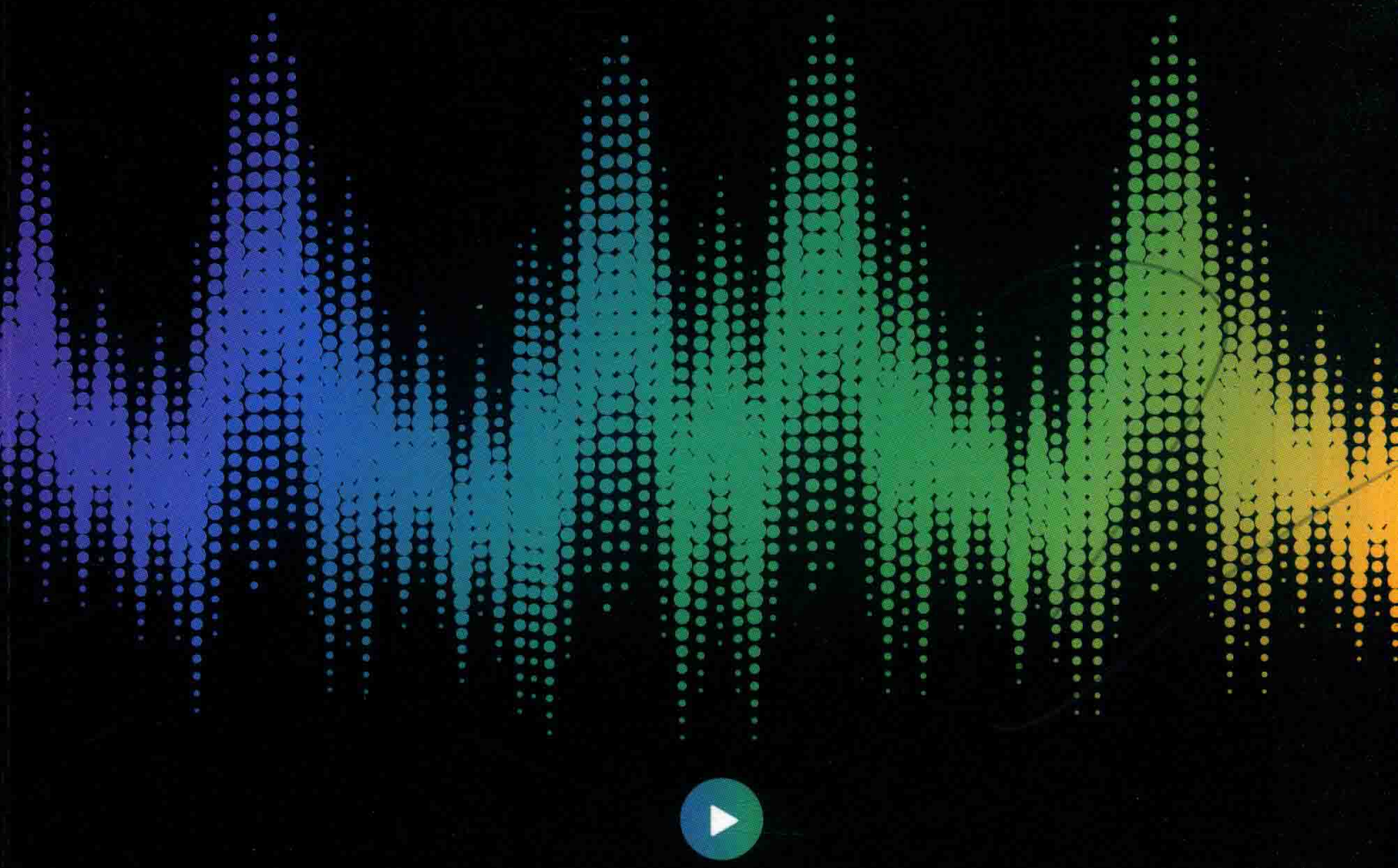
HZ BOOKS
华章科技

资深音视频技术专家撰写，详解 WebRTC 规范、API、信令系统、
底层技术、移动端和服务端实现，集大成之作



WEBRTC ESSENTIALS

Develop Video Conference From Scratch

WebRTC 技术详解

从 0 到 1 构建多人视频会议系统

栗伟 / 著

开源可商用视频会议系统，书中示例可直接应用于视频会议、
在线教育等实时音视频场景

机械工业出版社
China Machine Press

WebRTC 技术详解

从 0 到 1 构建多人视频会议系统

栗伟 / 著

WEBRTC ESSENTIALS

Develop Video Conference From Scratch

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

WebRTC 技术详解：从 0 到 1 构建多人视频会议系统 / 栗伟著 . -- 北京：机械工业出版社，2021.5
ISBN 978-7-111-67844-1

I. ① W… II. ①栗… III. ①移动终端 - 应用程序 - 程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字（2021）第 055562 号

WebRTC 技术详解：从 0 到 1 构建多人视频会议系统

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）

责任编辑：韩　蕊　　责任校对：殷　虹

印　　刷：北京诚信伟业印刷有限公司　　版　　次：2021 年 5 月第 1 版第 1 次印刷

开　　本：186mm × 240mm　1/16　　印　　张：20

书　　号：ISBN 978-7-111-67844-1　　定　　价：99.00 元

客服电话：（010）88361066　88379833　68326294　　投稿热线：（010）88379604

华章网站：www.hzbook.com　　读者信箱：hzit@hzbook.com

内容介绍

