

ArchSummit全球架构师峰会 深圳站2016

移动直播技术上的坑与优化经验

七牛直播云负责人 徐立



促进软件开发领域知识与创新的传播



关注InfoQ官方微信
及时获取ArchSummit
大会演讲信息

QCon

全球软件开发大会

[上海站] 2016年10月20-22日

咨询热线: 010-64738142

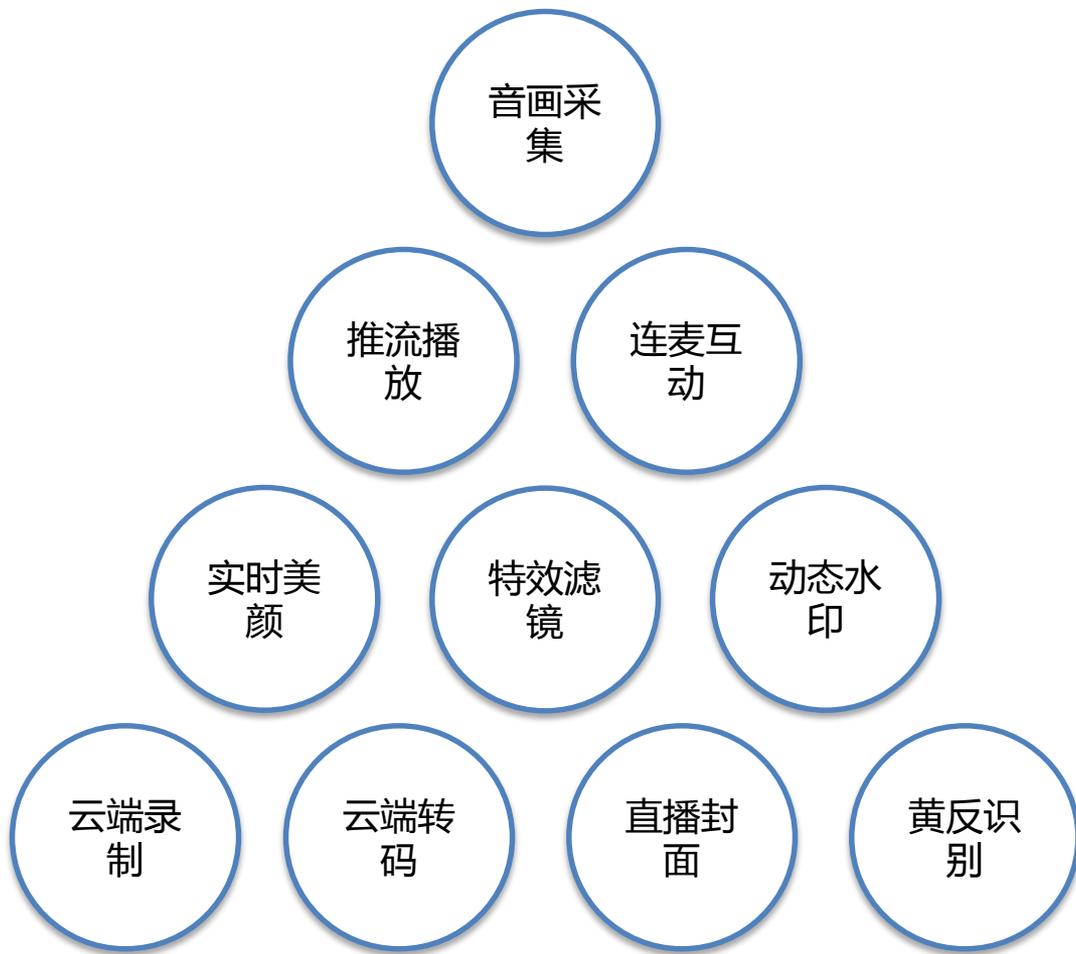
ArchSummit

全球架构师峰会 2016

[北京站] 2016年12月2-3日

咨询热线: 010-89880682

移动直播常见诉求



移动直播常见问题

- 推流发烫、崩溃、闪退，尤其是安卓
- 开启美颜，推流发烫
- 开启美颜和水印，推流发烫、画面卡顿
- 主播手机配置较差，推流发烫、观看画面卡顿
- 观众大量点赞“飘星”、送花，导致主播和观众画面卡顿
- 直播推流过程中接听电话、切换应用，导致丢失声音
- 延时较高，播放时间越长，延时越大
- 播放首屏打开慢，超过数秒打不开，有时黑屏、卡住
- 播放过程中间歇性卡顿
- 点击一个正在进行的直播，提示直播已经结束
-

第一个问题：首开耗时与用户感受

首屏打开时间	用户感受
0 ~100 ms	很好，很快
100~300 ms	还好，一般
300~1000 ms	略慢，再等等看
> 1000 ms	这么慢，我要切应用了
> 10000 ms	嘛，服务器宕机了么

首开耗时对业务运营的影响

- 必须 250 ms 内渲染页面，或者至少提供视觉反馈，才能保证用户不走开！
- 谷歌、微软和亚马逊的研究都表明，性能可以直接转换成收入。比如，Bing搜索网页时延迟 2000 ms 会导致每用户收入减少 4.3%。
- 类似地，一项覆盖 160 多家组织的研究表明，界面加载时间增加 1 秒，会导致转化率损失 7%，浏览量减少 11%，用户满意度降低 16%！

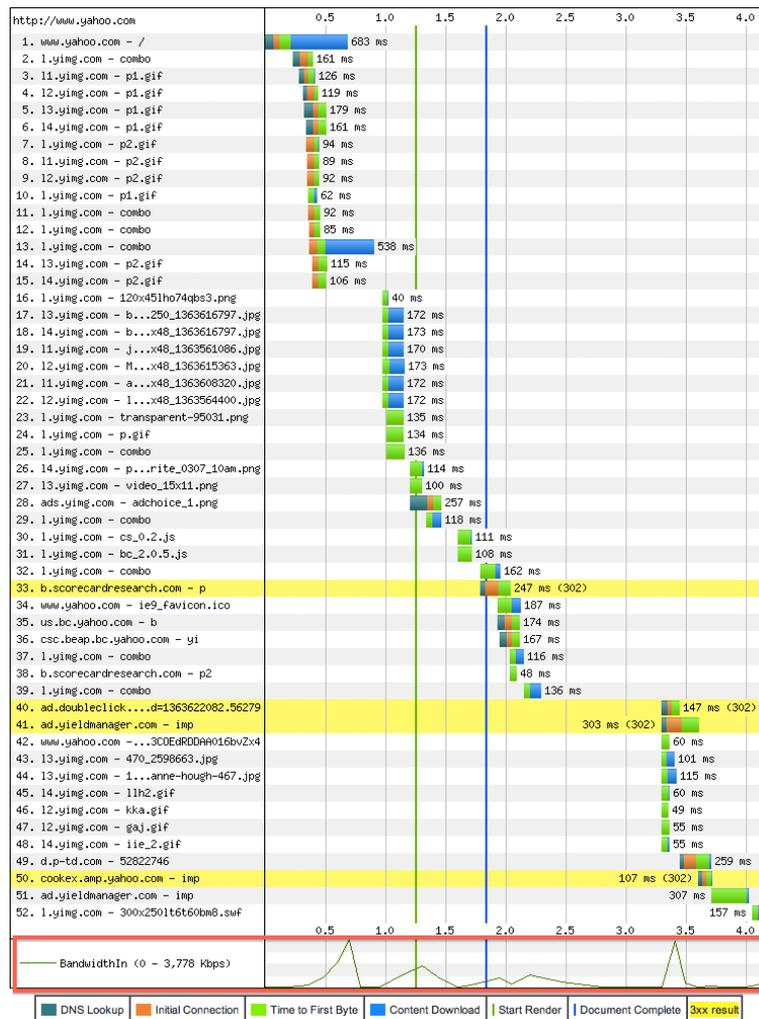
一个HTTP请求不为人知的背后



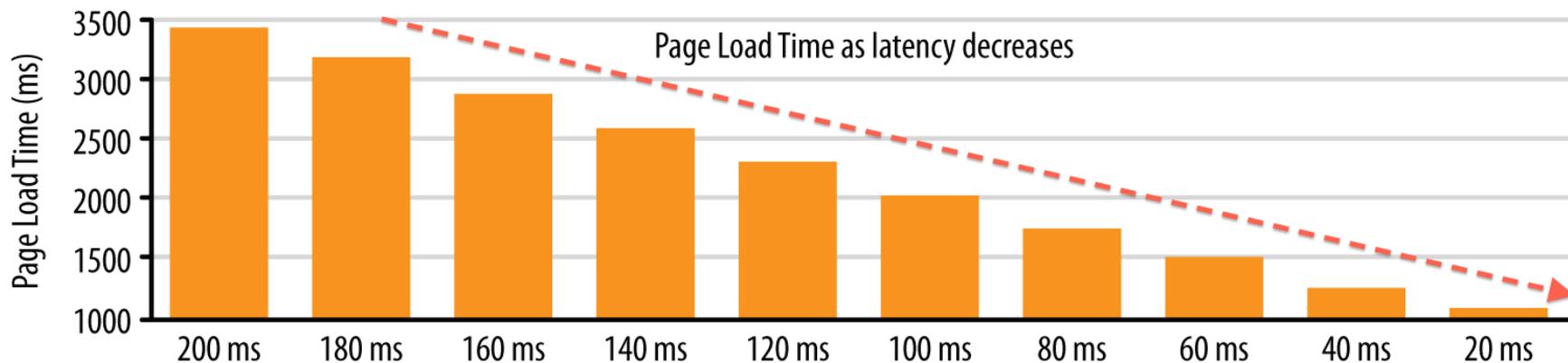
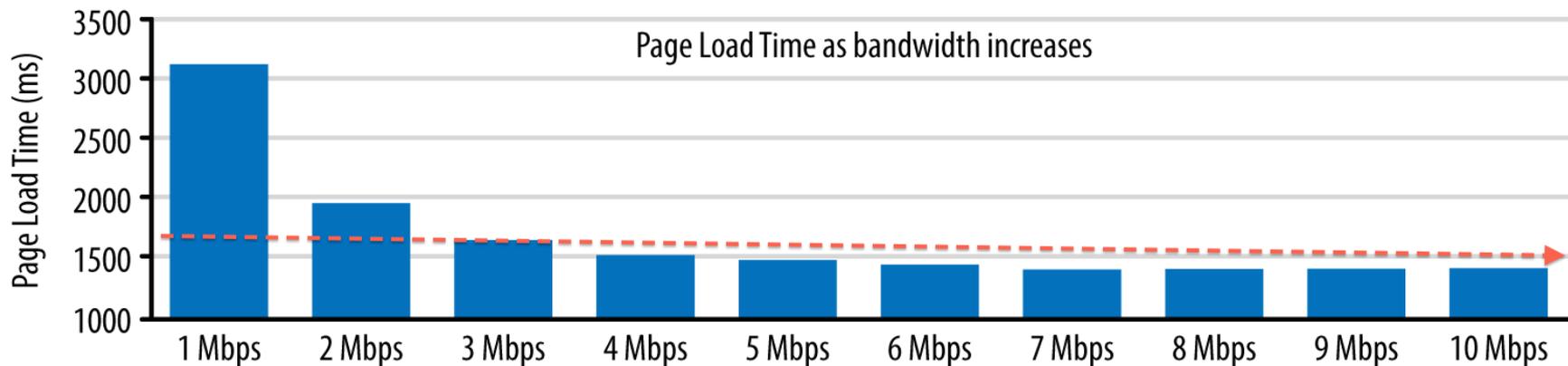
- 打开一个网页花了 683 ms
- 其中超过 200 ms 在等待网络就绪 (占到了请求耗时的30%)
- 1/3 时间用于准备网络 , 2/3 时间用于下载内容
- 为什么

性能剖析

- 1 个网页中附加 52 个资源
- 总大小 486 KB
- 然而，总耗时 4 秒(s)
- 不过，带宽利用却极低！



数据真相



数据结论

- 带宽 (Bandwidth) 并不是影响首开的根本原因
- 延迟 (Latency) 才是影响首开的性能瓶颈
- 要快，必须缩短 RTT (Round-Trip Time ，往返延时)

- 250ms 是 PC时代的网页加载的体验要求
- 移动互联网时代，内容消费升级 (文字、图片、视频、直播) ，内容越来越丰富 (体积越来越大) ，视频直播如何实现 “秒开” ？

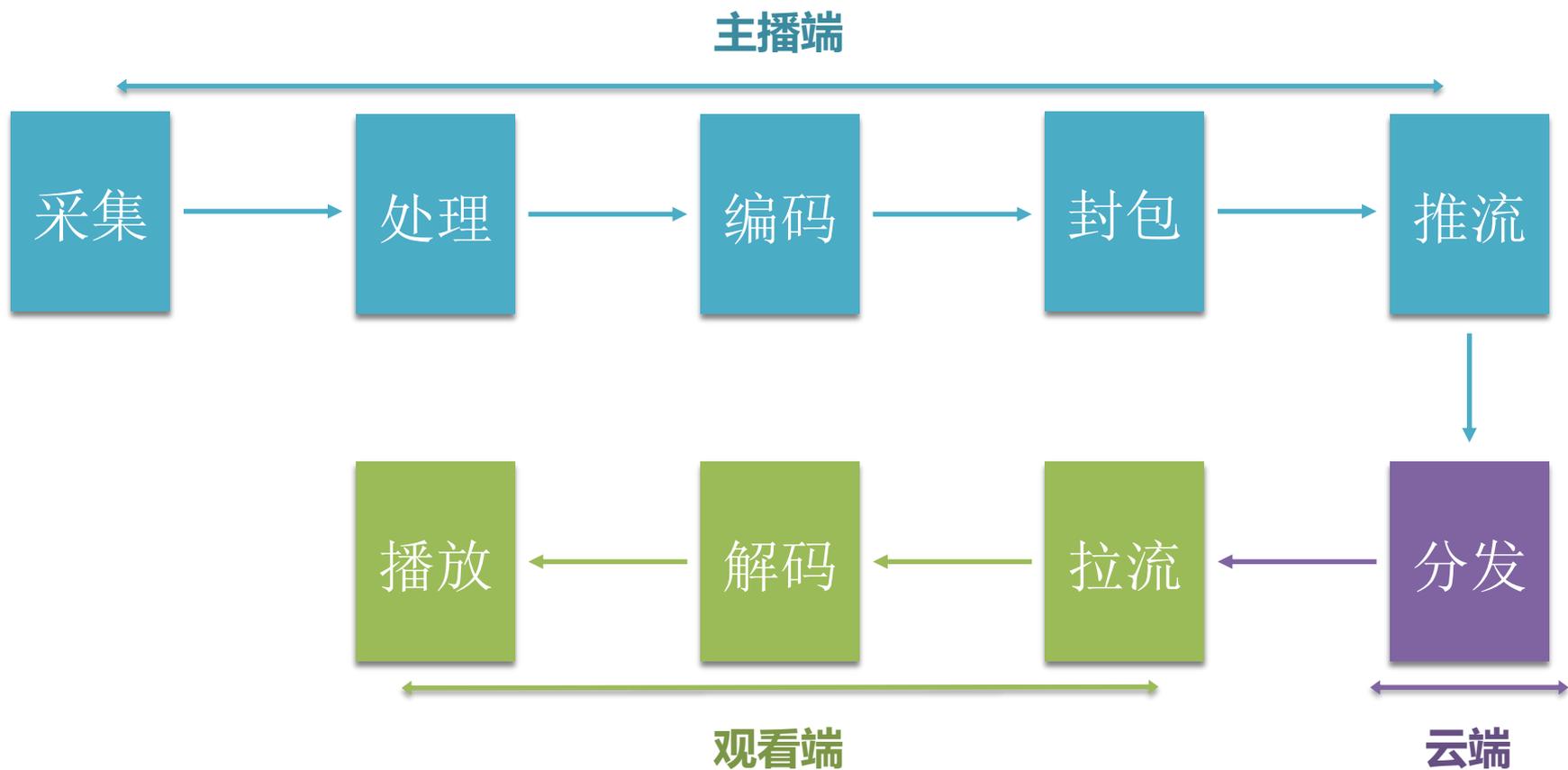
直播性能要素

首屏秒
开

流畅
不卡顿

低延时

APP直播流程



各环节资源消耗对比

	CPU/GPU消耗	I/O消耗
采集	低	低
处理	高	低
编码	高	低
封包	低	低
推流	低	高
分发	低	高
拉流	低	高
解码	高	低
播放	高	低

各环节资源消耗对比

	CPU/GPU消耗	I/O消耗	是否影响直播首开
采集	低	低	
处理	高	低	
编码	高	低	
封包	低	低	
推流	低	高	
分发	低	高	
拉流	低	高	是
解码	高	低	是
播放	高	低	

