

中国科学技术大学

博士学位论文

P2P系统中资源管理机制的研究

姓名：王菁

申请学位级别：博士

专业：计算机系统结构

指导教师：杨寿保

20070401

# 摘要

在过去的几年中，P2P系统规模不断扩大，应用不断增长。高效的资源管理机制成为P2P系统的关键技术。目前，P2P叠加网络拓扑的构建大多数没有考虑底层物理网络的实际情况，产生了叠加网络拓扑和底层物理网络不匹配的问题，导致了P2P节点定位资源的延迟和开销增加，严重影响了系统的效率，而P2P网络的动态性和匿名性又无法保证所有的响应节点都提供诚实良好的服务和可靠资源。同时，结构化P2P系统在提高良好可扩展性的同时存在的最大问题是仅限于精确关键字的匹配查询，而不支持基于内容的复杂查询。这些问题严重阻碍了P2P系统的迅猛发展。如何减少搜索成本，降低定位延迟，并提供支持基于内容查询、安全有效的P2P系统成为当前P2P研究领域的重要课题。

P2P系统的资源管理机制分为物理传输层、P2P叠加层、文件管理层以及检索模型四个部分，本文的研究重点为后三个部分。P2P叠加层，负责P2P协议的构建和优化。目前P2P系统中存在搜索成本高，定位延迟大以及由匿名性带来的安全隐患等问题，如何构建一个快速、高效、安全的叠加网络成为研究的新热点。文件管理层，负责管理数据仓库，按照关键字集合对文件进行有效地分类，同时提供文件到P2P叠加层的合理映射。实现从语义层到P2P层映射的最关键问题就是维数不匹配问题，因此采用有效的降维算法需要进一步的深入研究。检索模型，负责文本的插入和删除操作，并负责计算文本之间的相似性。“维灾”和高维稀疏矩阵是文本处理过程中急待解决的主要问题。本文从P2P资源管理机制的三个方面入手，针对上述问题展开深入研究，主要取得如下研究成果。

针对目前大多数P2P系统的构建没有考虑到物理因素，提出了基于延迟敏感的双向查询Chord系统，该系统根据节点和参考点之间的延迟对节点进行分组，减少平均相应时间；通过实现Chord上的双向查询，缩短搜索路径长度。

针对P2P系统中由节点匿名性所带来的恶意欺诈服务问题，提出了基于信誉感知的资源发现算法，基于Directed-BFS算法，融入信誉的概念，有效的抑制了P2P网络中恶意节点的欺诈行为，降低了系统的消息负载，保证了节点获得服务的可靠性和安全性。

为了减少女巫攻击对于P2P系统的影响，利用P2P的小世界特性，通过引入虚拟节点，依据最短增广路径算法寻找系统中的女巫攻击团体，有效地降低了女巫攻击节点在整个系统中的百分比。

针对当前结构化P2P文件存储系统存在仅支持单关键字的精确匹配，而缺乏支持内容查询的局限性，本文提出一种基于快速映射的结构化P2P存储系统FCAN。FCAN利用FastMap实现从语义空间到P2P空间的映射，在保证文本矢量语义完整性的同时对其进行有效降维，从而实现基于内容的查找；通过对质心法确定的最近邻居进行聚类，有效地减少了冗余信息和查询时间。

对于高维语义空间引起的文本相似度计算复杂,以及完全分布式非结构化P2P系统中广播算法带来的大量冗余消息,本文提出利用集合差异度实现基于内容聚类的P2P搜索模型。该模型利用集合差异度定义文本相似度,将文本相似性的计算复杂度控制在线性时间内而有效地减少了查询时间;利用节点之间的集合差异度实现基于内容的聚类,可以降低查询时间,减少冗余消息。

模拟试验表明,本文提出的资源管理机制可以有效减少P2P系统中的搜索成本,降低资源定位延迟,并且在保证安全性的情况下提供基于内容的有效查询。

关键词: P2P, 延迟敏感, FastMap, 向量空间模型, 质心法, 集合差异度, 女巫攻击, 小世界模型

# Abstract

The past several years witness the rapid development of P2P systems. Efficient resource management scheme becomes the key technology of P2P systems. On one hand, the fact that the construction of P2P overlay network doesn't consider the real network situation of physical layer leads to the mismatch of overlay layer and physical layer, which makes the latency and the cost of locating resources increase. At the same time, the characteristics of dynamics and anonymity of P2P can't ensure that all the peers will provide honest services and reliable resources. On the other hand, although structured P2P systems have good scalability, the main issue exists in them is that they are stricted to exact match and don't support content based query. All these problems prohibit the rapid development of P2P systems. How to reduce searching cost and decrease the latency of locating resources and how to construct a safe efficient P2P system that supports content based query becomes an important subject of the P2P system research field.

Resource management scheme in P2P system has four componets: physical transport layer, P2P overlay network layer, file management layer and information retrieval model. The main topics in this dissertation are the last three parts. P2P overlay network layer focuses on the construction and optimization of P2P protocols. Currently, there exist many problems in P2P system, which are high searching cost, long locating latency and hidden troubles that are brought by anonymity. Then, how to construct a fast, efficient and safe overlay network is challenging. File management layer aims at managing data depository, dividing the files efficiently according to keywords set and providing efficient mapping from file to P2P overlay network layer. The main issue to implement mapping from semantic layer to P2P layer is the mismatch of dimensions. How to reduce dimensions still needs further investigation. Information retrieval model mainly deals with the insertion and deletion of files and calculates the similarity between documents. The curse of dimensionality and the high dimension sparse matrix are the main issues to resolve when dealing with the documents. Detailed research works have been done on the above issues. The main contributions are as follows.

As current construction of most P2P systems doesn't take the physical elements into considerations, a bidirectional Chord system based on latency-sensitivity is proposed to decrease the latency and cost of resources locating. This system partitions the nodes according to the latencies among these nodes and the landmarks to decrease the average response time. And by the means of bidirectional query algorithm on Chord, the search path length is reduced.

To the problem of hostile service brought by node anonymity in P2P system, this paper proposed a reputation aware resource finding algorithm. This algorithm is based on Directed-BFS algorithm and

adopts the conception of reputation to efficiently restrain the deceiving behavior of hostile nodes in P2P system and reduce the messages load in the system, which can ensure the reliability and safety of the services.

To reduce the affection of Sybil attack on P2P system, a Sybil defense mechanism based on small world is proposed. Inducing the difficulty of finding Sybil group to max-flow/min-cut problem, this dissertation uses crawler to find Sybil group by adopting an artificial vertex, which can effectively isolate the Sybil attacker nodes from P2P system.

To solve the problem of existing DHT-based file storage systems being restricted to exact match and lacking semantic-based query, a DHT-based file storage system named FCAN is proposed in this paper. Mapping a document vector from semantic space to P2P space by FastMap, FCAN can effectively reduce the dimensions of document vector to implement semantic-based query without affecting the semantic integrality of this document. By clustering the nearest neighbors based on centroid, FCAN is further improved to reduce the redundant messages and searching time.

As the main issues that affect query efficiency and searching cost in content-based P2P search system are the complexity of computing the similarity of the documents brought by high dimensions and the great deal of redundant messages coming with flooding. This dissertation defines the document similarity by the way of set distance. This method restrains the complexity of computing the document similarity in linear time. Also, this dissertation clusters the peers based on content by their set distance to reduce the query time and redundant messages.

Simulation results show that the resource management schemes presented in this dissertation not only reduce the searching cost and decrease the latency to locate resources in P2P system effectively, but also provide content-based query and ensure the safety of the whole system.

**Keywords:** P2P, latency-aware, FastMap, Vector Space Model, Centroid, Distance, Sybil attack, small world mode

# 图表目录

图1.1 P2P系统的分类 .....	6
图1.2 Napster模型 .....	6
图1.3 Gnutella模型 .....	7
表1.1 结构化P2P系统与非结构化P2P系统的优缺点 .....	7
图1.4 论文结构图 .....	10
表2.1 Gnutella协议的消息类型 .....	13
表2.2 Gnutella协议的消息头格式 .....	13
图2.1 一个简单的P2P叠加网络拓扑图 .....	14
图2.2 节点a定位请求的过程 .....	15
图2.3 Chord结构图 .....	19
表2.3 Chord节点指针表中各项的含义 .....	20
图2.4 Chord的数据组织 .....	21
图2.5 CAN坐标空间的区域划分 .....	23
图2.6 CAN的路由算法图 .....	23
图2.7 一个pSearch查询的例子 .....	27
图2.8 SSW-1D的降维图 .....	28
图3.1 基于网络延迟的分组 .....	31
图3.2 基于网络延迟的分组 .....	32
图3.3 单向Chord系统中节点N1的指针表 .....	33
表3.1 节点n的前驱指针表中变量的定义 .....	34
图3.4 双向查询Chord系统中节点N1的指针表 .....	35
图3.5 双向查询Chord算法 .....	35
图3.6 随机分布的系统和基于延迟敏感的系统比较 .....	37
图3.7 Chord与双向查询Chord比较 .....	38
图3.8 两个系统的比较 .....	39
图4.1 洪泛算法 .....	41
图4.2 多路径推荐 .....	43
图4.3 扩展的P2P结构图 .....	43
图4.4 节点数据结构 .....	44
图4.5 算法流程图 .....	45
图4.6 RARDA的伪代码 .....	46
图4.8 信任节点比率 .....	47
图4.9 欺诈节点比例对成功下载文件的影响 .....	47

图4.10 资源搜索成功率 .....	48
图4.11 系统消息负载比率 .....	48
图5.1 分布式系统模型 .....	50
图5.2 基于最小割的女巫攻击团体 .....	52
图5.3 女巫防御系统中爬行器的工作原理图 .....	53
图5.4 引入虚拟节点的P2P系统 .....	54
图5.5 寻找女巫攻击团体的爬行器算法 .....	54
表5.1 基于小世界模型的女巫攻击防御系统中参数的说明 .....	55
图5.6 无防御系统时女巫攻击的节点数目与时间的关系 .....	56
图5.7 有防御系统时女巫攻击的节点数目与时间的关系 .....	56
图6.1 二维空间上的CAN结构 .....	59
图6.2 VSM模型以及相似度表示 .....	60
图6.3 FastMap的k维空间映射图 .....	61
图6.4 FCAN的映射算法 .....	62
图6.5 FCAN的基本搜索算法 .....	63
图6.6 最近邻居选择算法 .....	64
图6.7 基于质心法的d领域内最近邻居 .....	64
图6.8 超级节点连接的FCAN聚类结构图 .....	65
表6.1 FCAN系统参数 .....	66
图6.9 FCAN性能分析 .....	67
图6.10 SSW与FCAN平均路径长度比较 .....	68
图6.11 平均访问节点数比较 .....	68
图7.1 超级节点连接的基于内容的搜索模型 .....	72
图7.2 基于聚类特征向量的结构图 .....	72
图7.3 超级节点的倒排表 .....	74
表7.1 基于内容聚类搜索模型的试验参数 .....	75
图7.4 基于集合差异度的内容搜索模型与K-漫步者的平均召回率比较 .....	75
图7.5 不同TTL对平均召回率的影响 .....	76
图7.6 基于集合差异度的内容搜索模型查询效率分析图 .....	77

## 中国科学技术大学学位论文相关声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

本人授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

保密的学位论文在解密后也遵守此规定。

作者签名： 王菁

2007 年 4 月 28 日



# 第一章 绪 论

互联网系统的计算模式正在发生从客户机/服务器(client/server)模式到对等网络(Peer-to-Peer, 简称P2P)模式的转变<sup>[1]</sup>。对等网络的核心思想是所有参与系统的节点(指互联网上的某个计算机)处于完全对等的地位, 没有客户机和服务器之分, 也可以说每个节点既是客户机, 也是服务器; 既向别人提供服务, 也享受来自别人的服务。在P2P系统中, 资源是分布在各个对等节点上的, 而不是保存在集中的服务器上, 因此如何有效地管理P2P系统中的资源并对其进行快速定位成为P2P系统的关键技术。规模的增长对P2P系统中资源的定位不断提出更高的要求。资源管理机制成为影响P2P系统继续发展的重要因素。研究P2P系统中资源管理机制的关键技术, 提高P2P的资源定位性能, 成为一个重要的研究课题。

## 1.1 课题背景

互联网自产生以来一直以飞快的速度发展。首先, 网络规模日益扩大——目前接入互联网的个人计算机数量已经达到六十多亿<sup>[2]</sup>, 并且网络规模还在不断增长, 2004年增长率约为36%<sup>[3]</sup>。其次, 个人计算机的性能持续增强——主流微机的硬盘存储容量至少可以达到80GB以上, 微处理器时钟频率可以达到2GHz以上。按照摩尔定律预测, 微处理器性能每18个月可以翻一倍。第三, 网络带宽不断增大——带宽接入技术的应用使微机的网络带宽能达到512K以上。吉尔德定律预测未来25年内, 主干网的带宽将每6个月增加一倍。互联网的迅速发展给传统的网络应用模式带来了新的挑战, 促进了P2P系统的产生。

### 1.1.1 对等网络的产生

传统的互联网应用模式是基于客户机/服务器(Client/Server, 简称C/S)的。在这种应用模式下, 网络边缘的个人计算机处于客户机的地位, 不具备服务器的功能。随着个人计算机数目的增加, 服务器的负载过重, 难以满足客户机的服务请求; 同时个人计算机的性能增强, 已经具备小型服务器的能力, 但在传统的应用模式下只能处于客户机地位, 导致可用资源的闲置。这些空闲资源的潜力是巨大的, 假设个人计算机数目为1亿台, 每台个人计算机只要提供10MB的空闲存储空间, 用于存储的总空闲资源就能达到1000TB。一方面, 处在网络中心的服务器不堪重负; 而另一方面, 网络边缘却存在大量的空闲资源, 网络负载极不平衡。网络带宽的增长使得个人计算机之间具备直接通信的能力, 用户也希望能够不通过服务器直接进行资源的共享和交换。由此, 采用新型应用模式的P2P系统产生了。

P2P应用模式与C/S应用模式相比优势在于: 第一, P2P利用空闲资源降低了资源共享的开销; 第二, P2P极小化或无需集中控制提高了共享资源节点的自治性和系统的鲁棒性; 第三, P2P可以分散资源, 平衡网络负载。

### 1.1.2 对等网络的定义

从产生到现在,许多研究人员都试图给出P2P系统的定义,但是无论在学术界还是在工业界都没有一个统一的定义,下面理解几个常用定义。

英特尔公司P2P工作组(P2P Working Group)将P2P系统定义为通过在系统之间直接进行交换来共享计算机资源和服务的系统。

Schollmeier对P2P系统的定义如下<sup>[4]</sup>:如果一个系统的参与共享了自己的硬件资源,这些共享的资源是系统提供服务必需的,且可以被其他参与者不用通过中介而直接访问;系统的参与者既是资源的提供者也是资源的请求者,则此系统可以认为是P2P系统。

Graham通过描述P2P节点所需的三个必备条件来定义P2P系统<sup>[5]</sup>。三个必备条件如下:  
(1)具备服务器的能力;(2)拥有独立于DNS的寻址系统;(3)能够处理可变的连接。

Aberer通过描述P2P系统的特征来定义P2P系统<sup>[6]</sup>。这些特征如下:(1)没有集中的协调中心;(2)没有集中的数据库;(3)P2P节点没有整个系统的全局视图;(4)全局的行为依靠局部的相互作用;(5)所有存在的数据和服务都是可访问的;(6)P2P节点是自治的;(7)P2P节点及其相互之间的连接都是不可靠的。

Shirkey将P2P系统定义为一种利用互联网边缘的各种可用资源(如存储空间、计算能力、媒体能力、人力资源等)的应用程序<sup>[7]</sup>。因为访问这些分散的边缘资源意味着要在连接不稳定和IP地址不可预见的环境下工作,所以P2P节点必须能够独立于DNS系统且拥有独立于集中服务器的完全的自治。

惠普实验室(Hewlett-Packard Lab)的Milojicic将P2P系统定义为一类采取分布式方式利用分布式资源完成关键功能的系统<sup>[8]</sup>。分布式资源包括计算能力、存储空间、数据、网络带宽以及各种存在的可用资源。关键功能可以是分布式计算、数据内容共享、通信与协作或平台服务。分布式的方式可以应用到算法、数据、元数据或所有方面,但并不排除在系统或应用程序的某些部分保留集中式的方式。典型的P2P系统主要应用在互联网边缘或AdHoc网络环境中。

由于P2P系统采用了新兴网络应用模式,而且正在不断地发展变化,所以目前学术界仍然没有给出一个权威的定义,但这并没有妨碍P2P系统应用的迅速增长。

### 1.1.3 对等网络的特点

P2P技术的特点体现在以下几个方面。

**非中心化(Decentralization):**网络中的资源和服务分散在所有节点上,信息的传输和服务的实现都直接在节点之间进行,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。

**可扩展性:**在P2P网络中,随着用户的加入,不仅服务的要求增加了,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。

**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节点之间进行的,部分节点或网络遭到破坏对其他部分的影响很小。P2P网络一般在部分节点失效时能够自动调整整体拓扑,保持其他节点的连通性。P2P网络通常都是以自组织的方式建立起来的,并允许节点自由地加入和离开。P2P网络还能够根据网络带宽、节点数、负载等变化不断地做自适应的调整。

**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,个人计算机的计算能力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节点,将计算任务或存储资料分布到所有节点上。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。

**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。

**负载均衡:** 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。

### 1.1.4 对等网络的应用

在需求的推动下,P2P系统在文件共享、科学计算与协作、数据存储、数据搜索等多个领域得到了广泛的应用和研究,从而开辟了互联网应用的新时代。

#### (1) 文件共享

按照传统的文件共享模式,每个需要共享文件的计算机必须先把文件上载到集中的服务器上,而需要获取文件的计算机必须到服务器上下载所需的文件,这样才能实现个人计算机之间的文件共享。传统的方式不仅浪费了大量的服务器资源,而且存在单点失效的问题。利用P2P系统进行数据文件共享,可以让个人计算机用户之间不通过中心服务器而直接共享各种文件,用户直接到共享文件的计算机上去下载文件,这样不仅节约了资源,还提高了鲁棒性。

第一个P2P文件共享系统是1999年Fann开发的Napster<sup>[9]</sup>,它在发布不久便拥有了几百万用户,取得了极大的成功。此后,LimeWire<sup>[10]</sup>、Morepheus<sup>[11]</sup>、eDonkey<sup>[12]</sup>、iMesh<sup>[13]</sup>、

KaZaA<sup>[14]</sup>、BitTorrent<sup>[15]</sup>等P2P文件共享系统不断涌现。用户数量的持续增长和应用的迫切需求使得文件共享成为当前P2P系统中最主流的应用。

基于P2P的分布式Web缓存共享系统也得到了广泛的研究<sup>[16][17]</sup>，利用P2P技术，局域网内所有个人计算机能够共享本地缓存，中心代理服务器就无需缓存Web页面信息，而且所有的查询也无需通过中心的代理服务器发送到远程的服务器。这样不仅能节省费用，还能缩短请求的响应时间。

## (2) 科学计算与协作

在现代科学研究中，高能物理、大气、天文、生物信息和石油地质等许多重大科学领域正面临着巨大的挑战。这些领域的计算和存储要用Peta数量级来衡量。使用超级计算机存储和分析数据，不仅价格昂贵，还难以满足需求。同时单凭少数几个研究人员和几台计算机的传统科学研究方式已无法满足科研需求，需要全球各地的科研人员协作来完成。

P2P系统可以联接上百万台或更大规模的个人计算机，有效地利用处于网络边缘的空闲计算资源进行协同计算，完成超级计算机的工作。1999年开始的SETI@home<sup>[18]</sup>项目致力于搜寻外太空生命，到现在已经得到5,000,000以上用户的支持。2000年斯坦福大学开发的Folding@home<sup>[19]</sup>项目致力于研究蛋白质折叠、误折、聚合及由此引起的相关疾病，到目前为止已经吸引了400,000个用户加入。2003年Olson实验室主持的研究艾滋病的项目fightaids@home<sup>[20]</sup>也吸引到了9,020个用户。

英特尔公司研制的P2P分布式中间件NetBatch<sup>[21]</sup>使工程师能够在本地和全球的英特尔环境中寻找可用的计算能力，使计算机能够提高吞吐量、缩短运行时间并运行更复杂的工作，从而降低开发成本。企业级协同工作的P2P平台Groove<sup>[22]</sup>允许用户直接通信，为企业或是商务的应用提供支持。同时为了方便科学家之间的协作以及对大量研究数据的处理，P2P科学协作系统的研究工作也已经开始<sup>[23][24]</sup>。

## (3) 数据存储

当今社会处于一个信息爆炸的时代，需要大量的存储空间来存储消息。P2P技术允许数据分散存放在多个P2P节点上，而不是存放于专用服务器。这样不仅可以减轻服务器负担，节省大量开支，还可以提高数据存储的可靠性和传输的速度。

伯克利大学开发的分布式海量存储系统OceanStore<sup>[25]</sup>采用Tapestry<sup>[26]</sup>技术，提供了全球范围内的一致性数据存储。MIT研究的CFS项目采用Chord<sup>[27]</sup>技术，提供了一致性的分布协同文件存储。Rice大学和微软公司联合研究的PAST项目采用Pastry<sup>[28]</sup>技术，提供了大规模的、可扩展的、协同的分布文件存储服务。

Sun公司实验室和斯坦福大学图书馆联合研发的LOCKSS<sup>[29]</sup>项目是一个用于电子数字出版物的高度安全的系统。现在包括美国国会图书馆在内的49家图书馆正在采用LOCKSS软件。

## (4) 数据搜索

传统的搜索引擎采用集中式的体系结构,通过网络爬行器收集信息,在设计方面存在局限性,表现为数据不能动态跟踪、难以访问数据库喝动态Web页的信息、单点失效、可扩展性差以及资源浪费等。基于P2P的分布式搜索引擎可以克服这些缺点。首先,它通过对动态页面的访问使搜索范围更深更准确;其次,它采用的分布式结构避免了单点失效;最后,它可以充分利用网络边缘空闲资源,不必专门建立强大的服务器系统,从而节省资源。

2001年Sun公司开发的P2P分布式搜索引擎JXTASearch<sup>[30]</sup>可以基于某个元数据集合并进行分布式搜索。Miller提出的P2P分布式搜索引擎HyperBee<sup>[18]</sup>为网页建立数据库,利用互联网上数量庞大的个人计算机的运算能力,将每台个人计算机作为小型的搜索引擎搜索每个网站的信息。Suel讨论了P2P分布式搜索引擎的基本设计原则,并设计了一个原型系统ODISSEA<sup>[31]</sup>。

### (5) 实时通讯

基于P2P的通讯又可以分为两种,即时消息和实时游戏。即时消息(Instant Messaging,简称IM)在当今全球已经变得相当普遍。国外的ICQ、Yahoo Messenger、MSN Messenger以及国内的QQ等都已经吸引了大量用户使用。然后,目前IM软件还是基于C/S模型设计的,用户的帐户、好友列表都保存在Server上,甚至用户有时发出的消息也需要Server帮助转发。Serverless型的IM基本不需要Server的支持,只要人们以某种形式形成P2P网络互联,就可以相互之间识别并通讯,中间过程无需Server的帮助,比如Skype<sup>[32]</sup>。

网络游戏的发展速度同样惊人。然而与即时通讯应用相似,基于C/S模型的连线对战同样需要性能强劲的游戏服务器支持。P2P技术允许任何Peer可以单独建立区域性的P2P网络,可以让Internet上的任何人随时加入到其中,共同游戏娱乐。

## 1.2 问题的提出

### 1.2.1 对等网络的现状

文件共享已经成为当前P2P系统中最主流的应用,其发展速度非常迅速。1999年开发的Napster系统,在顶峰时期达到过6,000,000用户。LimeWire系统2005年统计的每天同时在线的用户达到105万<sup>[10]</sup>。Edonkey在线用户超过100万<sup>[12]</sup>。2005年2月19日,KaZaA文件共享系统软件下载累计达到384,191,334次<sup>[14]</sup>。文件共享系统规模还在以惊人的速度增长。

随着规模的扩大和应用的增长,P2P系统的流量在网络总流量中的所占的比例也急剧增长。根据威斯康星大学的统计,P2P应用所占的网络流量在2001年就达到总流量的30%,而Web应用的流量只占总流量的19%<sup>[33]</sup>。南加州大学在2002年统计发现P2P应用所占带宽为33.6%<sup>[33]</sup>。根据互联网数据分析合作协会(Cooperative Association for Internet Data Analysis,简称CAIDA)的统计,2002年各种P2P系统所占的总流量已超过美国城市骨干网的40%<sup>[34]</sup>。CAIDA还分别在2001年10月和2002年9月对MFN(Metromedia Fiber Network)

进行测量并统计各类应用程序所占带宽的百分比，结果发现P2P应用程序流量占总出入流量的百分比一年增长了6倍<sup>[34]</sup>。此后，虽然随着各种P2P系统软件的不断增加和端口号的可变性给P2P系统流量的测量带来困难，但2004年仍有报告证明P2P系统的流量还在不断增长，并没有下降的趋势<sup>[35]</sup>。为了进一步研究P2P系统的盛行对网络带宽造成的巨大影响，CAIDA还将分析P2P流量的工作列入其2007年的计划中。

1.2.2 对等网络的分类

目前P2P系统种类繁多，很多研究人员从不同的角度对它们进行分类<sup>[36][37]</sup>。本文从P2P发展的历程以及P2P系统的拓扑结构两个角度对P2P系统进行划分，如图1.1所示。

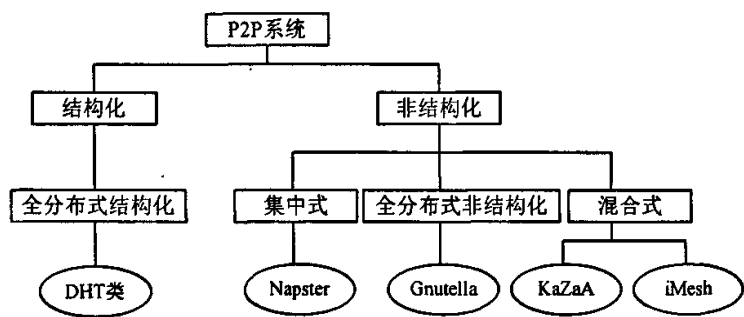


图1.1 P2P系统的分类

按照P2P发展的历程可以分为非结构化和结构化两种。非结构化P2P系统又分为集中式和完全分布式的，代表分别是Napster<sup>[9]</sup>和Gnutella<sup>[38]</sup>。Napster是最早出现的P2P系统之一，并在短期内迅速成长起来。Napster采用了集中式的目录服务器机制。目录服务器集中存放对等节点节点的地址信息和所保存数据的信息，可以对请求数据进行快速查找并能够返回最合适的目的节点。实际的文件传输将在请求节点和目的节点之间通过TCP连接直接进行，如图1.2所示。Napster首先实现了文件查询和文件传输的分离，有效地节省了中央服务器的带宽消耗，减少了系统的文件传输延时，但是存在单点失效的问题。

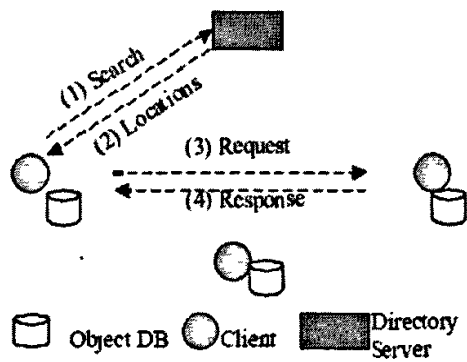


图1.2 Napster模型

和Napster不一样，Gnutella采用了完全分布式的策略，如图1.3所示。Gnutella无单点失效问题，可扩展性好，但是定位效率低。由于非结构化系统的不可扩展性，大量的研究集中在如何构造高度结构化的系统，结构化P2P系统应运而生。结构化P2P系统将每个资源定位在确定的节点上，提供了资源表示ID到资源所在位置的映射关系，从而保证在有限步数内定位到资源。结构化P2P系统以基于分布式哈希表(Distributed Hash Table, 简称DHT)的P2P系统为代表，如Chord<sup>[27]</sup>、Tapestry<sup>[28]</sup>、Pastry<sup>[28]</sup>、CAN<sup>[37]</sup>等。表1.1总结了结构化P2P系统和非结构P2P系统的优缺点。

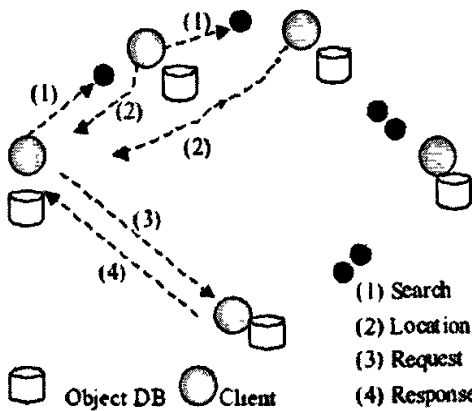


图1.3 Gnutella模型

表1.1 结构化P2P系统与非结构化P2P系统的优缺点

特 点	结构化P2P系统	非结构化P2P系统
资源定位	确定步数	不确定步数
可扩展性	好	较差
构建网络开销	大	小
关键字定位	精确匹配	模糊匹配
自治性	低	高
鲁棒性	低	高

拓扑结构是指分布式系统中各个计算单元之间的物理或逻辑的互联关系，节点之间的拓扑结构一直是确定系统类型的重要依据。根据拓扑结构的关系可以将P2P研究分为四种形式：中心化拓扑、全分布式非结构化拓扑、全分布式结构化拓扑和混合式拓扑。中心化拓扑的最大优点是维护简单发现效率高，但是对中心服务器的依赖比较大，一旦服务器失效，整个系统都会瘫痪，比如Napster。全分布式非结构化P2P系统采用了随机图的组织方式，节点度数服从“Power-law”<sup>[39]</sup>规律，从而能够较快发现目的节点，面对网络的动态变化体现了较好的容错能力，因此具有较好的可用性，并且支持负载查询，最典型的例子是Gnutella。Gnutella采用了基于完全随机图的洪泛算法和随机转发机制，为了控制搜索消息的传输，通

