



# 企鹅电竞Web P2P实践

谢庭 腾讯前端工程师  
& 开源爱好者



# 目录

## CONTENTS

1

个人介绍

2

项目背景

3

技术选型

4

技术实践

5

优化与开源

# 个人介绍



深圳大学硕士毕业



企鹅电竞前端开发组员工



游泳、旅游、做开源项目

# *Part 1*

背景



# flash即将退出历史舞台

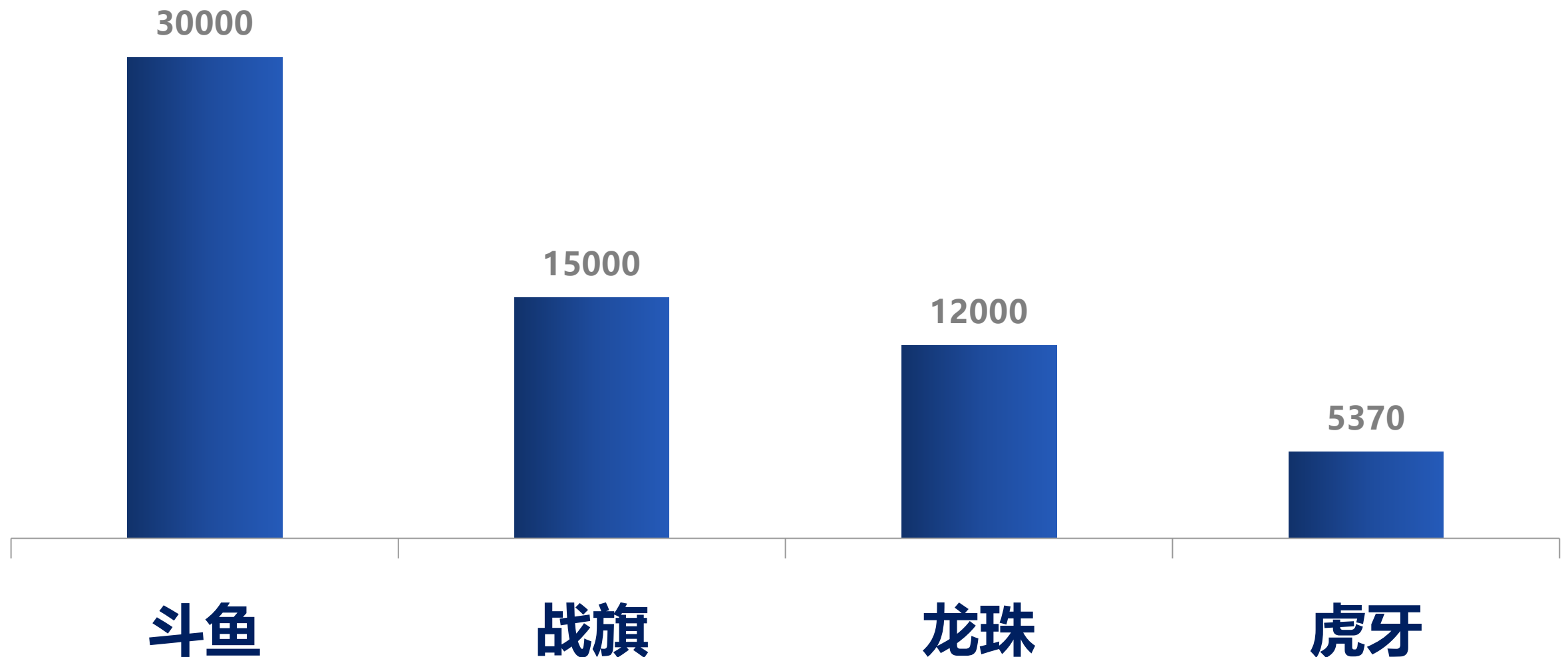
- 2020年将停止更新
- H5不用安装插件
- H5速度更快
- H5更节能
- H5更加安全



# 带宽成本高昂

各大直播平台一个月的带宽成本（万元）

带宽市场价：3000万元/100万人



# 运营商逐渐放开上行带宽

光纤时代，  
中国电信网络技术率先突破  
上行网速最高翻5倍

IT之家  
www.ithome.com

免费升级

宽带产品	原上行	新上行
1000M	20M	100M
500M	20M	50M
200M	10M	30M
100M	10M	20M
50M	4M	10M
20M		4M

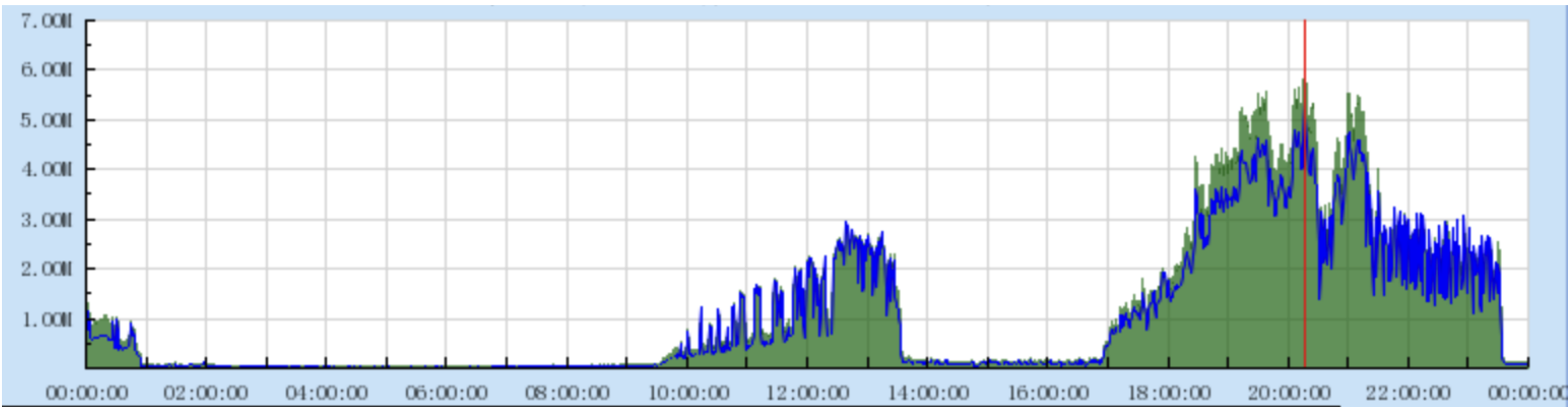
# WebRTC已得到主流浏览器支持

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			66		10.3				
			67		11.2				4
11	3 17	61	68	11.1	11.4	all	67	11.8	7.2
	3 18	62	69	12	12				
		63	70	TP					
			71						

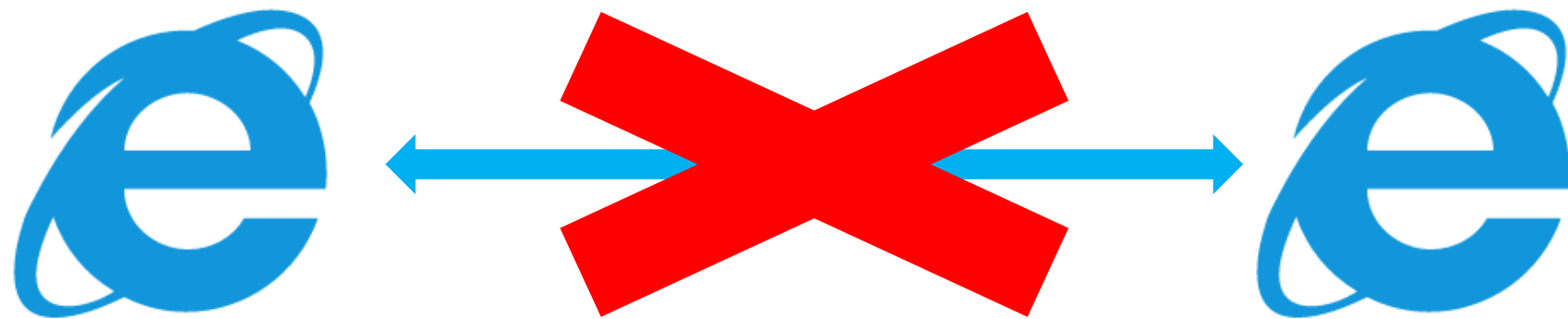




# 电竞直播具备潮汐现象，P2P技术可有效削低峰值带宽



移动端已经接入P2P，但web端...