各环节资源消耗对比

	CPU/GPU消耗	I/O消耗	是否影响直播首开	是否可能造成直播卡顿
采集	低	低		
处理	高	低		
编码	高	低		
封包	低	低		
推流	低	高		是
分发	低	高		是
拉流	低	高	是	是
解码	高	低	是	
播放	高	低		

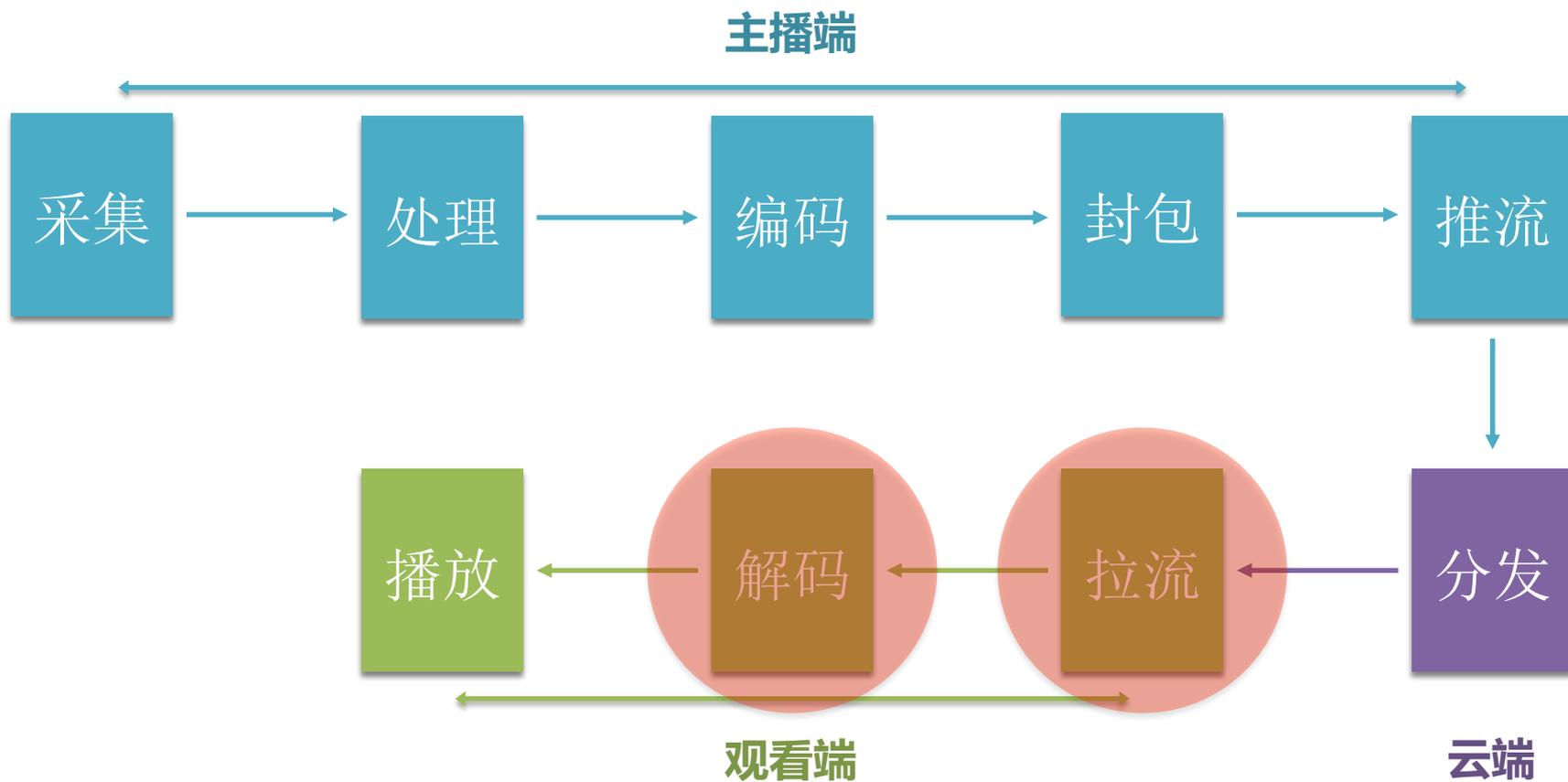
各环节资源消耗对比

	CPU/GPU消耗	I/O消耗	是否影响直播首开	是否可能造成直播卡顿	是否影响直播延时
采集	低	低			
处理	高	低			是
编码	高	低			是
封包	低	低			
推流	低	高		是	是
分发	低	高		是	是
拉流	低	高	是	是	是
解码	高	低	是		是
播放	高	低			

挑战一：如何实现



影响直播首开的环节



播放流程



播放为何不能秒开

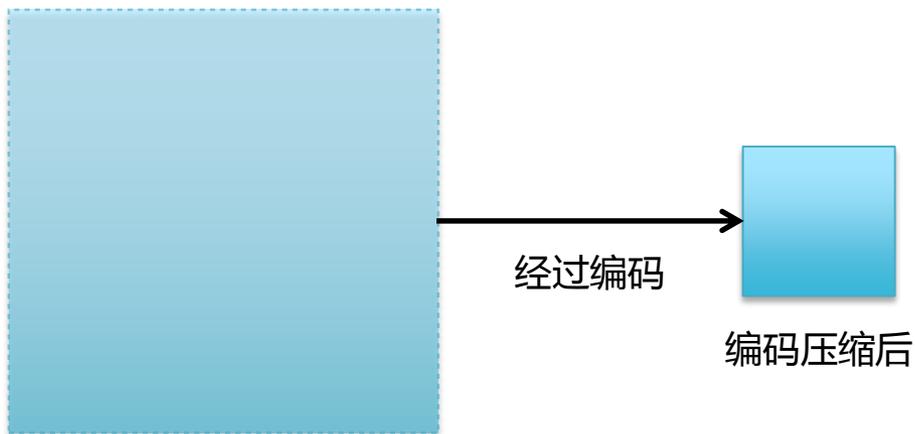
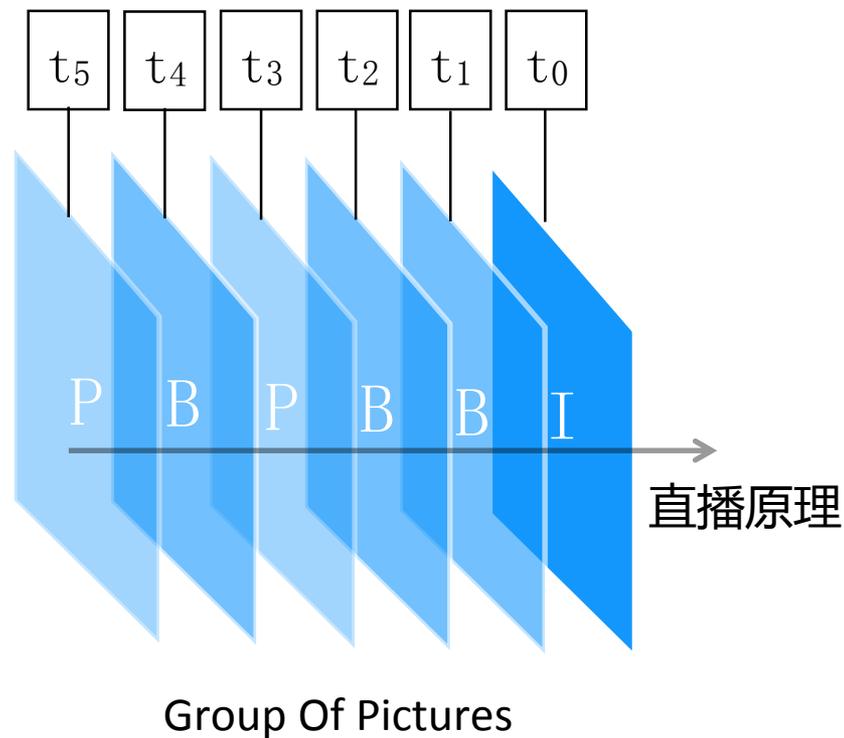
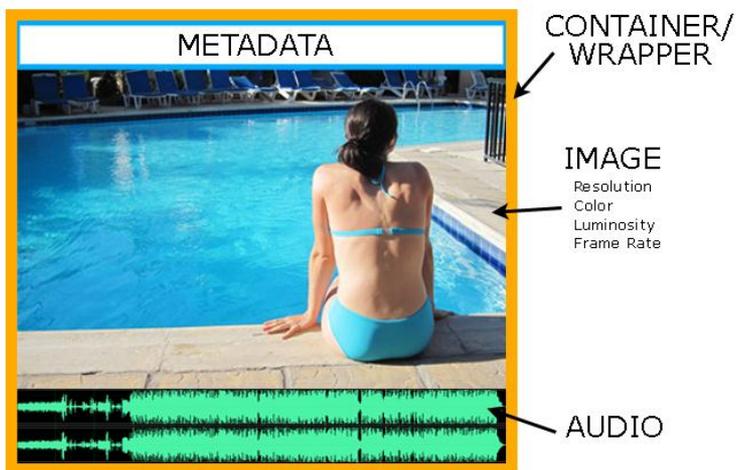
服务端：

1. 边缘节点没有GOP缓存，每次播放请求过来都需要回源，导致播放响应时间过长

客户端：

2. 只能下载整段GOP才能播放，不能识别GOP里边的首个关键帧进行解码播放
3. 等待音画时间戳同步，或者直到超时

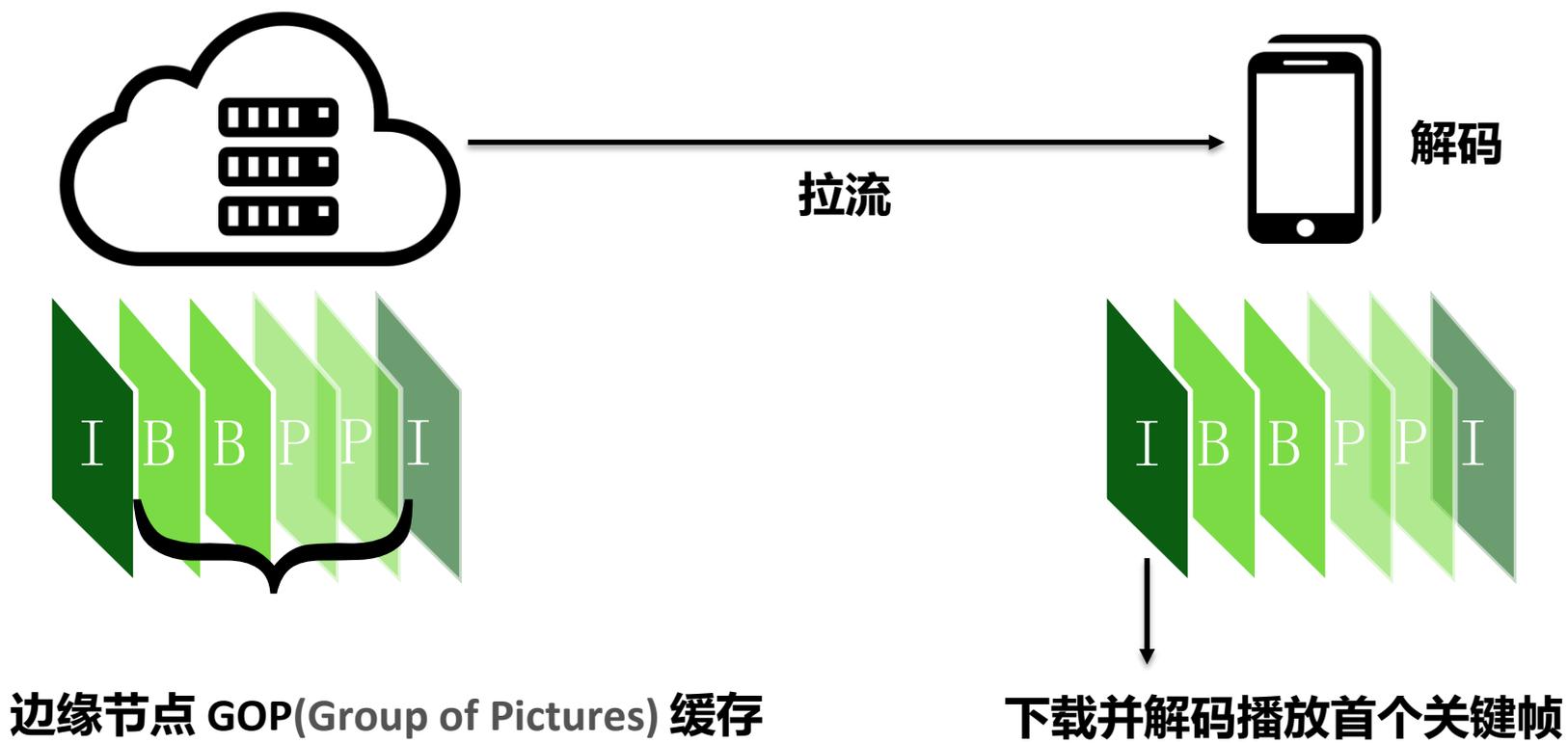
什么是 GOP ?



