第5章　P2P流媒体技术

随着移动互联时代发展得越来越快，人们已经不仅满足于现有的媒体资源，而是渴望寻求一种更加快速、便捷的技术手段来获得媒体服务，这也正是当初 P2P 技术由于自身理论上存在的无限的可扩展性被引入流媒体服务的原因。P2P 流媒体技术无论是在商业抑或是专业领域都被广泛使用，本章将围绕 P2P 流媒体系统中的关键技术展开论述。

5.1　P2P 流媒体技术概述

5.1.1　P2P 技术的引入

根据第二十届中国互联网大会发布的《中国互联网发展报告 2021》，截至 2020 年底，中国网民规模为 9.89 亿人，互联网普及率达到 70.4%；特别地，中国互联网网络音视频产业规模达到 2412 亿元，同比增长 44%，网络音视频用户规模持续增长。数字音乐产业市场规模达 732 亿元，同比增长 10%，网络视频活跃用户规模达到 10.01 亿人，网络音频娱乐市场活跃用户规模达到 8.17 亿人，同比分别增长 2.14%和 7.22%。网络音视频市场依托技术升级、用户代际更替、消费模式转换等变量推动，产业效能持续提升，网络直播规模持续增长，技术和商业模式持续拓展。为了使每个用户都能够得到相同的体验，扩大流媒体服务系统规模势在必行。目前主要有以下四种方法。

（1）媒体服务器集群。媒体服务器集群就是集合多个流媒体服务器一同工作，完成用户的大量请求，如图 5.1 所示。虽然这样的方法的确可以增大服务器规模，但是和用户带宽有一定关系。

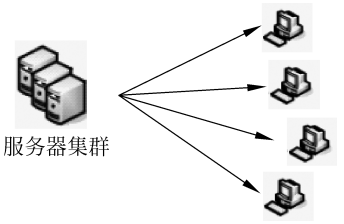

图 5.1　媒体服务器集群

（2）CDN（Content Delivery Network，内容分发网络）方式。如图 5.2 所示，CDN 的模式是在全世界范围内架设许多台服务器，它们共同承担用户的请求，当用户向服务器发起请求时，数据不会从主服务器发出，而是选择离用户最近的服务器发出。相比于媒体服务器集群，CDN 的可扩展性大大增加，但是用户分布情况需要与服务器相近，否则可能造成一些服务器负载过大，而一些服务器无人问津的情况发生，造成资源浪费和用户体验下降，且成本过高难以维护。

（3）IP 组播方式。即一份数据通过路由器复制的方式到达多个用户的客户端，如图 5.3 所示。因为资源服务器只需要发送一次数据，这样可以极大减轻服务器的压力。不过要能够真正落实 IP 组播非常麻烦且繁杂，导致这种情况的原因是路由器在每次工作时都

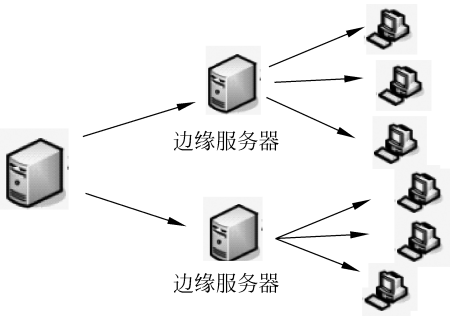

图 5.2　CDN 方式

会录入用户信息，完成之后才能正确转发消息，其次，IP 地址缺少也是不被大众认可的重要因素。

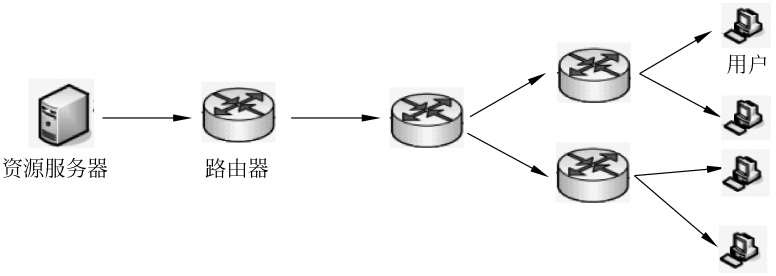

图 5.3　IP 组播方式

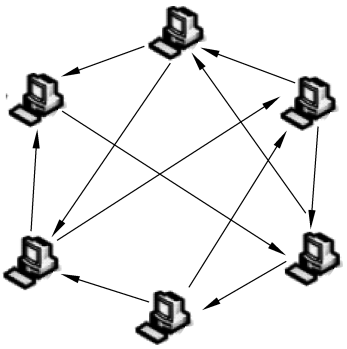

图 5.4　P2P 方式

（4）P2P 方式。P2P 即 Peer to Peer，也称对等网技术，如图 5.4 所示。它的核心思想与传统的 C/S 模式完全不同，数据的传输不需要通过服务器，而是直接进行传输，每一个用户都是一个网络中的节点参与数据传输，所以，可以将 P2P 技术与流媒体视频直播或者点播系统结合起来，有资源一方就可以分发数据，不再单一依靠服务器分发数据，减轻压力。理想的 P2P 模型可以无限制地拓展下去，所以将 P2P 与流媒体结合在一起拥有无限的可能。

虽然在日常应用中，往往会因为网络、节点等各种因素对采用 P2P 方式的系统造成不确定性，使得无限扩展只存在于理论上，但是综合来看，把 P2P 应用在流媒体系统中是最具有潜力的方式。

5.1.2　P2P 技术的发展

目前，国外研究 P2P 最著名的是 IRIS 计划以及英国斯坦福大学的 P2P 研究计划。IRIS 计划是由美国政府支持五所著名大学和研究所共同合作的一项致力于解决网络安全漏洞的存储系统。这个存储系统的基石就是 P2P 网络的支持。但与其他研究 P2P 网络计划不同的是，IRIS 计划主要研究的是 P2P 网络中的资源定位服务，通过研究 DHT 在 P2P

中的应用，开发出一款既可以共享文件又可以供给其他应用服务的平台软件，创造一个网络安全的分布式 P2P 网络。另一项则是斯坦福大学的研究，主要针对 P2P 资源搜索和与 IRIS 同样的构造网络安全环境。对于 P2P 资源搜索的研究集中在通过设定一定的搜索方式来使 P2P 网络能够更加有效、准确地展现出用户所想得到的结果。对于构建安全的 P2P 网络来说，研究侧重于网络的稳定、高效以及可靠。微软公司针对研发 P2P 技术专门设立了工作组，同样，在 2000 年 P2P 技术的早期，英特尔公司就已经宣布成立工作组专门致力于研究 P2P 项目。国内的 P2P 研究主要集中在知名的大学研究机构，如北京大学研究开发的 Maze 系统，当在个人的 PC 端安装 Maze 客户端后就可以加入 Maze 网络中，通过共享发送或接收来自好友的文件。华中科技大学自主设计的 anysee 直播软件也同样备受瞩目。anysee 不仅支持一对多进行传输，同时还支持防火墙的功能，使得 P2P 系统能够得到有效的扩展。还有诸如清华大学的 Granary 等都是 P2P 应用研究的领头羊。

国外相对于 P2P 和流媒体的研究起步较早。如果能同时提供给大量用户高画质、高帧率、高可靠性的流媒体服务，那么无疑这款应用是拥有极大前景的。但 20 世纪的传统 C/S 结构恰恰与之相反，由于服务器的瓶颈导致用户体验差、视频质量低等种种问题。1998 年正是因特网萌芽阶段，当时一个学生编写了一个搜索歌曲应用程序，程序会把搜索的音乐放入集中服务器，从而为用户过滤筛选，这个程序就是著名的 Napster，在最高峰时拥有 8000 万用户。Napster 由一个目录服务器和若干个客户端形成星状结构，用户提出需要的 MP3 后，客户端开始连接目录服务器，之后目录服务器会开始检索所需资源，然后服务器将检索后得出的结果反馈给客户端，用户再与音乐的作者连接，传输 MP3。这样大大降低服务器负担且使得用户获得广泛资源。Napster 使用的 P2P 技术开始进入流媒体研究者的视线。1998 年，一位美国的研究者提出了一种称作 Webcast 的流媒体播放系统，虽然这款基于 P2P 技术的流媒体应用采用最简单的树状结构进行传输数据，但是人们看到了 P2P 技术应用于流媒体服务中强大的可扩展性。到这里为止，P2P 流媒体正式开始被人们研究发展。

2000 年，P2P 流媒体技术得到了快速发展。2000 年出现的 ESM（End System Multicast）是第一个 P2P 流媒体直播系统，系统采用节点互相连接形成一个组播网，在节点之间传播数据，ESM 标志着 P2P 流媒体正式发展。之后出现许多 P2P 流媒体系统，如 Peercast，P2PRadin，Overcast，Coopnet，SplitStream 等，以及应用最广泛的 Goosip 协议。节点在传输数据时，首先把数据发送给邻居节点，再通过邻居节点进一步传输出去，以此类推。另外，还有一些研究机构将 P2P 技术与传统的流媒体技术结合起来，如美国奥利根大学研究的 PALS，利用分层编码技术，让网络中的不同节点将各个层次的媒体编码流传送到接收者，然后接收者根据自身的带宽资源等条件接收部分或者全部的编码流。

国内 P2P 流媒体技术由于早期进入时间不长所以发展缓慢，但是由于我国互联网技术不断的发展和对 P2P 的重视，P2P 正在迅速发展。2004 年，一款名为 CoolStreaming 的直播系统软件被香港科技大学研究出来，这款基于 Gossip 协议的应用在欧洲杯期间试行成功。CoolStreaming 系统所采用的 Gossip 协议具有十分高的可扩展性和稳定性，且支持多对多传输，使得用户的用户体验大大上升。也正是因为如此，当时得到了许多专家和用户的称赞。因为 CoolStreaming 系统是第一个成功问世且广泛应用的 P2P 直播系统，所以人们把它称为 P2P 直播的第三次革命。到此为止，P2P 直播逐渐开始商业化。