过TTL(Time To Live)的减值来实现。随着网络规模的扩大,这种基于洪泛算法的定位会导致网络中出现大量的冗余消息。对于全分布式结构化P2P系统的研究集中在采用新的拓扑图构建重叠路由网络,以减少路由表容量和路由延时。DHT类结构能够自适应节点的动态加入和退出,有良好的可扩展性和鲁棒性。混合式拓扑吸取了中心化拓扑和全分布式非结构化拓扑的优点,选择性能(处理、存储、带宽等方面性能)较高的节点作为超级节点,在各个超级节点上存储了系统中其他部分节点的信息,发现算法仅在超级节点之间进行转发。典型的系统有KaZaA系统<sup>[14]</sup>。

本文将分别选择结构化P2P文件共享系统和全分布式非结构化P2P系统作为对象,从安全性和基于内容查找两个方面研究P2P系统中的资源管理问题。

### 1.2.3 资源管理机制面临的问题

资源管理机制是P2P系统的关键技术,它包括资源的分配机制和资源的定位机制,其中前者决定了后者,并且最终为后者提供依据。资源管理机制的主要功能是返回请求资源的定位信息。如果资源管理机制差,将影响整个系统的运行。

目前存在的P2P叠加网络拓扑的构建大多数没有考虑底层物理网络的实际情况,产生了叠加网络拓扑和底层物理网络不匹配的问题,导致了P2P节点定位资源的延迟和开销增加,严重影响了系统的效率。

P2P网络的动态性和匿名性不能保证所有的响应节点都提供诚实良好的服务和可靠资源,并且P2P网络的自由开放性使得节点可随意创建身份加入系统,而引发女巫攻击。因此,在面向大量动态用户的匿名的P2P网络环境中,如何在资源搜索的过程中避免这些恶意节点的欺诈服务以及女巫攻击,保证服务提供节点提供可靠资源成为了一个亟需解决的问题。

结构化P2P系统具有较好的可扩展性和鲁棒性,并且提供了有效的节点加入和离开策略。但是结构化P2P系统的最大问题是仅限于精确关键字的匹配查询,而不支持基于内容的复杂查询,同时,结构化P2P系统所面临的另一个问题是维护机制较为复杂,尤其是节点频繁加入或退出所造成的网络波动。

全分布式非结构化P2P系统可以支持复杂查询,但是,它所采用的基于语义空间向量的相似度计算方法具有较高的时间复杂度,并且洪泛算法会消耗大量网络带宽,带来大量冗余消息。

低效的资源管理机制严重阻碍了系统规模的进一步扩大。如何设计P2P系统中高效的基于内容的资源管理机制成为P2P系统研究的重要方向。

### 1.2.4 P2P系统中资源管理机制的研究内容

P2P系统中的资源管理机制主要由四部分组成<sup>[40]</sup>:

- 物理传输层:负责解决两个节点之间的通讯。节点的IP地址和端口号可以被看作它在



在物理传输层标识符。由于IP地址的动态性,应该保持节点在上层的标识符和IP地址的一致性。

·P2P叠加层:负责构建P2P协议,可以分为结构化P2P系统和非结构化P2P系统。在P2P叠加层,每个节点都拥有逻辑命名空间中的唯一标识符,对于节点的查找算法依赖于所采用的P2P结构。把位于物理传输层上较近的节点组织在逻辑空间上相邻的范围内可以减少查询时间,并且优化的P2P结构也可以提高查询效率。P2P叠加层还负责整个系统的安全性问题,包括避免恶意节点的欺诈服务。

·文件管理层:负责管理数据仓库,按照关键字集合对文件进行有效地分类,同时提供文件到P2P叠加层的合理映射。文件和P2P中的节点共享同一个逻辑的命名空间,它们之间的有效映射算法可以提高查询效率。

·检索模型:负责文本的插入和删除操作,并负责计算文本之间的相似性。“维灾”和高维稀疏矩阵是文本处理过程中的主要问题,在有限维的P2P叠加层上实现基于内容的查询需要优化相似度的计算。

为了解决当前P2P系统资源管理机制中出现的问题,高效地定位资源,设计适应P2P系统特点的基于内容的查询机制,主要从P2P叠加层、文件管理层以及检索模型三部分技术入手。

## 1.3 论文工作

为了设计高效、安全的基于内容的资源管理机制,本文深入总结了前人在相关领域的研究成果,分别对P2P系统信息检索领域的三部分技术展开研究,主要研究工作如下:

1.全面深入的综述P2P系统中资源管理机制的相关技术。从资源管理机制的三个主要组成部分——P2P叠加层、文件管理层、检索模型出发,总结了相关研究工作中存在的主要问题,分析了未来研究工作发展的趋势。

2.研究P2P叠加层。针对目前大多数P2P系统的构建没有考虑到物理因素,提出了基于延迟敏感的双向查询Chord系统,该系统根据节点和参考点之间的延迟对节点进行分组,减少平均响应时间;通过实现Chord上的双向查询,缩短搜索路径长度。针对P2P系统中由节点匿名性所带来的恶意欺诈服务以及女巫攻击问题,分别提出了基于信誉感知的资源发现算法和基于小世界模型的女巫攻击防御机制。前者基于Directed-BFS算法,融入信誉的概念,有效地抑制了P2P网络中恶意节点的欺诈行为,降低了系统的消息负载,保证了节点获得服务的可靠性和安全性;后者利用P2P的小世界特性,通过引入虚拟节点,依据最短增广路径算法寻找系统中的女巫攻击团体,有效地降低了女巫攻击节点在整个系统中的百分比。

3.研究文件管理层。针对当前结构化P2P文件存储系统存在仅支持单关键字的精确匹配,而缺乏支持内容查询的局限性,本文提出一种基于快速映射的结构化P2P存储系统FCAN。FCAN利用FastMap实现从语义空间到P2P空间的映射,在保证文本矢量语义完整

性的同时对其进行有效降维，从而实现基于内容的查找；通过对质心法确定的最近邻居进行聚类，有效地减少了冗余信息和查询时间。

4.研究检索模型。对于高维语义空间引起的文本相似度计算复杂，以及完全分布式非结构化P2P系统中广播算法带来的大量冗余消息，本文提出利用集合差异度实现基于内容聚类的P2P搜索模型。该模型利用集合差异度定义文本相似度，将文本相似性的计算复杂度控制在线性时间内而有效地减少了查询时间；利用节点之间的集合差异度实现基于内容的聚类，可以降低查询时间，减少冗余消息。

本文的研究工作得到了国家自然科学基金项目——基于计算市场模型的网络资源管理研究（编号：60273041）和网络计算环境中信任感知的资源交易模型（编号：60673172），以及国家863项目——合肥网格节点的建设及若干典型网格应用的研制（编号：2002AA104560）的支持。

1.4 论文结构

论文共分七章，章节的基本关系如图1.2所示。

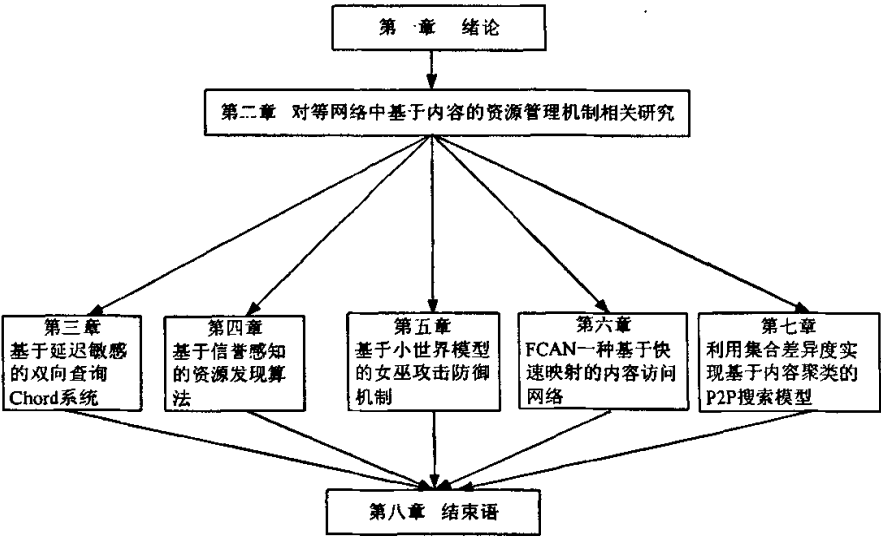


图1.4 论文结构图

第一章，绪论。介绍了P2P系统的发展及应用，阐述了P2P系统中基于内容的资源管理机制在P2P系统研究中的重要性，分析了P2P系统中基于内容的资源管理机制所面临的问题，简单介绍了本文的工作。

第二章，P2P系统中资源管理机制的相关研究。分析了P2P的构建与物理相关的重要性，回顾了P2P系统中资源定位机制的各种研究技术，从而明确本文的研究工作；提出从文本到P2P节点之间合理的映射方案；提出了利用集合差异度解决高维语义空间所引起的文本相似

度计算复杂性问题。

第三章，基于延迟敏感的P2P系统的构建。详细阐述了基于延迟敏感的双向查询Chord系统的设计与实现，并给出了性能分析和实验结果。

第四章，基于信誉感知的资源发现算法。基于Directed-BFS算法，融入信誉的概念，提出基于信誉感知的资源发现算法，有效地抑制了P2P网络中恶意节点的欺诈行为，降低了系统的消息负载，保证了节点获得服务的可靠性和安全性。

第五章，基于小世界模型构建女巫攻击防御机制。利用P2P系统的小世界特性，通过把寻找女巫攻击团体的问题转化为最大流最小割问题，引入虚拟节点，依据最短增广路径算法识别女巫攻击节点，降低了恶意节点在整个系统中所占的百分比。

第六章，基于快速映射的结构化P2P存储系统FCAN的构建。通过FastMap实现从语义空间到P2P空间的映射，在保证文本矢量语言完整性的同时对其进行有效降维，从而实现基于内容的查找。

第七章，利用集合差异度实现基于内容的查询和聚类。利用集合差异度定义文本相似度，将文本相似性的计算复杂度控制在线性时间内，利用集合差异度实现基于内容的聚类，减少冗余信息。

第八章，结束语。对本文工作的总结，以及对下一步工作的展望。

## 1.5 本章小结

本章首先介绍了对等网络的定义、特征、应用和分类，阐述了对等网络技术的发展以及研究意义。然后提出了对等网络中基于内容资源管理机制研究的重要性，分析了对等网络发展面临的挑战。最后给出了本文的研究内容和论文结构。

## 第二章 对等网络中资源管理机制的研究

由于P2P系统具有动态性的特点,资源管理机制的设计直接影响了P2P系统的性能,而研究P2P系统资源管理机制的重点是实现基于内容的查找以及保证P2P系统的安全性。P2P系统的资源管理机制主要由物理传输层、P2P叠加层、文件管理层以及检索模型四大部分组成。围绕这几个组成部分,目前已经展开了很多的研究。随着P2P系统规模的不断扩大,基于内容的资源管理机制的相关技术日益成为P2P系统研究的热点。本章将综述资源管理机制中P2P叠加层、文件管理层以及检索模型这三部分的相关研究工作,分析相关研究中存在的主要问题,从而明确本文的研究方向和内容。

### 2.1 引言

P2P系统中的资源管理机制负责资源的分配和资源的定位,其中前者决定了后者,并且最终为后者提供依据。资源管理机制的主要功能是返回请求资源的定位信息。综合前人的研究<sup>[39]</sup>,P2P系统中资源管理机制的三个主要组成部分是:

·P2P叠加层:负责构建P2P协议,可以分为结构化P2P系统和非结构化P2P系统。在P2P叠加层,每个节点都拥有逻辑命名空间中的唯一标识符,对于节点的查找算法依赖于所采用的P2P结构。把位于物理传输层上较近的节点组织在逻辑空间上相邻的范围内可以减少查询时间,并且优化的P2P结构也可以提高查询效率。P2P叠加层还负责整个系统的安全性问题,包括避免恶意节点的欺诈服务。

·文件管理层:负责管理数据仓库,按照关键字集合对文件进行有效地分类,同时提供文件到P2P叠加层的合理映射。文件和P2P中的节点共享同一个逻辑的命名空间,它们之间的有效映射算法可以提高查询效率。

·检索模型:负责文本的插入和删除操作,并负责计算文本之间的相似性。“维灾”和高维稀疏矩阵是文本处理过程中的主要问题,在有限维的P2P叠加层上实现基于内容的查询需要优化相似度的计算。

传统的P2P资源管理机制分为非结构化和结构化,其中全分布式非结构化P2P系统采用资源定位机制洪泛算法,并且实现了模糊查询。虽然实现简单,但不适应大规模P2P的特点,既给网络带宽带来严重的影响,也降低了资源搜索的性能;全分布式结构化P2P系统具有较好的可扩展性和鲁棒性,并且提供了有效的节点加入和离开策略。但是结构化P2P系统的最大问题是仅限于精确关键字的匹配查询,而不支持基于内容的复杂查询。研究人员围绕具有P2P系统特征的资源管理机制展开了大量研究。有的研究针对全分布式非结构化P2P系统的搜索机制进行优化,有的研究如何基于全分布式结构化P2P系统实现基于内容的查询。本章将综述这些相关工作。

本章各部分是这样组织的：2.2节首先分析传统洪泛资源定位机制的研究；2.3节总结传统结构化P2P系统中查询机制的研究；2.4节总结P2P系统中安全性问题的相关研究；2.5节总结结构化P2P系统中基于内容的查询机制；2.6节是小结。

## 2.2 传统洪泛资源定位机制的研究

传统的资源定位机制是基于洪泛算法的。采用洪泛算法传播消息无需节点保存其他节点的信息，实现最为简单，洪泛资源定位机制在P2P系统应用的初期得到广泛应用。Gnutella是实现洪泛资源定位机制的一种典型协议，它在P2P文件共享系统中得到广泛应用。本节将以Gnutella为例分析传统的洪泛资源定位机制。

### 2.2.1 传统洪泛资源定位机制的基本搜索技术

Gnutella协议主要包括叠加网络构建和定位请求信息处理两部分，表2.1给出了Gnutella协议定义的消息类型，其中Ping和Pong是构建类消息，用于构建叠加网络拓扑；Query和QueryHit是定位类消息，用于定位文件。各个消息的头格式如表2.2所示，其中的数字表示字节偏移量。

表2.1 Gnutella协议的消息类型

消 息	说 明
Ping	用于主动发现网络上的其他节点
Pong	Ping的响应消息，包含响应节点的地址以及其提供共享文件的信息
Query	用于定位资源的定位请求消息
QueryHit	Query的响应消息，提供请求文件的定位信息
Push	用于为防火墙后的节点提供共享文件

表2.2 Gnutella协议的消息头格式

MessageID	Payload Descriptor	TTL	Hops	Payload Length
0-15	16	7	18	19-22

Gnutella协议的叠加网络拓扑构建过程为：当一个节点要加入系统的时候，首先连接一些知名的节点，并向这些节点广播初始Ping消息。初始Ping消息携带初始的TTL值（通常为7）和全局唯一的ID。节点接收到Ping消息时，如果它以前接收过相同ID的消息，就不进行处理；否则，该节点将信息的TTL值减1，并转发此消息到它的所有邻居，同时沿原路回播包含其连接信息的Pong消息。直到所有Ping消息的TTL值等于0，整个过程才会结束。请求加入系统的节点收到Pong消息即可于这些节点连接。

Gnutella协议的定位请求消息处理的过程为：当用户需要定位某个文件时，向其所有邻居发送包含请求文件描述符的定位请求消息Query。转发请求消息的过程与洪泛构建算法相

似。当用户接收到Query消息时, 先将请求文件的描述符与本地共享文件索引进行匹配, 匹配成功发送包含请求文件定位消息的响应消息QueryHit。无论匹配成功与否, 只要TTL值不等于0就继续转发消息。

我们不难发现洪泛资源定位机制的叠加网络构建技术和定位请求消息处理技术都是基于洪泛算法的。采用洪泛算法可以快速传播消息, 且节点无需保存其他节点的信息, 是一种简单、易于实现的方法。洪泛算法构建的叠加网络拓扑图的特征与幂律图(Power-law)<sup>[38]</sup>和小世界模型(Small-world)<sup>[41]</sup>相似<sup>[42][43][44]</sup>。这种叠加网络拓扑图的直径比随机图的要小, 其资源定位时间较短; 但同时叠加网络拓扑图存在许多冗余路径, 因此会产生大量的冗余消息, 浪费大量的带宽。此外, 构建算法形成的P2P网络不能抵御有目的的攻击。针对大度数节点的攻击会使P2P节点形成多个互不相连的子网络<sup>[45]</sup>。

洪泛算法产生的消息的数目随着消息的传播成指数倍增长, 并包含大量冗余消息。Gnutella协议采用两种方法来控制冗余消息的数量, 以减少洪泛算法占用的带宽。

1. 消息生存时间(Time-To-Live, 简称TTL)。TTL是指消息在网络中传播时能够生存的时间, 它包含在消息头中, 在每个消息生成时被赋予一个初始值。当消息被发送出去后, 其他接收到此消息的节点将首先检查消息头中的TTL值, 如果大于0, 则将此消息的TTL值减1, 再将消息转发给它的所有邻居节点; 否则, 丢弃此消息。TTL值越大, 消息传播的距离就越远; 反之, 消息传播的距离就越近。

2. 消息全局唯一标识符(Unique Identification, 简称UID)。UID用于避免同一消息被同一节点重复传播, 它包含在消息头中, 不同消息的UID是不同的。每个节点维护一个UID列表。当消息发送到某个节点时, 该节点将取出消息头中的UID字段与本地UID列表进行比较。如果此消息的UID不在列表中, 则节点将此消息的UID保存到本地的UID列表, 并把消息转发给它的所有邻居节点; 否则, 直接丢弃此消息。

采用生存时间和全局唯一标识符只能避免部分消息的重复转发, 大量冗余消息仍将产生, 且并不能改变消息指数增长的趋势。

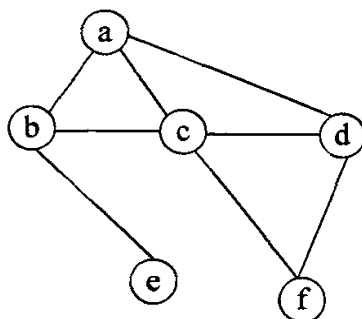


图2.1 一个简单的P2P叠加网络拓扑图

假设叠加网络拓扑图的节点平均度数为 $b$ , 消息的TTL值为7, 则粗略计算采用洪泛定位

机制进行一次文件定位请求产生的消息数为  $\sum_{i=1}^7 b^i$ 。若  $b$  等于 6，一次文件定位请求产生的消息数就达到 335,922 个，数量十分惊人。同时，由于叠加网络拓扑图存在许多冗余路径，因此在这些定位消息中又存在许多冗余消息，从而浪费了大量的带宽。举例说明，假设 P2P 节点构建成的叠加网络拓扑如图 2.1 所示，源请求节点  $a$  发送定位请求消息给节点  $b$ 、 $c$  和  $d$ 。这些节点接收到请求消息后先进行本地处理，再将请求消息转发给它们各自的邻居。整个定位请求过程如图 2.2 所示。

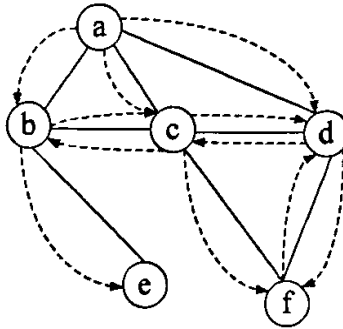


图 2.2 节点  $a$  定位请求的过程

图 2.2 表明节点  $c$  和  $d$  接收到了 3 个冗余消息，节点  $b$  和  $f$  接收到了 2 个冗余消息。整个过程共产生了 11 个请求消息，冗余消息就有 6 个。冗余消息浪费的带宽百分比高达 54.5%。

传统的洪泛定位机制虽然简单易实现，但其产生了大量冗余不必要的消息，占用并浪费了大量的带宽，因此十分低效并难以扩展。研究高效的资源定位机制势在必行。

## 2.2.2 洪泛资源定位机制中搜索技术的改进

为了减少资源定位占用的带宽，加快资源定位的延迟，研究人员主要在两个方面展开研究<sup>[52][53][54][55][56][57]</sup>，叠加网络拓扑的优化以及定位请求消息处理技术。其中前者的技术主要有：短直径的叠加网络拓扑、基于兴趣的叠加网络拓扑、基于能力的叠加网络拓扑、物理匹配的叠加网络拓扑以及可适应的叠加网络拓扑。后者主要分为盲目搜索(Blind Search)和有信息搜索(Informed Search)<sup>[58]</sup>。盲目搜索指节点随机选择所有邻居或邻居的子集来转发定位请求消息；而有信息搜索是指节点根据某种信息选择邻居的子集来转发定位请求消息。此外，这些研究技术再根据转发的方式又可以分为宽度优先搜索类(Breadth First Search-like，简称BFS-like)和深度优先搜索类(Depth First Search-like，简称DFS-like)。

### 2.2.2.1 叠加网络拓扑的优化

#### (1) 短直径的叠加网络拓扑

每个节点之间的距离越短，定位请求资源的延迟也就越短，因此构建短直径的叠加网络拓扑成为最直接的研究思路。

Pandurangan提出一种协议,将P2P节点构建成直径为 $O(\log N)$ 的叠加网络拓扑<sup>[59][60]</sup>,其协议简单,且易于网络规模的扩展,但协议需要集中式的服务器来协调工作,因此产生了单点失效问题。

Wouhaybi提出一种协议,将P2P节点构建成节点度数服从幂律分布的短直径叠加网络拓扑Phenix<sup>[61]</sup>,缩短了节点之间的距离。但是当幂律网络中的大度数节点受到攻击时,整个网络会分割成多个相互孤立的子网络,因此Wouhaybi还提出了节点受到攻击时的自恢复算法。

## (2) 基于兴趣的叠加网络拓扑

根据系统用户的爱好将相同兴趣的用户节点相连形成子网络,可以提高节点近距离定位资源的成功概率,从而减少节点定位资源的延迟。

Cohen认为存放相似文件的用户的兴趣相同,将存放相似文件的用户相连成一个组,称为相连叠加网络拓扑。模拟结果表明相连叠加网络拓扑能极大地提高用户定位稀少资源的能力<sup>[62][63]</sup>。

Ramathan提出评估“好”节点的方法<sup>[64]</sup>,并将“好”节点相互连接,以达到最小化搜索范围和最大可能地互联相似兴趣的节点的目的,但这种方法容易导致孤立节点的产生。

## (3) 基于能力的叠加网络拓扑

由于P2P节点的异构性,低访问带宽或低处理能力的小能力节点成为资源定位机制的性能瓶颈,影响了系统的可扩展能力,因此如何减轻小能力节点的负载,提高资源定位机制的性能也成为重要的研究方向。

ADP2PN模型<sup>[65]</sup>将原来骨干网络中的路由交换节点扩展成主动节点,并作为模型的簇节点。ADP2PN虽然提高了搜索效率但是对路由器节点和网络环境配置都有较高的要求。

Fastrack协议<sup>[66]</sup>和Gnutella0.6<sup>[67]</sup>协议采用超级节点叠加网络拓扑来减轻小能力节点的负载。

## (4) 物理匹配的叠加网络拓扑

P2P的叠加网络拓扑构建并没有考虑底层物理网络的实际情况,这就产生了叠加网络拓扑与底层物理网络不匹配的问题,导致P2P节点定位资源的延迟和开销增加。很多研究人员围绕这个问题展开了研究。

Padmanabhan利用IP地址判断P2P节点的物理拓扑,将距离相近的P2P节点连接在一起<sup>[68][69]</sup>,但这种方法的准确度不高,而且还会影响P2P节点的搜索范围。

Krishnamurthy提出网络相关簇技术<sup>[70][71]</sup>,利用BGP表的信息将底层拓扑相近的以及在同一自治域下的P2P节点连接成簇,但BGP表的信息不容易获取,给网络相关簇技术的实际应用带来难题。



### 2.2.2.2 盲目搜索

洪泛定位请求消息处理选择所有的邻居转发定位请求消息,产生了大量冗余和不必要的定位请求消息。实际上,定位某些资源并不需要查询所有的节点,减少处理定位请求消息节点的数目,既可以减少资源定位占用的带宽,又可以减轻节点的负载,因此如何减少处理定位请求消息节点的数目成为减少定位请求消息的主要方法。

#### (1) 宽度有限搜索类

Kalogeraki提出的Modified-BFS<sup>[72]</sup>是洪泛定位请求处理的改进,算法不是选择节点的所有邻居转发定位请求消息,而是随机选择邻居的子集进行转发。但如果只选择固定数目的邻居会出现局部覆盖问题(Partial Coverage Problem)。Zhuang提出了随机周期洪泛算法(Random Periodical Flooding)<sup>[73]</sup>,选择基于TTL值的可变数目的邻居转发定位请求消息。

Yang提出的迭代加深(Iterative Deepening)<sup>[74]</sup>算法根据用户需求逐个启动多个小于规定TTL值的洪泛搜索。因为定位请求消息的数目随着TTL值的增长呈指数增长,如果定位请求能在一个较小的TTL值内完成,那么定位请求消息的数目就会大量减少。否则,该算法反而会增加请求消息的数目。扩展环算法(Expanding Ring)<sup>[85]</sup>也与此类似。

Gnutella协议规定在超节点叠加网络中每个请求节点向所属超节点发送定位请求消息,每个接收到定位请求消息的超节点不仅查询自己维持的局部索引表,还向其他的超节点邻居广播定位请求消息<sup>[86]</sup>。

#### (2) 深度优先搜索类

Clarke利用深度优先搜索算法(DFS)<sup>[87]</sup>使得每个节点随机选择一个邻居转发请求消息。由于DFS每步产生的请求消息只有1个,总消息数为TTL个。因此算法定位资源占用的带宽很少。但是由于访问到的节点也最多只有TTL个,就很可能必须设置一个很大的TTL值来定位请求资源,从而增加了定位资源的延迟。

Lv提出的漫步算法(Random Walks)<sup>[85]</sup>是多个DFS算法的同步进行。源请求节点发送 $k$ 个定位请求消息,每个定位请求消息是一个漫步者(Walker)。每个漫步者到达一个P2P节点后,随机选择此节点的一个邻居节点进行访问。每个漫步者周期性地与源请求节点联系并询问节点需求是否已被满足,如果已满足,则停止;否则继续。增加漫步者的个数 $k$ (通常为16到64)就可以降低漫步算法定位资源的延迟。

### 2.2.2.3 有信息搜索

单纯地减少处理定位请求消息节点的数目会减少搜索资源的范围,从而导致定位某些资源的延迟增加,甚至定位不到请求的资源。因此,研究人员开始利用各种信息指导节点选择更有可能共享请求资源或找到资源的节点转发定位请求消息。

#### (1) 宽度优先搜索类

Yang提出定向宽度优先搜索算法(Directed BFS)<sup>[74]</sup>使节点根据统计信息选择最有可能

返回结果的邻居子集转发定位请求消息。Crespo提出的路由索引算法<sup>[88]</sup>与此类似,用户根据路由索引选择最有可能得到响应的邻居转发请求消息。Zhuang还针对随机周期洪泛算法提出了混合周期洪泛算法(Hybrid Periodical Flooding, 简称HPF)<sup>[73]</sup>根据多种度量标准选择可变数目的邻居转发定位请求消息。

Yang同时还提出局部索引(Local Indices)算法<sup>[74]</sup>。部分节点维持一个局部索引,包含距其 $\delta$ 跳步数以内的节点的共享资源索引。算法采取洪泛的方式传播定位请求消息,没有局部索引的节点只转发定位请求消息不进行本地请求处理,有局部索引的节点根据索引信息代表在离它 $\delta$ 跳步数范围内的每个节点处理请求。算法减少了每个节点处理请求的开销,但是并没有减少定位带宽开销,在动态变化频繁的环境下,更新消息的增多反而会加大定位的开销。

## (2) 深度优先搜索类

对于漫步算法,Adamic证明在具有幂律图特征的叠加网络拓扑中选择高度数的节点进行转发可以提高定位的性能<sup>[89]</sup>。Menasce提出的分布式资源定位协议(Distributed Resource Location Protocol, 简称DRLP)<sup>[90]</sup>,节点根据一定的概率将定位请求转发给每个邻居。Tsoumakos和Kumar分别根据各自的算法选择最有可能的邻居节点进行转发<sup>[91][92]</sup>,而Chawathe在其构建的叠加网络拓扑上使用定向漫步算法(Biased Random Walks)<sup>[93]</sup>,选择有最高能力的邻居节点转发定位请求消息。

## 2.3 结构化P2P系统中查询机制的研究

由于非结构化网络将重叠网络认为是一个完全随机图,节点之间的链路没有遵循某些预先定义的拓扑来构建。这些系统一般不提供性能保证,但容错性好,支持复杂的查询,并受节点频繁加入和退出系统的影响小,但是查询的结果可能不完全,查询速度慢,采用广播查询的系统对网络带宽的消耗非常大,并由此带来可扩展性差的问题。因此,大量的研究集中在如何构造一个高度结构化的系统。目前的研究重点放在了如何有效地查找信息上,最新的成果都基于分布式散列表(Distributed Hash Table, 简称DHT)的分布式发现和路由算法。

分布式散列表实际上是一个由广域范围大量节点共同维护的巨大散列表。散列表被分割成不连续的块,每个节点被分配给一个属于自己的散列块,并成为这个散列块的管理者。DHT的节点既是动态的节点,数量也是巨大的,因此非中心化和原子自组织成为两个设计的重要目标。通过加密散列函数,一个对象的名字或关键字被映射为128位或160位的散列值。最近的研究集中在采用新的拓扑图构建重叠路由网络,以减少路由表容量和路由延时。这些新的拓扑关系的基本原理是在DHT表一维空间的基础上引入更多的拓扑结构图来反映底层网络的结构。

DHT类结构能够自适应节点的动态加入/退出,有着良好的可扩展性、鲁棒性、节点ID分配的均匀性和自组织能力。由于重叠网络采用了确定性拓扑结构,DHT可以提供精确的

发现算法, 最经典的结构有Chord<sup>[27]</sup>, CAN<sup>[37]</sup>, Tapestry<sup>[28]</sup>和Pastry<sup>[28]</sup>。

## 2.3.1 Chord

### 2.3.1.1 相容哈希

Chord实现了这样一种操作: 给定一个关键字(key), 将key映射到某个节点。如果给对等网络应用的每个数据都分配一个key, 那么对等网络中的数据查找问题就可以用Chord很容易地解决了。Chord采用了相容哈希<sup>[94]</sup>的一种变体为节点分配关键字。相容哈希有几个很好的特点, 首先是哈希函数可以做到负载均衡, 也就是说所有的节点可以接收到基本相同数量的关键字。另外, 当第N个节点加入或者离开网络时, 只有 $1/N$ 的关键字需要移动到另外的位置。