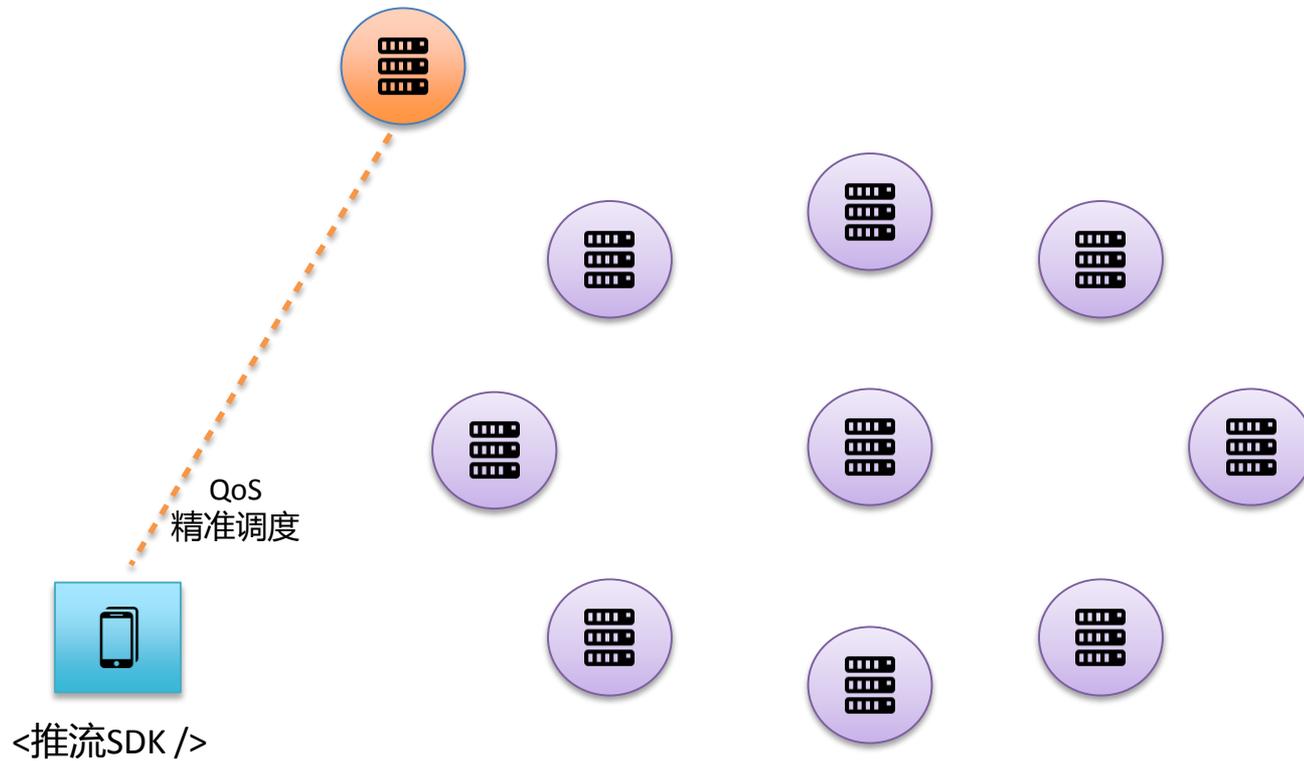
原始数据

编码压缩后

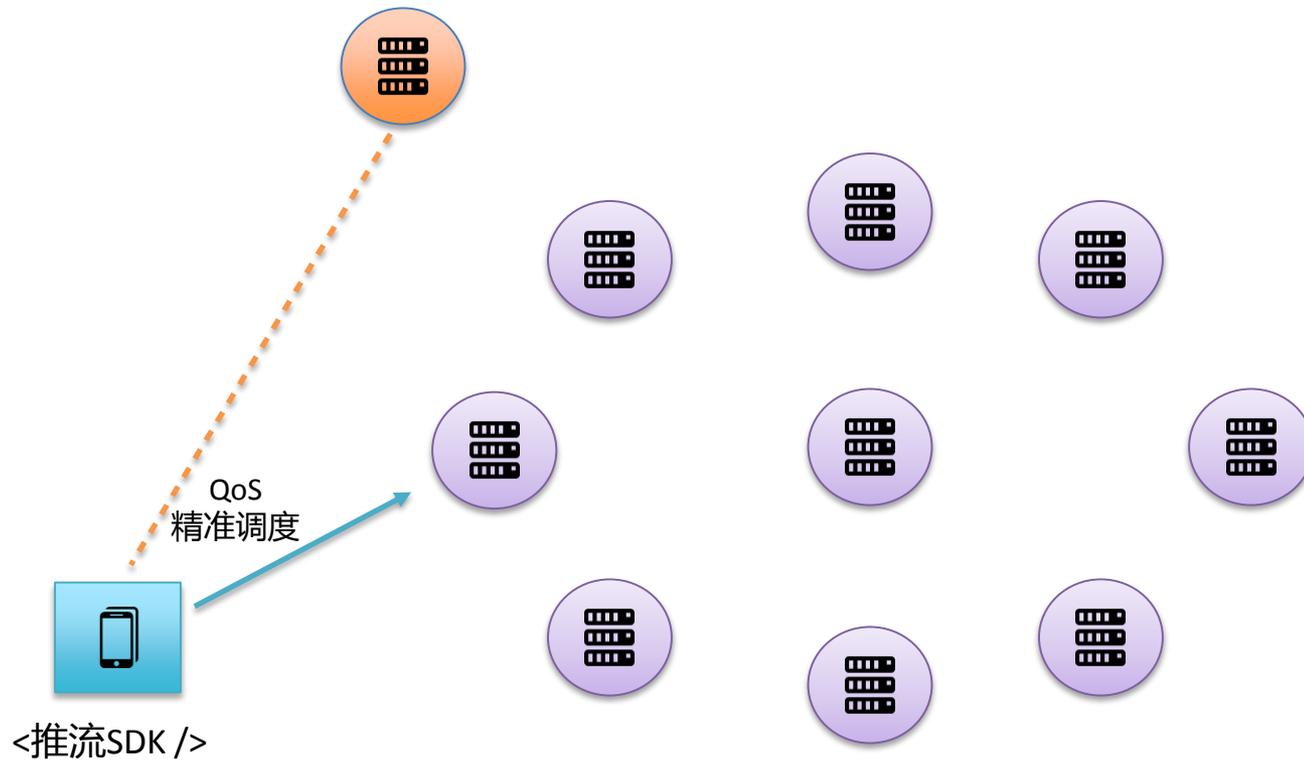
如何实现直播秒开



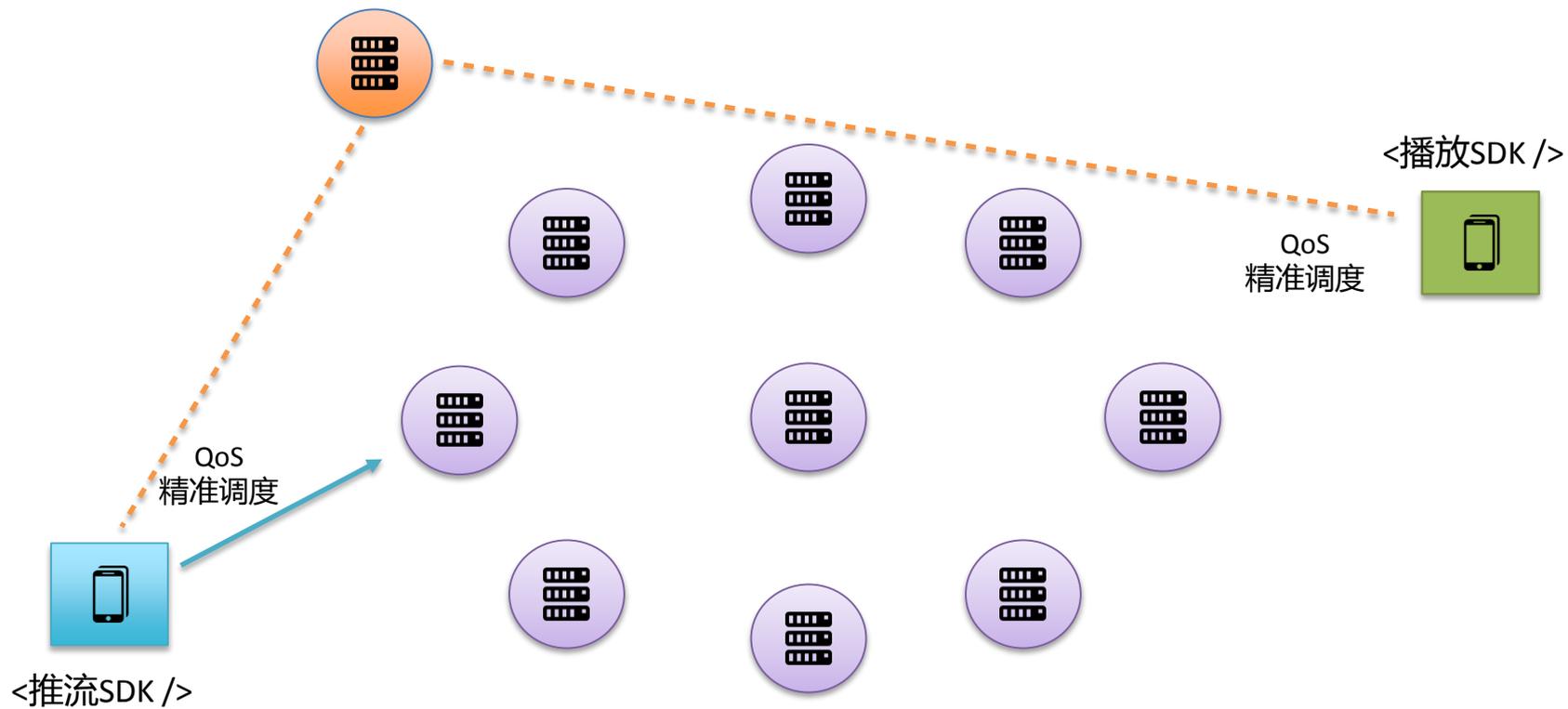
精准调度确定最佳接入点，缩短RTT便于秒开



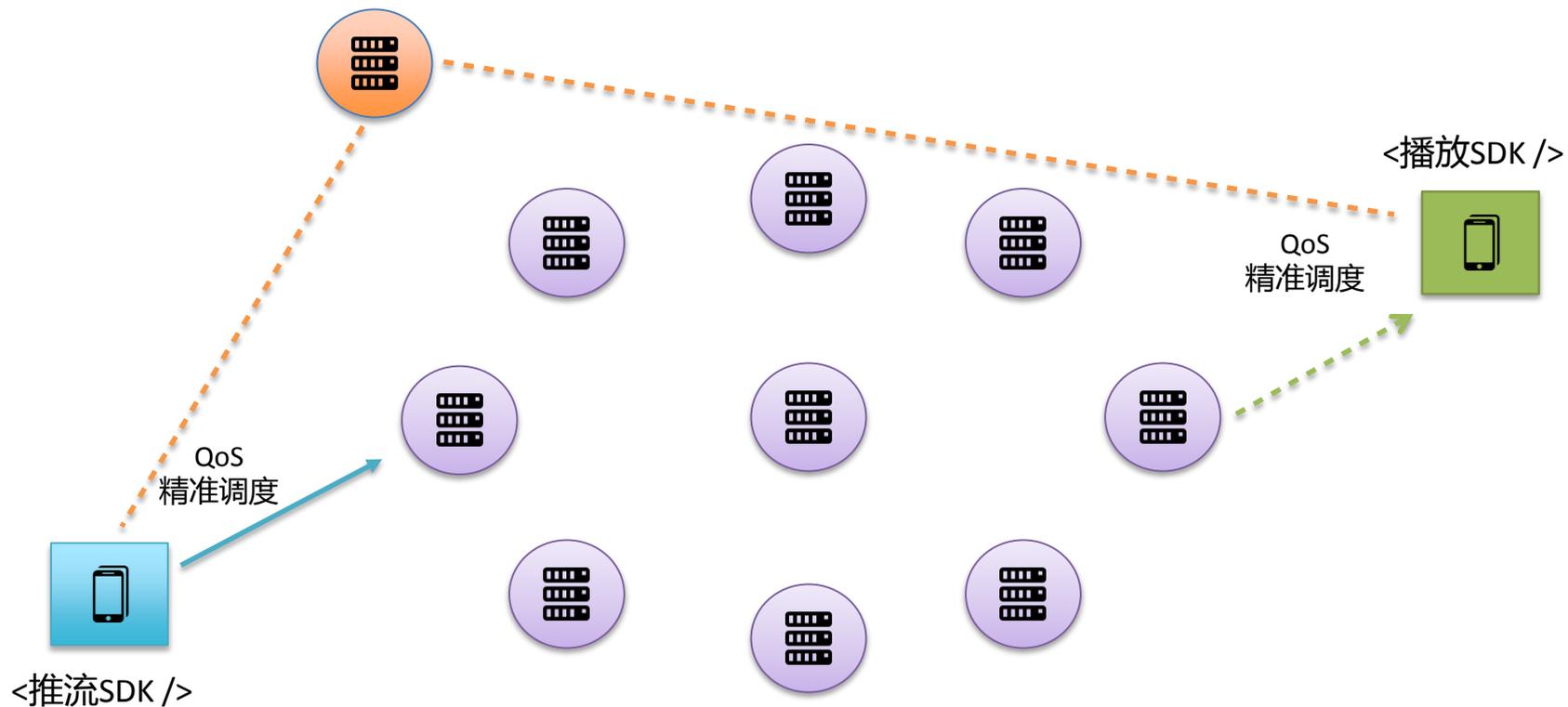
精准调度确定最佳接入点，缩短RTT便于秒开



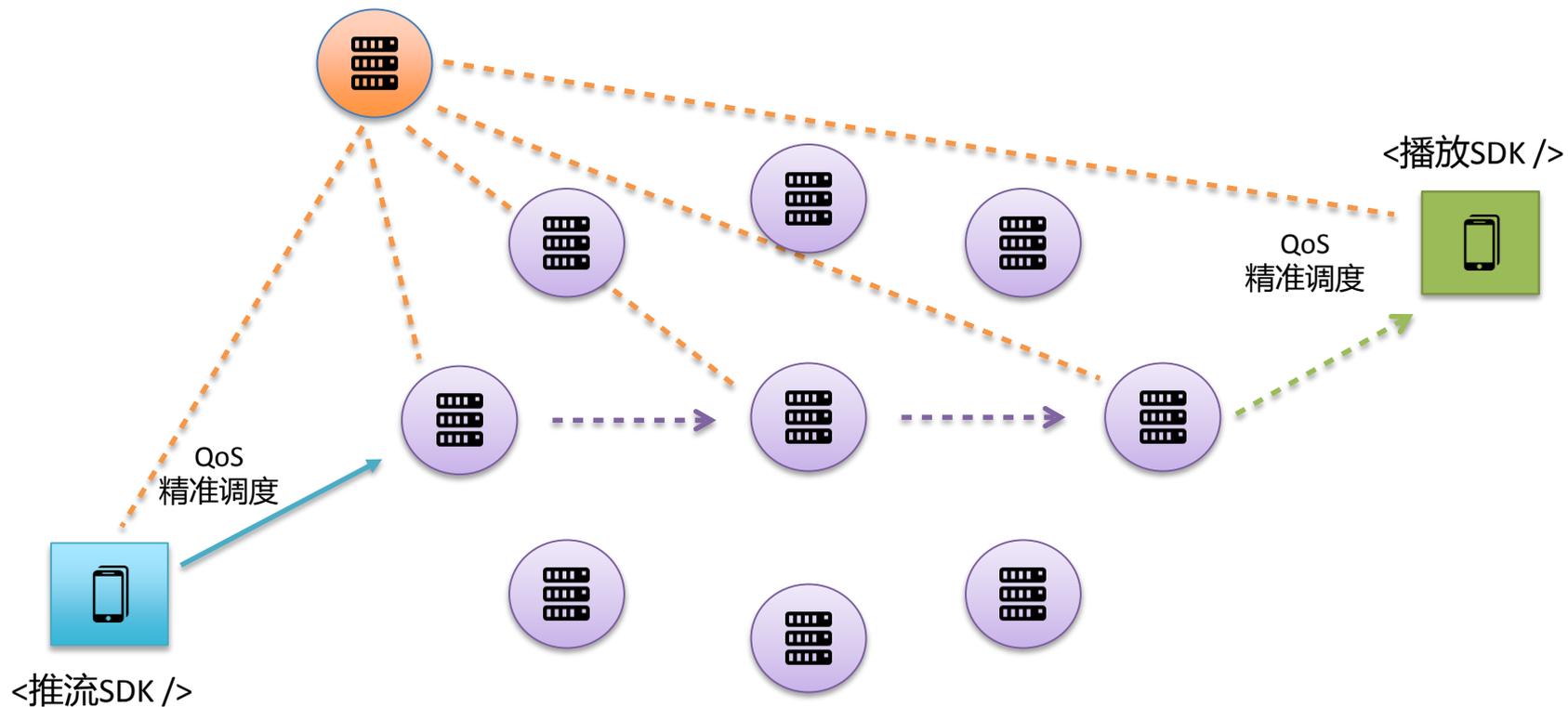
精准调度确定最佳接入点，缩短RTT便于秒开



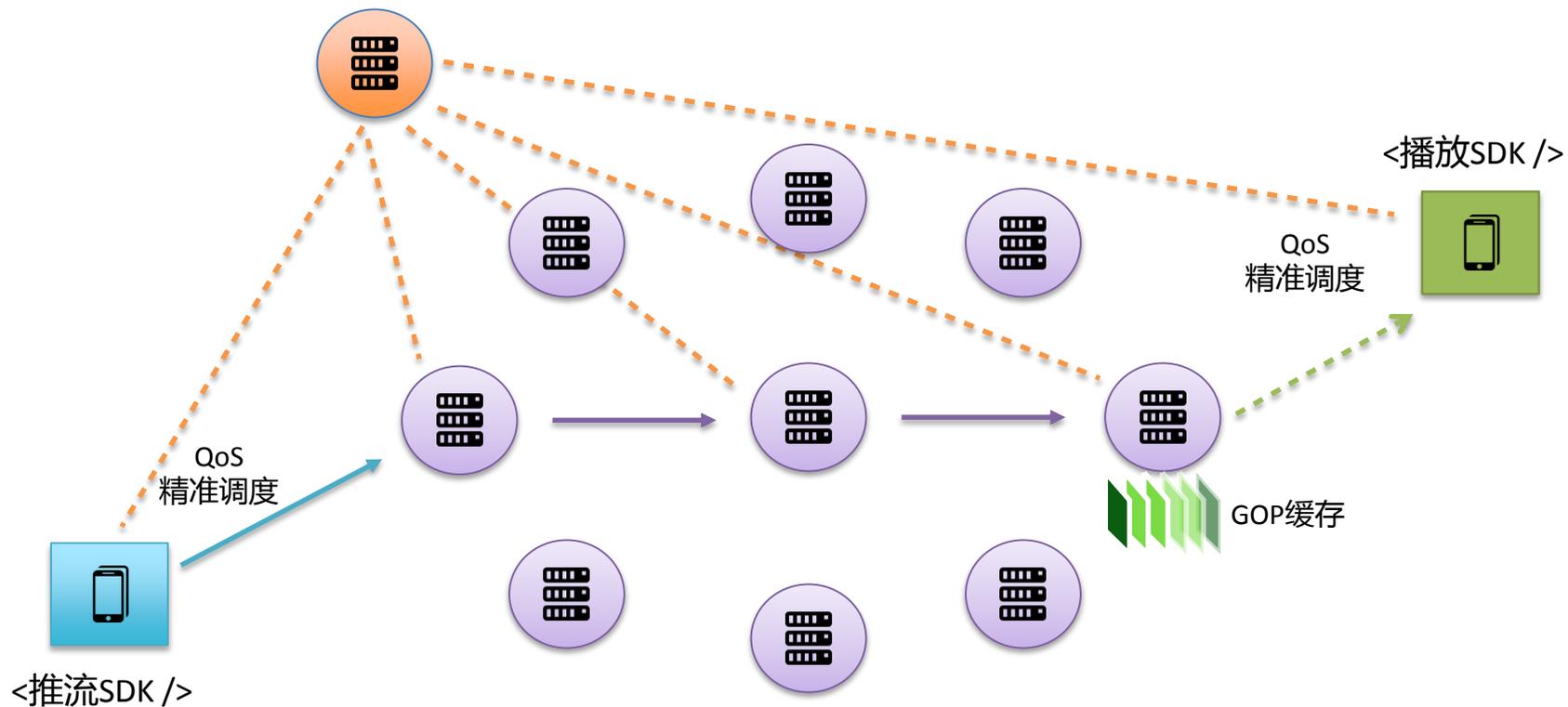
精准调度确定最佳接入点，缩短RTT便于秒开



