

直播卡顿问题及优化方案

整个直播链路可以分成推流端、云端以及播放端三块。

如何监控直播质量的好坏。一般情况下，业界通常使用 QoE 和 QoS 两种指标。

- **QoE 是一种与用户体验相关的指标**，通常包括观看时长、在线人数、完播率、评论数以及 GMV/打赏等指标。这类指标以用户的主观感受为主。
- **QoS 是相对客观的服务质量指标**，其中包括卡顿、秒开、端到端延时、开播成功率，开播跳帧等等。

卡顿现象一个简单的定义：

在直播的播放过程中因网络不稳、设备性能不足、直播流内容异常等导致的音视频播放不连续的现象。

瞬时卡顿定义：

播放器缓冲区从有数据到无数据，记为一次卡顿，连续无数据的时间记为卡顿时长。

有效卡顿定义：

当连续卡顿时长超过一定的阈值，比如 50ms 的时候，才将其记作有效卡顿。通过有效卡顿，来衡量卡顿严重情况。

卡顿监控指标

1. 客户端常用指标

基于音频

百秒卡顿时长： $\text{sum}(\text{卡顿时长}) / \text{sum}(\text{观看时长}) * 100$

百秒卡顿次数： $\text{sum}(\text{卡顿次数}) / \text{sum}(\text{观看次数}) * 100$

基于视频

视频渲染百秒卡顿时长： $\text{sum}(\text{视频渲染卡顿时长}) / \text{sum}(\text{观看时长}) * 100$

视频渲染百秒卡顿次数： $\text{sum}(\text{视频渲染卡顿次数}) / \text{sum}(\text{观看次数}) * 100$

客户端大部分是基于音频来统计卡顿的。其中百秒卡顿时长就是将所有参与评价的直播观看行为中出现的音频卡顿时长加和，然后除以全部直播观看时长加和，再乘以 100。百秒卡顿次数也是类似

的定义。除了音频外，还有一些 APP 会基于视频进行卡顿统计。这就是视频渲染百秒卡顿时长和次数，它统计的卡顿是基于视频帧的卡顿，只看视频帧的缺失或渲染缺失的情况。此时求和的便是视频渲染的卡顿时长或次数，然后除以整体观看时长的加和再乘以 100。

在服务端或云端，因为它没有客户端那样的直观数据，所以我们会有一个慢速指标。比如说接收慢速就是指 5 秒内接收到的音视频 dts 小于 4 秒。也就是说 5 秒只接收了不到 4 秒的音视频数据，缺了 1 秒，那我们就认为很可能会产生卡顿。发送慢速也是类似的，5 秒内，发送缓冲区堆积的音视频大于 1 秒，即 5 秒内只发送了不到 4 秒的数据，那我们就认为客户端很可能会发生卡顿。**慢速这个指标我们在理解或是观看上不是特别直观，所以又有了流畅度这个指标。它指 5 秒钟接收到或者发送出的音视频 dts 差值。**这项指标的呈现非常直观。上图中是一条接收流畅度的曲线。蓝色线条是本地流逝时间，即 5 秒钟均匀的流逝。绿色线条是视频的流逝时间，也就是每 5 秒钟接收到的视频第一帧跟最后一帧的时间戳差值。在视频及网络均正常的情况下，这个值基本上也是接近 5 秒的。当因各种原因出现接收数据不及时的情况，就会出现一个明显的下跌。恢复后数据再次传过来，接收到的数据时间就会上涨，便会形成类似上图这样波动的曲线，非常的直观。通过抖动的情况，我们就可以很容易地看到视频的稳定性以及直播流的质量。当然，这个 5 秒的间隔，我们可以根据实际的情况来进行调整。