这是一本全面、详细讲解WebRTC技术以及如何使用它构建一个可商用的视频会议系统的著作。

在技术维度，本书不仅详细讲解了WebRTC规范和全部API、信令系统、底层技术、移动端、服务端实现，还总结了作者多年来的“踩坑”经验；在实战维度，本书不仅提供了可直接应用于视频会议和在线教育等场景的真实案例，还开源了一个可商用的视频会议系统WiLearning，教读者如何从0到1搭建一个高并发、易扩展的视频会议系统。

全书一共10章

第1章 介绍了WebRTC的历史、技术架构、网络拓扑、兼容性等内容。

第2章 讲解了使用WebRTC API获取本地摄像头、话筒、桌面等媒体流的方法以及媒体流的录制、使用canvas操作媒体流的方法和示例。

第3章 讲解了WebRTC底层使用的传输技术，如ICE、SDP、STUN/TURN等。

第4章 介绍了使用RTCPeerConnection管理WebRTC连接的方法。

第5章 介绍了WebRTC的媒体管理方法，结合示例演示了切换编码格式、控制视频码率、替换视频背景的方法。

第6章 结合示例介绍了一种高效、实时的信令系统实现方法，并实现了一个可以在生产环境中使用的信令系统。

第7章 介绍了使用WebRTC数据通道传输任意数据的方法，结合示例演示了基于P2P的文本聊天以及文件传输功能的实现。

第8章 介绍了使用WebRTC获取媒体流相关统计数据的方法，在示例中演示如何使用Chart.js绘图展示实时码率。

第9章 介绍了在Android、iOS开发环境中使用WebRTC的方法，通过实例实现了基于WebRTC的视频聊天App。

第10章 结合笔者的开源项目WiLearning介绍了从0到1打造高性能视频会议系统的方法。

作者介绍

栗伟

资深音视频技术专家，在该领域有多年实践经验，对WebRTC有非常深入的研究。

曾任职于中科院计算所、CC视频，任职期间利用WebRTC技术开发了直播、在线课堂、视频会议等商业产品，并发用户数达到500万。

开源社区的积极参与者，在GitHub上开源了视频会议项目WiLearning，免费供中小企业使用。

封面设计 · 陈潇然

Foreword 序

栗伟跟我颇有渊源，他在上一家公司带领团队开发的直播产品，被我司采购使用了多年，但我们一直没有直接接触过。几年之后机缘巧合，我们成了同事，也让我对栗伟有了更多的了解。