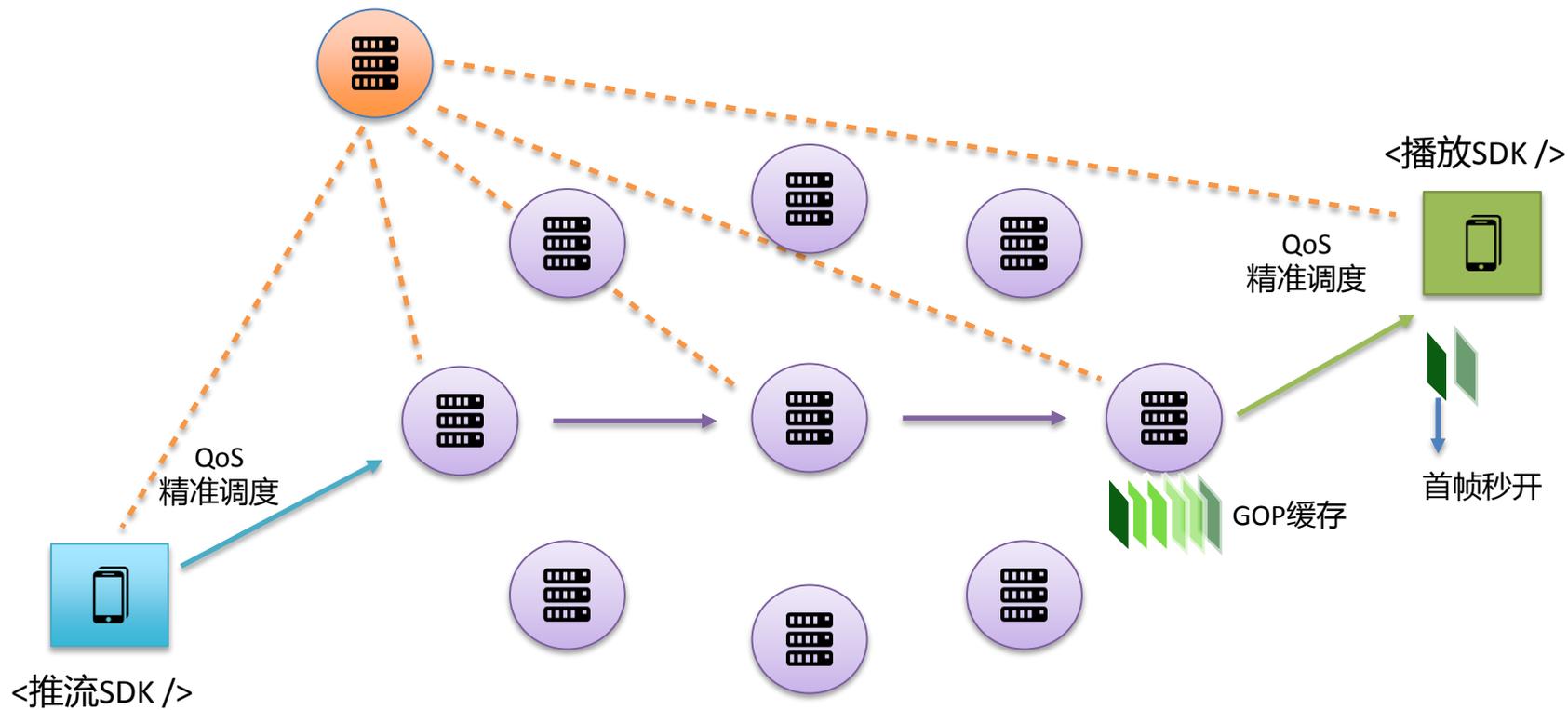
精准调度确定最佳接入点，缩短RTT便于秒开



播放请求时预判最佳路由，并主动触发预热GOP



播放SDK取边缘GOP缓存，解码首帧实现秒开



秒开经验总结

服务端：

1. 边缘节点缓存GOP，每次播放请求过来无需回源，省却回源时间
2. 主动预热GOP，省却来自边缘节点首次播放请求的路由和回源时间

客户端：

3. 无需下载整段GOP，下载关键帧后立即予以渲染显示
4. 画面首帧渲染环节避免音画时间戳同步校验

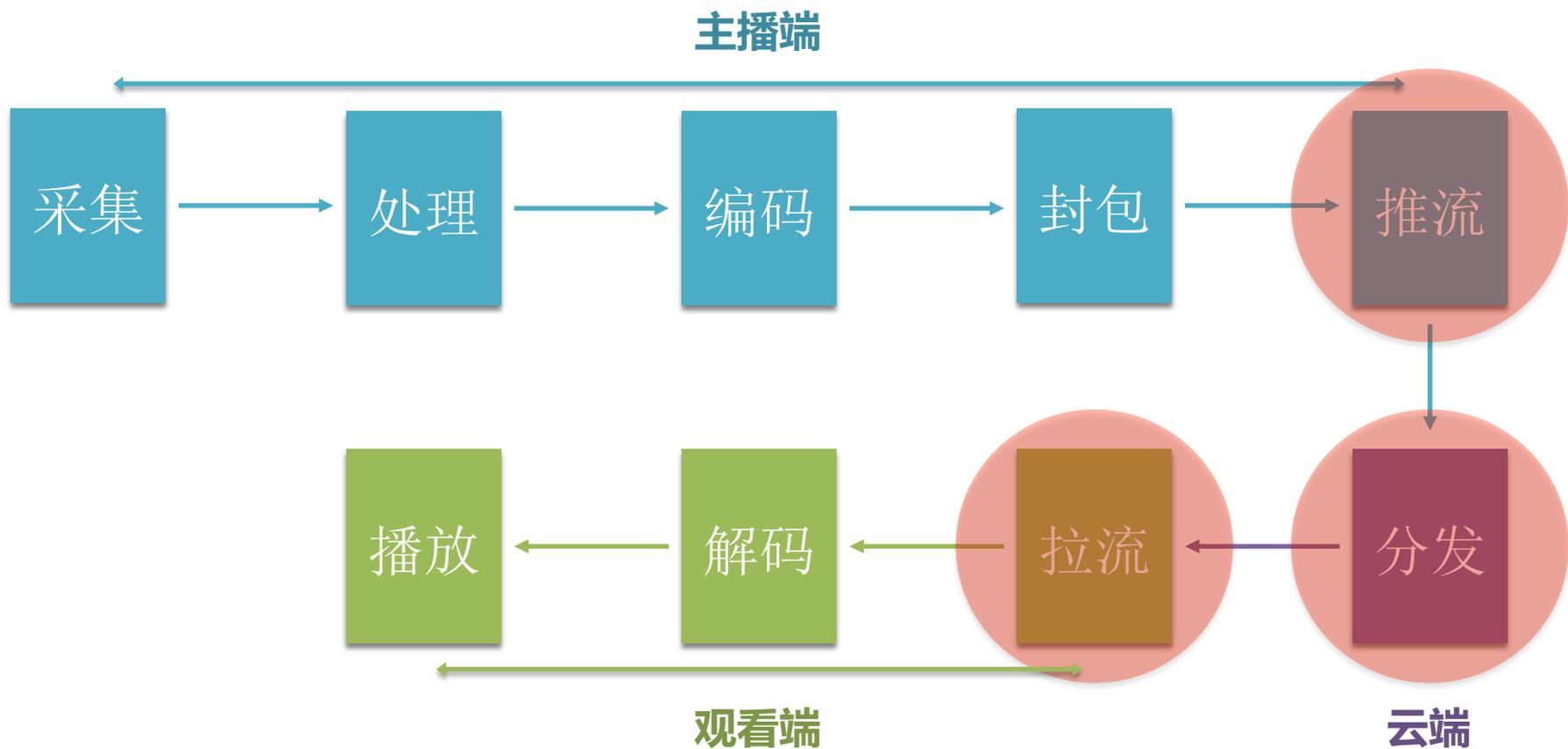
挑战二：如何实现

首屏秒
开

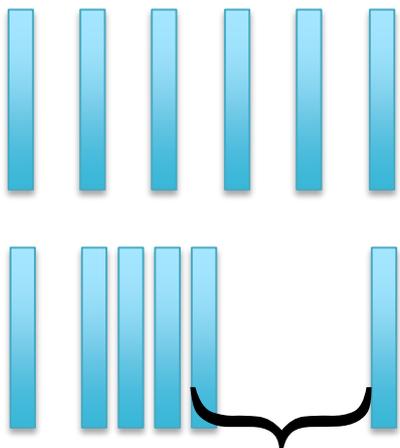
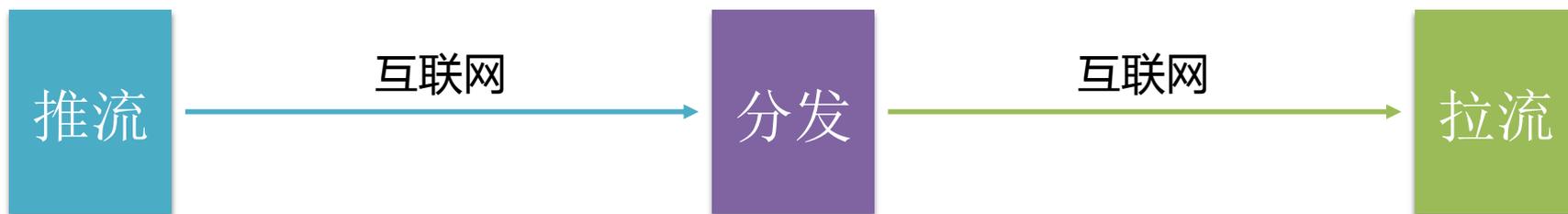
流畅
不卡顿