Chord进一步改善了相容哈希的可扩展性。在Chord中, 节点并不需要知道所有其他节点的信息。每个Chord节点只需要知道关于其他节点的少量的“路由”信息。在由N个节点组成的网络中, 每个节点只需要维护其他 $O(\log N)$ 个节点的信息, 同样, 每次查找只需要 $O(\log N)$ 条消息。当节点加入或者离开网络时, Chord需要更新路由信息, 每次加入或者离开需要传递 $O(\log^2 N)$ 条消息。

相容哈希函数为每个节点和关键字分配 $m$ 位的标识符, 此标识符可以用SHA-1<sup>[59]</sup>等哈希函数产生。节点的标识符可以通过哈希节点的IP地址产生, 而关键字的标识符可以直接哈希此关键字。比如IP地址为210.45.43.2的节点经过SHA-1哈希之后得到的标识符为216, 而关键字“ComputerNetwork”哈希之后的关键字为65。标识符长度 $m$ 必须足够长, 这样才能保证两个节点或者关键字被哈希到同一个标识符上的概率小到可以忽略不计。从图2.3可以看出Chord结构的特点。

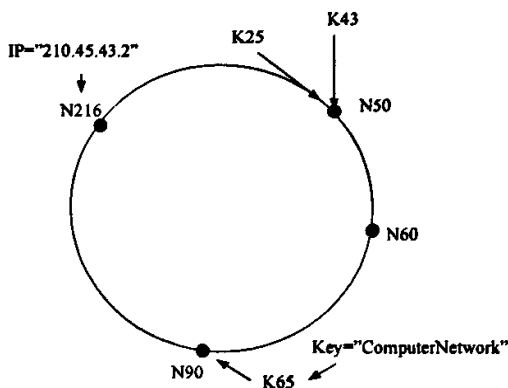


图2.3 Chord结构图

在Chord中, 每个关键字都保存在它的后继(successor)节点中, 后继节点是节点标识符大于等于关键字 $k$ 标识符的第一个节点, 我们将其记为 $\text{successor}(k)$ 。由于关键字

“ComputerNetwork”的标识符为65，因此它被保存在90节点中。如果标识符采用 $m$ 位二进制数表示，并且将从0到 $2^m-1$ 的数排列成一个圆圈，那么 $\text{succesor}(k)$ 就是从 $k$ 开始顺时针方向距离最近的节点。这一点，可以从图2.3中很清楚地得出。

Chord的一个特点就是当节点加入或者离开网络时对网络带来的冲击可以达到最小。当节点 $n$ 加入网络时，为了保持相容哈希映射，某些原来分配给 $n$ 的后继节点的关键字将分配给 $n$ 。当节点 $n$ 离开网络时，所有分配给它的关键字将重新分配给 $n$ 的后继节点。除此之外，网络中不会发生其他的变化。以图2.3为例，当节点N90离开网络时，关键字“ComputerNetwork”将被分配给节点N216。

2.3.1.2 关键字查找

在Chord中，每个节点维护少量的路由信息，通过这些路由信息，可以提高查询的效率。如果 $m$ 是关键字和节点标识符的位数（采用二进制表示），那么每个节点只需要维护一张最多 $m$ 个表项的路由表，我们称之为指针表(finger table)。节点 $n$ 的查找表的第 $i$ 个表项包括的是 $s=\text{success}(n+2^{i-1})$ ，这里 $1 \leq i \leq m$ 并且所有的计算都要进行 $\text{mod } 2^m$ ， $s$ 称为节点 $n$ 的第 $i$ 个指针，我们用 $n.\text{finger}[i].\text{node}$ 表示，指针表中的其他项的含义如下表所示。

表2.3 Chord节点指针表中各项的含义

符 号	定 义
Finger[k].start	$(n+2^{k-1}) \bmod 2^m, \quad 1 \leq k \leq m$
.interval	$[\text{finger}[k].\text{start}, \text{finger}[k+1].\text{start}]$
.node	第一个大于等于 $n.\text{finger}[k].\text{start}$ 的节点
successor	标识符环中的下一个节点; $\text{finger}[i].\text{node}$
predecessor	标识符环中的前一个节点

以图2.4为例，节点1的指针表的表项应该分别指向标识符 $(1+2^0) \bmod 2^3 = 2, (1+2^1) \bmod 2^3=3, (1+2^2) \bmod 2^3 =5$ 。而标识符2的后继是节点3，因为它是2之后的第一个节点，标识符3的后继是节点3，而标识符5的后继是节点0。

这一方案有两个重要的特性：首先，每个节点都只需要知道一部分节点的信息，而且离它越近的节点，它就知道越多的信息。其次，每个节点的指针表通常并不包括足够的信息可以确定任意一个关键字的位置。例如，图2.4中的节点3就不知道关键字1的位置，因为1的后继节点信息并没有包含在节点3的指针表中。

当节点 $n$ 不知道关键字 $k$ 的后继节点时怎么办？如果 $n$ 能够找到一个节点，这个节点的标识符更接近 $k$ ，那么这个节点将会知道该关键字的更多信息。根据这一特性， $n$ 将查找它的指针表，找到节点标识符大于 $k$ 的第一个节点 $j$ ，并询问节点 $j$ ，看 $j$ 是否知道哪个节点更靠近 $k$ 。通过重复这个过程， $n$ 最终将会知道 $k$ 的后继节点。

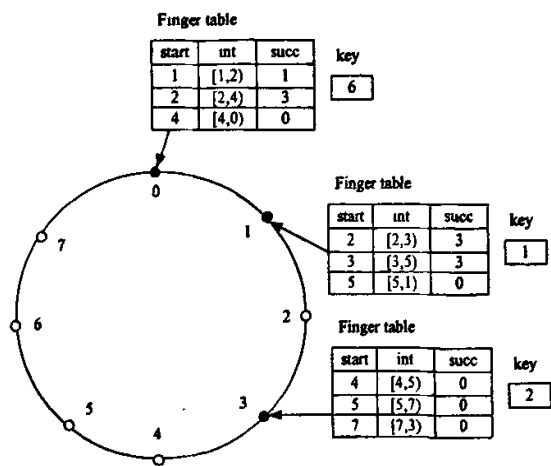


图2.4 Chord的数据组织

仍然考虑图2.4中的例子，节点3需要查找关键字1的后继节点。由于1属于循环区间[7, 3]，它属于3.finger[3].interval，因此节点3查找其指针表的第3项，返回0。由于0在1之前，因此节点3将要求0去寻找关键字1的后继节点。依此类推，节点0将查找它的指针表并发现1的后继节点是1本身，于是节点0将告诉节点3节点1是它要找的节点。

2.3.1.3 新节点的加入

节点n的加入分为三个阶段。

(1) 初始化新节点的指针表。

假设节点n在加入网络之前通过某种机制知道网络中的某个节点n'。这时，为了初始化n的指针表，n将要求节点n'为它查找指针表中的其他表项。

(2) 更新现有其他节点的指针表。

节点加入网络后将调用其他节点的更新函数，让其他节点更新其指针表。

(3) 从后继节点把关键字传递到节点n。

这一步是把所有后继节点是n的关键字转移到n上。整个加入操作的时间复杂度是O(log<sup>2</sup>N)，如果采用更复杂的算法<sup>[27]</sup>，可以把复杂度降低到O(logN)。

2.3.1.4 节点失效处理

在对等网络中，某个对等节点随时可能退出系统或者发生失效，因此处理节点失效是一个重要的问题。在Chord中，当节点n失效时，所有在指针表中包括n的节点都必须把n替换成n的后继节点。另外，节点n的失效不能影响系统中正在进行的查询过程。

在失效处理中最关键的步骤是维护正确的后继指针。为了保证这一点，每个Chord节点都维护一张包括r个最近后继的后继列表。如果节点n注意到它的后继节点失效了，它就用后继列表中第一个正常节点替换失效节点。

## 2.3.2 CAN

CAN(Content-Addressable Network, 内容访问网络)<sup>[37]</sup>可以在Internet规模的大型对等网络上提供类似哈希表的功能。CAN具有可扩展、容错和完全自组织等特点。

### 2.3.2.1 CAN的组成

CAN类似于一张大哈希表，CAN的基本操作包括插入、查找和删除（关键字，值）对。CAN由大量自治的节点组成。每个节点保存哈希表的一部分，称为一个区(zone)。此外，每个节点还保存少量的邻接区的信息。对每个特定关键字的插入（或者查找、删除）请求由中间的CAN节点进行路由直到到达包括该关键字的CAN节点所在的区。CAN的设计完全是分布式的，它不需要任何形式的中央控制点。CAN具有很好的可扩展性，节点只需要维护少量的控制状态而且状态数量独立于系统中的节点数量。CAN支持容错特性，节点可以绕过错误节点进行路由。

CAN基于虚拟的 $d$ 维笛卡儿坐标空间实现其数据组织和查找功能。整个坐标空间动态地分配给系统中的所有节点，每个节点都拥有独立的互不相交的一块区域。图2.5给出了一个2维的 $[0, 1] \times [0, 1]$ 的笛卡儿坐标空间划分成五个节点区域的情况。

虚拟坐标空间采用下面的方法保存（关键字，值）对。当保存 $(K1, V1)$ 时，使用统一的哈希函数把关键字 $K1$ 映射成坐标空间中的点 $P$ 。那么这个值将被保存在该点所在区域的节点中。当需要查询关键字 $K1$ 对应的值时，任何节点都可以使用同样的哈希函数找到 $K1$ 对应的点 $P$ ，然后从该点对应的节点取出相应的值。如果此节点不是发起查询请求的节点，CAN将负责将此查询请求转发到对应的节点。因此，有效的路由机制是CAN中的一个关键问题。

### 2.3.2.2 CAN中的路由机制

CAN中的路由机制非常简单，只需要计算目的点的坐标，然后寻找从发起请求的点到目的点的一条路径就可以。首先我们需要给出两个节点区域邻接的含义，在 $d$ 维坐标空间中，当两个区域在 $d-1$ 维上都覆盖相同的跨度而在另一维上相互邻接时，则称这两个区域邻接。例如，图2.5中D和E是邻接节点，而D和A就不是邻接节点。每个CAN节点都保存一张坐标路由表，其中包括它的邻接节点的IP地址和虚拟坐标区域。每条CAN消息都包括目的点坐标。路由时节点只要朝着目标节点的方向把请求转发给自己的邻接节点就可以了。

图2.6给出了查找过程的一个简单的例子。如果一个 $d$ 维空间划分成 $n$ 个相等的区域，那么平均路由长度是 $(d/4)(n^{1/d})$ ，每个节点只需要维护 $2d$ 的邻接节点信息。这个结果表明CAN的可扩展性很好，节点数增加时每个节点维护的信息不变，而且路由长度只是以 $O(n^{1/d})$ 的数量级增长。我们可以看到，在坐标空间中，两点之间可以有許多条不同的路径。因此，单个节点的失效对CAN基本上没有太大的影响。遇到失效节点时，CAN节点会自动沿着其他的路径进行路由。

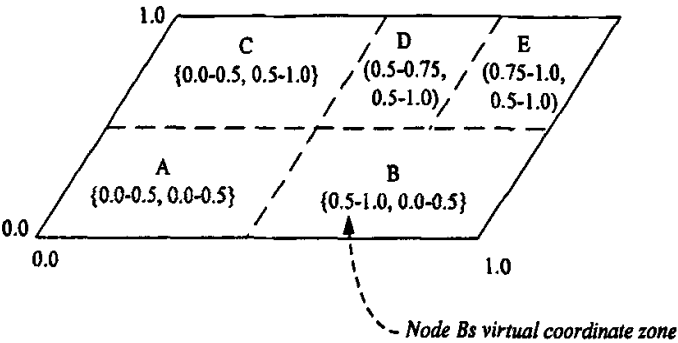


图2.5 CAN坐标空间的区域划分

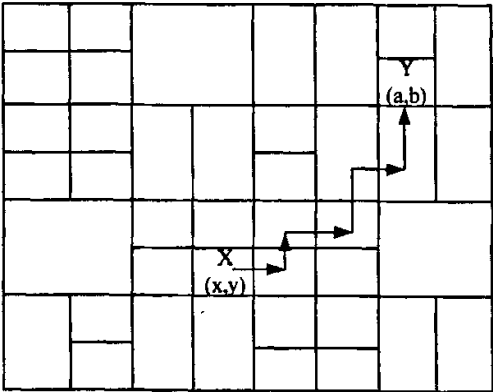


图2.6 CAN的路由算法图

2.3.2.3 节点的加入和退出

CAN是一种动态网络，当一个新的节点加入网络时必须得到自己的一块坐标空间。CAN通过分裂现有的节点区域实现这一过程。它把某个现有节点的区域分裂成同样大小的两块，把其中一块分给新加入的节点。整个过程分为以下三步：

- （1）新节点必须首先找到一个已经在CAN中的节点。
- （2）新节点使用CAN的路由机制找到一个区域将要被分隔的节点。
- （3）执行分裂操作，然后原有区域的邻接区域必须被告知发生了分裂，这样新节点才能被别的节点路由到。

当节点离开CAN时，必须保证它的区域被CAN系统收回，也就是分配给其他仍然在系统中的节点。一般过程是由某个节点来接管这个区域和所有的（关键字，值）数据库。如果这个区域可以和相邻区域合并形成一个新的区域，那么CAN将执行合并操作。如果合并不能进行，那么该区域将交给其邻接节点中区域最小的节点。也就是说，这个节点将临时负责两个区域。

### 2.3.3 Tapestry

Tapestry<sup>[26]</sup>是伯克利的Zhao, Kubiatiwicz和Joseph提出的一种用于在OceanStore<sup>[25]</sup>中进行查找和路由工作的基础结构。它的工作机制来源于Plaxton<sup>[96]</sup>方案,其命名、结构和核心查询和路由原理都类似于Plaxton,但它提供了自适应、容错性和自我优化等特性。

Tapestry算法中,每个节点都需要维护一个邻居图(路由表)、反向指针列表。对象位置指针和一些热点监视信息。路由表的大小为 $b \cdot \log_b N$ ( $b$ 行,  $\log_b N$ 列),每一项包含了网络中与当前节点最近的那些节点的信息,而且这些节点标识号中与列数相应数目的后缀也要与当前节点相同(即第 $i$ 列需要和第 $i$ 个后缀相同)。每个节点还要维护一个指向其邻居的反向指针列表,主要用于生成节点的路由表。对象位置指针采用拓扑形式<对象ID, 节点ID>,用于在对象向服务器发送路由请求时提供帮助;热点监视信息采用如<对象ID, 节点ID, 频率>这种形式的拓扑集合,这些信息用于帮助缓冲决定路由。

Tapestry的核心定位与路由机制类似于Plaxton算法。首先,当前节点计算自己的ID与自己的ID的相同后缀的数目 $j$ ;然后,从路由表中选择一个中间节点(一般在 $j+1$ 列),使得中间节点的ID与目的节点的ID相同的后缀数目大于等于 $j+1$ ,接着把查询请求任务转交给这个中间节点继续查询。这个过程递归执行,直到找到目的ID。可以看出每次任务转交,中间节点的ID和目的节点的ID的相同后缀数据至少会增加1,如果将ID作为字符串,则串长为 $\log_b N$ 。因此, Tapestry中,从任意一个节点开始查询,都能在 $\log_b N$ 跳内路由到目的节点。

### 2.3.4 Pastry

Pastry<sup>[28]</sup>是Microsoft和Rice大学共同发起的对等网络匿名存储系统中的定位和路由算法,该算法与Tapestry有许多相似之处:使用匹配前缀或者后缀地址的路由方法、插入和删除算法有类似的代价;它们在设计时都考虑了查询的实际路径(IP路由)和逻辑跳之间的关系,通过一定手段使得相对延迟(Relative Delay Penalty, RDP)尽可能少。Pastry与Tapestry的区别在于:第一, Pastry中对象复制无需由所有者来控制,在对象的出版/发布过程中,复制对象被放到与对象编号最接近的多个节点上。第二, Tapestry在服务器和根节点之间的路由过程中放置对象指针信息,而在Pastry中客户端通过对象编号直接路由到保存对象复制点附近,这样在实际网络中多个复制对象存在于实际物理网络的不同节点上,在多个节点存储复制对象带来了存储开销、安全性、保密性和一致性问题。第三,与Tapestry的代理路由算法(Surrogate Routing)相比, Pastry的路由跳数具有较弱的理论分析性, Pastry的路由距离具有可证明的理论分析性和概率界限,并保证能够找到已存在的可到达对象。最后, Pastry路由算法不能充分利用本地性提高路由效率。

## 2.4 P2P系统中安全性问题的研究

P2P系统的高度动态性和自适应性在带来系统灵活性和可扩展性优点的同时也带来了



一些问题。比如, P2P系统中节点的匿名性会导致系统中提供服务的节点中存在大量欺诈行为以及存在不可靠的服务质量, 从而降低了整个系统的可用性。由于P2P系统中节点的资源共享是用户的自愿行为, 服务的提供方随时可以中止服务, 甚至可能存在欺诈行为, 而无法保证服务质量。因此研究P2P系统中的安全性问题至关重要。目前, 在P2P安全性问题方面已经展开很多研究工作。

### 2.4.1 基于PKI的信任模型

在这类系统中, 存在少数超级节点, 超级节点负责整个网络的监督, 定期通告违规的节点。同时这些超级节点的合法性是通过CA颁发的证书加以保证的。这类系统的实例有: Onsale Exchange<sup>[70]</sup>, eBay<sup>[71]</sup>等。Onsale Exchange允许它的用户评估并提交关于卖者的评价, 一个卖者的信任是从它的客户收集到的所有评估的平均值。在eBay中, 每次拍卖结束后, 卖方接收关于它们的可靠性的反馈, 0或1。卖方的信任是过去六个月评估的总和。这类系统往往是中心依赖的, 具有可扩展性、单点失效等问题。

### 2.4.2 基于局部推荐的信任模型

在基于局部推荐的信任模型中, 参与者通过询问其他参与者来获得某个参与者的可信度。如Cornelli对Gnutella的改进建议就是采用这种方法<sup>[97]</sup>; PeerTrust<sup>[100]</sup>也是基于推荐的局部信任模型。该模型通过节点的交易反馈信息来量化、比较节点的可信度。但获取节点的可信度往往是片面的、局部的。

### 2.4.3 全局信任模型

全局信任模型认为对开放网络分布式应用中的每一参与者都存在一个全局信任度, 此类模型通过邻居参与者间相互满意度的迭代, 计算出参与者的全局可信度, 从而使分布式应用的任意一个参与者随时都可以较方便地获取其他参与者的全局信任度。EigenTrust<sup>[99]</sup>属于此类模型。

在Kamvar<sup>[99]</sup>等人提出的EigenTrust中, 局部信任度为参与者*i*根据事务交互情况对参与者*j*的评价的总和,

$$S_{ij} = \text{sat}(i,j) - \text{unsat}(i,j)$$

其中,  $\text{sat}(i,j)$ 为*i*认为满意的事务数, 而 $\text{unsat}(i,j)$ 为*i*不满意的事务数。

对局部信任度 $s_{ij}$ 进行正规化得到

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}$$

参与者*i*向邻居*j*询问他对其他参与者(*k*)的信任度, EigenTrust对正规化的局部信任度进

行聚集即可得到全局信任度

$$t_{ik} = \sum_j c_{ij} c_{jk}$$

通过综合*i*对*j*的信任 $c_{ij}$ 和*j*对*k*的信任 $c_{jk}$ ，即可得到*i*对*k*的全局信任 $t_{ik}$ 。EigenTrust将其表示为矩阵的形式，并对其进行迭代计算，即可得到整个网络中所有参与者的全局信任度。但是该模型对于同谋和诋毁攻击过分依赖预可信节点，而P2P环境中是否存在预可信节点是值得商榷的。

## 2.5 结构化对等网络中基于语义的信息检索技术

尽管上述结构化P2P系统都考虑到了存在大量节点时的可扩展性问题，解决这些系统中大量信息的语义性同样重要，从而激发了基于内容/语义聚类的发展。为了实现基于语义的搜索，数据对象通常用一些提取出来的属性值集合表示。这些属性拥有不同的格式和不同的域，可以逻辑地代表数据对象的语义。这样，每个数据对象可以被看作多维语义空间中的一个点。因此，在这样一个语义空间中对数据对象的查询可以根据这些属性来描述。典型的代表有pSearch<sup>[100]</sup>，SSW<sup>[101]</sup>(Semantic Small World)。

### 2.5.1 pSearch

pSearch<sup>[100]</sup>在*m*维的CAN上建立了基于语义信息的查询机制，其中 $m = 2.3 \cdot \ln(N)$ ，并且*N*是节点的总数。为了解决高维的问题，pSearch采用rolling index技术划分维数为*p*组（其中每组包括*m*维），并且将与每组相关的部分语义空间映射到CAN的相应关键字空间。为了进行基于语义的相似度查询，*p*个分割的查询在CAN关键字空间上分别进行。这*p*个查询中最相似的数据对象作为结果被返回。pSearch存在着四个不足。首先，它缺少足够的灵活性来调节系统中的动态变化。因为pSearch中CAN的维数是事先定义好的，所以对于pSearch如何适应网络规模的变化从而获得最优路径长度并不明确。其次，CAN利用了相对规则的关键字空间划分方案（对于空间的平均划分），会导致节点上的负载不平衡。这种无效被带到了pSearch中。第二，由于只有部分语义信息被映射到CAN，就需要经验采样来提高搜索效率，而这种采用会消耗大量带宽特别是当节点成员频繁变化时。除此之外，这种采样的有效性依靠存储在一个节点上数据对象的同构性。最后，pSearch语义空间中节点的位置并不完全体现它本地数据对象的语义。

### 2.5.2 SSW

和rolling index的降维技术不同，SSW<sup>[101]</sup>采用的方法是降低叠加网络的维数。尽管一个高维的叠加网络具有较高搜索效率（节点之间存在的大量可行路径），其维护的成本和负载则是相当高的。因此，要构建一个低维的叠加网络支持节点的连接和基于语义的查询。SSW

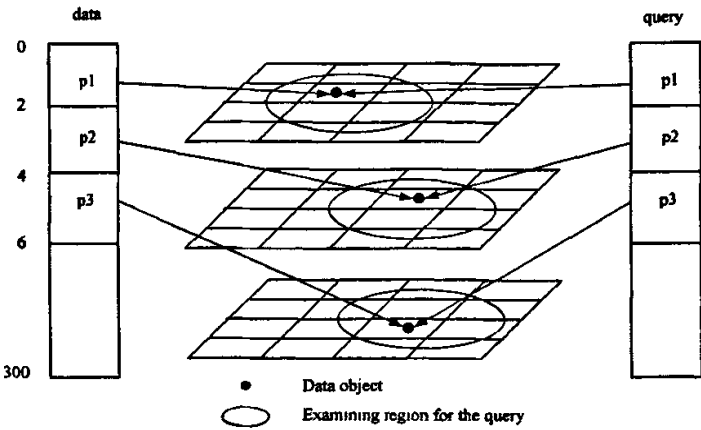


图2.7 一个pSearch查询的例子

通过自适应空间线性化方法(Adaptive space linearization)实现，它在语义小世界构建的聚类划分过程中将高维空间上的聚类线性化到一维语义小世界上。ASL尽量保持聚类间的语义相关性。ASL的作用和一些熟知技术Hilbert curve, Z-curve等等具有相似的作用。但是，这些技术只能用于将规则的高维空间映射到低维空间。高维语义空间是根据数据的密度动态自适应的，因此这些技术无法用于语义小世界的降维。

一维语义小世界的构造类似于双向链表，由语义聚类组成，这些语义聚类是通过每个节点的两个短程关联连接起来的。除了这种基于基本连接的线性网络结构，远程关联还为其他聚类提供了捷径从而有助于搜索。当原始语义空间被划分并且线性化时，一维语义小世界的聚类仍然和高维的原始语义子空间保持一致。这样，一维语义子空间的关键需求是如何在保证和原始语义空间相符的情况下，基于高维语义信息对命名空间进行编码，以进行有效的查询。或者说，一个查询应该在语义空间中快速到达其目的节点。

图2.8(a)给出的系统中语义空间被划分为11个聚类，其中每个聚类的ID都已经给出，本例子在一个二维空间中实现上述过程，其中垂直的线为第一维，水平的线为第二维。假设命名空间有4位。在这个例子中，语义空间首先沿“p=1”的垂直线划分。在这一点，线左边的节点和右边的节点分别获得ID“0000”和“1000”。然后左边再沿“p=2”的水平线划分，左下和左上的节点ID分别为“0000”和“0100”。实线给出了聚类ID的顺序，虚线（SSW是双向链表而自然形成的）表示查询可以双向进行。图2.8(b)给出了基于上面描述的命名空间中SSW-ID的建立。聚类4中的一个节点维护到邻居聚类2和5的短程关联，还维护了到聚类10的远程关联。

2.6 本章小结

当前和将来的P2P系统普遍具有规模大、动态性强和异构性强的特点。传统的资源定位机制基于洪泛算法，它们虽然实现简单，但已经不适用于当前和将来P2P系统的特点，不仅

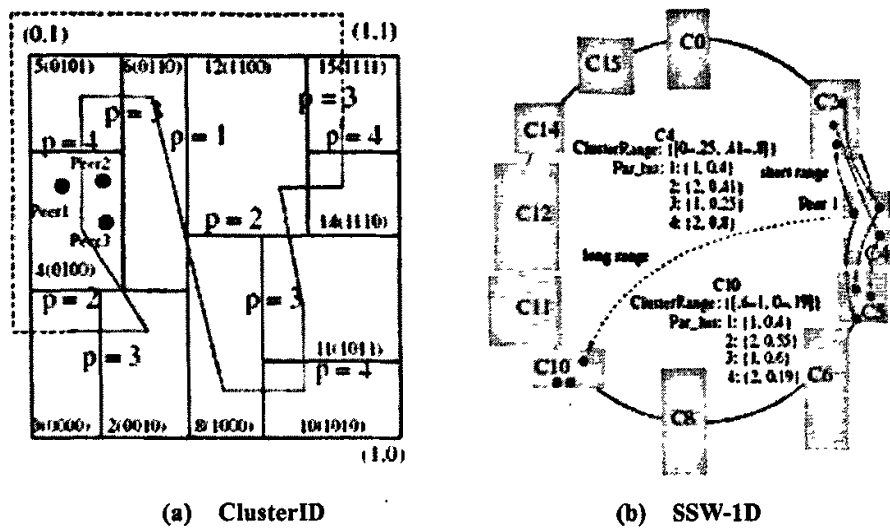


图2.8 SSW-1D的降维图

给网络带宽带来严重的影响，同时也降低了定位机制的性能。而基于DHT的结构化P2P尽管可以快速准确的定位资源，却不支持基于内容的查询。为了解决P2P系统中资源管理机制中出现的问题，相关研究主要从资源管理机制的三个组成部分入手进行改进。

在P2P叠加层，将针对目前大多数P2P系统的构建没有考虑到物理因素以及P2P系统中由节点匿名性所带来的恶意欺诈服务问题展开研究。

在文件管理层。针对当前结构化P2P文件存储系统存在仅支持单关键字的精确匹配，而缺乏支持内容查询的局限性，提出一种基于内容的资源定位方案成为有待解决的问题。

在检索模型方面，对于高维语义空间引起的文本相似度计算复杂，以及完全分布式非结构化P2P系统中广播算法带来的大量冗余消息，如何解决P2P层与语义层的不匹配问题有待进一步的研究。

本文将针对上述问题展开进一步研究,首先在第三章建立一个基于延迟敏感的双向查询Chord系统，根据节点和参考点之间的延迟对节点进行分组，减少平均相应时间；通过实现Chord上的双向查询，缩短搜索路径长度。

## 第三章 基于延迟敏感的双向查询Chord系统的构建

叠加层构建技术是P2P系统中资源管理机制的基础组成部分。目前大多数P2P系统的构建没有考虑到底层物理网络的实际情况,这就产生了叠加网络拓扑与底层物理网络不匹配的问题,从而导致P2P节点定位资源的延迟和开销增加。针对这个问题,本章提出了基于延迟敏感的双向查询Chord系统,根据节点和参考点之间的延迟对节点进行分组,减少平均相应时间;通过实现Chord上的双向查询,缩短搜索路径长度。

### 3.1 引言

P2P(peer to peer)作为建立在叠加层的网络,凭借灵活性(flexibility)和可扩展性(scalability)被广泛应用于文件存储、即时消息发送、对等计算等方面。在P2P应用中面临的最基本问题是如何有效快速的定位数据信息,即减少从源节点到目的节点的时间延迟(path latency)。这里的时间延迟包括两个因素,一个是从源节点到目的节点之间位于叠加层上的跳数,一个是每跳的响应延迟<sup>[102][103]</sup>。结构化P2P中基于DHT(distributed hash tables)的搜索技术,如Chord, CAN均提供了叠加层上有效的资源定位机制,保证查询在有限的路径长度内完成。但是,由于P2P叠加网络拓扑的构建没有考虑底层物理网络的实际情况,产生了叠加网络拓扑和底层物理网络不匹配的问题,导致P2P节点定位资源的延迟和开销增加。同时,一般Chord查询算法只沿顺时针方向进行查找,这种单向的查找方法使得路径长度和系统中节点的数目呈对数关系。在网络规模较大的情况下,对数级的转发次数仍然会带来大量的通讯量。

本章提出基于延迟敏感的双向查询Chord系统来解决上述问题。节点根据和参考点之间的延迟确定自己所属的分组,同一分组内的节点构建成Chord形式,并且组和组之间由超级节点相连。与此同时,在Chord上实现数据定位时,本系统采用双向查询。系统中的节点同时维护前驱指针和后继指针,并按照和目的节点之间的逻辑距离判断查询是沿顺时针还是逆时针进行。实验证明,在这样的划分机制中,同一Chord上节点之间的延迟比随机分配时同一Chord上节点之间的延迟要短。模拟结果显示,基于延迟敏感的双向查询Chord系统把路径长度控制在了常数范围内,并且基于延迟敏感的划分策略减少了同一分组内节点之间平均延迟,从而更加有效地减少资源定位的响应时间。

