腾讯云

腾讯云 | 云+社区

云+社区技术沙龙

腾讯云 | 云+社区

腾讯云X-P2P加速方案

Rocalzhang 张鹏

腾讯高级工程师

SPEAKER

Rocalzhang

腾讯XP2P负责人

腾讯云X-P2P是业内领先成熟的P2P产品，其中多个产品线均已成熟，包括不同平台、不同延迟场景下的P2P直播、点播P2P等，现已推广到斗鱼、企鹅电竞、英雄联盟等直播平台使用，经受住了大流量阅兵活动直播、赛事直播的考验。现在腾讯云视频P2P技术，在P2P网络传输、P2P技术对视频的侵入性、P2P网络拓扑组建上有了新的进展，在此分享给大家。

CONTENTS

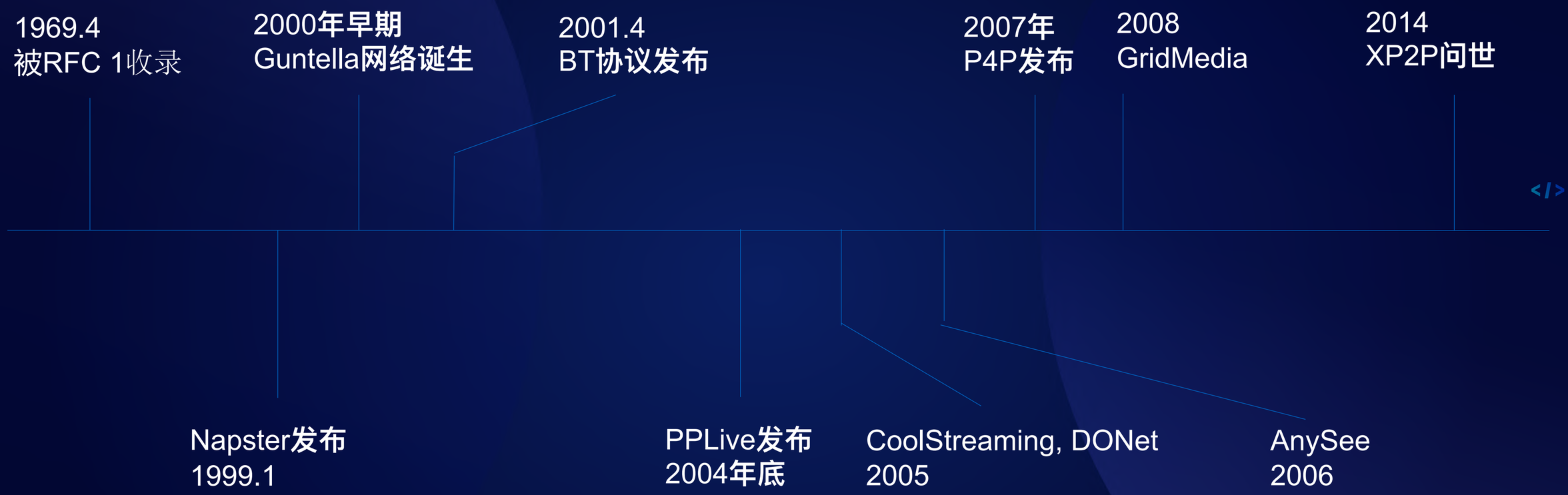
01 XP2P简介
peer-to-peer对等网络

02 XP2P产品功能
为企业减负，为观众增收

03 XP2P的应用场景
直播、点播、大文件分发

04 XP2P的思考与展望
大有可为

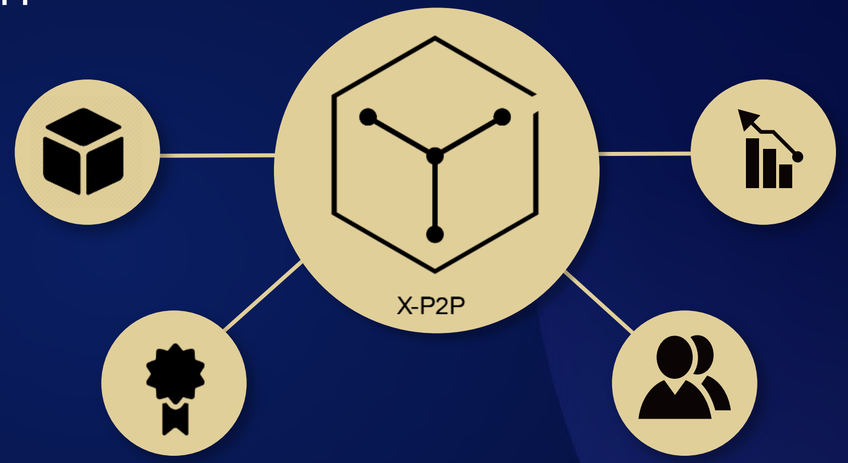
P2P发展历史


1969.4
被RFC 1收录
2000年早期
Guntella网络诞生
2001.4
BT协议发布
2007年
P4P发布
2008
GridMedia
2014
XP2P问世
Napster发布
1999.1
PPLive发布
2004年底
CoolStreaming, DONet
2005
AnySee
2006

更省成本、更流畅：业界领先的低延时P2P方案

S8赛事最高峰值

- Flash
- iOS/ANDROID
- H5
- 短视频
- DASH



- 平均卡顿率最低至2%
- 秒开时间低至400ms
- 平均分享率50%，可节约一半成本

- 直播时延业界最低

- 服务带宽超过8T
- 7家上线客户，包括top3游戏直播
- 涵盖游戏直播、广播电视、体育等

CONTENTS

腾讯云 | 云社区

01 XP2P简介
peer-to-peer对等网络

02 XP2P产品功能
为企业减负，为观众增收

03 XP2P的应用场景
直播、点播、大文件分发

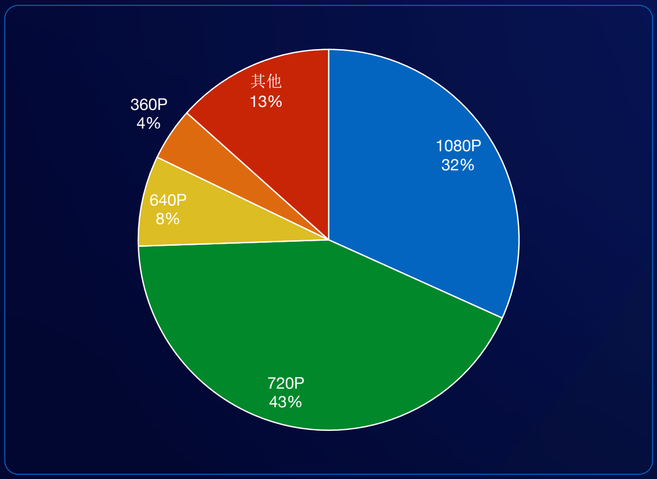
04 XP2P的思考与展望
大有可为

行业需求

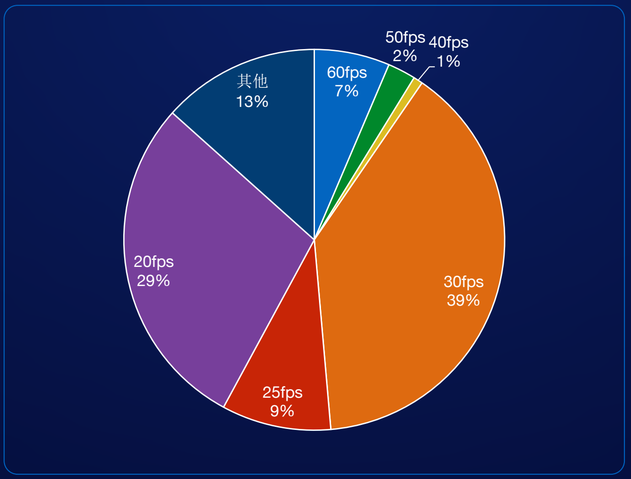
- 带宽成本居高不下
- 带宽需求增速大于带宽成本下降速度
- 用户体验要求较高，纯CDN方案接近优化极限
- 传统P2P方案无法解决播放体验差、延迟长等问题

