TLC 2017 EXCHANGE INNOVATION 交流·创新

主题：低延迟直播P2P

关俊辉

腾讯视频云技术总监

- 关俊辉　　中国科学院　七年音视频经验
- 腾讯云P2P平台总监，负责P2P技术研发
- 2014.11~2017.4　　苏州月光石网络科技　CEO
- 2013.9~2014.9　　风云直播　　架构师
- 2010.3~2013.8　　闪动科技　　Dopool手机电视

个人介绍

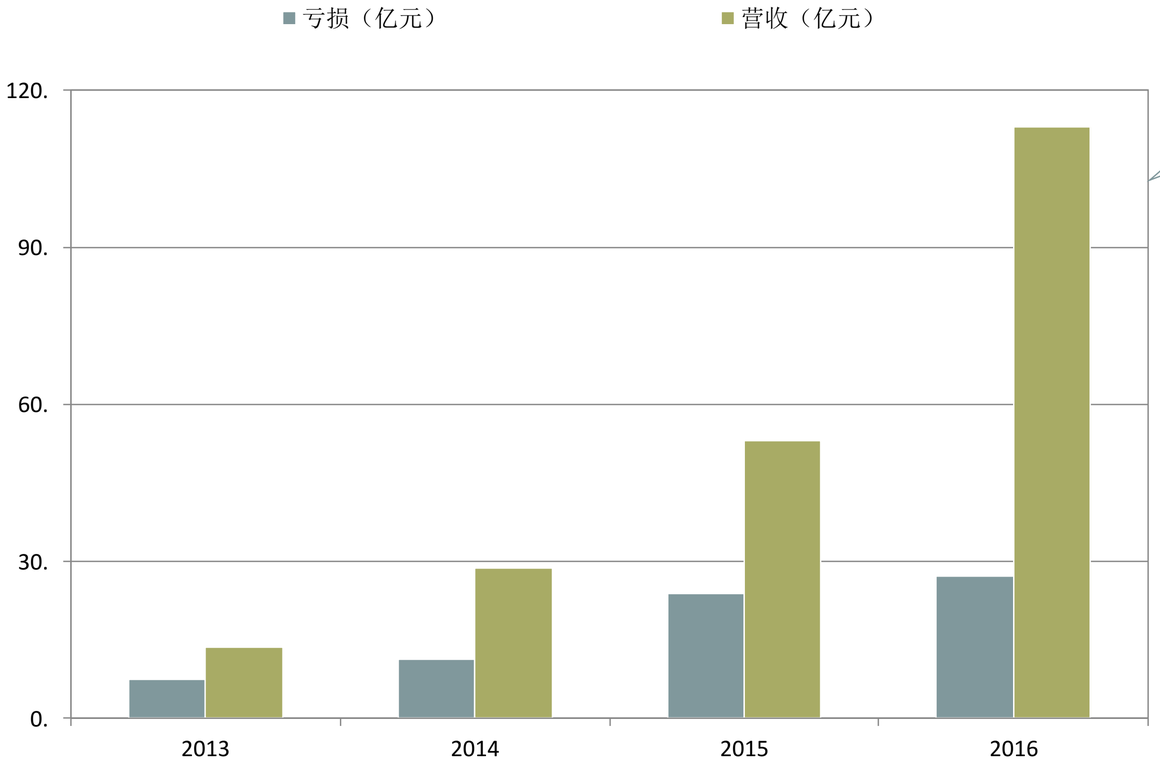
内容大纲

- P2P是成熟运营的必备
- P2P节约带宽的数学基础
- 混合P2P架构效果优于CDN
- 实际效果

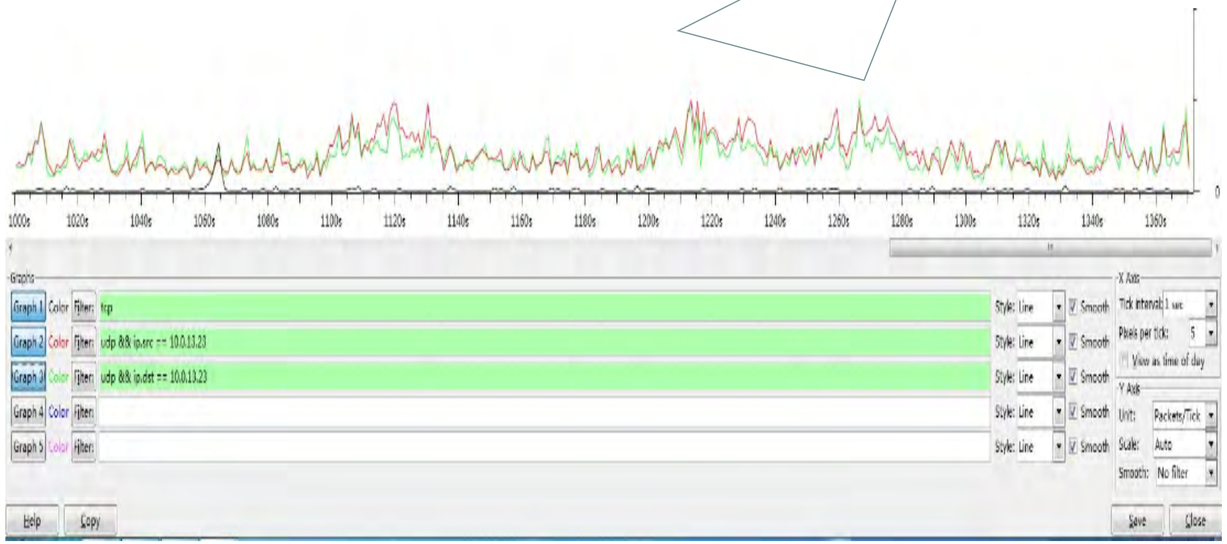
腾讯云

第一部分

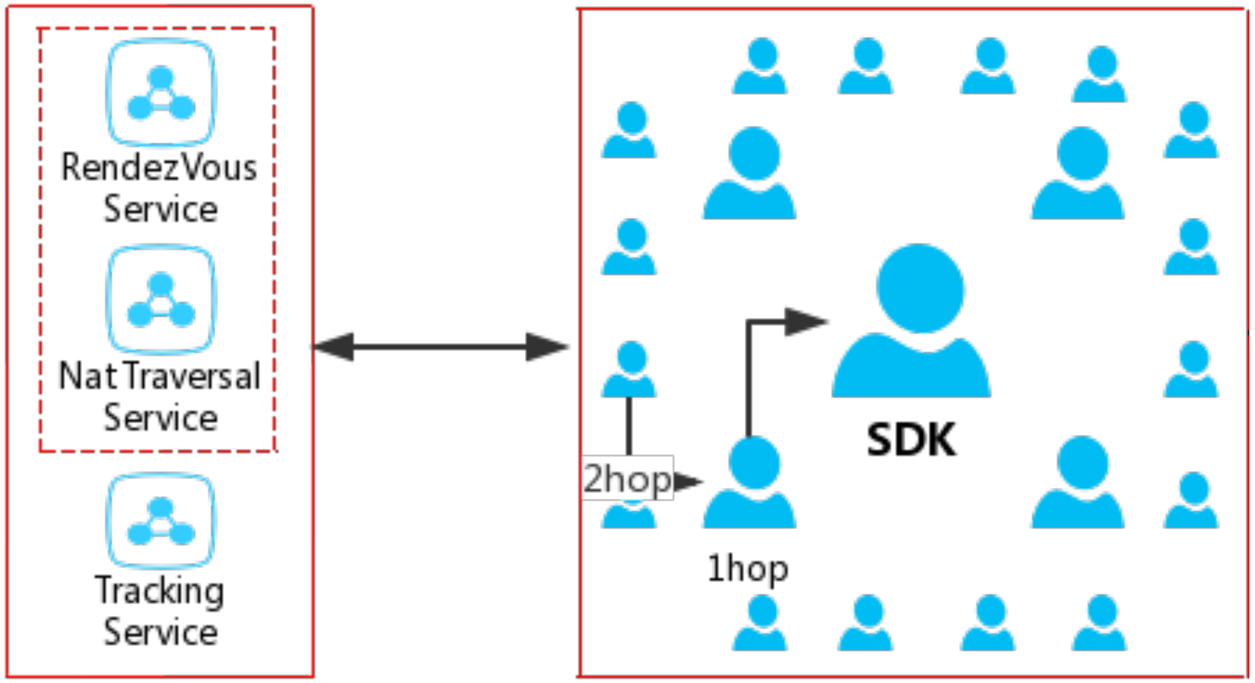
简要介绍



2015-2016年，营收增长113.1%，带宽成本增加58.7%，P2P是控制成本助力工具。



简要介绍



Node：具备P2P能力的一个线上客户端；

RendezVous Service：Node注册、归类、查询服务；

