# 摘　要

随着网络技术的进步，Internet 已经发展成为信息社会中最重要的内容发布系统，但 Internet 中传统的以应用服务器为中心的内容分发网络（Content Distribution Network，CDN）存在着性能瓶颈，不利于网络扩大和维护费用高等问题。与此同时，一种新的分布式资源利用模式——对等网络（Peer-to-Peer，P2P）计算产生了。与传统的 C/S 计算不同的是，P2P 计算一般不需要中心服务器。网络中每个节点既是客户端，又是服务器。P2P 允许计算节点之间的直接交流和协作。P2P 计算可以充分利用 Internet 边缘日益丰富的闲置资源，包括计算、存储、带宽等资源。内容发布和共享是 P2P 计算的一个主要应用领域，基于 P2P 的内容发布系统的特点是能够充分利用大量的客户端资源，减轻或者抛却应用服务器的负载。

P2P 内容分发机制的核心是负载均衡算法以及文件分块选择算法。其中负载均衡算法的目的是选择更接近的、性能更好的节点作为分发服务器。为适应 P2P 网络的各个节点随时变化的状态以及性能，内容分发机制采用动态的基于优先级的负载均衡算法——根据节点 CPU 和内存的状况、两节点之间的逻辑距离以及网络状况进行优先级计算并排序，然后顺序选择节点发出资源请求。文件分块选择算法依据文件两层分块的原理划分为两部分：部分选择算法以及块选择算法。前者采用基于优先级的原则：通过该部分在各节点的分布情况、完成度等因素计算出优先级，选择优先级最大的部分进行分发。后者采用顺序选择以及节点反馈相结合的方法：正常情况下，按照块的顺序进行选择；否则，跳跃到下载点所反馈回来的下一块进行顺序选择。

负载均衡算法使得下载者趋向于向更接近的、性能更好的上传者发出资源请求，以获得更好的分发速度；文件分块选择算法有助于增大网络中文件分块在各个下载者之间的差异性，以便加快下载者之间的分发速度。以此为核心的内容分发机制可以更快的速度从更接近、性能更好的节点处获得资源。

**关 键 字：** 内容分发、对等网络、负载均衡、分块选择、Kademlia

# Abstract

With the development of network technology, Internet has become the most important content distribution system in the information society, but the traditional server-centric content distribution mode is also confronted with performance bottle neck. At the same time, P2P computing as a new mode of utilizing distributed computing resources comes into being. It is different from Client/Server computing, commonly there is no special server in P2P network and nodes can communicate and collaborate directly with each other. P2P computing can utilize increasingly unused resources in the edge of the Internet. Content distribution and sharing is one of the main applications of P2P computing, P2P-based content distribution system can fully utilize resources of vast clients and lighten the load of application server.

The core of the P2P-based content distribution mechanism is the load balance arithmetic and the file's part choosing arithmetic. The purpose of the load balance arithmetic is to choose the node which is nearer or has a better performance as the distribution server. To adapt to the momentarily changing of the state and the performance of the nodes, the priority-based load balance arithmetic is used, whose principle is: calculating and sorting the priority by the states of the CPU and memory, the logic distance of the nodes and the states of network, then send a resource request to the node that has the highest priority. Like the principle of the file's partition, the file's part choosing arithmetic is divided into two: the Part-choosing arithmetic and the Block-choosing arithmetic. The Part-choosing arithmetic is priority-based: calculating and sorting every Part's priority by the situation of the Part in the nodes and the finish percentage, then sending the Part that has biggest priority. The Block-choosing arithmetic's principle is choosing in the order and the node's feedback: In the normal condition choosing the Block in the order; otherwise, choosing the Block beginning with the feedback.

The load-balance arithmetic could make the download-node to send the resource request to the upload-node which is nearer and has better performance; the file's part

choosing arithmetic is good for making the difference of the file's Part in the download-nodes bigger, and making the speed of the distribution faster. The content distribution mechanism whose core is the two arithmetics can make the node to get the resource faster from the nearer and better performance nodes.
**Keyword:** Content Distribution, P2P, Load Balance, Part Choose, Kademlia

# 目录

# 1 绪论

## 1.1 项目背景

目前，人类社会正处在一个信息时代，Internet 作为全球最大的内容发布系统，是信息社会中人们获取各种资讯不可缺少的工具。与此同时，虽然 Internet 中的信息量和用户数量与日俱增，但 Internet 中内容发布的基本方式并没有发生太大的变化。在传统的内容发布模式[1]中，内容的发布由 ICP（Internet Content Provider, Internet 内容提供商）的应用服务器完成，应用服务器通常处于网络的中心，客户端主机处于网络的边缘，客户端需要登录到应用服务器下载或浏览各种内容。在这种发布模式下，网络只表现为一个透明的数据传输通道，客户端也只是一个浏览工具, 而由于 Internet 的 IP 协议是“尽力而为”的，所以这种内容发布的 QoS 是依靠在用户和应用服务器之间端到端地提供充分的、远大于实际所需的带宽来实现的。网络访问对于带宽的要求呈现出端对端的形式，某段网络带宽瓶颈的限制将造成整个网络的拥塞，尤其当大量用户同时访问同一台服务器时，对连接服务器的链路带宽要求更高，不仅大量宝贵的骨干带宽被占用，ICP 的应用服务器的负载也变得非常重，而且不可预计。当发生一些热点事件或出现浪涌流量时，会产生局部热点效应（通常称为 Flash Crowds[2]，或 slashdot effect[3]，指的是对某些资源的访问请求在几分钟内突然之间剧增并且可能持续长达数天时间），从而使应用服务器过载退出服务：例如在 911 事件爆发时，空前的 Web 流量阻塞了相关的新闻网站[4]。为了满足日益增加的访问请求并提高服务质量，服务器必须保证 24×7 的可靠性且足够强大，服务器的处理能力逐渐成为网络发展的瓶颈，为此必须对服务器进行升级来提高性能，或采用多台服务器组成服务器集群来共同处理用户请求。但是，单纯地升级服务器性能的代价非常昂贵，而采用服务器集群的方法也难以进一步扩展。

另一方面，网络技术的发展和应用的需求推动了 P2P 计算（Peer-to-Peer computing）[5−8]

的产生。网络技术的发展主要体现在客户端主机能力的增强和带宽的增加上。现在主流的主机配置都可以达到存储空间在 160G 以上，CPU 速度在 2G 以上，这就使处于网络边缘的主机具备小型服务器的能力。另外，宽带接入技术的实现，使主机的带宽一般都能达到 256K 以上，使主机之间具备直接通信的能力。而传统的 C/S 应用模式使各种 Internet 应用必须通过集中式的服务器，浪费资源且操作复杂。此外，目前 Internet 主机的数目也在不断地增加，据统计已经上亿，而在 C/S 方式下，主机只能处于 Client 的地位实现 Client 的功能，资源得不到充分的利用，因此在网络边缘产生了大量的空闲资源，包括存储能力、CPU 计算能力、信息和人力资源，据估算其中空闲的存储能力和 CPU 能力都达到了上百 T 的数量级。网络边缘存在这么多大量的空闲资源，而处在网络中心的应用服务器随着用户的增加，负载过重，导致整个网络负载极不平衡，从而产生利用这些空闲资源的需求。需求是技术产生的动力。由此新的资源利用模式——P2P 计算产生了。P2P 计算技术出现的目的就是希望能够充分利用互联网中所蕴含的潜在计算资源，尤其是 Internet 边缘的客户端主机资源[9]。

P2P 技术应用非常广泛，其中一个重要应用领域即是内容共享和发布[10−11]。目前已经出现了很多流行的 P2P 内容共享和发布软件，人们可以抛开应用服务器，通过 P2P 软件自由、实时而廉价地共享、发布自己感兴趣的内容。更重要的是 P2P 内容发布和共享系统可以通过 P2P 节点之间的协作，实现内容的快速分发，大大减轻了应用服务器的负载。