在商业上也出现了许多 P2P 流媒体产品，如 PPLive、PPStream、UUSee 和沸点等。以 PPLive 为例，它采用的网络拓扑模型有效地提高了网络视频点播服务的质量，解决了服务器端带宽和负载有限的问题。同样颇具影响力的 PPStream，其最大优势在于几乎所有新加入的用户都可以在 1min 内播放所需视频数据，且实现了用户越多播放流畅性越高的特性。

PPLive 作为我国商业化形成的 P2P 流媒体视频直播与点播软件，被人们广泛认可。PPLive 系统中有多个 tracker 服务器和资源服务器，tracker 服务器之间负载平衡通过 DHT 算法来实现，资源服务器向用户提供最初的流媒体资源。当用户欣赏资源时，客户端就会把接收到的数据包根据相关的策略进行缓存，与此同时，用户节点定期向 tracker 服务器报告自己的缓存等相关状态信息，其他的节点就可以在服务器的辅助下找到需要的数据进行播放。随着 P2P 技术的不断发展更新，PPLive 也不断更新着服务器质量和用户体验，正是由于 P2P 网络的特点，只要能够保持一定量增长的用户，PPLive 服务器质量也会更加稳定和高效。由于点播系统相对于直播系统对于用户来说拥有更强的交互性和体验感，因此越来越多的专家利用 P2P 流媒体技术研究点播系统，如 PPStream 也开始提供点播服务。

5.1.3 P2P 技术的基本概念

目前 P2P 流媒体技术已经受到越来越多的关注，无论是学术或者商界，它都能对客户端的闲置资源进行最大限度的充分利用，大大缓解因用户过多而带来的网络带宽压力，因其所具备强大的可扩展性，也为人们在日常生活中解决流媒体内容分发问题提供了一个新的方向。

通过前面的内容我们知道，随着计算机的发展和人们日常生活需求的不断增长，流媒体(Streaming Media)技术应运而生，从“流媒体”这三个字上不难看出，所谓的流媒体就是流式媒体，流式媒体就是一种采用流式传输的方式在网络上传输多媒体数据。流媒体采用的是流式传输，即边下载边播放，不用完整的文件即可播放，因此它与传统媒体的区别主要体现在：①启动所需时间极大地缩短；②对缓存容量的上限大大降低；③流媒体需要特定的传输协议进行传输；④传输数据信息量比传统媒体大。

点对点技术(Peer-to-Peer，P2P)，也称为对等网技术，是一种分布式的应用体系架构，它的任务是对于分配到网络的各个节点，所有节点拥有相同的功能，在程序中等效应用，不存在主从之分，节点与节点之间互连形成了对等网络。当前，很多国内外研究人员都把 P2P 技术视作研究的重点项目。P2P 技术在共享、直播、点播、教育、金融等行业都已经取得了极大的发展和成果。它被《财富》杂志评为当今影响互联网发展的四大科技之一，回归了互联网最初设计的本质——网络中的节点可以互相通信，且不需要网络中其他节点参与，所以 P2P 技术更重要的是其设计思想。在 P2P 网络中，参与到网络中的用户共享它们拥有的一部分硬件资源(如处理能力、存储空间、网络带宽等)。在这个共享网络中，节点共享的资源和内容可以被其他节点使用，且不需要其他中间节点提供支持。网络中的节点既是资源的提供者，也是资源的利用者。

传统的基于 C/S(Client/Server)架构的网络，是以服务器为中心，所有客户端的资源均由服务器进行提供，客户端之间并没有关联通信。当客户端的规模数量不断增加时，受限于服务器承载能力的上限，服务器难以提供给每一个客户端高额的资源，成为整个网络架构的局限。此外，每一个客户端的请求与命令都需要服务器进行分析再给出应答，因此特别容易出现拥塞，不仅影响网络的效率，而且一旦服务器瘫痪，所有客户端也会随之一起瘫痪。

P2P 与传统 C/S 完全不同，图 5.5 与图 5.6 给出了两种架构的对比图。可以发现 P2P 网络结构与传统集中式的 C/S 网络结构截然相反，它是一种非中心化的结构，节点与节点之间相互联系，不用通过中间节点进行过渡，完美解决了 C/S 架构的缺点，因为没有中心服务器，所以网络中每个的节点既是服务器，又是客户端。

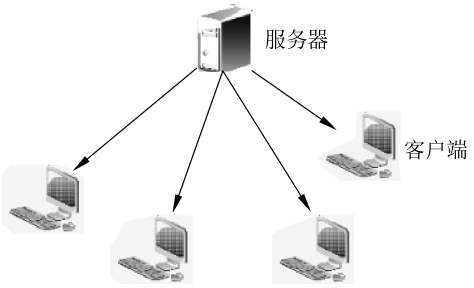

图 5.5　C/S 网络结构

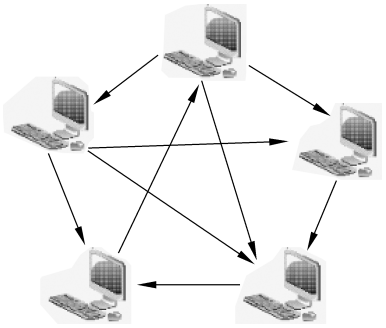

图 5.6　P2P 网络结构

通过对比 C/S 网络架构与 P2P 网络架构，不难看出 P2P 网络的特点。

（1）结构非中心化。传统 C/S 网络往往由于客户端过多而导致服务瓶颈，P2P 网络中的资源和服务在节点上进行传输和实现，又无需其他节点的帮助，使得 P2P 网络避免了 C/S 网络的弊端。

（2）可扩展性。随着源源不断的客户端加入 P2P 网络，节点数量越来越多，每个客户端可利用及传输的资源点也越来越多，拥有无限可能的扩展性。

（3）容错率高。由于日常网络中往往会出现网络延迟、拥塞、节点失联问题，这些问题都会影响整个网络的稳定和安全，在传统 C/S 网络中，一旦服务器出现问题，整个网络也会出错，而 P2P 网络中的节点出错时最多影响一部分，其余大部分节点仍然可以正常工作，容错率高。

（4）高性价比。随着网络硬件、存储空间快速发展成熟，若人们在空余时能把闲置的资源部分利用起来，这样的资源利用是不可估量的，P2P 网络能将这一功能实现，投入极少的资源，收取大量的性能。

综上所述，与传统的 C/S 结构网络相比，由于 P2P 网络中并没有绝对意义上的主机，每个节点既是主机也是客户端，因此彼此间可以直接通信、交互和共享信息。一个分布式的 P2P 网络可以提供传统网络不能实现的特性，而使真正的分布式计算成为可能。

5.1.4　P2P 流媒体的直播与点播系统

P2P 流媒体服务系统根据播放类型的不同，分为 P2P 流媒体直播系统和 P2P 流媒体点播系统。

1. P2P 流媒体直播系统

对于 P2P 直播系统，何谓直播？直播就是系统播什么用户看什么，是一个实时发生的播放系统，并不能让用户与系统通过交互互相联系在一起。在直播系统中，无论节点是在何时加入进来的，他们的观看进度总是相同的，节点所需的内容也基本相同，因此节点之间的共享资源大大增加。不同于点播要求交互的重要性，如何使直播更加稳定、高质量是直播系统所要解决的重要问题，因此节点间的资源共享方式相对比较简单。P2P 流媒体直播系统

主要分为以下三种。

1）基于树状结构的直播系统

美国斯坦福大学开发的基于应用层多播树 Spreadlt 系统是树状结构直播系统的代表。请求重定向技术是这个系统管理整个多播树的主要手段，再利用 P2P 节点转发流媒体数据。该系统的最大问题是一旦有一方节点退出，会使另外一方被孤立，而导致长时间去重连服务器。其他典型的树形直播系统有：微软研究院设计的 CoopNet 和 Splitstream 系统等。

2）基于网状结构的直播系统

基于网状结构的流媒体直播系统如中国香港科技大学张欣等人研发的 CoolStreaming 流媒体直播系统，基于 Goosip 协议进行数据的转发和传输。网状多播协议因其高可靠性、高可扩展性得以应用于点播系统中，得到了大量用户们的肯定。其他典型的网状直播系统有：PROMISE 系统、PRIME 系统等。

3）基于混合式结构的直播系统

基于混合式的流媒体直播系统如微软亚洲研究院与加拿大西蒙弗莱逊大学共同提出的 mTreebone 系统。顾名思义，即将树形与网状结构相结合起来，把性能强的节点作为树形的树干，把普通的节点用网状与树干结合。

2. P2P 流媒体点播系统

P2P 流媒体点播系统与直播恰恰相反，用户可以通过系统来选择自己想要的节目，同时还可以继续快进、后退等操作。用户的交互体验是点播系统的重中之重，因此，越来越多的用户喜欢使用点播服务。但正是由于点播系统强调交互性，使每个节点的位置都不一样，相互节点利用缓存资源少。因此，如何在点播系统中增加节点缓存资源利用率成为点播系统的重点问题。P2P 流媒体点播系统主要分为以下三种。

1）基于初始数据缓存的点播系统

P2Cast 点播系统基于 Patching 流技术，这个系统会在用户使用之前，向系统内的节点发送命令，使节点预缓存一定时长的流媒体文件，这样在用户发送请求观看命令后，不会因为网络波动而产生延迟，降低用户体验。这种系统的特点是比较容易构建、发展潜力大、花费低，但同时因为需要初始缓存，节点需要大量的存储空间。

