

# 历经5代跨越25年的RTC架构演化史

原创

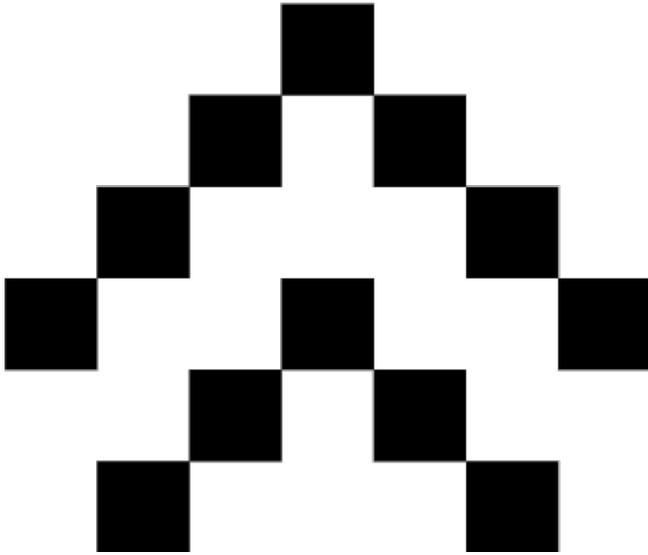
[LiveVideoStack](#) 于 2021-09-29 08:00:00 发布 7730 收藏 14

文章标签: [算法](#) [大数据](#) [emacs](#) [人工智能](#) [iot](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/vn9PLgZvnPs1522s82g/article/details/120558644>

版权



点击上方“LiveVideoStack”关注我们

作者 | 袁荣喜

技术审校 | 吴鹏强

内容编辑 | Alex

RTC架构演化史

视野

#008#

随着移动互联网普及和疫情叠加, 实时通信技术 (RTC) 一时间成为炙手可热的技术方向, RTC从1996年开始到如今已经发展成为一个非常复杂的技术领域, 其包含了网络传输、全局调度、媒体处理算法、媒体编解码、信令协议、输入输出设备、Web、操作系统等相关的技术, 至今为止发展了25年。这期间伴随互联网发展经历了多次技术迭代, 从网络通信架构演化过程来看可把它分为5个阶段 (这里称为5代), 每个阶段RTC从终端技术到通信架构都有大的技术变化。

第1代: MCU阶段 (1996~2004年)

第2代: P2P阶段 (2001~2007年)

第3代: 单SFU阶段 (2003~2012年)

第4代：云SFU阶段（2010~2017年）

第5代：全球实时云通信阶段（2016至今）

可以说每一代都是风起云涌的时代，每一代又都是“卷王“的时代。这里我不去“卷”RTC某一领域的技术细节和应用，而是通过分析这5个阶段的大环境背景和用户侧场景诉求是如何推动RTC技术演化变迁的，来帮助大家更好地理解RTC这个技术领域，也是对自己音视频从业以来的一个总结。

## MCU阶段

上个世纪90年代中期，随着WWW互联网的崛起，大量的终端用户通过电话线拨号涌入互联网，当时上网的终端网速只有56kbps（俗称56K猫），看网页、下载图片有时候需要数十秒钟。网速慢且流量昂贵，但企业和个人都有通过PC联网的强烈需求。

企业场景：通过PC互联实现内部线上会议替代传统的电话会议，企业信息融合通信。

个人场景：通过PC互联实现点对点实时语音沟通。

## H.323

在这种大背景下，ITU（国际电信联盟）基于传统PSTN架构在1996年率先推出H.323/RTP的IP网络多媒体标准，并在H.323协议中提出了MCU和网关的概念，与此同时IETF组织也在1996年发布了RTP规范（RFC1889），并在后续规范中进一步完善了RTP/RTCP体系（RFC3550/3551），ITU后来将这几个规范合并到H.323标准当中。H.323架构如下：

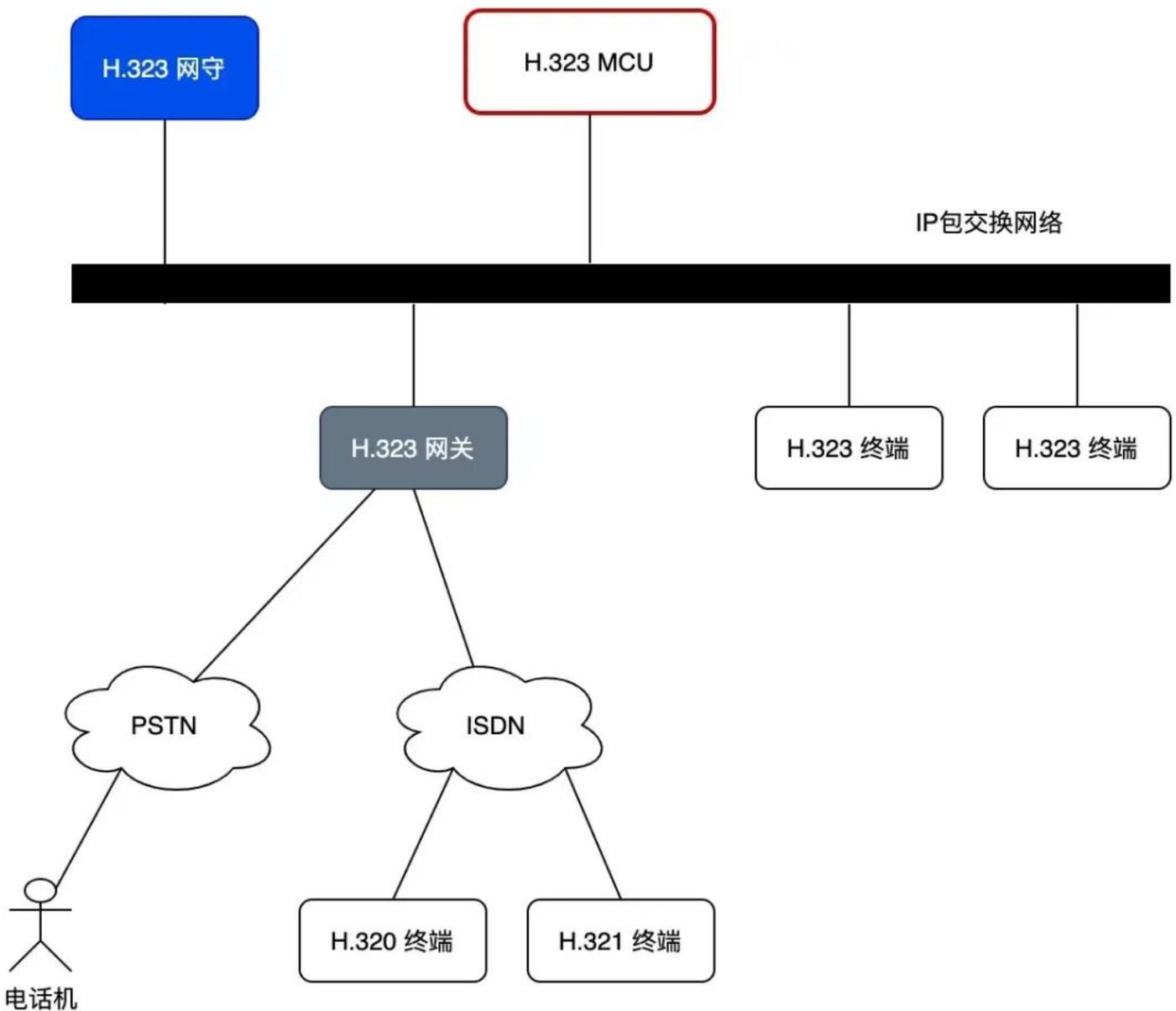


图-1:H.323架构

H.323标准不仅指定了传输协议RTP，它还制定了媒体编码规范（G.7xx/H.261/H.263）、媒体协商协议（H.245）、信令协议（RAS/H.225）、网络安全协议(H.235)等。其中媒体编码影响深远，几乎现在实时音视频编码算法都是从这里演化而来。H.323标准定义了点对点通信和多人会议通信，点对点通信和多人会议通信的差别是点对点通信的媒体是终端直接通信，而多人会议媒体需要经过MCU，MCU按需重编码合流后进行媒体数下发。这里不得不提一句，H.323的ASN.1协议编码技术确实精妙，但也没逃过被淘汰的命运。