*Part 2*

**技术选型**

# WebRTC

- 实现音视频流或者其他任意数据的点对点传输
- 不需安装插件，用户无感知
- 使用STUN来协助P2P打洞
- 在打洞失败的情况下用TURN服务器来中转数据
- JS API简单易上手，也可采用第三方SDK



Web  RTC



# RTCDataChannel

- *Same API as WebSockets*
- *Ultra-low latency*
- *Unreliable or reliable*
- *Secure*

# RTCDataChannel

```
var pc = new webkitRTCPeerConnection(servers,
  {optional: [{RtpDataChannels: true}]});

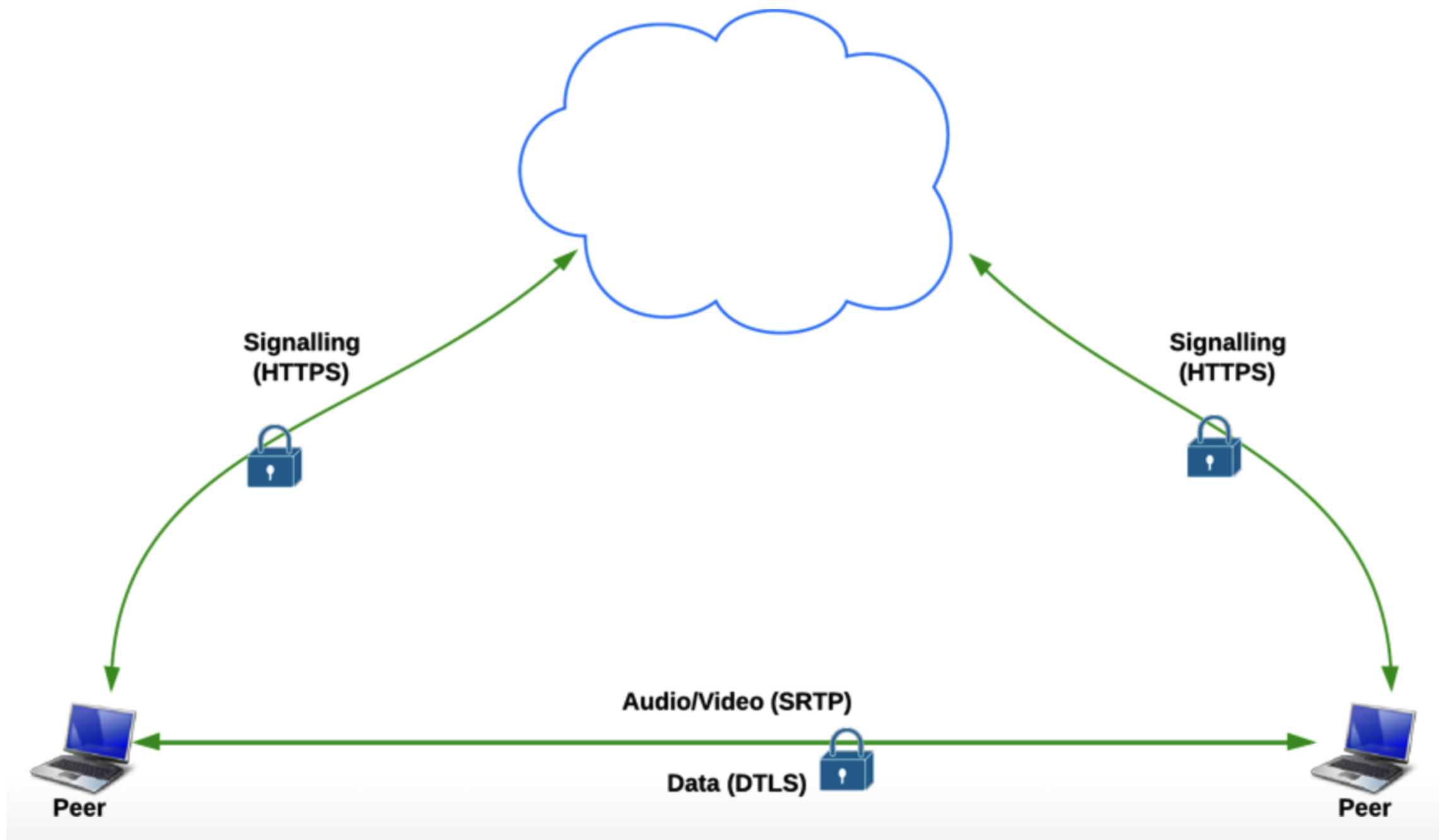
pc.ondatachannel = function(event) {
  receiveChannel = event.channel;
  receiveChannel.onmessage = function(event){
    document.querySelector("div#receive").innerHTML = event.data;
  };
};

sendChannel = pc.createDataChannel("sendDataChannel", {reliable: false});

document.querySelector("button#send").onclick = function (){
  var data = document.querySelector("textarea#send").value;
  sendChannel.send(data);
};
```



# RTCDataChannel





# P2P拓扑对比

拓扑结构	典型算法	数据流向	拓扑变动	适用情景	带宽利用率	实现难度
树型	Fastmesh	单向	频繁	直播	高	较难
网状	Bittorrent	双向	稳定	直播/点播	较低	简单



# Webtorrent

- 经典项目，已被广泛使用
- 可同时运行于浏览器和 `node.js` 环境
- 框架比较重
- 不支持直播
- 纯P2P传输，实际应用价值不大



WebTorrent

# 流媒体协议选型

名称	切片	直播	点播	延迟	兼容性
MP4	✗	✗	✓		好
RTMP	✗	✓	✗	低	差
FLV	✗	✓	✓	低	差
DASH	✓	✓	✓	较低	一般
HLS	✓	✓	✓	较高	好



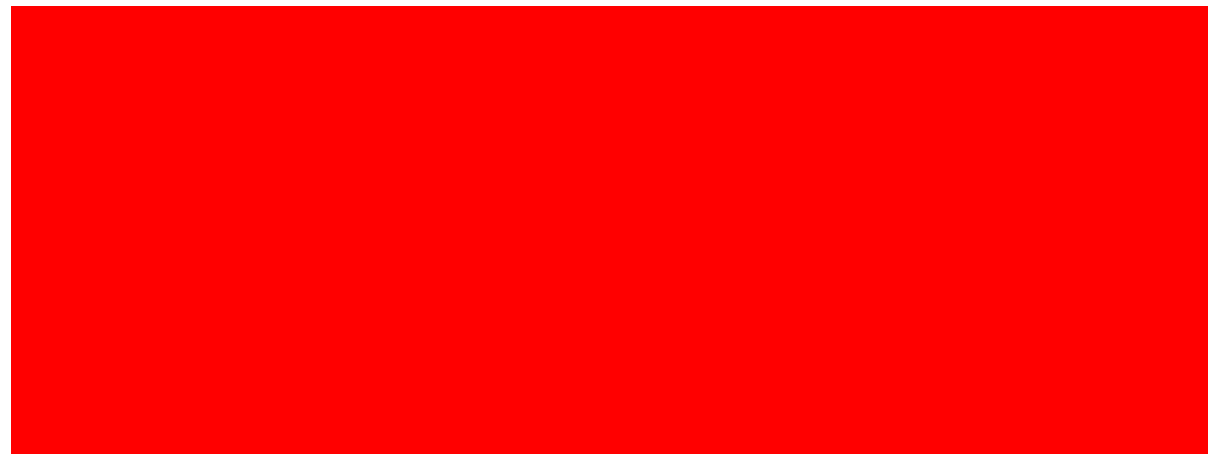
# Hls.js：有了它，我们只需要负责搬运数据

- 解决了HLS协议的平台兼容问题
- 架构优秀，可替换数据请求模块
- API稳定
- 已被多个网站使用，稳定可靠
- iOS Safari 不支持（已原生支持HLS播放）