NatTraversal Service：提供两个Node UDP直连服务；

Tracking Service：统计反馈P2P线上服务状况。

P2P算法统计基础

X%：单个Node下载某段数据的概率，1-X%未下载的概率；

NUM1：当前Node的一跳连接Node数目；

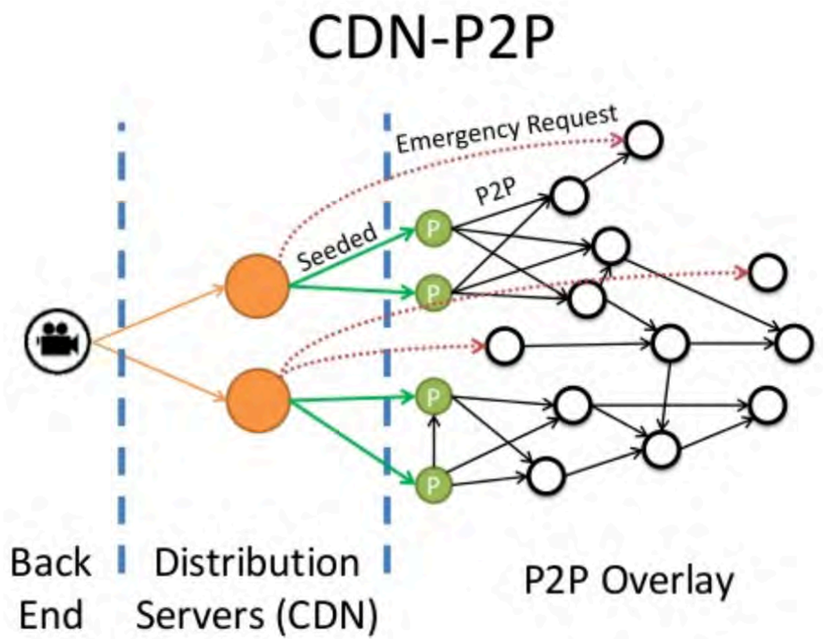
NUM2：当前Node的两跳连接Node数目，等于NUM1*NUM1；

(1-X%) ^ (NUM1+NUM2)：当前Node连接内未下载概率；

X=10，NUM1=8则未下载概率为0.005。

简要介绍

TCP	UDP
Reliable	Unreliable
Connection-oriented	Connectionless
Segment retransmission and flow control through windowing	No windowing or retransmission
Segment sequencing	No sequencing
Acknowledge segments	No acknowledgement