低延时

影响直播卡顿的环节

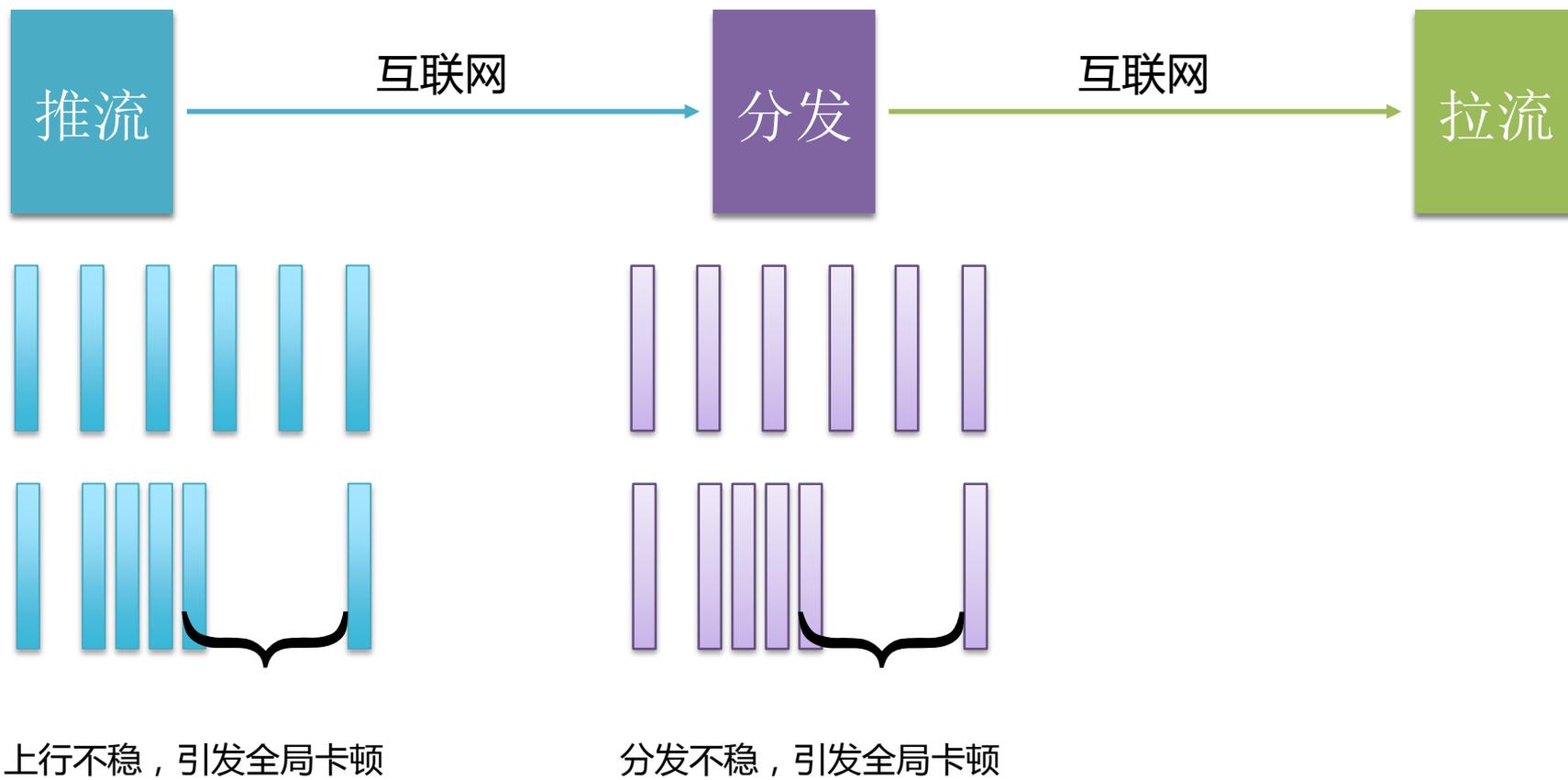


直播卡顿原因

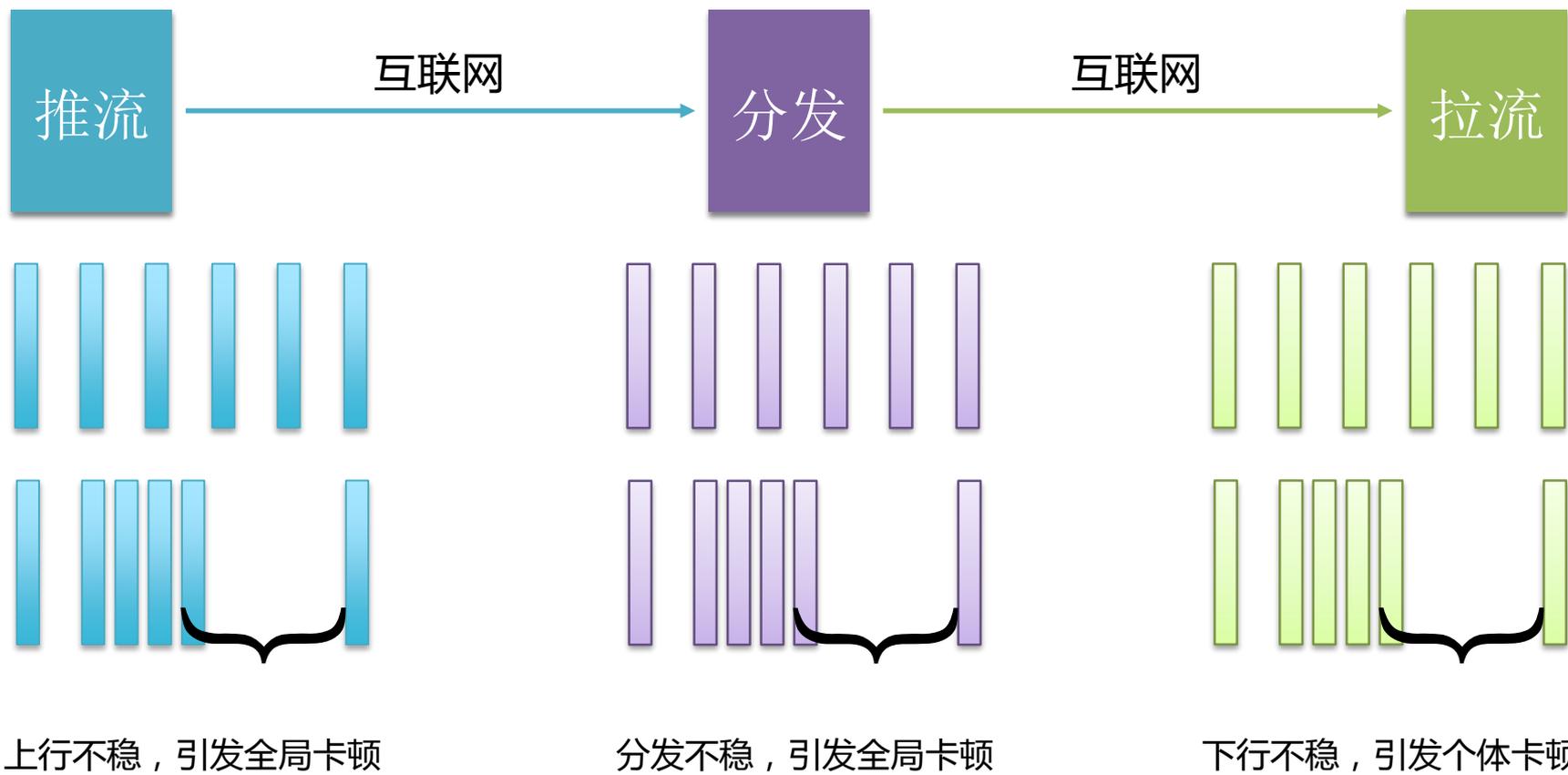


上行不稳，引发全局卡顿

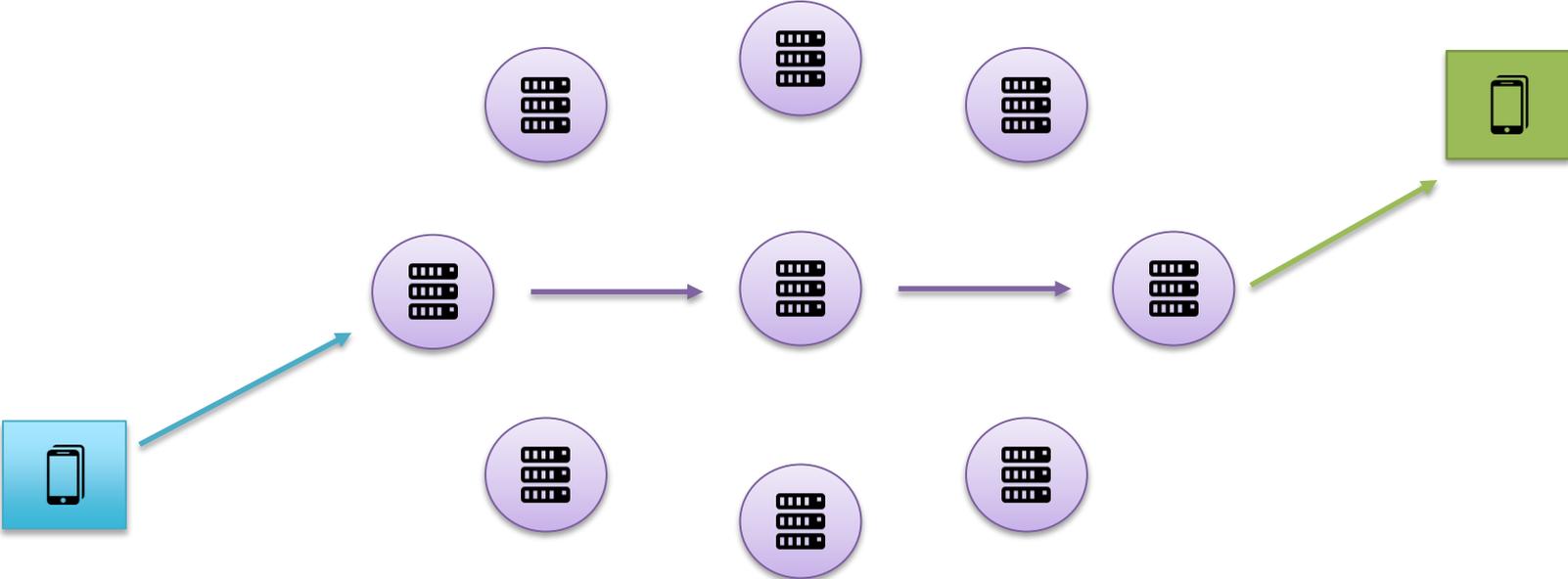
直播卡顿原因



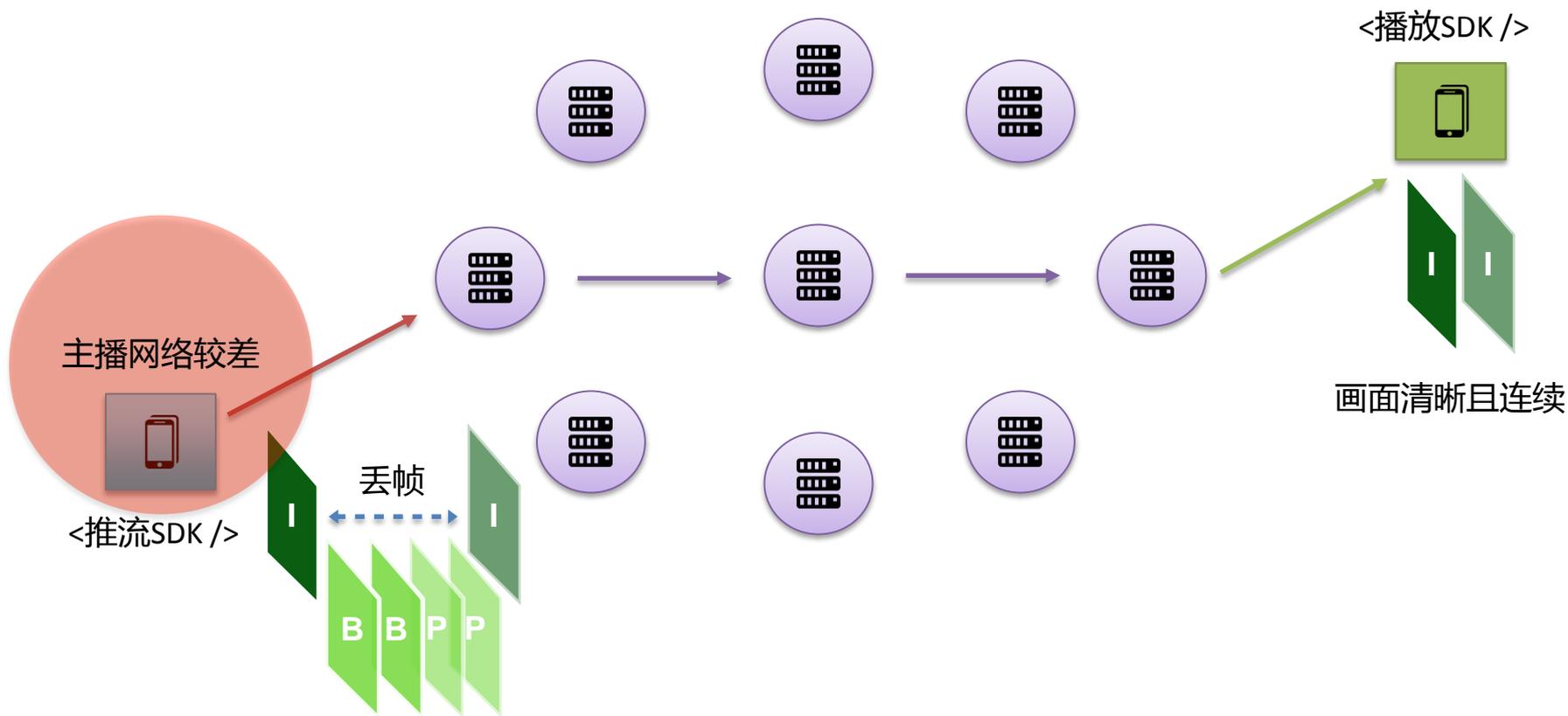
直播卡顿原因



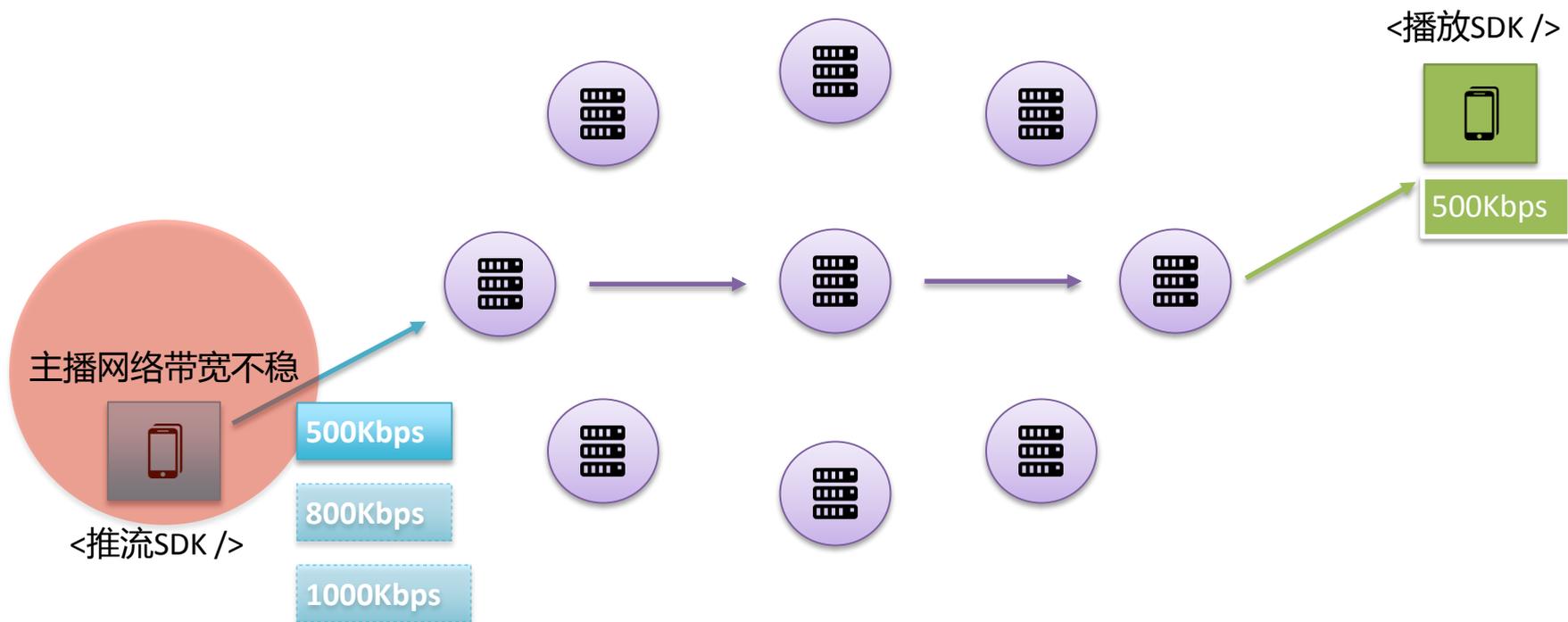
正常 推流、分发、拉流



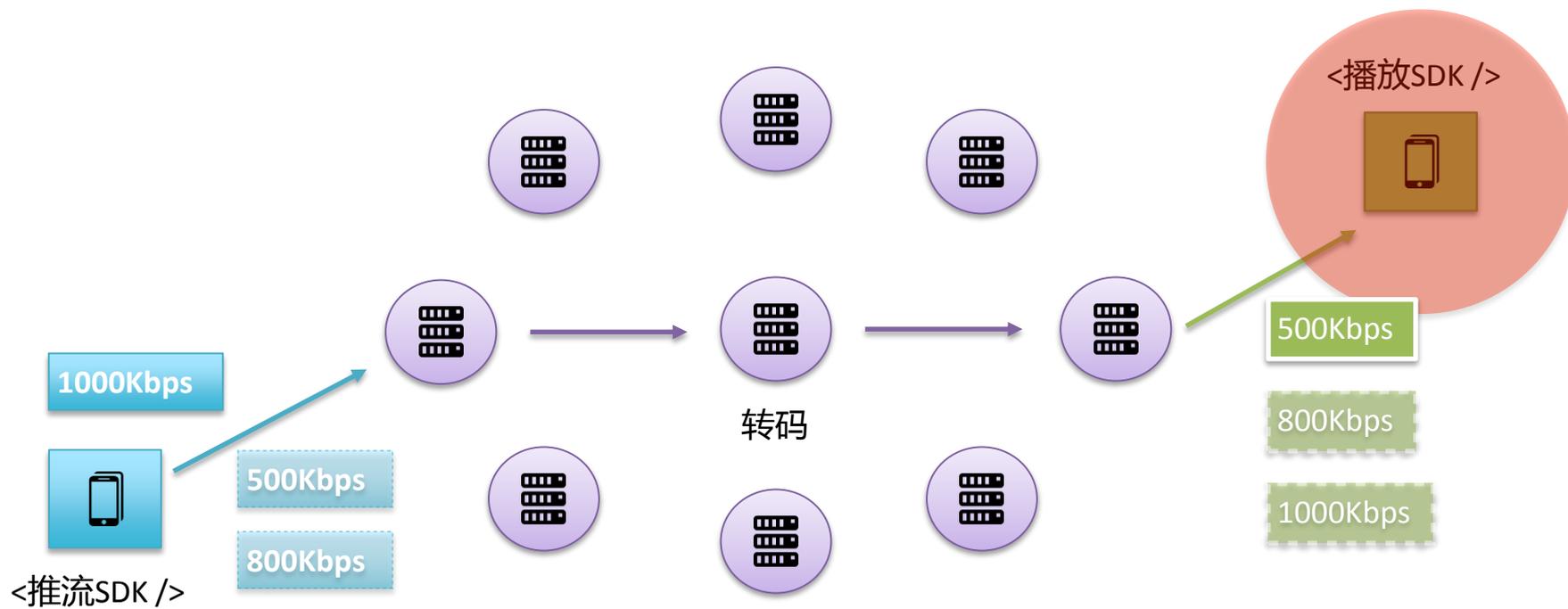
当主播网络质量较差时，进行弱网丢帧