本章第二部分介绍相关工作以及如何基于延迟敏感构建多个Chord互连的系统,以及系统中信息的查找;第三部分详细描述本文提出的双向查询Chord系统和算法实现;第四部分是仿真和结果分析,最后是小结。

## 3.2 相关工作

P2P中路由长度指的是位于叠加层的跳数,而不是网络层的,当前把叠加层和网络拓扑联系起来的研究并不多。对于叠加层的路由来说,考虑网络拓扑是很重要的,否则,在叠加层上的节点之间的消息传递就会花费较多的时间以及不必要的通信量。

目前,用于构建基于拓扑的P2P系统的策略主要有三种,分别是proximity routing, topology-based nodeID assignment, 以及proximity neighbor selection<sup>[104]</sup>。Proximity routing算法在选路时不仅考虑节点特性,还考虑选择延迟最小的下一跳节点。算法的基本操作是对于每一次查询给出 $n$ 个可选的邻居,由转发节点决定选择哪个路由对查询进行转发。这种算法每一步都有 $n$ 个选择,两节点之间的延迟可以从原来的平均延迟,减少到和 $n$ 个候选节点中最近的那个节点之间的延迟一样大。但是这种算法的局限是其优化程度取决于 $n$ 的大小。Topology-based nodeID assignment是把叠加层的逻辑ID空间映射到物理网络从而使得ID空间中的邻居节点在物理网络上也是相近的。但是这种方法破坏了ID空间划分的均衡性,并且不适用于一维ID空间的叠加层网络。和Topology-based nodeID assignment类似,Proximity neighbor selection也是建立了一个延迟敏感的叠加层网络。但是,它使用的不是ID空间的映射,而是从路由表的表项中所有拥有nodeID的节点中,选择拓扑结构上最近的节点。

基于延迟敏感是在构造叠加层网络时,根据节点与参考点之间的网络延迟对节点进行分组的划分,位于同一组的节点形成一个Chord。网络延迟的测量依据是节点向参考点发送ICMP协议中echo消息(ping)所得到的时间。

## 3.3 基于延迟敏感的双向查询Chord系统

这一部分首先简单介绍现有Chord的结构,包括指针表的构造和查找算法,以及对于节点加入和离开的信息维护,然后详细介绍本文提出的一种基于延迟敏感的双向查询Chord实现机制,最后是理论证明。

### 3.3.1 基于延迟敏感的Chord系统的构造

基于延迟敏感构建Chord的基本思想是按照网络上节点之间彼此的网络延迟把它们划分成不同的组,组内节点构成一个Chord,Chord与Chord之间由超级节点<sup>[105]</sup>相连。这样组成的结构中,组内节点之间的延迟比组与组之间的节点延迟相对要小。当一个节点要搜索数据时,首先在自己所在的Chord上进行查找,如果没有,则由超级节点把查询转发到附近的Chord上继续查找,直到返回结果,或者遍历所有节点返回失败信息。

节点的划分是按照它们到一些参考点的延迟来确定的。首先,在网络中选择 $m$ 台机器的IP作为参考点,记为landmarks。这 $m$ 个参考点按照一定的次序进行排列(可以有 $m!$ 种排序),

形成具有 $(l_1, l_2, \dots, l_m)$ 次序的 $m$ 维坐标空间, 其中沿每一维上的数值表示的是节点到该参考点的延迟, 节点加入之前要获得到这些参考点的网络延迟。然后, 基于延迟敏感的Chord系统把节点到这些参考点的相对距离映射到 $m$ 维的坐标空间上, 并定义一个阈值, 按照阈值把坐标空间划分成均匀的组, 分配到每个组内的节点组织成Chord的形式。这种基于网络延迟对节点进行分组而形成Chord的思想实现起来非常简单, 对于体系结构方面的唯一需求就是增加几台机器作为参考点。实验证明, 对于现有的网络规模来说, 参考点的个数应在8—12之间。这些机器对于整个P2P网络来说是安全的, 并且它们所处理的操作只是响应节点发来的ping消息。

如图3.1所示, 假定选取三台机器作为参考点, 分别记为L1, L2和L3。节点N1向这三台机器发送ping消息, 获得的网络延迟分别是35ms, 15ms, 27ms。如果选取10ms为阈值, 系统按照每10ms为一个单位对空间进行划分。因此, 节点N1位于组(3, 1, 2); 同时, 节点N3也位于组(3, 1, 2), 这样, N1和N3分在同一分组中, 如图3.2所示。对于位于同一个分组中的节点组织成Chord环的形式, 并且认为同一Chord中节点之间的网络延迟比不同Chord上节点之间的网络延迟小。

当一个节点要加入基于延迟敏感的Chord系统时, 主要执行以下两步:

- ✧ 向  $m$  个参考点发送 ping 消息, 并根据 $(l_1, l_2, \dots, l_m)$ 把自己加入相应的分组;
- ✧ 在相应的分组内组成Chord, 并按照SHA-1获得不同标识符, 以及这个分组内超级节点信息。

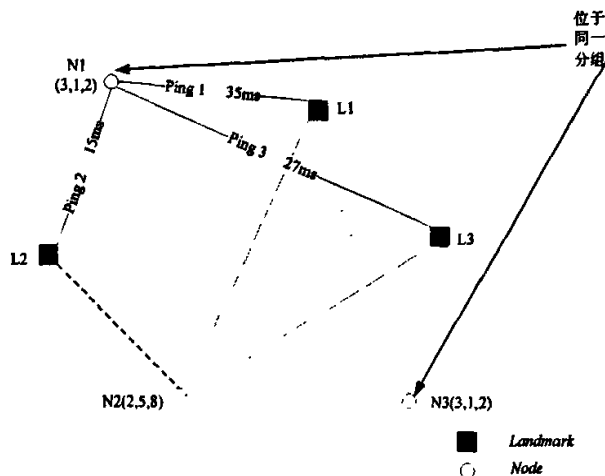


图3.1 基于网络延迟的分组

对于这种划分机制, 还要考虑两个问题, 一个是可扩展性, 另一个是这种划分方法是否确实把相近的节点划入了同一Chord。机制的可扩展性是显然的, 因为每个节点只需要获得到参考点的延迟。尽管参考点可能要接受百万个节点的ping消息, 其负载也是在可控范围内的, 这一点用快速back-of-the-envelope算法已经获得证明。

至于这种方法是否保证把相近节点划入同一分组, 在这里定义一个节点的latency ratio:

$$\text{latency ratio} = \text{intra-bin latency} / \text{inter-bin latency}$$

其中, intra-bin latency代表该节点和相同Chord中随机选取的一个节点之间的延迟, inter-bin latency代表该节点和其他Chord中随机选取一个节点的延迟。这里latency ratio越小, 说明相同Chord上节点之间的网络延迟越小, 最后用所有节点latency ratio的平均值衡量划分的有效性。通过模拟实验可以看到, 这种基于延迟敏感的Chord系统, 和节点随机加入的系统相比, 网络延迟明显减少。

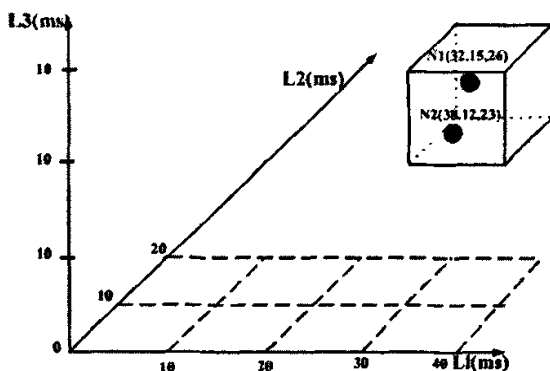


图3.2 基于网络延迟的分组

### 3.3.2 Chord系统的一般结构

结构化P2P一般基于DHT把节点node和关键字key分配到一个命名空间上, 例如, Chord是根据SHA-1把节点分配到 $m$ -位的环上。在现有的Chord系统中, 哈希函数为每个节点和关键字分配 $m$ 位的标识符, 这个标识符是通过分别对节点的IP地址和关键字使用SHA-1函数得到, 其中 $m$ 必须足够大从而保证任何两个节点或者关键字被分配同一个标识符的概率可以很小而忽略不计。关键字被分配在后继节点中, 后继节点定义为节点标识符大于或等于关键字 $k$ 标识符的第一个节点, 记为 $\text{successor}(k)$ 。如果标识符采用 $m$ 位的二进制表示, 并且把从0到 $2^m - 1$ 的标识符组成一个环, 那么后继节点就是从关键字 $k$ 开始沿顺时针方向的第一个节点。如图3.3所示, 节点被分配在一个 $m$ 为4的Chord环上, 其中包含有6个节点和4个关键字, 关键字5和6的后继是节点6, 因此, 关键字5和6被存放在N6中。

为了能够正确的找到关键字 $k$ 所在的位置, 每个节点只需要了解后继指针的信息。在简单的关键字查找过程中, 每个节点只要在环上沿后继指针进行查找, 直到找到关键字。但是这种查找算法在对数据进行搜索的过程中, 所转发的信息和节点数目是呈线性关系的, 当整个系统规模增大时, 进行查找所需要的信息就会成倍增长, 大大降低了查询的效率。为了加快查找速度, Chord为每个节点增加一个指针表, 记作finger table。

图3.3所示的是节点N1维护的指针表, 当一个节点要执行一个查询操作时, 它首先查询自己的后继节点中是否有所找关键字 $k$ , 如果有, 返回后继节点; 如果没有, 就在指针表中查找关键字 $k$ 所在的区间, 并返回该区间所对应表项的后继指针, 由这个指针继续查找关键



字 $k$ 。这样选择后继指针的依据是，距离 $k$ 越近的节点所包含的对于 $k$ 所在区域的标识符环的信息越多。在图3.3中，节点N1查找关键字13，首先它判断13不在N1和N1的后继节点之间，开始对指针表进行查询，发现13落在 $[9, 1)$ 这个区间内，于是由 $\text{finger}[4].\text{successor}$ （在这里是N10），继续查找13，最后在N14找到。

观察指针表可以发现，表中的区间是以2的幂函数分布的，节点每次转发查询都可以减少从源节点到目标节点之间至少一半的距离。所以在有一个有 $N$ 个节点的系统中进行查询所需要的消息数是 $O(\log N)$ ，大大改善了简单查询中的线性关系。

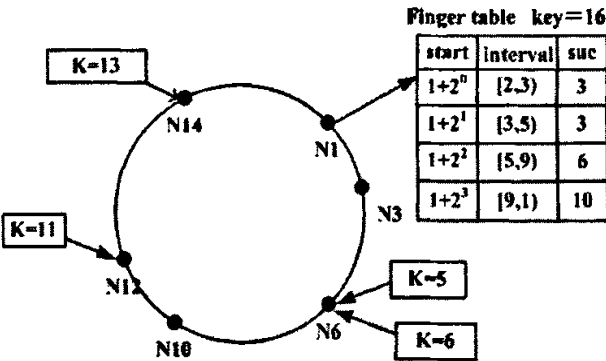


图3.3 单向Chord系统中节点N1的指针表

由于P2P系统是动态的，因此不断有节点加入和离开。只要保证后继指针的更新，就不会影响查询的正确性，但是如果指针表中的数据不及时更新，就会影响查找的效率。因此，Chord使用“stabilization”协议，系统中的节点周期性运行更新机制来维护指针表中的信息，使其保持最新。

3.3.3 双向查询Chord系统和算法

为了方便描述，首先介绍Chord系统中的两个基本组成部分，即节点(nodes)和路由(edges)。其中节点是指加入到P2P系统中的机器的标识符，它们在Chord系统中沿顺时针被标记为 $0, 1, 2, \dots, 2^b-1$ 的二进制表示。Chord环上的任意两个节点之间的边可以看作是一条路由，它代表叠加层上的一个连接。假设环上存在两个节点A和B，并且A和B之间的距离是 $2^k$ ，则A和B之间的路由(edge)可以定义为 $|A-B|=2^k$ 或者 $2^m-2^k$ ，这里 $m$ 是标识符的二进制表示位，并且 $0 \leq k < m$ 。

通过对图3.3的观察，会发现Chord系统中的算法只使用沿顺时针的方向上的路由，并且从源节点到目的节点之间的逻辑距离是呈2的幂减少的。这种只是在顺时针方向上的路径对于一个Chord环上的源节点和目的节点来说并不一定是最短路径。如图3.3中的节点N1要搜索关键字为13的文件，那么它首先要将查询转发给节点N10，然后再由N10依次向其后续节点转发来进行查找。但是，如果在查询中增加逆向查找功能，那么节点只要把查询转发给

它的前驱节点就可以直接找到,这样将大大减少了查询的跳数。因此,本文提出一种双向查询Chord,在Chord上实现对关键字的双向查询,减少路径长度。

为了实现Chord上的双向查询,需要为每个节点增加一个指针表,表中维护的表项是按照逆时针方向呈2的幂指数增长的节点排列的,如表3.1。在一般的Chord系统中,没有充分利用节点的邻居信息,从它的指针表中可以看出,节点对于顺时针方向邻居节点的信息了解得比较多,而对逆时针上的邻居信息却很少。实际上节点在环上的实际位置分配是随机和均匀的。在这种情况下,如果所查找的关键字在当前节点的前驱位置,也要经过当前节点的最后一个表项中节点的后继,并进行多次转发才可以到达目的节点,该算法大大降低了查找效率。如图3.3所示,如果节点N1要查找key为13的文件,在Chord系统中,首先在指针表中查找13所在区间,其后继节点是N10,然后由节点N10继续进行查询,返回N15。

表3.1 节点n的前驱指针表中变量的定义

符 号	内 容
$\text{finger}[i].\text{pre}$	$(n - 2^{i-1} + 2^m) \bmod 2^m, \quad 1 \leq i \leq m$
$\text{finger}[i].\text{interval}$	$[\text{finger}[i-1].\text{pre}, \text{finger}[i].\text{pre}), \text{ if } 1 \leq i < m; [n, \text{finger}[i].\text{pre}), \text{ if } i = m$
$\text{finger}[i].\text{node}$	第一个大于或等于 $n.\text{finger}[i].\text{pre}$ 的节点

本章提出的基于延迟敏感的双向查询Chord系统,通过增加逆时针邻居节点信息的指针表,利用邻居关系实现双向查找,发挥了Chord环状结构的基本特点;利用延迟敏感,减少同一分组内任意两节点之间的平均延迟。在该系统中,每个节点维护两个指针表,这两个指针表中的表项以当前节点为中心,分别呈顺时针方向和逆时针方向对称。这种对称式的构造可以使节点充分利用邻居信息,提高查找效率。在改进后的系统中,当一个节点发起查询的时候,它首先获得自己的ID和需要定位的关键字ID之间的差。通过差值,节点判断是沿顺时针还是逆时针进行查询。如图3.4所示,节点N1在查找key为14的文件时,通过ID的差值判断查询应沿逆时针方向执行,因此,依据前驱指针表直接把查询消息转发到节点N15,整个查找过程只需一步操作。假如在一般Chord算法中,这样的查找需要经过N10, N12等多次跳转,最后才能转发到N15。显然,双向查询Chord系统有效地减少了路径长度。

这种通过判断关键字和节点之间位于Chord环上边的长度,从而决定选择顺时针还是逆时针查询的算法,充分利用了节点逆时针方向上的邻居信息,并且仅通过增加一步判断操作而有效地减少了网络中信息的转发次数,相应地减轻了网络的负担。同时,双向查询Chord系统是基于延迟敏感的,对于划分在同一分组内的节点,彼此之间延迟较小而且分布均匀。因此,在基于延迟敏感的基础上建立双向查询Chord系统,可以有效地减少查询的响应时间。

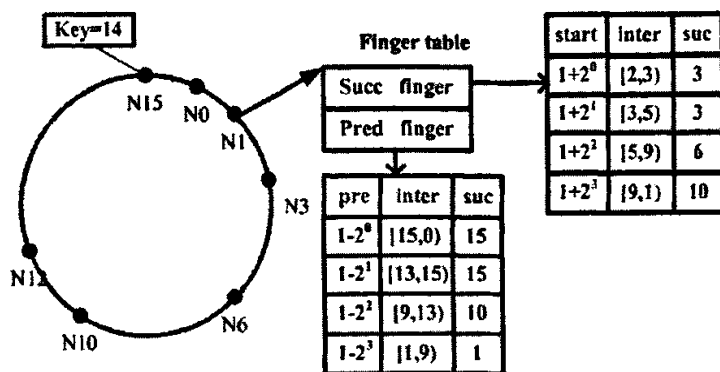


图3.4 双向查询Chord系统中节点N1的指针表

双向查询Chord算法的伪代码如下：

```
// ask node n to find the successor of ID
(1)n.BidiSearch_successor(ID)
(2)clockwise_length = ID - n + 2m mod 2m;
(3)anticlockwise_length = n - ID + 2m mod 2m;
(4)if (clockwise_length= anticlockwise_length)
//判断是沿顺时针还是逆时针
(5) edge = clockwise_length; //进行搜索
(6) else
(7) edge=min(clockwise_length, anticlockwise_length);
(8)if (edge = clockwise_length)
(9) return n.clocksearch(ID);
(10)else
(11)return n.anticlocksearch(ID);
```

图3.5 双向查询Chord算法

### 3.3.4 双向查询Chord的性能分析

基于延迟敏感的系统是依据节点和参考点之间的延迟对其进行分组的，那么和随机分配的系统相比，这里的分组中节点彼此之间延迟差异较小。这种延迟的平均性为我们的理论证明提供了有效的前提条件和保证。因此，在证明过程中，我们假设节点之间延迟相等。

设环上有两个节点A和B，则A和B之间的顺时针方向路径是 $d=(A-B+2^m) \bmod 2^m$ ，逆时针方向路径长度为 $d'=m-d$ ，并且设X为系统中任意两点之间最短距离之和，Y则为任意两点之间顺时针方向上的距离之和。那么 $X=\sum_{d \in \mathcal{D}} \min(d, d')$ ， $Y=\sum_{i=0}^{n-1} i \binom{n}{i}$ 。通过推导得到，对于双向查询系统的平均路径长度为 $X/2b + \theta(1)$ ；对于单向查找的Chord系统平均路径长度为

$Y/2b=b/2$ 。

如果 $m$ 为奇数，则 $X=X_{\text{odd}}$ ，否则 $X=X_{\text{even}}$ ，并且

$$X_{\text{odd}} = \sum_{i=0}^{(m-1)/2} i \binom{m}{i} + \sum_{i=(m+1)/2}^m (m-i) \binom{m}{i};$$

$$X_{\text{even}} = \sum_{i=0}^{m/2} i \binom{m}{i} + \sum_{i=b/2+1}^m (b-i) \binom{m}{i};$$

可以得到，

$$T - X_{\text{odd}} = \sum_{i=(m+1)/2}^m (2i-m) \binom{m}{i};$$

$$T - X_{\text{even}} = \sum_{i=(m+2)/2}^m (2i-m) \binom{m}{i};$$

根据 Stirling 近似，可以得到  $T - X_{\text{odd}} \approx [\sqrt{(m+1)/2\pi} + \theta(1)] 2^m$ ，以及  $T - X_{\text{even}} \approx [\sqrt{m/2\pi} + \theta(1)] 2^m$ ，即对于 $m$ 取偶数或者奇数时，双向查询Chord所减少的路由跳数，因此得到平均路径长度是  $m/2 - \sqrt{m/2\pi} + \theta(1)$ ；由于双向查询过程只在半个环上进行，因此最大路径长度是  $\lceil m/2 \rceil$ 。

### 3.4 仿真与分析

这里通过仿真来分析基于延迟敏感的双向查询Chord系统的性能。设path length是查询所经过的路由跳数，latency of each hop是每一跳的延迟，则系统中完成一个查询所消耗的时间是：

$$\text{path latency} = \text{path length} \times (\text{latency of each hop})$$

基于延迟敏感的双向查询Chord系统分别从两个方面对查询时间进行了优化，首先，双向查询Chord系统可以减少查询信息被转发的次数，相应的减少了路由跳数；其次，基于延迟敏感的系统可以减少每一跳的平均响应时间。

本章使用BRITE<sup>[106]</sup>作为拓扑产生器生成二层拓扑结构。BRITE有四种类型的拓扑参数，这里选择按照bottom-up类型生成。按照bottom-up类型参数生成的拓扑有两层，分别是Router-level和节点，并且每一层都有两种生成模式。实验中Router-level拓扑使用Waxman模式，router节点按照heavy-tailed的方法被分配到自治系统中，在此基础上用Java实现Chord系统。

#### 3.4.1 基于延迟敏感的节点划分对网络性能的影响

本实验验证参考点对于网络性能的影响，所用到的性能指标是gain ratio。一个节点分别获得与同一分组内以及不同分组内任意节点之间的延迟，并把两个结果做比较得到该节点的gain ratio。系统中所有节点gain ratio的平均值(average gain ratio)可以用来比较网络节点

划分的有效性。图3.6是参考点个数为4时基于延迟敏感和随机划分的系统的average gain ratio比较。通过观察可以发现,对于随机划分节点的系统,节点的latency ratio分布在1周围,即节点对于分别在组内和组外随机选取的节点之间的延迟差别不大,而基于延迟敏感的系统,节点与组内和组外随机选取的节点相比,延迟明显减小。实验结果表明, gain ratio分布在0.7附近。因此,基于延迟敏感的划分策略可以明显减少同一Chord内节点之间的延迟。同时,对于一个给定的拓扑,节点的密集程度不会影响系统的average gain ratio。

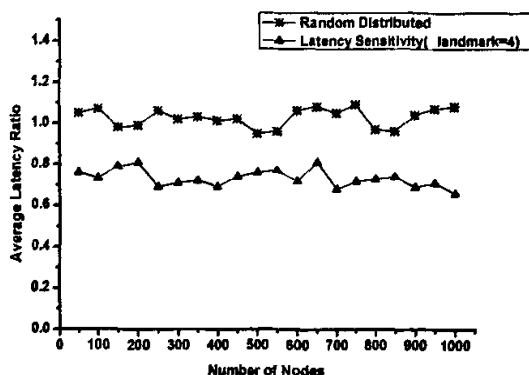


图3.6 随机分布的系统和基于延迟敏感的系统比较

### 3.4.2 双向查询Chord对路径长度的影响

一般的Chord系统可以在 $O(\log N)$ 内完成查询,其中 $N$ 为系统中节点的个数,也就是说最大路径长度和 $\log N$ 呈线性关系,而且它的平均路径长度是 $\log N/2$ 。在实现双向查询的Chord环上的最大路径长度是 $\lceil m/2 \rceil$ ,这里的 $m$ 是用二进制表示的节点的位数,并且平均路径长度是 $m/2 - \sqrt{m/2\pi}$ 。

为了验证这个结论,我们首先获得延迟敏感的Chord上的平均查询时间。因为Chord上的节点是基于延迟敏感的,根据3.1节所获得的结果,可以把位于同一Chord中的不同节点之间的延迟看作相等。这样,Chord中的平均路径长度可以通过平均查询时间和单跳的延迟之比来获得。我们在这里使Chord上的节点数目分布在1000到10000之间,并取 $m$ 的值为14,从而可以提供足够大的标识符空间。仿真试验结果如图3.7所示。随着节点数目的增加,单向查询算法的Chord的平均路径长度是和 $\log N/2$ 呈正比增长的,而双向查询算法的Chord平均路径长度保持在 $m/2 - \sqrt{m/2\pi}$ 附近,基本符合3.2.4节的证明结果,有效地把路径长度从一般Chord的对数增长控制在了常数范围内。

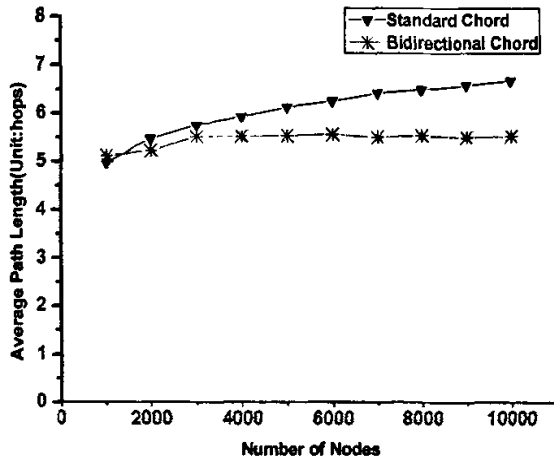


图3.7 Chord与双向查询Chord比较

### 3.4.3 两个系统的比较

基于延迟敏感的双向查询Chord系统,根据节点和参考点之间的延迟对这些节点进行划分,降低了同一分组内节点之间的延迟;通过实现Chord上双向查询算法减少路径长度,这样降低了整个系统的平均查询响应时间。如图3.8所示,基于延迟敏感的双向查询Chord系统与随机分布的单向Chord系统相比,执行查询的平均时间明显减少。但是,我们还发现,在节点达到一定数目之前,随机分布的单向Chord系统中查询时间是相对较短的。这是因为:

- a)和随机分布相比,基于延迟敏感的系统虽然有效的减少了组内节点之间的延迟,但势必会增大位于不同分组节点之间的延迟;
- b)双向查询Chord系统只是减少了同一Chord上的平均路径长度。因此当节点数目不大,不同分组内节点相对稀少,并且发起的查询具有随机性时,对结果的搜索更多的在不同Chord上执行,查询时间相对较长。随着网络规模的扩大,查询会集中在各自分组内进行,平均查询时间相对减少很多,并趋向于图3.7的结果。和图3.7的结果相比,图3.8和理论结果有差异是由于选取了不同的节点范围,这里存在部分不同分组之间的查询,而影响了平均查询效率。这种情况可以根据不同需求,改变分组的规模(通过改变阈值,见第3.3.1小节),做适当调整。

## 3.5 本章小结

本章提出的基于延迟敏感的双向查询Chord系统分别从路由长度和节点之间的延迟两个方面来减少查询时间。根据节点和参考点之间的网络延迟来对节点进行分组,可以使网络上相近的节点作为邻居从而减少延迟。针对Chord中查询采用单向搜索的不足,本章采用双向查询的方法,充分利用节点逆时针方向上的邻居信息,可以把Chord上查询的平均路由长

度减少到 $m/2 - \sqrt{m/2\pi}$ ，并且最大路由长度减少到 $\lceil m/2 \rceil$ 。双向查询Chord系统通过增加一步判断操作大大减少了网络中信息的转发次数，从而减轻了网络的负担。但是这种双向查询

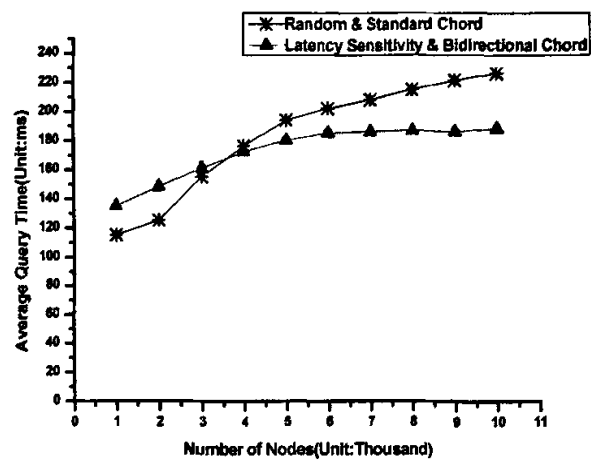


图3.8 两个系统的比较

只可以减少在查找位于节点逆时针邻居上数据时的路径长度，而对位于节点对剖方向上的节点的搜索路径改进不明显，因此在今后的工作中需要考虑复制或者其他相关技术来进行改善。同时，本系统在设计时增加了维护逆向邻居信息的指针表，可能带来了维护指针表的开销，在将来的工作中可以考虑改进的方法。对于P2P系统中由于节点匿名性所带来的安全性问题，本文将在下一章提出基于信誉感知的资源发现算法抑制P2P网络中恶意节点的欺诈行为。

## 第四章 P2P网络中基于信誉感知的资源发现算法

P2P节点的异构性以及匿名性无法保证节点都提供可靠的服务和资源,而叠加层构建的对等性使得这些不可靠节点成为影响资源管理性能的主要问题之一。本文提出的基于信誉感知的资源发现算法依据历史交易信息对每个节点评定信誉度,在智能选取邻居集以及选择服务响应节点的时候考虑信誉因素,有效地抑制了P2P系统中恶意节点的欺诈行为,降低了系统的消息负载,并保证了服务的可靠性和安全性。

### 4.1 引言

P2P网络是用来进行数据共享、协同计算和获取大容量存储的一种分布式应用模式。P2P网络中不存在中央服务器,每个节点同时具备资源提供、资源消费和信息通讯三个方面的功能,节点和节点之间不需通过中间代理实体即可直接互相访问。P2P网络的动态性、自组织性和匿名性不能保证所有的相应节点都提供诚实良好的服务和可靠资源,一些节点不提供其所声称的服务,某些节点甚至提供恶意欺诈服务,对于服务的请求者以及其他节点造成危害,从而导致所发现资源的非法性和不确定性。因此,在面向大量动态用户的匿名的P2P网络环境中,如何在资源搜索的过程中避免这些恶意节点的欺诈服务,保证服务提供节点提供可靠资源和服务成为了一个亟需解决的问题。本章基于Directed-BFS算法<sup>[74]</sup>,融入信誉的概念,提出基于信誉感知的资源发现算法: RARDA (Reputation-Aware Resource Discovery Algorithm),以保证请求节点得到可靠的资源和服务。模拟实验表明,该算法有效地抑制了P2P网络中恶意节点的欺诈行为,降低了系统的消息负载,保证了节点获得服务的可靠性和安全性。

本章其余部分安排如下:第二部分介绍目前P2P网络中的相关资源搜索算法;第三部分描述了信任以及基于投票评估的信誉值计算;第四部分基于Directed-BFS算法,融入信誉因素,提出基于信誉感知的启发式资源发现算法RARDA;第五部分为仿真及结果分析;最后是小结。

### 4.2 相关工作

P2P应用目前主要有文件共享、协同处理、即时通信三个方面,所有的应用都不外乎两个方面:资源发现和资源使用。因此,资源发现是P2P系统中一个基本的服务。所谓的资源发现主要是为了发现资源的结构和状态信息,给出期望资源的一个具体的描述,发现一系列和资源描述相匹配的资源。



完全分布式P2P系统中文件的发布和网络拓扑松散相关, 资源定位可以采用BFS(Breadth-First Traversal)技术。BFS技术就是每个节点接收到一个查询消息后, 使用洪泛Flooding算法, 把查询消息传递给所有的邻居, 直到查询消息达到了预先确定的跳数。其Flooding算法查询过程如图4.1所示。