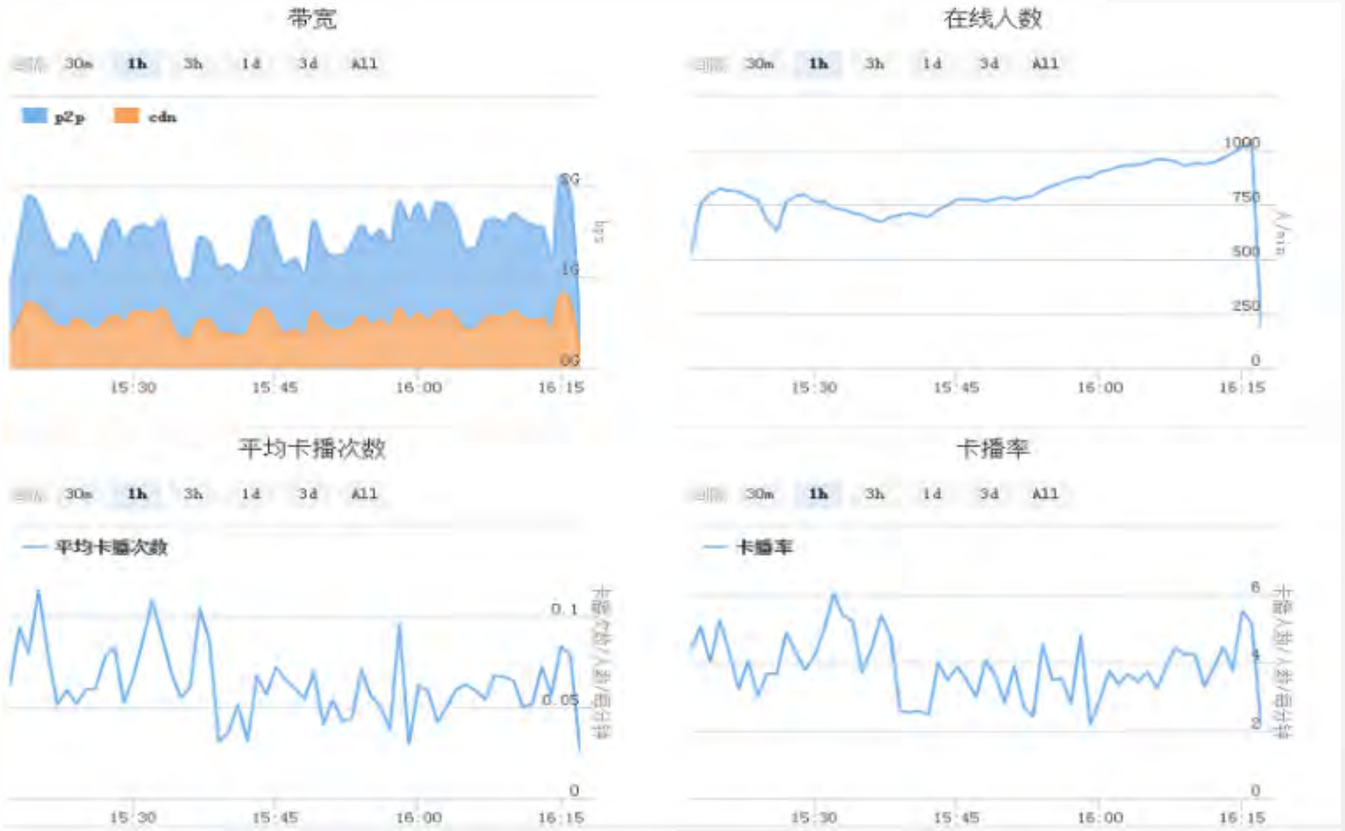
简要介绍

- 显著降低卡播率

纯CDN方案	5~10%
P2P方案	2~5%

- 低延迟

纯CDN方案	3~10s
传统直播P2P	30~90s
低延迟P2P方案	低至6s

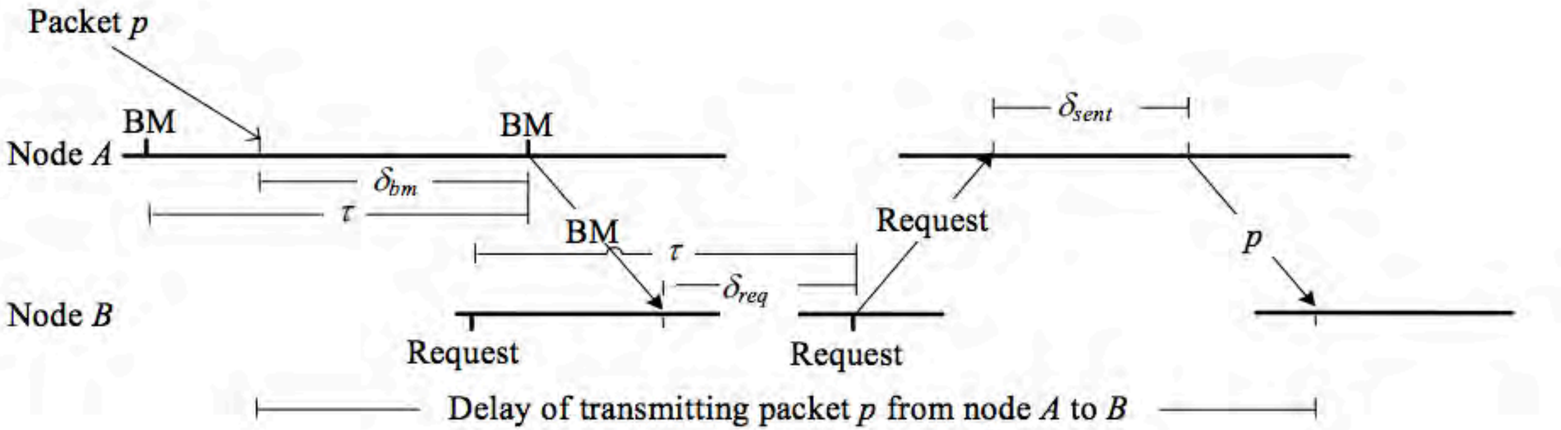


- 传统P2P模型分析
- 如何降低P2P延迟
- 测试数据

腾讯云

第二部分

延迟分析



τ：两次Request或者Bitmap发送的时间间隔;

δ_{bm}：数据接收到，需要一定时间检测到Bitmap的更新，平均时长τ/2;

δ_{req}：接收到远程Bitmap，需要一定时间来发送Request，平均时长 τ/2;

δ_{send}：Socket按照包序列发送完毕需等待时间，平均时长τ/2;

$\overline{\delta_{EED}}$：单次发送数据包到达的平均时间间隔;

平均传输一跳的数据时间为：$\delta_{bm}+\delta_{req}+\delta_{send}+3\overline{\delta_{EED}}=3\frac{\tau}{2}+3\overline{\delta_{EED}}$。

延迟分析

所有的优化都针对此公式：

$$\delta_{\mathrm{bm}} + \delta_{\mathrm{req}} + \delta_{\mathrm{send}} + 3\overline{\delta_{\mathrm{EED}}} = 3\frac{\tau}{2} + 3\overline{\delta_{\mathrm{EED}}} \text{。}$$