## 1.2 课题的提出

虽然 P2P 技术在内容共享和发布领域有着广阔的前景，但是现今在这个领域的商业应用上，传统的内容分发网络[12−13]（Content Distribution Network，CDN）依然是主流产品。造成这种情况的原因主要有以下几个：

（1）法律问题。由于 P2P 网络是一个分布式的对等网络，网络中每个用户在享受别人的资源的同时，也在向别的用户共享自身的资源。而在内容分发网络中，共享的资源就是各种各样的文件，其中尤以多媒体文件为多。这些文件中有相当大的

部分是具有版权，而未经版权拥有者同意就进行共享的

行为是需要负法律责任的。在传统的内容分发网络中，服务器是由明确的法人单位来建立、管理与维护的。对于每一个进入网络的资源都有管理者进行监控，这样可以大量地避免侵权行为。而对于 P2P 网络来说，网络是分布式的，并不存在服务器的概念，并且 P2P 网络实行的是 ID 认证，而不是实名制。如此 P2P 网络中的用户就可以肆无忌惮地对某些具有版权的文件进行共享分发，而不怕负上法律责任。正是由于 P2P 网络的这种特点，使得某些想把 P2P 内容分发技术进行商业应用的公司很容易被其他公司或者个人控告而负上法律责任。

（1）网络硬件。P2P 内容分发的宗旨是尽其所能的分享资源。众多拥有被请求资源的节点同时向请求者进行内容分发，如此大量的网络数据包涌向请求节点。虽然现今的网络通信技术已经取得非常大的进步，但是在中国使用高速的宽带上网的成本还是比较高的。故此中国很多的网民还是使用着比较低速的网络通信，甚至还有很多是共用一条网络的。在这种情况下，过多的数据包同时到达会造成网络拥塞。对于共用一个实 IP 地址的局域网（特别是这个网络使用的是低速网络）来说，局域网中一台主机加入到某个 P2P 分发网络中并共享和下载资源会抢占别的主机的带宽，造成整个网络的网速缓慢，某些时候甚至达到难以忍受的地步。

（2）算法问题。在 P2P 内容分发网络中，P2P 网络拓扑结构和内容分发机制是最重要的两个方面。P2P 网络拓扑结构是整个 P2P 网络的基石，他决定着网络的资源查找与发布的原理与效率。当前，P2P 网络拓扑结构已经发展到第四代了，技术上已经比较成熟。内容分发机制则是整个系统的核心，决定着分发的效率以及速度。一个差的内容分发机制会导致网络出现过多的冗余数据包，并且可能出现某个时间段内某段网络的负载过大。而且过多的节点信息交换也会出现上述情况。故此，一个设计良好的网络拓扑结构以及内容分发机制是解决网络拥塞的有效途径之一。

在上述的三个原因中，法律问题可以通过立法以及协商合作来解决，而硬件问题也随着中国的网络硬件的升级逐步得到解决。最后剩下就是技术问题了，虽然现今很多的研发单位和企业都投入了大量的精力到 P2P 内容分发技术的研发中，但是 P2P 内容分发技术依然具有较大的发展空间。本文正是基于这种情况，对 P2P 环境下的内容分发机制进行研究，试图设计一个较高效的 P2P 内容分发系统。

## 1.3 国内外研究概况

随着 P2P 网络应用越来越广泛，制定 P2P 相关标准的呼声越来越高。和其他研究 Internet 技术的国际组织相比，研究 P2P 应用的工作组还比较少，目前 IETF 还没有关于讨论 P2P 的相关工作组，但其他国际组织已经开始着手制订 P2P 的相关标准。主要工作包括：

（1）Internet2 的 P2P 工作组（Peer-to-Peer Working Group）

Internet2 是一个由 207 所大学联合企业和政府组成的试验新的网络应用和技术、加快下一代互联网建设的组织，它是今天的 Internet 在其发展初期组织的重构。Internet2 P2P 工作组的建立目标是提供一个报告 P2P 和分布式计算领域最新进展和发展趋势的论坛、一个在教育和研究领域使用新的 P2P 和分布式计算应用的场所。Internet2 P2P 工作组成立以后，对 P2P 计算中的术语进行了统一，也形成了相关的草案，但是在标准化的工作方面进展缓慢。目前，Internet2 P2P 工作组已经和全球网格论坛（Global Grid Forum，GGF）合并，由该论坛管理 P2P 计算相关的工作。GGF 负责网格计算和 P2P 计算等相关的标准化工作。 主要进行 P2P 应用系统的确认和 P2P 安全方面的工作，一些具有研究意义的 P2P 网络都包括在这个工作组的研究范围内，如 SETI@home，Chord[14]，JXTA，Groove，OceanStore[15]，Tapestrv[16]。

（2）Intel 的 P2P 工作组

Intel 是 P2P 的热心推动者，并且试图以 P2P 开发组织领导者的身份去领导 P2P 的未来，它试图演绎 Sun 公司在 Java 领域扮演的角色。Intel 建立了包括业界主要厂商 Entropia，Groove Networks，Hewlett Packard，IBM，Sun 等参加的工作组，希望分析 P2P 部署中存在的安全、存储管理和互用性等问题，并促进 P2P 标准的制定和体系结构的建立。但各个厂商出于自己的利益，在标准问题上一直没有达成一致意见。

在国内，P2P 网络正在兴起，越来越多的学术机构研发投身于 P2P 系统的研究中去：

（1）北京大学——Maze。

Maze 是北京大学网络实验室开发的一个中心控制与对等连接相融合的对等计算文件共享系统，在结构上类似 Napster，对等计算搜索方法类似于 Gnutella。网络上的一台计算机，不论是在内网还是外网，可以通过安装运行 Maze 的客户端软件自由加入和退出 Maze 系统。每个节点可以将自己的一个或多个目录下的文件共享给系统的其他成员，也可以分享其他成员的资源。Maze 支持基于关键字的资源检索，也可以通过好友关系直接获得。

（2）清华大学——Granary。

Granary 是清华大学自主开发的对等计算存储服务系统。它以对象格式存储数据。另外，Granary 设计了专门的结点信息收集算法 PeerWindow 的结构化覆盖网络路由协议 Tourist。

（3）华中科技大学——AnySee。

AnySee 是华中科大设计研发的视频直播系统。它采用了一对多的服务模式，支持部分 NAT 和防火墙的穿越，提高了视频直播系统的可扩展性；同时，它利用近播原则、分域调度的思想，使用 Landmark 路标算法直接建树的方式构建应用层上的组播树，克服了 ESM 等一对多模式系统由联接图的构造和维护带来的负载影响。除了 AnySee 以外，华中科技大学的网格实验室还研发了一个 GridCast 的视频点播系统。GridCast 是由华中科技大学计算机学院集群与网格实验室的 P2P e-Learning 组开发。该软件利用 P2P 的多点传输技术实现基于时间组织策略的动态 Overlay 结构，使得加入用户可以很快加入当前的 P2P 网络并获取数据，达到即点即看的效果。与传统 P2P 系统不同的是，GridCast 结合 C/S 模式和 P2P 模式，既可以保证流媒体的服务质量又可以利用 P2P 减轻服务器负担和中心网络的网络流量，极大程度地增强的系统的可扩展性，降低运营成本。总的来说，GridCast 具有如下特点：

1）利用 P2P，人越多越流畅，缓冲次数越少。

2）数据仅在内存当中，无需频繁的磁盘读写，不伤硬盘（除资源拥有者以外）。

3）不在硬盘上存放数据，节省磁盘空间资源（除资源拥有者以外）。

除了以上用于研究的 P2P 内容分发系统以外，在商业应用上国内也有很多企业研发产品。

（1）广州数联软件技术有限公司——POCO。

POCO 是中国最大的 P2P 用户分享平台，是有安全、流量控制力的，无中心服务器的第三代 P2P 资源交换平台，也是世界范围内少有的盈利的 P2P 平台。目前 POCO 已经形成了 4800 万海量用户，平均在线 58.5 万，在线峰值突破 79 万，并且全部是宽带用户的用户群， 成为中国地区第一的 P2P 分享平台。

