 setAutoAdjustCache 开关打开，即为自动模式。在该模式下播放器会根据当前网络情况，对延迟进行自动调节（默认情况下播放器会在 1s - 5s 这个区间内自动调节延迟大小，您可以通过 setMinCacheTime 和 setMaxCacheTime 对默认值进行修改），以保证在足够流畅的情况下尽量降低观众跟主播端的延迟，确保良好的互动体验。

- **极速模式**：主要适用于**秀场直播**等互动性高，因而对延迟要求比较苛刻的场景。

极速模式设置方法是 `setMinCacheTime = setMaxCacheTime = 1s`，自动模式跟极速模式的差异只是 MaxCacheTime 有所不同（极速模式的 MaxCacheTime 一般比较低，而自动模式的 MaxCacheTime 则相对较高），这种灵活性主要得益于 SDK 内部的自动调控技术，可以在不引入卡顿的情况下自动修正延时大小，而 MaxCacheTime 反应的就是调节速度：MaxCacheTime 的值越大，调控速度会越发保守，卡顿概率就会越低。

- **流畅模式**：主要适用于**游戏直播**等大码率高清直播场景。

当把播放器中的 setAutoAdjustCache 开关关闭，即为流畅模式，在该模式下播放器采取的处理策略跟 Adobe Flash 内核的缓存出策略如出一辙：当视频出现卡顿后，会进入 loading 状态直到缓冲区蓄满，之后进入 playing 状态，直到下一次遭遇

无法抵御的网络波动。默认情况下缓冲大小为 5s，您可以通过 `setCacheTime` 进行更改。

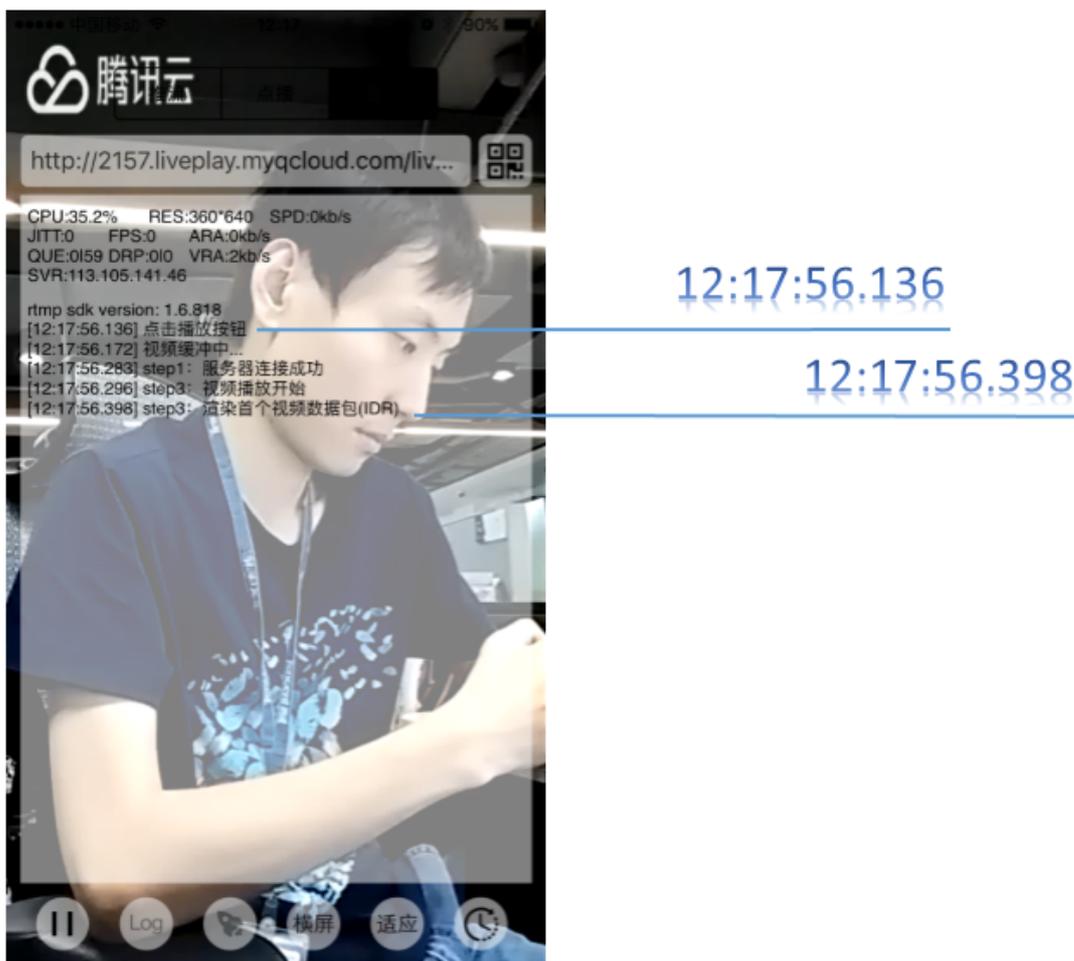
在延迟要求不高的场景下，这种看似简单的模式会更加可靠，因为该模式本质上就是通过牺牲一点延迟来降低卡顿率。

# 如何实现秒开

最近更新时间：2018-07-10 14:57:29

## 什么叫做“秒开”？

**秒开** 即从视频播放开始到真正看到第一帧画面所消耗的时间要尽可能的短（几百毫秒时间），不能让观众有明显的等待时间：



这种能力主要依靠云端服务的优化以及播放器的配合，如果您组合使用腾讯云音视频 SDK 配合视频云服务实现直播能力，最快可以实现 **200ms 左右** 的首屏打开速度，如果网络下行足够好的话甚至可以瞬开。

## 如何实现“秒开”？

### App 端

使用 [腾讯云音视频 SDK](#) + FLV 播放协议即可实现秒开：

- **HTTP + FLV 播放协议**

HTTP + FLV 协议是目前直播行业使用最普遍的播放协议，它的数据组织格式比较简单，可以做到一旦连通服务器就能获取到音视频数据。相比之下，RTMP 协议由于连接初期不可避免的几次协商握手过程，导致在首帧速度方面略逊于 FLV 协议。

- **腾讯云音视频 SDK**

秒开的云端实现原理其实非常简单，服务器始终缓存一组 GOP 画面（至少包含一个可以用于解码的关键帧），这样播放器一旦连通服务器就可以获取到一帧关键帧（I 帧），进而可以解码和播放，但这种云端的缓存也会带来副作用：播放器在连通服务器后，通常会一口气被塞过来几秒钟的音视频数据，从而产生不小的播放端延迟，我们称之为“秒开后遗症”。

一款好的播放器，除了具备秒开能力，还要具备优秀的**延迟修正**能力，能够在无损观看体验的情况下，自动修正播放端延迟到一个合理的范围内（比如 1 秒以内），而腾讯云音视频 SDK 在这方面就做的非常优秀，您甚至可以指定播放器的延迟修正模式（[iOS](#) & [Android](#)）。