### 架构小结

H.323架构后续在互联网应用中受限于网络NAT和终端带宽问题，从而迅速演化成单中心MCU服务模式，MCU既负责网络媒体合流也负责信令接入，即第一代MCU架构：

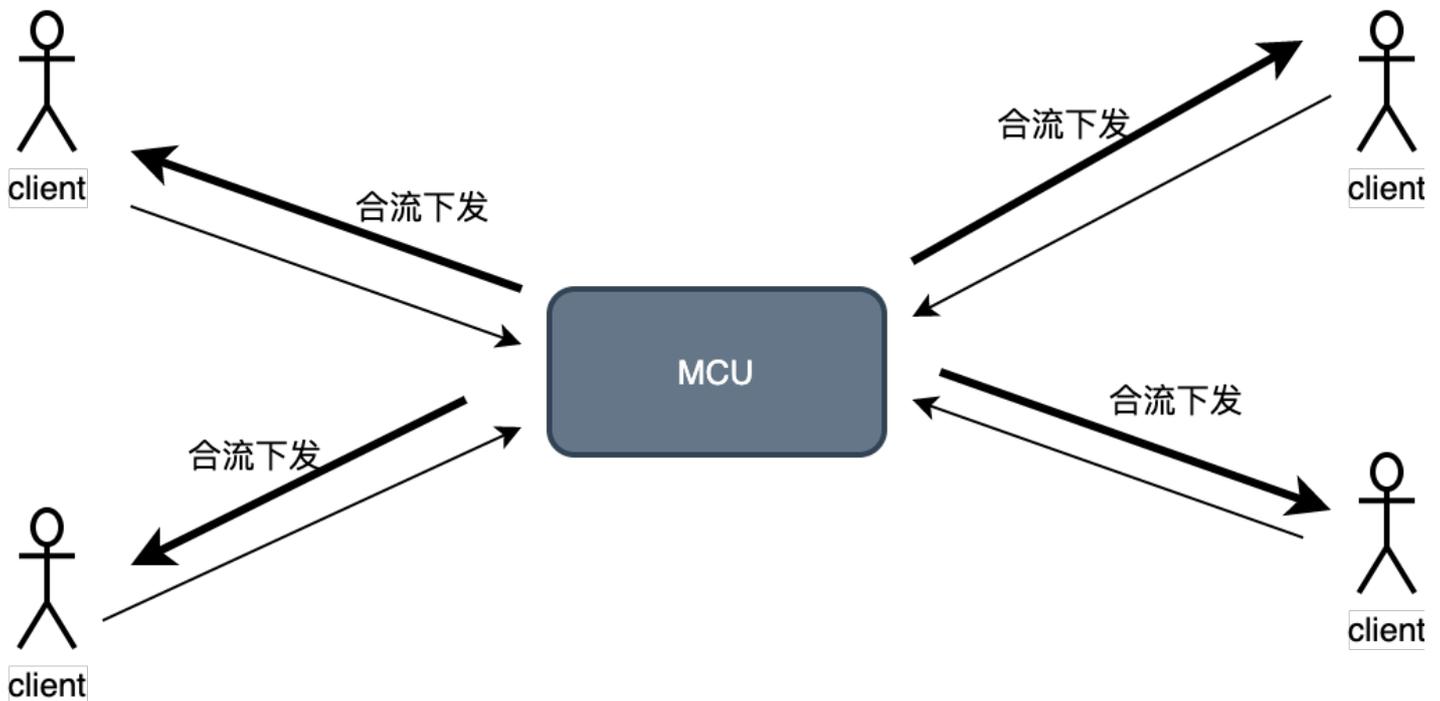


图-2: MCU架构

优点：增强了Client的联通性，合流让终端的下行带宽缩小，让Client在当时带宽下可以获得可通话的质量。

缺点：MCU的计算量大，中心合流带来的扩展性比较差，容易引来系统瓶颈。

MCU架构是在互联网初期终端带宽稀缺且昂贵的条件下产生的架构，终端的形态也各式各样，PC、IP话机、PBX网关等。随着PC互联网的到来，这一切迅速发生了改变。

### P2P阶段

PC互联网的发展促使了PC终端网络的快速迭代。1999年，ITU批准了ADSL技术成为宽带标准，下载速度可达8Mbps，上行可以到1Mbps，ADSL利用原有电信线路搭建起来的终端接入方式迅速铺开。ADSL解决了终端带宽瓶颈问题，一时间IM社交（OICQ/MSN）和P2P下载服务（KaZaA/BT/eMule/迅雷）成为PC互联网的主宰。同时摩尔定律继续发挥效应，终端PC的计算能力进一步加强，已经出现双核CPU。终端设备的升级使得个人用户的互联网通信成为主流并形成了新的实时通信需求，同时也给RTC通信带来新的变革和机会，其主要场景包括：

企业场景：基于VPN-Lan的VoIP通信，企业希望通过互联网接入能力构建自己内部的IP语音通信网络。

个人场景：社交IM场景下的1V1实时音视频通信，免费的IP语音服务替代传统的PTSN收费服务，同时社交类的视频需求变得强烈。

### RTC技术黄金期

正是终端计算能力和带宽能力大幅提升，RTC技术形成了黄金发展时期，出现了众多影响至今的技术和模型。

#### SIP/SDP协议

IETF针对互联网实时通信提出自己的信令标准SIP和媒体协商协议SDP，SIP协议采用UDP+HTTP文本方式实现协议编码，连接过程原语遵循了电话呼叫的流程。其简单直接的方式一经推出后就获得了很多厂商的支持，这时就面临与原来H.323协议交换互通的问题。在此情况下，各大厂商雄心壮志地试图借鉴传统交换机模型移到IP软件交换上来统一各端设备通信，形成了当时流行的软交换系统。

值得思考的是，为什么当时简洁的SIP协议在VoIP上迅速替代了成熟的H.323技术体系？

#### P2P实时通信

终端带宽能力的增强和P2P文件下载技术的大规模应用，促使实时音视频通信可以完全基于终端之间完成，企业的VoIP在这个时候得到了大量的落地（图-3），这种模式更加经济和高效。