（2）深圳市点石软件有限公司——OP。

OP——又称为 Openext Media Desktop，一个网络娱乐内容平台，Napster 的后继者，它可以最直接的方式找到您想要的音乐、影视、软件、游戏、图片、书籍以及各种文档，随时在线共享文件容量数以亿计——“十万影视、百万音乐、千万图片”。它整合了 Internet Explorer、Windows Media Player、RealOne Player 和 ACDSee ，是国内的网络娱乐内容平台。

（3）基于 P2P 的在线电视直播——PPLive。

PPLive 是一款用于互联网上大规模视频直播的共享软件。它使用网状模型，有效解决了当前网络视频点播服务的带宽和负载有限问题，实现用户越多，播放越流畅的特性，整体服务质量大大提高！（2005 年的超级女声决赛期间，这款软件非常的火爆，同时通过它看湖南卫视的有上万观众）。

## 1.4 论文结构

本文共分六章来介绍课题的研究内容。第一章为概述，介绍课题的来源、目的和意义，分析了国内外对 P2P 内容分发网络的研究现状。第二章详细介绍了 P2P 的概念，特点以及发展状况，并给出了开源软件 Emule 的分布式结构化覆盖网络 Kademlia 的介绍。第三章详细介绍了内容分发机制中分发方式，负载均衡算法以及文件分块选择算法。第四章给出了原型系统的设计与实现，第五章给出了原型系统的测试与结果分析。最后对全文的研究工作进行总结和展望。

# 2 P2P 技术概述

## 2.1 P2P 技术概述

### 2.1.1 P2P 的含义与特点

尽管 P2P 网络只是在最近几年才声名鹊起，但实际上 P2P 的概念很早就已经提出[17]。20 世纪 60 年代后期的 ARPANET 和 80 年代后期出现的 Usenet 都具有 P2P 网络的性质：它们都是分布的、分散的用于用户之间文件传输和共享的系统。但是在上世纪 90 年代初期随着 Internet 的迅猛发展、视窗操作系统的普遍应用以及当时网络硬件的限制，WWW 作为一种新兴的分布式系统逐渐代替了这些早期 P2P 系统的使用。90 年代后期一系列新技术的发展又促使 P2P 网络的新发展：首先是 MP3 音频编码技术和 Divx 等视频技术，以及与之配套的免费 MP3 播放器软件的出现，使得多媒体文件的压缩和播放更加方便和有效，尤其是 1997 年 Winamp 的推出，极大地推动了多媒体文件的广泛应用；其次，宽带技术的普及使得用户能廉价、高速地接入 Internet。这两种技术的出现极大地刺激了音乐文件在网络上进行免费交换和传输，终端用户无须再支付大量的金钱给唱片公司来购买 CD 音乐。1999 年推出的 Napster 和 2000 年推出的 Gnutella[18]系统迅速流行，速度之快超乎想象。从这时开始 P2P 应用重新引起了学术界和产业界的浓厚兴趣，它们开始投入大规模的人力、物力来展开对这种分布式系统的研究和开发。

按照维基百科全书的定义，P2P 网络是指不依赖于专门的服务器，而由用户计算机之间连接彼此的对等的通信方式。它能够在计算机之间直接交换服务或者数据。在一个纯粹的 P2P 系统中，唯一的组件是称作 Peer 的节点，组件没有客户机和服务器的区别，通过贡献计算机资源（如 CPU 处理能力、存储空间）以及分享其他节点贡献的资源的同时起到服务器和客户机的作用。一个更精确的定义是：P2P 网络是一个分布式系统，其中称为“Peer”

的网络可寻址计算单元具有相互可比较的信息、共享资源以及可使用的服务。P2P 网络利用大量可扩展的自组织节点的联合存储效应提供服务，重点是强调网络以无中心方式进行工作的特点，这吸引了每一个家用计算机用户的加入，因为他可以独立地选择自己的工作策略进行工作并能受益于整个系统。

P2P 技术的特点体现在以下的几个方面：

（1）非中心化（Decentralization）[19]：网络中的资源和服务分散在所有结点上，信息的传输和服务的实现都直接在结点之间进行，可以无需中间环节和服务器的介入，避免了可能的瓶颈。P2P 的非中心化基本特点，带来了其在可扩展性、健壮性等方面的优势。

（2）可扩展性：在 P2P 网络中，随着用户的加入，不仅服务的需求增加了，系统整体的资源和服务能力也在同步地扩充，始终能较容易地满足用户的需要。整个体系是全分布的，不存在瓶颈。理论上其可扩展性几乎可以认为是无限的。

（3）健壮性：P2P 架构天生具有耐攻击、高容错的优点。由于服务是分散在各个结点之间进行的，部分结点或网络遭到破坏对其它部分的影响很小。P2P 网络一般在部分结点失效时能够自动调整整体拓扑，保持其它结点的连通性。P2P 网络通常都是以自组织的方式建立起来的，并允许结点自由地加入和离开。P2P 网络还能够根据网络带宽、结点数、负载等变化不断地做自适应式的调整。

（4）高性能/价格比：性能优势是 P2P 被广泛关注的一个重要原因。随着硬件技术的发展，个人计算机的计算和存储能力以及网络带宽等性能依照摩尔定理高速增长。采用 P2P 架构可以有效地利用互联网中散布的大量普通结点，将计算任务或存储资料分布到所有结点上。利用其中闲置的计算能力或存储空间，达到高性能计算和海量存储的目的。通过利用网络中的大量空闲资源，可以用更低的成本提供更高的计算和存储能力。

（5）隐私保护：

在 P2P 网络中，由于信息的传输分散在各节点之间进行而无需经过某个集中环节，用户的隐私信息被窃听和泄漏的可能性大大缩小。此外，目前解决 Internet 隐私问题主要采用中继转发的技术方法，从而将通信的参与者隐藏在众多的网络实体之中。在传统的一些匿名通信系统中，实现这一机制依赖于某些中继服务器节点。而在 P2P 中，所有参与者都可以提供中继转发的功能，因而大大提高了匿名通讯的灵活性和可靠性，能够为用户提供更好的隐私保护。

（1）负载均衡[20]: P2P 网络环境下由于每个节点既是服务器又是客户机，减少了对传统 C/S 结构服务器计算能力、存储能力的要求，同时因为资源分布在多个节点，更好的实现了整个网络的负载均衡。

### 2.1.2 P2P 的体系结构模型

P2P 系统最大的特点就是用户之间直接共享资源，其核心技术就是分布式对象的定位机制，这也是提高网络可扩展性、解决网络带宽被吞噬的关键所在。迄今为止，P2P 网络已经历了三代不同网络模型[21−22]，各种模型各有优缺点，有的还存在着本身难以克服的缺陷，因此在目前 P2P 技术还远未成熟的阶段，各种网络结构依然能够共存，甚至呈现相互借鉴的形式。

（1）集中目录式结构

集中目录式 P2P 结构是最早出现的 P2P 应用模式，因为仍然具有中心化的特点也被称为非纯粹的 P2P 结构。用于共享 MP3 音乐文件的 Napster[23]是其中最典型的代表（如图 2-1 所示，S 表示服务器，P 表示对等节点 Peer。），其用户注册与文件检索过程类似于传统的 C/S 模式，区别在于所有资料并非存储在服务器上，而是存贮在各个节点中。当某个用户需要某个音乐文件时，首先连接到 Napster 服务器，在服务器上进行检索，并由服务器返回存有该文件的用户信息或者主机信息，再由请求者直接连到文件所有者传输文件。这种网络结构非常简单，但是它显示了 P2P 系统信息量巨大的优势和吸引力，同时也揭示了 P2P 系统本质上所不可避免的两个问题：法律版权和资源浪费的问题。当系统的中心服务器出现故障而瘫痪时，整个系统将会停止运行。而且当网络中的用户和资源增加时，中心服务器的维护、更新和查询