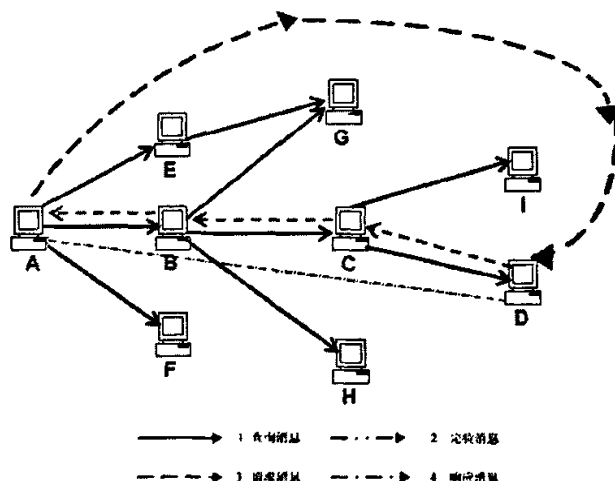


图4.1 洪泛算法

图4.1所示的Flooding资源定位方法只适应于以发起者为中心的某个特点半径范围内, 当超出这个范围时就无法继续查询, 因此Flooding方法不能保证查询的有效性。但Flooding资源定位方法是一种完全分布式的, 它能有效地避免单点失败问题。

Flooding洪泛算法采用了广播机制来进行查询, 从理论上讲是理想的, 因为它把查询请求尽可能快地发送给每个可能的节点; 但是算法所产生的数据流量按指数级增长, 浪费了大量的资源以及网络带宽, 造成网络上被大量的查询广播消息所占用, 系统没有很好的可扩展性。因此在[74]中, 提出了折衷而不降低查询结果质量的方法: Directed-BFS算法。Directed-BFS算法利用了如下的策略: 把查询消息依据某些历史经验传给其邻居的一个子集, 因而减少了网络开销。这种技术关键在于智能选取邻居节点达到不降低查询结果质量的目的。关于Directed-BFS算法效率模拟在[74]中具有详尽的描述。但是Directed-BFS算法并不能保证每个资源提供节点都提供可靠良好的服务, 会存在一些恶意节点和伪装节点提供欺诈和不可靠服务<sup>[107]</sup>。为了保证P2P系统的可靠性, 保证QoS, 本文改进Directed-BFS算法, 依据历史交易信息对每个P2P节点评定信誉度, 在智能选取邻居集以及选择服务响应节点的时候考虑信誉因素, 提出基于信誉感知的启发式资源发现算法。目的是为了获取可信节点提供的诚实服务和降低网络通信量, 提高P2P系统的可扩展性。

### 4.3 信誉和信誉值计算

本章采用Diego Gambetta给出的信任定义<sup>[108]</sup>, 节点间的信任关系分为两类: 直接信任(Direct Trust)和推荐信任(Recommendation Trust)<sup>[109]</sup>。直接信任是指两个实体之间曾经

有过直接的交易, 根据双方的交易情况得出的直接经验来评估其信任度。推荐信任是指两个实体之间没有进行过直接的交易, 而是根据其他实体的推荐建立的一种信任关系, 它们之间的信任度是根据其他实体的评估得出的结果, 而且两节点之间的信任关系是单向、不可逆的。

本章采用信誉值(ReputationValue)和信任类别(TrustClass)的概念来描述每一对节点之间的信任关系。信任类别是服务提供者提供的某类服务。信誉值是基于某种信任类别的两节点之间的信任程度, 是对节点之间信任的量化的评价。其信誉值计算公式可形式化描述为:

$$\text{ReputationValue}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) = \lambda \times \text{DRV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) + (1 - \lambda) \times \text{RRV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) \quad (4.1)$$

$\text{DRV}$  表示两节点之间直接信誉值,  $\text{RRV}$  表示任两节点之间的推荐信誉值,  $\lambda$  表示一个节点对于直接交易评估的信任程度。  $\text{ReputationValue}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B})$  可以简单描述为  $\text{RV}_{\text{A}}(\text{B})$ 。

### 4.3.1 直接信誉值计算

两节点之间的直接信誉值根据历史交易信息进行投票评分, 从而得到的一个节点对另外节点的在某种行为上的信任能力, 形式化描述为  $\text{DRV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B})$ 。其信誉值的计算公式表示为:

$$\text{DRV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) = \frac{1}{n} \times \sum_n \text{Score}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) \quad (4.2)$$

其中  $\text{Score}_{\text{A}}^{\text{TrustClass}_A^B}$  表示一次历史交易中节点A对节点B的某种行为投票评分值,

$\text{Score}_{\text{A}}^{\text{TrustClass}_A^B} \in [-1, 1]$ 。如对待节点A每次从B下载文件时, 如果下载的文件可信, 则取评估值为正数; 如果被下载的文件是不可信的, 或是被篡改过的, 或者下载被中断等, 则评估值取负数。 $n$  为节点A和节点B之间的历史交易的次数

### 4.3.2 推荐信誉值计算

节点除了根据节点的历史经验获得直接经验外, 还会接受其他推荐节点对目标节点的推荐。本文利用信誉值传递的方法来进行推荐信任的合成, 其形式化的描述为  $\text{RRV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B})$ 。节点A有多条路径都可以到达节点B, 本文假设在A和B之间有三条推荐路径, 推荐路径如图4.2所示。 $\alpha, \beta, \gamma$  分别为每条推荐路径的权重,  $\alpha + \beta + \gamma = 1$ 。其推荐信誉值公式表示为:

$$\text{RRV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) = \alpha * (\text{RV}_{\text{A}}^m * \text{RV}_m^B) + \beta * (\text{RV}_{\text{A}}^j * \text{RV}_j^k * \text{RV}_k^B) + \gamma * (\text{RV}_{\text{A}}^n * \text{RV}_n^B) \quad (4.3)$$

综上所述, 请求节点A和响应节点B之间的信誉  $\text{RV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B})$  为:

$$\text{RV}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) = \frac{\lambda}{n} \times \sum_n \text{Score}_{\text{A}}^{\text{TrustClass}_A^B}(\text{B}) + (1 - \lambda) \times (\alpha * (\text{RV}_{\text{A}}^m * \text{RV}_m^B) + \beta * (\text{RV}_{\text{A}}^j * \text{RV}_j^k * \text{RV}_k^B) + \gamma * (\text{RV}_{\text{A}}^n * \text{RV}_n^B)) \quad (4.4)$$

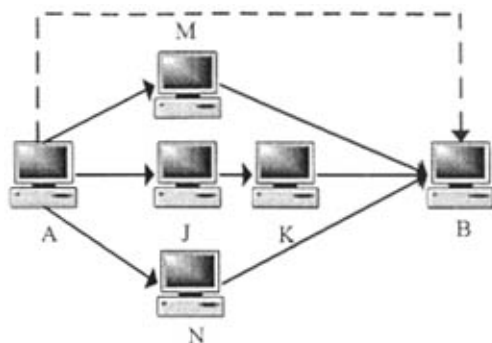


图4.2 多路径推荐

## 4.4 启发式资源发现算法

P2P系统没有固定的网络拓扑结构,其网络拓扑结构是在节点的加入和退出中动态自发形成的。系统中的每个节点在加入和退出系统时都要发送消息给其邻居节点,使得P2P系统能够动态的进行拓扑更新。基于信誉的P2P系统对原有的系统进行了扩展,其扩展结构如图4.3所示。

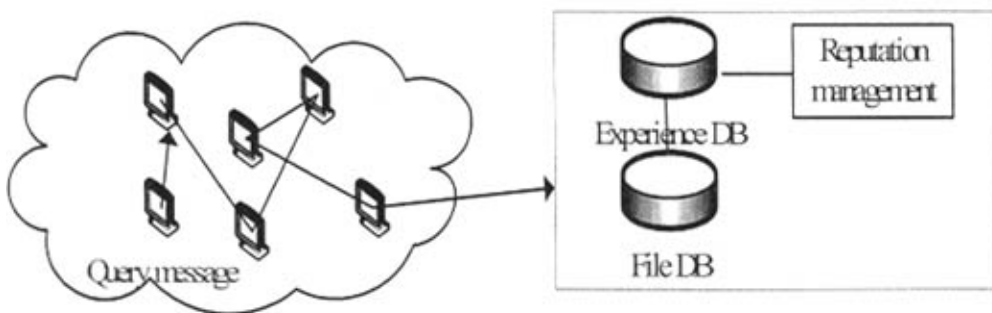


图4.3 扩展的P2P结构图

扩展的P2P系统中每个节点除了存储文件共享信息库之外,还扩展了一个历史经验信息库,通过信誉管理机制对经验信息库中的数据进行管理。P2P系统中的每个节点具体维护如下的数据结构,如图4.4所示。

- File Table** 存储了文件的哈希关键字及其一些描述信息。
- Neighbor Table** 存储了其邻居节点标识及交互的历史信息,包括邻居节点的消息转发次数和消息响应提供服务次数。
- Experience Table** 存储和当前节点有过交易历史节点的地址、信任类型和交易反馈信息以及时间衰减因子等。信誉值会随着时间的增加而逐渐衰减。

**Reputation Table**根据信誉值的计算公式存储节点之间的信任程度。

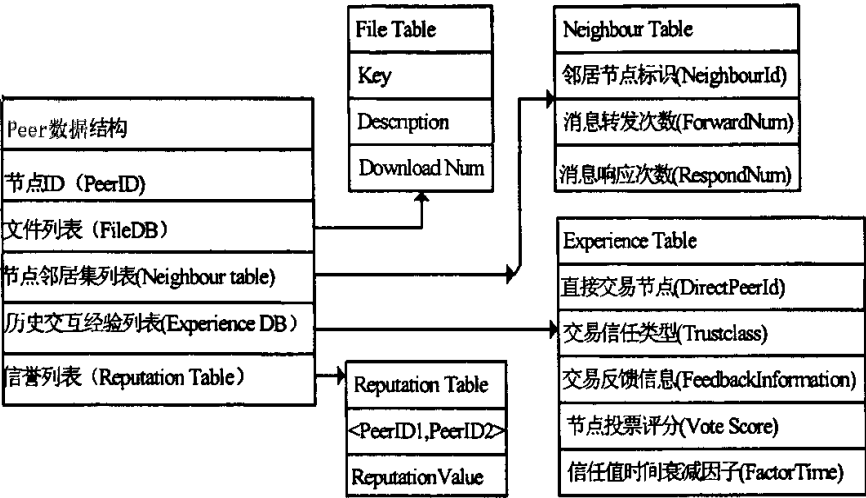


图4.4 节点数据结构

4.4.1 资源搜索流程

文献[74]中所提到的Directed-BFS算法相对于Flooding算法在网络带宽消耗、处理代价方面都有了一定的提高,但不能保证搜索到的服务提供节点都能提供可信的服务和可靠的资源。因此,本文基于Directed-BFS算法,融合信誉,提出信誉感知的启发式资源发现算法,其信誉的使用表现在以下三个方面:

- a、依据信誉值智能选取消息转发邻居集
- b、多个节点响应请求提供服务,选择信誉值较高的节点作为资源服务节点
- c、转发节点如没有保存邻居节点的信誉值,则依据公式(4.4)进行对节点信誉列表进行计算更新。

下面给出改进后的资源发现算法,如图4.5。

其流程描述如下:

(1)请求节点发送查询消息,  $Message=\{SourcePeerId, FileId, Rv_{min}\}$ ,  $SourcePeerId$  表示请求节点标识符,  $FileId$ 表示请求资源标识符,  $Rv_{min}$  表示请求节点需要的最低信誉值。

(2)选择其迭代加深策略,如策略为  $Policy=\{Depth1,Depth2\}$ 。首先选  $Policy.Depth1$  作为迭代洪泛的跳数。

(3)根据经验列表历史交易信息,搜索历史的直接交易节点,是否这些节点中存在满足请求的资源。如存在满足请求的节点,且誉值大于  $Rv_{min}$ ,则响应。否则,更新节点的信誉列表,选择信誉值大于等于请求消息  $Rv_{min}$  的节点作为消息转发的邻居集子集,然后在邻居集子集中选择消息响应次数或者消息转发次数较多的节点作为消息转发的节点。

(4)判断是否有节点满足查询请求,如果存在多个节点满足条件,则选择信任值较高的节点响应请求,算法结束,否则,转到(5)。

(5)判断迭代的跳数是否超过 Policy.Depth1, 如果没有超过, 则转到(2)。否则, 重发Resend消息, 冻结 Policy.Depth1 时的所有搜索节点, 转到(1), 选择 Policy.Depth2 作为迭代洪泛的跳数, 重复步骤(2)(3)。

(6)超时或者是迭代深度结束, 没有响应资源请求, 则利用简单Flooding算法查询。

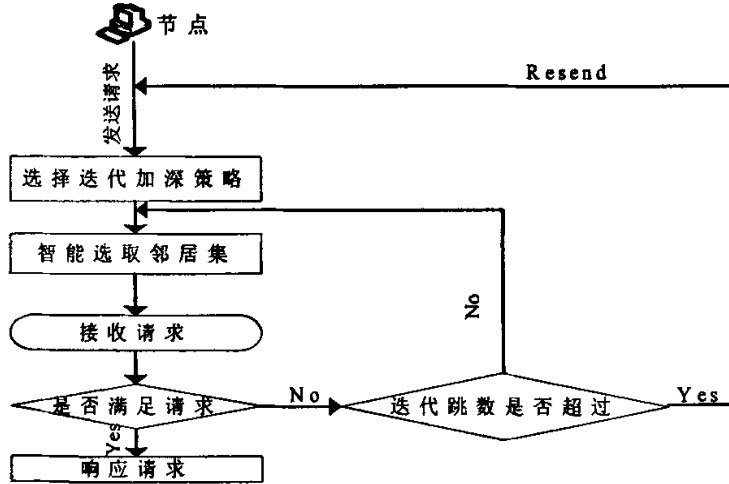


图4.5 算法流程图

#### 4.4.2 伪代码描述

基于信誉感知的启发式算法进行消息的转发定位时并没有改变网络的底层路由算法, 而是在底层的网络拓扑结构上添加了一个额外的叠层<sup>[110]</sup>(Overlay), 并且本文所提出的算法也是覆盖在Flooding算法基础之上, 在最坏的情况下, 算法执行简单Flooding算法, 因此具有较强的灵活性, 便于算法进行移植。本文是通过对节点之间的历史交易信息进行投票评分来计算节点之间的信任程度。算法既有效的避免欺诈节点提供恶意或虚假服务, 保证了服务质量(QoS)的需求, 也降低了网络负载, 增强了系统的可扩展性。其伪代码算法如图4.6所示。

```

policy p={a,b,c}
Message={ SourcePeerID, FileId, RVmin}
Resource-Discovery Algorithm(Message):
do{
(1) hop=SelectPolicyDepth(p);/*选择迭代深度*/
(2) while(i<=hop){
(3) IntelligentSelectNeighbourSets(peers);/*智能选取邻居节点集*/
(4) SendMessage(Message);
(5) if(FileId exist Peer) return success;
(6) else
(7) i=i+1;}
(8) Frozen(PastForwardPeers);/*冻结过去搜索过的节点*/

```

```

(9) Resend(message);
}
(10) until((i>=SelectPolicy)or(overtime))
(11) Flooding(FileID);/*没有发现资源, 使用洪泛算法*/

```

图4.6 RARDA的伪代码

当一个请求消息发出时, 资源请求节点选择迭代加深策略, 查找节点经验列表和邻居列表。如果节点文件列表存在请求资源并且满足信誉等级的要求, 则节点响应消息; 否则, 节点依据信誉列表和邻居节点响应次数和转发次数智能选择消息转发邻居节点。如信誉列表中没有保存相关两节点之间的信誉值, 则根据公式(4.4)进行推荐信誉值的计算。

### 4.4.3 算法性能分析

RDARA算法在保证节点提供可靠诚实服务的同时能够减少网络中的通信量, 下面具体分析算法性能。

假设P2P网络中的任意节点A和B之间存在边,  $S(A)$ 表示节点A及其邻居节点集合。如果节点A为请求节点, 向其邻居节点发送消息表示为  $S(A)^{translate}$ ,  $S(A)^{translate}$  表示通过智能选择策略所选择的节点A的转发邻居节点, 经过一次循环, 搜索节点数为:  $S(A)^{translate} \cup S(B)$ 。算法在最坏的情况下执行Flooding算法, 假设网络生成的连通图最大直径用  $d(n)$  (节点数为  $n$  的网络直径) 进行表示, 则其循环复杂度即为  $d(n)$ 。

算法的通信复杂度依赖于网络中的边的个数, 用  $E_{num}(n)$  表示节点数为  $n$  的网络中初始的边的个数, 则最坏情况下, 网络中的所有节点同时进行资源请求, 并且都经过最大的路径长度, 即为  $d(n)$ , 因此, 网络的通信复杂度为  $\Omega(E_{num}(n) \times d(n))$ 。

## 4.5 仿真及结果分析

本文采用BRITE<sup>[106]</sup>网络结构生成器, BRITE生成网络结构与实际网络结构相似, 而且符合power laws规则<sup>[39]</sup>:  $y = x^\alpha$ 。仿真的网络拓扑结构是采用自底向上的两层层次拓扑结构。整个结构生成1000个节点, 3个自治域, 其路由结构使用Waxman模式, 最大带宽和最小带宽分别为1024kbps和10kbps, 利用CAIDA的可视化显示工具Otter0.9显示网络拓扑结构。

在实验中, 以非结构化的文件共享系统Gnutella协议为基础进行设计评测。本文假设10000个文件随机的分布在这1000个节点上, 假设每个节点分配100个互不相同的共享文件, 不予考虑节点之间的信任类型, 任意两节点之间的信任值仅取决于历史交易评估信息。仿真试验分别实现了Flooding算法、Directed-BFS算法以及RARDA算法。

图4.8表示三种资源搜索算法下载不同数量文件时, 响应节点中诚实节点占有的比率。实验结果表明, 基于信誉感知的启发算法能够保证更多的下载资源由信任可靠的响应节点提供, 保证QoS服务。图4.9表示欺诈节点所占比例不同的情况下系统所能下载诚实文件的比例。其结果表明, 在信誉感知启发搜索算法中, 即使在不诚实节点所占的比例为90%的情况下, 仍然可以下载到可靠诚信的文件。

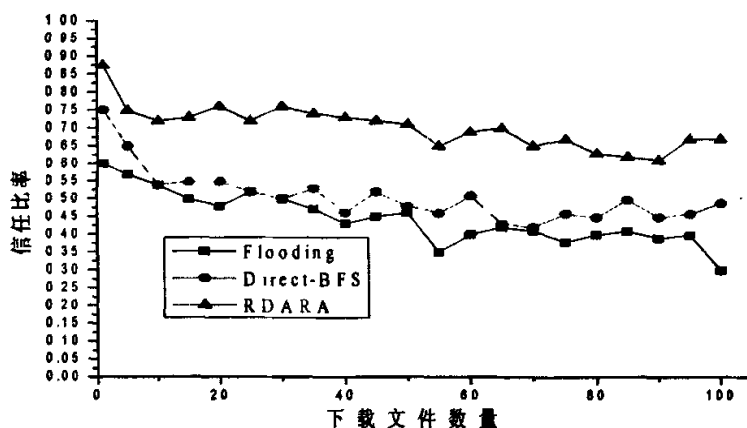


图4.8 信任节点比率

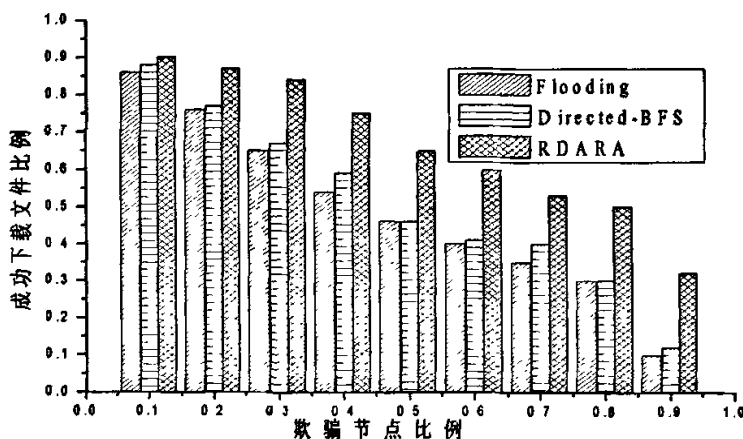


图4.9 欺诈节点比例对成功下载文件的影响

成功率是在某个时间段中查询成功的次数与这段时间中总的查询次数之比, 图4.10的三条曲线起伏不大, 比较接近, 则表明RDARA算法保证了资源搜索的命中率, 算法具有较好的可用性。图4.11为系统查询负载比较, 即为算法执行时系统中的消息数目。结果显示, RDARA算法大大降低了系统消息的负载, 增强了系统可扩展性。

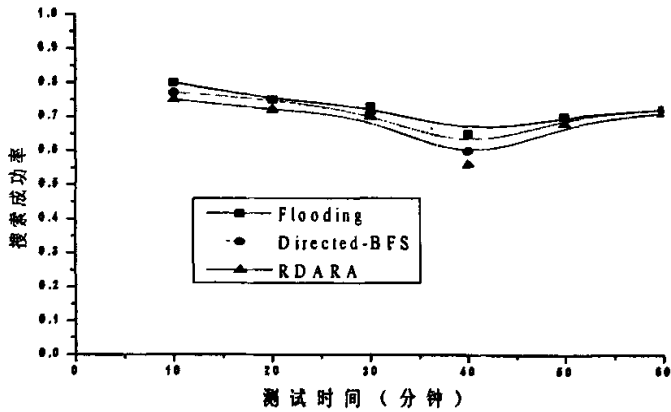


图4.10 资源搜索成功率

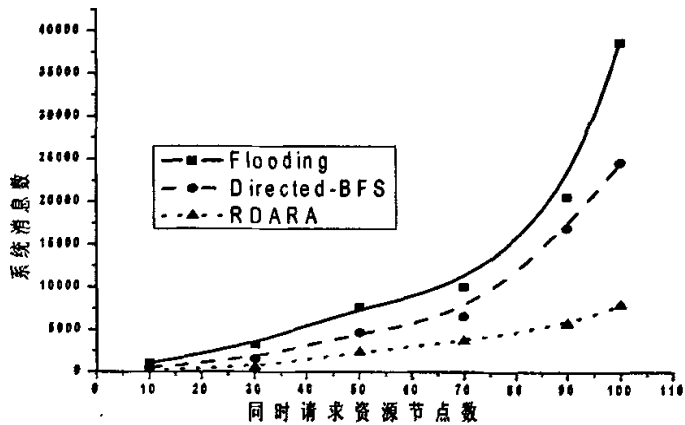


图4.11 系统消息负载比率

由以上实验结果分析可以得出，在基于信誉感知的启发式资源搜索算法中，能够一定程度的发现更多的不信任、不可靠的节点的数目，从而使得资源搜索时避免选择这些不可靠节点，减少了恶意节点的欺诈行为和虚假服务，提高了系统的可靠性和可扩展性。

## 4.6 本章小结

本章针对P2P系统中资源搜索所面临的节点欺诈和虚假服务的问题，提出了非结构化P2P中一种基于信誉感知的启发式的资源搜索算法。该算法是洪泛算法与信誉机制的有效结合，保证了搜索到资源的安全性和可靠性，减少了恶意节点的欺诈行为和虚假服务，提供了可靠的QoS服务。实验表明，该算法对于降低网络负载、有效抵御恶意节点的虚假服务都有较好的性能，但是对于女巫攻击问题却并不是非常有效。本文下一章将提出基于小世界模型的女巫攻击防御机制，用于控制女巫攻击对于系统的影响。



## 第五章 基于小世界模型的女巫攻击防御机制

P2P节点的异构性以及匿名性无法保证节点都提供可靠的服务和资源,而叠加层构建的对等性使得这些不可靠节点成为影响资源管理性能的主要问题之一。本文提出的基于信誉感知的资源发现算法依据历史交易信息对每个节点评定信誉度,在智能选取邻居集以及选择服务响应节点的时候考虑信誉因素,有效地抑制了P2P系统中恶意节点的欺诈行为,降低了系统的消息负载,并保证了服务的可靠性和安全性。

### 5.1 引言

和传统的C/S模型相比,P2P<sup>[75]</sup>弱化了服务器的概念。但是P2P系统的开放性和匿名性使得系统容易遭受攻击。并且随着P2P系统规模的扩大,恶意节点的出现也非常普遍。针对P2P系统的安全问题,已经展开了很多研究<sup>[76]</sup>。像数字签名VeriSign<sup>[77]</sup>一样的标识符认证是比较有效的安全策略。比如,CFS协作存储系统通过对节点的IP地址进行哈希后得到的标识符实现对每个节点的认证。SFS网络文件系统<sup>[78]</sup>通过为一个DNS域名附加一个主机标识符实现对远程路径的命名。但是上述系统都需要一个中心认证机构实现标识符的分配,并且在处理女巫攻击问题时并不十分有效。

本文提出一种基于小世界模型<sup>[79]</sup>的女巫攻击防御机制。通过把找出女巫攻击团体归结为最大流/最小割问题,本文引入虚拟节点,并使用爬行器找到女巫攻击团体。这种方法有效地将女巫攻击者和P2P系统分开。模拟结果显示这种机制明显地减少了整个系统中女巫攻击节点的百分比。

第二部分是相关工作;第三部分介绍了女巫攻击;第四部分阐述如何设计女巫攻击防御系统;第五和第六部分分别是试验和结论。

### 5.2 相关研究工作

到目前为止,针对女巫攻击问题已经展开了很多研究<sup>[80][81][82][83]</sup>。一个信任的中心认证机构通过为每个人分配唯一的标识符能够有效地组织女巫攻击。这种中心认证结构可能同时会需要为每个标识符付出一定代价。但是,这种防御机制在很多情况下不适用。比如,很难选择或者建立一个整个系统都信任的独立实体。同时,中心认证机构会成为一个瓶颈,并可能引起QoS攻击。

对于女巫攻击问题的最初研究结果有些不尽人意,因为当时的研究都是基于很多假设条件的。目前大多数研究都是依靠多个独立的远程实体来防御恶意节点的攻击,比如

EigenTrust<sup>[99]</sup>。在文献[99]中详细论述了采用EigenTrust算法如何降低P2P网络中非授权文件的共享，从网络中隔离恶意节点等一系列节点信誉问题。尽管它可以将P2P系统和很多恶意节点隔离开，但是对于女巫攻击问题却没有很好的解决。

### 5.3 女巫攻击

在具有可信任的认证中心节点的P2P网络环境中，一个未知远程节点呈现不同的身份是不可能的。但在没有逻辑中心和可信任认证中心节点的完全分布式P2P网络里，节点和节点之间进行一对一的通信，无法对对方身份进行认证，这样使得未知远程节点可以表现为不同的身份。如果本地节点和未知远程节点没有直接的物理连接，本地节点便会把远程节点仅仅看作信息的抽象，我们这里称之为身份。P2P系统必须保证不同的身份属于不同的节点。否则，当一个本地节点错误的选择了一个恶意主机进行文件下载时，恶意主机就会冒充好节点向它发送非法数据。结果，好节点的信誉值降低，网络中的资源会被浪费。

P2P系统里这种伪造多个身份的攻击就被称为女巫攻击。这种攻击产生的主要原因是用户无需任何代价就可以创建新的身份和节点。但是攻击者就利用了这种漏洞在网络中传输非法文件，从而破坏系统中文件的安全性，消耗了网络中节点的连接资源而无需担心自己会受到影响。

图5.1给出了分布式网络模型。其中包括以下几个方面：

- ✧ 节点 $e$ 的集合 $E$ ；
- ✧ 广播通信的网络；
- ✧ 将每个节点与通信网络相连接的管道。

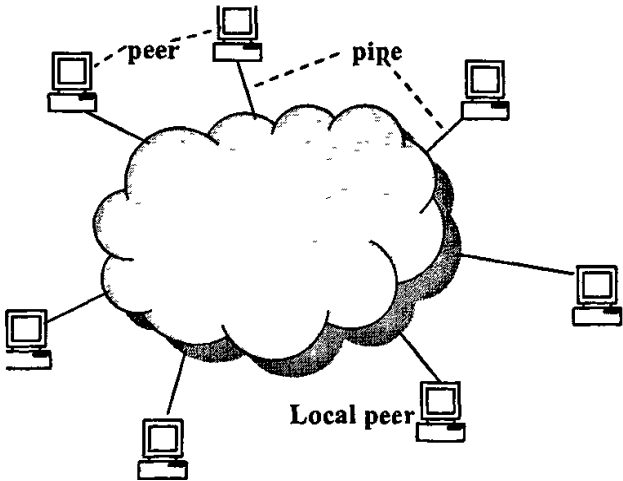


图5.1 分布式系统模型

集合 $E$ 分为两个不同的子集， $C$ 和 $F$ 。在集合 $C$ 里的每个节点 $c$ 都是正常节点，遵循所定义的网络协议规则。集合 $F$ 里的每个节点 $f$ 都是恶意节点。节点间通过消息机制完成通信。节

点之间通过消息进行通信。一个消息是一个连续的、有限长度的字符串，它所代表的内容是根据节点之间明确的协议或者暗含的规定决定的。一个节点可以通过管道发送一个消息，并且把它广播给其他节点。这个消息会在规定的时间间隔内被所有的节点收到。

每个节点试图向系统中的其他节点表明自己的身份。假设有一个本地的诚实节点 $h$ ，具有身份 $l$ ，如果节点 $e$ 将它的身份 $i$ 成功地告知节点 $h$ ，就认为身份 $l$ 接受了身份 $i$ 。

所有的诚实节点以及恶意节点都试图表明自己有一个合法身份。但是，恶意节点还会有多个伪造的身份，这种攻击称为女巫攻击。本文针对女巫攻击提出一种防御系统。

## 5.4 女巫攻击防御系统的设计

本文基于对社会网络的分析和研究提出一种新的女巫攻击防御机制。利用社会关系网络图，节点代表身份，边代表人为的信任关系。连接诚实团体和女巫攻击团体的边是女巫攻击边。本文设计的基本想法是如果恶意节点创建了太多的女巫攻击身份，关系网络图的节点分布会受到很大影响，也就是出现最小割，将女巫攻击节点和整个系统分离。直接寻找这样的最小割非常困难，因为无法获取整个网络的拓扑结构以及每条边的端点。即使可以获得整个网络的拓扑，寻找最小割仍然是一个NP问题。