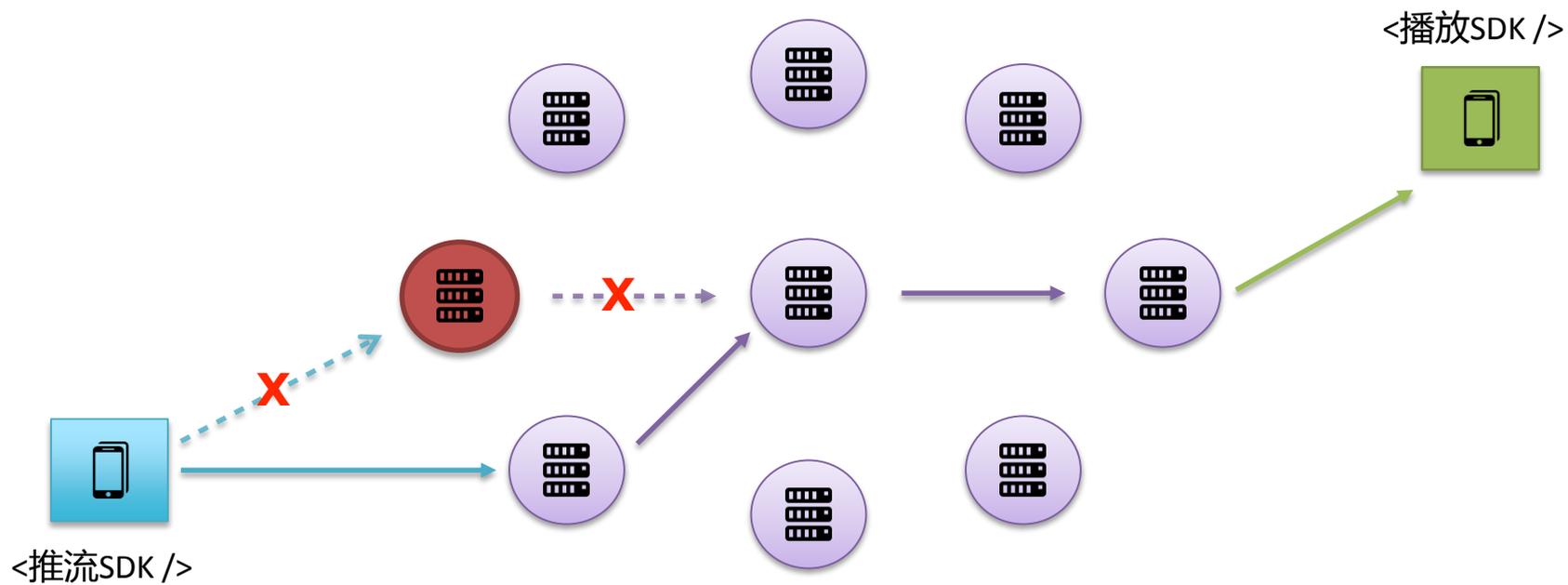
当主播网络带宽不稳时，动态调节推流码率



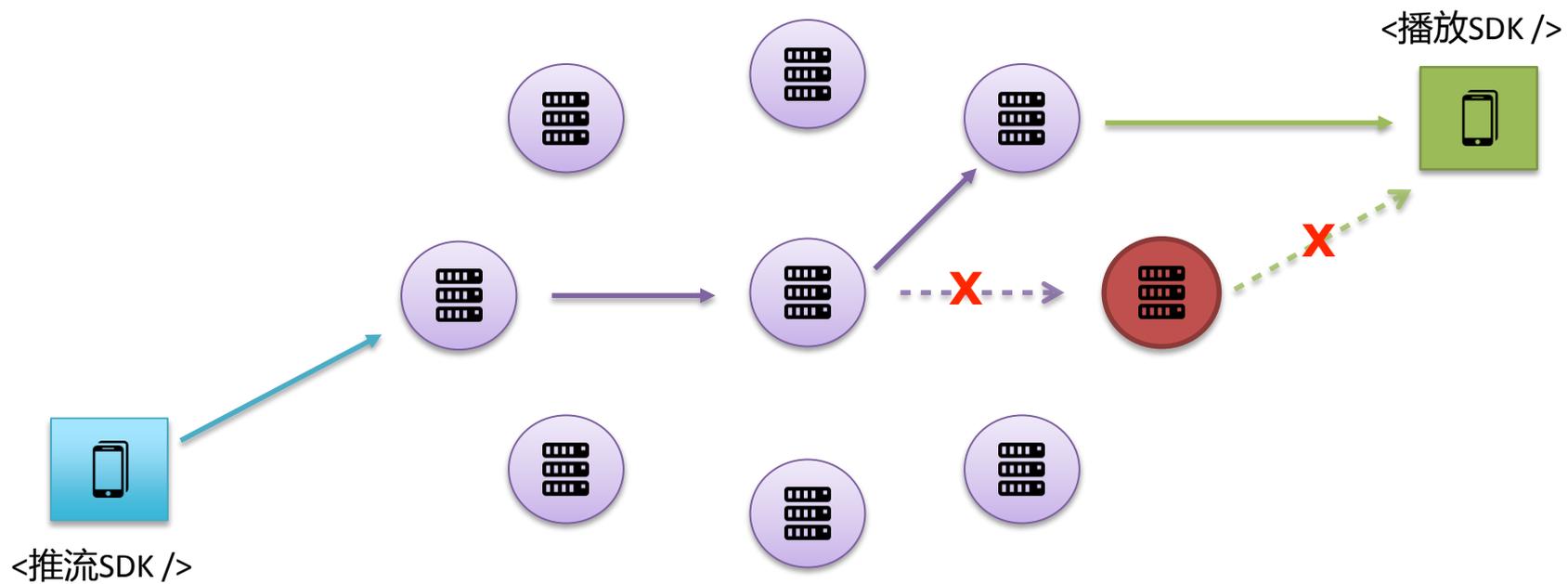
当观众网络带宽不稳时，播放端动态切换码率



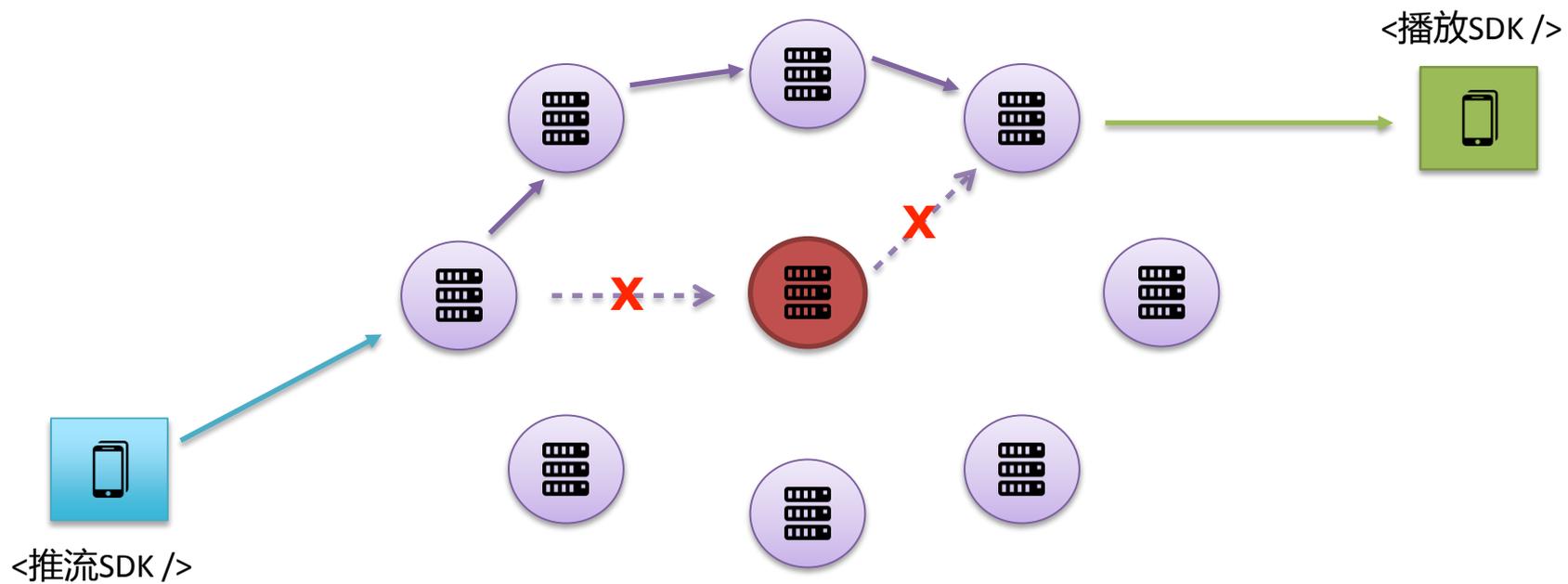
故障容错，线路动态切换，保障直播流畅稳定



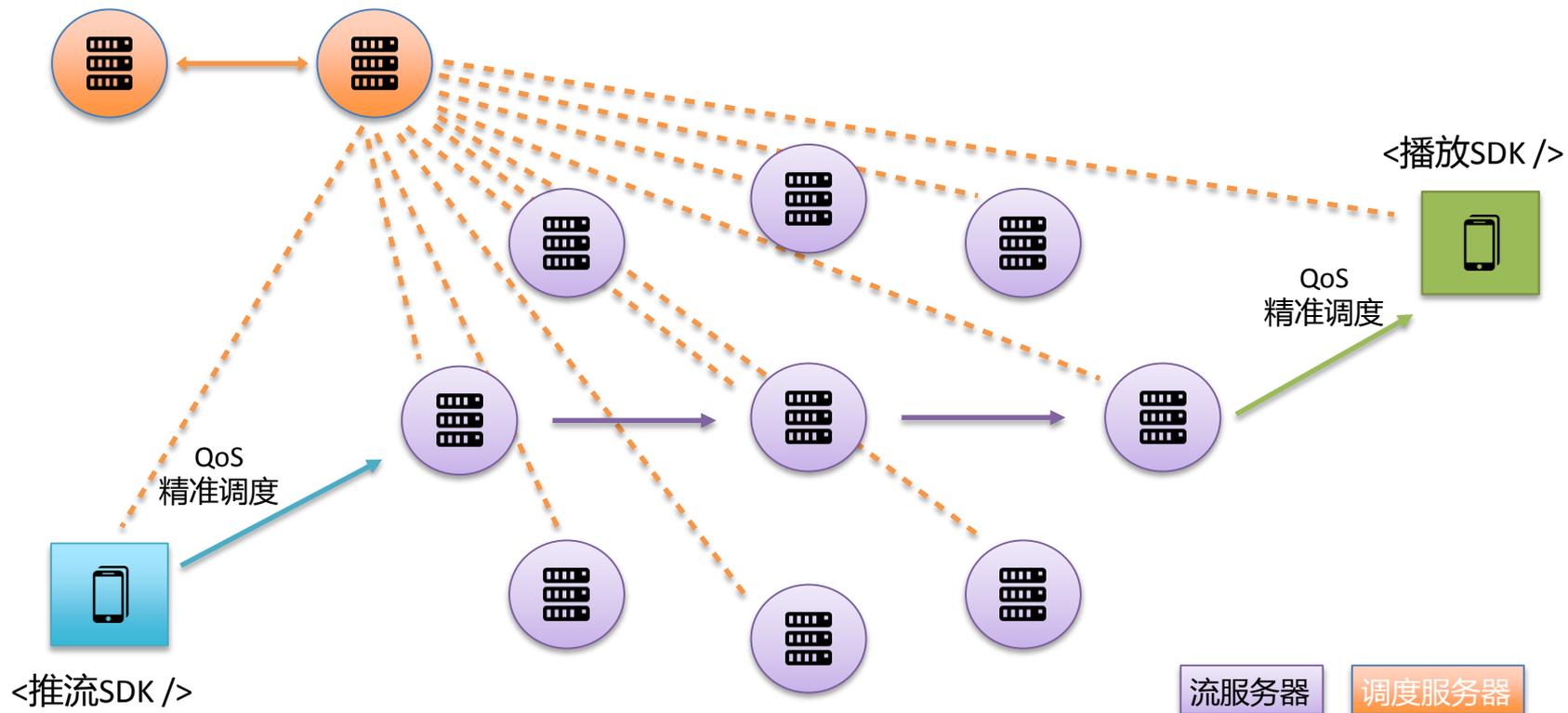
故障容错，线路动态切换，保障直播流畅稳定



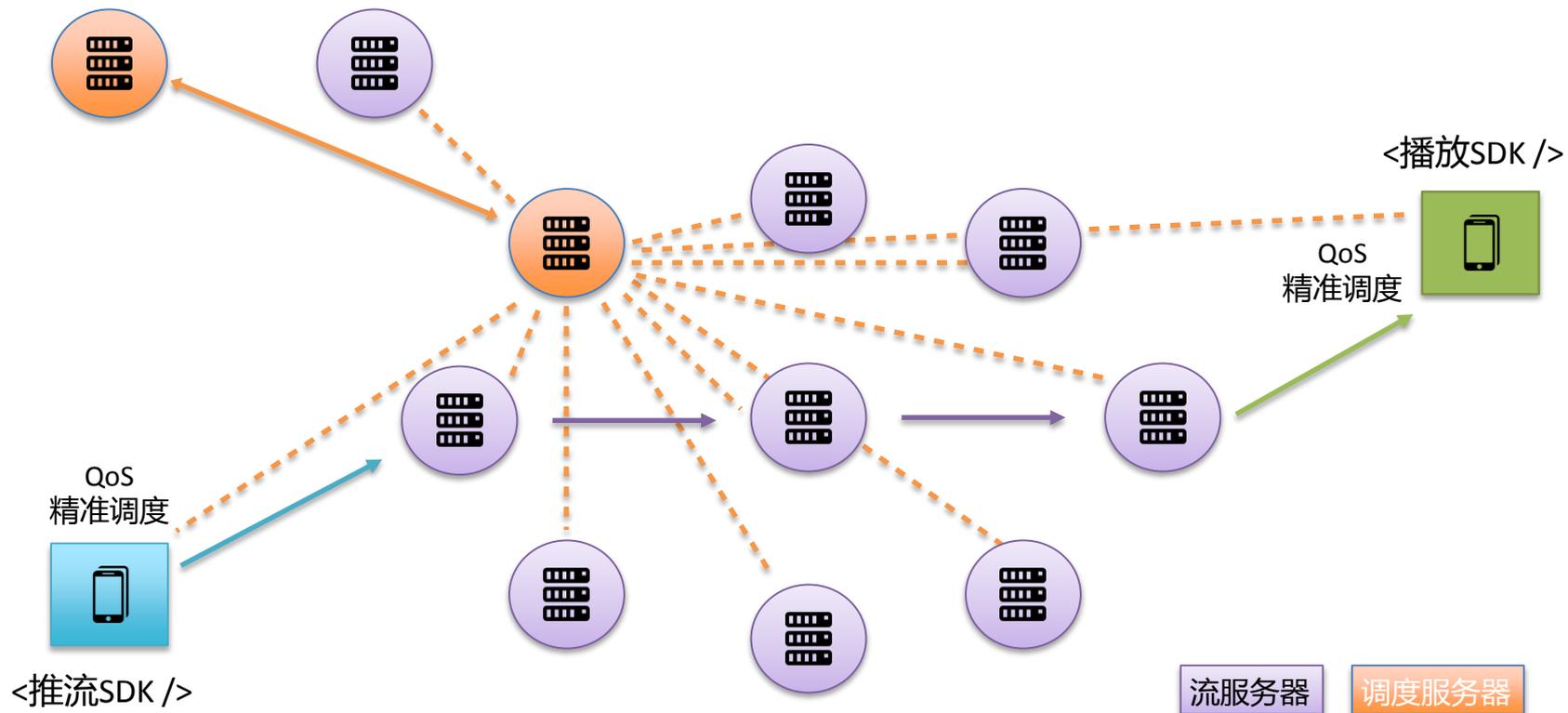
故障容错，线路动态切换，保障直播流畅稳定



基于数据实时侦测，优化线路，保障直播流畅稳定