的压力将随之增加，成本相应增加。而且因为中心目录索引服务器为用户提供的相关资源信息是通过“自由”方式共享的，当涉及版权或知识产权时，中心服务器将负有法律责任。当初 Napster 公司就是因为法律问题而被勒令关闭的。

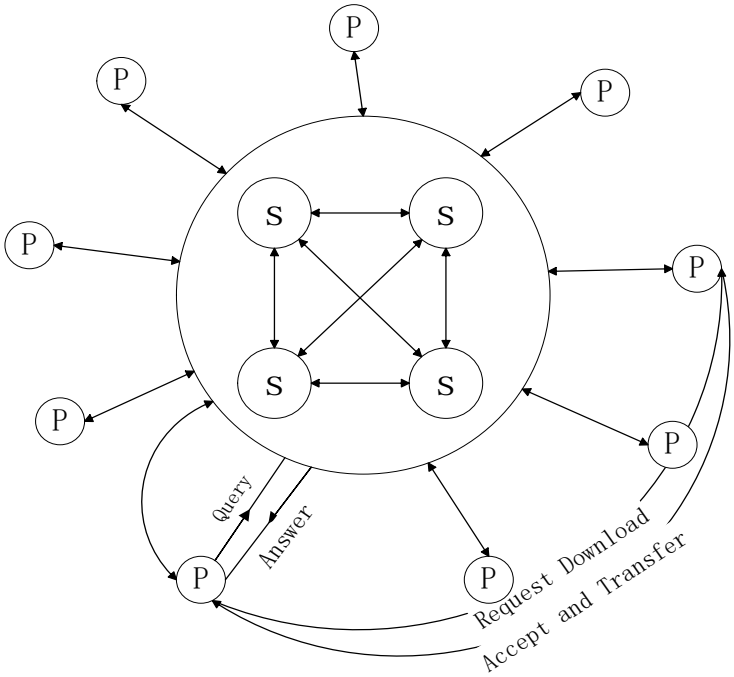


图 2-1Napster 集中目录结构原理图

（1）纯 P2P 非结构化网络模型

纯 P2P 模式[24]也被称作广播式的 P2P 模型。它取消了集中的中央服务器，每个用户随机接入网络，并与自己相邻的一组邻居节点通过端到端连接构成一个逻辑覆盖的网络。对等节点之间的内容查询和内容共享都是直接通过相邻节点广播（基于完全随机图的洪泛 Flooding 发现和随机转发 Random Walker 机制）接力传递，同时每个节点还会记录搜索轨迹，以防止搜索环路的产生。为控制搜索消息的传输，通过 TTL（Time To Live）的减值来实现。

Gnutella[25]模型是现在应用最广泛的纯 P2P 非结构化拓扑结构（如图 2-2 所示），它解决了网络结构中心化的问题，扩展性和容错性较好，但是 Gnutella 网络中的搜索算法以泛洪的方式进行，控制信息的泛滥消耗了大量带宽并很快造成网络拥塞甚至网络的不稳定。同时，局部性能较差的节点可能会导致 Gnutella 网络被分片，从而导致整个网络的可用性较差，另外这类系统更容易受到垃圾信息，甚至是病毒的恶意攻击。另外，由于没有确定拓扑结构的支持，非结构化网络无法保证资源发现的效率，即时需要查找的目的节点的确存在，发现也有可能失败。

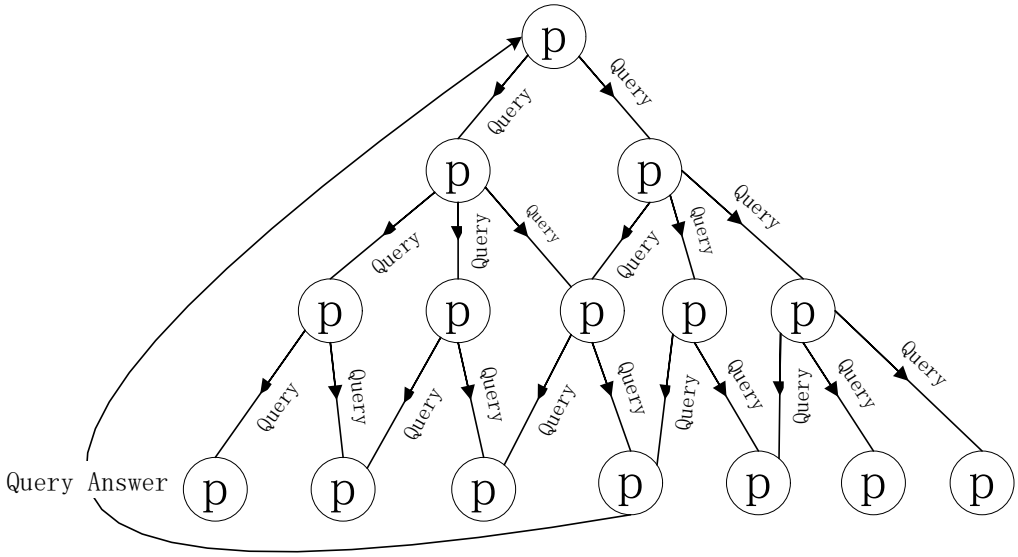


图 2-2 纯 P2P 非结构化模型资源查找原理图

（1）混合式网络模型

Kazaa 模型是 P2P 混合模型的典型代表（如图 2-3 所示，SP 表示超级节点 Super Peer，P 表示普通节点 Peer，双箭头连线表示两者是可连通的。），它在纯 P2P 分布式模型基础上引入了超级节点的概念，综合了集中式 P2P 快速查找和纯 P2P 去中心化的优势。Kazaa 模型将节点按能力不同（计算能力、内存大小、连接带宽、网络滞留时间等）区分为普通节点和搜索节点两类（也有的进一步分为三类节点，其思想本质相同）。其中搜索节点与其临近的若干普通节点之间构成一个自治的簇，簇内采用基于集中目录式的 P2P 模式，而整个 P2P 网络中各个不同的簇之间再通过纯 P2P 的模式将搜索节点相连起来，甚至也可以在各个搜索节点之间再次选取性能最优的节点，或者另外引入一新的性能最优的节点作为索引节点来保存整个网络中可以利用的搜索节点信息，并且负责维护整个网络的结构。

由于普通节点的文件搜索先在本地所属的簇内进行，只有查询结果不充分的时候，再通过搜索节点之间进行有限的泛洪。这样就极为有效地消除纯 P2P 结构中使用泛洪算法带来的网络拥塞、搜索迟缓等不利影响。同时，由于每个簇中的搜索节点监控着所有普通节点的行为，这也能确保一些恶意的攻击行为能在网络局部得到控制，并且超级节点的存在也能在一定程度上提高整个网络的负载平衡。