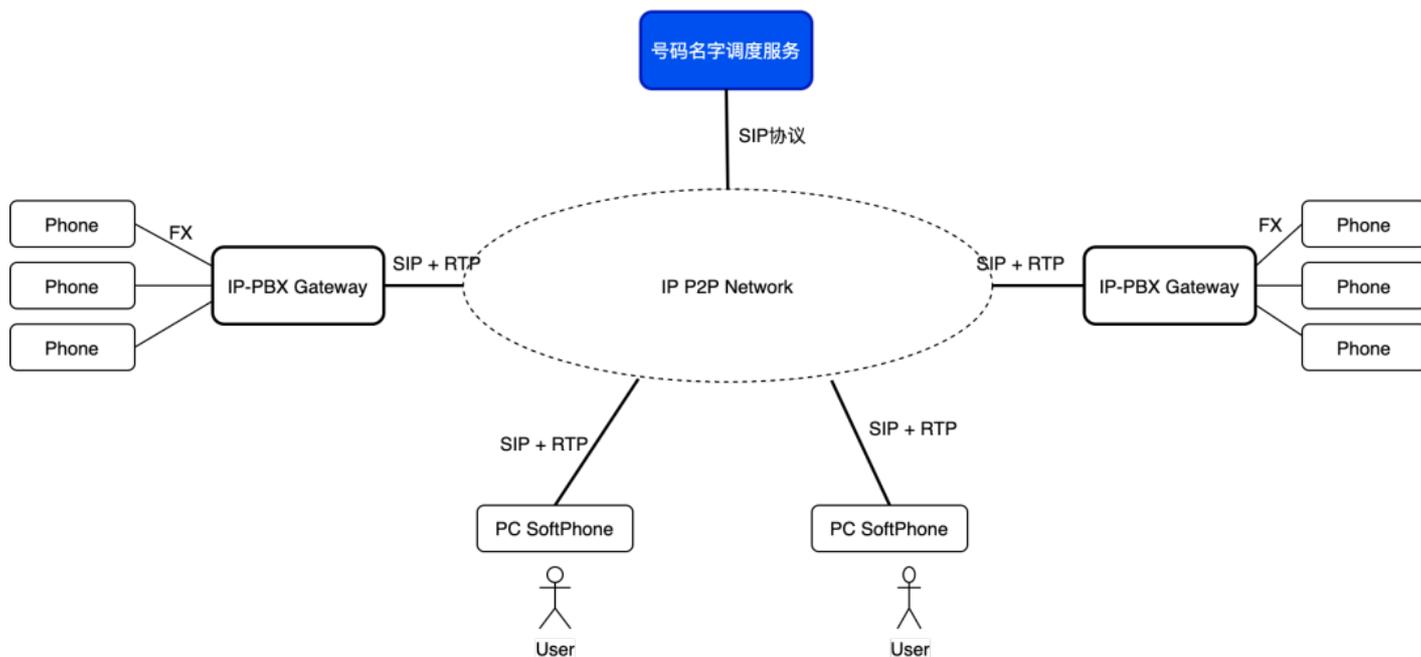


图-3:基于SIP的VoIP架构

除了通信关系上的变化外，在P2P系统中发展出来了很多新的网络传输技术，例如：STUN/TURN穿越技术、ICE技术、RUDP技术、数据切片组帧技术等等。在此之前并没有这些技术标准，这些技术是在当时遇到实际问题时创造出来的。

### GIPS 封神

由于终端大量使用PC，早期的PC系统上并没有对声音的DSP处理，这导致使用PC进行实时语音通信的时候会出现大量的回声、噪声、声音抖动、丢包颤音等问题。当时优秀的声音处理算法都是以DSP捆绑硬件的方式提供，PC获得和传统电话一样的通话质量成为最迫切的需求。

瑞典GIPS公司创造性地将DSP的声音算法移植到PC系统上，充分利用PC强大的计算能力对声音算法进行大胆创新，并开发出了独特的3A算法（AEC/AGC/ANS）、网络抗抖算法（NetEQ）、高质量语音编码器（iSAC）。当时大部分互联网巨头都使用了GIPS，QQ2018PC版本里还有GIPS的版权说明呢。后来GIPS被Google收购，它与GTalk的libjingle合并成为早期WebRTC的一部分，这是后话了。

### 视频编码王者H.264

个人社交服务的兴起，在需求侧实时视频通信已经迫在眉睫，当时实时视频编码分布在两个阵营：ITU的H.263和ISO的MPEG4，H.264是ITU-T以H.26x系列为名称命名的视频编解码技术标准之一，该标准结合了H.323协议中的H.263协议和MPEG-4协议，解决了目前基于软件视频会议MPEG-4标准没有办法与H.323协议的终端兼容问题，使其成为目前最好的视频压缩协议。H.264在计算功耗和压缩率之间有非常好的平衡，是RTC视频通信的基石，纵横视频编码二十载，如今依然是编码器中的王者。

值得思考的是，为什么后来的视频编码器没有替代掉H.264？仅仅是专利问题么？

### 第一个全球实时通信网络——Skype

当时个人用户的社交需求使得各大IM厂商都实现了自己内部的实时音视频通信功能，不过很不幸的是，当时IM厂商主要还是集中在文字消息上，很多软件音视频功能的连通率不到30%，网络一旦波动，通话质量无法保证，甚至会出现断线情况。P2P音乐下载软件KaZaA的作者用独特的**FastTrack**协议构建了第一个全球覆盖的实时通信网络，结合GIPS语音引擎在这个实时网络上进行语音通信，这就是当时著名的Skype。Skype解决了当时社交产品上的几个问题：

- 1) 语音连通性和NAT穿越问题，将连通性提高到95%以上。
- 2) 结合GIPS解决了PC端上通话音质问题。
- 3) 利用P2P的全局索引与UDP overlay技术解决了全球覆盖问题。
- 4) 利用P2P的超级节点技术优化通信路径延迟和跨国跨运营商网络问题，全球通话延迟在500ms以下。
- 5) 利用数字加密技术解决端到端媒体数据安全问题。

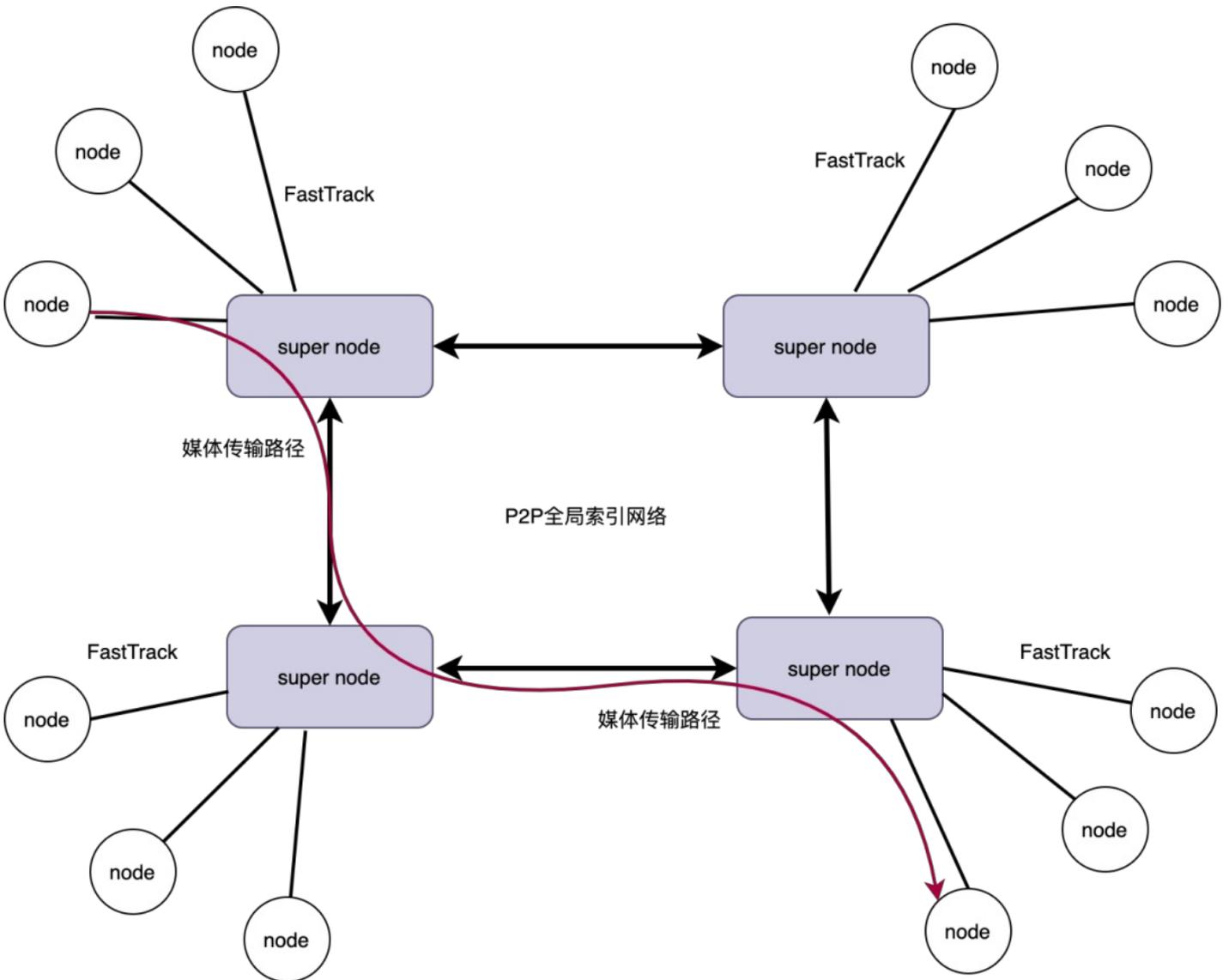


图-4:Skype网络架构示意图

Skype开创了非常多的新技术，主要包括以下几方面：

- 1) 全局索引调度技术，可以通过P2P算法在网络中找到并调度所需要的通话资源节点。
- 2) Lastmile就近super node接入，而不是单纯的点对点直连。
- 3) 路由实时计算，通信时实时计算和调整传输路径。