## PC 浏览器

PC 浏览器的视频播放内核一般都是采用 Flash 控件（目前 Chrome 也支持 MSE，但并不比 Flash 有明显优势），Flash 播放器策略是比较刚性的**强制缓冲**模式，所以视频打开速度没有什么优化空间，一般很难做到 1 秒以内，这一点可以通过各大视频网站和直播平台的 PC 端表现就能发现。

## 手机浏览器

- **iPhone**

Safari 对 HLS (m3u8) 的支持很好，甚至直接使用 iPhone 的硬解芯片协助视频播放，在具备 DNS 缓存的情况下，视频打开速度通常都有保障，但这也仅限于 iOS 平台。

- **Android**

Android 上的表现就具有比较大的随机性，由于碎片化严重，各个版本和机型的系统浏览器实现都有差异，QQ 和 微信内的浏览器甚至采用了腾讯自己的 X5 内核，所以具体表现会有比较大的差异。

# 如何降低延迟

最近更新时间：2018-07-10 15:02:22

按正常情况 RTMP 推流+FLV 播放的正常延迟在 2-3 秒左右，太长则是有问题的，如果您发现直播延迟时间特别长，可以按照如下思路来排查：

## Step 1. 检查播放协议

不少客户播放协议采用 HLS(m3u8)，并感觉延迟较大，这个是正常的。苹果主推的 HLS 是基于大颗粒的 TS 分片的流媒体协议，每个分片都有 5s 以上的时长，分片数量一般为 3-4 个，所以总延迟在 20s - 30s 就不足为怪。

换用 FLV 作为播放协议即可解决这个问题，但是要注意，如果您要在手机浏览器上观看直播视频，只有 HLS(m3u8) 这一种播放协议可以选择，其它的直播协议在苹果的 Safari 浏览器上都是不支持的。

## Step 2. 检查播放器设置

腾讯云 RTMP SDK 的播放器支持极速、流畅和自动三种模式：

- **极速模式**：能保证绝大多数场景下延迟都在 2-3 秒以内，美女秀场适合这个模式。
- **流畅模式**：绝大多数场景下延迟都在 5 秒以内，适合对延迟不敏感但对流畅度要求高的场景，比如游戏直播。



## Step 3. 后台不要打水印

腾讯云支持后台打水印，目的是满足一些不能使用腾讯云 RTMP SDK 的推流器（支持直播端加水印）但依然要打水印的客户，但是这种方案会引入额外的三秒延迟，故如果您本身使用的是腾讯云 RTMP SDK 来推流，就把后台水印关闭后在主播端的 App 上加水印。

## Step 4. 第三方推流器？

我们只能确保在腾讯云一体化解决方案中保持理想的效果，如果您使用的是第三方推流软件，建议您先换成腾讯云 RTMP SDK 的推流 Demo 做个对比，排除一下第三方推流器的编码缓存引入大延迟的可能，因为很多第三方的推流器对会暴力的采用无限缓冲的方式来解决上行带宽不足的问题。