在这一部分，首先介绍什么是最大流/最小割问题，然后是如何找到理想状态下的女巫攻击团体，并且通过引入虚拟节点利用爬行者获得现实状态下的女巫攻击团体，最后是爬行器的实现算法。

### 5.4.1 最大流/最小割问题

$s$ - $t$ 最大流问题是这样描述的。对于一个给定的有向图 $G=(V, E)$ ，其中每条边 $(u, v) \in E$ 均有一个非负容量 $c(u, v) \geq 0$ 。 $G$ 中存在两个特殊点，源点 $s$ 和汇点 $t$ ，利用 $s$ 和 $t$ 之间不同边的容量找到从 $s$ 到 $t$ 的最大流即为 $s$ - $t$ 最大流问题。可以形象的描述为，如果边是输水管道，点是临时储备库，那么最大流问题就是寻找从源点到汇点最多可以输送的水的流量。

Ford和Fulkerson提出的“最大流/最小割”理论证明了寻找一个网络的最大流问题等同于划分 $s$ 和 $t$ 的最小割问题。因此，本文基于“最大流/最小割”理论给出了基于最小割问题的社会团体的定义。

**定义5.1** 如果 $[S, T]$ 是图 $G(V, E)$ 中的最小割，并且 $s \in S$ ， $t \in T$ ，那么 $S$ 是相对与 $t$ 的一个社会团体。

对于最大流问题的大多数解决方案都是基于一个假设，就是整个图的结果可以很容易的获得。这显然不适用于动态的P2P系统，因为P2P系统的整体结构是随着节点的自由加入和离开而不断变化的。本文选择最短增广路径算法找出P2P系统中源点和汇点之间的最短路径，因为它只需要利用系统中的部分拓扑结构就可以有效地解决最短路径问题。

## 5.4.2 女巫攻击团体

这一部分将详细阐述女巫攻击团体的定义以及如何识别女巫攻击。首先,给出理想状态下女巫攻击团体的发现,然后是如何引入虚拟节点并利用爬行器寻找现实状态下的女巫攻击团体。

### 5.4.2.1 理想状态

这一部分定义无向图中的通信,其中每条边都具有单位容量。在理想状态下,每条边都被用于计算并且不具有平凡割,首先给出一些定义。

**定义5.2** 女巫攻击团体是一个节点集合  $G' \subseteq E$ , 对于集合中的所有节点  $v \in G'$ ,  $v$  在  $G'$  中具有的连接数至少等于它和  $(V-G')$  中节点所具有的连接数。

**定义5.3** 给出变量  $s'$ , 代表  $s$  和集合  $(G'-s)$  中节点之间的连接数; 同样, 变量  $t'$ , 代表  $t$  和集合  $(V-G'-t)$  中节点之间的连接数。

**结论5.1** 寻找P2P系统中的女巫攻击团体  $G'$ , 可以转化为寻找  $G$  中以  $s$  为源点,  $t$  为汇点的  $s$ - $t$  最小割问题, 其中  $s'$ ,  $t'$  要大于最小割的大小。一旦最小割的边被删除, 所有对于  $s$  可达的点都在女巫攻击团体里。

这个结论可以用反证法验证, 此处不做详细介绍。

图5.2给出了将女巫攻击团体和整个系统分割开的例子。在结论1中给  $s'$  和  $t'$  的条件是为了避免平凡割问题。但是一个女巫攻击团体会包括很多类似的平凡割, 为了解决这个问题, 本文引入一个虚拟节点, 并且选择多个种子节点和虚拟节点相连。其中种子节点和虚拟节点之间的容量为无穷大。

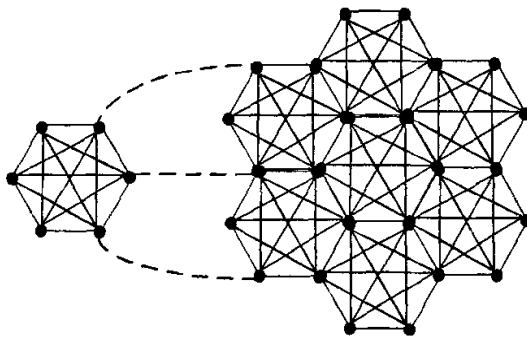


图5.2 基于最小割的女巫攻击团体

### 5.4.2.2 现实环境中女巫攻击团体的发现

在构造P2P系统时可以基于Jon Kleinberg模型, 因为很多相关研究<sup>[39][41]</sup>表明, P2P网络拓扑是符合小世界模型的, 因此节点的入度和出度可以遵循power-law分布。

在理想状态下寻找女巫攻击团体的主要问题是需要快速获取很多节点的入度和出度信

息。图5.3显示了本文的爬行器如何寻找属于女巫攻击团体的节点。首先,选择一些种子节点,如图5.3所示。然后,爬行器从种子节点开始获得它们的入度和出度。用这种方法,可以获得图5.3中的集合(c)。当集合(c)中的节点被确定之后,同样可以获得它们的出度。这些出度信息可以被分为两类,第一是从集合(c)到集合(c)以及从集合(c)到种子节点。所有其他的出度都被看作从集合(c)到集合(d),而集合(d)中的节点最终汇聚为一个虚拟节点。

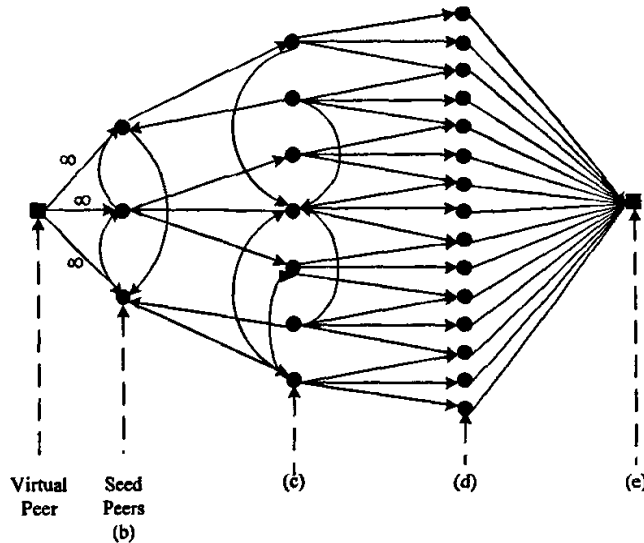


图5.3 女巫防御系统中爬行器的工作原理图

为了使用最小割算法将女巫攻击团体和P2P系统分割开,本文采用一个虚拟节点连接所有节点,如图5.4所示。假设为每个节点都建立到虚拟节点之间的无向边,边的容量为 $\alpha$ ,则图G变为扩展图 $G_\alpha$ 。扩展图 $G_\alpha$ 中的 $\alpha$ 对于寻找女巫攻击团体起着至关重要的作用。它的值与系统聚类的数目有着密切的关系。

当 $\alpha$ 为0时, $G$ 中所有节点与 $t$ 之间的最小割都为割 $(t, V)$ ,将 $t$ 和所有 $G$ 中的节点分离。这时,最小割算法将整个图 $G$ 聚为一类。

当 $\alpha$ 的值趋于无穷大时,最小割算法将 $G$ 中所有的 $n$ 个节点划分为 $n$ 个类(即 $n$ 个团体),每个团体只包含其自身。

当 $\alpha$ 的值在这两种极限情况之间变化时, $G$ 的 $n$ 就会被聚成1到 $n$ 个团体,聚类的形式主要取决于 $G$ 的结构和边的容量分布。并且,团体的数目随着 $\alpha$ 的增加是非递减的。

P2P系统中寻找女巫攻击团体的具体算法如图5.5所示。算法主要分为两步。首先是利用最大流算法产生信任团体。通过分析历史信息,获取网络的连接状态,并利用最短增广路算法产生信任团体 $G$ 。其次是选择种子,增强团体,并通过爬行器不断对种子集合进行扩充。

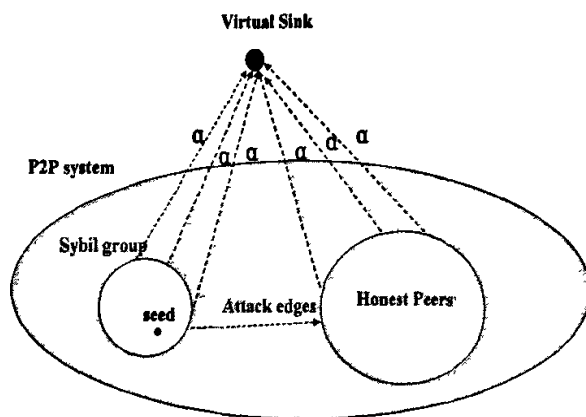


图5.4 引入虚拟节点的P2P系统

Algorithm *SybilCrawler* (P2P System:  $G=(N, E)$ ; peers:  $p, q \in N$ )

//  $N$ 为系统中节点的数目,  $E$ 为边的数目

While number of iterations is less than desired do

Set  $k$  equal to the number of peers in seed set,

Perform maximum flow analysis of  $G$ ,

Get the group  $G'$ .

Identify non-seed peers,  $p^* \in G'$ ;

With the highest in-degree relative to  $G$ ;

For all  $r \in G'$  such that in-degree of  $r$  equals  $p^*$

Add  $r$  to seed set;

Add edge  $(p, r)$  to  $E$  with infinite capacity.

End for

Identify non-seed peers,  $s^*$ ;

With the highest out-degree relative to  $G$ ;

For all  $s \in G'$  such that out-degree of  $s$  equals  $s^*$

Add  $s$  to seed set  $s$ ;

Add edge  $(p, s)$  to  $E$  with infinite capacity.

End for

Re-crawl so that  $G$  uses all seeds.

Let  $G$  reflect new information from the crawl.

End while

End

图5.5 寻找女巫攻击团体的爬行器算法

一旦女巫攻击团体被确定, 爬行器会把消息传播到系统中的其他节点。那么当一个节点选择另一个节点进行交易时, 它会首先判断对方是否属于女巫攻击团体, 再决定是否从对方下载文件或者将文件共享给对方。本文通过将女巫攻击团体和P2P系统分开, 可以保护整个系统的安全性。模拟试验证明了基于小世界模型的女巫防御机制的有效性。

## 5.5 模拟试验及性能分析

在这一部分，本文将通过仿真验证系统的有效性。假设节点到达系统遵循泊松分布。节点的生存周期呈指数分布，也就是说，在本文描述系统中，大部分节点只生存很短的时间。这也是和实际P2P应用的情况相符的，因为在现实P2P的应用中，大多数用户在下载完所需文件后都会离开，也就是常说的“搭便车”<sup>[64]</sup>。在这种情况下，只要很少一部分节点会长时间停留在系统中继续提供服务。

试验是基于P2P文件共享系统的，并且假设了三个场景：没有攻击者，一个攻击者和四个攻击者。我们将通过验证P2P系统在有无防御机制情况下，被女巫攻击者影响的节点数来证明本文提出的防御机制的有效性。假设攻击者节点用于无穷大的生命周期，因为它们一般不会离开系统，并且也不考虑恶意节点出现故障的情况。女巫攻击者的主要目标就是尽可能多的获取系统中的IDs，因此我们假设一旦获取一个新的IDs就意味着新节点的加入。试验使用C++编写源码，相关参数如表5.1所示。

表5.1 基于小世界模型的女巫攻击防御系统中参数的说明

参 数	定 义
$N$	系统中节点的数目
$N_{\text{sybil}}$	女巫攻击者的数目
$k$	种子的个数
$\alpha$	节点和虚拟节点之间的权值
$R_a$	合法节点的到达率
$L$	一个合法节点的平均生命周期

首先给出没有防御机制情况下，P2P系统中受女巫攻击的情况。在系统达到稳定状态之后，女巫节点开始进行攻击，如图6所示。在 $t=20$ 分钟时，网络中用于1000个节点，并达到稳定状态。女巫节点与 $t=40$ 分开始进行攻击。本文试验结果显示一个攻击者在1个小时以内可以控制系统中将近10%的节点数，而4个攻击者只需半个小时就可以达到同样的效果。图5给出了在使用基于小世界模型防御机制情况下P2P系统中被控制节点的百分数。可以看出，此时，一个攻击者在相当长的一段时间内只能够控制系统中不到2%的节点数，而4个攻击者对于节点的控制能力也不到5%。通过对比图5.6和图5.7，可以得出结论，本文提出的基于小世界模型的防御机制有效的控制了女巫攻击的程度，降低了女巫攻击者所控制的节点数目在整个系统中的百分比。

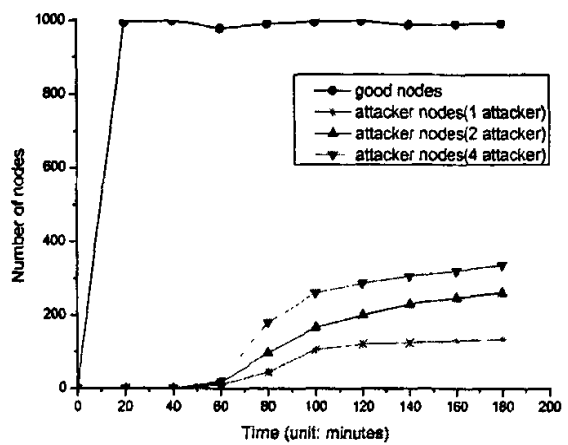


图5.6 无防御系统时女巫攻击的节点数目与时间的关系

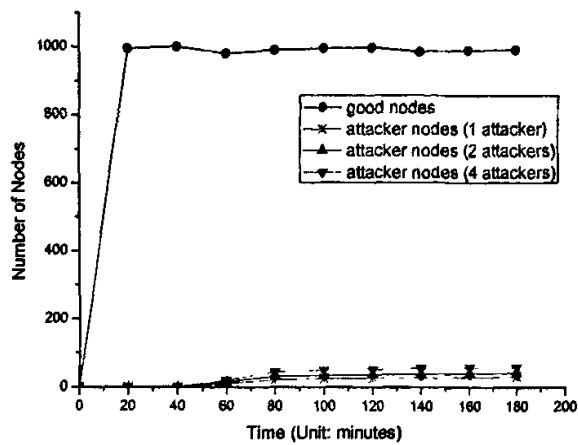


图5.7 有防御系统时女巫攻击的节点数目与时间的关系

5.6 本章小结

针对P2P系统中存在的女巫攻击问题，本文提出了基于小世界模型的女巫防御机制。防御P2P系统中的女巫攻击最关键的问题是找到女巫攻击团体，通过把女巫攻击团体的发现归结为最大流/最小割问题，本文引入了虚拟节点，并利用爬行器找到女巫攻击团体。这种方法有效地将女巫攻击者和P2P系统分开。模拟结果显示基于小世界模型的女巫防御机制有效地减少了整个系统中女巫攻击节点的百分比。

本文前面几章设计并实现了安全高效的P2P资源管理机制，但是这样的P2P系统仍然只支持基于关键字的精确查询。下面的工作将围绕如何构建基于内容查找的P2P系统展开。



## 第六章 基于快速映射的内容访问网络

文件管理层是P2P系统中基于内容资源管理机制的核心组成部分。针对当前结构化P2P文件存储系统存在仅支持单关键字的精确匹配,而缺乏支持内容查询的局限性,本章提出一种基于快速映射的结构化P2P存储系统FCAN。FCAN利用FastMap实现从语义空间到P2P空间的映射,在保证文本矢量语义完整性的同时对其进行有效降维,从而实现基于内容的查找;通过对质心法确定的最近邻居进行聚类,有效地减少了冗余信息和查询时间。

### 6.1 引言

尽管基于P2P的内容存储和检索相对于集中式存储和检索有着可扩展性强、实时性好、适合动态网络等优点,同时还可以充分挖掘分布在边缘网络的各类异构信息,然而正是由于P2P系统的灵活性导致这些存储系统中存在的关键问题是如何在分布式环境下快速准确地索引和定位资源对象。

已有很多关于P2P系统中资源定位问题的解决方案。Gnutella和Random Walk<sup>[65]</sup>分别通过洪泛算法和系统中随机选取的一条路径进行搜索。尽管这两种方法的维护成本很低,可是搜索代价却很高,因为它们分别会产生大量的冗余消息和冗余路径。Local Index<sup>[74]</sup>和Neighborhood Signature<sup>[111]</sup>通过维护邻居信息可以提高搜索效率。但是,当系统中内容发生变化时,对索引消息进行更新和维护都需要很大的代价。结构化P2P系统的核心思想是在资源的索引编码与P2P网络节点的标识符之间建立映射关系。这种方法的可扩展性非常好,但是仅限于精确匹配,不支持基于内容的查找。

本章旨在CAN的基础上设计一种支持快速映射的内容访问网络FCAN。实现结构化P2P系统中基于内容的查询,需要解决几个主要问题。首先,P2P叠加层的构建要保证P2P系统中节点的组织以及数据对象的放置和它们所属的语义空间一致。其次,实际应用中表示数据对象的特征信息往往很多,设计系统时要考虑在高维空间中有效完成查询的同时维护尽量少的信息。最后,系统是基于一P2P叠加层的,因此要尽量维护P2P系统的灵活性、可扩展性以及负载均衡等特点。

可以看出,实现结构化P2P系统中基于内容的查询,要解决的关键问题是实现P2P叠加层和语义空间的维数一致。因为语义空间的“维灾”(dimension curve)问题是影响P2P系统中对对象进行有效查询的主要因素。本章提出FCAN(FastMap-based CAN)系统,利用FastMap<sup>[112]</sup>对语义空间进行有效降维,实现文本矢量从多维语义空间到P2P空间的映射;根据对象的质心选择最近邻居完成聚类,并通过超级节点连接。FCAN不仅有效地实现了结构化P2P中基于内容的查询,并且模拟显示其具有较短的搜索时间和高召回率。

本章第一部分是相关工作,第二部分详细描述FCAN的系统结构以及基本搜索算法,第

三部分是对FCAN系统的改进，第四部分是仿真实验，最后是结论。

## 6.2 相关研究工作

### 6.2.1 DHT上基于语义检索的方法

基于广播的搜索方法，可以实现基于内容的查询。然而，这种方法在对稀疏资源进行定位时往往效率非常低。以CAN, Chord等为代表的结构化P2P采用了基于DHT的搜索方案，尽管可以解决对等网络中快速定位的问题，却只能支持基于单关键字的精确匹配，而未实现基于内容的查询。

实现结构化P2P系统中基于内容查询的关键技术是对语义空间的有效降维。pSearch<sup>[100]</sup>在语义向量空间上实现了基于rolling index的降维。Rolling index把多维空间的向量划分为 $p$ 个子向量，每个子向量包括 $m$ 维。在这里， $m=2.3*\ln(N)$ ， $N$ 是系统中节点的数目。pSearch在CAN的空间上分别执行 $p$ 个子查询完成基于内容的搜索。这种方法实现起来很简单，但是容易造成语义的丢失，同时会引起很大的负载。和pSearch不同的是，SSW(Semantic Small World)<sup>[101]</sup>是基于ASL(adaptive space linearization)对叠加层网络实行降维。这种方法克服了pSearch中存在的负载过大和语义丢失的弊端，但是仍然存在语义空间和P2P叠加网络空间之间的维数不匹配问题。基于质心法的内容索引算法<sup>[113]</sup>通过LSI(Latent Semantic Index)挖掘单词间的语义关系，并通过对各个单词向量进行加权平均聚类来获得文档的内容索引。但是基于质心法的内容索引通过对矩阵进行SVD(Singular Value Decomposition)分解的作法不适合动态环境，因为在动态的P2P环境中要使搜索性能不下降，必须不断重新进行SVD变换，而计算SVD的时间复杂度为 $O(n*d^2)$ 。

针对这些问题，本章提出了基于FastMap对语义空间实现降维的FCAN，这种方法实现简单，保证了语义空间和P2P空间的一致性，同时具有较低的时间复杂度和搜索时间。

### 6.2.2 内容访问网络CAN

内容访问网络CAN可以在Internet规模的大型对等网络上提供类似哈希表的功能。CAN基于虚拟的 $d$ 维笛卡儿坐标空间实现数据的组织和查找功能。整个坐标空间动态地分配给系统中的所有节点，每个节点拥有独立的区域。CAN的路由机制非常简单，如图6.1所示。如果一个二维空间划分成 $n$ 个相等的区域，那么平均路由长度是 $(d/4)(n^{1/d})$ ，每个节点只需要维护 $2d$ 的邻接节点信息，这表名CAN的可扩展性好，能够在 $O(\log n)$ 跳之内找到目的节点，并且节点数增加时每个节点维护的信息不变，路由长度只是以 $O(n^{1/d})$ 的数量级增长。

本章利用CAN这种 $d$ 维拓扑空间的逻辑网络特点，在其上搭建基于内容的检索系统FCAN。FCAN的基本思想是利用FastMap技术对语义空间实现降维，并映射到CAN系统中，使得所有文本的索引信息和CAN中节点的坐标映射到同一空间中。当发布某个文本信息时，

首先建立该文本的内容索引index，然后把索引信息存放在地址和index相近的CAN节点上。在执行某个查询操作时，由查询索引所在的节点利用洪泛算法完成查找。

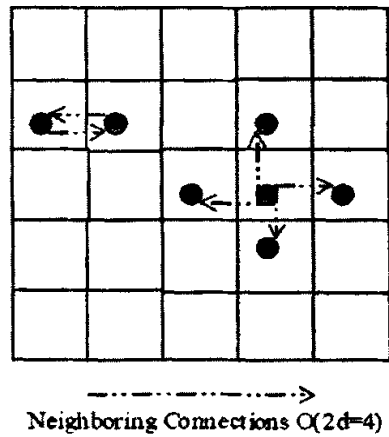


图6.1 二维空间上的CAN结构

### 6.3 FCAN基本结构设计

#### 6.3.1 FCAN系统中文本索引的表示

在基于内容检索的结构化P2P系统中，文本的索引必须满足两个条件：索引能够正确地体现文本的内容，并且索引之间的相似度可以体现文本之间的相似度；索引的表示应该和P2P系统中节点的地址具有一致性，从而保证查询可以在P2P系统中顺利进行。为了满足第一个条件，本文采用传统的向量空间模型VSM(Vector Space Model)<sup>[114]</sup>。VSM是近年来应用较多而且效果较好的文本特征表示方法。在该模型中，文本空间被看作是由一组正交词条向量所组成的矢量空间，每个文本表示为其中的一个范式特征向量：

$$V(d) = (t_1, \omega_1(d); \dots, t_i, \omega_i(d); \dots, t_n, \omega_n(d)) ,$$

其中 $t_i$ 为词条项。因此，要将文本表示为向量空间中的一个向量，就首先要将文本分词，由这些词作为向量的维数来表示文本。 $\omega_i(d)$ 为 $t_i$ 在 $d$ 中的权重，一般定义为 $t_i$ 在 $d$ 中出现频率的函数，即 $\omega_i(d) = \Psi(f_i(d))$ 。最初的向量表示完全是 0、1 形式，即，如果文本中出现了该词，那么文本向量的该维为 1，否则为 0。这种方法无法体现这个词在文本中的作用程度，所以逐渐 0、1 被更精确的词频代替，词频分为绝对词频和相对词频，绝对词频，即使用词在文本中出现的频率表示文本，相对词频为归一化的词频，其计算方法主要运用 TF-IDF公式一种比较普遍的TF-IDF公式：

$$W(t, \bar{d}) = \frac{tf(t, \bar{d}) \times \log(N/n_t + 0.01)}{\sqrt{\sum_{t \in \bar{d}} [tf(t, \bar{d}) \times \log(N/n_t + 0.01)]^2}} \quad (6.1)$$

其中,  $W(t, \bar{d})$  为词  $t$  在文本  $\bar{d}$  中的权重, 而  $tf(t, \bar{d})$  为词  $t$  在文本  $\bar{d}$  中的词频,  $N$  为训练文本的总数,  $n_t$  为训练文本集中出现  $t$  的文本数, 分母为归一化因子。为了简单起见, 本文此处使用0、1表示。

通过VSM表示的文本向量之间的夹角可以反映文本内容的相似性, 如图6.2所示。两个文档内容相关程度的大小可以用文档向量间的距离来衡量, 一般使用夹角的余弦来计算文本相似度。两者夹角越小说明相似度越高, 由于查询也可以表示为同一空间里的一个查询向量, 可以通过公式(6.2)给出的相似度计算得出每个文档向量与查询向量的相似度, 排序这个结果后与设立的阈值进行比较。如果大于阈值则页面与查询相关, 保留查询结果; 否则, 过滤掉该文本。

$$\text{Sim}(D_1, D_2) = \cos\theta = \frac{\sum_{k=1}^n w_{1k} \times w_{2k}}{\sqrt{\left(\sum_{k=1}^n w_{1k}^2\right) \left(\sum_{k=1}^n w_{2k}^2\right)}} \quad (6.2)$$

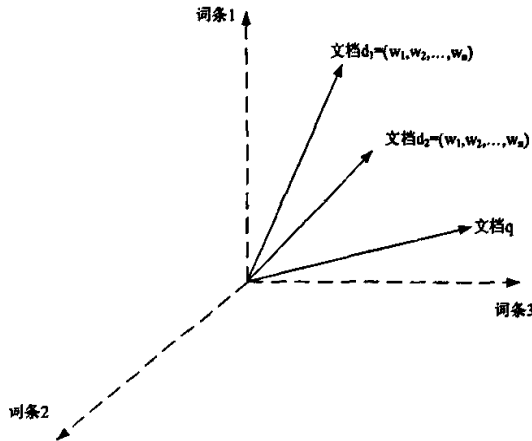


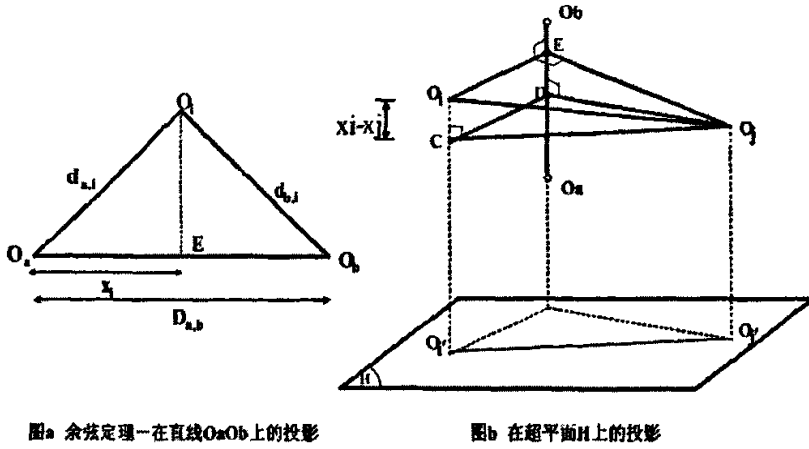
图6.2 VSM模型以及相似度表示

对于第二个条件, 本章采用的是FastMap映射算法对高维空间向量进行降维。

### 6.3.2 FastMap映射算法

由6.3.1节可以看出, 任一文本可用存在于 $n$ 维空间的一个向量表示。那么, 数目为 $A$ 的文集可以表示为 $n$ 维空间中的 $A$ 个点, 即  $O_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n}), i = 1, 2, \dots, A$ 。FastMap算法通过距离函数  $Diff(O_i, O_j)$  将这些点映射到互相正交的 $k(k < n)$ 维空间上, 实现降维。本文的

$Diff(O_i, O_j)$  采用的是  $O_i, O_j$  之间的 Euclidean 距离。



图a 余弦定理—在直线  $O_a O_b$  上的投影

图b 在超平面  $H$  上的投影

图6.3 FastMap的  $k$  维空间映射图

如图6.3(a)所示,  $O_a, O_b$  是  $n$  维空间中的两个距离最远的点, 对三角形  $O_a O_i O_b$  应用余弦定理, 则  $d_{b,i}^2 = d_{a,i}^2 + d_{a,b}^2 - 2x_i d_{a,b}$ , 从而可得  $O_i$  在直线  $O_a O_b$  上的投影  $x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{2d_{a,b}}$ ,

其中  $d_{i,j}$  代表的是  $Diff(O_i, O_j)$ ,  $i, j = 1, \dots, A$ 。

图6.3(a)描述的是二维空间中的映射, 同理可以应用到多维空间上。如图6.3(b),  $H$  是垂直于  $O_a O_b$  的  $(n-1)$  维空间超平面, 现将目标  $O_i, O_j$  映射到  $H$  超平面。在超平面  $H$  上, 映射点  $O'_i, O'_j$  之间的差异度  $Diff'(O'_i, O'_j)$  可以通过  $O_i, O_j$  之间的差异度计算得到,

$$Diff'(O'_i, O'_j)^2 = (Diff(O_i, O_j))^2 - (x_i - x_j)^2 \quad (6.3)$$

公式(6.3)解决了  $n$  维空间向  $n-1$  维空间的映射问题。类似可以进行  $k$  次迭代完成到任意  $k$  维空间的映射。关于  $O_a$  以及  $O_b$  的选取, 文献[72]给出了详细的迭代求解过程。

### 6.3.3 FCAN基本搜索算法

本节主要介绍FCAN系统中基本的文本索引发布算法以及查询算法。根据6.2.2节和6.3.1节, FCAN通过对关键字向量表示的文本建立索引, 并把该索引发布到CAN网络中, 利用索引信息和CAN在同一空间的特点, 实现结构化P2P系统中基于文本内容的查询。实现结构化P2P系统中基于文本内容的查找, 需要获得文本向量在P2P空间中的映射。本文采用FastMap映射算法。首先, 需要分别设置包含  $k$  个元素的二维数组  $LM[k]$  和一维数组  $M[k]$ , 用于存放每次映射选取的参考点  $O_a, O_b$  以及文本在不同坐标系的映射。具体算法如图6.4:

Algorithm *FastMaponCAN*( $n$ ,  $Diff()$ ,  $Doc_{SematicVec}$ )

输入: document的 $n$ 维向量表示 $Doc_{SematicVec}$ ;

CAN的维数 $k$  ( $k \ll n$ );

输出: document在 $k$ 维CAN空间上的映射点 $P_{docmaponCAN}$

(1) landmark=0; // landmark记录文本已经被映射的次数

(2) if ( $n=k$ ) {return;}

(3) else {landmark++;}

(4) 根据  $Diff()$  选取语义空间中差异度最大的对象  $O_a$ ,  $O_b$ , 并令  $LM[1, landmark]=$

$O_a$ ,  $LM[2, landmark]= O_b$ ;

(5) 对于文本document矢量, 利用图6.2(a)所示的余弦定理获取其在直线  $O_a O_b$  上投影 $x$ , 并令  $M[landmark]=x$ ;

(6) 调用 *FastMaponCAN*( $n-1$ ,  $Diff'()$ ,  $Doc_{SematicVec}$ ); //  $Diff'()$  通过公式(6.3)获得;

(7) 最终获得document在 $k$ 维超平面上的投影  $P_{docmaponCAN}=(M[1], M[2], \dots, M[k])$ .