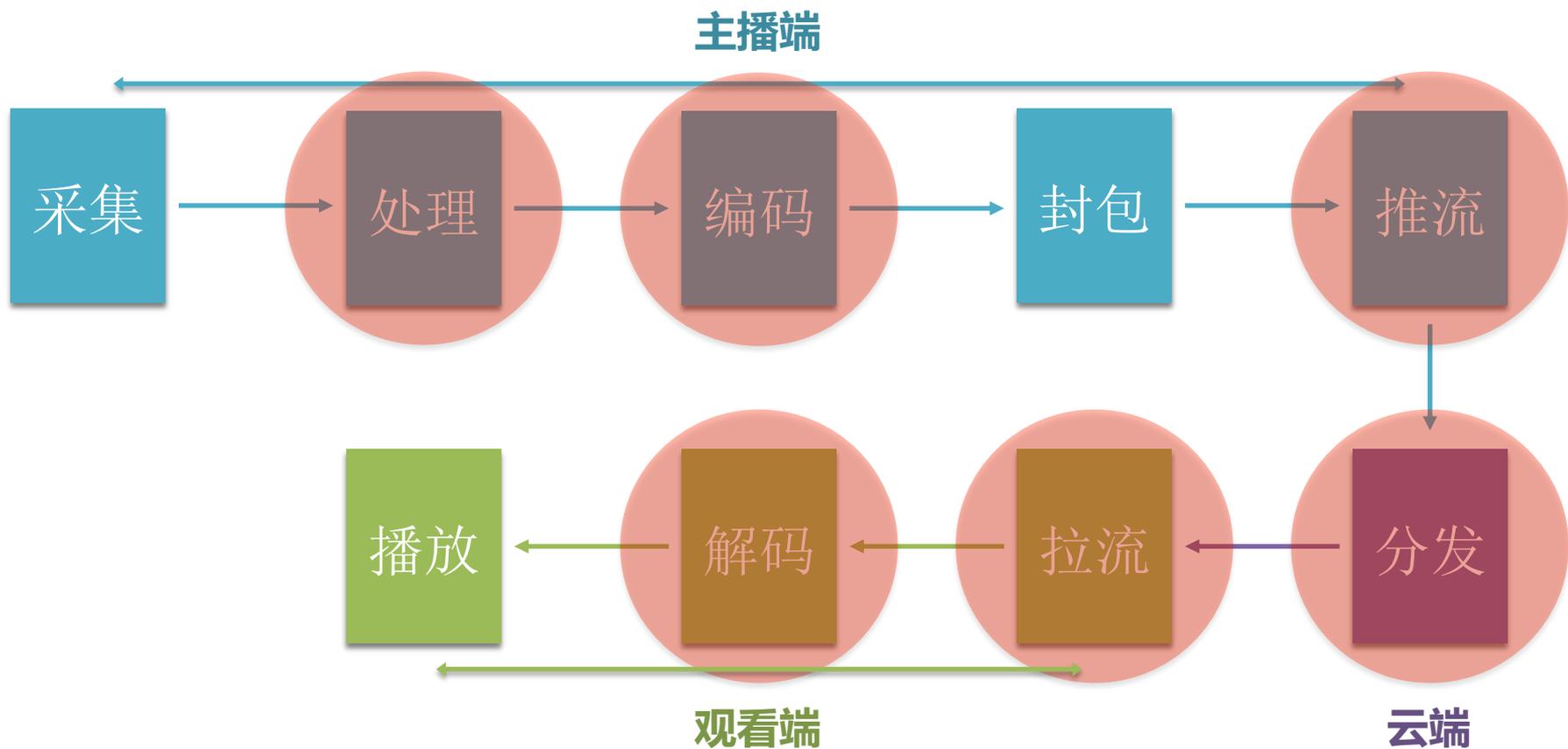
对称部署，无缝切换，服务容错，扩点灵活



挑战三：如何实现

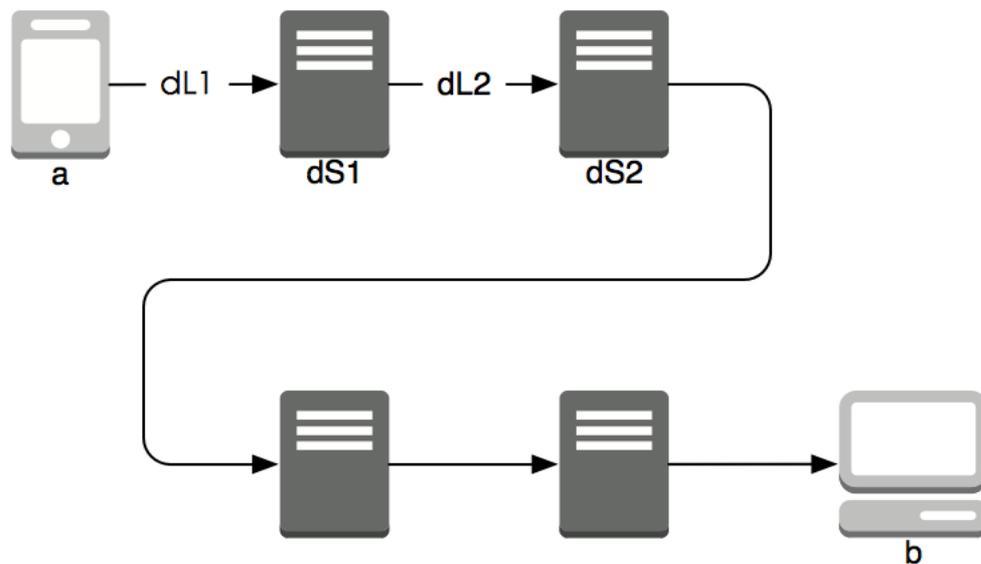


影响直播延时的环节



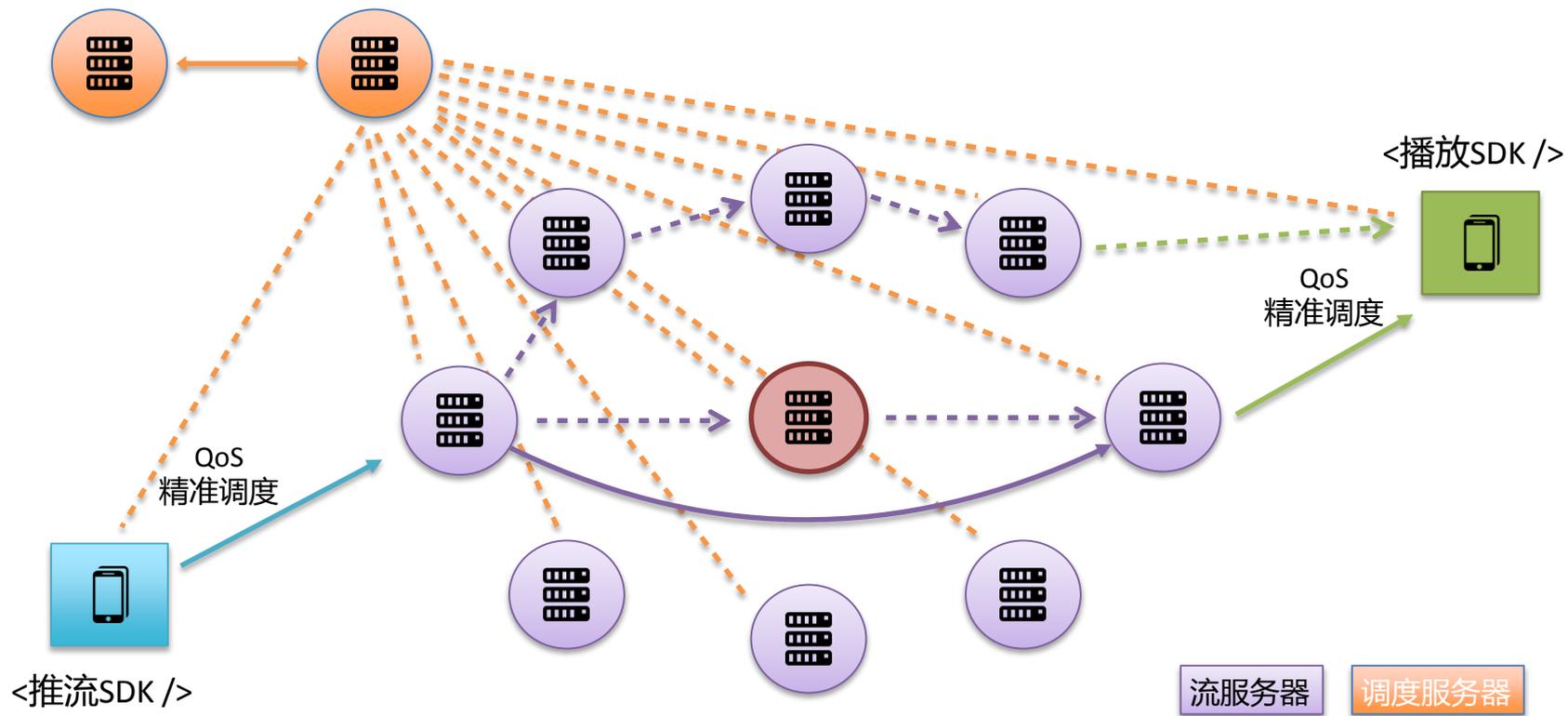
问题一：传输延时

- 发送和接收的时间差
- 转发环节越多，延时累加越大



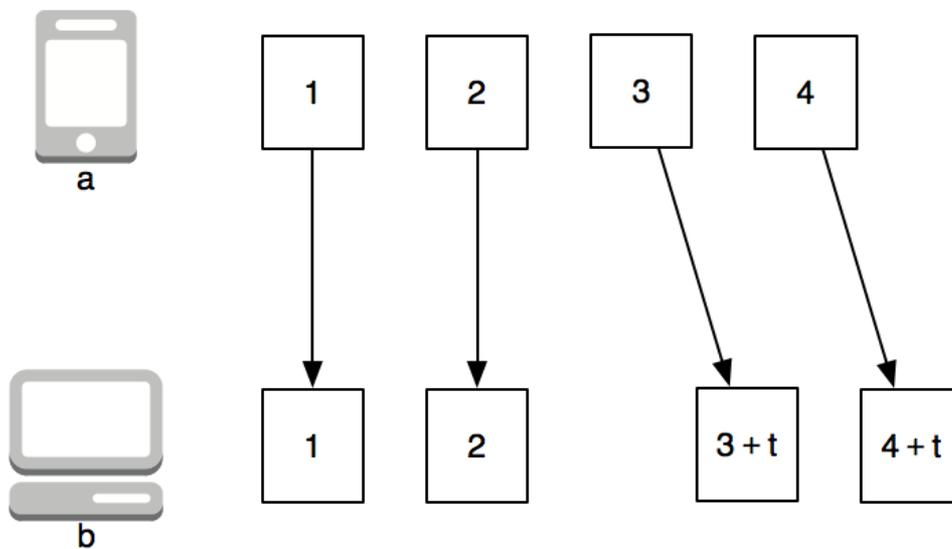
$$\text{延迟 } b-a = \text{Sum}(dL1+dL2+\dots) + \text{Sum}(dS1+dS2+\dots)$$