一，bitmap 是否是必须的，如何交换信息

- ✓在传输data packet的数据包中加入bitmap；
- ✓采用hash算法对peer进行分组，连接分组后的Peer；

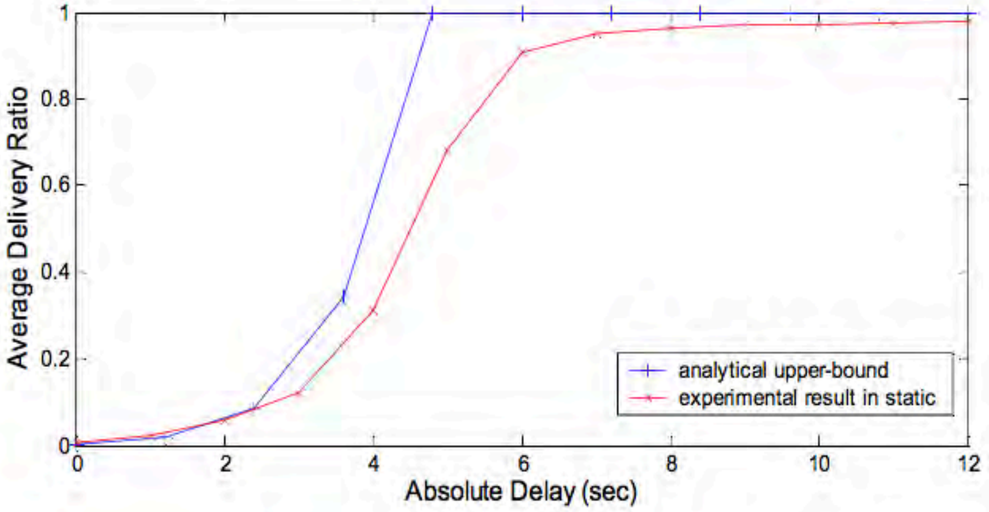
二，一次data packet需要等待三次数据发送，如何缩减

- ✓获取完整数据的一刻即开始p2p分发，减少需求确认；
- ✓去除p2p系统对bitmap的依赖后数据传输需要其他规则定义；

三，可靠UDP传输和不可靠UDP传输对P2P质量的影响

- ✓低延迟情况下，UDP传输时间有限，需提高发送的到达率；
- ✓数据的分块对于传输的效率有较大影响。

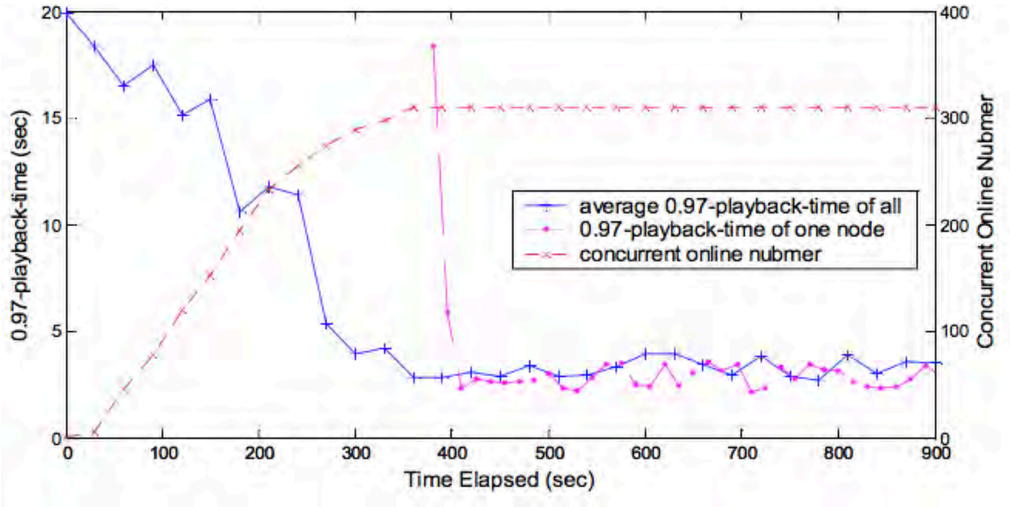
延迟分析



(1)绝对延迟(absolute-delay)——从源节点发出数据包到节点处理数据的延迟.

(2) 数据传递率(delivery-ratio)——截至某个延迟时间,节点收到的数据包的比例.

(3) α延迟(α-playback-delay)——当数据传递率达到一定值α (0≤α≤1)时的绝对延迟.