4) STUN/TURN/ICE技术, Skype首次提出并使用ICE技术。

5) 端到端媒体数据加密技术, Skype首次大规模使用数字签名和媒体加密 (RSA/AES/rc4) 保证通话安全。

6) UDP Overlay实时通信覆盖网络。

#### 架构小结

由于PC终端能力增强, RTC的架构从MCU架构演化成了基于每个流的P2P Mesh架构(图-5), 这种架构脱离中心服务的束缚, 转变成去中心化的服务模型。最先把这个模型用到极致的是Skype, 它的语音服务质量和稳定性堪比PSTN。

2005年, Skype被eBay以26亿美金收购。后面诸多社交IM服务产品跟进形成内卷的局面, 就连技术宗神Google当时也山寨了一个GTalk。

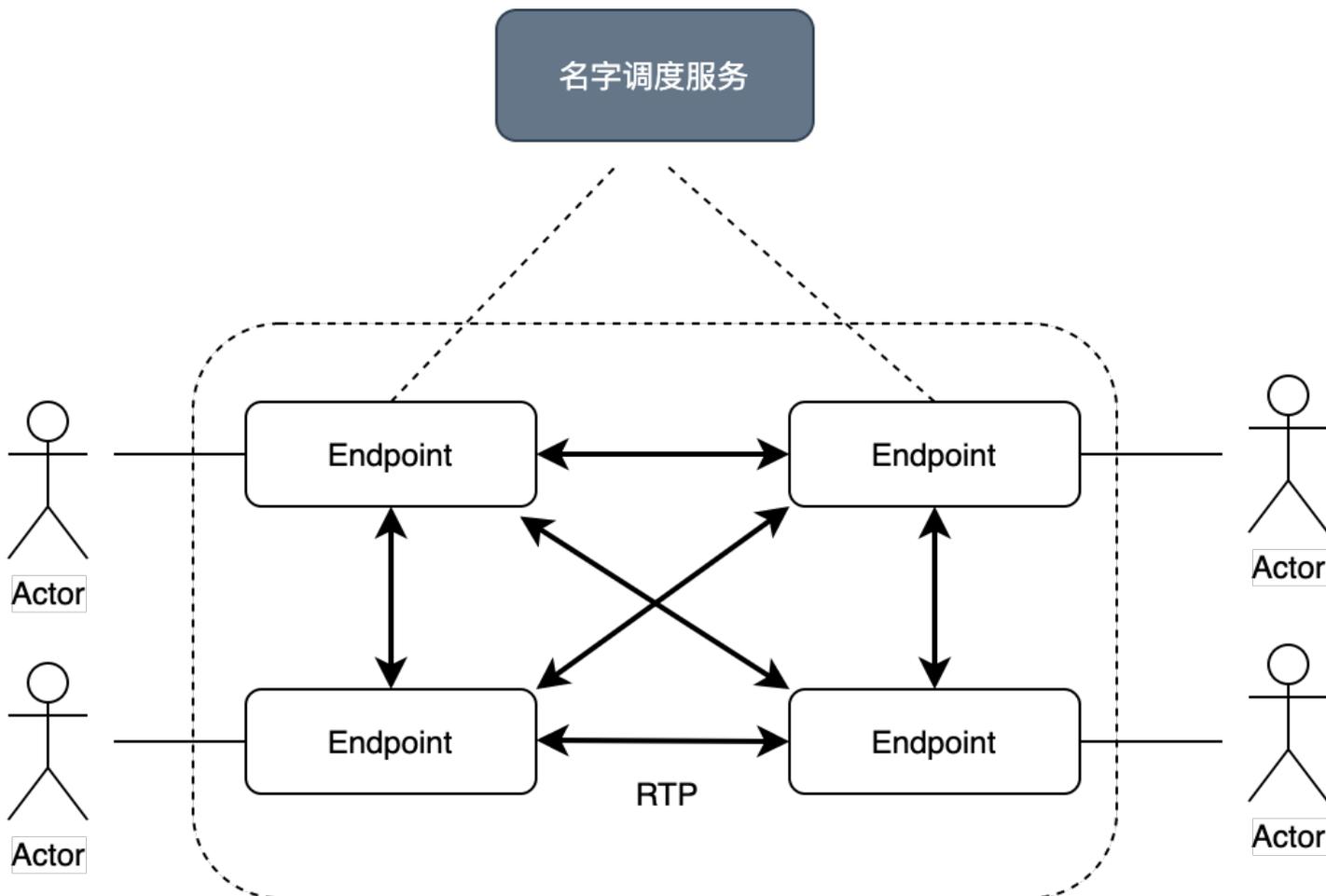


图-5:RTC P2P架构示意图

**优点:** 后台服务轻量, 架构扩展性强, 开发编程简单, 只需要名字定位和信令协商等。通信由Endpoint之间自己完成, 适合1V1方式通信。

**缺点:** 服务质量完全依赖于各个终端的网络环境和稳定性, 连通性差, 各端能力差异会引来通信差异。对于跨国跨运营商的通信无法保证通话QoS, 去中心化后对于敏感通话信息无法进行中心管理监控, 有政治安全风险。

#### SFU阶段

网络MMO RPG游戏的兴起使得游戏社区开始萌芽，游戏领域的实时沟通成了一个强需求，国外首先诞生了基于speex语音引擎的TS软件。在国内，由于传奇和魔兽世界的流行，国内模仿TS软件诞生了iSpeaker和YY。游戏内部实时协作不仅对通信质量和稳定性要求很高，也同时要求减少PC的资源占用。由于大量用户使用类直播的多人沟通方式，在游戏语音软件首先出现了Web 2.0模式，用户自发组织成了一个内容社区（家族和频道），频道内部除了游戏沟通外，平时也会出现直播类娱乐节目。

非典时期，新浪UC在IM竞争中败北，转而研发出基于多视频互动的直播聊天室大放异彩，后续从中又演化出9158、六间房等产品。2011年底，YY在其大频道加入单视频直播模式，直播方式在这个阶段成了主要流量。由于直播表演可以更好地展现主播形象，出现了专业声卡硬件设备、美颜摄像头设备、虚拟直播伴侣软件。

这个阶段企业应用发展相对缓慢，但值得一提的是，视频会议软件逐步开始代替企业VoIP信息融合通信。这个时期出现了WebEx、Ploycom等视频会议厂商，视频会议软件采用简单、容易部署的架构（类似现在的SaaS 服务）。

值得思考的是，为什么单纯的视频会议会替代掉融合PSTN的VoIP？

## 社区模式下的RTC技术

社区模式使得RTC重新回到了传输的Client/Server模式，Client负责个人媒体数据收发和录制播放，Server负责控制和传输转发，前后端侧重的技术点不一样，给社区场景带来了后面的技术转变。

### 基于媒体流的SFU

频道对象的实时通话不可能基于MCU去混流转发，为了满足客户端按需拉流的需求，发展出了基于Publish/Subscribe的流订阅技术。这个技术在游戏语音场景首先实现，后被延展到秀场直播和视频会议场景。

### 万路语音频道

为了满足频道多人互动需求，YY在2008年左右开发出万人语音频道，很快iSpeaker和其他的游戏语音厂商也开始跟进。万人语音频道还是有很大的技术挑战，当时大部分厂商采用的是单物理机的SFU模式，相当于单个机器并发UDP包事件在20万/秒左右。在2012年的ChinaJoy上，YY利用自己的分布式UDP Overlay架构成功支持了单频道120万人线上观看直播，这是RTC技术上的里程碑。后来直播逐步Web化，媒体分发开始大量使用CDN，催生了CDN与直播的发展，这是另外的话题。

### Proxy接入技术

基于流的SFU模型虽然利用了Server强大的控制转发能力，减轻了终端上传压力，但不能解决用户接入的网络分区和长距离传输问题，在SFU架构基础上，有些厂商开始引入Proxy接入技术来解决客户端接入分区问题（见图-6）。