音视频处理一直是门槛比较高的领域，实时音视频技术尤其如此，栗伟在这个领域深耕多年，打造了音视频方面优秀的商业产品，可谓经验丰富。如今，他把沉淀多年的经验编写成书，同时把自己的项目开源，积极回馈社区，让我十分钦佩。开源社区正是有了千千万万个这样的贡献者，才如此生机勃勃。作为一个享受开源达 20 年之久的互联网技术人，我本人也获益匪浅。希望能有更多人参与到开源社区，希望更多人从本书以及本书介绍的开源项目中获益。

祝本书能够大卖！

正保集团副总裁　林杨

2021 年 2 月

前 言 Preface

为什么要写这本书

最早接触 WebRTC 技术是在 2015 年，那时需要在直播产品中增加实时连麦的功能，经过对几种技术进行对比，最终我选择了 WebRTC。当时 WebRTC 技术还不够成熟，相关资料非常少，在产品中使用 WebRTC 技术的难度非常大，往往为了弄清楚某个概念、某个 API 的用法，需要查阅大量的英文资料，而且遇到问题解决起来非常棘手。

从最初的原生 WebRTC，到多点控制单元（MCU），再到各种选择性转发单元（SFU），我在使用 WebRTC 的过程中一直不断学习新的知识，不断解决新的问题，同时也逐步加深了对 WebRTC 技术的理解和认识。

因为踩过许多坑，所以我深刻体会到了 WebRTC 技术的难度和广度。WebRTC 技术包含了音视频编解码技术、传输技术、流媒体服务器技术等，涵盖了音视频处理和传输的方方面面。这些技术中任意一个都能成为独立的课题，都值得花大量时间深入研究。除此之外，理解 WebRTC 相关 API，还必须掌握现代 Web 技术，尤其是 ES6、Promise 等语法知识。可见，学习 WebRTC 技术需要掌握大量的预备知识，这对于初学者来说有一定的门槛。

非常遗憾的是，时至今日仍没有一本中文书能够系统地涵盖 WebRTC 的技术内容，剥离层层技术面纱将 WebRTC 呈现给国内技术人员。

在实时通信产品大爆发的时期，为什么 WebRTC 的中文技术资料如此之少？我想可能有以下几个原因。

- WebRTC 技术规范都是英文文档，缺少使用示例，故而读起来晦涩难懂，加大了 WebRTC 的学习难度。
- WebRTC 技术较新，专业性较强，能真正理解并掌握其精髓的技术人员较少。
- 国内技术人员工作压力大，资深 WebRTC 技术人员忙于项目，没有时间总结经验并分享。
- WebRTC 技术覆盖面广，难以讲深、讲透，针对某个技术点的分享容易实现，但要系统讲解技术内幕则非常难。

撰写一本能够降低国内技术人员使用 WebRTC 的门槛，能够帮助研发人员更好地将 WebRTC 技术应用到产品中的书，是我编写本书的出发点。

作为一名较早使用 WebRTC 的技术人员，我一直关注 WebRTC 技术的发展，在日常使用过程中积累了大量学习笔记和经验，这些都为撰写本书提供了素材。

本书对 WebRTC 1.0 规范的内容进行了系统整理，以一种易于理解的形式呈现给读者。书中还给出了我的“踩坑”经验和一些实用的案例，帮助读者全面认识 WebRTC。

WebRTC 降低了实时通信技术的门槛，使得之前只有互联网巨头才能掌握的实时通信技术得以普及，使得我们能够在家远程办公，孩子们能够“停课不停学”。相信在 5G 普及之后，WebRTC 还会迎来更加蓬勃的发展。

可以预见，未来将有更多技术人员学习并应用 WebRTC，希望本书能够帮助大家轻松踏入 WebRTC 的技术殿堂！

读者对象

实时通信产品的售前、售后、研发人员，音视频行业的架构师、CTO 等。

本书特色

- 全面涵盖 WebRTC 1.0 规范。
- 详细讲解 WebRTC 底层技术。
- 结合示例演示 WebRTC API 的使用。
- 从零起步实现高效、实时的信令系统。
- 使用 WebRTC 技术从 0 到 1 打造开源视频会议系统。