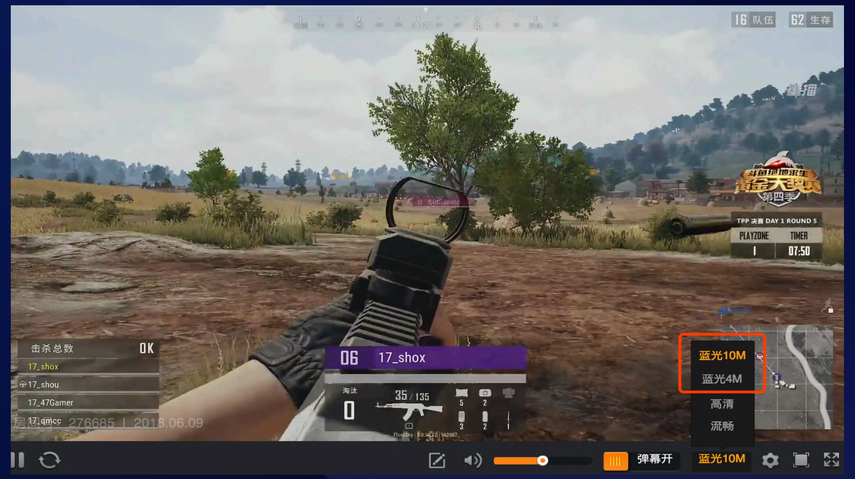
4K HDR ULTRA HD

1080P由3%上升到32%

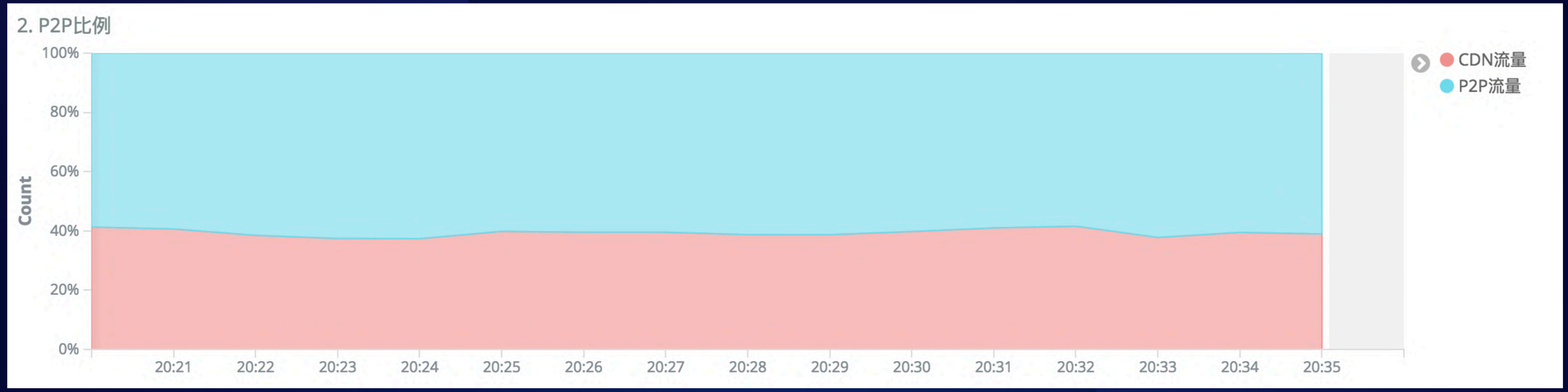


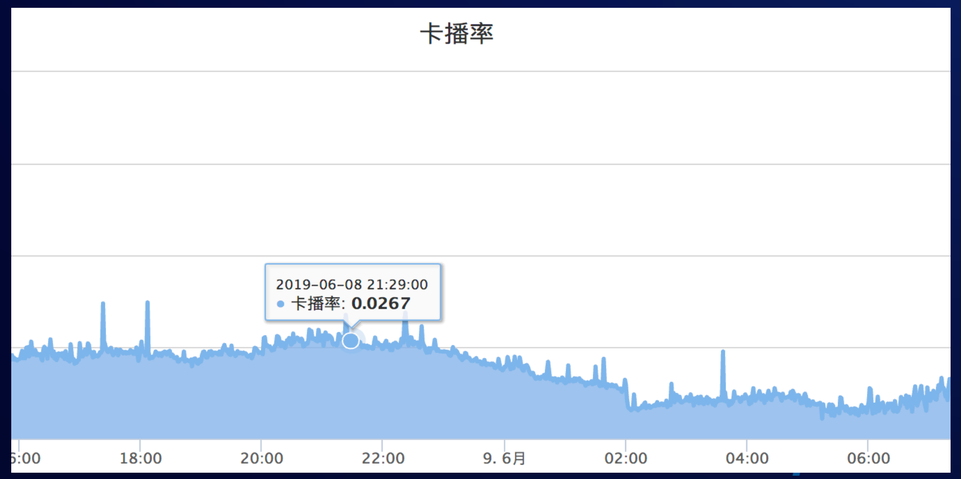
主流帧率上升到30FPS

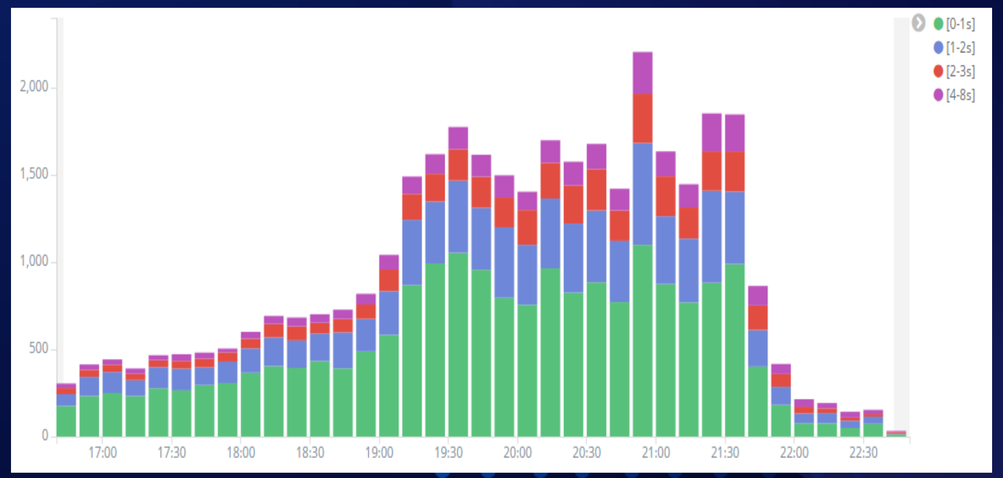




P2P的能力—节省带宽，为企业减负；提高体验，增加收视







P2P的必要条件-切片

Why ?
- 全网对等节点，要有统一的数据块一致性，这是能进行P2P前提需求

点播：
- 可以按照offset制定数据块，比如：1MB一片
- 可以按照播放时长制定，比如：1秒的数据一片

直播：
- 不能够按照offset制定数据块
- 可以按照dts制定，比如：
  $\text{Piece}_i = \{\boldsymbol{dts} \in \left[\boldsymbol{1000i}, \boldsymbol{1000(i+1)}\right), \boldsymbol{dts}\text{为解码时间戳}\}$
- 将数据变为私有协议格式，插入切片边界信息
- 将直播数据预先切好分片文件，以分片格式去分发，如HLS、Dash