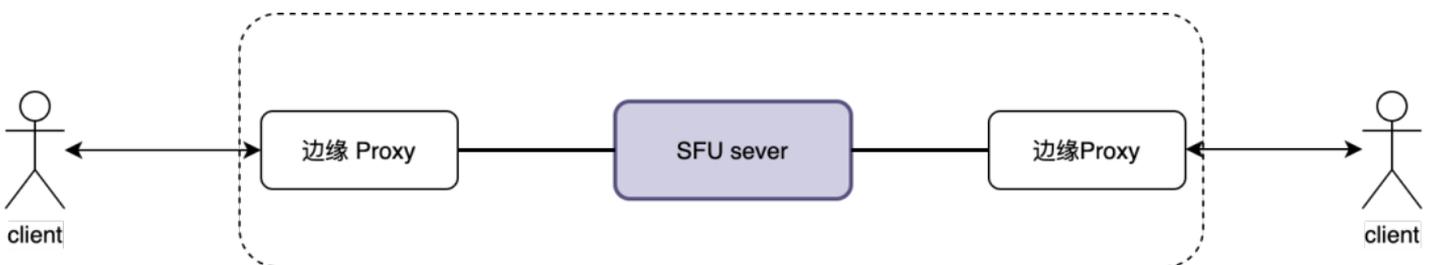


图-6: SFU与Proxy示意图

Proxy可以很好地解决Client就近接入的问题，Proxy利用机房间的稳定的传输网络或者专线能力，可以提供比较稳定的传输质量。Proxy具有数据同机交换转发能力，但转发关系还需要中央的SFU来进行控制决策。后面的云SFU架构是在Proxy基础上演化而来的。

### 前端媒体处理

这个阶段出现了频道直播场景，表演性质决定了主播需要展示更好的一面。在这种强需求推动下，首先出现了媒体前处理的硬件设备，而后又出现了带有滤镜、特效、简单美颜的PC直播伴侣软件，其中最具有代表性的是MVBox和六间房直播伴侣。

## Web化与开源

PC互联网的后期，用户增长出现了瓶颈，为了降低用户使用门槛和增加留存，市面上产品都开始Web化，当时最具代表性的是Web桌面应用。RTC技术领域也不例外，在Skype早期就使用了浏览器激活协议在Web上激活native软件并进行呼叫，这个方式被WebEx和Zoom一直沿用到现在，如今所有视频会议产品都有Web入会的功能。

2010年，Google收购GIPS引擎和ON2，将GIPS、ON2（VP8前身）和GTalk通信库合并；2011年，Google将库纳入Chrome体系并开源，并命名为WebRTC。第一版本的WebRTC只提供native库，2012年它获得各大浏览器厂商支持并纳入W3C标准，自此才基本能支持Web形式的实时音视频通信。WebRTC的开源是RTC技术领域的里程碑事件，大大降低了RTC开发的门槛，催生了后来移动互联网RTC应用的大时代。

值得思考的是，实时音视频一直游离在Web领域的边缘，为什么一直没有出现主流的实时音视频产品？截止如今native依然是RTC领域的主流，假如WebRTC早5年出现或许会成为RTMP这样的CDN分发技术。

## 架构小结

这个阶段RTC出现了大规模直播表演类的应用，使得SFU架构迅速成型并大规模应用，SFU架构很好地分离了客户端和服务端功能，SFU专注于控制与传输转发。由于应用场景变化和终端计算能力进一步增强，终端出现了图像声音相关的前处理应用。SFU架构简洁（见图-7），适合大规模和超大规模实时音视频互动场景。

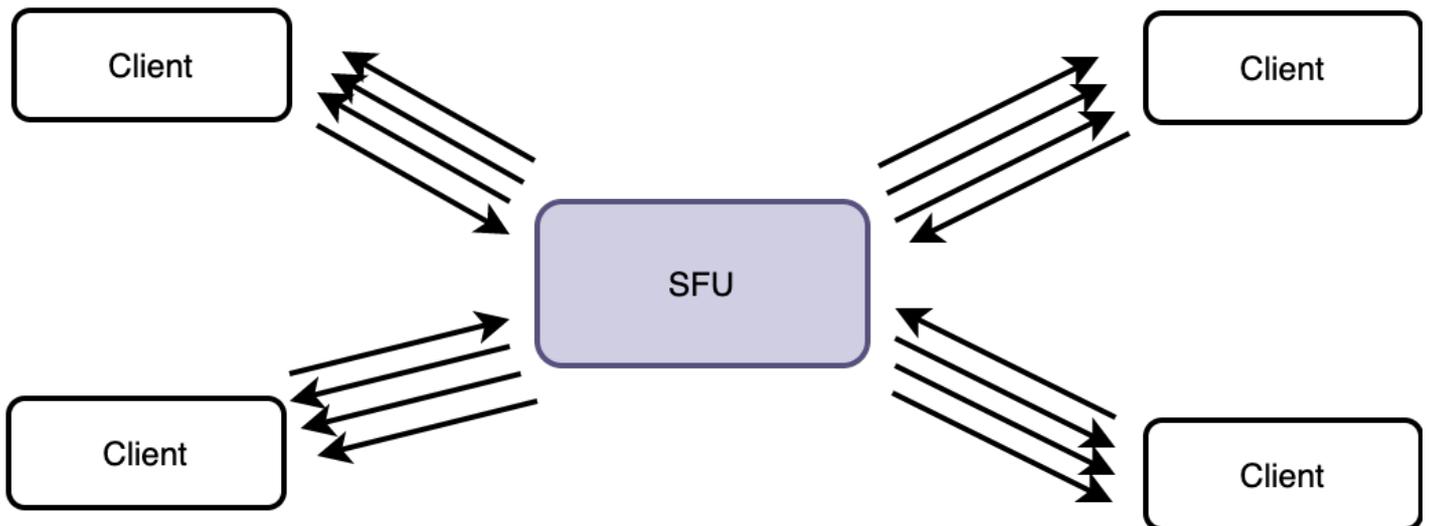


图-7:SFU架构示意图

**优点：**基于流Publish/Subscribe模式灵活，可以非常简单地实现各种多人互动场景，也可以节省Client端的带宽和上行能力，比较容易实现旁路监控和风险管控。

**缺点：**单SFU应对网络传输的分区性差，就近接入需要引入Proxy。单SFU服务可用性差。对于跨地区长距离很难保证实时通信质量，分发效率和质量并与后来的CDN有差距。

## 云SFU阶段

智能手机的诞生让互联网从PC时代迈向了移动时代，终端由计算能力强的PC开始向计算能力弱的手机转移，终端的网络瞬间由原来的有线接入转向了WiFi和3G/4G。由于终端的变迁和用户的激增，为了解决服务后端扩展性、稳定性和运维效率问题，在原来分布式系统基础上衍生出来了云计算。为了消除终端与服务的连通性问题，机房服务端的网络开始大量部署BGP。这个巨大的改变使RTC应用场景发生了转变。这个阶段有三个特点：

1)终端计算能力削弱，网络接入向不稳定的无线（WiFi/3G/4G）接入迁移。

2)后端计算能力增强，网络消除了分区连通问题。

3)联网用户激增，RTC原来的社交、游戏、娱乐直播等领域进一步渗透，RTC扩展到教育、电商和金融。

## 云时代方向

正因如此，RTC在2014年出现了对外提供RTC云计算PaaS服务商网易云信和声网，而后toB的PaaS服务商如雨后春笋，RTC领域正式进入指标内卷时代。RTC PaaS开始阶段是部署在云IaaS上的SFU，直接利用了IaaS基础设施能力，简化了SFU架构，这个架构简称为云SFU。

在线教育开始萌芽，RTC除了音视频通信以外还产生了富媒体的实时通信需求：课件同步与书写同步。富媒体同步需求让RTC产生了实时可靠传输数据能力，在此基础上演化出来实时互动白板。

娱乐直播领域视频分辨率整体提高，甚至出现720P/1080P的实时视频直播，此时CDN接管了音视频分发，RTC主要应用在直播连麦和推流测，在这个场景当中RTC产生了诸多新技术。

企业办公移动化，视频会议产品开始移动化，标志性产品Zoom。

除了以上几个典型RTC方向外，还有更多其他领域的应用方向，这里也就不细说了。总之云时代旺盛需求让RTC迎来了一次巨大的技术迭代。

## 云时代的RTC技术

### 终端传输算法的升级

视频的分辨率和码率爆炸性的增加，而终端接入从原来的有线接入转变为无线接入方式，网络质量受周围环境的影响加大，而且时刻处于高度变化当中，那么在一个高度变化的网络上提供稳定的实时通信的需求就变得异常强烈。在这个阶段，各大RTC PaaS使出浑身解数，各展绝技，不断提高技术指标和效果，toC产品的技术部门紧跟其后，所以我们在LiveVideoStackCon技术大会上总能听到这方面的议题。这些传输算法主要包括以下几个方面：