图6.4 FCAN的映射算法

图6.4所示的*FastMaponCAN*算法给出了一个文本向量从 $n$ 维语义空间到 $k$ 维FCAN空间的映射。对于给定的文本, 一旦获得它在CAN空间上的投影 $P_{docmaponCAN}$ , 就可以根据 $P_{docmaponCAN}$ 确定和其相近的系统中的节点, 并为该文本建立索引, 发布给该节点。发布索引信息之前, 首先为文本建立四元组属性(DocID, PeerID, IPdoc,  $Doc_{SematicVec}$ ), 其中DocID为文本唯一的标识符, IPdoc为发布该document的节点IP地址。然后, 通过 $P_{docmaponCAN}$ 确定应该保存该索引信息的节点PeerID, PeerID收到四元组索引信息后, 组织成本地节点的倒排文档, 以供本地文件的检索。

如图6.5所示, 对于一个给定的查询 $q$ , 首先利用*FastMaponCAN*映射算法获得 $q$ 在FCAN上的映射 $q'$ 。然后, 查询被转发给该映射区域所属的节点peer1, peer1在搜索完本地倒排文档之后, 将进行洪泛查找, 直到返回的结果数符合事先实现给定的数目。其中,  $P_{docmaponCAN}$ 作为 $Doc_{SematicVec}$ 在FCAN上的映射, 被包括在peer1的索引范围之内, 因此将被作为结果返回。

FCAN系统通过FastMap实现语义空间到P2P空间的映射, 在保证语义完整性的同时有效地对语义空间进行了降维。算法*FastMaponCAN*在第4、6步的递归都在线性时间内完成的。因此, 整个算法的时间复杂度为 $O(n)$ 。

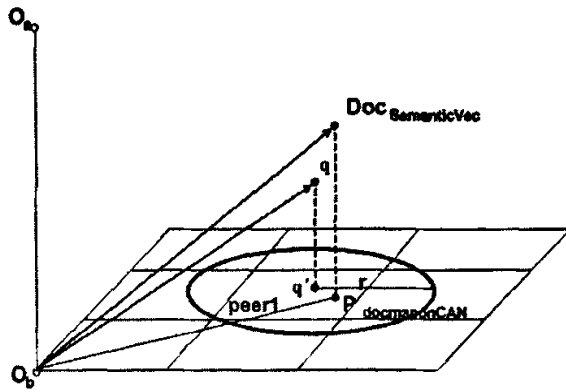


图6.5 FCAN的基本搜索算法

## 6.4 基于最近邻居优先聚类的FCAN

尽管已经通过FastMap对于语义空间实现了有效的降维，并且维护了文本的语义信息。但是，对于CAN本身来说，仍然是一个多维空间。在这样一个多维的结构上，每个节点同时存在多个邻居。因此，当使用6.3节算法在有限范围内基于洪泛进行查找时，会不可避免地带来大量的冗余信息。为了减少冗余消息以及系统中的通讯量，本文在FCAN的基础上实现基于质心法的最近邻居优先聚类，节点在完成本地检索后，将查询转发给聚类的超级节点，继续执行。这样，通过避免采用洪泛算法而有效地减少了系统中的通讯量。基于最近邻居优先的聚类算法主要有两步，首先是获取CAN中每个节点的最近邻居，然后通过给定阈值对节点进行分类。

#### 6.4.1 基于质心法最近邻居的选择

FCAN是实现了语义空间到结构化P2P映射的查询系统，因此，每个节点除了拥有其位于P2P系统中的标识符之外，还需要拥有语义标识符，从而可以描述该节点在语义空间中的位置。本章采用质心法表示节点在语义空间中的位置。采用质心法描述节点具有明显的优势。通过6.3.3节描述的文本映射算法可以发现，语义相近的文本被尽可能的发布在同一节点上，每个节点可以看作是具有相似语义的文本对象的集合。因此，质心法可以更加清楚的表示存储在节点上的文本信息。

FCAN是建立在多维空间上的查询系统,因此,FCAN中的每个节点都存在多个邻居。在这样一个系统中搜索节点的最近邻居,如果通过两两比较和每个邻居之间的距离,将会带来非常大的计算量。本章提出SearchSmallGroup算法,只需比较节点和它某一范围内邻居节点之间的距离,有效的减少了查询时间。首先需要证明如下定理:

**定理6.1** 设 $P=\{p_1, p_2, \dots, p_m\}$ 是 $n$ 维空间中的点集,  $p_i$ 是集合 $P$ 中的任一点, 距离阈值为 $d$ , 则到 $p_i$ 的距离小于 $d$ 的点一定在每一维上都在 $p_i$ 相应维的 $d$ 邻域内。

证明: 设点集  $P$  中存在一点  $p$ ,  $p$  到  $p_i$  的距离小于  $d$ ,  $p$  到  $p_i$  的距离,

$$D(p, p_i) = \sqrt{(p_1 - p_{i1})^2 + (p_2 - p_{i2})^2 + \dots + (p_k - p_{ik})^2 + \dots + (p_n - p_{in})^2} \leq d \quad (6.4)$$

假设,  $P$  在第  $k$  维上的投影不在  $p_i$  的  $d$  邻域内, 则  $p_k - p_{ik} > d$ , 那么  $(p_k - p_{ik})^2 > d^2$ , 所以  $D(p, p_i) > d$ , 与题设矛盾。所以, 原定理得证。

依据上述定理, 本文提出 FCAN 系统中选择最近邻居的 *SearchSmallGroup* 算法如下。

#### Algorithm *SearchSmallGroup*

输入 节点  $p$  在 FCAN 的质心表示  $p = (V_1, V_2, \dots, V_n)$ , 距离阈值为  $Dis_{sem}$ ;

输出 节点  $p$  的六元组属性 (PeerID, PeerCentroid, ClusterID, ShortestDistance, Neighbors, AsNeighbors)

- (1) 找出节点  $p$  在不同维数空间上的邻居点集合  $P$ ;
- (2) 找出集合  $P$  中在某一维上不属于  $p$  的  $Dis_{sem}$  邻域内的点, 删除;
- (3) 计算  $P$  中节点和  $p$  的  $Diff()$ , 并取最小的 ShortestDistance;
- (4) if  $p$  和某点  $p'$  之间的 ShortestDistance  $\leq Dis_{sem}$ , 则把  $p'$  写入  $p$  的 Neighbors, 同时将  $p$  写入  $p'$  的 AsNeighbors, ClusterID 为空;
- (5) else ClusterID, ShortestDistance, Neighbors, AsNeighbors 均为空,  $p$  为孤立点。

图6.6 最近邻居选择算法

观察图6.6所示的 *SearchSmallGroup* 可以发现, 该算法不必求  $p$  到所有邻居节点的  $Diff()$ , 而是比较一个很小邻域内的点, 从而有效的减少了运算量。如图6.7, 节点  $p_1$  如果按照图6.3所示的洪泛算法选择最近邻居节点, 要计算8次  $Diff()$ , 而本文只需要计算和  $d$  邻域内节点  $p_2, p_3, p_4, p_5$  之间的  $Diff()$  就可以了, 计算量明显减少。在这里, 算法的第(2)步中对于排序维的选取是一个非常重要的问题。本章采用随机取样, 并选取方差最大的维。这里不做过多的讨论。

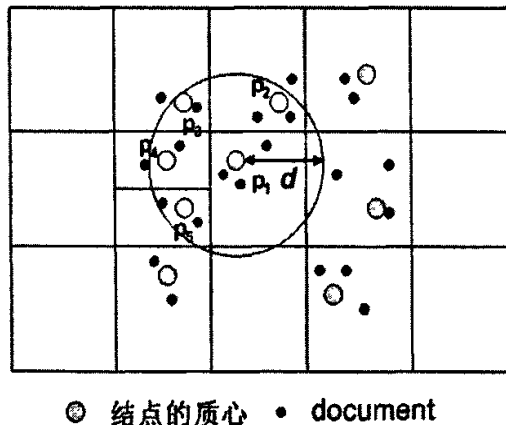


图6.7 基于质心法的  $d$  邻域内最近邻居



## 6.4.2 基于最近邻居聚类的实现

本章聚类算法的基本思想是将由质心法确定的最近邻居归为同一个类,不同聚类之间通过超级节点连接,类的ClusterID依据质心法确定。超级节点的选择可以结合节点的处理能力以及和质心的距离大小来确定。对于一个新加入的节点 $p$ ,在应用SearchSmallGroup算法确定了其最近邻居集Neighbors和以 $p$ 作为AsNeighbors的节点集合之后,可以通过其邻居节点的聚类属性确定自己的ClusterID。为了防止聚类不规则带来的负载不平衡问题,本章为聚类设置阈值ClusterNumber。当一个聚类的数目超过ClusterNumber时,超级节点会依据节点质心和聚类质心之间的距离选择出相对差异度较大的节点并删除。被删除的节点会自动根据邻居节点重新进行分类,或者成为孤立点。

如图6.8所示,当节点A执行某个查询 $q$ 时,首先在本地进行查找,如果返回对象的数目不满足需求或者未找到符合条件的对象,则 $q$ 被转发给超级节点。超级节点首先在自己的cluster内进行查找,返回结果。如果仍然不满足要求,则超级节点继续转发查询 $q$ 。最后A对返回结果进行排序,找到最符合条件的对象,本例中位于节点H上,查询结束。

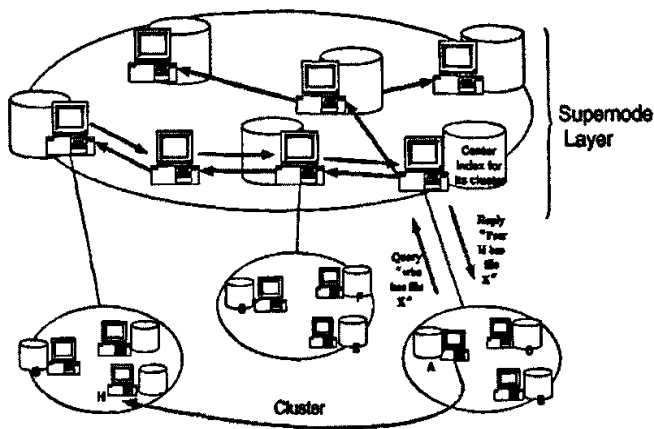


图6.8 超级节点连接的FCAN聚类结构图

## 6.5 模拟试验及性能分析

本节通过模拟实验来验证基于快速映射的内容访问网络FCAN有效性。实验的目标是为给定查询找出最符合条件的文本集合,并且分别对FCAN、pSearch、SSW以及质心法在查询效率以及维护成本方面进行比较。以下是实验环境以及相关的参数设置。

### 6.5.1 试验环境

本章使用BRITE<sup>[106]</sup>作为拓扑产生器生成二层拓扑结构。BRITE有四种类型的拓扑参数,这里选择按照bottom-up类型生成。按照bottom-up类型参数生成的拓扑有两层,分别是

Router-level和节点，并且每一层都有两种生成模式。实验中Router-level拓扑使用Waxman模式，router节点按照heavy-tailed的方法被分配到自治系统中，在此基础上用Java实现CAN系统。同时，本实验采用Otter0.9可视化工具显示生成的拓扑。CAN系统的初始化状态为只存在一个节点，并依次有新节点加入，直到节点数目达到N，系统呈现相对稳定的状态。本文的实验只考虑稳定状态下的系统性能，并假设此时的节点加入比率和离开比率相同。

采用的样本数据是从目前信息检索领域中最大的文集TREC中抽取528543个份文档，并由SMART为它们生成文档向量。查询样本采用TREC中第351—450的主题查询。

表6.1是实验中使用的参数，若无特殊说明，本章采用的都是表中所设的默认值。

本章的目标是在尽量少的查询时间内提高搜索效率，即获得尽可能高的查全率和精度。因此，我们在分析实验结果时采用下列指标：

- (1) recall ratio: 召回率，是检索出的相关文本数和文本库中的所有的相关文本数的比率，衡量的是查全率；
- (2) precision: 精度，是检索出的相关文本数与检索出的文本总数的比率，衡量的是查准率；
- (3) path length: 从源节点到目的节点之间位于叠加层上的跳数；
- (4) visited nodes: 完成每个查询需要访问的节点数目。

表6.1 FCAN系统参数

参 数	定 义	默 认 值
m	CAN的维数	100
N	系统中节点的个数	256—32K
ClusterNumber	聚类的节点数目	16
n	每个节点上存储的文本数目	100
N <sub>n</sub>	系统中样本文本的总数	528543
q <sub>n</sub>	查询的数目	100
R <sub>n</sub>	返回的文本数目	150
r	利用洪泛算法进行查找时的广播半径	3

6.5.2 FCAN查询性能分析

实验通过和pSearch、SSW以及质心法做比较，验证FCAN聚类前后的查询效率。本章主要从两个方面考虑查询效率，即查询质量和查询时间。其中查询质量由两个指标来衡量，即召回率和精度。对于一个系统来说，召回率和精度往往不能兼顾。如图6.9(a)和图6.9(b)所示，FCAN的召回率以及精度明显比pSearch、SSW以及质心法高。这说明基于FastMap的语义空间降维技术可以有效地实现文本向量到结构化P2P系统的映射，并且基于FastMap的降维所引起的语义信息的丢失是最少的。同时，观察图6.9(a)可以发现，随着节点数目的增

加,实际上是随着由节点加入所引起的文本数目的增加,实验中五个系统的召回率都呈现下降的趋势。这一点可以通过召回率的计算公式验证,由于系统中相关文本数是位于分母位置的,因此其数量的增加会带来召回率的下降。和图6.9(a)相反,图6.9(b)中显示的精度则是呈现上升趋势。并且,对FCAN进行改进之后,其查询结果的召回率降低了,然而精度得到相应提高。

验证查询效率的另一个方面是查询时间。本文假设节点之间的延迟是相等的,则查询时间可以通过P2P叠加层的跳数进行衡量。由于pSearch、质心法以及改进前的FCAN都是基于语义空间范围的洪泛查找,因此,其逻辑跳数和洪泛查找的半径呈线性关系,而与系统的查找算法无关。因此,这里只对SSW和聚类的FCAN进行验证。如图6.10所示,实现聚类的FCAN系统中查询时间比SSW短。这说明,通过对基于质心法选择出的最近邻居进行聚类,可以把具有相近语义的节点组织在一起,从而有效的控制查询范围,减少查询时间。

通过以上两个实验可以得出结论,基于聚类的FCAN即可以在保证语义完整性的情况下进行文本从语义空间到CAN的映射,从而实现了降维技术;又可以利用基于质心法的最近邻居聚类有效减少冗余消息,并且降低查询时间。

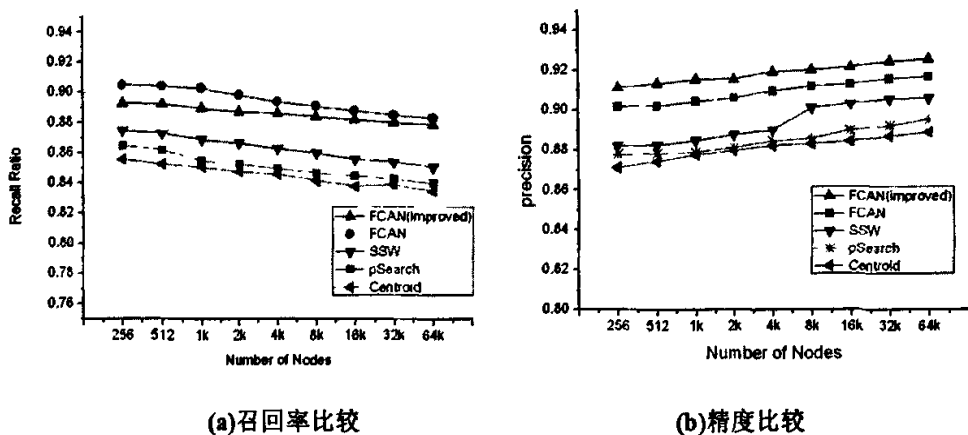


图6.9 FCAN性能分析

### 6.5.3 FCAN查询成本

本节通过比较查询需要访问的节点数衡量系统的搜索代价。图6.11显示,SSW和基于聚类的FCAN平均访问的节点数目明显比FCAN、pSearch以及质心法少。这是因为后三者采用区域内洪泛算法会带来大量的冗余消息,而SSW以及改进的FCAN采用聚类的方法可以减少访问的节点数,同时降低冗余消息。观察图6.11,还可以发现随着网络规模的扩大,pSearch、FCAN以及基于质心法的内容索引系统中访问节点数目增加的速度远大于SSW和改进后的FCAN。

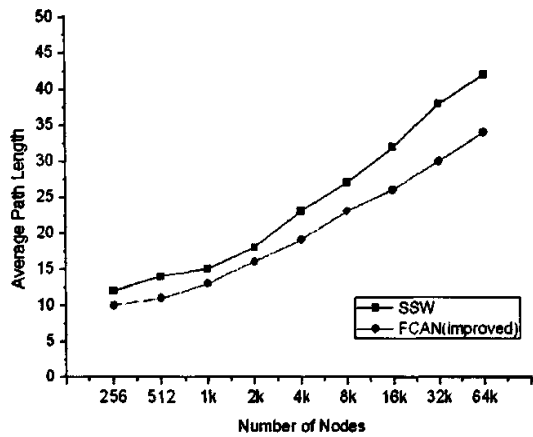


图6.10 SSW与FCAN平均路径长度比较

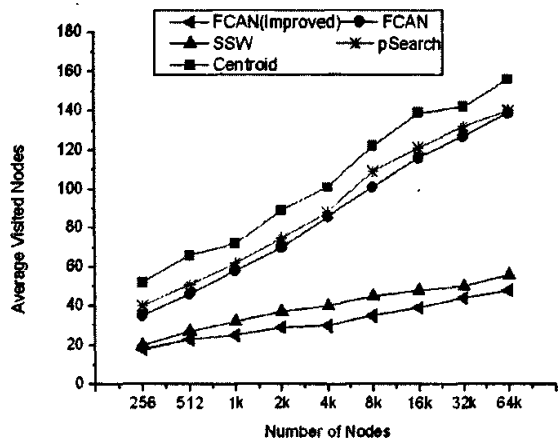


图6.11 平均访问节点数比较

6.6 本章小结

本章提出的FCAN系统利用快速映射实现了文本矢量从语义空间到P2P空间的映射，既有效地对语义空间实现了降维，又保证了语义信息的完整性；通过提出基于质心法的最近邻居优先聚类对FCAN进行改进，减少了原有系统中由于洪泛算法产生的冗余信息，提高了搜索效率和查询结果的精度。实验结果表明，在改进的FCAN系统中，查询结果具有比较高的精度和召回率；同时，基于最近邻居的聚类有效地减少了FCAN系统中的查询时间和代价。但是本文在设计系统时假设系统为一个相对稳定的状态，没有考虑到P2P系统的动态性，并且相关搜索信息的增加势必带来节点需要维护的成本，这些问题可以在将来的工作中考虑改进的办法。除此之外，本节使用的VSM文本表示方法具有计算量大的问题，因此本文在第七章提出基于集合差异度表示文本相似度的方法，处理文本向量中存在的高维稀疏矩阵问题，用于减少计算量。

## 第七章 利用集合差异度实现基于内容聚类的P2P搜索模型

文本相似度的计算是P2P系统中基于内容资源管理机制的重要组成部分。对于高维语义空间引起的文本相似度计算复杂,以及完全分布式非结构化P2P系统中广播算法带来的大量冗余消息的问题,本文提出利用集合差异度实现基于内容聚类的P2P搜索模型。该模型利用集合差异度定义文本相似度,将文本相似性的计算复杂度控制在线性时间内而有效地减少了查询时间;利用节点之间的集合差异度实现基于内容的聚类,可以降低查询时间,减少冗余消息。

### 7.1 引言

基于对等网络的内容存储和检索相对于集中式存储和检索有着可扩展性强、实时性好、适合动态网络等优点。同时还可以充分挖掘分布在边缘网络的各类异构信息。然而,P2P系统的核心问题是如何在分布式环境下快速准确地索引和定位资源对象。

结构化P2P网络需要建立在DHT(distributed Hash table)上,在给定资源索引的情况下,能够在 $O(\log N)$ 跳之内定位到目的节点。DHT方法的核心思想是在资源的索引编码与P2P网络节点的标识码之间建立映射关系。这些系统的可扩展性非常好,但是仅限于精确匹配,不支持基于内容的搜索。尽管非结构化P2P系统的典型代表Gnutella支持基于内容的模糊查询,但是,它所采用的基于语义空间向量的相似度计算方法具有较高的时间复杂度,并且洪泛算法会消耗大量网络带宽,带来大量冗余消息。

目前减少Gnutella系统中冗余消息的有效作法之一是实现节点的聚类。因此,本章采用基于内容的聚类模型用于减少计算文本相似性的时间复杂度和冗余消息。实现P2P系统中基于内容的聚类,需要解决两个主要问题。首先,文本的表示方法可以体现文本之间的相关度。其次,基于文本相关性的聚类方法可以有效地组织节点。本章提出一种非结构P2P环境下利用集合差异度实现基于内容搜索的模型,能够较好的体现文本之间的相似性,并依据这种相似性实现节点之间基于内容的聚类,不同的聚类之间通过超级节点连接。该模型从集合的角度定义了一种新的文本相似性计算方法,不仅对数据进行了有效压缩,使得数据处理量大大减少;而且利用文本集合的差异度衡量节点之间内容的相似性,可以实现基于内容的聚类,减少冗余消息。实验证明,本章提出的搜索模型不仅具有较高的召回率,而且将搜索成本和查询时间分别降低到了原有Gnutella系统的40%和30%左右。

本章第二部分是相关工作;第三部分提出一种基于集合差异度的文本相似性计算方法;第四部分介绍本章提出的基于内容的搜索模型及其维护;第五部分是模拟实验;最后是结论。

## 7.2 相关研究工作

对于非结构化P2P系统基于内容查询的研究,已经有一些相关工作<sup>[115]</sup>。文献[116][117][118][119]都提出了对于包含相似语义文本的节点聚类算法。但是,[116]依靠中心服务器对文本和节点进行聚类,会产生瓶颈问题;[117]提出对于有相同兴趣的节点进行聚类,但是对于节点之间兴趣相似性的定义以及如何形成聚类没有提及。[118]依靠在节点之间周期性地交换消息来保证节点之间的语义相关性,但是这样会带来大量的冗余消息。并且,以上工作普遍采用传统的向量空间模型VSM(Vector Space Model)<sup>[114]</sup>描述文本向量,并通过向量之间的夹角余弦衡量文本的相似度。[119]对本地文献按照ACM Topic进行分类并统计从而描述Peer的语义信息。这些做法使得文本有限的属性在庞大的语义矢量空间中构成稀疏矩阵,增加了计算量,降低了搜索效率。

本章提出基于集合差异度的对等网络搜索模型,利用集合的稀疏特征向量表示文本,并通过特征向量之间的差异度衡量文本的相似性,有效地降低了文本相似性的计算复杂度。同时,基于超级节点的聚类方法可以减少洪泛查找带来的大量冗余消息。

## 7.3 基于集合差异度的文本相似性表示

有效完成任何搜索的前提是必须对要搜索的文本提供一种完整而准确的表示方式。而文本的内容是人类使用的自然语言,计算机很难处理语义。这就需要对文本进行预处理,抽取代表其特征的元数据,这些特征可以用结构化的形式保存,作为文本的表示形式。

传统的向量空间模型VSM是近年来应用较多的文本特征表示方法。在该模型中,文本空间被看作是由一组正交词条向量所组成的矢量空间,每个文本表示为其中的一个特征向量,通过文本特征向量之间的夹角余弦反映文本内容的相似性。但是当整个文本空间中抽取的属性或者关键词数目很大,而用于描述每个文本的属性却很少时,对于每个文本可能存在很大一部分属性值为0。此时,文本空间则相应成为高维稀疏矩阵,导致信息量的冗余以及余弦计算量的增大。本章基于集合差异度定义一种基于集合稀疏差异度的文本相似性计算方法,可以有效压缩数据,减少数据的运算量。这里首先给出集合差异度定义。

**定义7.1.**在一个包含 $n$ 个文本, $m$ 个关键字的文集中,每个文本由 $m$ 个关键字组成的特征向量表示,关键字取值为1或0(称为稀疏特征值)。X为其中的一个文本子集,其中的文本个数为 $|X|$ ,该子集中所有对象稀疏特征值均为1的关键字个数为 $a$ ,稀疏特征值不全相同的关键字个数为 $b$ ,集合X的稀疏差异度 $SD(X)$ 定义为,

$$SD(X) = \frac{b}{|X| * a} \quad (7.1)$$

一个文本集合的稀疏差异度可以表明该集合内部各文本间的差异程度。利用这种集合的差异度,定义7.2提出一种新的文本相似性度量方法:

**定义7.2.**假设存在集合D,从中任取两个文本 $d_1$ 和 $d_2$ 的组成子集D'。两文本中关键字均

为1的个数为 $Num1$ ，关键字不全相同的个数为 $Num2$ ，则利用 $D'$ 的集合差异度计算 $d_1$ 和 $d_2$ 的相似度 $Sim(d_1, d_2)$ 为，

$$Sim(d_1, d_2) = \frac{1}{SD(D')} = \frac{2 * Num1}{Num2} \quad (7.2)$$

比如,用于描述文本集合 $D$ 的关键字为 $(A_1, A_2, \dots, A_6)$ 。假设 $d_1$ 表示为 $(1, 0, 1, 1, 1, 1)$ ， $d_2$ 表示为 $(1, 0, 0, 0, 0, 1)$ ， $d_3$ 表示为 $(1, 0, 1, 1, 0, 0)$ ，那么 $Sim(d_1, d_2)$ 的值为1.3333， $Sim(d_1, d_3)$ 的值为3。这说明 $d_1$ 和 $d_3$ 的内容更接近，同时也证明了定义的正确性。

传统的VSM利用文本向量的夹角余弦比较文本之间的相似性。但是对于高维语义向量空间中的文本向量，其余弦的计算具有较高的计算复杂度。而本文给出的基于集合差异度的文本相似性计算方法可以将相似度的计算时间控制在线性范围内。观察定义7.1可以发现，利用集合的稀疏差异度不仅可以表示文本的相似性，还可以实现基于内容的聚类，并且聚类的效果可以依据差异度获得。下面给出聚类特征向量的定义。

**定义7.3.** 假设某文本集 $corpus$ 有 $n$ 个文本，描述每个文本的特征值有 $m$ 个，取值为1或0。P2P系统中某聚类 $A$ 的文本集合为 $corpus$ 中的一个子集，其中文本个数为 $Num_A$ 。那么， $A$ 中所有文本的稀疏特征取值都为1的关键字个数为 $a$ ，对应的关键字序号为 $j_{k_1}, j_{k_2}, \dots, j_{k_a}$ ，稀疏特征值不全相同的关键字个数为 $e$ ，对应的关键字序号为 $j_{m_1}, j_{m_2}, \dots, j_{m_e}$ 。向量 $SV(A) = (|A|, S(A), NS(A), SD(A))$ 为聚类 $A$ 的特征向量。其中 $|A|$ 为 $A$ 中文本对象的个数， $S(A) = j_{k_1}, j_{k_2}, \dots, j_{k_a}$ ， $NS(A) = j_{m_1}, j_{m_2}, \dots, j_{m_e}$ ， $SD(A)$ 为聚类 $A$ 的稀疏差异度，计算方法见公式(7.1)。同时，可以发现， $a = |S|$ ， $e = |NS|$ 。

聚类特征向量概括了一个文本聚类的稀疏特征以及该聚类内各文本之间的差异度。这样，对于一个文本聚类，只需存储其特征向量就可以描述其特征，有效地减少了数据量。同时，两个聚类合并时，它们的特征向量具有可加性。

**定理7.1(聚类特征向量可加性定理).** 假设有 $n$ 个文本,描述文本的关键字个数有 $m$ 个,取值为0或1,  $A$ 和 $B$ 为其中不相交的两个文本子集,  $A$ 和 $B$ 合并后的文本集为 $A \cup B$ , 则,  $SV(A \cup B) = SV(A) + SV(B)$ 。

利用定理7.1，可以对节点实现基于文本内容的聚类，将搜索控制在某个特定范围内，既可以降低查询时间，又能够减少冗余消息。

## 7.4 基于内容聚类的搜索模型

基于洪泛算法的非结构化P2P系统因盲目搜索导致效率低，并产生大量冗余消息。本文在定义7.2的基础上提出基于内容聚类的搜索模型。该模型根据节点的特征向量差异度对其

实现聚类，并通过超级节点连接。基于超级节点的查询有效地避免了盲目搜索，既节省了时间又减少了冗余信息。这里首先介绍基于内容搜索模型的聚类过程，然后是查询算法的实现，最后是系统的维护。

7.4.1 基于内容的聚类模型

本文先把系统中的每个节点看作一个聚类，则每个节点具有一个依据定义3所描述的聚类特征向量。根据节点的特征向量差异度对其实现聚类，并由超级节点连接不同聚类，如图7.1所示。其中，下层为待聚类的 $n$ 个节点，上层为代表 $k$ 个不同聚类的超级节点。基于内容的聚类过程可以用一个自底向上的两层结构来描述，图7.2显示了图7.1所对应的基于聚类特征向量的结构图。