2. 服务端常用指标

接收慢速：

5s 内接收到的音视频 DTS 小于 4s

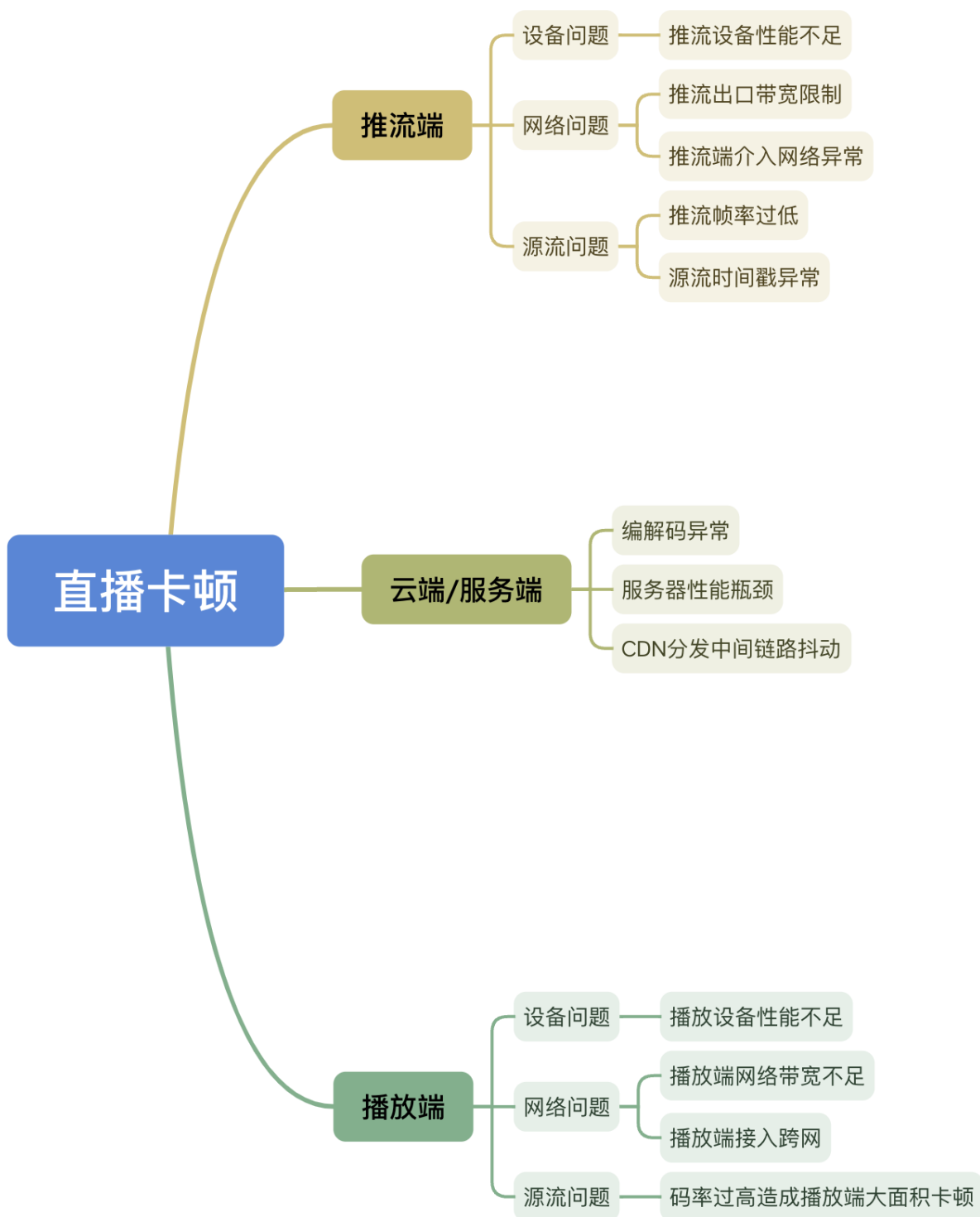
发送慢速：

5s 内发送缓冲区堆积音视频 DTS 大于 1s

流畅度：

5s 内接收或发送的音视频 DTS 差值

卡顿原因分析



卡顿定位思路

1. 先判断是大面积卡顿还是个例？

控制台查看推流是否稳定？

播放端在带宽足够时是否流畅？

是否有地区、运营商、设备、版本聚集？

1) **腾讯云官网的控制台**，这里可以查到每一条流的推流数据，包括这条流的推流帧率、码率等情况。如果图中可以看到码率有一些抖动，这是正常的，因为码率会随着画面，码控方式选择等等原因产生一些抖动。但是一般情况下，帧率都是比较平稳的。所以我们可以根据帧率来判断推流是否稳定。

2) **可以试一下在播放端带宽充足的情况下是否流畅**。比如在网络带宽良好的情况下，自己来播一下这个流，看看卡不卡。如果不卡，那至少证明推流端或者 CDN 那里大概率是没有问题的。

2. 对于大面积卡顿

推流稳定：通过降低码率来减少卡顿，常规直播推荐 1mbps+15fps 或 2mbps+30fps

推流不稳定：继续分析推流端及内容是否正常

推流端分析

1. 可能同时推多路流，导致 CPU 和带宽不够

2. 服务器公网 IP 带宽上限不足

3. 推流端接入网络异常，中小运营商多出口问题（每次推流出口变化），localDNS 设置问题一般是中小运营商多出口或者 localDNS 设置异常问题。中小运营商是租借的三大运营商的出口。有时候为了自身的成本考虑，它们会在各个出口之间进行调度。这时就有可能出现跨运营商的情况。