#### 1.拥塞控制

WebRTC在2015年左右引入基于卡尔曼滤波的GCC拥塞控制算法，经过不断的试验调整，在2018年左右改为发送端计算的TCC。而后WebRTC又引入了Google的BBR和PCC，后续测试当中BBR在视频传输不理想又删除了，PCC还在测试当中。国内的厂商后续跟进，都在拥塞控制方面做了很多的微创新。

#### 2.FEC、ARQ与NACK

**FEC：**WebRTC在第一次推出时带有最简单的Xor的FEC，后续加入了ulpFEC和FlexFEC。国内云厂商创造性地用RS纠删码实现了新的FEC算法，实验室环境抗丢包可以达到80%，真实环境效果有多少就不得而知了。

**ARQ/NACK重传：**重传逻辑在早期的WebRTC版本中没有实现，2015年后开始加入其中，NACK会拉大通信的延迟，所以在链路RTT延迟较小时效果比较明显。

#### 3.Simulcast与SVC

由于终端网络高度动态变化，各个端的网络能力可能有差异，RTC厂商把视频会议中的Simulcast大小流技术引入到RTC作为标准能力，解决SFU到终端之间网络状态适配问题。但Simulcast会增加上传的带宽和视频编码压力。除了Simulcast，有些厂商也引入SVC来解决此类问题。

#### 4.语音带内冗余编码技术

声音在移动弱网下比视频敏感，有些业务声音传输质量要求高于视频，例如在线教育。这就带来了一个媒体优先保证的需求，所以带内语音编码技术成为RTC PaaS厂商的标配。

## 5. QUIC与RUDP

RUDP在2005年左右就有相对应的开源项目，因为NAT穿越的关系，当时最主要的应用是用来做点对点文件传输。2013年，Google开始研发可靠UDP技术，并将这个项目命名为**QUIC**，旨在应用于HTTP/2.0的多路复用优化上。QUIC从提出到成为标准道阻且长。不过QUIC率先在直播推流端解决推流延迟和抗弱网，因为比传统TCP效果好，一时间QUIC在推流端大量应用。

由于实时富媒体通信的场景需求，RTC在原有的实时通信链路上实现了一个基于可靠UDP的数据传输通道来解决实时富媒体交互问题，这类技术和QUIC不谋而合，大有合并之势。

### 媒体处理AI化

由于直播时代大发展和深度学习技术突破，前期简单的滤镜美颜功能已经不能满足需求了，各大厂商从美图秀秀这类照片应用中获得灵感，将其算法应用到实时视频当中，开启了全民美颜直播模式。

除了美颜滤镜AI化外，媒体算法还有：超分显、AI编码、虚拟渲染、头像替换、背景抠图、变声、AI回声消除、AI降噪等等，一大票优秀的算法正在路上。

### SFU Fullmesh

单SFU架构有用户接入和可用性方面的问题，云SFU用BGP接入解决用户接入问题。将SFU做了平行扩展（见图-8），同一个频道用户可以接入到多个对等的SFU进行实时音视频通信，很好地消除了服务异常造成不可用的缺陷，这种架构还能跨IDC部署，实现7x24小时的云服务。

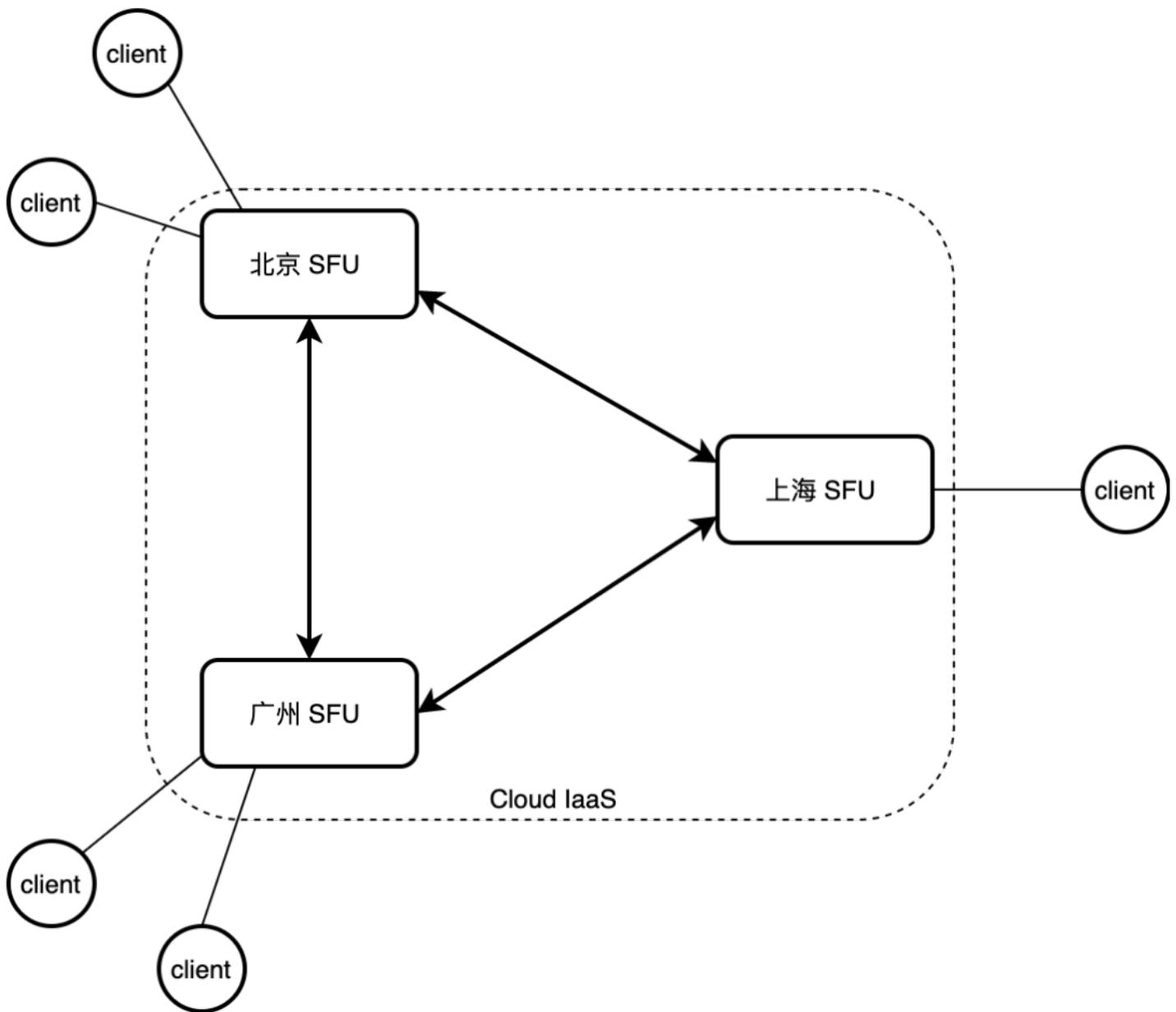


图-8: SFU fullmesh示意图

### 旁路服务

由于直播、教育等业务都是需要审核监控和录制相关的需求，RTC后端除了SFU Fullmesh变化外，还引入了旁路网关单元，旁路网关功能繁多，包括：实时录制、AI内容审核、合流CDN推流、其他协议接入（小程序网关/PSTN）等。

### RTC QoE

RTC PaaS厂商为了数值化媒体通信质量和体验，提出了基于用户体验的QoE指标，将音视频的卡顿次数和时长、连接时长和成功率、码率波动、首屏时间、链路异常恢复时长等数据QoE化，用QoE来保证RTC技术迭代的质量。

### 架构小结

在移动互联网大发展时期，终端设备和接入网络发生了很大变化，这迫使RTC在终端传输技术和媒体处理算法上做了大的革新，并催生了云计算的大规模落地。云计算大规模部署BGP网络，提供便捷的IaaS能力，促使了RTC PaaS厂商的兴起，云SFU Fullmesh架构与直播CDN、AI等技术融合，如下图：