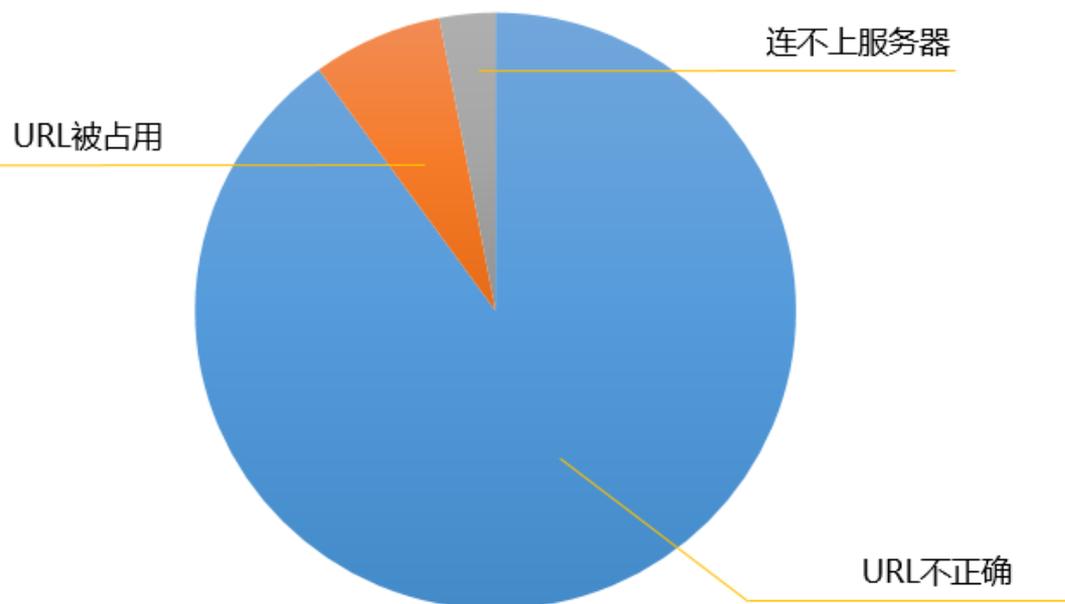
### Step 5. 检查 OBS 设置

不少客户反馈采用 OBS 推流，在播放端延迟比较大。建议按照 [PC 推流](#) 文档中配置对应的参数，并注意要把关键帧间隔设置为 1 或者 2。

# 为何推流不成功

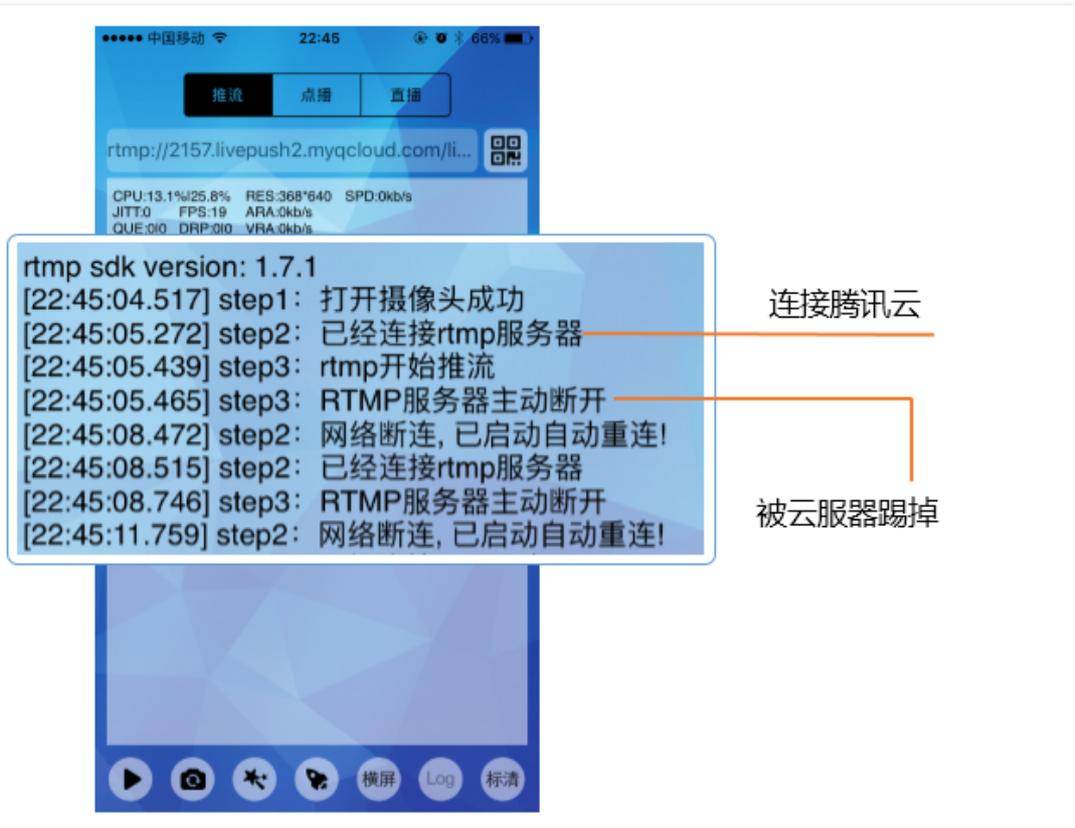
最近更新时间：2018-07-24 16:21:41

腾讯云统计客户投诉中关于推流不成功的原因，主要集中在如下三个点上：



## txSecret 不正确

腾讯云目前要求推流地址都要加防盗链以确保安全，防盗链计算错误或者已经过了有效期的推流 URL，都会被腾讯云踢掉，这种情况下 RTMP SDK 会抛出 `PUSH_WARNING_SERVER_DISCONNECT` 事件，[RTMP SDK DEMO](#) 此时的表现如下：



阅读 [如何获取推流 URL?](#) 了解如何获取可靠的推流 URL。

### txTime 已过期

有些客户担心自己的直播流量被人盗用，会将 txTime 设置的过于保守，比如从当前时间开始往后推 5 分钟。其实由于有 txSercet 签名的存在，txTime 的有效期不用设置的太短。相反，如果有效期设置的太短，当主播在直播过程中遭遇网络闪断时会因为推流 URL 过期而无法恢复推流。

txTime 建议设置为当前时间往后推 12 或者 24 小时为宜，也就是要长于一场普通直播的直播时间。

### 推流 URL 被占用

一个推流 URL 同时只能有一个推流端，第二个尝试去推流的 Client 会被腾讯云拒绝掉，这种情况下 RTMP SDK 会抛出 `PUSH_WARNING_SERVER_DISCONNECT` 事件。

### 连不上云服务器

RTMP 推流所使用的默认端口号是 **1935**，如果您测试时所在网络的防火墙不允许 1935 端口通行，就会遇到连不上与服务器的的问题。此时您可以通过切换网络（比如 4G）来排查是不是这个原因导致的问题。

### 小直播推流地址

小直播的推流 URL 可以用调试的办法获取，您可以全局搜索代码寻找关键字 `startPush`，然后在此处打下调试断点，这里是小直播对 RTMP SDK 的调用点，startPush 的参数即为推流 URL。

# 为何直播看不了

最近更新时间：2018-07-24 16:24:15

如果您发现直播无法观看，完全搞不懂里面出了什么情况，按照下面的思路进行排查，一般都能在几十秒内确认问题原因：



## step 1. 检查播放 URL

在所有检查开始之前，您务必要先检查一下地址是否正确，因为这里出错概率最高，腾讯云的直播地址分推流地址和播放地址两种，我们要首先排除**误拿推流地址来播放**的错误。

RTMP推流地址	<code>rtmp://6666.livepush.myqcloud.com/live/6666_xxxxxxxxxxxx?bizid=6666</code>
RTMP播放地址	<code>rtmp://6666.liveplay.myqcloud.com/live/6666_xxxxxxxxxxxx</code>
FLV播放地址（荐）	<code>http://6666.liveplay.myqcloud.com/live/6666_xxxxxxxxxxxx.flv</code>
HLS播放地址	<code>http://6666.liveplay.myqcloud.com/6666_xxxxxxxxxxxx.m3u8</code>

**小直播的播放 URL：**  
 小直播的播放 URL 可以用调试的办法获取，您可以全局搜索代码寻找关键字 **startPlay**，然后在此处打下调试断点，这里是小直播对 RTMP SDK 的调用点，startPlay 的参数即为播放 URL。