hls.js

## *Part 3*

**实践**



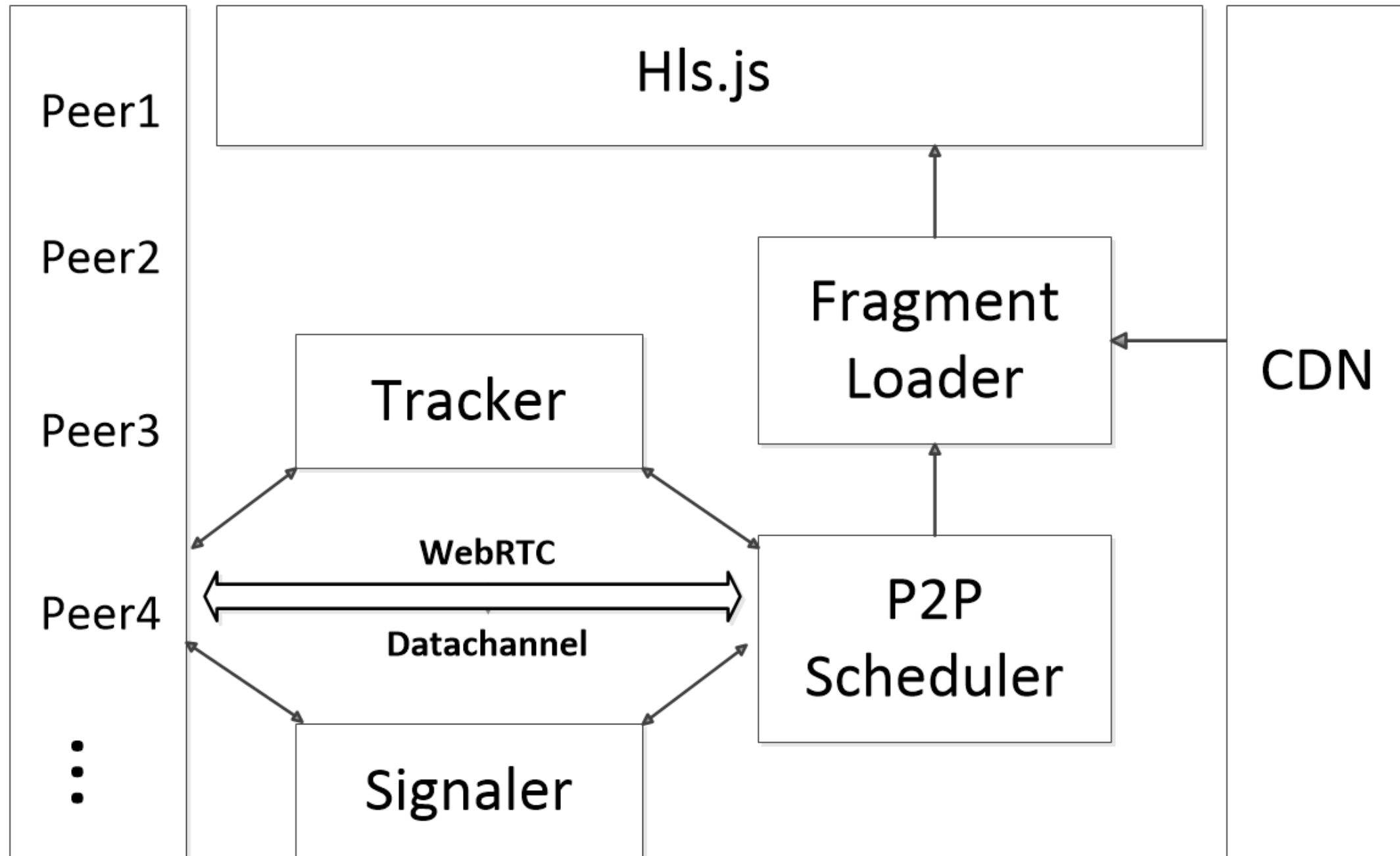
# 设计原则

- 同时适用于较高延迟直播和点播
- 作为hls.js的插件，不改动hls.js源码
- 集成简单，可在现有项目中快速集成
- 高可配置化，用户可以根据特定的使用环境调整各个参数
- 通过有效的调度策略在P2P和CDN之间无缝切换，在保证用户的播放体验的前提下最大化p2p率
- 后台通过IP调度提高P2P连接的成功率和稳定性

