



链滴

# 开发一个完整的直播 app? 看这里就够了 (五) 延迟优化

作者: [wulalala](#)

原文链接: <https://ld246.com/article/1474600932094>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

七牛云于 6 月底发布了一个针对视频直播的实时流网络 LiveNet 和完整的直播云解决方案，很多开发者对这个网络和解决方案的细节和使用场景非常感兴趣。

结合七牛实时流网络 LiveNet 和直播云解决方案的实践，我们用七篇文章，更系统化地介绍当下大热视频直播各环节的关键技术，帮助视频直播创业者们更全面、更深入地了解视频直播技术，更好地技术型。

本系列文章大纲如下：

- (一) 采集
- (二) 处理
- (三) 编码和封装
- (四) 推流和传输
- (五) 延迟优化
- (六) 现代播放器原理
- (七) SDK 性能测试模型

在上一篇推流和传输中，关于「直播的第一公里」的关键因素我们展开了详细的介绍。本篇是《解密视频直播技术》系列之五：延迟优化。

我们在很多线上和线下场合分享了如何优化直播体验，详细讲解了各部分造成低延迟和卡顿的原因和应的优化原理。实际上，音视频的直播系统是一个复杂的工程系统，要做到非常低延迟的直播，需要杂的系统工程优化和对各组件非常熟悉的掌握。这里面我们再分享几个简单而常用的调优技巧。

## 编码优化

1. 确保 Codec 开启了最低延迟的设置。Codec 一般都会有低延迟优化的开关，对于 H.264 来说其果尤其明显。很多人可能不知道 H.264 的解码器正常情况下会在显示之前缓存一定的视频帧，对于 Q1F 分辨率大小的视频（176 × 144）一般会缓存 16 帧，对于 720P 的视频则缓存 5 帧。对于第一帧读取来说，这是一个很大的延迟。如果你的视频不是使用 H.264 来编码压缩的，确保没有使用到 B，它对延迟也会有较大的影响，因为视频中 B 帧的解码依赖于前后的视频帧，会增加延迟。
2. 编码器一般都会有码控造成的延迟，一般也叫做初始化延迟或者视频缓存检验器 VBV 的缓存大小把它当成编码器和解码器比特流之间的缓存，在不影响视频质量的情况下可以将其设置得尽可能小也以降低延迟。
3. 如果是仅仅优化首开延迟，可以在视频帧间插入较多的关键帧，这样客户端收到视频流之后可以快解码。但如果需要优化传输过程中的累计延迟，尽可能少使用关键帧也就是 I 帧（GOP 变大），在证同等视频质量的情况下，I 帧越多，码率越大，传输所需的网络带宽越多，也就意味着累计延迟可越大。这个优化效果可能在秒级延迟的系统中不是很明显，但是在 100 ms 甚至更低延迟的系统中就非常明显。同时，尽量使用 ACC-LC Codec 来编码音频，HE-ACC 或者 HE-ACC 2 虽然编码效率高但是编码所需时间更长，而产生更大体积的音频造成的传输延迟对于视频流的传输来说影响更小。
4. 不要使用视频 MJPEG 的视频压缩格式，至少使用不带 B 帧的 MPEG4 视频压缩格式（Simple profile），甚至最好使用 H.264 baseline profile（X264 还有一个「-tune zerolatency」的优化开关）这样一个简单的优化可以降低延迟，因为它能够以更低的码率编码全帧率视频。
5. 如果使用了 FFmpeg，降低「-probesize」和「-analyze duration」参数的值，这两个值用于

帧信息监测和用于监测的时长，这两个值越大对编码延迟的影响越大，在直播场景下对于视频流来说 analyzeduration 参数甚至没有必要设定。

6. 固定码率编码 CBR 可以一定程度上消除网络抖动影响，如果能够使用可变码率编码 VBR 可以节省一些不必要的网络带宽，降低一定的延迟。因此建议尽量使用 VBR 进行编码。

## 传输协议优化

1. 在服务端节点和节点之间尽量使用 RTMP 而非基于 HTTP 的 HLS 协议进行传输，这样可以降低视频的传输延迟。这个主要针对终端用户使用 HLS 进行播放的情况。

2. 如果终端用户使用 RTMP 来播放，尽量在靠近推流端的收流节点进行转码，这样传输的视频流比原始视频流更小。

3. 如果有必要，可以使用定制的 UDP 协议来替换 TCP 协议，省去弱网环节下的丢包重传可以降低延迟。它的主要缺点在于，基于 UDP 协议进行定制的协议的视频流的传输和分发不够通用，CDN 厂商持的是标准的传输协议。另一个缺点在于可能出现丢包导致的花屏或者模糊（缺少关键帧的解码参考），这就要求协议定制方在 UDP 基础之上做好丢包控制。

## 传输网络优化

1. 我们曾经介绍过七牛直播云的 [实时流传输网络](#)，它是一种新型的节点自组织的网状传输网络，适合国内多运营商网络条件下的传输优化，也适合众多海外直播的需求。

2. 在服务端节点中缓存当前 GOP，配合播放器端优化视频首开时间。

3. 服务端实时记录每个视频流流向每个环节时的秒级帧率和码率，实时监控码率和帧率的波动。

4. 客户端（推流和播放）通过查询服务端准实时获取当前最优节点（5 秒一次），准实时下线当前障碍节点和线路。

## 推流、播放优化

1. 考察发送端系统自带的网络 buffer 大小，系统可能在发送数据之前缓存数据，这个参数的调优也要找到一个平衡点。

2. 播放端缓存控制对于视频的首开延迟也有较大影响，如果仅优化首开延迟，可以在 0 缓存情况下数据到达的时候立即解码。但如果在弱网环境下为了消除网络抖动造成的影响，设置一定的缓存也有必要，因此需要在直播的稳定性和首开延迟优化上找到平衡，调整优化缓冲区大小这个值。

3. 播放端动态 buffer 策略，这是上面播放端缓存控制的改进版本。如果只是做 0 缓存和固定大小缓存之间进行选择找到平衡，最终还是会选择固定大小的缓存，这对亿级的移动互联网终端用户并不公平，他们不同的网络状况决定了这个固定大小的缓存并不完全合适。因此，我们可以考虑「动态 buffer 策略」，在播放器开启的时候采用非常小甚至 0 缓存的策略，通过对下载首片视频的时来决定下一个时间片的缓存大小，同时在播放过程中实时监测当前网络，实时调整播放过程中缓存大小。这样即可做到极低的首开时间，又可能尽量消除网络抖动造成的影响。

4. 动态码率播放策略。除了动态调整 buffer 大小的策略之外，也可以利用实时监测的网络信息来动态调整播放过程中的码率，在网络带宽不足的情况下降低码率进行播放，减少延迟。

以上，是我们在低延迟优化方面的部分技巧。实际上我们优化低延迟的时候并不是只关注「低延迟」而是在保证其它条件不影响用户体验的情况下尽量做到低延迟，因此它的内容涉及到更多广泛的话题而视频直播的优化也包含方方面面，这里只分享了其中经过我们实践的部分。随着实践的积累，我们

下来会在线上和\*\* 线下分享更多关于视频直播甚至点播的优化技巧。

### 【没看过瘾？直接来上免费公开课】

为了让大家能够将技术理论快速应用到实践开发中，七牛云联合慕课网、StuQ 特别制作了一期课程专门针对移动直播应用开发，供大家学习参考。



慕课网：

<http://www.imooc.com/learn/707>

StuQ：

<http://www.stuq.org/course/detail/1077>