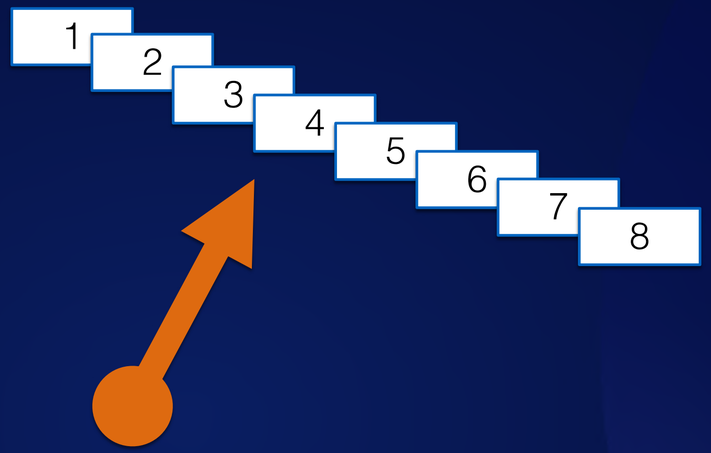
一部电影，长度、时长固定，拆分切片很方便

直播而言，新内容稍纵即逝，全网用户有差别随机位置开始起播

P2P针对直播的切片

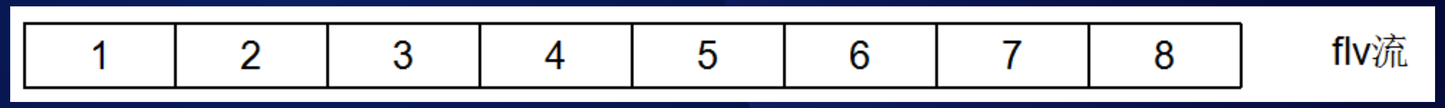
HLS、Dash

- 天生就是切片segment，不需再切片



？问题：

- 带宽不均匀
- 前进需要节奏
- 每次还要客户端去请求



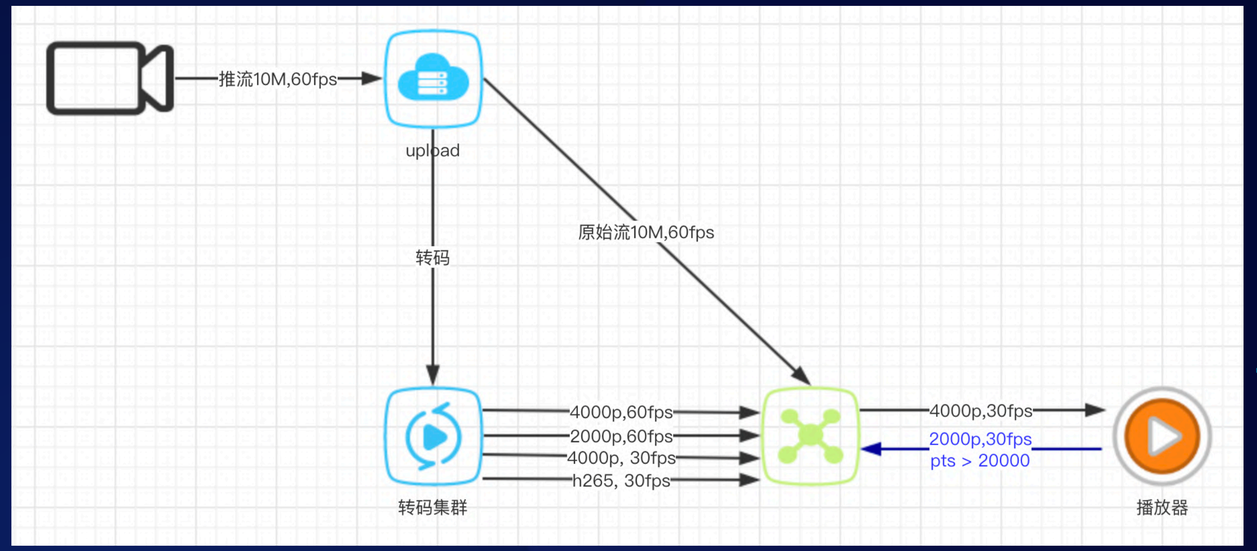
Flv、FMP4

- 突破在原始直播流上无法进行切片的限制
- 对直播流无任何损害，其mux、codec内容都不变，就算不使用p2p，直播流也可无缝送交播放器直接播放

P2P的自适应码率

HLS、Dash

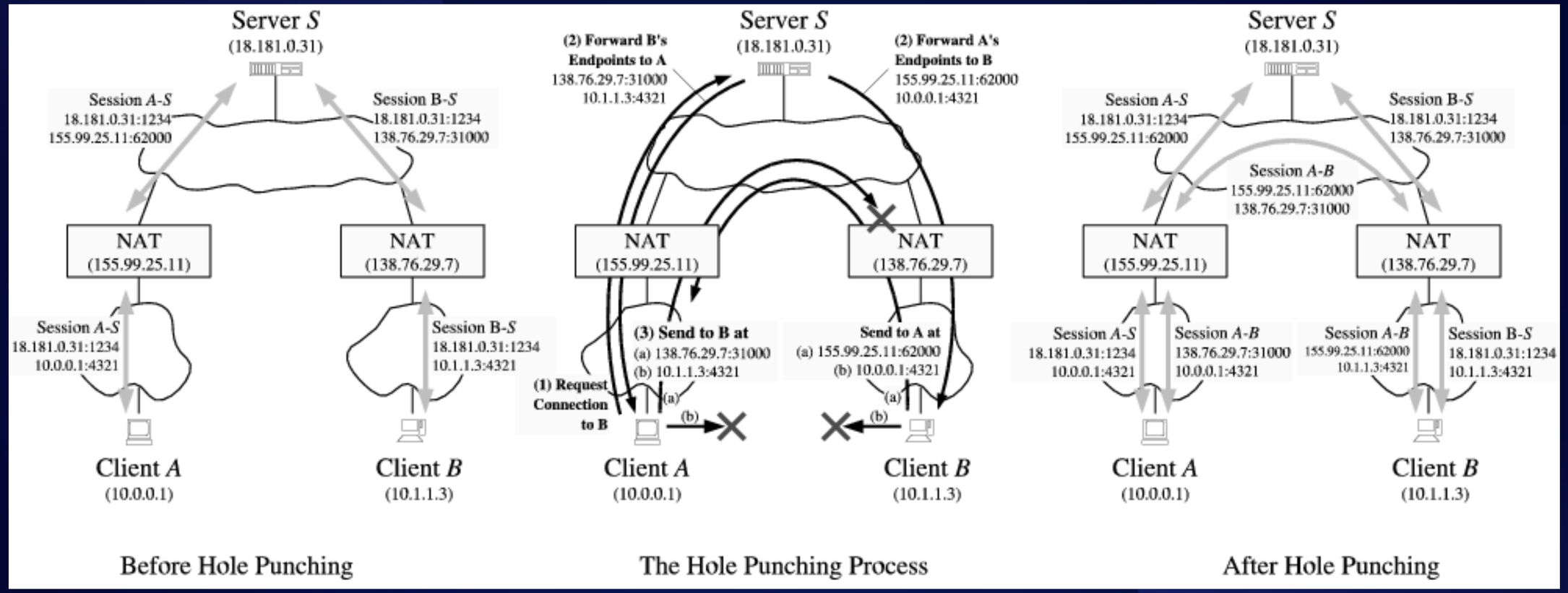
- 为自适应码率而生



Flv、FMP4-怎么办？

- 切换码率，无非就是将解码器等信息重置，再交给播放器
- P2P SDK感知网络带宽，自适应切换码率，且内容连续无缝切换
- 配合腾讯云播放器，可以做到无感知、无缝切换，通过事件告知view层切换到了目标目标
- 而SDK内部还是使用类似长连接单请求来获取数据