# 分层模型

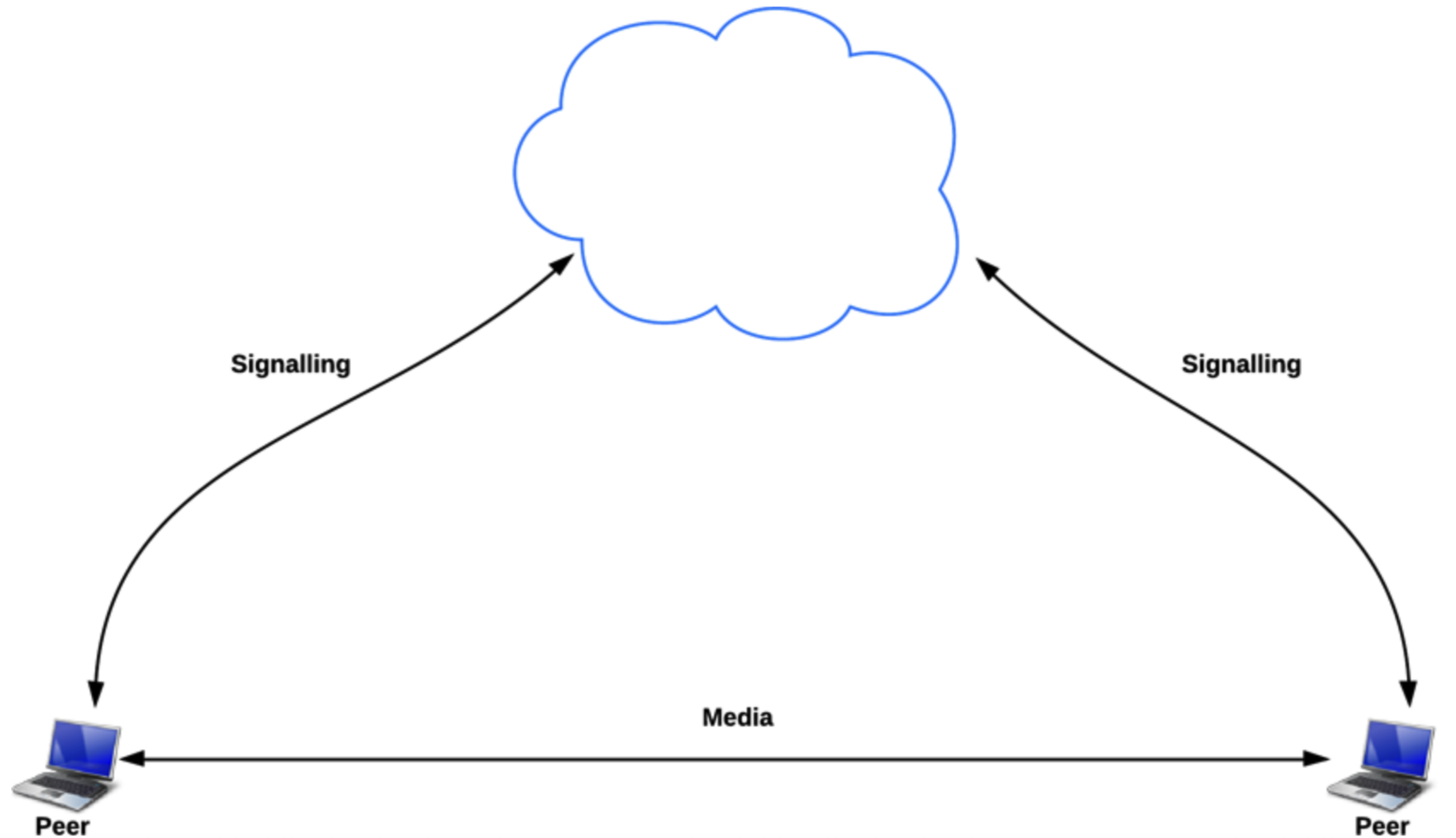


# 整体架构

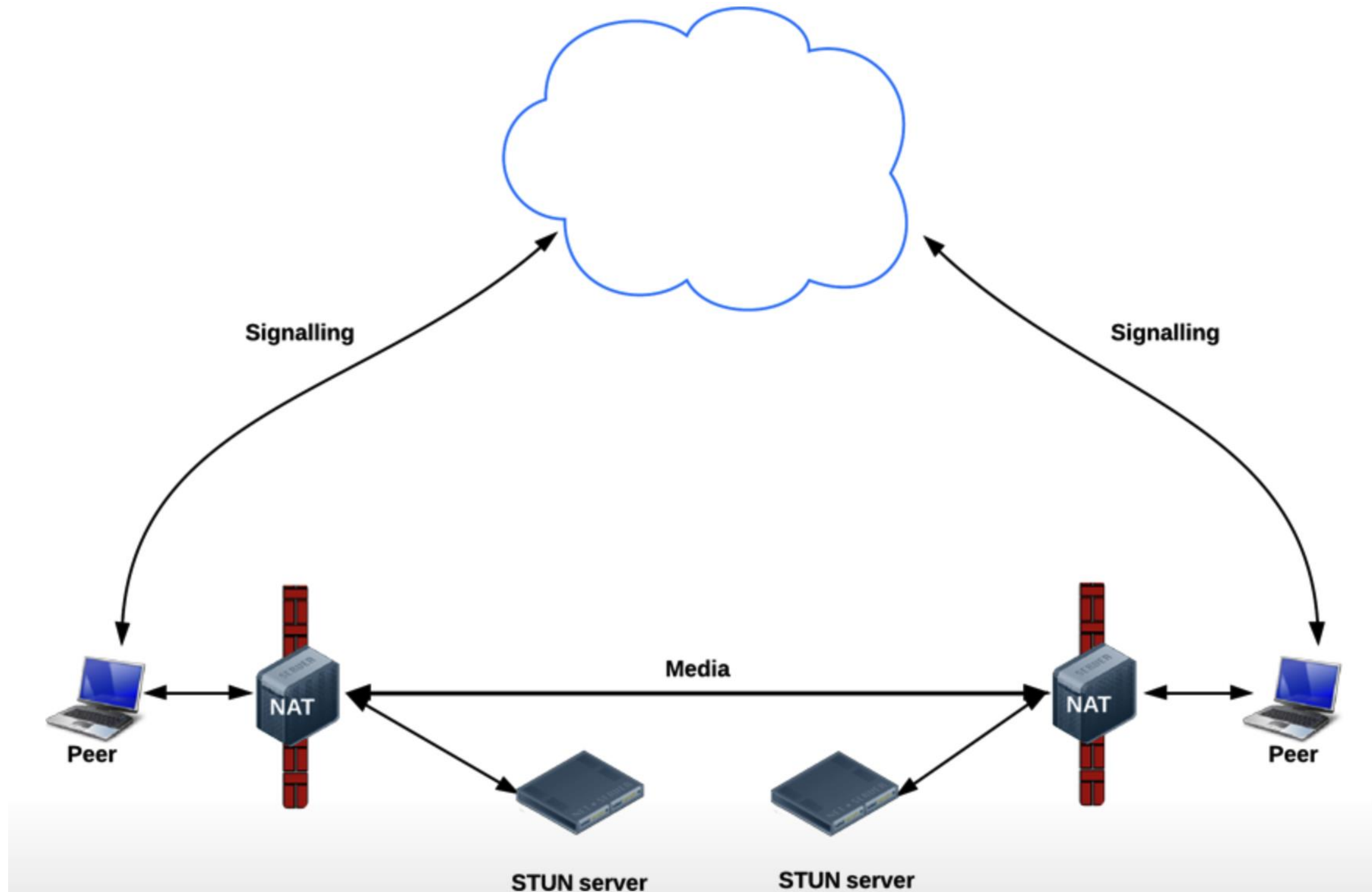




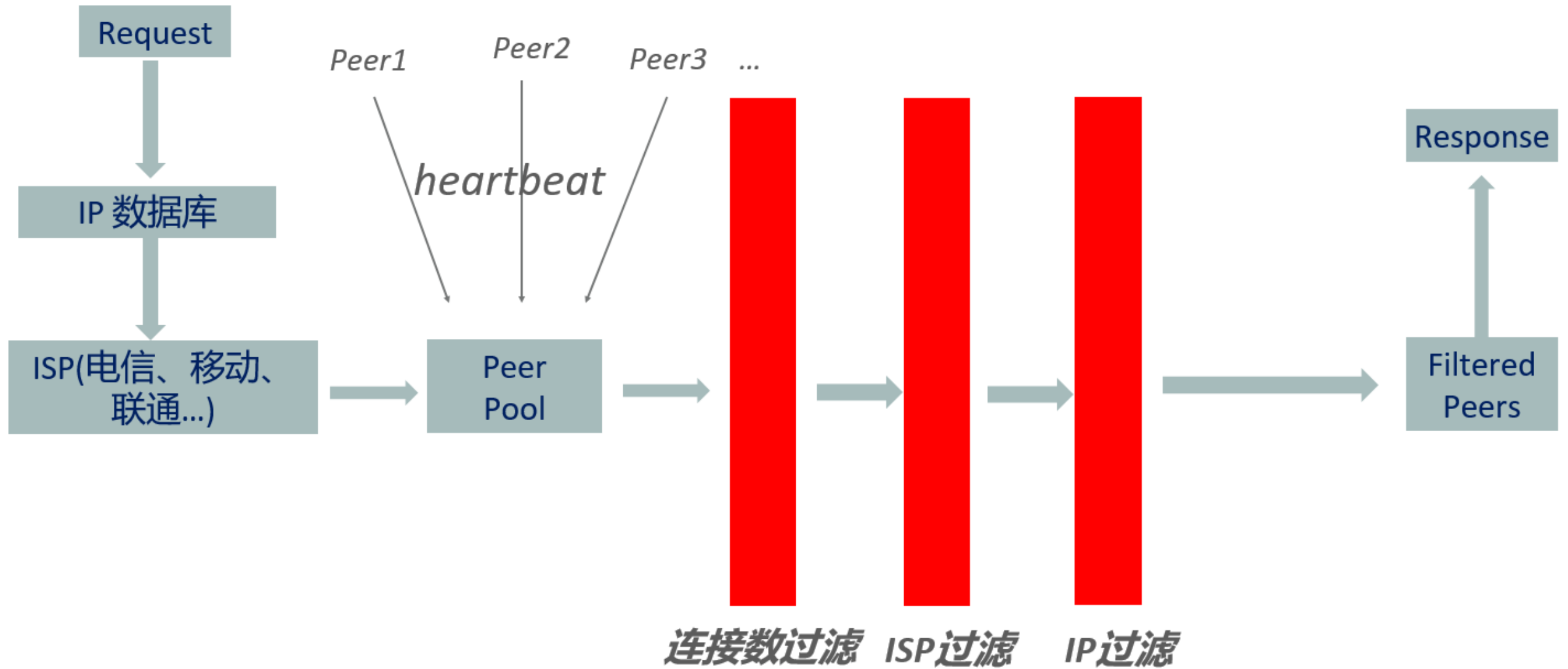
# Signaling Server



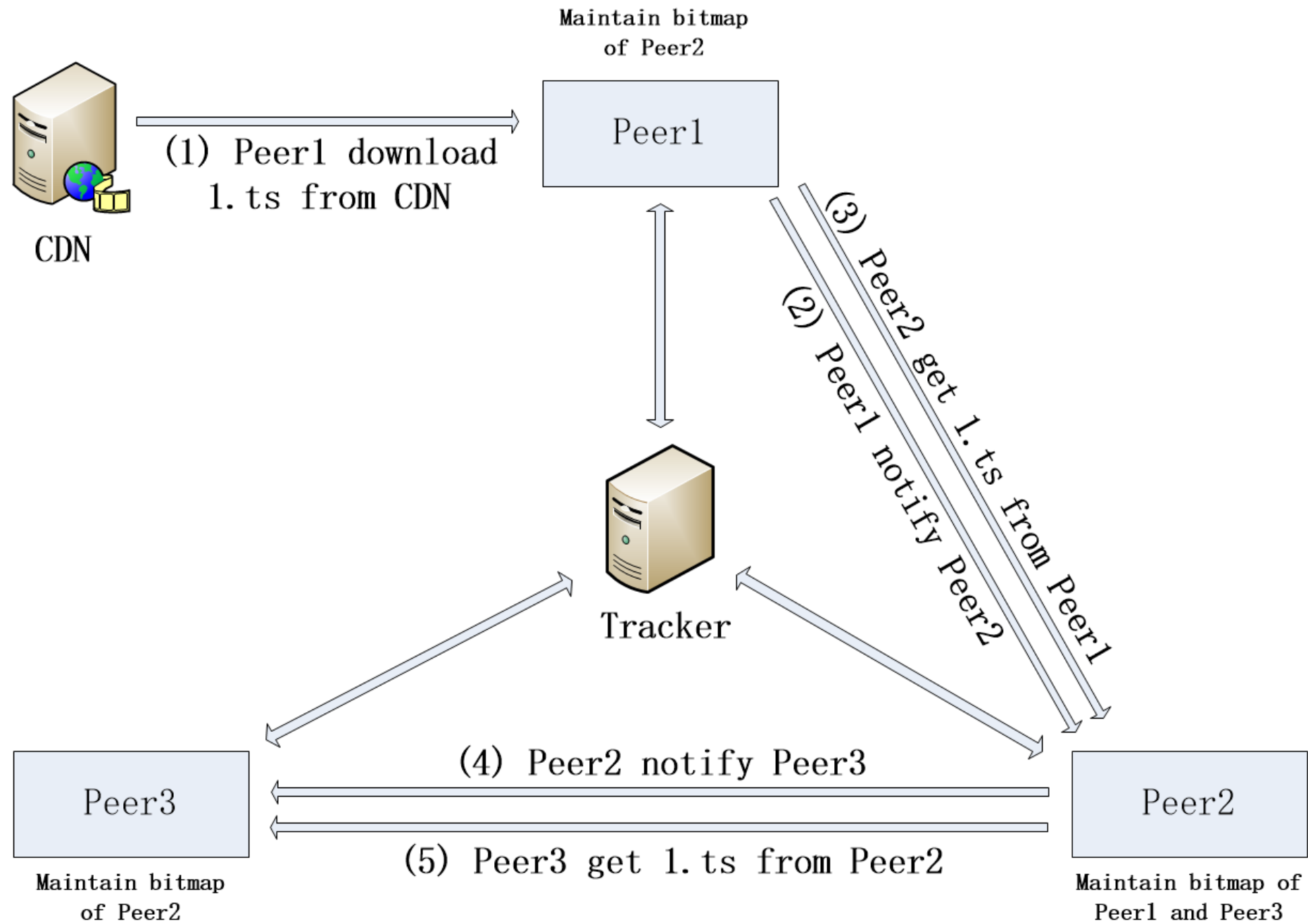