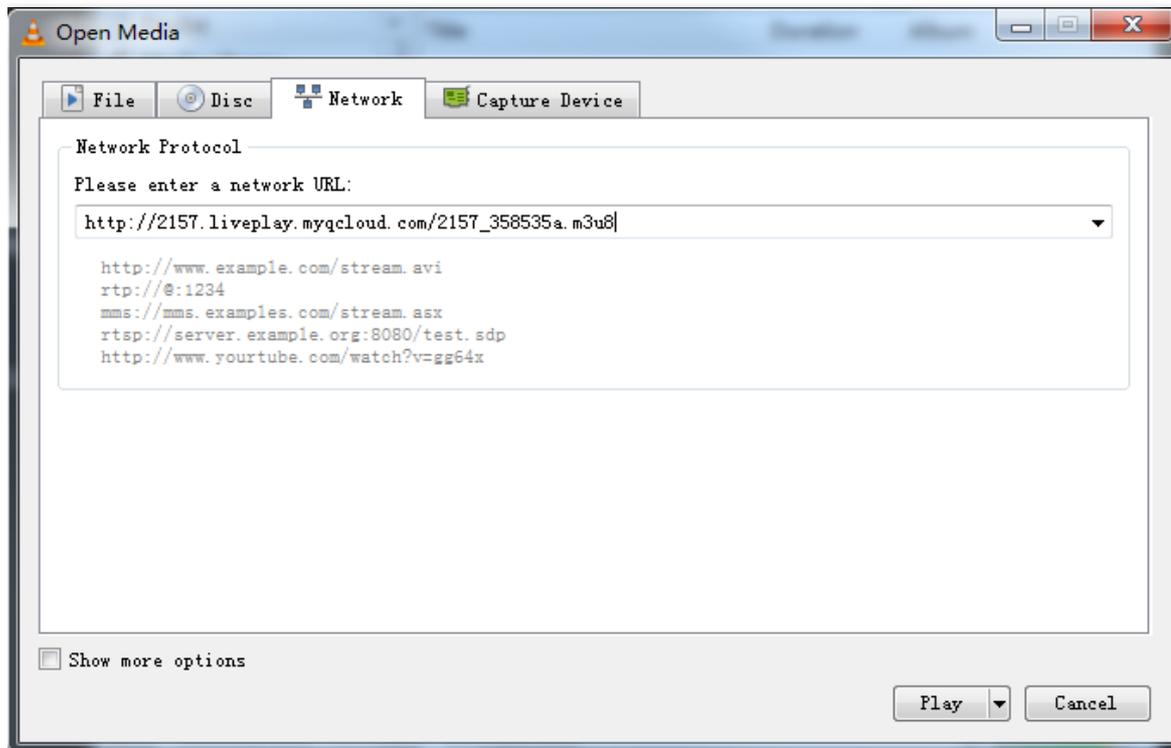
## step 2. 检查视频流

播放 URL 正确不代表视频就能播放，所以要检查视频流是否正常：

- 对于**直播**，如果主播已经结束推流，直播 URL 就不能观看。

- 对于点播，如果云端的视频文件已经被移除，同样也是不能观看。

常用的解决办法就是用 VLC 检查一下，VLC 是 PC 上的一款开源播放器，支持的协议很多，所以最适合用来做检查：



## step 3. 检查播放端

如果视频流非常健康，我们就要分情况检查一下播放器是否 OK：

### 3.1 Web 浏览器 ( A )

- **格式支持**：手机浏览器 只支持 HLS ( m3u8 ) 和 MP4 格式的播放地址。
- **HLS ( m3u8 )**：腾讯云 HLS 协议是懒启动的，简言之，只有当有观众请求 HLS 格式的观看地址后，腾讯云才会启动 HLS 格式的转码，这种懒启动策略的目的是规避资源浪费。但也就产生一个问题：**HLS 格式的播放地址要在全球首个用户发起请求后 30 秒才能观看。**
- **腾讯云 Web 播放器**：支持同时指定多种协议的播放地址，能够根据所在的平台（PC/Android/iOS?）采用最佳的播放策略，同时内部的选择性重试逻辑也能针对性解决 HLS(m3u8) 懒启动的问题。

### 3.2 RTMP SDK ( B )

如果 [RTMP SDK DEMO](#) 本身播放没有问题，推荐您参考 RTMP SDK 的播放文档（[iOS & Android](#)）检查一下对接逻辑是否正确？

## step 4. 防火墙拦截 ( C )

这是常见的一种情况，不少客户的公司网络环境会限制视频播放，限制的原理是由防火墙侦测 http 请求的是否是流媒体资源（公司老板都不希望员工上班看视频吗）。如果您使用 4G 进行直播观看没有问题，而用公司的 Wi-Fi 网络无法观看，即说明公司的网络策略有

---

所限制，您可以尝试跟网管沟通，让它给您的 IP 做一下特殊处理。

## step 5. 检查推流端 ( D )

如果是直播 URL 根本不能播放，而且没有 Step4 中防火墙限制的可能，那么很大概率是推流不成功，可以到 [为何推流不成功](#) 继续问题的排查。

# 如何实现好的画质

最近更新时间：2018-07-10 14:56:58

## 场景一：美女直播

### Step 1：更新最新版 SDK

每一版 SDK 我们都或多或少的对美颜效果进行一定的提升，比如：

- 1.9.1 我们更新了新的美颜引擎，在前景侧重、算法优化、曝光改善等方面做了不少改进，同时性能也有很大的提升。
- 1.9.2 我们优化了降噪效果，使夜间的噪点大幅减少，人物的清晰度也有提升
- 2.0.0 我们又为 iOS 增加了一些新的滤镜，来解决肤色过黄的问题
- ... ..

### Step 2：设置画质级别

我们要知道直播端看到的画面跟观众端看到的画面是不一样的：

#### • 主播 vs 观众：

主播看到的画面是经过采集后的视频直接渲染到了手机屏幕上，所以清晰度最高，该画面还要经过【视频编码】>【网络传输】>【视频解码】才能到达观众的手机屏幕，由于视频编码会导致画质的损失，所以观众看到的画面在清晰度上要逊于主播端。不合理的参数设置会导致画质损失严重，比较典型的一个错误配置就是“高分辨率配低码率”，这样的配置会导致画面模糊，并且画面马赛克严重。

#### • setVideoQuality

我在 1.9.1 版本的 TXLivePusher 中新加入的 setVideoQuality 函数，并给出了几个级别的设置，您选择 高清 模式即可达到最佳的美女直播效果，详情请关注 [iOS 平台](#) & [Android 平台](#)。

### Step 3：Android 增加手动曝光

同样美颜算法在不同的 Android 手机上差异可能很大，主要是曝光的差异产生了视觉感受的大不同。

在 iOS 平台我们采用了系统的自动曝光，但是 Android 机型差异太大，很多千元机的自动曝光效果一般，所以我们还是推荐在您的 UI 界面上提供一个自动曝光的操作滑竿，让主播可以自己调节曝光值大小，以符合他/她的预期。

Android 版 RTMP SDK 中的 TXLivePusher::setExposureCompensation 接口提供了调节曝光的能力，参数 value 为 -1 到 1 的浮点

数：0 表示不调整，-1 是将曝光降到最低，1 表示是将曝光加强到最高。



#### Step 4：增加色彩滤镜

滤镜也是很关键的一环，不同的色彩滤镜能够营造不同的氛围，主播配合衣服的颜色或者房间灯光选择合适的滤镜，会让整个画面的氛围有更好的效果。



1.9.1 版本的 RTMP SDK 开始支持颜色滤镜，TXLivePusher 中新加入的 `setFilter` 可以设置滤镜效果，我们设计师团队提供了八种素材，默认打包在了 Demo 中，您可以随意使用，不用担心版权问题。

### Step 5 : Android 马赛克严重

有些客户会发现 Android 版 RTMP SDK 推流出来的画面马赛克特别严重，尤其是在画面有运动时，这是 Android 硬件编码的常规表现，解决方案有两个：

- **比较关注耗电**

如果您比较关注 App 耗电情况，那么可以考虑直接调大推流的码率，或者使用 `setVideoQuality` 的 **高清** 档位（如果设定的码率比较低，Android 的硬编码模块会通过大幅降低画质来确保码率稳定）。

- **比较关注带宽**

如果您比较关注带宽成本，那么直接调高码率可能并不适合您的需求，这时您可以通过关闭硬件加速解决问题（关闭方法见 [setHardwareAcceleration](#)）。

### Step 6 : 关闭网络自适应

`TXLivePushConfig` 中的 `AutoAdjustBitrate` 选项是网络自适应开关，开启后，当主播网络变差时，会通过降低画质来保证流畅性，但这一点 **并不适合** 美女秀场模式。网络自适应是适合游戏直播场景的技术，因为游戏直播场景中观众对流畅性的追求高于画质，如果在战斗时间主播网络有波动，画质可以差一点，但绝对不能卡，所以降低画质来保证流畅性（帧率）就是必选项。但美女秀场场景中，清晰度更加重要，不少客户反馈有些直播间画质很好，有些直播间画质很差，导致这个现象的一个高概率原因就是开启了网络自适应。建议您关闭网络自适应，对于出现网络波动的情况，改用 **主动提示** 的方式进行优化，这种方式更加釜底抽薪。

## 场景二：游戏直播

### Option 1 : 简单应对

在直播开始的界面上提供三种清晰度选项：标清、高清和超清，**让主播自己选择**，游戏主播一般相对比较偏专业，会自己摸索出适合自己当前在玩的游戏的效果，三种档位对应的配置如下：