总的来说，基于超级节点的混合式 P2P 网络结构比以往有较大程度的改进。

然而，由于超级节点本身的脆弱性也可能导致其簇内的结点处于孤立状态，因此这种局部索引的方法仍然存在一定的局限性。这导致了结构化的 P2P 网络模型的出现。

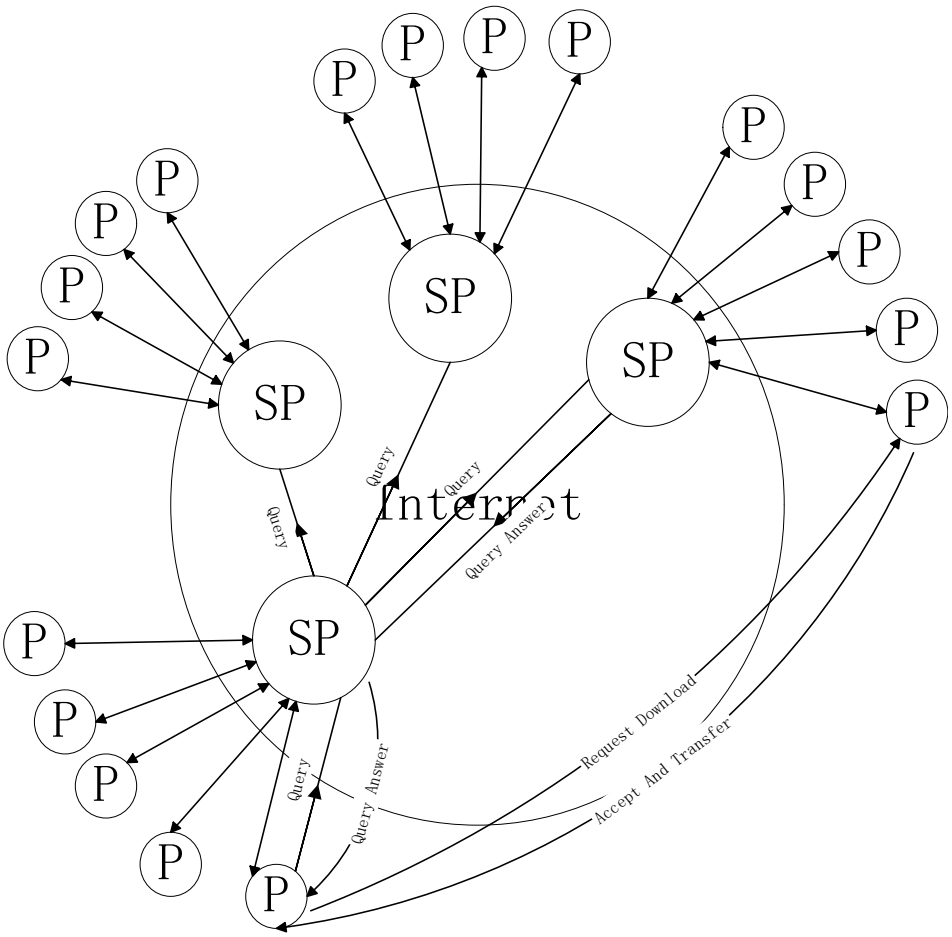


图 2-3P2P 混合模型原理图

（1）结构化网络模型

所谓结构化与非结构化模型的根本区别在于每个节点所维护的邻居是否能够按照某种全局方式组织起来以利于快速查找。结构化 P2P 模式是一种采用纯分布式的消息传递机制和根据关键字进行查找的定位服务，目前的主流方法是采用分布式哈希表（DHT，Distributed Hash Table）技术[26]

，这也是目前扩展性最好的 P2P 路由方式之一。由于 DHT 各节点并不需要维护整个网络的信息，只在节点中存储其临近的后继节点信息，因此较少的路由信息就可以有效地实现到达目标节点，同时又取消了泛洪算法。该模型有效地减少了节点信息的发送数量，从而增强了 P2P 网络的扩展性。同时，出于冗余度以及延时的考虑，大部分 DHT 总是在节点的虚拟标识与关键字最接近的节点上复制备份冗余信息，这样也避免了单一节点失效的问题。

目前基于 DHT 的代表性的研究项目主要包括加州大学伯克利分校的 CAN[27−28]项目和 Tapestry 项目（如图 2-5 所示），麻省理工学院的 Chord 项目（如图 2-4 所示）、IRIS 项目，以及微软研究院的 Pastry[29−30]项目等。这些系统一般都假定节点具有相同的能力，这对于规模较小的系统较为有效。但这种假设并不适合大规模的 Internet 部署。同时基于 DHT 的拓扑维护和修复算法也比 Gnutella 模型和 Kazaa 模型等无结构的系统要复杂得多，甚至在 Chord 项目中产生了“绕路”的问题。事实上，目前大量实际应用还大都是基于无结构的拓扑和泛洪广播机制，现在大多采用 DHT 方式的 P2P 系统缺乏在 Internet 中大规模真实部署的实例，成功应用还比较少见。

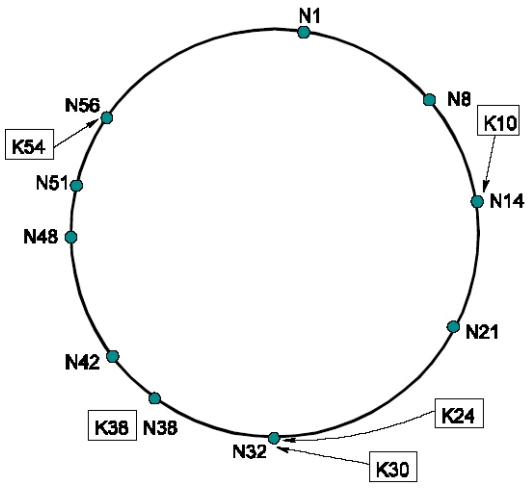


图 2-4Chord 的资源发现

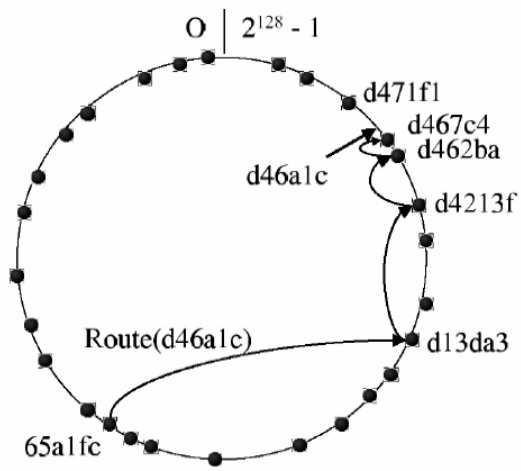


图 2-5Pastry 的资源发现

综合上面的四个 P2P 网络模型，它们的之间的性能差异比较如表 2-1 所示。从这四个 P2P 网络拓扑模型的性能比较中，可以看出纯 P2P 结构化模型的总体性能是最好的。

表 2-1P2P 四种网络模型的性能比较

| 比较标准/拓扑结构 | 集中目录式结构 | 纯 P2P 非结构化拓扑 | 混合式拓扑 | 纯 P2P 结构化拓扑 |
|---|---|---|---|---|
| 可扩展性 | 差 | 差 | 中 | 好 |
| 可靠性 | 差 | 好 | 中 | 好 |
| 可维护性 | 最好 | 最好 | 中 | 好 |
| 发现算法效率 | 最高 | 中 | 中 | 高 |

| | | | | |
|---|---|---|---|---|
| 复杂查询 | 支持 | 支持 | 支持 | 不支持 |

## 2.2 Emule 分析

### 2.2.1 Kademlia 网络

Emule 是一个支持 P2P 文件上传与下载的软件，在文件传输上采用的是 P2P 模式。但是对于资源发现以及资源发布，Emule 拥有两种不同的方式。这两种方式采用不同的网络模式：第一种是服务器模式，该方式拥有一群专用于资源发现与发布的服务器，这些服务器具有分层结构，大体上类似于内容分发网络（CDN）的服务器集群构造；另外一种是纯 P2P 结构化模式，没有任何的服务器，每个节点都是对等的，这就是 Kademlia 网络[31]。本系统的目标是 P2P 环境下的内容分发机制，因此本系统的资源发现与发布采用的是第二种方式——Kademlia 网络。

Kademlia（简称 Kad，如图 2-6 所示）属于一种典型的结构化 P2P 覆盖网络（Structured P2P Overlay Network），以分布式的应用层全网方式来进行信息的存储和检索是其尝试解决的主要问题。在 Kademlia 网络中，所有信息均以<key, value>的哈希表条目形式加以存储，这些条目被分散地存储在各个节点上，从而以全网方式构成一张巨大的分布式哈希表。

Kademlia 网络的基本原理是：网络中的每个节点都分配一个 128 位的 ID（该 ID 与节点所在的物理网络无关），两节点的远近用两个节点的 ID 的异或值的大小来表示；每个节点都保存着两个字典——关键字字典和文件索引字典，这些关键字字典都使用对关键字进行 Hash 计算得到的 Hash 值作为 key，关键字字典保存的 value 是文件的 Hash 值，而文件索引字典保存的 value 是保存文件的节点的详细信息（这些信息包括用户 ID、IP、端口等），其 key 则是关键字字典的 value 值（即文件名的 Hash 值）；当用户要发布一个文件时，首先把各个关键字进行 Hash 计算获得 128 位的关键字 Hash 值，然后再把文件进行 Hash 得到文件 Hash 值，然后把这些信息保存到两个字典中，保存位置（即这些信息保存到 Kademlia 网络中的哪个节点上）