如何阅读本书

本书对 WebRTC 技术进行了全面的介绍，涵盖 WebRTC 1.0 规范全部 API、WebRTC 底层技术、WebRTC 在移动端和服务器端的应用等内容，并提供了具体的示例，力求做到理论结合实践。本书最后使用这些 WebRTC 知识打造了一个真实的视频会议系统，同时对高并发、易扩展的视频会议架构进行了详细讲解。

本书分为 10 章。

第 1 章介绍 WebRTC 的历史、技术架构、兼容性等内容。

第 2 章介绍使用 WebRTC API 获取本地摄像头、话筒、桌面等媒体流的方法，以及媒体流的录制、使用 canvas 操作媒体流的方法和示例。

第 3 章介绍 WebRTC 底层使用的传输技术，如 SDP、ICE、STUN/TURN 等。

第 4 章介绍使用 RTCPeerConnection 管理 WebRTC 连接的方法。

第 5 章介绍 WebRTC 的媒体管理方法，结合示例演示切换编码格式、控制视频码率、替换视频背景的方法。

第 6 章结合示例介绍一种高效、实时的信令系统实现方法，并实现一个可以在生产环境中使用的信令系统。

第 7 章介绍使用 WebRTC 数据通道传输任意数据的方法，结合示例演示基于 P2P 的文字聊天以及文件传输功能的实现。

第 8 章介绍使用 WebRTC 获取媒体流相关统计数据的方法，结合示例演示如何使用 Chart.js 绘图展示实时码率。

第 9 章介绍在 Android、iOS 开发环境中使用 WebRTC 的方法，并实现基于 WebRTC 的视频聊天 App。

第 10 章结合我的开源项目 WiLearning 介绍从 0 到 1 打造视频会议系统的方法。

本书提供的示例代码以及开源项目 WiLearning 可以在 GitHub 上免费获取，地址为 https://github.com/wistingcn。

致谢

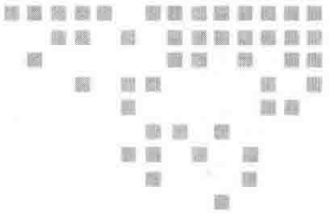
感谢我的家人，他们给我提供了最大的支持。在写书期间，我每天早出晚归，没有一个完整的周末，我的爱人承担起了所有的家务。还有我两个可爱的小天使，每天晚上回到家里，她们都会跑过来喊着："欢迎爸爸回来！"这是我一天中最开心的时刻，所有的疲劳和烦恼都一扫而光。

感谢开源社区贡献了 WebRTC 这样一个优秀的实时音视频框架。正是出于回馈开源社区的愿景，我才投入了大量的精力开发 WiLearning。

感谢机械工业出版社的杨福川和各位编辑为我写书提供了指导，并不辞劳苦地修订、校稿。

谨以此书献给我最亲爱的家人以及众多热爱 WebRTC 技术的朋友们！

Contents 目 录



第 1 章 Chapter 1

WebRTC 概述

随着网络基础设施日趋完善以及终端计算能力不断提升，实时通信技术已经渗透到各行各业，支撑着人们的日常生活。在 WebRTC 诞生之前，实时通信技术非常复杂，想获得核心的音视频编码及传输技术需要支付昂贵的专利授权费用。此外，将实时通信技术与业务结合也非常困难，并且很耗时，通常只有较大规模的公司才有能力实现。

WebRTC 的出现使实时通信技术得以广泛应用。WebRTC 制定、实现了一套统一且完整的实时通信标准，并将这套标准开源。这套标准包含了实时通信技术涉及的所有内容，使用这套标准，开发人员无须关注音视频编解码、网络连接、传输等底层技术细节，可以专注于构建业务逻辑，且这些底层技术是完全免费的。

WebRTC 统一了各平台的实时通信技术，大部分操作系统及浏览器都支持 WebRTC，无须安装任何插件，就可以在浏览器端发起实时视频通话。

WebRTC 技术最初为 Web 打造，随着 WebRTC 自身的演进，目前已经可以将其应用于各种应用程序。

随着 4G 的普及和 5G 技术的应用，实时音视频技术正在蓬勃发展。在互联网领域，花椒、映客等直播平台吸引了大量的用户；在教育领域，通过实时直播技术搭建的“空中课堂”惠及全球数亿学生；在医疗行业，随着电子处方单纳入医保，互联网看病、复诊正在兴起，地域之间医疗资源不均衡的问题被实时直播技术逐步消除。