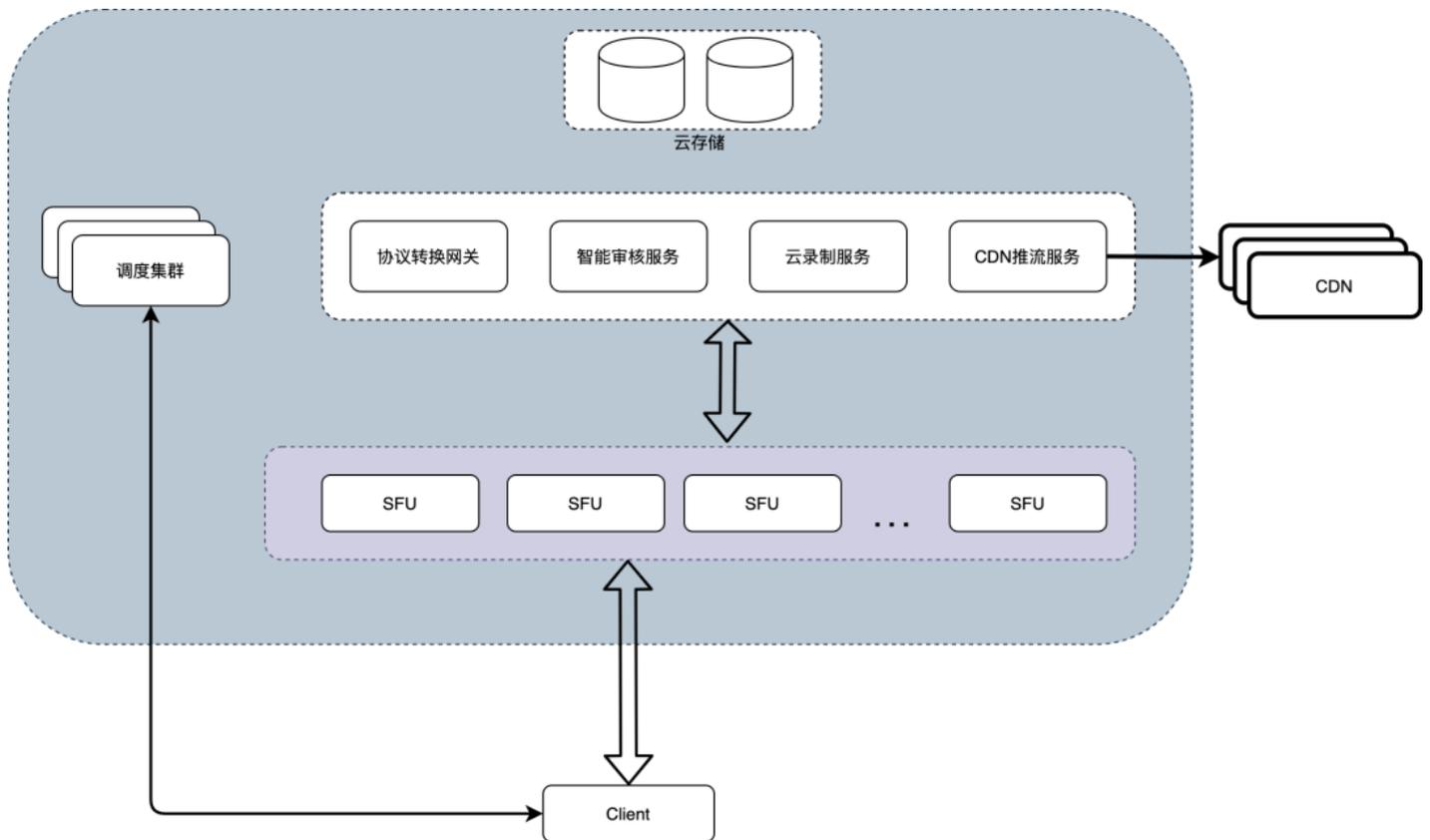


图-9:RTC云SFU示意图

**优点：**充分利用了云计算基础设施的稳定来解决网络接入异构性问题，提高运维效率，优化客户端的传输算法来提高通信的稳定性和质量，保证7x24服务。

**缺点：**在大规模RTC领域BGP带宽成本过高，无法像CDN利用边缘节点带宽来提升传输效率和降低成本；在跨国超长距离RTC通信当中虽然可以利用分布式SFU架构解决就近接入，但很难降低SFU之间的传输延迟，而且架构过分依赖公有云导致可扩展性和灵活性不够。

#### 全球实时云通信阶段

教育大规模线上化和中国音视频产品出海预示着全球实时通信阶段的到来，2020年开始全球疫情加速了RTC音视频通信在各行业的落地，一方面在线课堂、在线办公、企业协同、直播带货等出现了大量RTC应用场景，另外一方面RTC流量成本居高不下，场景流量暴增和成本居高不下之间的矛盾日益严峻，催生了新一代RTC架构迭代。这个阶段主要是两个问题：

- 1.如何将RTC流量成本降下来？
- 2.如何保证下沉用户和跨国长距离通信的质量？

为了解决上面的问题，声网2016年首先研发出基于SDN架构的SD-RTN技术来构建全球实时通信网络（图-10），利用廉价的边缘节点和独特的智能路由加速技术让全球端到端延迟控制在400ms以内。

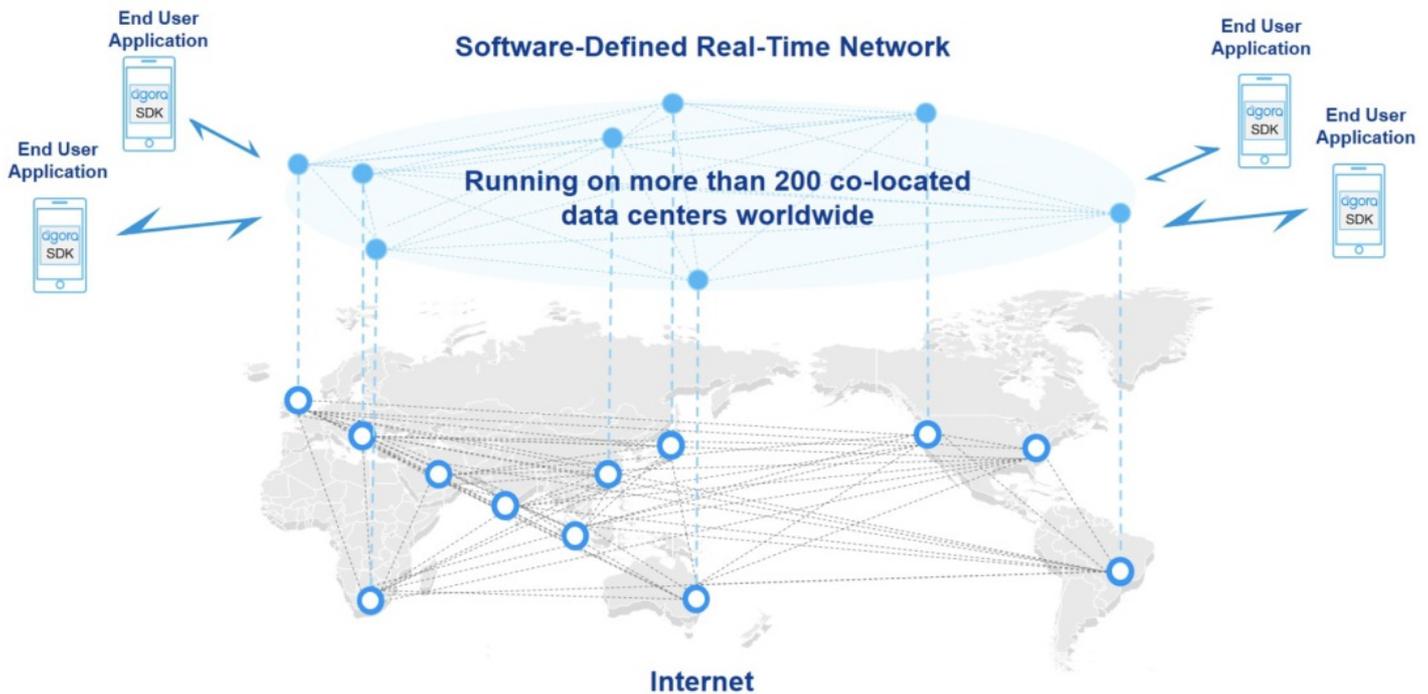


图-10: 全球RTN示意图

从架构上来看，SD-RTN借鉴了早期Skype的P2P覆盖网架构，并开创性地与SDN架构相结合来定义全新的RTN网络，用边缘计算节点代替P2P super node，增强了稳定性，而采用中心计算全局的路由又充分地利用了云计算特点来控制边缘计算节点的传输行为，不仅增加了传输网络的可控性，也保证了分布式节点之间的数据安全。