- 为什么要采用segment格式
- 腾讯云直播P2P架构
- 架构优缺点一
- 架构优缺点二

腾讯云

第三部分

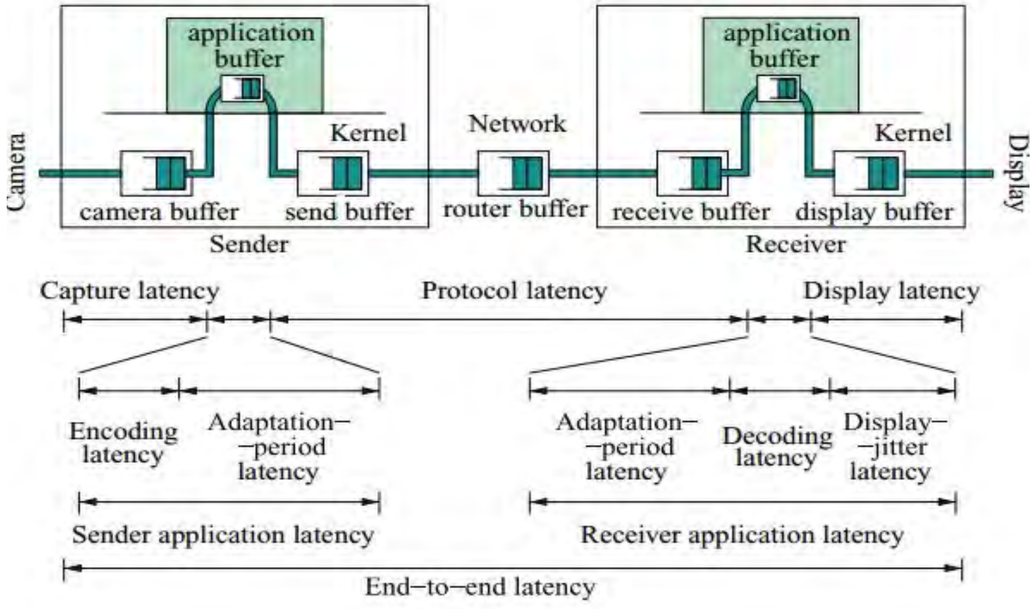
架构简介

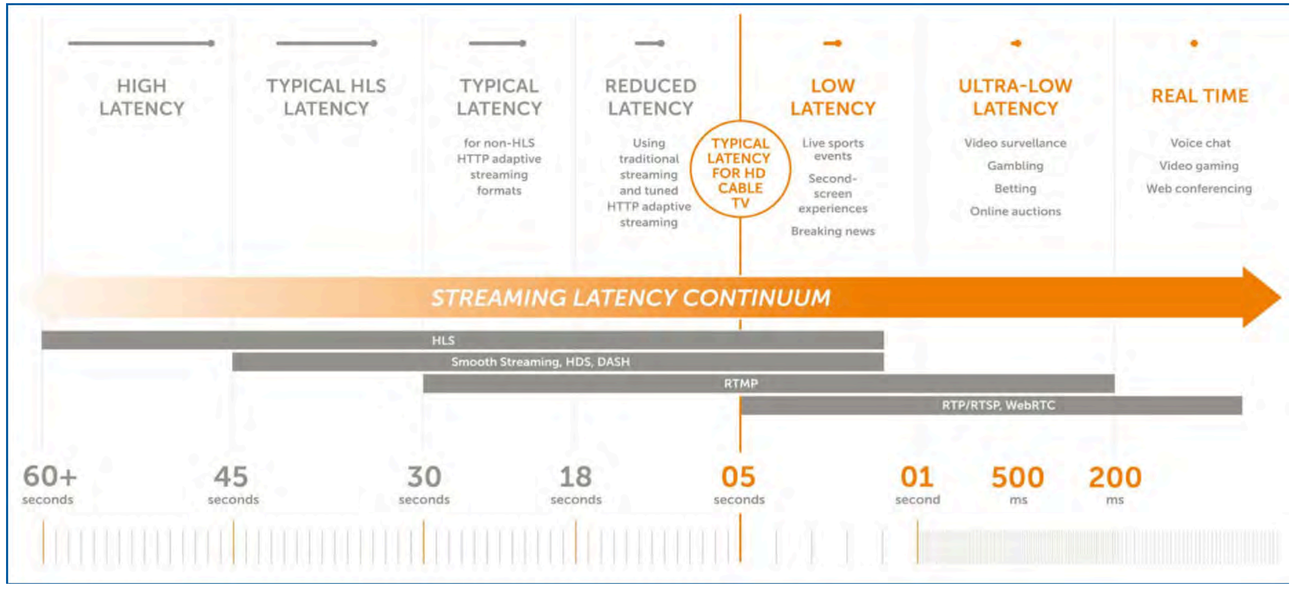
◆系统上游带来的延迟不可控，network耗时与协议相关；

◆根据Akamai的研究，Segment格式可以满足互动直播延迟要求；

◆P2P要求数据的强一致性，在CDN分发之前进行切片保证了这一点；

◆视频短连接代替了长连接，同时增加了请求数据，对回源请求合并要求高。





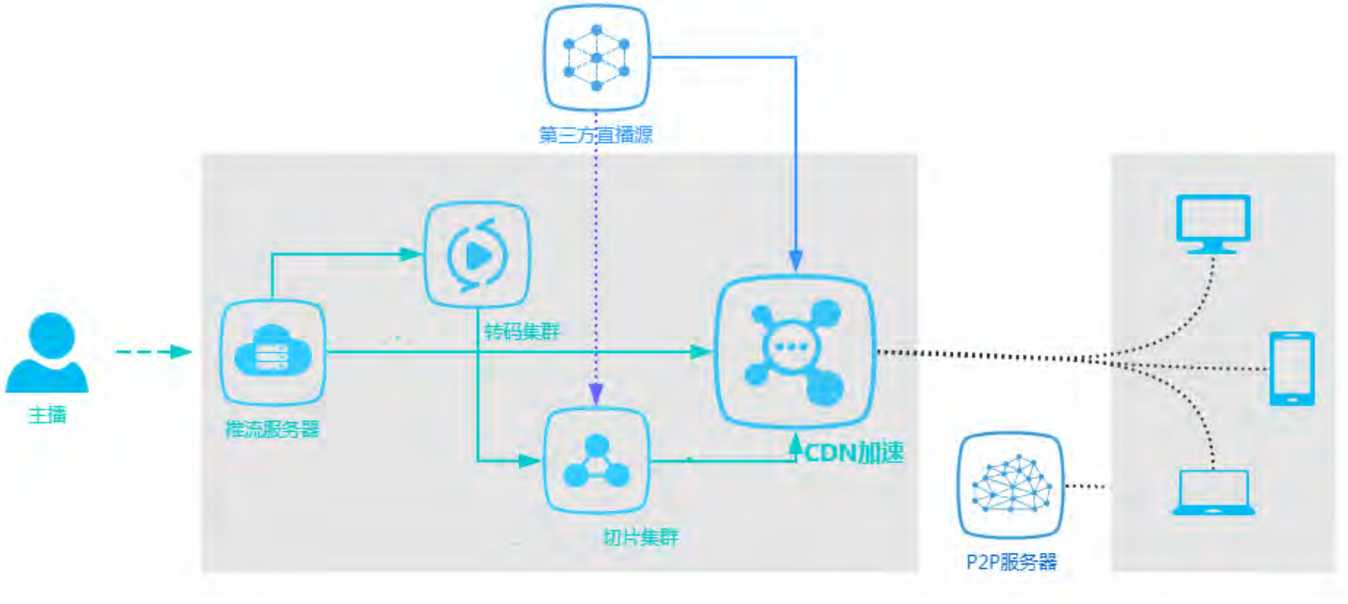
架构简介

◆推流和分发均与现有CDN兼容且解耦和，应用既有资源解决问题；

◆技术上可支持多家CDN，具备简洁有效的调度轮询机制；

◆具备强大的防盗能力，解决了精细运营中出现的防盗防刷等严重问题；

◆传输采用TCP Friendly的速率控制协议，不使用4G流量，不影响其他链路。

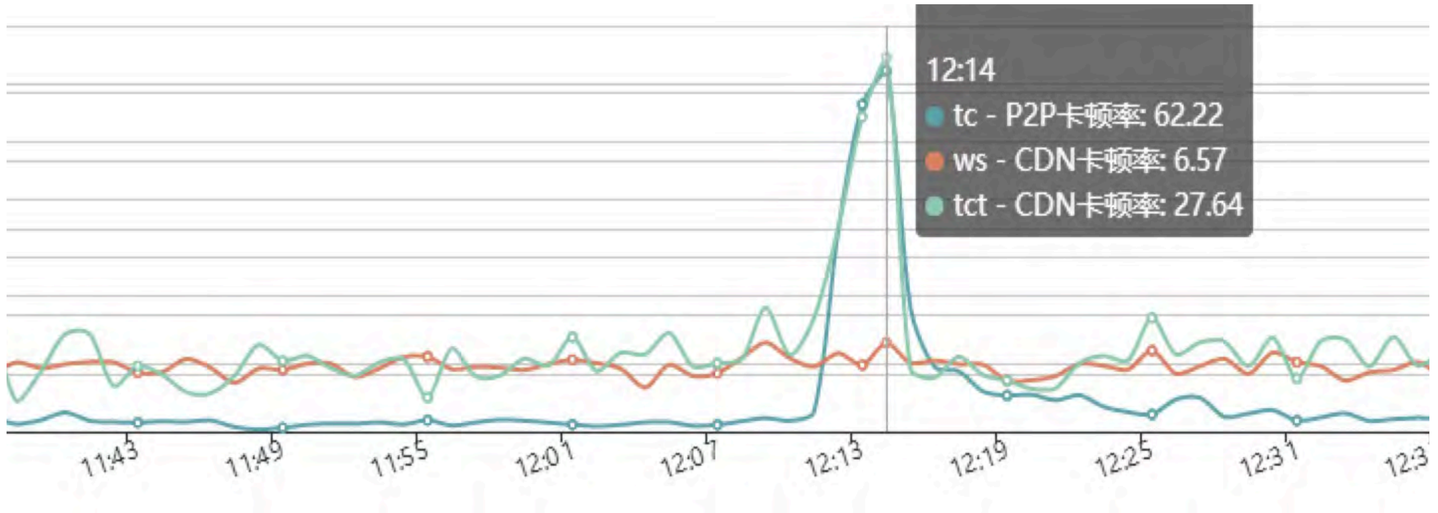


架构简介

◆相较于RTMP协议，系统至少4s延迟，但可以满足低延迟需求；

◆卡播整体表现很好，但是对Upload要求高，会造成整体不稳定性；