P2P的NAT穿透



1a. Client A去连接stun服务器，获取自己的公网地址
1b. Client B去连接stun服务器，获取自己的公网地址
2.Client A通过种子服务器拿到了Client B的地址，然后去请求

3. Client A先向Client B发送一个包
4.Client A向stun服务器发送一个包，请求stun服务器代为转发给Client B
5.Client B收到stun服务器转发的包，响应Client A，于是连接建立

P2P之NAT类型

（1）Open Internet：主机具有公网IP，允许主动发起和被动响应两种方式的UDP通信。

（2）UDP Blocked：位于防火墙之后，并且防火墙阻止了UDP通信。

（3）Symmetric Firewall：主机具有公网IP，但位于防火墙之后，且防火墙阻止了外部主机的主动UDP通信。

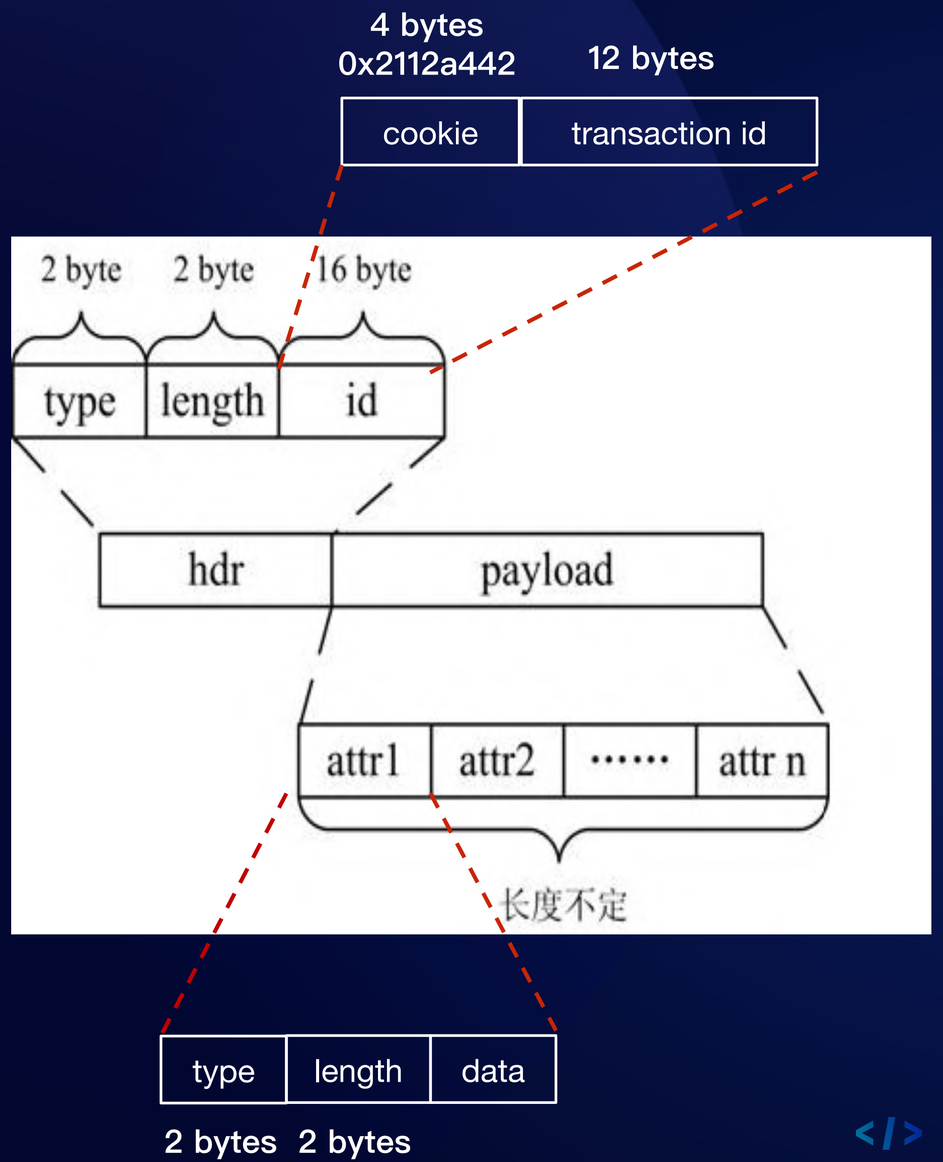
（4）Full Cone NAT：完全锥型，行为类似公网IP，任何外部主机就可以直接向NAT后的公网地址发起UDP通信。

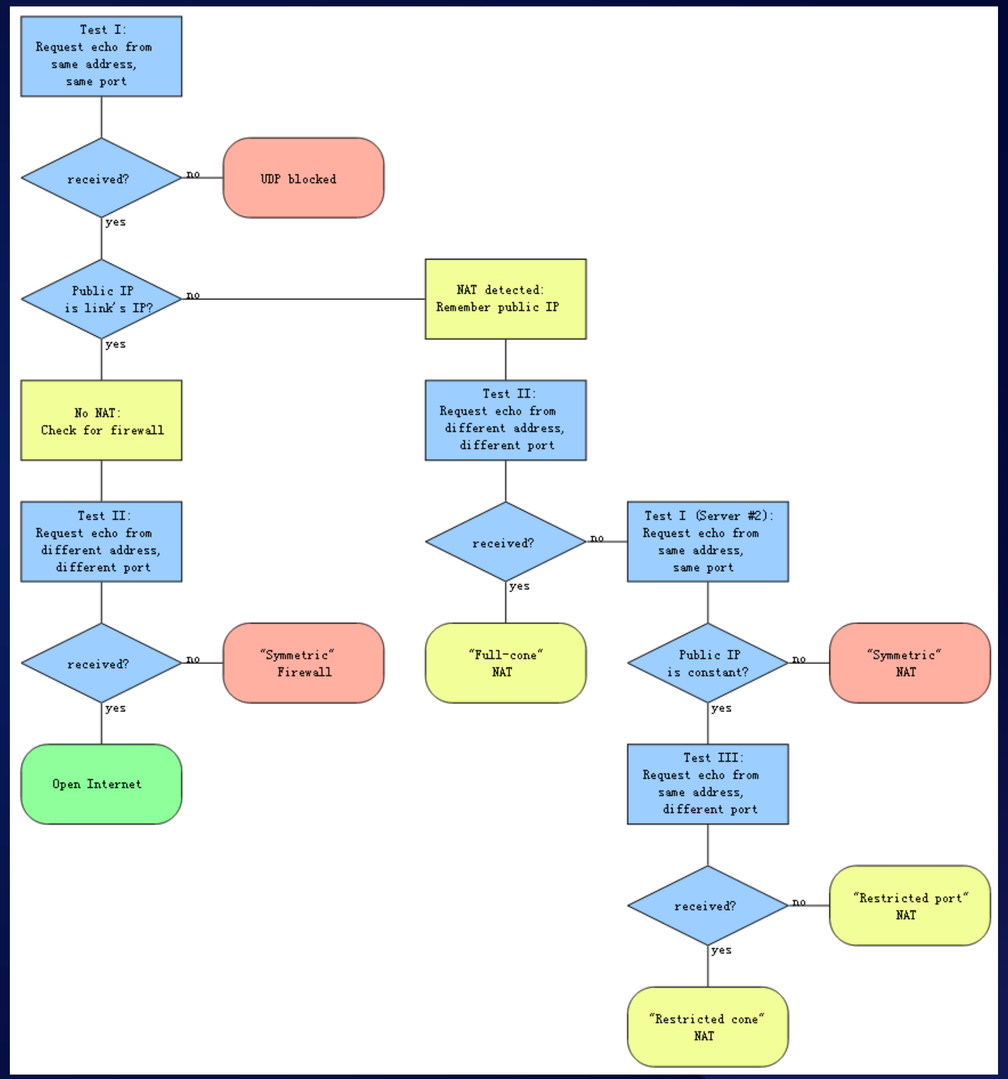
（5）Restricted Cone NAT：受限圆锥型，验证通信时对方的IP。因此，要想外部主机能够主动向该内部主机发起通信，必须先由该内部主机向这个外部发起一次通信。