# Signaling Server



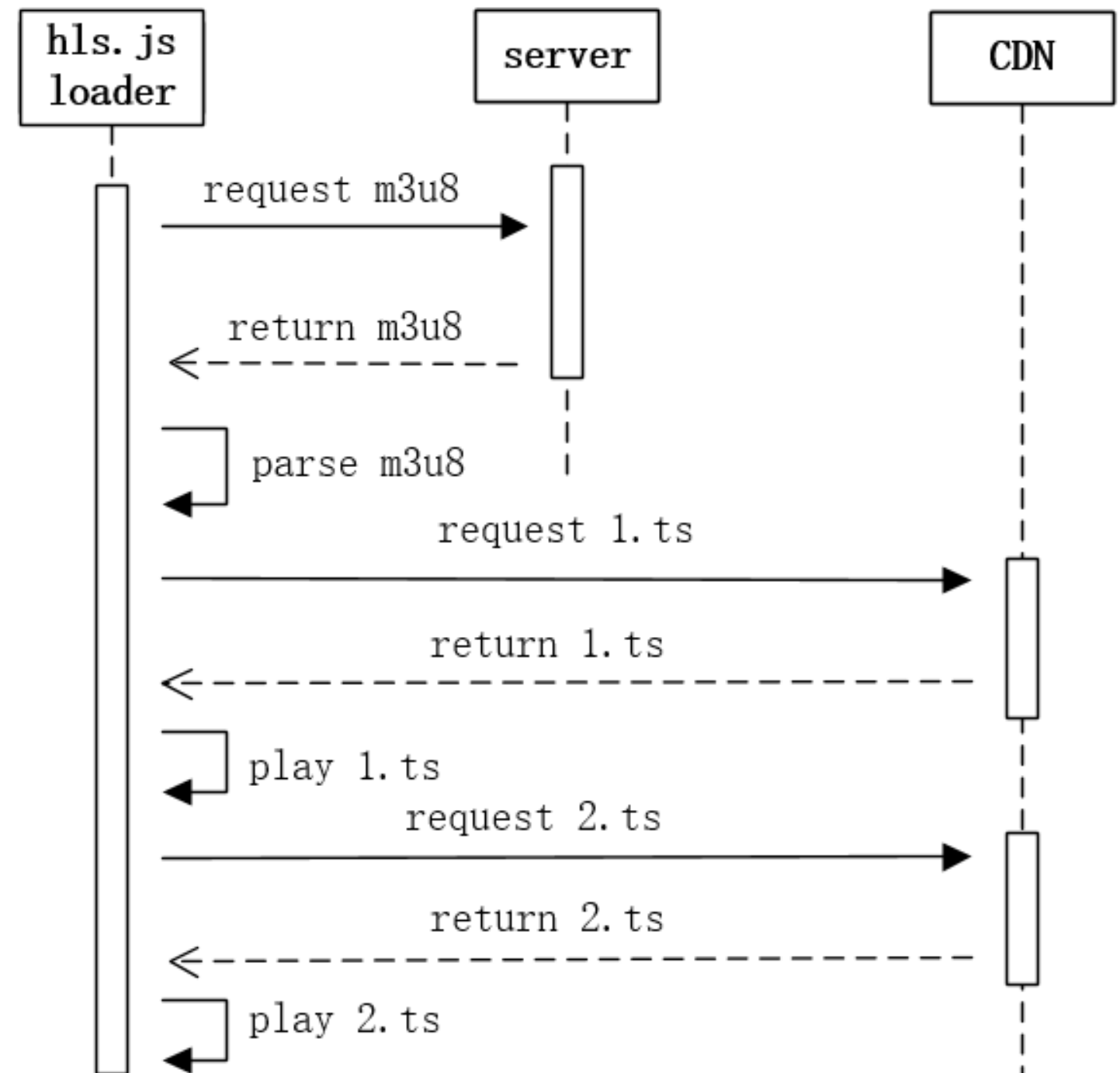
# Tracker Server



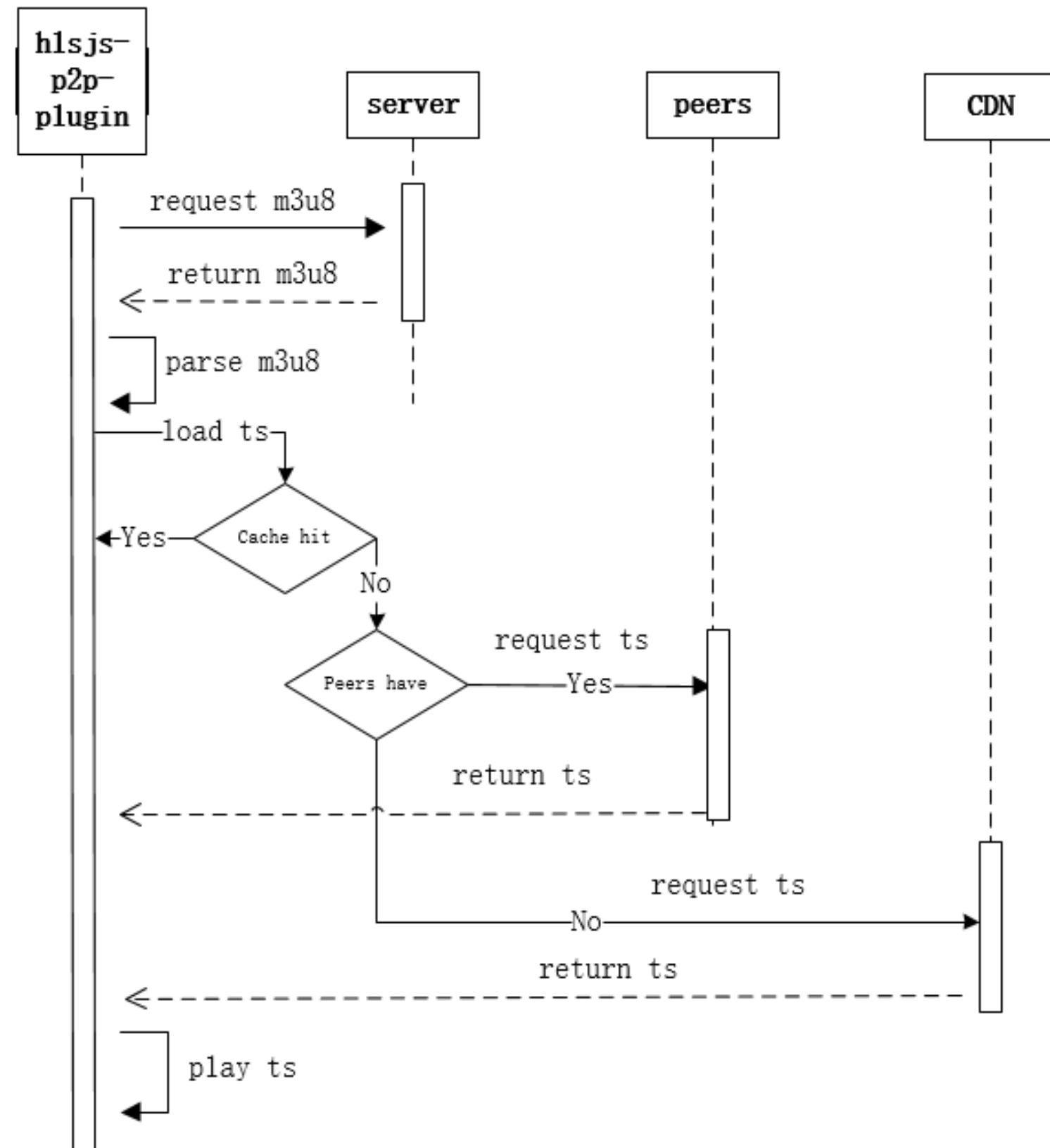
# P2P Scheduler



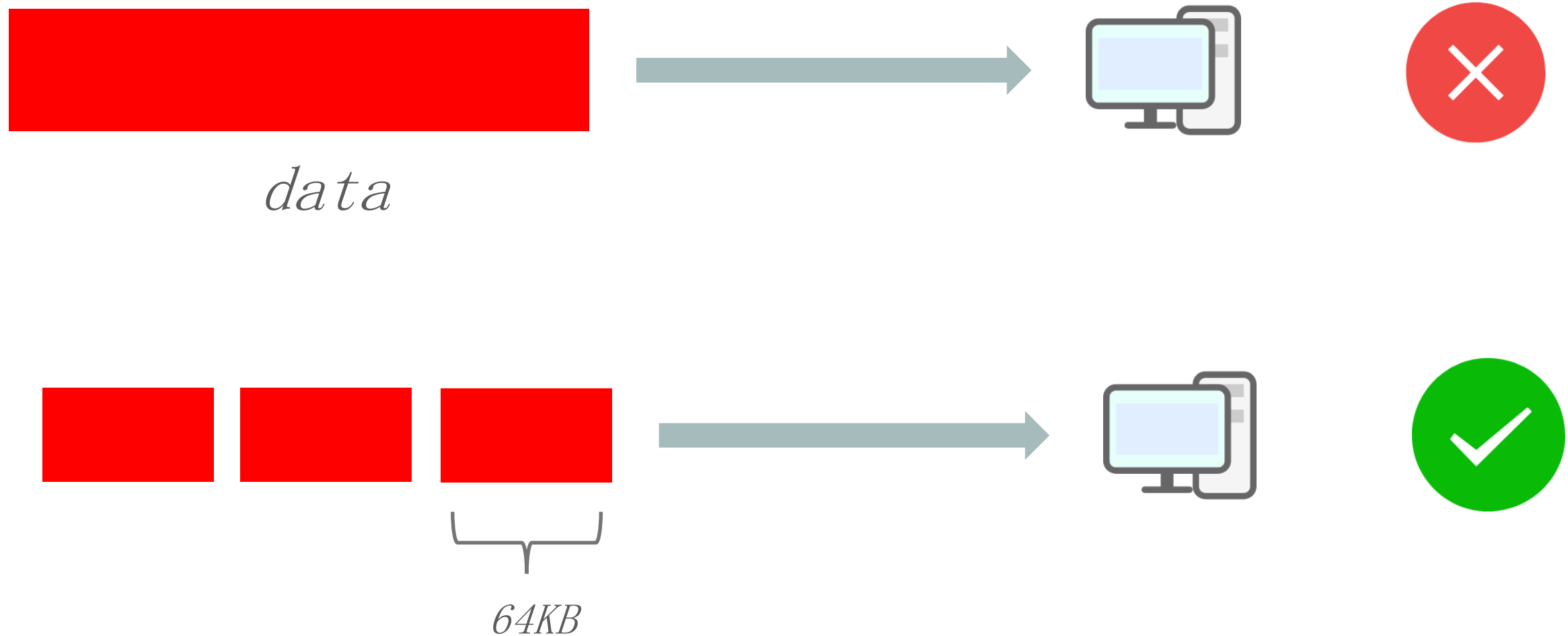