档位	分辨率 ( Resolution )	FPS ( FPS )	码率 ( Bitrate )
标清	VIDEO_RESOLUTION_TYPE_360_640	20	800kbps
高清	VIDEO_RESOLUTION_TYPE_540_960	20	1000kbps
超清	VIDEO_RESOLUTION_TYPE_720_1280	20	1800kbps

**注意：**  
 游戏直播场景，FPS 最低是 20 ，否则观众端的表现是卡顿感严重。

### Option 2 : 专业应对

根据不同的游戏设置不同的分辨率和码率，比如：

- **皇室战争** - 这类画面变化幅度不大的游戏，推荐选择 **960 \* 540** 的分辨率，800kbps-1000kbps 的码率就可以输出不错的效果。
- **捕鱼达人** - 这类画面变化幅度较大的游戏，推荐选择 **960 \* 540** 的分辨率，码率相对要高一点，比如1200kbps - 1500kbps。
- **神庙逃跑** - 这类画面变化幅度超大的游戏，推荐选择 **640 \* 360** 的分辨率，码率也要很大，比如 2000kbps，否则妥妥的满屏马赛克。

## 音视频小科普

### Point 1 : 720P 一定更清楚吗？

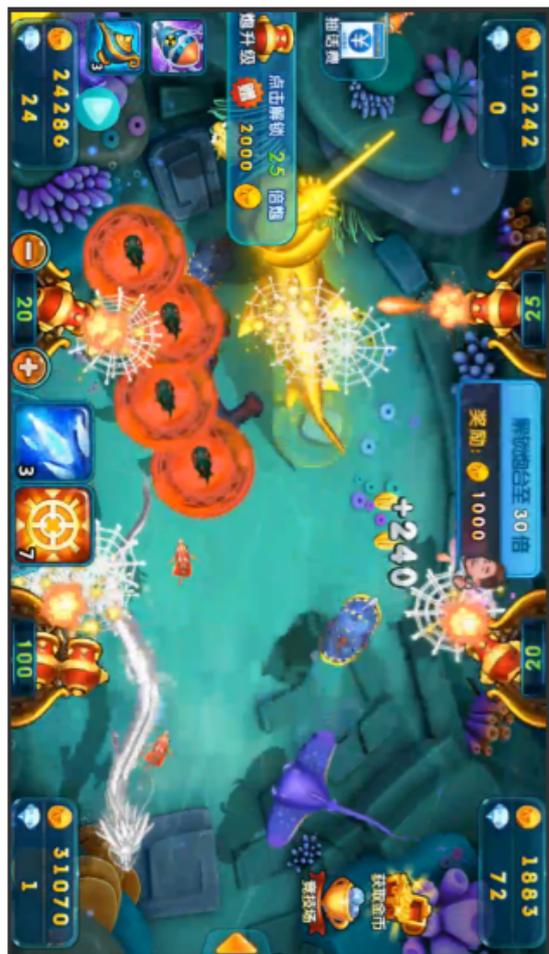
如果限定一个码率，比如 800kbps，那么**分辨率越高就会让编码器越难做**。可以想象，编码器必须拆东墙补西墙，通过减少色彩信息或者引入马赛克这种“鱼目混珠”的手段来承载足够多的像素点，所以同样的是 2G 的一个电影文件，1080p 画质的版本可能不如 720p 画质的版本看起来更清晰。

同时，如果您的观众都是在小手机屏幕上观看视频，那么 **960 \* 540 1000kbps** 同 **1280 \* 720 1800kbps** 的差距其实也不明显，比如下面两张图片就是基于 airplay 技术的 iOS 录屏直播：



960 \* 540 1000kbps

VS



1280 \* 720 1800kbps

**注意：**

用 32 寸的 LCD 显示器全屏看，还是有差距的。

### Point 2 : 帧率不要超过 24

如果限定一个码率，比如 800kbps，那么帧率越高，编码器就必须加大对单帧画面的压缩比，也就是通过降低画质来承载足够多的帧数。如果视频来自摄像头，24FPS 已经是肉眼极限，所以一般 20 帧的 FPS 就已经可以达到很好的用户体验。有些玩过 3D 游戏的朋友可能会说：“游戏的帧率越高越流畅吗？比如 60FPS，120FPS?”

这里要注意一定不要混淆场景：游戏追求高帧率是**渲染帧率**，其目的是为了尽可能让 3D 模型渲染出来的运动效果更加接近真实运动轨迹，所以帧率越高越好，但采集帧率不需要这么高，比如手机摄像头，它采集的目标是真实世界的物体，真实世界的物体本来就是连续运动的，而不是用一阵阵画面刷新来模拟的，所以 20FPS 的采集就足以。

对于游戏直播，帧率达到 24FPS 当然最好，但也要考虑到系统编码开销，手机的温度和 CPU 占用等。

# 如何适配苹果 ATS

最近更新时间：2018-07-11 10:11:37

苹果在 WWDC 2016 中表示，从 2017 年 1 月 1 日起所有新提交的 App 默认不能使用 `NSAllowsArbitraryLoads=YES` 来绕过 ATS 的限制，腾讯云将于 12 月 12 日正式支持 HTTPS，届时您只需要使用新版 SDK（接口无变化），并且将原来的视频地址的前缀从 `http://` 换成 `https://` 即可，SDK 内部会自动适配。

但需要提醒的是，https 相比于 http 在引入更多安全性（视频方面并不是特别需要）的同时也牺牲了连接速度和 CPU 使用率，所以如果您 App 在新的上架政策下还要需要继续使用 HTTP，方法修改 Info.plist，将 `myqcloud.com` 加入到 `NSExceptionDomains` 中，具体的修改如图所示：

▼ App Transport Security Settings	◇	Dictionary	(1 item)
▼ Exception Domains	◇	Dictionary	(3 items)
▼ myqcloud.com		Dictionary	(3 items)
NSExceptionRequiresForwardSecrecy		Boolean	NO
NSExceptionAllowsInsecureHTTPLoads		Boolean	YES
NSIncludesSubdomains		Boolean	YES