一些设置可能**修改了 DNS 或者使用了 VPN**等等，就导致出现了跨运营商。

4. 源流问题分析

1) 采集帧率正常，发送帧率过低；

2) 推流帧率码率持续波动；

3) 流内容时间戳不连续（回退、跳变、不规律等）

服务端用 ffprobe 进行查看，可以看到时间戳忽然跳大，然后又跳小，反复抖动。时间戳的不连续，最终在播放端就会导致播放器无法正常播放，产生各种卡顿的现象。

一般推流端出现问题，就会引起大面积的卡顿。那如果是播放端出现问题的话，一般会是一些个例的问题。所以大面积的卡顿问题，我们就往推流端以及云端/服务端查找定位。那播放端的一些个例卡顿，我们推荐先确认一下设备的终端性能如何。我们可以结合客户端的日志来查看当时 CPU

等的占用率情况。之后可以通过工具，查看是否有跨网、localDNS 异常等问题。我们还可以测一下本地的网络速度情况，可以使用 speedtest 这个工具来测试，它可以很方便的看到上传以及下载速度以及客户端 IP 等信息。

播放端性能分析

1. dump 文件分析

条件允许的话可以将视频保存下来，wget, curl, http 网页等方式保存；然后采用 ffprobe 或 flv-debugger 查看文件信息。

通过工具看到视频每一帧的 dts 以及 pts 情况，还可以看到 dts 差值等等。这个流视频的 dts 差值以及音频的 dts 的差值，如果值均匀，没有回退等情况，那这个视频就是一个比较好的视频。

云端/服务端分析

云端异常：比如说编解码的异常，一些新的编码格式可能还不支持，或者是一些硬件编码，编出来的格式不是特别的规范，导致云端兼容不了。还有服务器性能出现瓶颈。例如对一个帧率码率都很低的流，忽然提高它的码率，然后服务器没能及时将这个任务调度到资源充足的机器上去，就会出现性能瓶颈的问题。另外在 CDN 往全国甚至全球的外网进行分发的过程中，可能因为外网络的抖动，出现短暂的卡顿情况。

优化方案

OBS 推流部份

在这个工具中，我们可以在推流之前对各选项做一些设置。输出模式直接选择高级，**编码器推荐大家选 X264，因为这个软件编码的适用性更好一些**。如果你选了硬件编码，那可能不同的硬件设备编出来的音视频格式或内容会有一些差异。下面的码率控制，如果没什么特殊要求，建议选 CBR 就可以了，这样推出来会比较平稳的。比特率，也就是码率，没有特殊的要求的话，1Mbps 到 2Mbps 就够了，在一些特殊的高清场景，可以再选大一些。关于关键帧间隔，也就是我们说的 GOP 的大小，一般在 1-4 秒就好，尽量不要超过 4 秒。如果超过 4 秒可能会在延迟、兼容性等方面出现小问题。一般没有特殊要求，推荐选 2 秒就可以。关于 CPU 的预设，选择默认的 veryfast 就可以了。配置 profile 一般没什么要求，选择 main profile 或者 baseline 都是可以的。

输出分辨率这里一般按自己的要求，选择 720P 或者 1080P 来进行输出就可以。帧率 FPS 一般不用超过 30。如果是一些静态的，像 PPT 等类型，15 帧就够了。大部分情况下，都不需要超过 30 帧。

推流端部份

1. 推流网络验证

推流前验证推流网络，防止 localDNS 域名被劫持以及跨网访问问题

2. 云端转推

同时向多个平台推流的需求，推荐采用服务器转推或多个设备推流，避免单设备性能瓶颈

使用**腾讯云拉流转推**的功能。它可以非常方便的向多个平台进行转推。当要添加一个转推任务时，在控制台点击创建任务就可以了。创建任务只需要填三个信息。一个是时间，这个任务的起止时间。一个是源流的地址，源流地址就是我先推一路流出来，推到一个平台，然后拿它的播放地址作为源流地址。第三个是目标地址，就是我要转推的其他平台地址。这三个信息一填，就可以非常方便的创建一个转推任务。而且它还规避掉了自己进行服务器转推或者设备推流的各种不稳定的问题。因为它具有自动容灾，任务异常自动拉起等功能。

3.