2）基于就近数据缓存的点播系统

DirectStream 是基于中心索引服务器而开发的点播系统。节点内存在 FIFO 队列，用服务器索引的方式来了解节点的情况和缓存状况。但其对中心索引服务器的能力和带宽要求过高，导致可扩展性差。

3）基于全局指定缓存的点播系统

P2PStream 是基于全局供应与需求的分段缓存的点播系统。该系统中每个节点缓存的流媒体数据段全部由中央服务器根据全局策略进行统一分配指定。P2PStream 系统中节点根据其 IP 地址的前缀信息被组织到多个网络集群中，集群中的每个节点根据 Round-Robin 策略来支配节点的缓存操作。节点缓存流媒体分段数据后，向所在的集群中心节点报告其所缓存的片段以及上传速率。当节点请求资源时，每个节点都尽量优先从本地集群中的其他节点处获取流媒体资源。当节点新加入系统以后，中央服务器根据每个集群中分段数据的需求与实际供应情况进行评估，然后根据二者关系决定新入节点该缓存什么数据和缓存多少数据。新入节点从中央服务器中获取分段供应。

近年来，越来越多的研究机构开始对 P2P 流媒体进行研究。2007 年，PPS 直播兼点播

的流媒体系统的出现，实现了极少带宽供几万用户同时使用的功能。

在点播系统中，节点播放不同导致差异性，容易造成网络共享资源的减少从而出现网络卡顿、播放异常等问题。所以，如何在提高视频质量的同时改进服务器负荷是当前的一个研究方向。

5.2 P2P 流媒体网络拓扑结构

网络拓扑(Network Topology)是指由各个节点组成的网络中每个节点的位置与状态。节点之间或是节点与主机之间相连接而形成的连接图就是拓扑图。人们在日常中常常见到的拓扑图如星状拓扑、环状拓扑、树形拓扑等。每一种拓扑在网络的扩展性、安全性方面都有区别。和计算机网络拓扑一样，P2P 流媒体网络也分为物理拓扑和逻辑拓扑两种。而人们常常把 P2P 网络拓扑分为四种，分别是中心化拓扑、结构化 P2P 拓扑、非结构化 P2P 拓扑以及分层式拓扑。下面对这四种模型进行分析与比较。

5.2.1 中心化拓扑

中心化 P2P 拓扑结构也叫集中式拓扑结构，它把 P2P 架构和传统 C/S 结合使用，是最早一代的 P2P 技术。中心化拓扑结构如图 5.7 所示，它是由中央目录服务器和各个连接的节点组成，所以它不是纯粹的 P2P 网络结构。在传统的 C/S 架构中，中心服务器往往对资源进行垄断式的管理，客户端只有被动地接收资源，并不能进行交互或者选择，虽然 P2P 中心化拓扑结构中也存在中心服务器，但是中心服务器仅仅是起到资源的索引、维护功能，真正管理和存储资源的是网络结构中的每一个对等节点，它们每个都是独立的中心服务器，当有一个节点请求资源时，中心服务器会指引目标节点，让节点与节点对接。

Napster 是中心化拓扑结构中的典型代表之一，结构如图 5.8 所示。Napster 由一个目录服务器和若干个客户端形成星状结构。目录服务器只负责目标资源的索引和维护功能。使用者在终端上安装客户端后方可加入网络，之后根据自己的需要向目录服务器提出请求。用户提出需要的 MP3 后，客户端开始连接目录服务器，之后目录服务器会开始检索所需资源，然后服务器将检索后得出的结果反馈给客户端，用户再与音乐的作者连接，传输 MP3。Napster 把检索交给服务器，把传输交给用户之间，这样大大地减少了服务器的负荷，同时缩短了传输 MP3 的时间。

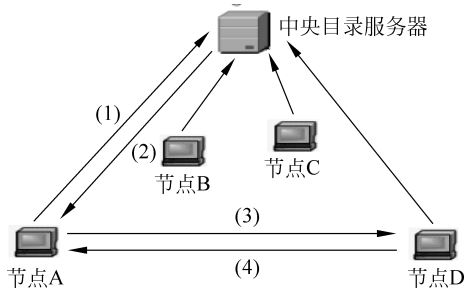

图 5.7 中心化 P2P 网络结构

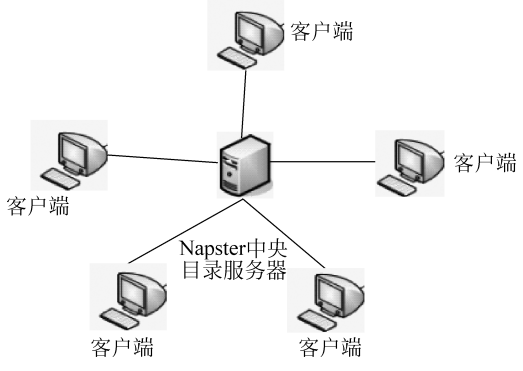

图 5.8 Napster 网络拓扑结构

这种拓扑架构的主要优点如下。

(1) 结构简单,方便维护。

(2) 索引快,目录服务器不负责传输,所以效率高。

然而,它也存在许多的不足。

(1) 和传统 C/S 服务器一样,目录服务器是整个架构的核心,如果服务器崩溃,整个网络将随之崩溃。

(2) 成本高。目录服务器需要不断更新以跟上网络的扩展,容易成为瓶颈。

综上所述,中心化 P2P 网络结构结合了传统 C/S 网络结构,既有优点也延续了缺点,服务器的瓶颈效应使得它不适用于大型网络。

5.2.2 全分布式非结构化拓扑

中心化 P2P 网络模型的瓶颈效应导致网络的不断扩大得不到相应的服务支持,产生了第二代网络拓扑结构。全分布式非结构化拓扑是真正意义上的 P2P 网络拓扑结构,如图 5.9 所示。它移除了中心服务器,节点随机进入网络,然后连接与自己相邻的节点,形成一个逻辑上的覆盖网络。全分布式非结构化拓扑广播发布要查询和发布的资源,各个节点维护一个邻居节点。

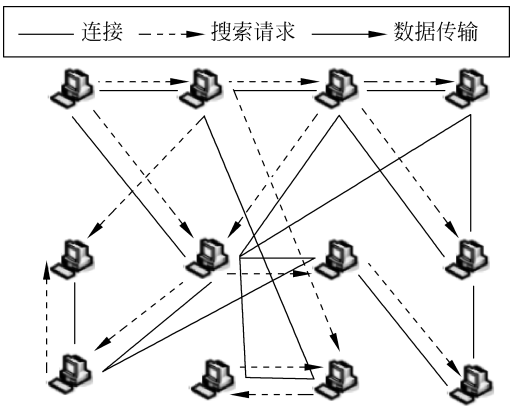

图 5.9 全分布式非结构化网络拓扑

非结构化的意思是每个节点维护的邻居节点都是随机的,网络资源的存储点和拓扑结构是无关的,同时也是随机的。分布式非结构化拓扑的构造图也是随机产生的。图虽然是任意形状的,但还是符合一定的规律,如幂次法则、小世界模型。因为节点位置都是随机的,所以单个节点是不知道整个网络拓扑结构的。一个节点寻找另外一个节点时,首先会向它发出一个查询节点的洪泛请求广播,当邻居节点不是想要寻找的节点时,那么邻居节点会向它的邻居节点查询,直到查询到目标节点为止。在查询时会记下走过的路径以防止回路浪费资源。

Gnutella 模型是使用最广泛的全分布式非结构化拓扑模型,图 5.10 描述了 Gnutella 的工作原理。

若 M2 想要寻找资源 E,那么它会向它的邻居节点 A 和 C 发送寻求广播,如果 A 和 C 都没有资源 E,那么 A 和 C 就会向它们的邻居节点发送请求,M6 和 M4 会进行寻找,此时

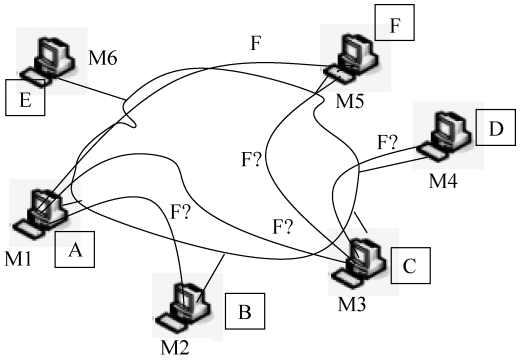

图 5.10　Gnutella 工作原理

发现 M6 有资源 E,那么 M6 就会把资源传输给 M2 完成传输。

从 Gnutella 模型可以看到,全分布式非结构化拓扑模型是纯粹的 P2P,比起一代 P2P 的容错与扩展有了很大的提升。但是逐个节点的洪泛广播在网络规模不断扩大后会造成网络的阻塞和时延,消耗大量带宽的同时阻止了网络继续扩展的可能性。

全分布式非结构化拓扑有以下两个好处。

(1) 容错性高。

(2) 查询结果多。

这种模型也有不足。

(1) 扩展性差。随着网络的扩大,查询量增加,在同一时间网络中可能存在大量的查询信息,导致网络崩溃。

(2) 查询速度慢。需要相邻节点逐个传递,且不保证查询信息的精准程度。

(3) 网络难控制和管理。

5.2.3　全分布式结构化拓扑

由于非结构化 P2P 网络模型存在扩展性不够大、查询效率低等问题,为了进一步提高资源的利用和速度,人们想出了纯 P2P 结构化网络模型。如图 5.11 所示,全分布式结构化网络运用的是第三代技术,并且也是没有中心服务器的网络,它有效地解决了无结构网络的扩展性问题,不再使用洪泛的方式广播请求消息。与无结构网络不同的是,网络中的节点不再是无规律的随机性分布,而是由网络统一管理调度分配,对于资源的索引查询信息也按照