（6）Port Restricted Cone NAT：端口受限圆锥型，验证通信时对方的IP、PORT。

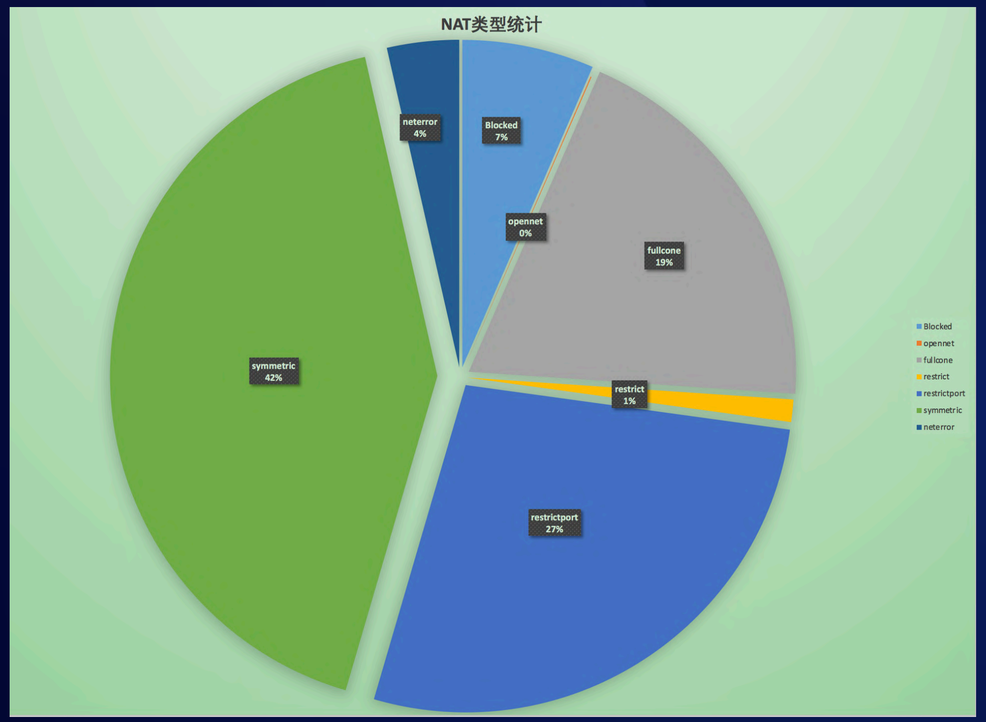
（7）Symmetrict NAT：对称型。它连接外部peer1和peer2时拥有随机不同的ip和端口。难以穿透

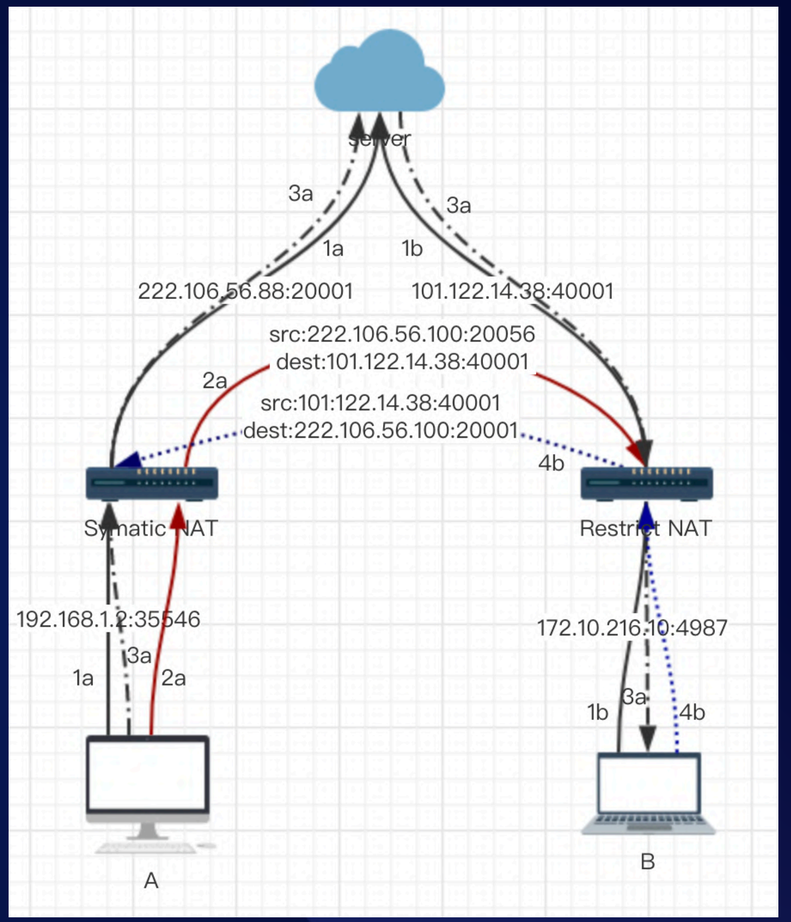
P2P之STUN协议


4 bytes
0x2112a442
12 bytes
cookie
transaction id
2 byte
2 byte
16 byte
type
length
id
hdr
payload
attr1
attr2
……
attr n
长度不定
type
length
data
2 bytes
2 bytes


Test I:
Request echo from
same address,
same port
received?
no
UDP blocked
yes
Public IP
is link's IP?
no
NAT detected:
Remember public IP
yes
No NAT:
Check for firewall
Test II:
Request echo from
different address,
different port
Test II:
Request echo from
different address,
different port
received?
no
Test I (Server #2):
Request echo from
same address,
same port
received?
no
"Symmetric"
Firewall
yes
"Full-cone"
NAT
Public IP
is constant?
no
"Symmetric"
NAT
Open Internet
yes
Test III:
Request echo from
same address,
different port
received?
no
"Restricted port"
NAT
yes
"Restricted cone"
NAT

P2P之NAT分布

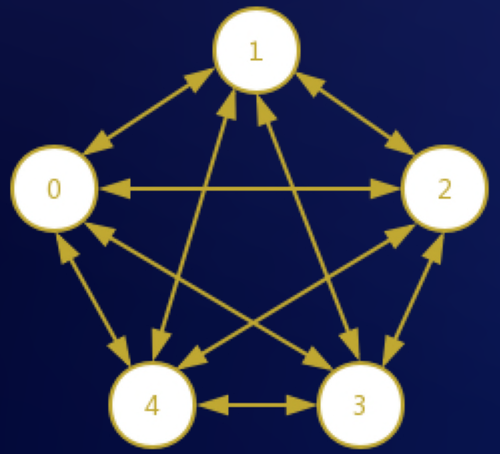




对称型的NAT占比大，且标准STUN对对称型很难穿透！

然而，腾讯云XP2P再一次攻破了这种限制..