WebRTC 1.0 规范发布以来，以 Chrome、Firefox 为代表的浏览器对 WebRTC 提供了全方面的支持，Safari 11 也开始对 WebRTC 提供支持。

1.1 WebRTC 的历史

WebRTC（Web Real-Time Communication）是一个谷歌开源项目，它提供了一套标准

API，使 Web 应用可以直接提供实时音视频通信功能，不再需要借助任何插件。原生通信过程采用 P2P 协议，数据直接在浏览器之间交互，理论上不需要服务器端的参与。

“为浏览器、移动平台、物联网设备提供一套用于开发功能丰富、高质量的实时音视频应用的通用协议”是 WebRTC 的使命。

WebRTC 的发展历史如下。

- 2010 年 5 月，谷歌收购视频会议软件公司 GIPS，该公司在 RTC 编码方面有深厚的技术积累。
- 2011 年 5 月，谷歌开源 WebRTC 项目。
- 2011 年 10 月，W3C 发布第一个 WebRTC 规范草案。
- 2014 年 7 月，谷歌发布视频会议产品 Hangouts，该产品使用了 WebRTC 技术。
- 2017 年 11 月，WebRTC 进入候选推荐标准（Candidate Recommendation，CR）阶段。

1.2 WebRTC 的技术架构

从技术实现的角度讲，在浏览器之间进行实时通信需要使用很多技术，如音视频编解码、网络连接管理、媒体数据实时传输等，还需要提供一组易用的 API 给开发者使用。这些技术组合在一起，就是 WebRTC 技术架构，如图 1-1 所示。