RTN设计目标是通过横向分层架构来降低系统的耦合度，让音视频RTC系统专注业务逻辑，让网络算法和网络架构工程师发挥自己的长处来满足通信QoS的需要。RTC相关的服务之间相当于内网通信，让服务不需要关心机房和物理位置的限制，整体像是在同一个IDC机房当中，简化开发和部署。

RTN的出现直接导致了终端到Server、Server到Server之间通信的解耦，RTN专注于Server之间的加速，终端与Server之间慢慢演变成了类QUIC的半可靠RUDP协议通信（如图11）。正因为RTN架构能有效解决以上两方面的问题，又符合未来边缘计算趋势，所以各大RTC厂商都纷纷跟进研发自己的RTN组网技术，所以最近几年各种名字的RTN网络都出来了。

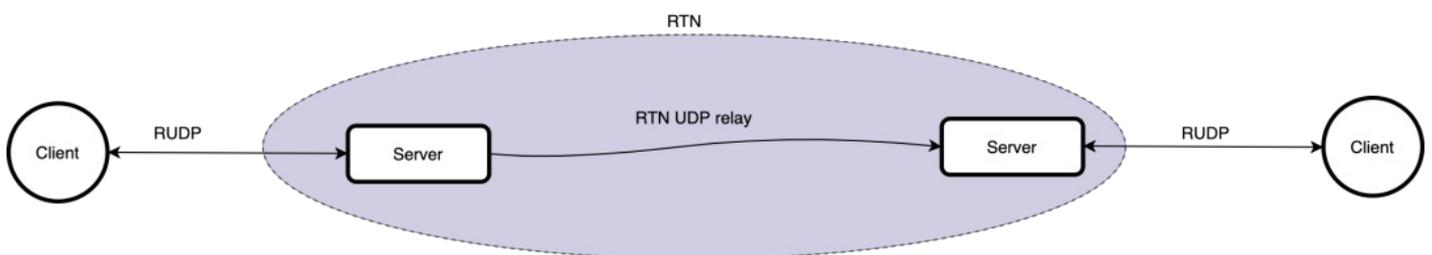


图-11: Client/Server与RTN关系图

## 关键技术

### 边缘计算容器

RTN要实现简化部署和充分利用边缘节点的能力，那么必不可少地需要引入边缘容器技术，通过Docker和K8s实现计算资源的统一管理和调度。

## 智能调度

智能调度有两个方面：终端Lastmile接入调度与系统带宽流量调度。终端Lastmile是调度系统精确将终端调度到最近的边缘RTN节点上进行业务接入，流量调度是根据业务的特点进行资源预留和削峰填谷，让通信成本和质量控制在一个平衡点上。

## 智能路由与多链路备份

RTN网络的边缘节点会实时向中心控制器汇报自己的通信状态，中心控制器收到这些通信状态后会对正常和异常的数据进行处理，并作为计算路由的数据依据。中心控制器得到各个节点数据会通过实时智能最短路径等算法进行路由计算。对于优先级比较高的业务会采用多路径备份策略来保证通信的稳定和质量。

## 虚拟组网技术

RTN边缘计算节点分布在不同的机房，甚至有些横跨洲洋上万公里，这就需要实现一套VPN网络来进行服务之间寻址，简化通信架构复杂度。虚拟组网技术是边缘计算和RTN的关键技术之一。

## 终端接入协议

终端接入协议有类QUIC统一的趋势，目前不管是直播推流还是RTC通信，都在朝着统一化走，快手2017年率先研发出基于可靠UDP的KTP协议来优化客户端接入问题，声网AOT最近也从RTC中独立出来作为单独模块应用在它的FPA产品当中。传统的RTP/RTCP扩展已经不堪重负（见WebRTC源码），类QUIC模型减轻了RTP/RTCP负担，让网络算法工程师摆脱协议束缚专注于传输优化，将会成为未来的趋势。

## 分层解耦

RTN是一个路由网络，从RTC中分离出来。RTC专注于通信业务，RTN专注于通信路由的计算和链路异常failback，那么RTC与RTN需要有一套通用的分层接口来实现分层解耦。

## 架构小结

RTN是RTC超大规模业务场景的底层支撑系统，RTN从架构层利用下沉的边缘计算节点解决路由链路和通信带宽成本问题，让网络算法、RTC、传输路由三者之间解耦。它是当前RTC系统最为负载的系统。

目前RTN还没有统一的标准和做法，各家厂商都是根据自身的业务特点构建自己的RTN系统。在这里我把RTC+RTN分层架构定义为第六代RTC架构。RTN需要庞大的边缘计算节点资源，能进行RTN部署的都是互联网大企业和有一定规模的RTC PaaS厂商。目前看来，RTN可能会让整个RTC行业呈现马太效应。

## 一些总结

综上所述，每一次大的技术浪潮并不是由技术本身推动的，而推动技术浪潮要有两个变化：终端计算带宽能力升级（1、2、4代）和实时通信场景的迁徙（3、5代），这两个变化都会带来旺盛的用户需求。整个RTC技术发展过程中出现过很多技术和架构，大部分技术沿用至今，依然保持生命力，所以说RTC音视频领域技术迭代并不是很快，但每一个技术背后的门槛却很高，技术人员成型时间较长，所以**一般RTC音视频领域没有35岁问题**。

RTC技术领域有其自身的特点，关注用户侧感受和诉求是从事这方面技术人员很容易忽视的。例如：流媒体在用户侧的感受并不敏感，技术上HEVC/AV1比AVC提高多少倍压缩效率，用户侧感受到的可能是手机烫不烫手，耗不耗电。宣传固然重要，但技术不应该忽略用户感受去谈先进性。

技术迭代不是一个数字比武过程，不是谁的数字指标高就会成为主流技术的，技术迭代过程是一个趋同效应，能契合某一类大规模应用场景往往会成为主流或者标准，作为从业人员不应该死盯技术指标上，用更高的技术指标去打败行业先行者是非常困难的，所以在固有领域里面盲目的技术精进也是一种故步自封，后来者应该尽力找到技术更广阔的应用场景形成新趋势。

后疫情时代RTC成为内卷严重的领域，一方面终端能力没有升级，另一方面疫情期间带来的应用场景流量出现了消退的迹象，巨头横行，而新场景还没有出现。但高分辨率、实时虚拟现实等高码率应用刚刚萌芽，超大码率会让UDP协议给kernel带来的负担越来越大，高带宽与低延迟、大并发的矛盾将会在新的场景更加尖锐，新一代的RTC架构有可能会出现TCP/UDP孪生模式。作为技术人的我依然对未来抱有希望，现在最期待的是第六代RTC架构什么时候到来，想必那时定是一个更加波澜壮阔的时代。

扫描图中二维码或点击阅读原文

了解大会更多信息



2021 北京

NEW 新技术 新机会  
TECHNOLOGY OPPORTUNITY

 <p><b>肖潇</b> 即构科技 RTC后台技术总监</p>	 <p><b>刘岩</b> 陌陌 直播媒体中台运维负责人</p>	 <p><b>张建磊</b> Opera 技术副总监</p>	 <p><b>周世付</b> 声网Agora 深度学习算法工程师</p>	 <p><b>吉奇</b> 网易云信 服务端首席架构师</p>
---	---	---	--	--

2021年10月29-30日  
北京丽亭华苑酒店



了解更多演讲内容  
与大会日程相关信息

喜欢我们的内容就点个“在看”吧！