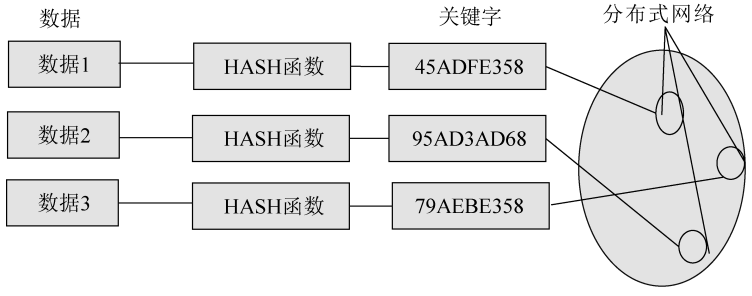

图 5.11　分布式散列表示意图

一定的规则保存在节点中，采取分布式传递，以最少的时间最快速查询到相应节点。此模型的典型代表有 Chord、Kademlia、Tapestry 和 Pastry。

分布式结构化网络主要使用分布式散列表(Distributed Hash Table，DHT)来组织网络中的节点。当进行文件索引时要用到一个(key，value)键值对，key 表示关键词或文件名，value 表示存储文件的节点。当查找文件时会通过搜索 key word 的方式通过 HASH 函数得到 key 值再找到节点，比起洪泛广播更加便捷和快速。从全局上看，所有节点维护的邻居节点路由表都按照某种全局方式将节点组织起来，形成了一个整体的结构化 P2P 网络。

综上所述，分布式结构化网络的长处是：维护量小；扩展性好，分布式散列比起洪泛更加迅速便捷且不会造成阻塞。分布式结构化网络的不足是：DHT 的维护比较复杂，系统会花费时间和精力去应对节点的波动，因此不适合高动态的网络环境。

5.2.4 分层式拓扑

中心化 P2P 网络拓扑结构和分布式网络拓扑结构尽管有明显优点，但同时也有较大缺点，为了更好的网络架构并充分利用前者的优势，研究者提出了分层式 P2P 网络拓扑结构，也称为混合式 P2P 结构，如图 5.12 所示。

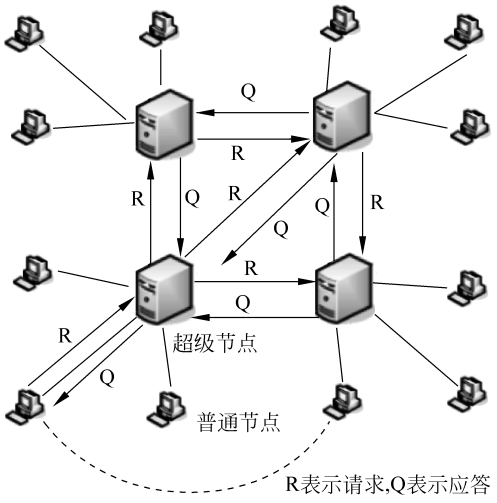

图 5.12 分层式网络拓扑结构

分层式拓扑结构网络吸收了之前网络结构的优点，在增加了检索速度的同时也保证了结构的扩展性。它把网络中的节点分为超级节点和普通节点，超级节点就是如内存大、容量高、运算强等具有优势的节点，这些超级节点在共享资源方面与普通节点功能相同，但是它们还存储网络中其他节点的信息。超级节点与周围的普通节点形成一块区域，这个区域内的超级节点领导其他节点进行查询，不同区域之间使用的是纯 P2P 模式的查询机制。

分层式结构的主要优点如下。

(1) 节点利用率高。每一个节点围绕超级节点进行工作，查询速度大大加快。

(2) 扩展性强，理论上拥有无限可能大的扩展性。

分层式结构的主要缺点：网络拓扑实现较难，同时对超级节点的能力要求高。

5.2.5 P2P 网络拓扑结构比较

P2P 流媒体网络最基本的就是网络拓扑结构，由于 P2P 网络没有传统 C/S 架构中的主服务器，因此选择合适的网络拓扑类型来为后续的数据传输、调度等十分关键。从可靠性、扩展性、发现效率、可维护性和复杂查询几方面对四种结构进行比较，结果如表 5.1 所示。可以得出，在网络扩展性、可靠性、可维护性方面，全分布式结构化拓扑是综合最优的，但它不支持复杂查询，中心化拓扑结构和非结构化拓扑在资源查找方面做得最好，但因为服务器垄断式管理导致扩展性较差。

表 5.1 主要拓扑结构性能比较

拓扑结构 / 比较标准	中心化拓扑	全分布式非结构化拓扑	全分布式结构化拓扑	分层式拓扑
扩展性	差	差	好	中
可靠性	差	好	好	中
可维护性	最好	最好	好	中
发现效率	最高	中	高	中
支持复杂查询	支持	支持	不支持	支持

5.3 P2P 流媒体视频自适应技术

自适应流媒体技术就是能够智能感知用户的下载速度，然后动态调节视频的编码速率，为用户提供最高质量、最平滑的视频演播的技术。

5.3.1 可伸缩视频编码

网络中每一个客户端的实际下载或是上传速度都会受到当时的许多因素影响而每时每刻发生变化。同时，每一个用户对于视频的质量、清晰度都有着不同的要求。所以，P2P 流媒体系统需要提供不同的视频流来满足每一个用户的不同需求。人们将普通的转码、联播技术应用于 P2P 流媒体系统中时，会造成一定的时延并且使服务器负荷运作，导致不能解决该问题。2004 年，为开发一种新的方便灵活的可伸缩视频编码标准，SVC 技术草案应运而生。2004 年 10 月，JVT 最终确定采用德国 HHI 研究所提出的基于 H.264 的扩展架构作为 SVC 标准的实现框架和研究起点，MPEG 和 VCEG 的联合组织 JVT 将此标准定义为 H.264-SVC 标准。通过对该标准的不断修订和完善，H.264-SVC 标准终于在 2007 年 10 月成为正式标准。

SVC 编码是以 H.264 为基础来制定的，与其他的编码方式不同，它通过 H.264 的高效运算只对视频进行一次编码，就可以产生空间、图像质量、时间上的可伸缩。将压缩码流进行提取、解码再传输，重新构造出不同画面质量、帧数的视频资源，如图 5.13 所示。SVC 将编码进行分层，分为基础层和增强层两层，基础层只含有一层，顾名思义是整个视频的基础数据，如分辨率、帧率、图片质量等，增强层则是对视频进行空间、时间、质量增强。所以编码的前提是基础层，增强层并不能进行单独的操作。如图 5.14 中 SVC 码流的模型，每个立方

块代表一个分层组合。

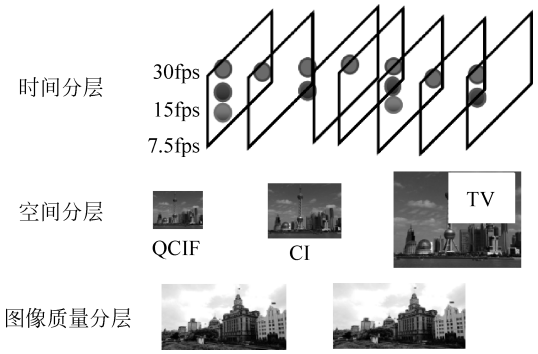

图 5.13　SVC 分层示意图

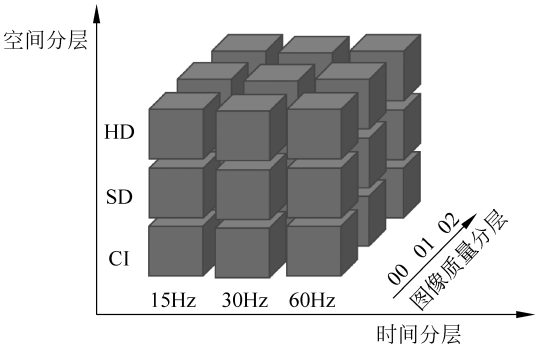

图 5.14　SVC 码流的模型

如图 5.15 所示，为达到空间上的可伸缩，需要通过采样生成不同空间分辨率的数据。层次较低的运动和纹理信息可以被更高层次空间进行利用预测。SVC 的基础就是采用区域编码，在区域内能够实现预测编码的功能，图像往往选取可伸缩编码的方式存储于空间层内。

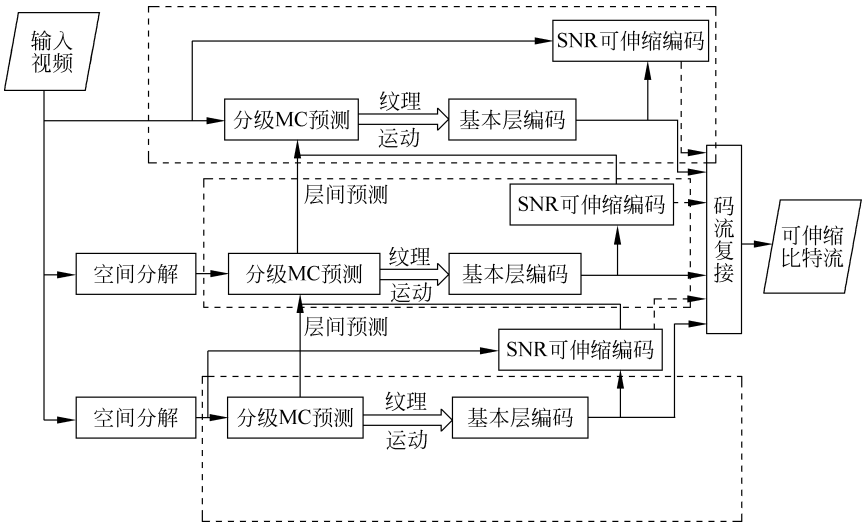

图 5.15　SVC 编码原理

1. 时间可伸缩

JVT 采用层次 B 帧预测技术来实现时间可伸缩，使每一层具有不同的帧率。以图 5.16 为例，图中 SVC 以 8 个图像为一组进行编码，即 Group of Pictures(GOP)等于 8，帧序号为 0～8。GOP 的大小决定时间分级数量，图中时间分为 4 层。每个图像组包含一个关键帧，该关键帧被编码成 I 帧或 P 帧，如帧 0 和帧 8。关键帧之外的帧全部采用双向预测 B 帧进行编码。如帧 0 和帧 8 预测出帧 4，即 B_1 帧。帧 0 和帧 4 预测出帧 2，即 B_2 帧。同样，帧 0 和帧 2 预测出帧 1，即 B_3 帧。以此类推，完成整个图像组的时间可分层编码。层次 B 帧预测时，时间基础层只是用本层前一个图像进行预测，而时间增强层使用两个低时间层的图像进行双向预测。