# Hlsjs Fragment Loader



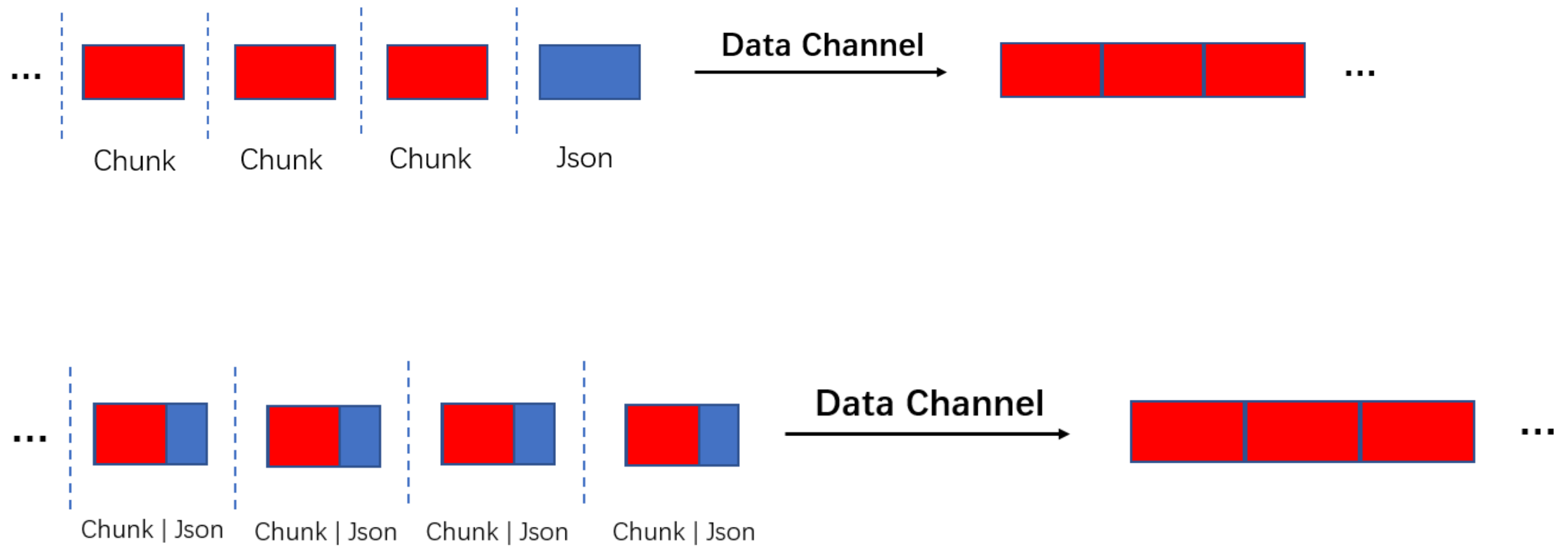
# Modified Fragment Loader



# RTCDataChannel限制



# 传输协议





# Channel问题

*http:example.m3u8*



*Channel Id*

*example*

传输协议更新会造成冲突!