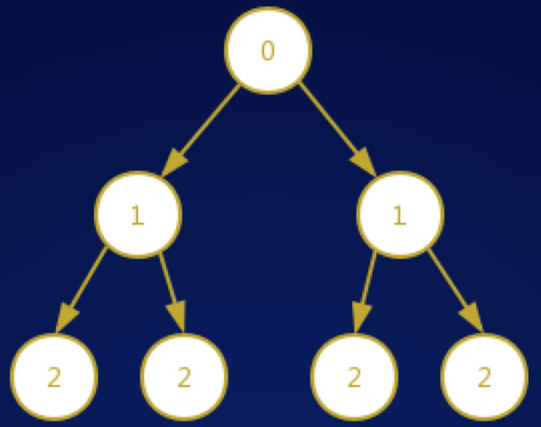
P2P网络拓扑结构



平均分流模型：

把一道流均分成5份，每个节点平均负责一份，并分享给其他4个peer，其他peer亦然

各节点带宽平摊效果好，各节点间播放观看无明显时差

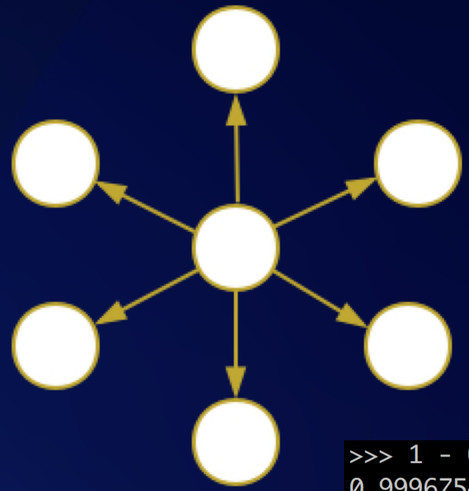


树状模型：

由顶层节点获取源数据，一带二，二带四地层层分享给子节点

随着层数的增加，虽能达到更高的分享率，但缺点也很明显：

- 有一半的底层叶子节点无法贡献数据，也就无法均摊带宽，父节点带宽压力大
- 上层节点离开产生的影响大
- 底层节点与顶层节点有明显时差



网状模型：

每个节点都极少自主请求源数据，按需向周围节点请求，考虑到子节点的子节点数量是级数增加的，所有总能从子节点拿到缺少的数据

- 此模型经验证能达到很高的分享率，但是要求很高的延迟，故十分适用于高延迟直播
- 此模型需要比较高频的相互信息交换

```
>>> 1 - 0.8 ** 36
0.9996754814463416
```

XNTP的传输能力

要先探测上行带宽，好控制最大可用上行？

不该探测：
- 动态的带宽变化，根本无法探测
- 探测浪费带宽

应该：
- 要不停地探测
- 在使用中自然探测

要强占TCP带宽吗？

不应该抢占：
- TCP用的剩下的，才是P2P最多该使用的

应该：
- 要公平竞争

TCP的薄弱能力

启动慢

从初始速度增加到理想速度，即便是倍增也要好几个回合。

拥塞控制差

加性增乘性减，导致带宽不均，且利用率最多只达75%。

抗抖动、抗丢包差

丢包率超20%，基本就废了。

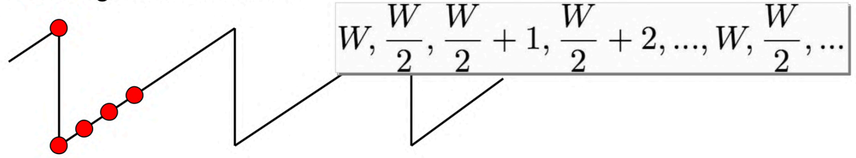
重传歧义

不能确定重传包是真丢了传回来，还是之前的包没丢延迟到的

TCP Modelling: The "Steady State" Model

The model: Packet size B bytes, round-trip time R secs, no queue.

- A packet is dropped each time the window reaches W packets.
- TCP's congestion window:



- The maximum sending rate in packets per roundtrip time: W
- The maximum sending rate in bytes/sec: $W\,B\,/\,R$
- The average sending rate T: $T = (3/4)W\,B\,/\,R$
- The packet drop rate p: $p = \frac{1}{\frac{3}{8}W^2}$
- **The result:** $T = \frac{\sqrt{6}B}{2R\sqrt{p}} = \frac{\sqrt{3/2}B}{R\sqrt{p}}$

5

XNTP的传输能力

腾讯云 | 云+社区

快速启动

迅速检测出理想带宽。

基于公式的速率控制

使用基于数学模型公式的带宽控制，最大化带宽利用。

以丢包率反馈传输速率

即便丢包率30%，也能充分利用那剩下的70%。

双序号包索引

学习quic协议，完美的重传设计

An Improved "Steady State" Model

A pretty good improved model of TCP Reno, including timeouts, from Padhye et al, Sigcomm 1998:

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)}$$

T : sending rate in bytes/second]
R : round trip time
p : fraction of packets lost
t_{RTO} : TCP retransmission timeout

Would be better to have a model of TCP SACK, but the differences aren't critical.

6

XNTP的丢包率计算

if n=8, the values of w_0 to w_7 are:
[1.0, 1.0, 1.0, 1.0, 0.8, 0.6, 0.4, 0.2]

$$\frac{1}{p} = \frac{\sum_{i=0}^{8} Interval_i \cdot Weight_i}{\sum_{i=0}^{8} Weight_i}$$

We calculate the average loss interval I_mean as follows:

```
I_tot0 = 0;
I_tot1 = 0;
W_tot = 0;
for (i = 0 to k–1) {
    I_tot0 = I_tot0 + (I_i * w_i);
    W_tot = W_tot + w_i;
}
for (i = 1 to k) {
    I_tot1 = I_tot1 + (I_i * w_(i–1));
}
I_tot = max(I_tot0, I_tot1);
I_mean = I_tot/W_tot;
```