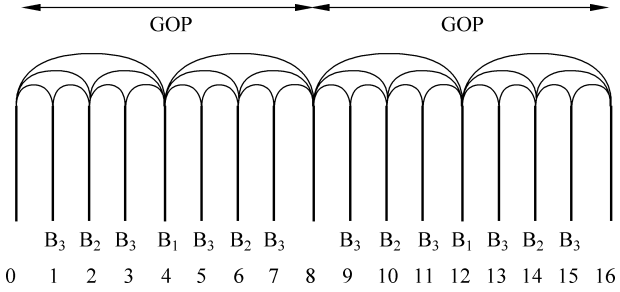

图 5.16　时间分层的编码示意图

2. 空间可伸缩

如图 5.17 所示为空间可伸缩的空间分层示意图。同一时间层内一定存在互相关联且对应的高低层，即层间冗余。空间可伸缩指的是首先对原视频进行采样，然后分出不同分辨率的高低层，再在各层内进行时间和质量的改变。消除在不同层域内的冗余就可以大大提高效率，来达到空域的变化。SVC 提供三种层间预测方案，一是层间帧内预测，这种预测方式的宏块信息完全由层间低分辨率的参考帧上采样实现；二是层间运动预测，宏块预测采用层间参考帧相应块的预测模式，其对应的运动矢量也利用层间参考帧相应块的运动信息预测编码；三是层间残差预测，利用残差的层间相关性对帧间预测的图像继续进行层间预测，以进一步压缩码流。

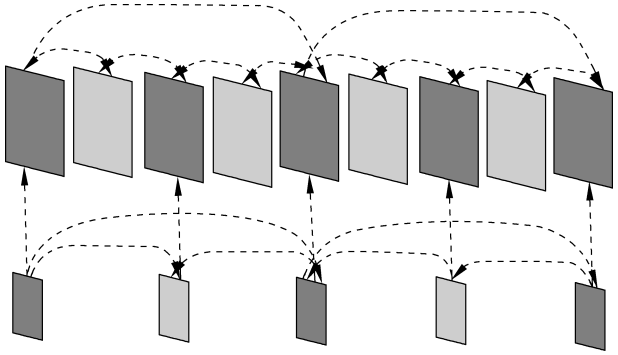

图 5.17　空间分层示意图

3. 质量可伸缩

每一层逐步减小量化步长来达到质量可伸缩的效果。常用的方法有两个：粗粒度质量可伸缩编码(CGS)和中等粒度质量可伸缩编码(MGS)。CGS 和空间可伸缩中的层间预测相似，不同的地方在于它不用进行向上采样，通过采用比前一 CGS 层更小的量化步长来重新量化残差，这样就可以采取到更加精细的纹理，但 CGS 存在的主要问题是它在调整视频质量时只可以使用几个不变的码率点，导致灵活性大大降低。所以 JVT 提出了 MGS 方案，运用关键帧的概念来控制误差传播和质量增强层编码效率两者之间的折中。

5.3.2 P2P 流媒体视频质量自适应技术

现代网络状况波动极大，网络的稳定性会随时改变，最明显的就是网络带宽的波动，带宽很难固定在某一个数值。当网络出现波动时，网络变差，带宽突然降低，用户在观看视频时会出现突然的卡顿，大大影响用户的观影体验；当网络情况良好时，用户在观影时想要更高的清晰度，即更好地利用带宽。此外，为了让 P2P 流媒体系统更好地应用在不同的终端上，需要开发不同类型的视频流来增加用户群。因此，需要研究更好利用带宽的适应网络波动的视频质量自适应技术。

1. 传统的流媒体视频质量自适应技术

流媒体视频质量自适应技术主要包括转码、联播、自适应编码以及多描述编码等。

1) 转码的视频质量自适应

20 世纪 90 年代后期，转码技术是网络上多媒体视频质量自适应的核心解决方案，成为当时的研究和应用热点。转码是指在服务器上保存一个质量足够好的压缩视频流，根据网络变化和用户端情况，对高质量视频流进行部分解码和编码，以输出合适的视频流。主要实现方法是：在发送端和接收端之间建立一个反馈信道，接收端将网络状态(如带宽、丢包率、时延等)反馈给发送端，在可接受的时间范围内，发送端转码器根据这些反馈信息选择性丢弃压缩数据中不会严重影响视频质量的部分，如选择性丢帧或丢弃 DCT 系数的高频分量，以获得较低码率/分辨率/帧率的视频流。这种转码技术虽然能够适应网络带宽的变化，但由于其需要为每个用户定制合适的码流，当大量用户点播时，会加重转码服务器运行负载，并带来少量的视频转码延时。

2) 联播的视频质量自适应

联播技术的基本思想是将同一段视频编码成多个不同分辨率、帧率和码率的压缩视频流，将这些压缩视频流用独立的组播通道发送到网络，用户根据自己的需求和网络的变化选择合适的视频流。这种方案在一定程度上解决了用户和网络的异构问题，并且服务器的计算复杂度很低，不受限于特定的编码标准。视频被编码的码流数目越多，该方案对动态网络变化的适应性就越好。但同时增加了流媒体服务器的存储空间和管理复杂度。

3) 自适应编码的视频质量自适应

自适应编码通常采用 RTP/UDP/IP，发送端将压缩视频流封装成 RTP 包发送给接收端，接收端监测 RTP 数据包的传输时延和丢包率等，来判断网络的实时带宽变化，并通过 RTP 中的 RTCP 将网络情况反馈给发送端，发送端根据网络带宽情况，通过多种方式(例如改变分辨率、跳帧、调节量化参数等)来生成与网络环境匹配的码流。这一方法能很好地适应网络带宽变化，但码率的调整范围受限，重新生成码流会带来一定时延，且不适用于离线

的编码系统。

4）多描述编码的视频质量自适应

多描述编码是将原始视频分成多个描述，对每个描述独立编码形成多个码流。其中，任何一个码流都可以被独立解码成一个满足基本质量的视频。接收端接收到的码流越多，恢复的视频质量越好。多描述编码是一种可兼顾数据传输实时性要求，同时能够解决数据失真问题的编码方案，在多径传输和多播中具有优势。

2. 采用 SVC 的 P2P 流媒体质量自适应技术

因为 SVC 只需一次编码，就可生成一个基础层码流和若干个增强层码流，以自动适应网络的变化和终端设备。所以相比于其他的编码类型，SVC 编码有效地降低了所需资源的数量并且提高效率，它无须固定码率，只需要一个大范围就可以有固定的视频质量和适应大多设备的视频流，灵活且高效。

采用 SVC 的 P2P 流媒体系统可实现以下两方面的视频质量自适应。

（1）在视频接入的初始就进行自适应，根据视频的大小、类型以及网络情况来确定自适应的视频质量。

（2）在视频被用户进行下载的同时进行自适应过程。例如，当用户的网络情况良好时，通过增加增强层下载数量来提供用户更好的观看质量；若用户网络情况差时，就会降低视频质量来保证用户获得更长时间的观看。

3. SVC 的视频质量自适应关键问题

那么如果需要将 SVC 引入 P2P 流媒体系统，仍然需要解决以下几点问题。

（1）视频质量的评判标准。视频质量自适应的最终目标是使用户获得更好的观看体验，因此首先要明确什么才是更好的观看体验。目前的研究中多数是通过 QoS 的指标或者终端下载视频层数来衡量视频质量。

（2）视频质量调整的依据。就是需要考虑何时对视频质量进行改变，例如，将缓存中待播放数据量作为调整的依据。

（3）如何选择下载的分层数据。SVC 可实现时间、空间、质量三个维度的可分层，满足同一网络条件的视频流可能会有多种层组合，这些层组合也可能具有不同等级的质量，只有选出能获得最好的视频观看体验的层组合才能最大化满足用户需求。

5.4 节点缓存策略

节点缓存，就是在使用数据之前，提前将数据缓存在节点内，由于 P2P 网络的特性，每个节点既可以接收也可以发送，所以数据缓存在节点，资源就可以最大化利用了。在 P2P 流媒体服务系统中，为了达到最好的利用效果，系统中的每个节点都会把部分流媒体数据缓存到节点来提供给其他节点使用。对于 P2P 流媒体直播系统，播放时序由流媒体服务器决定，所以即使每个节点在播放时所需的流媒体数据不同，但是由于与后续播放节点时序大致相同，节点缓存数据重合度高，缓存相对简单。对于 P2P 流媒体点播系统，节点的播放时序与加入点播系统的时间相关，节点之间缓存数据重合度低，缓存相对复杂。

5.4.1 节点缓存机制的重要性

P2P 网络流媒体系统有很多方面的性能特征表现，从单个用户角度来看，主要的性能指标包括：启动时延、播放连续度、与服务器的同步性（针对直播系统）、Seeking 时延（针对点播系统）、带宽利用的合理性等。而对于绝大多数用户来说，能最直接体验到的性能是播放的时延和连续度。优化时延和连续度就要从最根本的节点入手，通过适当的节点选择、数据块调度等策略进行改良，因此，节点缓存机制对整个系统性能的优化具有极其重要的意义。