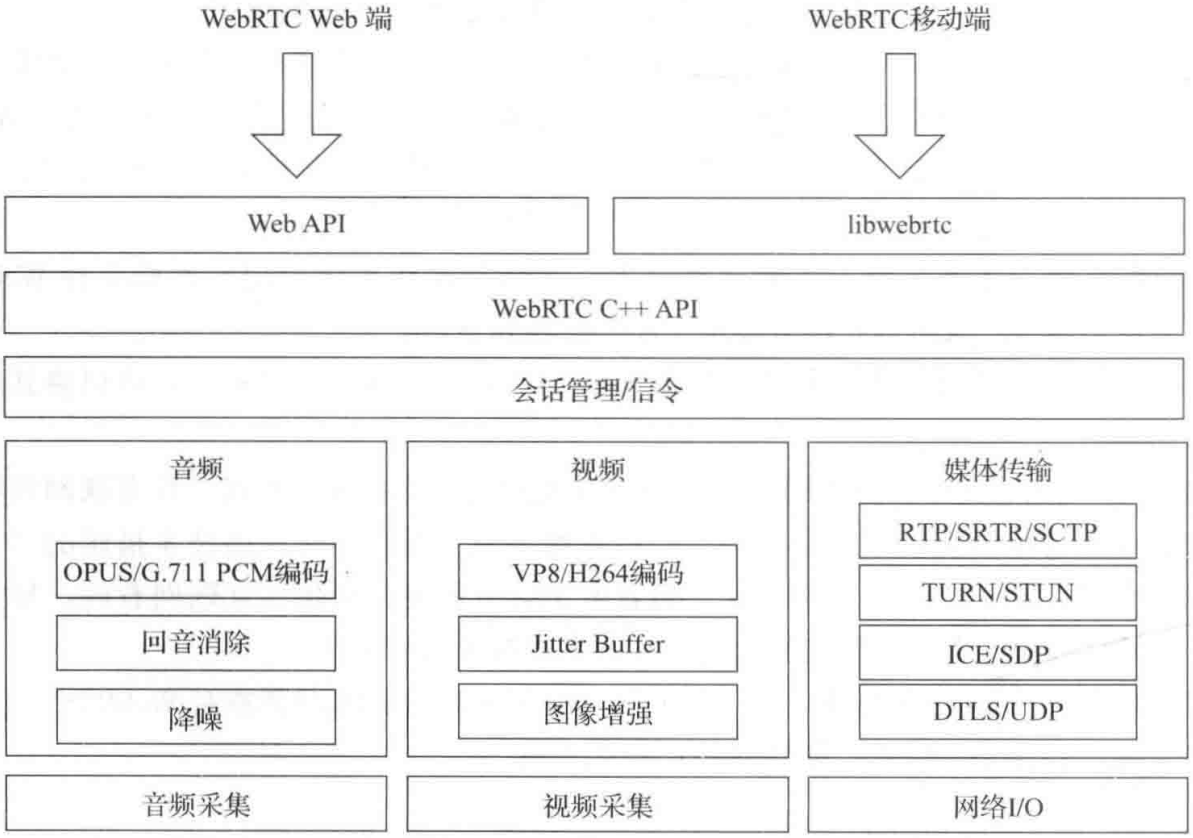


图 1-1 WebRTC 技术架构

WebRTC 技术架构的顶层分为两个部分。一部分是 Web API，一组 JavaScript 接口，由

W3C 维护，开发人员可以使用这些 API 在浏览器中创建实时通信应用程序。另一部分是适用于移动端及桌面开发的 libwebrtc，即使用 WebRTC C++ 源码在 Windows、Android、iOS 等平台编译后的开发包，开发人员可以使用这个开发包打造原生的 WebRTC 应用程序。

第二层是 WebRTC C ++ API，它是 Web API 和 libwebrtc 的底层实现。该层包含了连接管理、连接设置、会话状态和数据传输的 API。基于这些 API，浏览器厂商可以方便地加入对 WebRTC 的支持。

WebRTC 规范里没有包含信令协议，这部分需要研发人员依据业务特点自行实现。

WebRTC 支持的音频编码格式有 OPUS 和 G.711，同时还在音频处理层实现了回音消除及降噪功能。WebRTC 支持的视频编码格式主要有 VP8 和 H264（还有部分浏览器支持 VP9 及 H265 格式），WebRTC 还实现了 Jitter Buffer 防抖动及图像增强等高级功能。

在媒体传输层，WebRTC 在 UDP 之上增加了 3 个协议。

- 数据包传输层安全性协议（DTLS）用于加密媒体数据和应用程序数据。
- 安全实时传输协议（SRTP）用于传输音频和视频流。
- 流控制传输协议（SCTP）用于传输应用程序数据。

WebRTC 借助 ICE 技术在端与端之间建立 P2P 连接，它提供了一系列 API，用于管理连接。WebRTC 还提供了摄像头、话筒、桌面等媒体采集 API，使用这些 API 可以定制媒体流。

我们将在后面的章节详细讨论 WebRTC 架构的主要技术（不包含 C++ 部分），并结合实例展示这些技术的应用。

1.3　WebRTC 的网络拓扑

WebRTC 规范主要介绍了使用 ICE 技术建立 P2P 的网络连接，即 Mesh 网络结构。在 WebRTC 技术的实际应用中，衍生出了媒体服务器的用法。

使用媒体服务器的场景，通常是因为 P2P 连接不可控，而使用媒体服务器可以对媒体流进行修改、分析、记录等 P2P 无法完成的操作。实际上，如果我们把媒体服务器看作 WebRTC 连接的另外一端，就很容易理解媒体服务器的工作原理了。媒体服务器是 WebRTC 在服务器端的实现，起到了桥梁的作用，用于连接多个 WebRTC 客户端，并增加了额外的媒体处理功能。通常根据提供的功能，将媒体服务器区分成 MCU 和 SFU。

1. Mesh 网络结构

Mesh 是 WebRTC 多方会话最简单的网络结构。在这种结构中，每个参与者都向其他所有参与者发送媒体流，同时接收其他所有参与者发送的媒体流。说这是最简单的网络结构，是因为它是 WebRTC 原生支持的，无须媒体服务器的参与。Mesh 网络结构如图 1-2 所示。