针对特定域名禁用 ATS 是可以被苹果审核所接受的，您可能需要在审核时进行说明，`myqcloud.com` 是用于视频播放的域名。

# 如何实现连麦功能

最近更新时间：2017-12-22 17:24:57

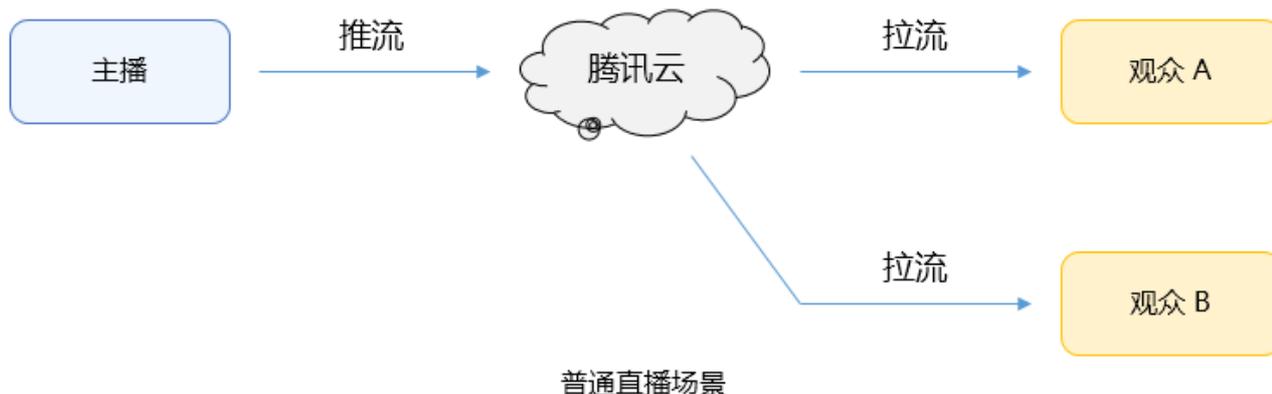
## 什么叫连麦？

连麦（也叫上麦）是今年比较热门的直播功能。所谓连麦，是指一个直播间中可以不仅只有一个主播，观众（或其它房间的主播）也可以参与进来于主播进行视频互动，从而增加视频直播的趣味性。

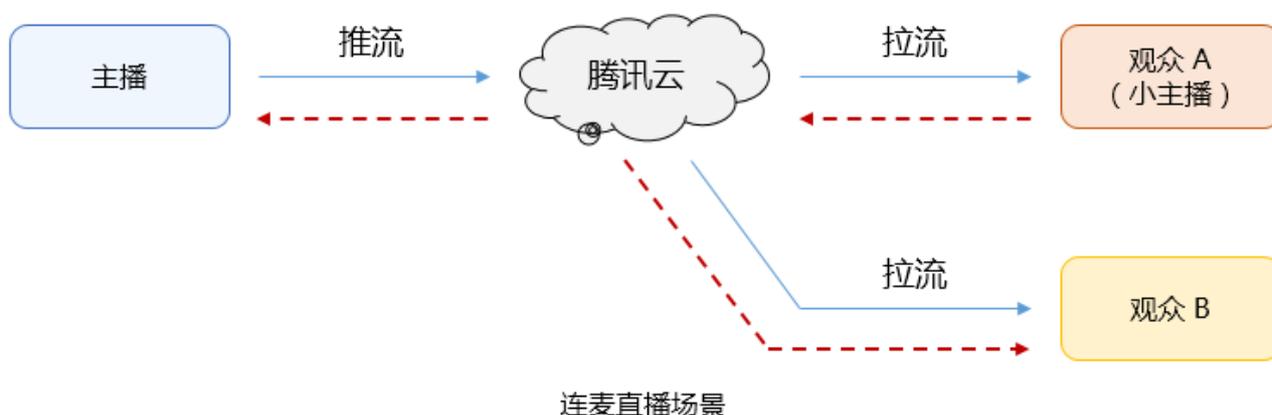


## 从“单向”到“多向”

我们先从普通的直播模式说起，目前常规的直播遵循如下图所示的模式：主播端（TXLivePusher）将音视频数据推送给云服务器，多个观众端（TXLivePlayer）就可以从云端拉流并播放音视频数据。



既然要做连麦，那么反向的一条线路就必不可少，我们这里做个假设，**观众 A** 从原来的普通观众变成了小主播，那么下图中就多出了一条直播流（图中红色虚线所示）：



**注意：**

腾讯云 RTMP 直播支持跨房间连麦互动，所以小主播（们）可以是原房间里的普通观众，也可以是另一直播间里的其他主播。

从“单向”到“多向”，这看似很简单，而且直接用 RTMP SDK 按这个思路也是可以实现的。但效果却很难达到商用的要求，因为有两个难题需要我们去解决：

• **延迟问题**

常规直播解决方案中从推流端到播放端延迟一般在 2-3 秒，但是连麦场景中大主播和小主播们之间的延迟如果超过 1s，和谐的语音沟通就基本不可能了。

• **回音问题**

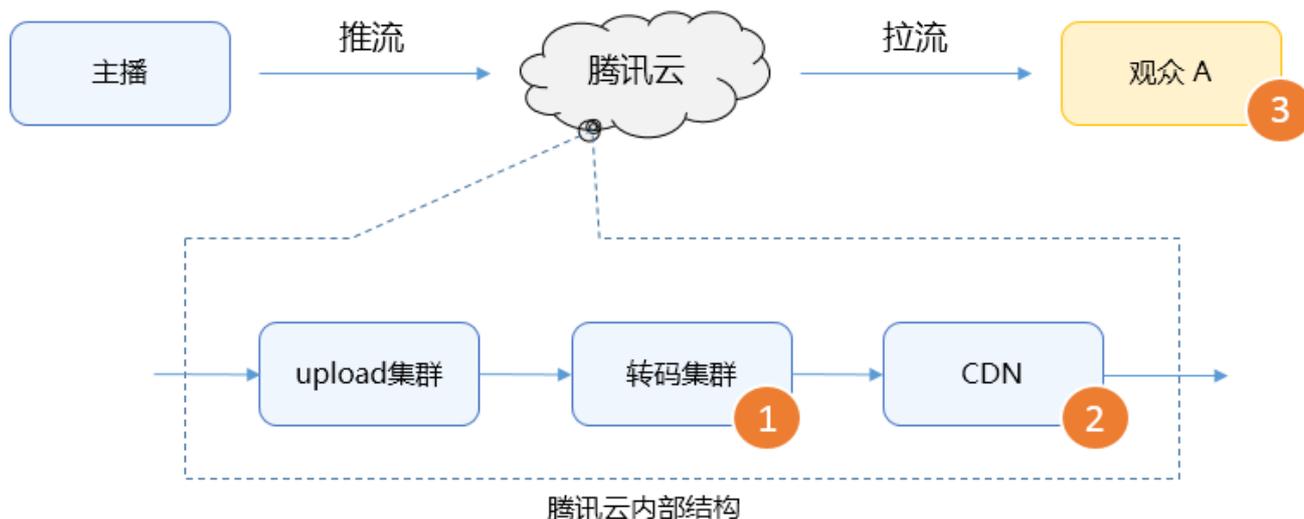
常规直播解决方案中，由于语音是单向的（主播说 => 观众听），所以没有必要去做 AEC（回音消除）。但是连麦场景中有双向（或多向）的语音沟通，主播的声音流向小主播那一端的扬声器，如果不做回音消除，这些声音会再经由麦克风采集后返还给主播本人。

• **混流问题**

解决完延迟问题和回音问题，大主播和小主播（们）之间是可以互动了，但是要让普通的直播观众看到才算真正成功，所以多路画面必须要完成混流才能在观众端正常展示。

## 如何降低延迟？

我们先看看怎么解决延迟问题，要解决延迟，先要弄明白延迟是怎么来的？



上图中红色标记的三处是整条链路的主要延迟来源，一场延迟大约 3 秒的直播中，以上三个模块“贡献”了 80% 的力量。

### 1. 转码模块

- **延迟的原因：**

转码模块的主要工作是对主播推上来的音视频数据做进一步的加工处理，同时，如果您有多清晰度（超清、高清、标清）以及多格式（比如适合 Web 播放的 HLS）的需求，这也是转码模块需要去操心的。