第一，节点缓存资源显著影响着系统中流媒体播放的启动时延与 Seeking 时延。因为当用户启动播放时，节点早就已经缓存并准备好了播放所需要的数据块，那么节点就没必要再从头向其他相邻节点要求传输数据，也就避免了中间所产生的传输时延，从而大大减小了启动时延和 Seeking 时延。但是在选择节点缓存策略时需要更加仔细，因为这不仅是降低时延，也会同时影响播放质量等其他方面。

第二，合理的缓存机制可以有效地降低网络抖动、网络拥塞，进而改善流媒体播放连续性。当用户在播放 P2P 流媒体文件时，由于网络的状况是每时每刻都在变化的，因此往往会产生播放的卡顿和不连续，正是因为节点所携带的数据没有按时在播放点到达，所以合理的节点缓存机制可以预先将所需数据缓存在节点中，避免网络波动而卡顿，影响流畅性。

第三，对于 P2P 网络流媒体系统中的源服务器负载来说，节点缓存机制带来的影响更为突出。因为对等网的特点就是每个节点既是客户端也是接收者，节点既可以发出数据也可以接收数据，所以当节点作为负载为服务器缓解压力时，节点缓存机制就显得尤为重要了。

综上所述，制定有效的、合理的节点缓存机制，以及有效的流媒体缓存策略对 P2P 流媒体系统具有重要的意义。

5.4.2 节点缓存机制问题分析

对于 P2P 网络流媒体系统中的节点缓存机制，需要考虑几个主要的问题：①对等节点应该缓存哪些媒体数据块？②数据块怎么存储在节点之中？节点中的逻辑划分如何展开？③如何更有效地预缓存数据块？④当有限的缓存区数据已满时，缓存数据块的替换问题如何操作？对于这些问题，找到最适合的解决方案就能最大限度地发挥节点缓存机制的作用。

5.4.3 数据块预缓存策略

过去在研究 P2P 流媒体系统时，都默认用户在观看或者点播时都会从头至尾地浏览视频文件，因而忽略了用户的交互操作来设计预缓存策略，然而，经过大量的分析调研用户的观看习惯，事实证实这种假设完全错误。通常情况下，观看视频的用户更加倾向会选择一种随机搜索的方式观看视频。随机搜索指的是用户从视频当前的播放片段随机地跳转到另一个片段进行播放。这种现象产生的原因，一种情况是用户对当前播放的内容没有观看的意向，于是想要跳转至另一个内容进行观看；另一种情况则是用户没有办法可以完整地观看整个视频，于是他们选择快速地浏览视频，或是选择视频最精彩的部分进行观看。随机搜索是流媒体点播服务中的常见行为，是区别于流媒体直播服务的重要特征。当用户进行随机

搜索时，当前节点的视频播放数据就会发生改变，它会向相邻节点请求传输相关数据，而原本的邻居节点所存储的视频数据并不能提供。因此，节点需要重新获取一个新的邻居节点，而在此之前，节点必须向媒体服务器发送数据内容请求，从而增加了媒体服务器负载。另外，若节点等待新的邻居关系建立后，再从邻居节点获取媒体数据进行播放，会极大地增加随机搜索时延(Seeking Delay)。所以，设计有效的节点缓存机制，使节点尽可能地在本地缓存中获取到随机搜索时所需的媒体数据内容，对于减少随机搜索时延以及媒体服务器负载就具有非常重要的意义。

最近几年，对于 P2P 视频点播系统在随机搜索操作发生后，如何制定有效的策略来降低随机搜索时延的问题越来越受到研究者们的关注。例如，VMesh 方案通过有效地利用网络中节点的限制资源来提高数据冗余，从而有效地支持随机搜索等交互性操作。视频媒体被分成多个数据块分布式地存储在网络中的节点上，节点通过结构化的覆盖网络被组织起来。VMesh 在选择需要缓存的数据时，将媒体数据块的受欢迎程度作为主要考虑因素，通过在系统中缓存更多的高流行度的数据块来降低服务器负载。再如，基于指导性搜索的预缓存方案能有效地缩短用户的搜索行为发生后的时延。指导性搜索是指通过收集网络中各个节点的随机搜索信息，计算出视频中从数据块 x 跳转到数据块 y 的概率，从而根据跳转概率给用户的随机搜索操作提供指导性意见。

5.4.4 节点内部的逻辑划分

1. 存储对象的逻辑划分

存储对象的逻辑划分指的是通过各种方式方法对存储对象进行划分，来存储数据。一开始的缓存策略是把文件整个进行存储，但对于流媒体系统来说难以真正实现，因为流媒体文件占用空间通常比较大，如果把整个流媒体文件存储在单个节点内，对单个节点的性能要求极强，且会导致其他节点不能得到充分的利用，还会造成单个节点的瓶颈问题。所以 P2P 流媒体系统采取分段缓存方式，这样一来就可以应对节点在网络中动态地进出和异构性。分段缓存策略的关键在于分段的大小。目前典型的流媒体分段方法主要有前缀分段、固定分段、指数分段和 Adaptive & Lazy 分段。

1) 前缀分段

前缀分段就是将文件的前缀作为一个分段点，把文件分为前缀和后缀，分别存储在代理服务器和资源服务器中。当播放文件时，首先从代理服务器下载前缀文件，再从资源服务器下载后缀文件，因为后缀文件需要从服务器下载，所以当播放时跳过前缀文件直接播放后面时，这个缓存方法就不能有效地缓解带宽压力，也就失去了意义。

2) 固定分段

固定分段就是将流媒体文件等长地分为若干个分段存储，分段流媒体在传输及存储时以逻辑段为处理单元。固定长度的分段可以大大降低分段的复杂性，让系统更加简单地去执行分段操作。如何确定分段的固定长度是这个缓存策略的关键所在。

3) 指数分段

首先把流媒体对象分为各个独立的逻辑块，再由块与块不同的位置进行编号，编号由块到一开始之间的位移距离根据指数进行区分。由于指数的特性，越靠后指数增长倍率越大，越靠后的流媒体分段长度也就越长。由于该方法仅依据分段在流媒体文件中的位置进行长

度区分，缺乏一定的灵活性。

4）Adaptive & Lazy 分段

Adaptive & Lazy 分段方式是借助数据被使用频率来展开分段的。与前几种分段方式相反，前几种分段方式在数据被使用前就规定好分段规则，而 Adaptive & Lazy 分段方法是在数据被使用后统计频率展开分段。因此对于不同的文件都有各自的分段区间，Adaptive & Lazy 分段也就具备了很大的灵活性。基于分段的流媒体缓存策略将流媒体分段作为缓存与替换的基本单元。

表 5.2 对前缀分段、固定分段、指数分段和 Adaptive & Lazy 分段展开对比。通过对比发现每一种分段方法都有各自的优缺点，如 Adaptive & Lazy 分段缓存灵活，固定分段简单易实现；但也有各自的缺点，如 Adaptive & Lazy 分段需要一定量的访问等。因此对于不同的环境，需要选择合适的分段方法。

表 5.2　数据分段策略比较

分段策略	前缀分段	固定分段	指数分段	Adaptive & Lazy 分段
分段大小	前固后不固	分段都是一个固定长度	距离越大，长度越长	由数据被使用频率决定
分段时机	数据使用之前	数据使用之前	数据使用之前	完成数据使用频率调查后
分段优点	对象初始部分请求延迟少	简单易实现	易于快速调整部分缓存对象	独立处理对象，缓存灵活
分段缺点	所需时间过长	缺乏灵活性	划分依据单一	忽略冷门数据

2. 缓存空间的逻辑划分

节点缓存空间的逻辑划分和存储对象的划分不同，主要负责的是特定节点在节点里面如何进行分割的方法，从而来存储流媒体文件信息。节点缓存空间的划分主要分为两种：单频道缓存（Single Video Cache，SVC）和多频道缓存（Multi Video Cache，MVC）。

SVC 的特点是它只把数据分享给与之处于相同频道内的其他用户，不支持不同频道的流媒体数据共享与缓存。用户 A 和 B、C、D 如果在同一时间收看某个视频，那么就把 A、B、C、D 一同加入一个 P2P 网，节点之间互相利用彼此的缓存来播放视频减少数据的传输。SVC 对于节点多的频道具有显著的作用，节点越多，共享的缓存也就越多。

但 SVC 也存在以下几点不足。

（1）SVC 对于节点少用户不多的效果作用不大。因为每个流媒体频道的用户数量都不相同，有的多，有的少，当频道内的节点少时，组成的 P2P 网络可共享的缓存数据也会变少，导致往往直接从流媒体服务器发出请求传输数据。

（2）SVC 不能充分利用其他频道的空闲节点。因为 SVC 把上传与下载分为两个频道，节点在请求数据时只看单个频道的节点，而不在一个频道的就不会采用。

（3）SVC 普遍采用先下载先删除或者采用滑动窗口技术进行流媒体分段的替换，这可能造成某些热门分段也被替换的情况。

因为 SVC 缓存模式的不足之处，研究人员提出了 MVC 缓存模式。MVC 缓存模式可以实现不同频道的节点相互共享，并且支持同一时间下载上传不同频道的数据。

相对于 SVC，MVC 缓存模式的改进之处如下。

(1) SVC 的节点只会接入当前频道组成的 P2P 网络中，而 MVC 可以加入多频道的 P2P 网络中，当在相同频道时，MVC 和 SVC 的缓存模式相同，当其他频道需要节点加入缓存时，MVC 会根据已缓存的数据去上传到其他频道。