The loss event rate, p is simply:

```
p = 1 / I_mean;
```

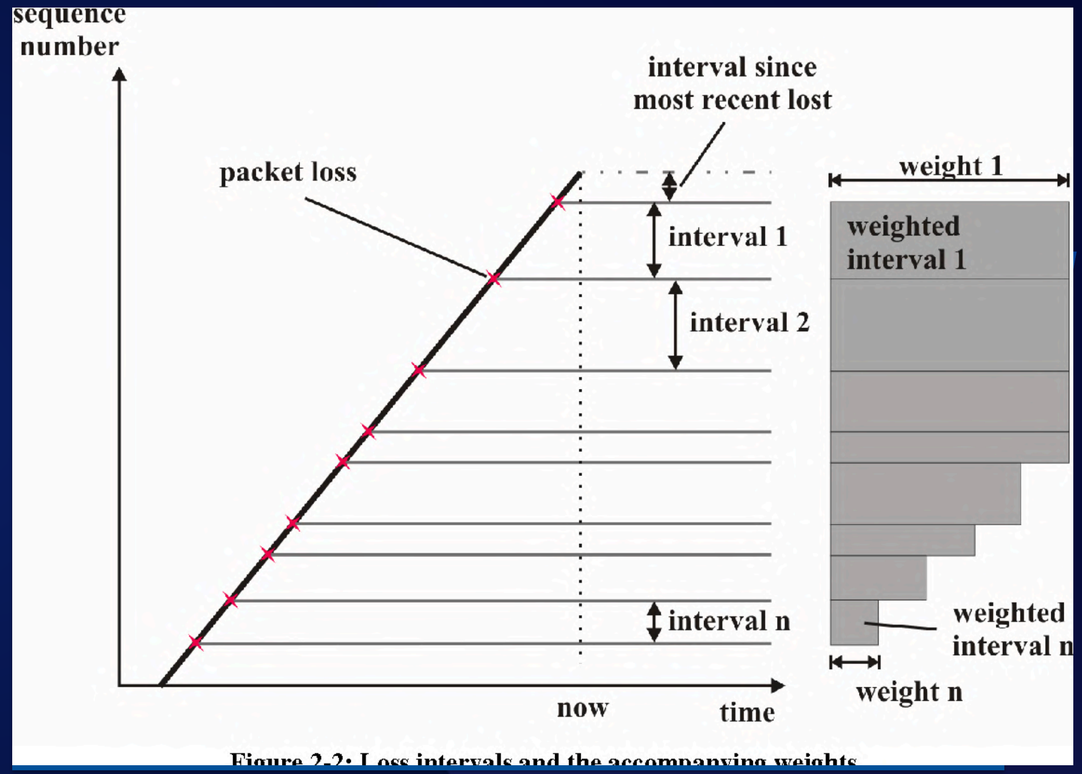
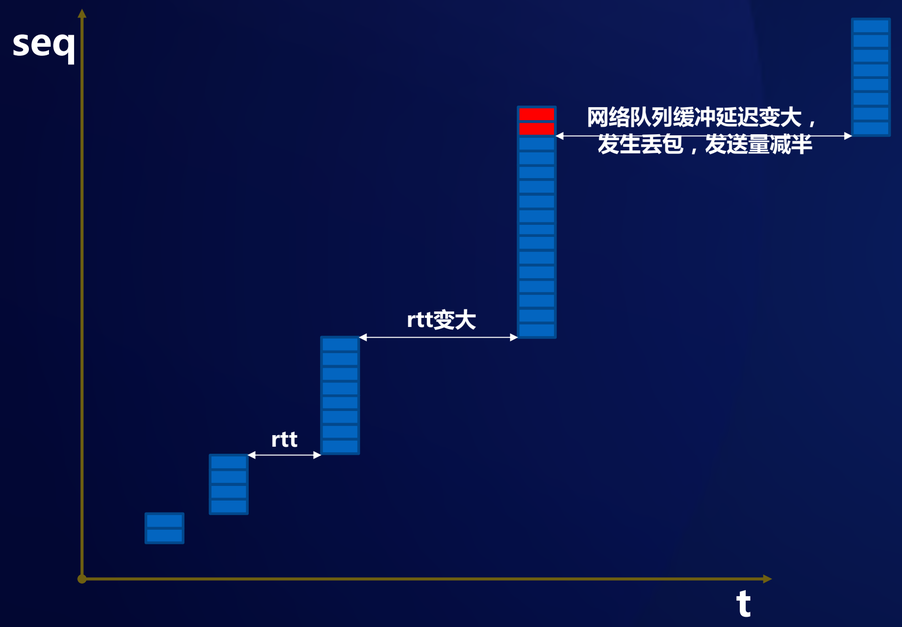