◆相对于传统流式传输，请求量增大较多，存在不稳定性风险。

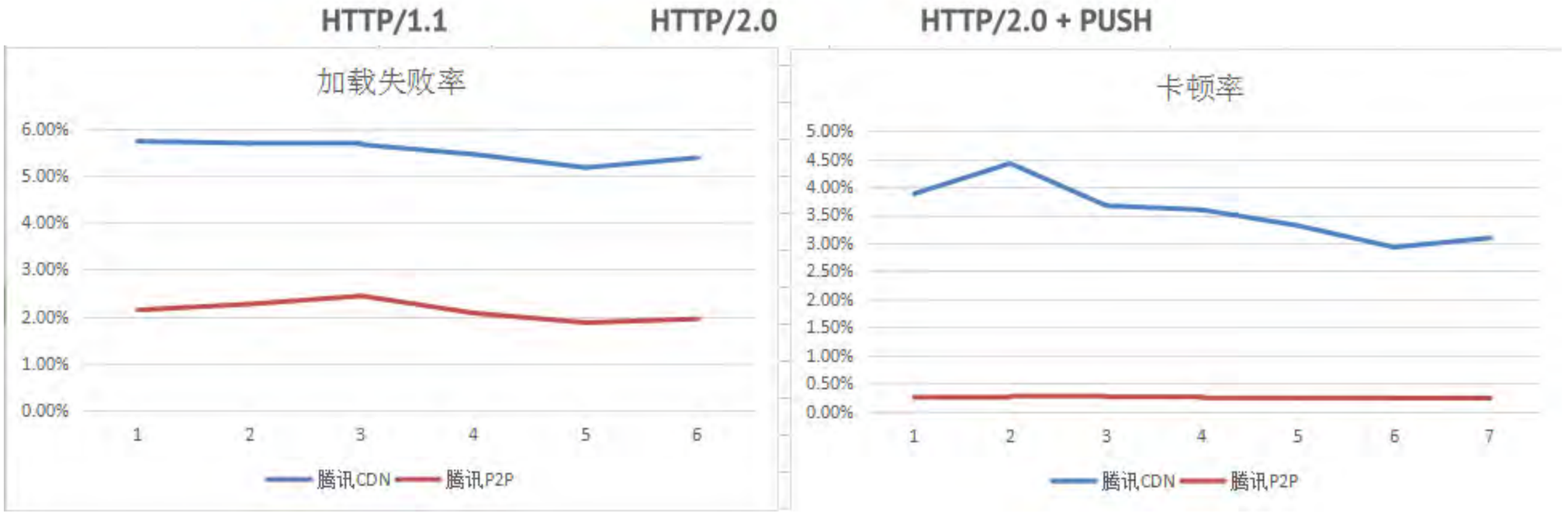


架构简介

◆扩展性非常好，不依赖直播协议服务器，无需特殊部署，保证了系统整体稳定性，排查问题更加清晰；

◆P2P模型很大程度上限制了HTTP的请求量，抵消一部分系统性风险；

◆DC->OC使用HTTP推送以及HTTP2的使用都会降低系统性风险，符合下一代CDN发展趋势。

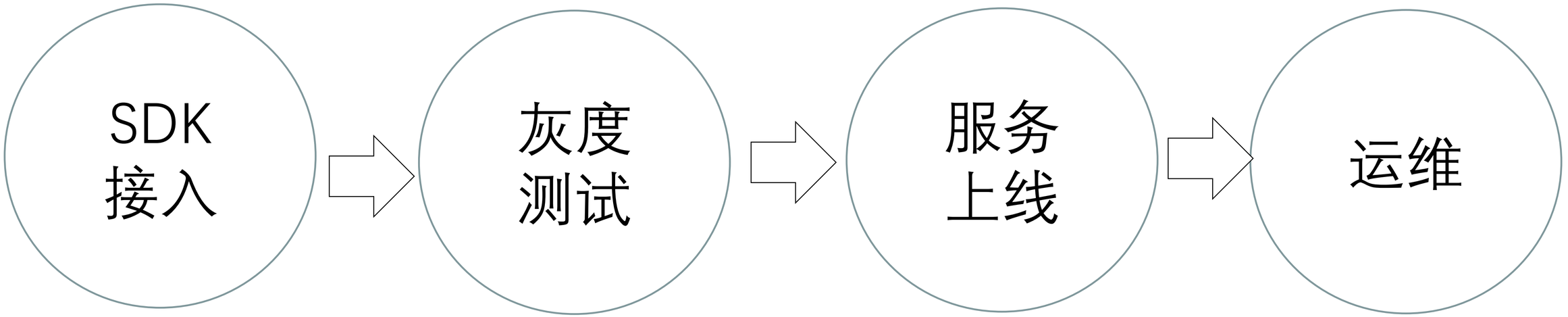


- 对接流程
- Flash平台对接
- 移动端对接
- 移动端SDK参数

腾讯云

第四部分

对接流程



对接流程

```
ns = new NetStream(nc);
ns.addEventListener(NetStatusEvent.NET_STATUS, onPlayStatus);
video.attachNetStream(ns);
ns.play("http://flvtx.plu.cn/onlive/543a354ed2a.flv?txSecret=78453e5ded3f9032&txTime=58c60ff4");
```

```
var loader:Loader = new Loader();
loader.contentLoaderInfo.addEventListener(Event.COMPLETE, function(e:Event):void {