- **应对的思路：**

在连麦场景中，大主播和小主播（们）之间如果都使用 RTMP 协议构建链路，那就不需要转码集群的参与，因为能够省掉这部分延迟。

### 2. CDN 集群

- **延迟的原因：**

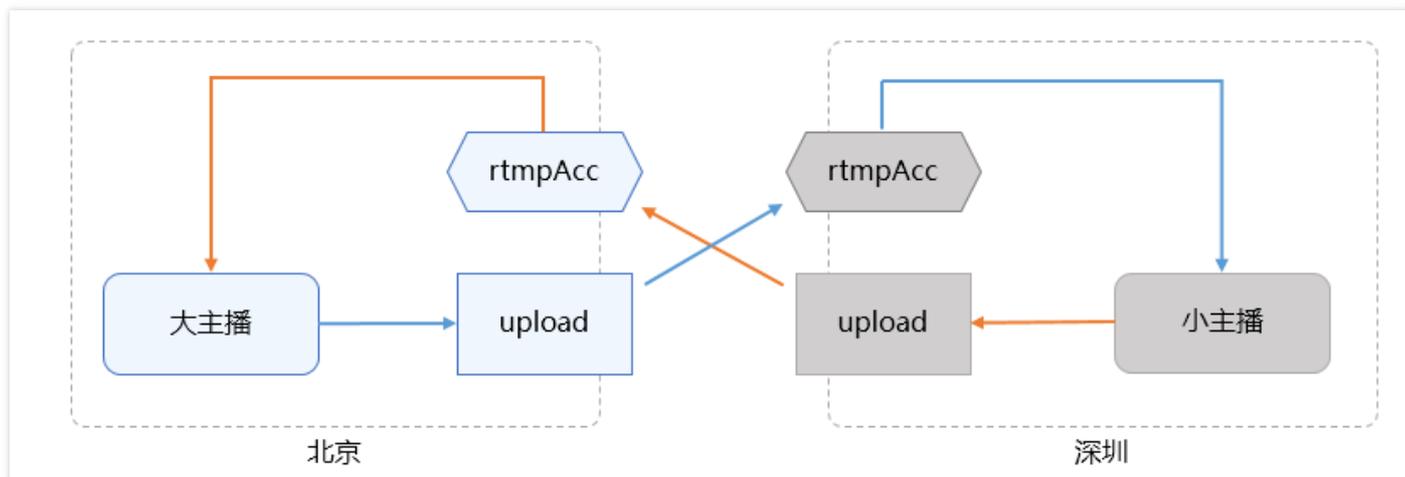
CDN 集群存在目的是 **分发** 数据流：如果主播在上海，那么他/她肯定是向上海的服务器推流，这样才能保证较好的上传质量，问题来了，桂林和哈尔滨的观众怎么办呢？难道从上海的服务器上去拉流吗？显然这并不是一个好主意，可行的方案是通过 CDN 分发集群将音视频流按需分发到桂林和哈尔滨两个城市。

- **应对的思路：**

连麦场景中，大主播和小主播（们）之间如果都不走 CDN 集群，他们之间的延迟可以缩短很大一截。但这样一来，地域问题如何解决？比如有两位主播要连麦互动，一位在北京，一位在深圳，相隔千里，如何才能构建低延时且高质量的直播链路呢？

腾讯云的解决方案是采用 **RtmpAcc 加速节点**，它是我们专门为连麦场景所设立的超低延迟加速集群，在全国各大关键网络节点均

有部署。这些加速节点全部由专线连接，唯一的职能便是为处于不同地域、不同运营商的主播，提供可靠而优质的低延时链路。



### 3. 播放器缓冲

- **延迟的原因：**

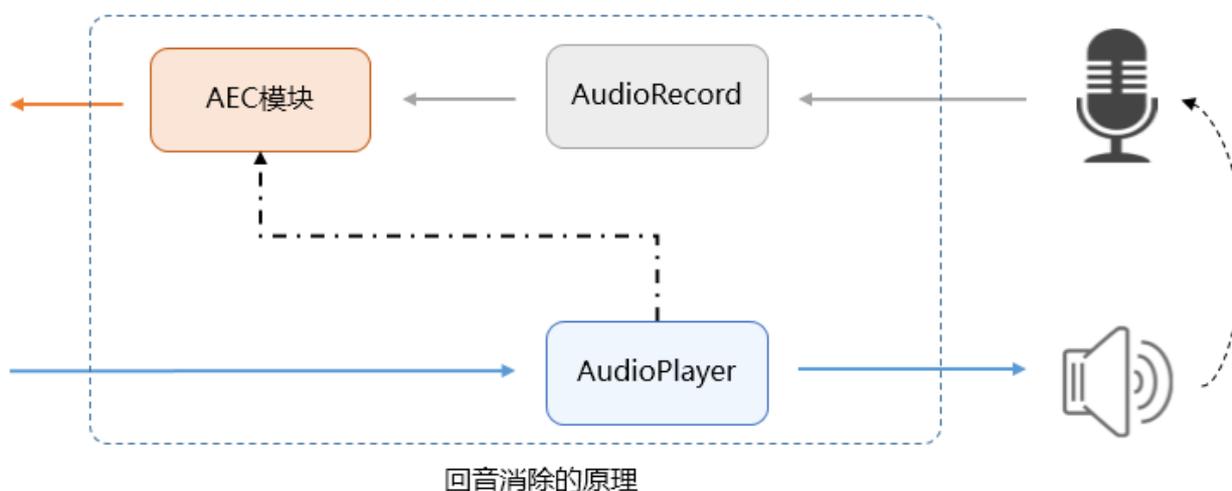
播放器的缓冲是多多少少都要有的，因为下行网络不可能均匀平滑不抖动。缓冲区越长，抗网络抖动的效果就越好，视频的观看体验也就越流畅，同时，这也意味着更大的延迟。常规解决方案中，我们一般设置 500ms 或者 1000ms 以上的播放器缓冲。

- **应对的思路：**

在连麦场景中，这里就要激进一些，比如 200ms 的延迟相对适中。于此同时，常规开源解决方案的播放器一般不具备 **延迟修正** 能力，所以随着卡顿次数的增加，延迟也越积越多。在连麦场景下，延迟是不能容忍的，所以延迟修正不仅不可或缺，而且策略也要非常激进。

## 回音消除 ( AEC )

如果要做双向的语音通讯，回音消除是不可或缺的，我们从 RTMP SDK 1.8.2 开始，在 iOS 和 Android 两个平台引入了回音消除模块，从而避免主播在手机的喇叭里听到 1 秒前自己说话的声音。