基于数据实时侦测，调度线路，保障最优路径

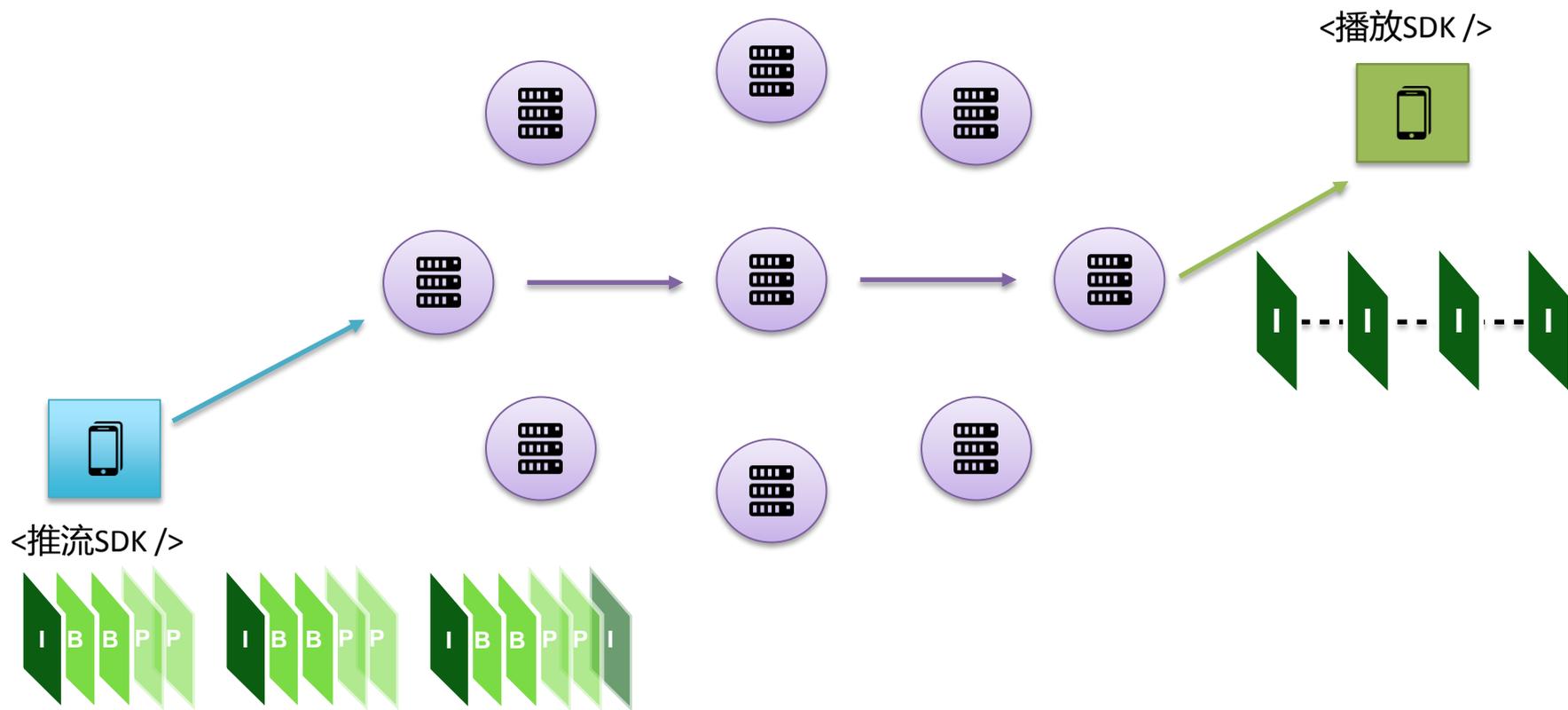


问题二：丢包重传累计延时

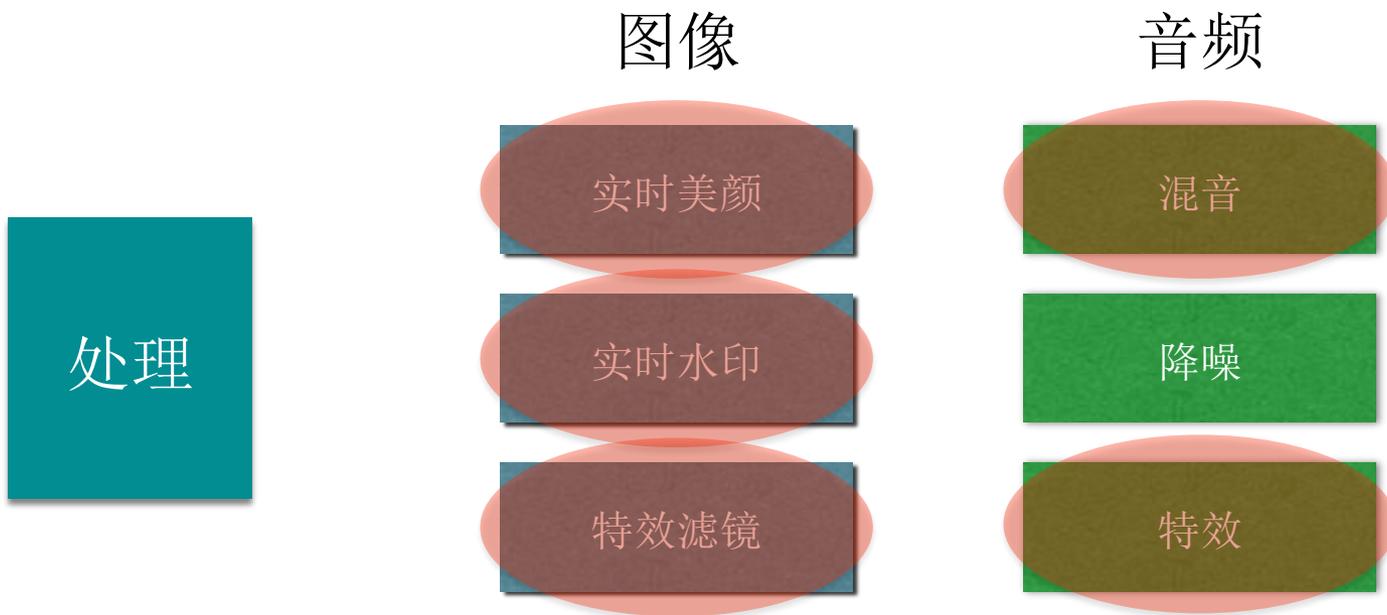
- 网络抖动
- 丢包重传
- 增大延时



播放SDK动态追帧，消除累积延时



问题三：推流端 采集音画数据后的 处理、编码 延时

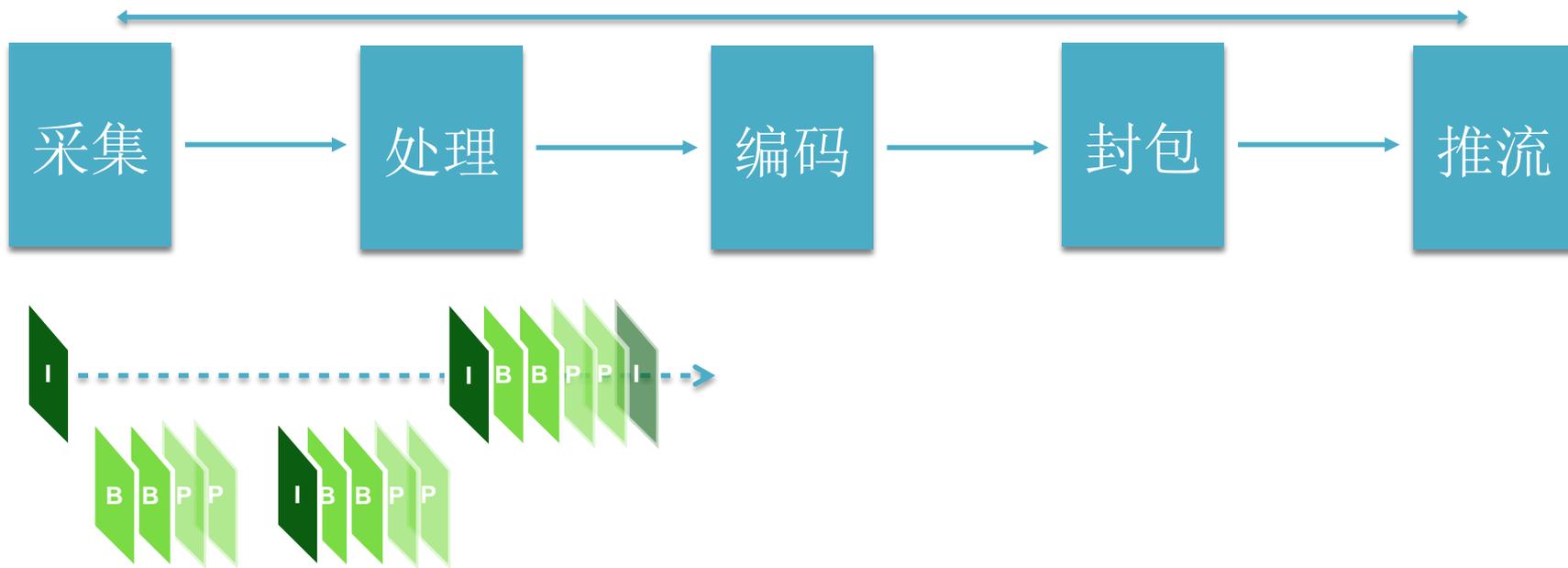


问题三：推流端 采集音画数据后的 处理、编码 延时



编码前丢帧，减少编码耗时、带宽开销

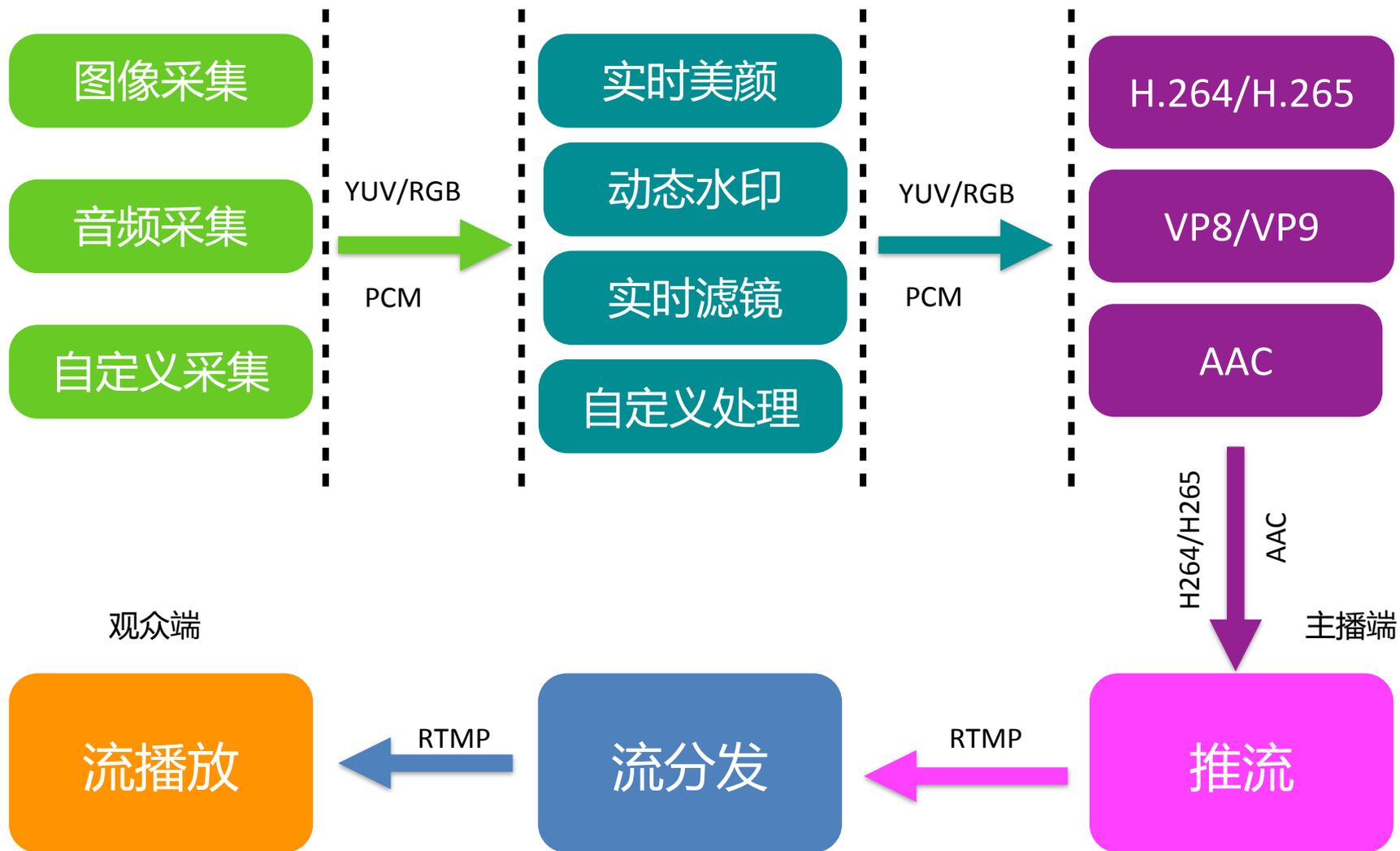
主播端



移动直播优化经验总结

- 支持硬编和软编，提供软编解决安卓机型推流兼容性和稳定性问题
- 软编还能解决码率精确控制、画质清晰度等问题
- 视频编码与图像处理分离（美颜、水印、滤镜等图像处理走GPU），减少处理耗时
- 播放优先硬解（性能优先），其次软解（兼容性适配）
- 支持HTTPDNS防DNS劫持，避免黑屏卡住等问题
- 根据QoS数据实时反馈进行动态精准IP调度，缩短RTT便于秒开、保障直播流畅度
- 支持弱网丢帧策略，保障直播流畅度
- 支持弱网动态码率调节，保障直播流畅度
- 支持编码前丢帧策略，不损失画质的前提下减少编码耗时、保障直播低延时、流畅度
- 支持动态追帧，消除由于网络抖动导致的累积延时
- 支持后台播放暂停视频解码，降低功耗节省流量

七牛直播SDK工作原理（采集、处理、编码、推流、分发、播放）



开放式架构：采集层



开放式架构：处理层



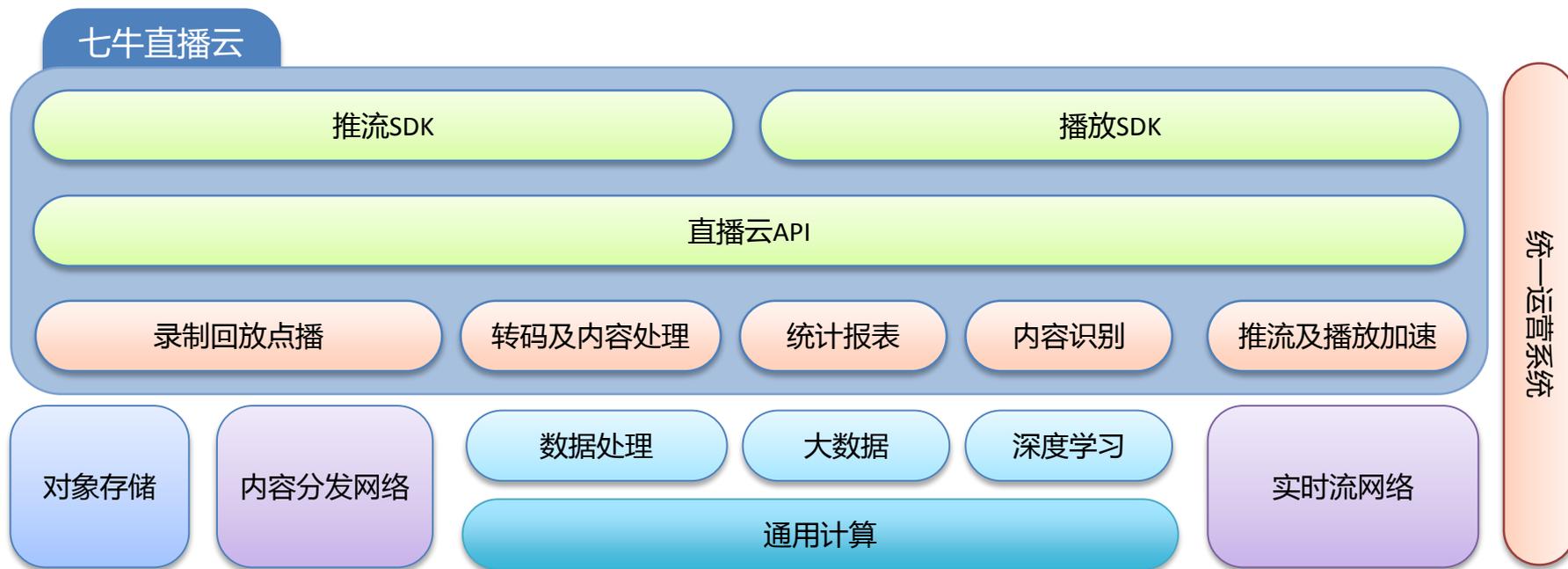
开放式架构：编码层



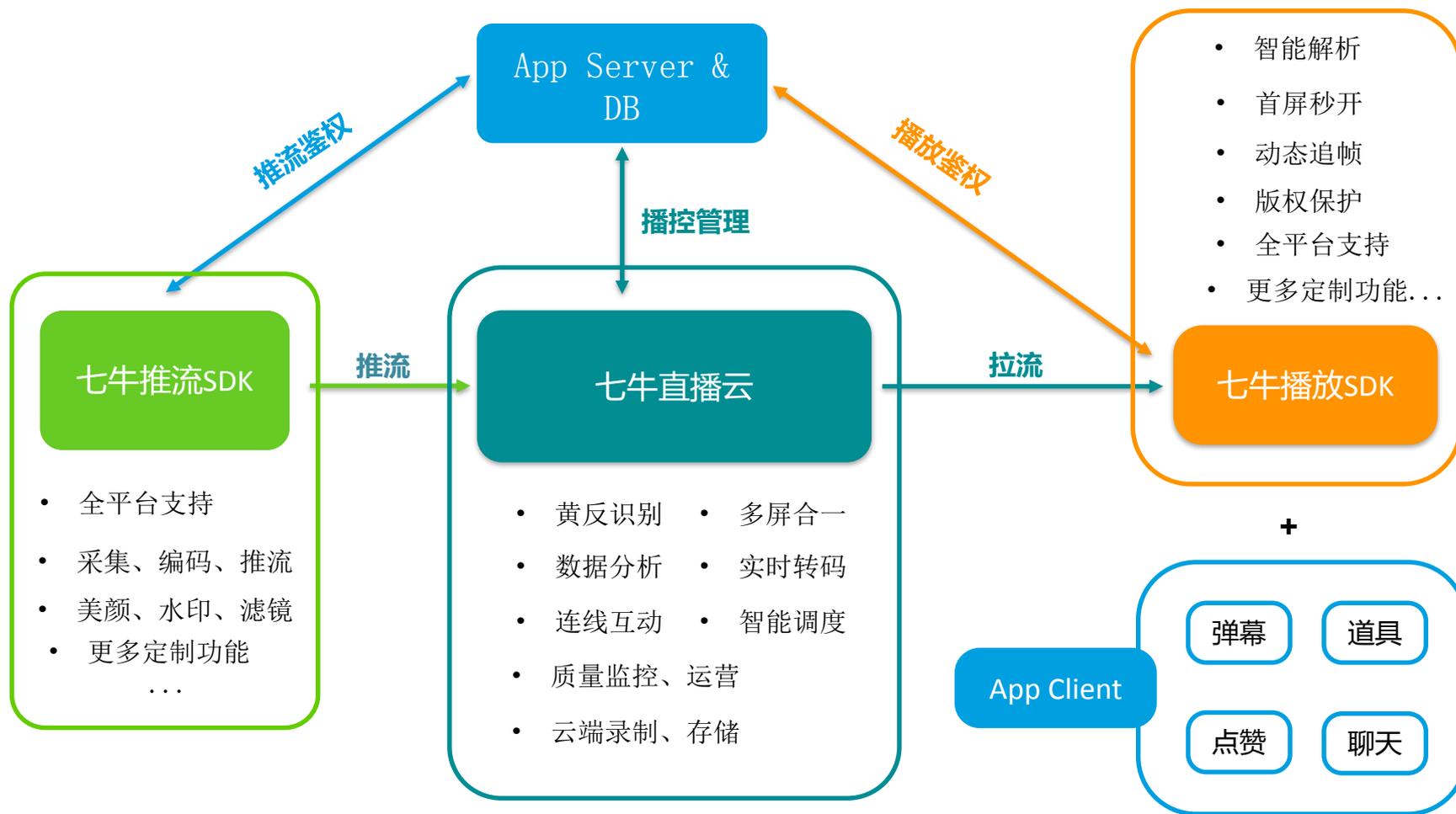
开放式架构：最小内核



七牛直播云架构



移动直播APP速成指南



Thanks!