与 key 有关，也就是把条目保存到拥有与 key 最接近的 ID 的节点处（这些保存节点的数目设定是根据系统动态或者静态设定的）；当进行资源查找时，首先对关键字进行 Hash，根据得到的 key 值向拥有与关键字 Hash 值接近的用户 ID 的节点查找关键字字典，查找者通过关键字查找得到文件名的 Hash 值以后，然后再根据文件名 Hash 值到用户 ID 与文件名 Hash 值接近的节点去通过文件索引字典查找具体的文件存放节点；一般来说，系统会返回一个节点集合，然后系统可以根据系统策略来选择节点进行内容传输。

从上面的原理可以看出，Kademlia 的资源发布与查找跟用户 ID、文件名 Hash 值以及关键字 Hash 值有很大关系。事实上用户 ID、文件名 Hash 值以及关键字 Hash 值都是 128 位的二进制数。它们的位数以及计算方法都是一样，这样的目的是为了实现上面介绍的资源发布与查找功能。通过 Hash 值与用户 ID 的异或得到两者的差别，如果差别小的话，则该文件名或者关键字信息就保存到该接点的对应的字典中。

在 Kademlia 网络中，每个节点并不需要知道整个网络的拓扑结构，而只是知道与本节点相邻的或者有联系的节点的信息即可。故此在 Emule 中，每个 Kademlia 网络中的节点都有一个长度为 128 的数组，该数组的每个元素保存的是一个链表，该链表的每个节点保存的信息为一个 Kademlia 为网络中的节点信息。对于数组中的第 i（$0 \le i \le 127$）个元素，其保存的链表为与本节点的距离为 distance 的节点信息（其中 $2^i \le distance < 2^{i+1}$）。如果节点需要查找某个节点或者某个文件的时候，首先根据节点的 ID 或者文件的 Hash 值得到与本节点的距离 distance，然后根据该 distance 的值向数组的第 k 个元素链表中的各个节点发出查找信号（$2^k \le distance < 2^{k+1}$），这样查找就会向着目标越靠越近，直到查找成功或者超时。

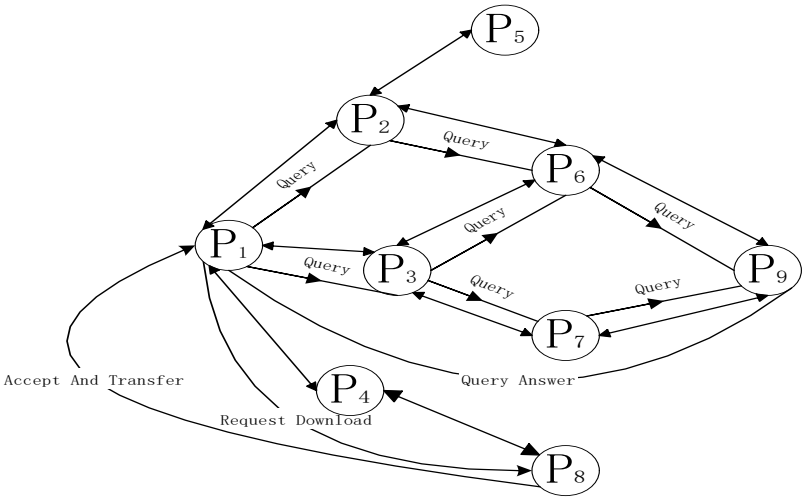


图 2-6Kademlia 网络的资源发现原理图

## 2.2.2 Hash 算法 SHA1

散列(Hash)算法，就是把任意长度的输入（又叫做预映射），通过散列算法，变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，而不可能从散列值来唯一的确定输入值。

数学表述为：h=H(M)，其中 H()为单向散列函数，M 为任意长度明文，h 为固定长度散列值。

在信息安全领域中应用的 Hash 算法，还需要满足其他关键特性：

（1）单向性。也就是说 Hash 算法从预映射，能够简单迅速的得到散列值，而在计算上不可能构造一个预映射，使其散列结果等于某个特定的散列值，即构造相应的 $M = H^{-1}(h)$ 不可行。这样，散列值就能在统计上唯一的表征输入值，因此，密码学上的 Hash 又被称为“消息摘要”，就是要求能方便的将“消息”进行“摘要”，但在“摘要”中无法得到比“摘要”本身更多的关于“消息”的信息。

（2）抗冲突性。即在统计上无法产生 2 个散列值相同的预映射。给定 M，计算上无法找到 $M'$ 满足 H(M)=H($M'$)，此谓弱抗冲突性;计算上也难以寻找一对任意的 M 和 $M'$，使满足 H(M)=H($M'$

)，此谓强抗冲突性。当预映射的空间很大的情况下，算法必须有足够的强度来保证不能轻易找到相同映射值的人。

（3）映射分布均匀性和差分分布均匀性。散列结果中，为 0 的比特数和为 1 的比特数,其总数应该大致相等；输入中一个比特的变化，散列结果中将有一半以上的比特改变，这又叫做“雪崩效应”；要实现使散列结果中出现 l 比特的变化，则输入中至少有一半以上的比特必须发生变化。其实质是必须使输入中每一个比特的信息，尽量均匀的反映到输出的每一个比特上去；输出中的每一个比特，都是输入中尽可能多比特的信息一起作用的结果。

SHA1[32]是目前应用最为广泛的 Hash 算法之一，而它是以 MD4 为基础设计的。

（1）MD4

MD4（RFC1320）是以 MIT 的 Ronald L . Rivest 在 1990 年设计的，MD 是 MessageDigest 的缩写。它适用在 32 位字长的处理器上用高速软件实现——它是基于 32 位操作数的位操作来实现的。它的安全性不像 RSA 那样基于数学假设，尽管 DenBoer、Bosselaers 和 Dobbertin 很快就用分析和差分成功的攻击了它 3 轮变换中的 2 轮，证明了它并不像期望的那样安全，但它的整个算法并没有真正被破解过，Rivest 也很快进行了改进。

（2）SHA1

SHA1 是 由 NIST NSA 设计为同 DSA 一起使用的，它对长度小于 $2^{64}$ 的输入，产生长度为 160 位的散列值，因此抗穷举性更好。SHA1 设计时基于和 MD4 相同原理，并且模仿了该算法。因为它将产生 160 位的散列值，因此它有 5 个参与运算的 32 位寄存器字。

消息首先被拆成若干个 512 位的分组，其中最后 512 位一个分组是“消息尾+填充字节（100……0 ）+64 位消息长度”，以确保对于不同长度的消息，该分组不相同。64 位消息长度的限制导致了 SHA1 安全的输入长度必须小于 $2^{64}$ 位 ，因为大于 64 位的长度信息将被忽略。

接着各个 512 位消息分组以 16 个 32 位字的形式进入算法的主循环。主循环也同样是 4 轮，但每轮进行 20 次操作。最后生成 160bit 的输出。

在 NIST 新的 Advanced Encryption Standard（AES）

中，使用了长度为 128、192、256 位的密钥，因此相应的设计了 SHA256、SHA384、SHA512，它们将提供更好的安全性。

Hash 算法在信息安全方面的应用主要体现在以下的 3 个方面：

（1）文件检验

在文件传送后，将得到的目标文件计算哈希值，与源文件的哈希值对比，由两者的一致性，可以从统计上保证两个文件的每一个码元也是完全相同的。这可以检验文件传输过程中是否出现错误，更重要的是可以保证文件在传输过程中未被恶意篡改，起到二进制文件“数字指纹”的作用。