从上图可以看出，AEC 模块是躲在 RTMP SDK 里面的，所以在使用上您不需要额外的编程。

## 多路混流

腾讯云有两种技术可以实现混流：客户端混流和服务端混流。

- **客户端混流**

源自 RTMP SDK 1.8.2 开始支持的多实例播放，也就是可以并行的播放多个直播 URL，视频 View 也可以相互叠加。这样一来，只要观众端能拿到多个主播的 URL 就可以实现客户端混流。

- **服务端混流**

服务端混流是近期推出的一项新解决方案，目前外网已经可以支持。它是腾讯云视频转码集群的一个附加模块，可以将多路视频流直接在云端混成一路，减少下行的带宽压力。

混流方案	优势	不足	关键因素
客户端混流	由于观众端的表现由 App 自行控制，能够支持更灵活的界面排布	下行数据是多路，所以带宽消耗要高于服务端混流。	TXLivePlayer 的缓冲区必须要用极速模式 + 1 秒固定缓冲
服务端混流	下行数据只有一路，所以对于高并发的直播场景，能有效降低带宽消耗。	目前最多支持 1v3 混流。	需要调用混流 CGI 并正确设置混流参数

## 对接攻略

- [视频连麦 \( iOS \)](#)
- [视频连麦 \( Android \)](#)

# 频道模式和直播码模式ID删除问题

最近更新时间：2018-07-11 10:12:58

## 1. 频道删除接口问题

直播码模式没提供创建流的接口，也就没有提供删除流的接口。

频道模式提供了创建流和删除流的接口。删除接口：<https://cloud.tencent.com/document/product/267/4722>

## 2. 删除频道问题

频道模式客户可在控制台上删除频道，直播码模式不可客户删除，直播后台自动删除最近7天未推流的直播码流，如果某个直播码 id 一直有推流则不删除。

## 3. 频道数量问题

直播码模式创建的数量没有限制，如果是频道限制的问题可不担心。

频道模式现在默认只能创建 50 个，如果要更多，需要申请，后台开放限制。

另外补充下，如果推流的数量多了月并发推流数也可能会增加，会增加频道的费用。

# 禁播断流的区别

最近更新时间：2018-07-11 10:11:12

## 1.断流功能

一条直播中的流，如果断流，则本次推流会被中断，观众端将无法观看直播。断流后，主播端可以再次发起推流，继续直播活动。

## 2.禁播功能

一条直播中的流，如果禁播，则本次推流会被中断，观众端将无法观看直播。断流后，主播端在一段时间内无法再次发起推流，禁播功能生效的截止时间可控制台设置，最长不超过为一个月。

例：2018-01-01 12:00 对流 test123 设置了禁播功能，禁播截止时间设置为 2018-01-05-12:00，则在这段时间内，test123 这条流对应的主播端均无法推流，播放端无法观看此直播。

# 关于直播域名备案和资质审查常见问题

最近更新时间：2018-09-29 15:47:39

## 什么是ICP备案？

网站备案+域名备案=ICP备案，现在一般说的网站备案和域名备案泛指ICP备案。

## 如何进行ICP备案？

腾讯云为您提供协助使用大陆服务器开办网站的用户办理备案的业务。腾讯云网站备案拥有快速初审，免费幕布，7\*24小时咨询等优点。您可在腾讯云的协助下高效地为您的网站办理备案。详情请见“[网站备案](#)”。

## ICP备案是否是必须的？

是的，全国“扫黄打非”办公室会同工业和信息化部、公安部、文化和旅游部、国家广播电视总局、国家互联网信息办公室联合下发《[关于加强网络直播服务管理工作的通知](#)》（以下简称通知）要求网络直播服务提供者应向电信主管部门履行网站ICP备案手续，并向有关业务部门申请取得许可。

## 域名分发服务是怎样的？

腾讯云直播将对播放域名进行审核，若通过审核，则可接入并分发直播内容，若未通过则不允许添加和分发。腾讯云将于2018年10月30日停止用自有域名为用户提供分发服务。

## 为什么要进行资质审核？

《通知》要求，网络直播服务提供者应向电信主管部门履行网站ICP备案手续，涉及经营电信业务及互联网新闻信息、网络表演、网络视听节目直播等业务的，应分别向相关部门申请取得许可。

## 有哪些许可证，我涉及的应该是哪个许可证？

例如：电信业务—《电信业务经营许可证》；  
互联网新闻信息业务—《互联网新闻信息服务许可证》；  
网络视听节目直播业务—《信息网络传播视听节目许可证》；  
网络表演业务—《网络文化经营许可证》；  
其中《网络文化经营许可证》又包含了以下类别：

- i. 游戏类；
- ii. 音乐娱乐类；
- iii. 动漫类；
- iv. 演出类；
- v. 展览展出及比赛类；
- vi. 节目表演类；
- vii. 美术艺术品类；

另：教育类直播暂时无对应的对应的许可证类别，视后期政策而定。

您可以根据贵司业务类型去确认需要哪个许可证。

## 许可证是必须的吗？我现在没有怎么办？

《通知》要求未提供ICP备案手续或者相关业务许可材料的网络直播平台在两个月内补充相关材料。《通知》强调，对两个月后仍然无法提供相关材料的，要停止服务；对拒绝提供相关材料的，要立即停止服务。

若您现在没有许可证，请先将ICP备案的域名进行提交至腾讯云直播控制台——[域名管理](#)，此功能已于2018年9月20日上线)。后再进行许可证的申请，以备后期不时之需。

### 不知道去哪儿办理许可证，是否有推荐？

建议您去许可证官网查询办理途径，或者进入[腾讯云市场](#)咨询相关供应商，我们也会不定期进行供应商的更新。

### 如何查看黑名单？怎养确定自己不在黑名单里？

该黑名单为有关部门所建，目前暂不对外开放，只要贵司业务一切正常，便不会出现在黑名单里，若后期出现任何问题，腾讯云会第一时间与您进行联系。

### 该通知的正式执行的开始时间以及截止时间是？

腾讯云直播控制台已于2018年9月20日开放添加域名的入口，请您添加已备案域名；若您有对应的许可证，您可同时通过工单形式提交相关许可证。腾讯云直播服务将于2018年10月30日前对所有直播客户进行备案审查和资质审核，若届时未满足《通知》的要求、腾讯云直播将于2018年10月30日停止为您提供服务，并在2018年11月15日清除直播数据。

### 海外客户有什么不同？

如果用户通过国际站进行海外直播，则可不需要进行备案域名及许可证提交，该通知仅针对使用国内直播服务的用户。

如果用户开通是国内直播服务，经营是海外直播业务需要引导用户提交工单进行申请，后台开通服务后用户便可以在控制台接入未备案的域名；否则无法接入自定义域名。

为保障网络直播内容的安全、直播平台的健康发展，请您尽快根据通知完成备案或提供相关资质，感谢您的理解与配合。若您对域名备案有疑问，请参考[文档域名备案指引](#)，域名备案需要时间周期较长、建议您尽早联系服务器租用商协助办理备案。

感谢您对腾讯云的支持和信赖，如有疑问，欢迎您拨打支持热线4009-100-100或登录[控制台提工单](#)反馈，我们会及时为您核查处理。