(2) MVC 在切换频道后仍然存储之前频道所下载的缓存数据，而 SVC 会在切换频道后把之前的缓存数据全部清空。

MVC 缓存模式的主要优点是：提高了节点的利用率，不仅可以对热门资源进行缓存，同时对于冷门资源也有顾及，使节点的缓存数据完全不被浪费，均衡资源。

综上可以看出，SVC 相较而言缓存方式便捷，但 MVC 更贴合实际用户体验。如 PPLive 系统使用的就是 MVC 的缓存模式。

5.4.5 流媒体缓存预取

流媒体缓存预取指的是节点在播放之前就进行预载数据，预载后续的数据或其他频道的数据，这些数据可以与其他节点共享，提高了播放性能。

(1) 按照空闲对象的不同，流媒体缓存预取分为网络预取和节点预取。网络预取指的是网络状态良好带宽充足的情况下，节点开始预取操作。节点预取指的是节点状态良好存储空间足够时开始预取操作。当在网络状态良好带宽充足时，节点预先把一部分数据缓存下来，这样当网络出现繁忙时，带宽压力增大后，不会出现卡顿或者传输不流畅的情况。这样做有助于节点与节点之间更好地共享数据，但要保证节点不会提前退出或者失连，否则无法与其他节点共享，如果节点退出还会使预取资源浪费，造成后续网络的传输压力。

(2) 按照频道的不同，流媒体缓存预取分为：单个频道和多个频道。单个频道指的是节点会预缓存节点所在频道的后续数据，用户在观看前一部分视频的同时，后续视频就缓存完毕，因此可以有效改善系统的播放性能。多个频道是指节点在用户观看视频时，不仅缓存当前频道的后续资源，还会缓存额外多个频道的视频。在节点缓存完毕之后并不会下线，而是保持在线与其他节点共享资源。

(3) 按照预取的分段序列号是否连续，将缓存预取算法分为连续分段数据预取与非连续分段预取。连续分段预取指的是对节点当前位置后的后续数据进行分段，然后按照顺序进行预取。非连续分段预取是指节点预取的分段序列不再连续，而是以一定的规则对播放位置较远的流媒体数据进行缓存预取。近年来，许多研究者提出了基于概率的非连续分段预取方案。基于概率的预取机制是指节点按照一定的规则计算预取数据段的概率，然后依照概率的大小进行缓存。预取概率的大小通常由数据段距离节点当前播放位置的远近来决定。

5.4.6 流媒体缓存替换

流媒体缓存替换指的是流媒体数据在缓存中的更新方式，它决定了在流媒体数据中需要被更新的对象和更新的对象。它主要包括缓存空间频道替换和缓存存储单元替换两种。缓存空间频道替换是指当节点缓存的空间不足时，节点会选择删除几个流媒体频道进行更新。此方案有利于目的节点的定位和数据调度，但是忽略了流媒体文件内部的流行度因素，在一定程度上降低了节点缓存空间的命中率。缓存存储单元替换则是节点把存储的单位作为替换对象。由于流媒体文件的不同部分具有不同的访问程度，为提高节点缓存空间的命

中率，基于流媒体分段的缓存替换策略成为当今缓存替换的主流方案。

现有的流媒体分段缓存替换算法主要有如下几种。

(1) 基于访问时间的缓存替换。将缓存替换的对象按时间进行排序，典型的缓存替换算法有 FIFO、LRU 等。LRU 算法会换出最少访问的缓存，主要考虑访问时间性。但它只考虑了近期的访问而忽略了访问的频率与换出的大小。如果一个对象只是最近较少访问但总体来说访问量是最大的，那么就大大降低了对象请求的命中率。

(2) 基于访问频率的缓存替换。它以访问频率来进行对象的更新。典型算法有 LFU、LRU-K、LIRS 等。LFU(Least Frequently Used)在更新时首先找到存储空间中访问频率最小的对象然后将它更新替换。但是 LFU 算法存在缓存污染问题，因为 P2P 流媒体系统是不断更新且不断在更新的，随着时间的推移不断改变，LFU 会把历史访问量高却随着时间推移而访问量越来越小的对象仍然保存在节点的缓存中，大大浪费了空间。

(3) 基于访问时间和访问频率的缓存替换。该方案同时兼顾缓存对象的访问时间和访问次数，使得该方案在数据访问模式的更新变化中仍然有着较好的性能。典型算法有 FBR、LRFU 等。LRFU(Least Recently Frequently Used)算法是将 LRU 和 LFU 进行结合，同时衡量访问时间和访问次数，采用权值函数对所有以前访问过的对象进行加权评估，最近一次访问所占权值最大，越往前其权值越小。该算法为每一个缓存项记录一个权值 $W(x)$，每一次直接替换 $W(x)$ 最小的对象。

以上三种流媒体缓存替换更新方法主要以访问时间、访问频率以及时间与频率相结合的方式进行替换的衡量因素。这些方案也被应用于许多基于 P2P 的流媒体分布式缓存中。但是由于在 P2P 网络中单个节点的变化会影响其他节点，使用这些方案进行替换时往往效率不高，如何找到一个合适的替换更新方案值得进一步研究。

5.5 P2P 流媒体数据传输与调度

为使用户能够得到最佳体验和使用感受，节点缓存策略的选择解决了启动时延和播放流畅程度的问题，而选在节点缓存策略之后，节点之间的传输效率也是必不可少的一个关键要素。本节根据在传输过程中角色的不同，主要从传输路径、结构、传输方式的内容来分析传输对整个系统的影响。

5.5.1 P2P 流媒体数据拓扑

当数据在节点与节点之间互相传输时，如何去构建节点互相的组合方式才能高效传输数据，即传输拓扑要解决的问题。通常情况下分为两种结构，即基于树(Tree-based)结构和基于网状(Mesh-based)结构。其中，基于树结构又分为单棵树与多棵树两种结构。

1. 单棵树结构

单棵树结构指的是把所有需要播放同一个视频资源的节点组成一个组播树，由流媒体服务源作为这棵树的根，如图 5.18 所示。传输数据时，首先由根节点

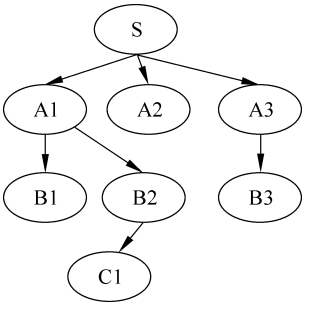

图 5.18 单棵树结构数据传输拓扑

向下发出数据，分发给连接它的子节点，再由子节点继续向下分发给后面的子节点，这样不断往下，前面的子节点作为下一个的父节点传输数据。传输并不会因为父节点的离开而终止，系统会启动备用或推举新的父节点来继续工作，如 DirectStream、P2Cast。

图 5.19 为 P2P 流媒体点播系统 DirectStream，它的主要特点是设置一个目录服务器，负责记录服务器和用户端的重点消息，如 IP 地址、缓存大小等，按就近服务策略引导节点的加入。由于 P2P 网络的特性，每个节点即是主机也是客户端，既接收数据又发送数据，同时把最近的视频资源缓存于节点里面，并提供给后续节点。在接收到用户的请求后，首先查找目录服务器记录的节点信息，将用户节点指引到拥有缓存资源的节点，这样不仅快速而且降低了服务器的负荷。

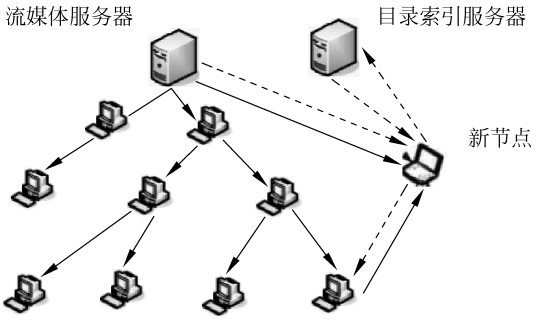

图 5.19　DirectStream P2P 流媒体分发树

图 5.20 为运用了 Patching 技术的 P2P 视频点播服务系统 P2Cast。P2Cast 的构想是假设节点请求到达时间为 t，以请求时间域值 T 为调节因子和约束条件创建流数据组播树。在到达时间 $t \leqslant T$ 时，把所有请求该资源的节点集中起来创建一个单棵组播树，这样就不会因为加入时间的不同而增加服务器负载，同时使得各个节点能够快速获取资源。按就近服务的原则，系统引导新加入节点搜索距其加入之前最近的节点作为 Patching 服务器获取最近的流媒体数据。

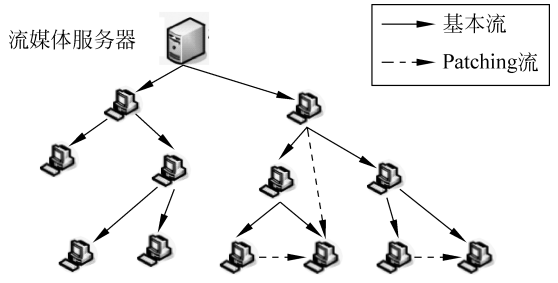

图 5.20　P2Cast P2P 流媒体组播树

单棵树拓扑结构具有如下特点。

(1) 树形拓扑结构，数据分发调度简单，使得用户能够快速获得流媒体资源。

(2) 一些节点不参与数据分发，导致树干中的节点分发压力过大，形成网络瓶颈。

(3) 对网络系统和前驱节点性能要求高，前驱父节点退出或失效而引起的网络抖动导致流数据的不连续，很难实现不间断连续播放，不适合大规模应用。

2. 多棵树结构

Microsoft 研发出的 SplitStream 与 CoopNet 是典型的多棵组播树结构的流媒体方案，都是以多描述编码(MDC)为基础提出的。容错能力强是多棵树与单棵树的最大区别，这对于 P2P 网络不断动态变化的结构而言是至关重要的。因为选取了 MDC 的方式，所以资源解码时不需要相互依靠，同时能够分成多个层分别传输于多棵树上，节点从它们自己所在的多棵树结构上获得视频资源然后进行整合还原成视频，如图 5.21 所示。由于最后完整视频是由多个视频流汇合而成，所以视频流聚集得越密，最后呈现出的画质越好。与单棵树不同的是，当多棵树结构中的一个父节点丢失时，并不会因为父节点的失效而引起数据传输分发的停止导致无法播放，节点还可以从其他树接收数据，只不过会降低最终视频所展现出的画质。绝大多数用户相比于无法播放更倾向于略微地降低画质。