（2）数字签名

Hash 算法也是现代密码体系中的一个重要组成部分。由于非对称算法的运算速度较慢，所以在数字签名协议中，单向散列函数扮演了一个重要的角色。在这种签名协议中，双方必须事先协商好双方都支持的 Hash 函数和签名算法。签名方先对该数据文件进行计算其散列值，然后再对很短的散列值结果（如 SHA1 是 20 字节），用非对称算法进行数字签名操作。对方在验证签名时，也是先对该数据文件进行计算其散列值，然后再用非对称算法验证数字签名。

（3）鉴权协议

在传输信道是可被侦听，但不可被篡改的情况下，利用哈希算法是一种简单而安全的方法。需要鉴权的一方，向将被鉴权的一方发送随机串（“挑战”），被鉴权方将该随机串和自己的鉴权口令字一起进行 Hash 运算后，返还鉴权方，鉴权方将收到的 Hash 值与在己端用该随机串和对方的鉴权口令字进行 Hash 运算的结果相比较，如相同，则可在统计上认为对方拥有该口令字，即通过鉴权。

在本系统中，Hash 算法 SHA1 的主要作用是对文件分块的检验。系统采用的是 Emule 的文件分块功能，具体的分块思想在下一章的文件分块选择算法再进行详细说明。对于某个文件的每个分块的内容作为 SHA1 算法的输入，得到该分块的 Hash 值，然后再进行文件传输之前，先把文件各个分块的 Hash 值发送给请求者，之后当请求者每获得一个分块的时候，都进行 Hash 值计算，确定分块的正确性。

另外，对于每一个用户，系统会根据用户信息产生一个 Hsah 值的 ID，该 ID 的主要作用已经在上一小节进行介绍，这里不再重复。

## 2.3 本章小结

本章主要介绍了 P2P 网络的含义以及特点，简要的分析了 P2P 的四个网络拓扑模型，并对开源软件 Emule 的 Kademlia 结构化覆盖网络进行分析，介绍了 Hash 算法的功能与原理。

# 3 P2P 环境下的内容分发机制

## 3.1 内容分发机制的设计思想

在传统的内容分发网络（CDN）中的内容分发机制的主要原理是：首先用户向中心服务器发送请求，中心服务器把请求发向全局负载均衡中心，然后全局负载均衡中心根据负载均衡算法得到最优的地区服务器，然后由地区服务器向用户提供服务。传统的 CDN 网络是一个分层的网络结构，而 P2P 网络是一个分布式的对等网络。因此不能把 CDN 网络下的内容分发机制一成不变地用到本系统的内容分发机制中。根据 P2P 的网络特点，本系统的内容分发机制的核心思想如下：首先资源请求者通过资源查找功能[33−34]获得拥有该资源的所有资源提供者的信息，然后根据负载均衡算法从中得到最优的资源提供者子集；把这个子集中的资源提供者看成是独立的对等服务器，然后请求者分别向各个服务器发出资源请求信号，服务器接到请求信号后，根据文件分块选择算法选出合适的文件分块，并向请求者发送分块。

## 3.2 分发方式

内容分发方式的形式主要有三种：推方式、拉方式和混合式，其中混合式是指推方式和拉方式的结合。

推方式是一种主动分发的技术。通常，推方式是由资源拥有者发起，将内容从源或者中心媒体资源库分发到各个资源请求者。对于推方式分发需要考虑的主要问题是分发策略，即在什么时候分发什么内容。一般来说，内容分发策略主要分为静态策略和动态策略。静态策略是指分发内容可以由内容拥有者或者管理员人工确定，而动态策略是指通过智能的方式决定，即所谓的智能分发，它根据用户访问的统计信息，以及预定义的内容分发的规则，确定内容分发的过程。

拉方式是一种被动的分发技术，它通常由用户请求驱动。当用户需要某资源，而该资源不存在时，用户就可以启动拉方式从资源拥有者或者其他拥有部分资源的

节点实时获取内容。在拉方式下，内容的分发是按需的。

在实际情况中，推方式和拉方式都存在着或多或少的缺点。推方式对分发策略性能要求太高，甚至可以说分发策略性能决定着推方式性能。静态的分发策略太死板，无法适应网络的变化；而动态的分发策略需要面对各种情况，无法做到面面俱到。拉方式是在明确知道向哪个资源拥有者发出相应的资源请求，但是一般来说对于一个资源只能向众多的资源拥有者的其中一个发出请求，但是网络的状况是实时变化的，之前最优的被请求者可能在下一刻变成最差的。

因此在实际的内容分发系统中，一般两种分发方式都支持，但是根据内容的类型和业务模式的不同，在选择主要的内容分发方式时会有所不同。在本系统中，分发方式也是采用混合式。整个资源采用的是拉方式，而对于资源各个细分部分则采用推方式，其中推方式的分发策略采用动态策略。具体的内容在文件分块选择算法以及负载均衡算法中进行说明。

## 3.3 文件分块选择算法

随着计算机硬件以及软件技术的发展，文件的大小也变得越来越大，特别是对于多媒体文件来说，动辄就是接近 1GB 的大小。对于如此大容量的文件，分发系统如果以整个文件来作为一个单位进行分发，那么性能可想而知是非常低下的。因此分发系统必须对文件进行分块。另外分块还有一个好处就是如果保留了每一分块的 hash 值，就能在只下载到文件的一部分时判断出下载内容的有效性。本系统的文件分块采用开源软件 emule 的文件分块原则。

如图 3-1，在文件分块系统中，文件的分块有两层：第一层是把整个文件分成 n 个部分（part），每个部分的大小都为 9500KB（除了最后的一部分）；第二层则是把每个部分划分为 52 个块（block），除最后的一块为 140KB 以外，每块的大小为 180KB。其中块是作为网络传输的最小单位。而采用两层分块模式的原因是：采用一层模式的系统只能以块（180KB）为单位进行分块选择，如此每次发送完一块以后都要重新进行分块选择，分块选择过多会导致性能下降；而采用两层模式时，可以以部分为单位进行分块选择，这样可以适当的减少分块选择次数。

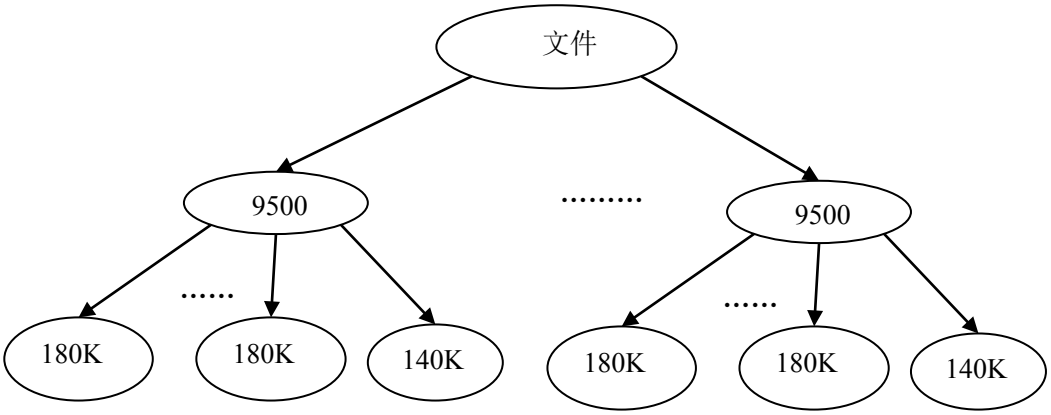


图 3-1 文件两层分块模型图

### 3.3.1 算法目标

文件分块选择算法的主要目标是：

（1）尽可能使各个下载者所获得的文件分块之间的差异性最大化。这样可以避免下载集中在某一部分，而其他部分却无人下载；还可以促进各个下载者之间分享资源。

（2）算法具有可行性，并可以有效快速的运行。由于算法的调用比较频繁，故有效高速应该是一个重要的标准。

### 3.3.2 分块选择算法设计

系统采用的是开源软件 Emule 的两层文件分块模式，因此分块选择算法[35]也应该分为两层：部分（part）选择算法和块（block）选择算法。其中部分选择算法主要功能是从文件的 n 个部分中选择合适的部分作为下一次传输的部分，只有在上一个部分传输完成以后，或者该部分已拥有的部分已经全部传输完成以后，才能启动下一次的部分选择算法；块选择算法的功能是从部分选择算法已得到的部分中选择合适的块进行分发。