Figure 2-2: Loss intervals and the accompanying weights

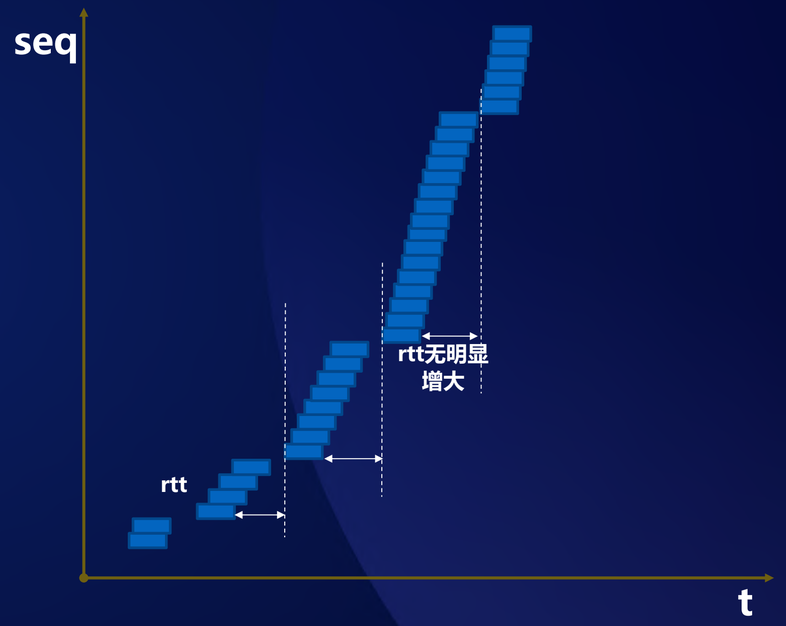
XNTP的Pacing

问题简化：在一个窗口内，计算出了下一次rtt要发生的数据量，怎么发送？
比如：假设rtt=40ms，发送速率8Mbps，每包1kB，则需要发送40个包，怎么发？

方法一：突发式一次性全送交网络发完



方法二：均匀地每隔1ms，送交网络一个包

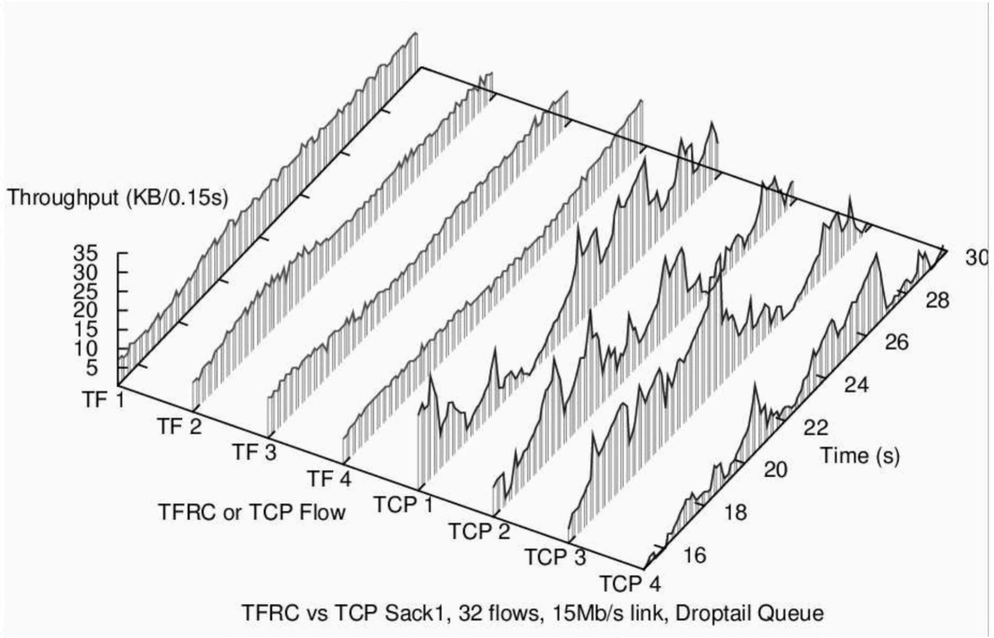


Pacing发送更能真实表达发送速率的意义，更贴近真实网络，
其在消除发送队列缓存延迟方面表现出色，发送更加平稳

XNTP VS TCP

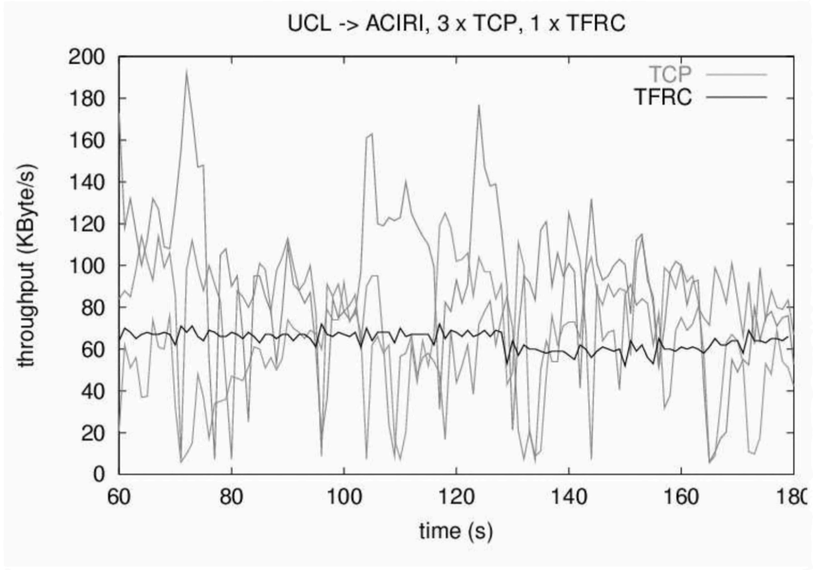
腾讯云 | 云+社区

TFRC Performance (Simulation)



TFRC vs TCP Sack1, 32 flows, 15Mb/s link, Droptail Queue

TFRC Performance (Experimental)



数据来源：https://people.eecs.berkeley.edu/~istoica/classes/cs268/06/notes/7-BeyondTCPx2.pdf

CONTENTS

腾讯云 | 云社区

01 XP2P简介
peer-to-peer对等网络

02 XP2P产品功能
为企业减负，为观众增收

03 XP2P的应用场景
直播、点播、大文件分发

04 XP2P的思考与展望
大有可为

XP2P应用场景

直播

尤其适用于大型直播活动。
卡顿、首屏均要优于CDN网络加速

点播

短中长视频均可。
节省带宽，不输于CDN的体验

文件

类似于BT文件下载。
下载速度更快，存储成本更低

XP2P应用场景

- ✓ 实现了某种程度的多播协议，优化直播带宽传输，即降低网络负载，又降低带宽成本
- ✓ 4K视频加速：有P2P的助力，4K体验将更胜一筹
- ✓ 大型直播活动，如赛事、春节联欢晚会等场景
- ✓ 短视频、常规视频P2P分发
- ✓ 大规模、大文件分发

接入简单，优质服务

CDN配置

云官网控制台启用直播，并配好P2P加速域名。

SDK接入

接入P2P SDK，包体积小，接入简单。

测试上线

发布上线，观察数据，反馈效果，极致优化体验。

运维

快速响应需求，稳定运行，不出故障。

CONTENTS

腾讯云 | 云社区

01 XP2P简介
peer-to-peer对等网络

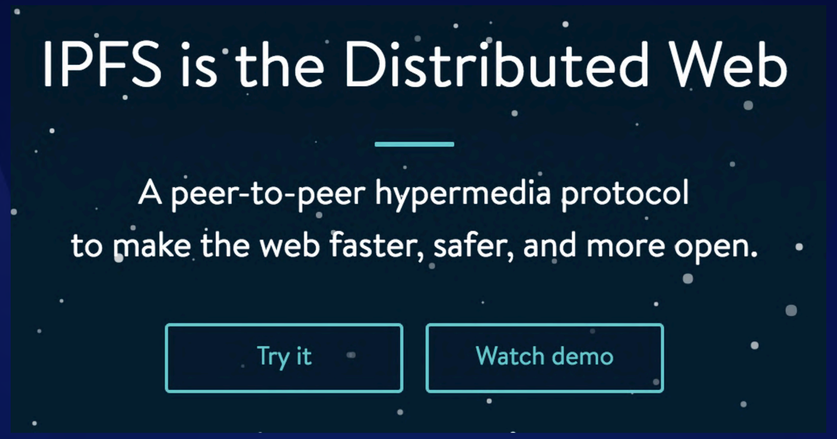
02 XP2P产品功能
为企业减负，为观众增收

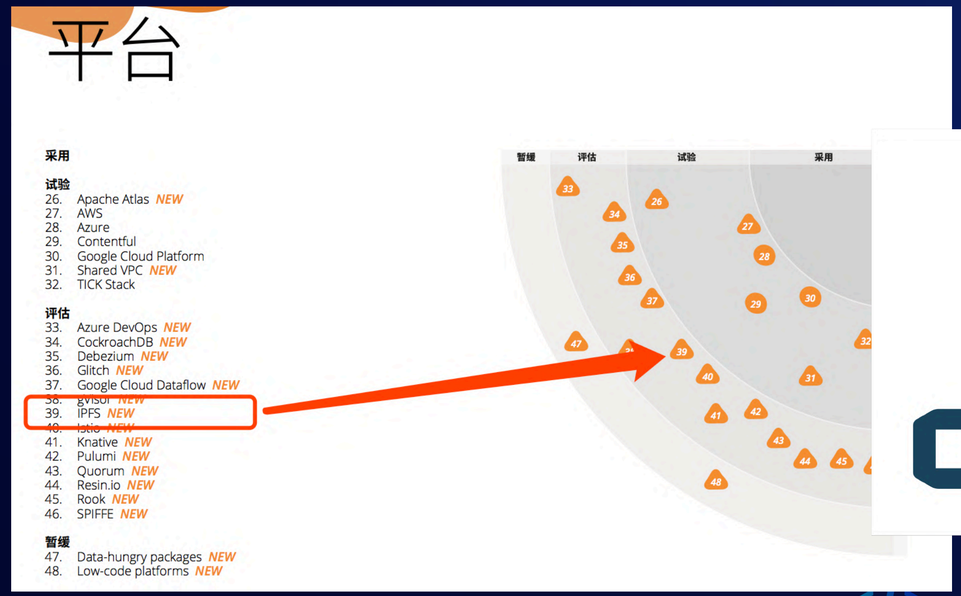
03 XP2P的应用场景
直播、点播、大文件分发

04 XP2P的思考与展望
大有可为

腾讯云XP2P之未来展望

- **IPv6**
- **5G**
- **边缘计算、盒子加速**
- **区块链**
- **标准化**





比肩IPFS和Filecoin，为什么说libp2p将成为区块链和物联网 ...
2019-6-20 · 10月20日，在由晨雾科技主办的中国IPFS开发者沙龙上，晨雾科技的联合创始人徐潇，他带来了名为《libp2p详解》的主题演讲。 libp2p为什么重要呢？今年7月份 ...
https://www.8btc.com/article/294956

引介 | 为什么 ETH2.0 要选择 libp2p？ | 闪马块讯
2019-4-10 · 如果你有持续关注Polkadot、Ethereum2.0或Substrate的开发近况，你应该听说过libp2p。Libp2p是一个便于使用者开发去中心化点对点应用的网络框架，它原先是 ...
https://shanma.pro/tutorial/36439.html

最终，可能会回归最初的愿望：

彻底去中心化！

XP2P的客户

关注云加社区公众号

腾讯云 | 云社区

THANKS