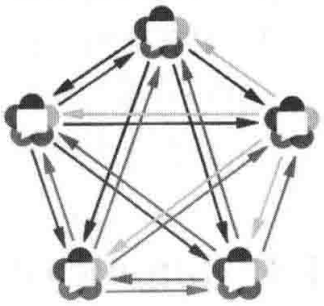


图 1-2　Mesh 网络结构

在 Mesh 网络结构中，每个参与者都以 P2P 的方式相互连接，数据交换基本不经过中央服务器（部分无法使用 P2P 的场景，会经过 TURN 服务器）。由于每个参与者都要为其他参与者提供独立的媒体流，因此需要 $N-1$ 个上行链路和 $N-1$ 个下行链路。众多上行和下行链路限制了参与人数，参与人过多会导致明显卡顿，通常只能支持 6 人以下的实时互动场景。

由于没有媒体服务器的参与，Mesh 网络结构难以对视频做额外的处理，不支持视频录制、视频转码、视频合流等操作。

2. MCU 网络结构

MCU（Multipoint Control Unit）是一种传统的中心化网络结构，参与者仅与中心的 MCU 媒体服务器连接。MCU 媒体服务器合并所有参与者的视频流，生成一个包含所有参与者画面的视频流，参与者只需要拉取合流画面，MCU 网络结构如图 1-3 所示。

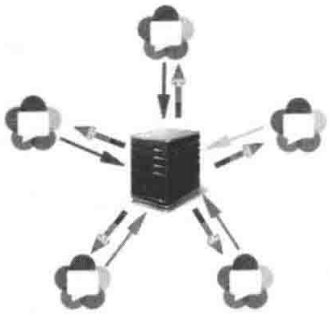


图 1-3 MCU 网络结构

这种场景下，每个参与者只需要 1 个上行链路和 1 个下行链路。与 Mesh 网络结构相比，参与者所在的终端压力要小很多，可以支持更多人同时在线进行音视频通信，比较适合多人实时互动场景。但是 MCU 服务器负责所有视频编码、转码、解码、合流等复杂操作，服务器端压力较大，需要较高的配置。同时由于合流画面固定，界面布局也不够灵活。

3. SFU 网络结构

在 SFU（Selective Forwarding Unit）网络结构中，仍然有中心节点媒体服务器，但是中心节点只负责转发，不做合流、转码等资源开销较大的媒体处理工作，所以服务器的压力会小很多，服务器配置也不像 MCU 的要求那么高。每个参与者需要 1 个上行链路和 $N-1$ 个下行链路，带宽消耗低于 Mesh，但是高于 MCU。

我们可以将 SFU 服务器视为一个 WebRTC 参与方，它与其他所有参与方进行 1 对 1 的建立连接，并在其中起到桥梁的作用，同时转发各个参与者的媒体数据。SFU 服务器具备复制媒体数据的能力，能够将一个参与者的数据转发给多个参与者。SFU 服务器与 TURN 服务器不同，TURN 服务器仅仅是为 WebRTC 客户端提供的一种辅助数据转发通道，在无法使用 P2P 的情况下进行透明的数据转发，TURN 服务器不具备复制、转发媒体数据的能力。

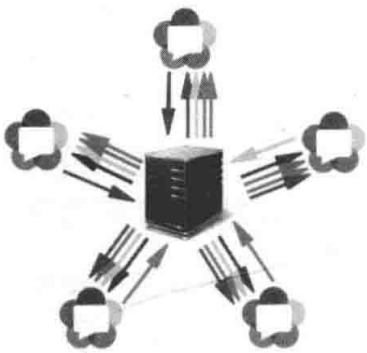


图 1-4 SFU 网络结构

SFU 对参与实时互动的人数也有一定的限制，适用于在线教学、大型会议等场景，其网络结构如图 1-4 所示。

1.4 Simulcast 联播

在进行 WebRTC 多方视频会话时，参与人数较多，硬件设施、网络环境均有差异，这