（1）部分（part）选择算法

简单的循环分块选择

虽然可以把分块请求平均的分布到各个部分中，但是该方法属于静态方法，无法应对某些特殊情况（譬如很多节点缺少相同的某一部分时，无法快速的把该块分发给这些节点）。而采用单纯的随机分块选择，则会导致太多的未知情况，难以对性能进行估计。同时随机分块选择对随机算法也有要求，而现实中的随机算法只是一个伪随机算法，并没有真正做到随机。从小节 0 的算法目标设定可以看到，本系统采用以上的两种算法都不合适，因此本系统采用了能够适应大多数情况的优先级算法。

优先级分块选择算法的主要设计思想是：对于每一个请求者，所请求的文件的每个部分都有着相应的优先级（即优先级对于不同的请求者来说可能是不一样的）；首先对文件的每个部分（part）进行优先级计算，从中找出拥有最大优先级的部分（part），该部分就是要选择的分块所在的部分。而算法的重点在于优先级的设定与计算。根据设计思想，优先级的计算参数具有多个方面，某些与文件的全局信息有关，某些则与请求者信息有关。综合起来，优先级主要考虑以下的几个方面：

1）该部分在其他节点的分布情况，即该部分在其他的节点的拥有数目。如果数目比较多，则表示该部分有更多的机会从其他的节点获得；反之，则比较难获得。

2）该部分的完成度，即该部分中本节点已经下载完的块所占的百分比。完成度越高，意味着该部分作为上传部分的效果更好。

3）该部分是否已经被发送到请求者。如果该部分已经被发送到了请求者，那表示完全没有必要再次发送。

4）该部分的正在上传状况，即获得本节点该部分上传的接受者的数目。数目越大，代表着差异性越小。

假设要计算优先级的部分为第 i 部分，记为 $Part_i$，设该部分接受者为 j，则 $Part_i$ 对于接受者 j 的优先级为 $pro_{ij}$。$Part_i$ 在其他节点的分布数目为 $num_i$，完成度为 $finish_i$，正在获得上传的接受者数目为 $upload_i$，而接受者 j 上次已经获得该部分上传后所剩余没有得到上传的块数为 $uploaded_{ij}$。其中 $num_i$、$upload_i$ 和 $uploaded_{ij}$ 用整数表示，$finish_i$ 用百分比表达。根据上述的 4 个原则，$pro_{ij}$ 与 $num_i$ 和 $upload_i$ 成反比，

与 $finish_i$ 和 $uploaded_{ij}$ 成正比。由此可见优先级的计算公式为：

$$pro_{ij} = uploaded_{ij} * finish_i * (\max - num_i) / upload_i \qquad 3\text{-}1$$

其中 max 表示该文件各个部分中的最大上传次数加上一。这样可以避免：文件首次共享后接受到的首次请求时，max 为零的情况；以及出现 $\max - num_i$ 为零的情况。而且 $upload_i$ 不能为零值，即 $upload_i$ 的值应该为该部分已有的接受上传者数加上一。

（2）块（block）选择算法

当部分选择算法结束以后，就进入到块选择算法。块选择算法不采用优先级的方式，而是采用简单的连续选择算法：第一次发送该部分时，从第一块开始连续发送五块，如果接受端已经全部拥有这五块，则返回该部分第一个没有拥有的块，下一次发送从该块开始连续发送五块，然后接着发送下五块；如果连续三次发送了接受端已经拥有的块，或者该部分已经下载完成，则接受端应该返回下一部分的请求通知。

## 3.4 负载均衡算法

在 P2P 网络中，拥有同一资源的各个节点之间存在着不同的硬件与软件差异，而且不同节点之间的网络状况总是实时变化的。这些差异会导致节点提供服务的性能出现差异。每个资源请求者都渴望得到更好更快的服务，这就导致竞争的出现。同时由于这些竞争，高性能的服务者由于获得太多的请求，导致高负载从而变成低性能，而低性能则因为低负载变为高性能。如此如何动态获得高性能服务者就成为一个问题。这个问题的解决方法就是使用负载均衡。

### 3.4.1 算法目标

与大多数使用负载均衡算法的应用软件的目标一样，本系统的负载均衡算法的目标如下：

（1）解决网络拥塞问题，服务就近提供，实现地理位置无关。

（2）为资源请求者提高更好的服务质量。

（3）提高服务提供者的响应速度。

### 3.4.2 负载均衡算法设计

现今对负载均衡算法[36-40]的研究已经比较深入，特别是在传统的内容分发网络（CDN）中的负载均衡算法已经很成熟了。这些负载均衡算法一般都是基于集中式的分层服务器的负载均衡而设计的。而本系统的网络环境是分布式的结构化的 P2P 环境——Kademlia，因此本系统的负载均衡算法不能采用传统内容分发网络的负载均衡算法，而应该充分考虑 Kademlia 网络的特点，并借鉴已有的负载均衡算法来设计新的负载均衡算法。

在 Kademlia 网络中，由于其分布式的特点，对于拥有被请求资源的节点来说，它们就是服务器，但这些服务器并没有主次或者分层的概念存在，因为它们都是平等的。它们之间的区别只在于本身性能以及网络环境的差别。

本系统所采用的负载均衡算法为动态负载均衡，需要考虑的因素主要有：

（1）资源提供者与请求者之间的逻辑距离；

（2）资源提供者与请求者之间的网络状况；

（3）资源提供者的 CPU 状况；

（4）资源提供者的内存状况。

假设资源请求者所请求的文件为 $file_i$（$0 < i \le m$，m 为该节点所请求的文件数目），在资源查找中获得的资源提供者 $provider_j$（$1 \le j \le n$，n 为资源提供者的数目），则对于 $file_i$， $provider_j$ 的优先级为 $pri_{ij}$。 $provider_j$ 与请求者之间的逻辑距离为 $distance_j$，与请求者之间的网络状况为 $net_j$，CPU 的状况为 $cpu_j$，内存状况为 $mem_j$。则优先级 $pri_{ij}$ 的计算公式为：

$$pri_{ij} = cpu_j * mem_j / (distance_j * net_j) \quad (3\text{-}2)$$

$$cpu_j = cpuFre_j * cpuUse_j \quad (3\text{-}3)$$

$$mem_j = memSize_j * memUse_j \quad (3\text{-}4)$$

$$distance_j = IP_j \oplus IP_{local} \quad (3\text{-}5)$$

$$net_j = (time_{return} - time_{start}) / 2 \qquad 3\text{-}6$$

其中 $cpuFre_j$ 为 $provider_j$ 的 CPU 频率， $cpuUse_j$ 为 $provider_j$ 的 CPU 利用率；$memSize_j$ 为 $provider_j$ 的内存容量， $memUse_j$ 为 $provider_j$ 的内存利用率； $IP_j$ 为 $provider_j$ 的 IP 地址， $IP_{local}$ 为资源请求者的 IP 地址； $time_{start}$ 表示资源请求者向 $provider_j$ 发送网络状况访问包的开始时间， $time_{return}$ 为返回时间。

当资源请求者通过资源搜索获得资源提供者 $provider_j$ 的相关信息（$IP_j$、端口等相关信息）以后，请求者向提供者 $provider_j$ 发送网络状况访问包，并从中获得 $cpuFre_j$、$cpuUse_j$、 $memSize_j$、 $memUse_j$、 $time_{start}$ 和 $time_{return}$，然后根据公式 3-2 到 3-6 计算出 $pri_{ij}$。根据计算出来的 $pri_{ij}$ 对各个提供者进行排序，请求者向前十个提供者发出资源请求命令。如果提供者数量太少，则向所有的提供者发出请求命令。

## 3.5 本章小结

本章主要介绍了 P2P 环境下的内容分发机制的设计思想，确定了系统的分发方式为混合式，设计了关于内容分发机制的两个重要算法：两层的文件分块选择算法以及负载均衡算法。