        ns = e.currentTarget.content.stream;
        ns.addEventListener(NetStatusEvent.NET_STATUS, onPlayStatus);
        video.attachNetStream(ns);
        ns.play("http://flvtx.plu.cn/onlive/543a354ed2a.flv?txSecret=78453e5ded3f9032&txTime=58c60ff4");
});
loader.load(new URLRequest("http://qvb.qloud.com/sdk/superp2p.swf")) ;
```

对接流程

- 移动端对接--Android可以提供gradle和aar两种嵌入方式，示例仅介绍gradle

gradle编译（推荐）

Android SDK托管于第三方android lib平台jcenter上，如果您的项目是一个使用gradle编译的AndroidStudio项目，只需添加如下依赖，并经gradle同步之后，即可使用该SDK的各种接口:

```
1.  android.defaultConfig.ndk {
2.      // 设置支持的abi，不设置就是默认全部支持
3.      abiFilters 'armeabi-v7a' //,'x86','armeabi','x86_64','arm64-v8a'
4.  }
5.  dependencies {
6.      // 加入下面依赖
7.      compile 'cn.vbyte.p2p:libp2p:1.3.3'
8.      compile 'cn.vbyte.p2p:libp2pimpl:1.3.15'
9.  }
```

- 移动端对接—IOS提供framework供编译嵌入

对接流程

移动端性能分析报告	
内存	开启P2P 40 分钟内会有大约15M 左右内存增长，并保持稳定。该部分增长主要来自视频数据缓存
CPU	开启P2P 会有1~3%左右的CPU 增长
耗电量	开启P2P 播放一个小时会有1~3%的耗电消耗；低端机型3%左右，高端机型1%左右
发热量	开启P2P 播放一个小时会有1~4 度的温度升高；低端机型1.2 度左右，高端机型4 度左右
首帧时间	开启P2P，首帧时间平均500ms
分享率	秀场类：平均码率1200，延时6s，分享率40%~50%
	赛事类：平均码率2500，延时10s，分享率40%
卡顿率	秀场类：平均码率1200，延时6s，卡顿率5%
	赛事类：平均码率2500，延时10s，卡顿率4%

END

感谢您的聆听.

-关俊辉