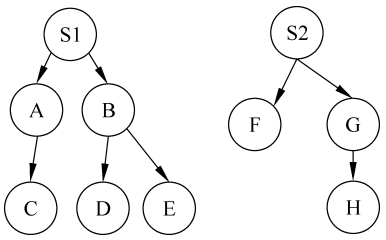

图 5.21　多棵树结构数据传输拓扑

SplitStream 与 CoopNet 相似，它的主要想法是在传输和分发数据的过程中把任务分配给每一个参与形成组播树的节点上，这样不会对单一的一个源节点形成过大的压力。但有所不同的是，它的 Peer 节点只在某一棵组播树中充当中间节点参与数据转发而在其余组播树中只作为叶节点接收数据，这样就可以尽可能地减少节点的动态进出而造成的影响，实现流数据的分发传输任务均匀分布到系统中的所有节点，增强容错性。

多棵树结构具有如下特点。

(1) 最大限度地利用树组的节点，提高视频质量。

(2) 容错能力大，不仅靠单一节点进行数据的传输与接收，而是由许多节点共同工作。

(3) 传输路径固定，一般要求同一组播树内具有相同的传输速率，带宽利用率不高，抗抖动性能弱。

3. 网状结构

尽管人们对单棵树与多棵树这两种基于树形结构而产生的拓扑类型采取了许多不同的改良维护方法，但由于节点流动性大的特性，使得视频最终呈现出来的质量仍然不能满足人们的日常需求。于是人们开始研究基于网状结构的数据传输拓扑类型，对此许多基于网状拓扑结构的流媒体分发方案涌现出来，较为典型的系统包括 Donet、Anysee 等。它们都是采用 Gossip 协议来创造出一个完全随机的拓扑结构。当节点在进行数据传输的过程中，每个节点会根据已缓存的数据形成数据位图，然后在传输过程中，每个节点交换本地已缓存的数据位图，节点会根据自身的视频播放进度和形成的数据位图向相邻节点提出请求来接收数据，这样每个节点都知道自己所需要的是哪一部分资源，以及相邻节点拥有哪一部分资源，形成一个多对多的数据传输模型。

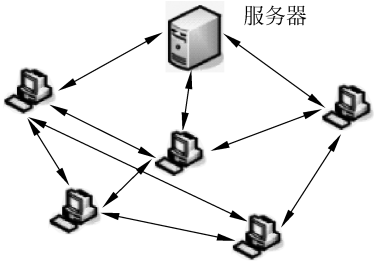

图 5.22　网状结构数据传输拓扑

如图 5.22 所示，网状结构的数据拓扑不再是单一连接在拓扑主干上，而是一个节点同时接收多个节点的数据传输。

Donet 是网状拓扑结构流媒体分发系统的一个典型系统。它利用轻量级的 Gossip 协议来构建一个应用层的覆盖组播网。它包含三个主要模块：①负责维护系统中部分其他在线节点的成员节点管理模块；②负责与邻居节点建立协作关系的伙伴管理模块；③负责和其他节点进行数据交换的数据调度模块。在网中的每个节点都维护两张表，其中一张是系统中的部分在线节点的成员关系列表，另外一张是和本节点交换媒体数据的伙伴关系列表。系统节点利用可扩展成员管理协议（SCAM）来得到其他节点状态信息，当新节点加入系统时将被多次重定向，从而每个节点拥有均匀分布的局部节点视图。在 Donet 中视频数据被分割成相同大小的片断用缓存映射数据位图来表示节点是否拥有某个片断的数据，节点和邻居通过不断交换数据位图来了解相互间的缓存情况，节点根据邻居的情况和在线状态来对邻居节点列表进行管理。

在网状网络拓扑结构中，节点与节点的相连完全是根据它们自身的需求而产生的。节点独立地请求和下载所需的流数据。动态的传输路径导致数据到达节点的时间和顺序也是动态的，所以接收数据的节点也需要动态地改变播放时间，让数据重新按照顺序依次播放。不同于树形结构的节点失效或离开会造成网络的波动，由于拓扑结构是动态的，节点对系统抖动的影响十分微小，因此网状网络拓扑结构具有系统维护开销少、鲁棒性和可靠性高的特点。但在流媒体播放过程中，需要持续对节点和网络性能状况进行实时监测，更新活动候选节点集，选择问题依赖于对底层网络的了解和参数约束，算法复杂度高。

5.5.2 P2P 流媒体数据调度模式

数据调度就是把数据通过调度方式以最高的效率发送到网络中的每一个节点。一个好的调度机制需要满足以下三点：①能够充分利用网络中其他节点的资源（计算能力、存储能力、网络带宽、I/O 端口）；②让用户在观看视频时感到满意，如播放连续性、播放时延等；③能应对不同的网络拓扑结构。目前，“推”“拉”“推拉结合”是主要的三种数据调度机制，由于网络结构的组成很大程度上决定了数据的调度方式，所以通过结合上文不同的网络传输结构来介绍这三种分发方式。

1. “推”模式

在 P2P 流媒体研究的早期，人们根据 IP 组播的经验，提出了“推”模式的数据分发方式。这种模式是应用于树形结构的架构上，流媒体数据依靠树进行分发。

推策略是一种由源节点驱动的数据调度策略。在上文提到过，在树形结构中，P2P 流媒体从树的根节点将流媒体数据推送给其第一级子节点，第一级子节点接收到数据后继续向其下一级子节点推送，以此类推直到最低一级子节点（叶子节点）接收到数据为止。树形结构中的子节点只从一个父节点接收数据（多棵树结构中节点也只从一个父节点接收一个条带），易受节点动态性的影响。为此，研究者提出了候选父节点、联合随机数据转发等辅助机制。另外，为支持树形结构 P2P 视频点播系统，研究者还提出了 Patch 流。这种从上至下逐级推送数据的调度方式就称作推模式。推模式主要用于树结构，父节点直接将数据推给子节点，如 TURINstream。但是由于网状结构的节点都是动态连接，所以没有明确的父子节点关系，这就可能导致两个节点同时发送相同的数据到同一个节点上，不仅浪费了传输带宽，同时也增加了负载。

2. “拉”模式

当前很多流行的 P2P 流媒体点播系统如 CoolStreaming，PPLive 都是采用“拉”这种模

式的数据分发策略。在这种策略中，节点间一般形成网状的网络结构。每一个节点加入网络时，会选择一些节点作为自己的邻居节点，与这些节点在逻辑上形成一个网状的覆盖网。节点与自己的邻居节点定时交换数据的缓存信息，统计数据的分布情况，然后根据自己的需求和数据的分布情况，向拥有需求数据块的邻居节点要求建立数据互通，邻居节点得到响应，就转发所需要的数据。“拉”模式避免了“推”模式的弊端，因为它会要求系统内的节点定时地互相沟通，了解彼此的数据位图信息，这样节点与节点的传输效率大大增加，消除时延。然而，频繁地交换数据位图信息和请求信息会增加系统的开销，导致传输延迟。

3. “推拉结合”模式

总结了“推”“拉”两种模式的数据调度方式后，人们发现无论是树形网络的“推”还是网状网络的“拉”，都存在一定的缺点，于是结合两种模式的优点提出了一种全新的调度模式，即“推拉”结合模式。

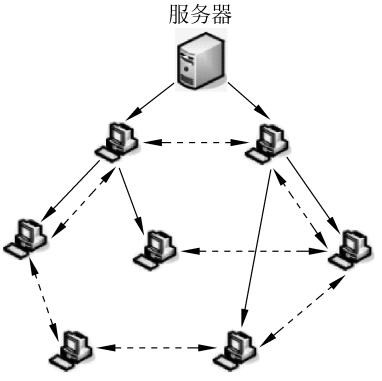

图 5.23　mTreebone 系统架构

图 5.23 是 mTreebone 的系统架构。该系统采用的就是“推拉”结合的方式，例如，首先节点从树形结构的树根节点用“推”的方式获取资源，但如果在这中间发生了因某一个节点而产生的数据丢失或者失效，那么节点就可以采用“拉”模式，主动地去向相邻节点发起请求，传输刚刚丢失的数据。使用这种方式，可以减少数据的调度时延和网络中控制消息的流量，增强了网络的鲁棒性，充分利用了网络中节点的带宽。

5.5.3　数据调度模式比较

下面从带宽利用率、鲁棒性、时延、控制信息开销四个方面对三种数据调度模式进行比较，如表 5.3 所示。从表中可以得出，综合来看“推拉”模式无论在稳定性、时延和系统开销等方面都表现良好，只有带宽效率稍稍低于“拉”模式。而“拉”模式时延长、开销大，并不适合广泛运用。最后，“推”模式虽然时延低、系统开销微乎其微，但是效率低且稳定性差，同样不适用于当今成熟的 P2P 流媒体系统。

表 5.3　各种数据调度模式性能比较

分发方式	带宽效率	稳定性	时延	系统开销
“拉”模式	高	高	长	大
“推”模式	低	低	短	没有
“推拉结合”模式	中	高	短	小

习　　题

1. 简述 P2P 流媒体技术的基本概念。
2. 简述 P2P 流媒体网络的拓扑结构。
3. 举例说明 P2P 流媒体的关键技术。