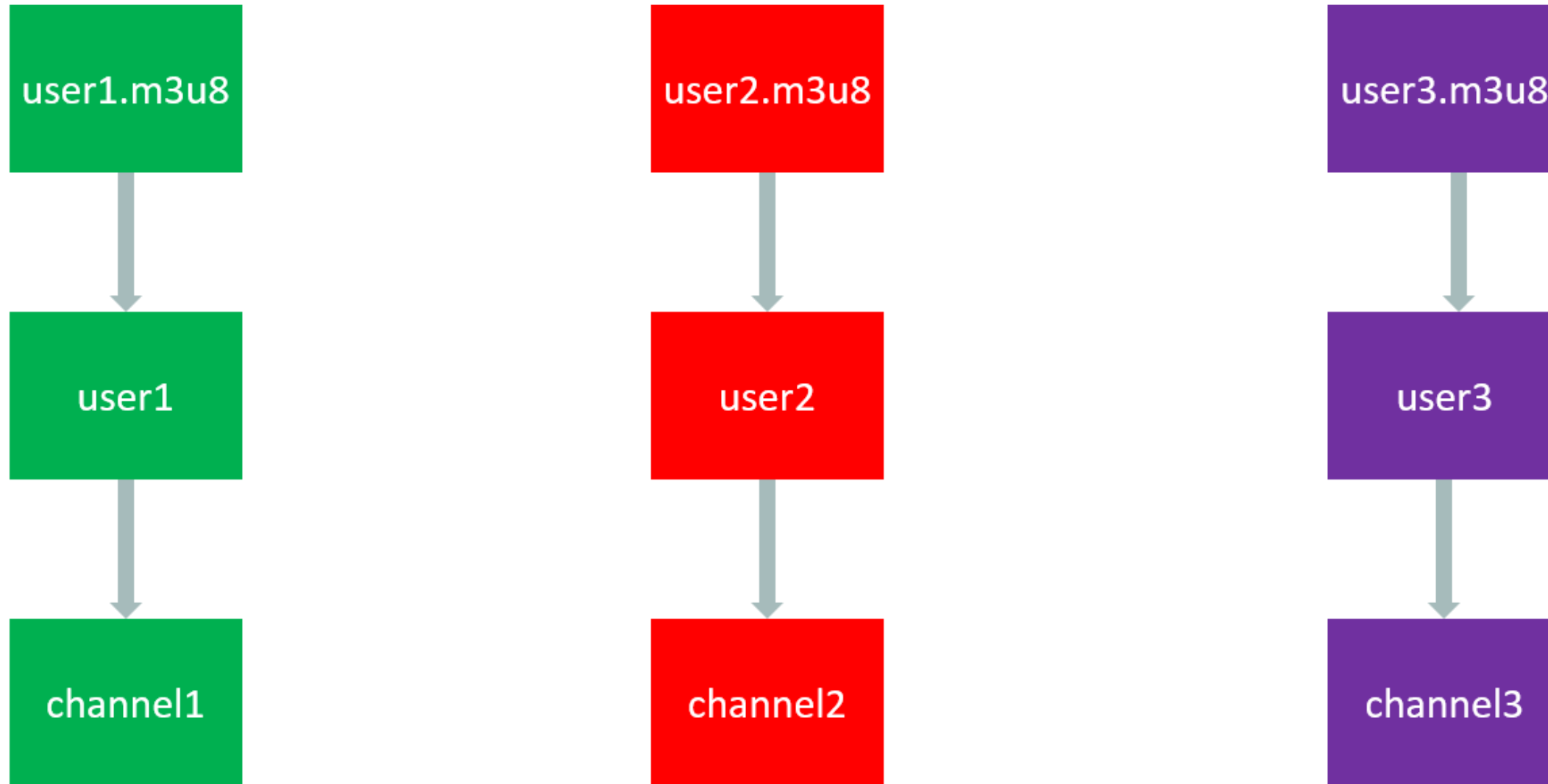


*Channel Id*

*example + 协议版本号*

无冲突问题

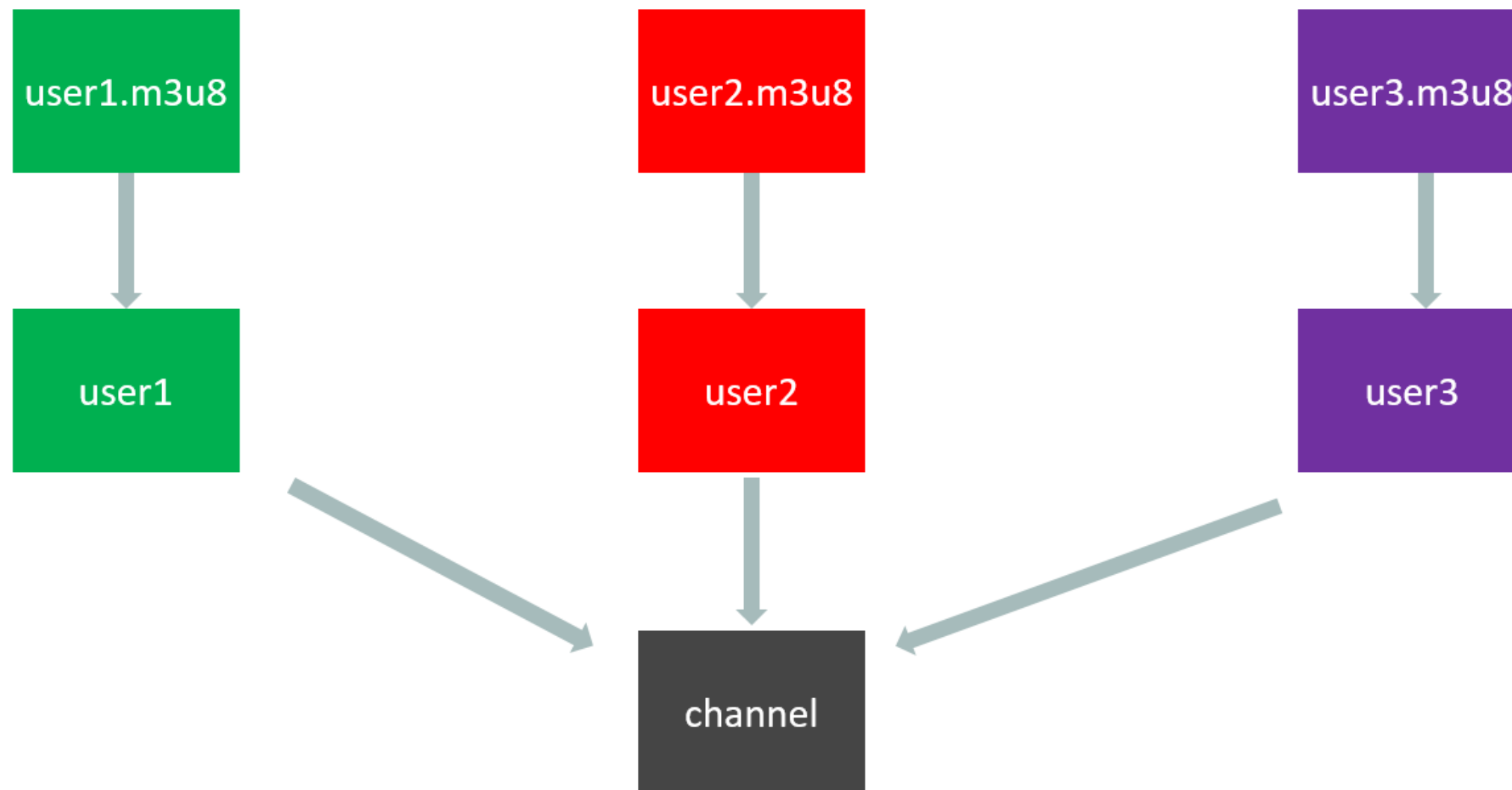
# 动态url问题



# 动态url问题

```
p2pConfig: {  
    channelId: function (url) {  
        return video_name;  
    }  
}
```

# 动态url问题



# Tag与参数调整

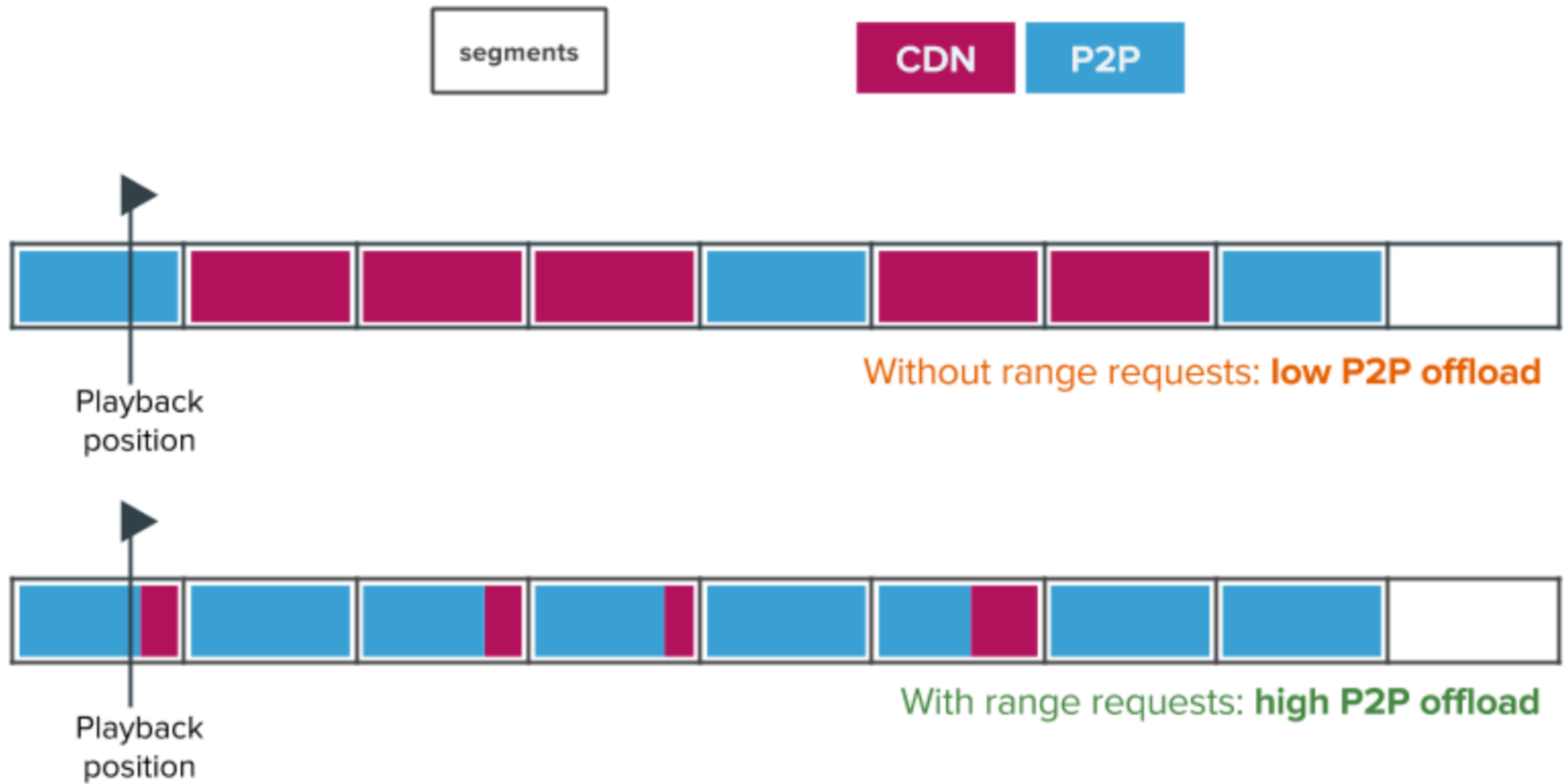
```
p2pConfig = {  
    tag: “1077”           //tag上报并存入数据库  
}
```

1077: 调整了哪些参数...

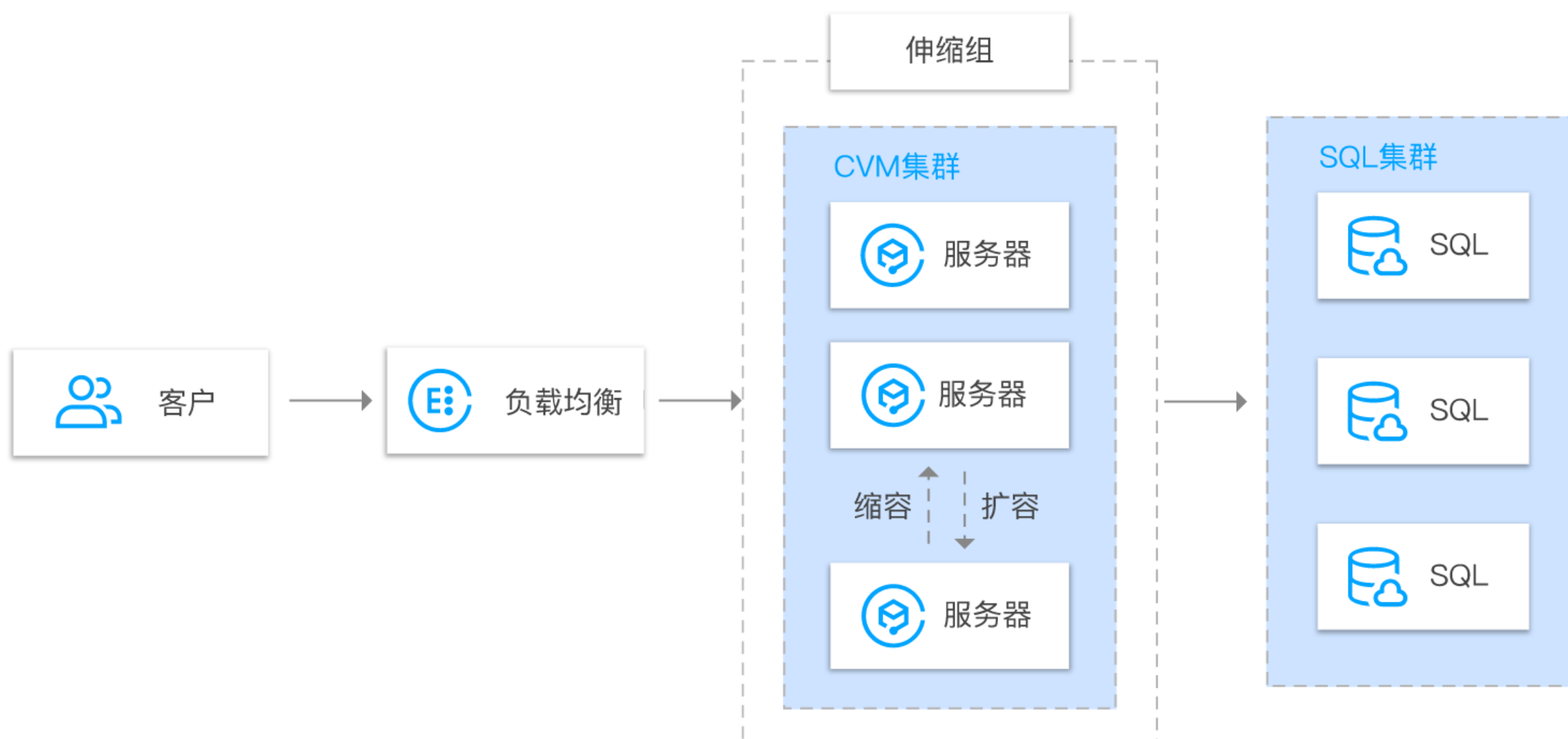
## *Part 4*

# *优化与开源*

# 高码率优化



# 弹性伸缩





# 移动Web P2P策略

蜂窝网络只下载不上传

```
if (navigator.connection) {  
    //Network Information API is supported  
    var type = navigator.connection.type;  
    console.log(type);  
}
```

# 其它优化

- 上行带宽估计 (*speedtest*、动态估算)
- 更精细的调度 (*ISP*、省份、城市、经纬度、在线时长、上行带宽)
- 缩小包体积 (*100+KB*  $\rightarrow$  *50+KB*)
- 与移动端、PC端打通 (需实现*WebRTC*协议栈)
- 雾计算 (利用闲置的路由器、*NAS*等设备)

# 开源项目

不想重复造轮子？

不想踩我们已经踩过的坑？

那就试一试我们的开源项目吧！

扫一扫，用黑科技般的**WebRTC**，  
节省带宽成本的同时，为用户带来流  
畅的播放体验：)



开源免费的Web P2P解决方案！

# 在现有项目中快速集成

## 在现有的hls.js项目中集成

只需要将原有的引入hls.js的script标签如：

```
<script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
```

替换为

```
<script src="https://cdn.jsdelivr.net/npm/cdnbye@latest"></script>
```

# 支持第三方播放器

- *Videojs*
- *FlowPlayer*
- *JWPlayer*
- *Dplayer*
- *Clappr*
- *MediaElement.js*

# END

“

感谢您的聆听.