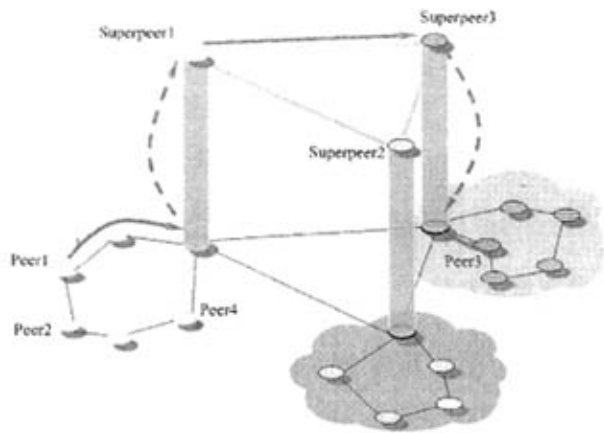


图7.1 超级节点连接的基于内容的搜索模型

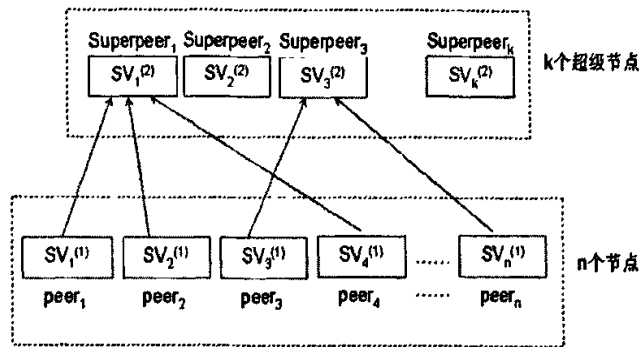


图7.2 基于聚类特征向量的结构图

从图7.2可以看出，每个节点拥有一个特征向量SV，便于实现节点之间基于内容的聚类。假设在本文的系统中，有 $n$ 个节点，并且文本集中提取 $m$ 个关键字，描述第 $i$ 个节点的 $m$ 个稀疏特征取值分别对应于二态变量值 $p_{i1}, p_{i2}, \dots, p_{im}$ ，同时一个聚类的集合差异度上限为 $Threshold_{cluster}$ 。那么，基于内容的聚类算法处理步骤如下：



STEP1:为每一个节点建立一个文本集, 分别  $X_i^{(0)}$ ,  $i \in \{1, 2, \dots, n\}$ ;

STEP2:根据可加性定理, 计算  $SV(X_1^{(0)} \cup X_2^{(0)}) = SV(X_1^{(0)}) + SV(X_2^{(0)})$ , 如果合并后形成聚类的稀疏差异度不大于  $Threshold_{cluster}$ , 则对节点1和节点2实行聚类, 记为  $X_1^{(0)}$ ; 否则, 将  $X_1^{(0)}$  和  $X_2^{(0)}$  分别作为一个初始类, 记为  $X_1^{(0)}$  和  $X_2^{(0)}$ 。并将聚类的个数设为  $Number_{cluster}$ ;

STEP3:对于节点3, 计算  $SV(X_3^{(0)} \cup X_k^{(0)}) = SV(X_3^{(0)}) + SV(X_k^{(0)})$ ,  $k \in \{1, 2, \dots, c\}$  寻找某个节点  $i'$ , 使得  $SD(X_3^{(0)} \cup X_{i'}^{(0)}) = \min_{k \in \{1, 2, \dots, c\}} SD(X_3^{(0)} \cup X_k^{(0)})$ 。如果  $SD(X_3^{(0)} \cup X_{i'}^{(0)})$  不大于  $Threshold_{cluster}$ , 则合并  $X_3^{(0)}$  和  $X_{i'}^{(0)}$ , 并记为  $X_{i'}^{(0)}$ ; 否则, 将  $X_3^{(0)}$  作为一个新的初始类, 记为  $X_{c+1}^{(0)}$ , 同时  $Number_{cluster}$  的个数增加1;

STEP4:对于  $X_i^{(0)}$ ,  $i \in \{4, 5, \dots, n\}$ , 依次执行STEP3, 最终形成聚类  $X_k^{(0)}$ ,  $k \in \{1, 2, \dots, c\}$ 。

通过不断的迭代, 最终将节点划分成为  $Number_{cluster}$  个聚类, 并且聚类之间可以明显的区分, 而聚类内的节点具有较高的内容相似性, 并且聚类的算法复杂度为线性时间。

## 7.4.2 查询的实现

系统中每个节点都维护一个本地文本的倒排表, 记录关键字在文本中出现的情况。同时, 每个超级节点还需要维护一个词汇表, 用于记录在该聚类内抽取的所有文本的关键字及其在节点上的分布, 如图7.3所示。当某个节点  $p$  发起向量为  $q=(t_1, t_2, \dots, t_m)$  的查询时, 具体步骤为:

STEP1: $p$  首先在本地倒排表中进行查找, 如果能够获得一些相关文本的 DocID, 则计算查询向量  $q$  和文本向量之间的相似度(定义2给出), 并返回相似度大于  $Similarity_{document}$  的文本, 如果文本数目达到事先给出的数目, 则结束查询; 否则,  $q$  被转发给超级节点 SN;

STEP2:SN 收到查询  $q$ , 首先进行本地查询; 然后, 依据聚类稀疏特征向量之间的相似度在词汇表中查找具有相关文本的  $K$  个节点  $p_1, p_2, \dots, p_k$ , 并将  $q$  转发给这些节点;

STEP3:节点  $p_1, p_2, \dots, p_k$  分别将结果返回给 SN, SN 按照相似度对返回的 DocID 进行排序并进行选择;

STEP4:如果返回结果没有达到给定数目, SN 使用洪泛算法将查询转发给其他超级节点, 并重复STEP2和STEP3, 直至结束。

7.4.1节介绍的基于内容的聚类方法已经将文本相似度较高的节点汇聚在一起。因此,

这里的查询算法在进行查找时，高效地选择了最有可能包含查询结果的节点，避免了广播消息带来的冗余信息，提高了查询效率。

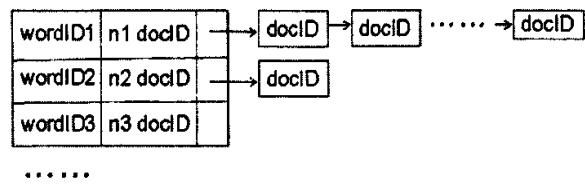


图7.3 超级节点的倒排表

7.4.3 查询系统的维护

在具有高动态性的P2P环境中，由于节点的自由加入或离开所导致文本的频繁增加或删除，必然会引起超级节点上词汇表的不断变化。当一个文本的加入带来新的词汇时，节点会把消息发送给超级节点，超级节点自动更新词汇表，并把词汇增加信息广播给其他超级节点，从而保证整个系统词汇信息的一致性。

7.5 模拟试验

本节通过实验从两个方面验证利用集合差异度实现基于内容聚类的搜索模型的有效性，即算法的查询效果和查询效率。查询效果主要从用户的角度考虑，评价指标是查全率；查询效率主要从系统的角度考虑，评价指标是查询时间和访问的节点数。以下是实验的环境设置以及实验结果。

7.5.1 试验环境

研究显示，Gnutella的结构分布规律和幂律图Power-law<sup>[39]</sup>相似，因此，本文实验使用BRITE<sup>[66]</sup>作为拓扑产生器生成基于Power-law规律的网络拓扑结构。同时，采用Otter0.9可视化工具显示生成的拓扑。模拟程序用JAVA编写。

采用的样本数据是从目前信息检索领域中最大的文集TREC中抽取2000个和基因科学相关的文档摘要(TREC 2005)，以及第100-300的主题查询。文本在系统中呈Zip-f<sup>[14]</sup>规律分布，即只有少部分节点共享文件，大部分节点搜索或下载文件。表7.1是相关参数。

7.5.2 查询效果

衡量基于集合差异度的内容搜索模型的查询效率，本文采用信息检索系统中的查全率(Recall)，Recall的计算公式为：

Recall=Relevant matched Documents/Relevant Documents. (7.3)

其中Relevant matched Documents代表查询结果中与查询相关的文本数目，Relevant

Documents为文本集中与查询相关的文本数目。

表7.1 基于内容聚类搜索模型的试验参数

参 数	默认值	定 义
$Number_{peer}$	1000	网络中的节点数目
$Number_{words}$	1000	超级节点维护词汇表包含的关键字个数
$Key_{peer}$	150	每个节点的倒排表中关键字个数(100-200)
$TTL$	6	Time-to-Live,消息洪泛的跳数
$Threshold_{cluster}$	1	节点聚类的阈值
$Similarity_{document}$	1	文本相似度的阈值
$K$	5	超级节点在聚类中选择查询要转发的节点数

7.5.2.1 基于内容搜索模型的查询效果

本实验将基于集合差异度的内容搜索模型和K-漫步者模型做比较， $K$ 和 $TTL$ 都取表1给出的默认值。结果如图7.4所示。从图中可以看出，在节点比较少的时候，基于集合差异度的内容搜索模型的平均召回率比K-漫步者要小。这是由于此时节点之间的相似度过小，即所包含的文本内容差异度大，基于内容的聚类内包含的节点数很少，因此平均查全率比K-漫步者相对要小。但是随着系统规模的扩大，节点逐渐基于内容实现聚类，查全率相应得到提高。因此，和K-漫步者相比，本文提出的模型具有较好的查询效果。

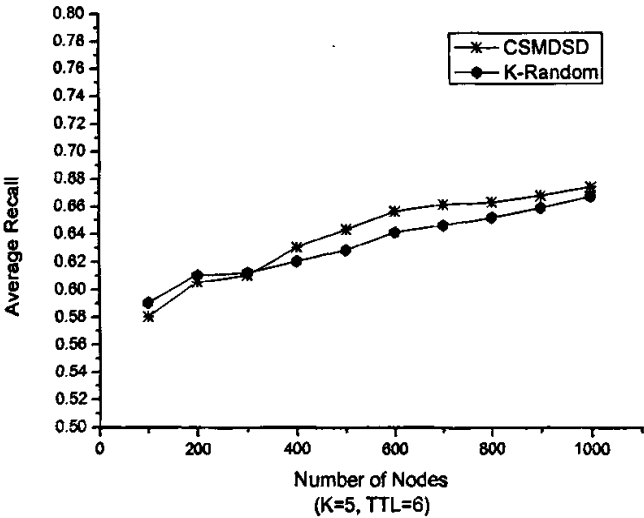


图7.4 基于集合差异度的内容搜索模型与K-漫步者的平均召回率比较

7.5.2.2 TTL对查询效果的影响

基于集合差异度的内容搜索模型的查询效果受到TTL的限制，并且无法访问TTL之外的节点。从图7.5可以发现，增大TTL，可以提高系统的查询效果，并且在TTL为2时，平均召回率增长的斜率最大。随着TTL的增大，召回率的增长越来越慢。这说明本文提出利用集合差异度实现基于内容的聚类模型将文本内容相近的节点有效地聚集在一起，并且类与类之间比较明显的区分开。这样，可以将查询有效地控制在某个范围内，减少了查询时间，优化了查询效果。

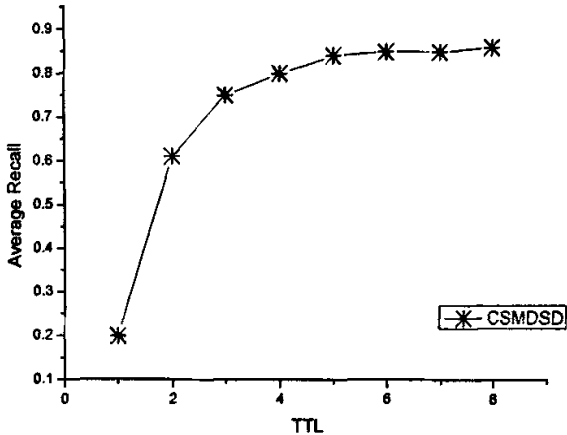


图7.5 不同TTL对平均召回率的影响

7.5.3 查询效率

从两方面考虑基于集合差异度的内容搜索模型的查询效率，包括搜索代价和查询时间。其中，搜索代价的衡量指标使用完成查询所访问的节点数。

7.5.3.1 基于内容搜索模型的查询效率

实验通过对Gnutella、K-漫步者以及基于集合差异度的内容搜索模型比较进行搜索时访问的平均节点数衡量它们的查询效率，模拟结果如图7.6(a)所示。

从图7.6(a)可以看出，K-漫步者和本文模型的平均访问节点数不到Gnutella的一半，并且在系统中节点数目较少时，K-漫步者和基于集合差异度的内容搜索模型之间平均访问节点数差别不大。但是，随着网络规模的扩大，K-漫步者算法的平均访问节点数目明显比基于内容聚类的搜索模型增加的快。这是由于在初始状态，基于内容的聚类模型尚未形成，查询主要按照洪泛算法进行。但是随着节点数目的增多，系统逐渐将节点按照内容相近的形式实现了聚类，相应的查询在某个范围内即可完成。而此时K-漫步者执行的仍然是盲目搜索，因此，访问节点的数目相对要多。但是，二者的搜索代价都大大小于广播式搜索，实验数据表明，在基于内容聚类的搜索模型中，平均访问节点数目是Gnutella系统的40%左右。

### 7.5.3.2 基于内容搜索模型的搜索代价

利用集合差异度实现基于内容聚类的基本思想是利用集合差异度衡量文本之间的内容相关性,并依据节点之间的内容差异度对节点实现聚类。前者克服了传统的VSM利用语义向量之间夹角余弦表示文本相似度所引起的计算复杂的问题,既有效地表示了文本内容,又有效地压缩了数据;后者通过基于内容的聚类可以在某个范围内有效地完成查询,减少了查询时间。实验对基于内容聚类的搜索模型和分别采用VSM以及基于集合差异度作为文本相似度衡量方法的Gnutella系统进行比较,从实验结果中看出,利用集合差异度实现基于内容聚类的查询时间是Gnutella系统的30%左右,如图7.6(b)所示。并且,基于VSM的查询比基于集合差异度的查询所花费的时间要多。

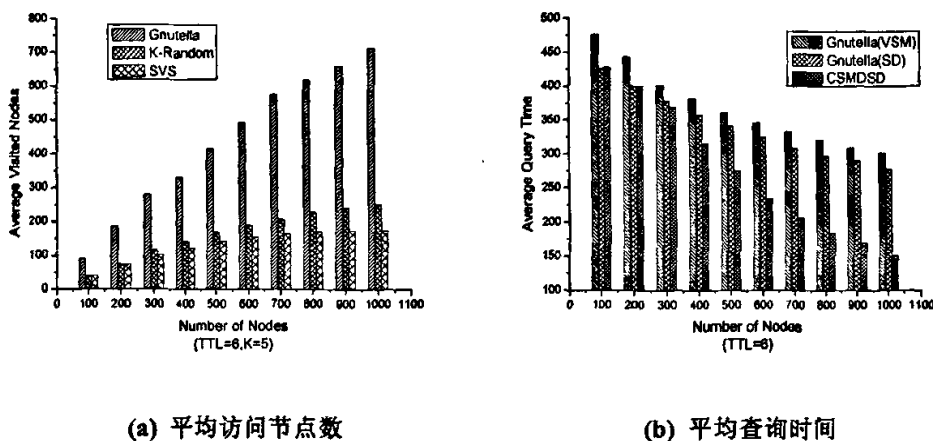


图7.6 基于集合差异度的内容搜索模型查询效率分析图

## 7.6 本章小结

目前P2P系统中基于内容的搜索机制采用传统的向量空间模型VSM描述文本向量,并通过向量之间的夹角余弦衡量文本的相似度。这种方法对于高维语义向量空间中的文本向量,其余弦的计算具有较高的计算复杂度,并且存在语义层和P2P叠加成之间的维数不匹配问题。本章通过利用集合差异度提出一种新的文本相似度表示法,对于多维空间中的文本向量进行了有效的数据压缩,减少了运算量,提高了查询效率;通过提出聚类特征向量的概念,对P2P系统中的节点实现基于内容的聚类模型,将查询控制在一个有效的区域内,减少了冗余消息,降低了查询时间。实验结果表明,该系统不仅具有较高的召回率,而且将搜索成本和查询时间分别降低到Gnutella系统的40%和30%左右。但是,本章在聚类特征向量可加性定理中是假设两个不相交的子集,即没有考虑副本的管理,这是可望在今后的工作中加以改进的。

## 第八章 结束语

### 8.1 本文工作总结

对等网络作为建立在叠加层的网络,凭借灵活性和可扩展性被广泛应用于文件存储、即时消息发送、对等计算等方面。资源管理机制是P2P系统的关键技术,它包括资源的分配机制和资源的定位机制,其中前者决定了后者,并且最终为后者提供依据。资源管理机制的主要功能是返回请求资源的定位信息。如果资源管理机制差,将影响整个系统的运行。为了设计高效的基于内容的资源管理机制,本文深入总结了前人在相关领域的研究成果,分别对P2P系统信息检索领域的三部分技术展开研究,为进一步的工作奠定了基础。

本文首先阐述了对等网络的产生、特征、分类和应用,归纳比较了目前主要的P2P结构的特色,指出了对等网络发展面临的挑战。在全面深入地分析了P2P系统中资源管理机制的基础上,针对P2P叠加层、文件管理层以及检索模型这三个关键技术展开研究,主要解决P2P系统中基于内容搜索模型的效率和性能问题。本文的主要贡献和创新如下:

1.全面深入的综述P2P系统中基于内容的资源管理机制的相关技术。从基于内容的资源管理机制的三个主要组成部分——P2P叠加层、文件管理层、检索模型出发,总结了相关研究工作中存在的主要问题,分析了未来研究工作发展的趋势。

2.提出基于延迟敏感的双向查询Chord系统。针对目前大多数P2P系统的构建没有考虑到物理因素的问题,根据节点和参考点之间的延迟对节点进行分组,减少平均响应时间;通过实现Chord上的双向查询,缩短搜索路径长度。

3.提出了一种基于信誉感知的资源发现算法。针对P2P系统中由节点匿名性所带来的恶言欺诈服务问题,基于Directed-BFS算法,融入信誉的概念,提出信誉感知的资源发现算法,有效的抑制了P2P网络中恶意节点的欺诈行为,降低了系统的消息负载,保证了节点获得服务的可靠性和安全性。

4.基于小世界模型构建女巫攻击防御机制。利用P2P系统的小世界特性,通过把寻找女巫攻击团体的问题转化为最大流最小割问题,引入虚拟节点,依据最短增广路径算法识别女巫攻击节点,降低了恶意节点在整个系统中所占的百分比。

5.提出了一种基于快速映射的内容访问网络。针对当前结构化P2P文件存储系统存在仅支持单关键字的精确匹配,而缺乏支持内容查询的局限性,本文提出一种基于快速映射的结构化P2P存储系统FCAN。FCAN利用FastMap实现从语义空间到P2P空间的映射,在保证文本矢量语义完整性的同时对其进行有效降维,从而实现基于内容的查找;通过对质心法确定的最近邻居进行聚类,有效地减少了冗余信息和查询时间。

6.提出一种利用集合差异度实现基于内容聚类的搜索模型。对于高维语义空间引起的文本相似度计算复杂,以及完全分布式非结构化P2P系统中广播算法带来的大量冗余消息,本文提出利用集合差异度实现基于内容聚类的P2P搜索模型。该模型利用集合差异度定义文本相似度,将文本相似性的计算复杂度控制在线性时间内而有效地减少了查询时间;利用节点之间的集合差异度实现基于内容的聚类,可以降低查询时间,减少冗余消息。

## 8.2 未来工作展望

P2P系统中基于内容的资源管理机制是一个重要的研究课题,是建立各种不同P2P应用系统的基础。由于课题工作量和时间的关系,本文在这方面也只是作了一些初步的研究,我们的工作还不完善,还有许多问题需要进一步深入研究。

1.全面深入的构建混合P2P系统。由于非结构化P2P系统和结构化P2P系统各有优缺点,因此可以考虑将非结构化P2P系统和结构化P2P系统进行综合,取长补短,适应复杂的P2P环境并满足不同的需求。

2.继续深入研究P2P叠加网络拓扑与物理网络拓扑不匹配的问题。由于叠加网络拓扑与物理网络拓扑不匹配,导致P2P流量增大,因此如何高效地构建与物理网络相匹配的叠加网络拓扑正在成为研究的新热点。

3.探求解决P2P系统的免费乘车现象的方法。由于P2P系统中存在大量的免费乘车者,即只索取资源而不共享资源,导致系统的定位性能受到影响。因此如何采取激励机制减少免费乘车者的数量也是一个值得研究的方向。

4.满足新的需求。随着P2P系统的应用不断涌现,不同的应用对P2P系统的资源定位技术产生新的需求。如实现匿名搜索可以保护节点隐私,避免恶意跟踪和攻击。

本文提出的技术能够有效地提高P2P系统资源定位的性能,并且提供了高效、安全的基于内容的资源定位功能,有助于P2P系统在互联网上得到更加广泛的应用。在未来的工作中,我们将继续探讨解决新问题的相关技术。

# 参考文献

- [1] 郑纬民, 胡进锋, 代亚非, 等. 对等计算研究概论. 技术发展年度报告. 中国计算机学会. 2004. <http://www.ccf.org.cn>
- [2] Inc MI. Internet usage statistics, 2006 URL <http://www.internetworldstats.com/stats.htm>
- [3] Consortium IS. Isc internet domain survey host count , 2004 URL <http://www.isc.org/index.pl?/ops/ds/reports/2004-07>
- [4] Schollmeier R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. Proceedings of the 1st International Conference on Peer-to-Peer Computing (P2P'01), 101-102. cLinkoping, Sweden:IEEE Computer Society, 2001
- [5] Graham RL. Peer-to-Peer: Towards a definition. Proceedings of the 1st International Conference on Peer-to-Peer Computing (P2P'01), Lecture. Linkoping, Sweden:IEEE Computer Society, 2001
- [6] Aberer K and Hauswirth M. Peer-to-Peer information systems: Concepts and models, state-of-the-art, and future systems. Proceedings of the 18th International Conference on Data Engineering (ICDE'02), Tutorial Slides. San Jose, CA, USA, 2002.
- [7] Shirr C. What is P2P and what isn't 2001 URL <http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html>
- [8] Milojicic DS, Kalogeraki V and Lukose RM. Peer-to-Peer computing. Technical report, Hewlett-Packard Company, 2002
- [9] Napster website, 1999 URL <http://www.napster.com>
- [10] LLC LW. LimeWire website, 2000 URL <http://www.limewire.com>
- [11] Morpheus. Morpheus file sharing network, 2002 URL <http://www.musiccity.com>
- [12] eDonkey.eDonkey home page, 2002 URL <http://www.edonkey.com>
- [13] iMesh.imesh home page, 2002 URL <http://www.imesh.com>
- [14] LTD SN. KaZaA media desktop, 2002 URL <http://www.kazaa.com>
- [15] Inc B.Bittorrent, 2001 URL <http://www.bittorrent.com>
- [16] Iyer S, Rowstron A and Druschel P. Squirrel: A decentralized peer-to-peer web cache. Proceedings of the 21th ACM Symposium on Principle of Distributed Computing (PODC'02), 213-222. Monterey, California, USA, 2002
- [17] 凌波, 王晓宇, 周傲英, Wee-Siong. 一种基于Peer-to-Peer技术的Web缓存共享系统研究. 计算机学报, 28(2):170-178, 2005
- [18] Society TP. Seti@home project home page, 1999 URL <http://setiathome.ssl.berkeley.edu/>
- [19] University S. Folding@home project home page, 2000 URL <http://folding.stanford.edu/>
- [20] Lab O. fightaids@home project home page, 2003 URL <http://fightaidsathome.stripps.edu/>
- [21] Intel. Netbatch, 1990 URL <http://www.intel.com>
- [22] Networks G. Groove home page, 2002 URL <http://www.groove.net/home/index.cfm>
- [23] Iamnitchi A, Ripeanu M and Foster I. Locating data in (small-world?) P2P scientific



- collaborations. Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), 232-241. Cambridge, MA, USA: Springer-Verlag, 2002
- [24] Iamnitchi A. Resource Discovery in Large-Scale Distributed Environments, Ph.D. thesis, Computer Science, University of Chicago, 2004
- [25] Kubiawicz J, Bindei D, Chen Y, Czerwinski S, Eaton P, Geels D, Gummadi R, Rhea S, Weatherspoon H, Weiner W, Wells C. OceanStore: An Architecture for Global-Scale Persistent Storage. Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 190-201.
- [26] Zhao B, Kubiawicz J and Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, Computer Science Division, U.C.Berkeley, 2001
- [27] Stoica I, Morris R, Liben-Nowell D, Karger D R, Kaashoek MF, Dabek F and Balakrishnan H. Chord: A scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking, 11(1):17-32, 2003
- [28] Rowstron A and Druschel P. Pastry: Scalable, decentralized object location and routing for largescale peer-to-peer systems. Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01), 329-350. Heidelberg, Germany: Springer-Verlag, 2001
- [29] University S and Inc S M. Lots of copies keep stuff safe (lockss) project home page, 2002 URL <http://lockss.stanford.edu>
- [30] Waterhouse S. JXTASearch: Distributed search for distributed networks. Technical report, Sun Microsystems Inc., 2001
- [31] Suel T, Mathur C, Wu J, Zhang J, Delis A, Kharrazi M, Long X and Shanmugasunderam K. ODISSEA: A peer-to-peer architecture for scalable web search and information retrieval. Proceedings of the International Workshop on Web and Databases (WebDB'03), 67-72. San Jose, California, 2003
- [32] Skype. Skype homepage. URL <http://www.skype.com/intl/zh-Hans/helloagain.html>
- [33] University W. traffic statistics, 2003 URL <http://www.stats.net.wisc.edu/>
- [34] For Internet Data Analysis CA. traffic statistics, 2002 URL: <http://www.caida.org/analysis/workload/oc48/>
- [35] Karagiannis T, Broido A, Brounlee N, claffy K and Faloutsos M. Is P2P dying or just hiding? Technical report, Cooperative Association for Internet Data Analysis, 2004.
- [36] Androutsellis-Theotokis S and Spinellis D. A Survey of peer-to-peer File Sharing Technologies. White paper, Electronic Trading Research Unit (ELTRUN), Athens University of Economics and Business, 2002. URL <http://citeseer.ist.psu.edu/androutsellis-theoto02survey.html>
- [37] Ratnasamy S, Francis P, Handley M, Karp R and Shenker S. A Scalable Content-Addressable Network. Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication,

- 161-172. San Diego, CA, USA:ACM Press, 2001
- [38] Clip2com. The Gnutella protocol specification v0.4, 2001  
URL:<http://rfc-gnutella.sourceforge.net/Development>
- [39] Faloutsos M, Faloutsos P and Faloutsos C. On Power-law Relationship of the Internet Topology. Proceedings of the ACM SIGCOMM 1999 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, 251-262. Cambridge, Massachusetts, USA: ACM Press, 1999.
- [40] Karl Aberer, Fabius Klemm, Martin Rajman and Jie Wu. An Architecture for Peer-to-Peer Information Retrieval. School of Computer and Communication Science EPFL, Lausanne, Switzerland, July 2, 2004.
- [41] Watts D J and Strogatz S H. Collective Dynamics of small-world networks. *Nature*, 393:440-442, 1998.
- [42] Ripeanu M, Foster I and Iamnitchi A. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6-1:50-57, 2002.
- [43] Jovanovic M, Annexstein F S and Berman K A. Scalability issues in large Peer-to-Peer networks – a case study of Gnutella. Technical report, University of Cincinnati, 2001
- [44] Jovanovic M A. Modeling Large-scale Peer-to-Peer Networks and a Case Study of Gnutella. Master's thesis, University of Cincinnati, 2000.
- [45] Saroiu S, Gummadi P K and Gribble S D. A Measurement Study of Peer-to-Peer File Sharing systems. Proceedings of the Multimedia Computing and Networking (MMCN'02). San Jose, California, 2002.
- [46] Karagiannis T, Broido A, Brownlee N, claffy K and Faloutsos M. Is P2P dying or just hiding? *Technical report*, Cooperative Association for Internet Data Analysis, 2004.
- [47] DSS C. Gnutella: To the bandwidth barrier and beyond. Technical report, <http://dss.clip2.com>, 2002
- [48] DSS C. Bandwidth barriers to Gnutella network scalability *Technical report*, <http://dss.clip2.com>, 2002
- [49] Ritter J. Why Gnutella can't scale. No, really, 2001 URL  
<http://www.darkridge.com/~jpr5/doc/gutella.html>
- [50] Ripeanu M, Foster I and Iamnitchi A. Mapping the Gnutella network: Proceedings of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6-1:50-57, 2002
- [51] Daswani N and Garcia-Molina H. Query-flood dos attacks in Gnutella. Proceedings of the 9<sup>th</sup> ACM Conference on Computer and Communications Security (CCS'02), 181-192. Washington, DC, USA: ACM Press, 2002.
- [52] Li J, Loo BT, Hellerstein JM, KaashoeK MF, Karger DR and Morris R. On the feasibility of peer-to-peer web indexing and search. *Proceedings of the 2<sup>nd</sup> International Workshop on*

- Peer-to-Peer Systems (IPTP'03)*, 207-215. Berkeley, CA, USA: Springer-Verlag, 2003.
- [53] Maymounkov P and Mazieres D. Kademlia: A peer-to-peer information system based on the xor metric. *Proceedings of the 1<sup>st</sup> International Workshop on Peer-to-Peer Systems (IPTP'02)*, 53-65. Cambridge, MA, USA: Springer-Verlag, 2002.
- [54] Metamachine. The overnet file-sharing networks, 2002 URL <http://www.overnet.com>
- [55] Ganesan P, Sun Q and Garcia-Molina H. YAPPERS: A peer-to-peer lookup service over arbitrary topology. *Proceedings of the Conference on Computer Communications (IEEE INFOCOM'03)*, volume 22, 1250-1260. San Francisco, CA, USA: IEEE Computer Society Press, 2003.
- [56] Loo BT, Huebsch R, Stoica I and Hellerstein J. The case for a hybrid P2P Search infrastructure. *Proceedings of the 3<sup>rd</sup> International Workshop on Peer-to-Peer Systems (IPTP'04)*, 141-150. La Jolla, CA, USA: Springer-Verlag, 2004.
- [57] Castro M, Costa M and Rowstron A I T. Should we build Gnutella on a structured Overlay. *Computer Communication Review*, 34 (1): 131-136, 2004.
- [58] Tsoumakos D and Roussopoulos N. Analysis and comparison of P2P search methods. Technical Report CS-TR-4539, UMIACS-TR-2003-107, University of Maryland, 2003.
- [59] Pandurangan G, Raghavan P and Upfal E. Building P2P networks with good topological properties. *Technical Report*, Brown University, 2001.
- [60] Pandurangan G, Raghavan P and Upfal E. Building low-diameter P2P networks. *IEEE Journal on Selected Areas in Communications*, 21(1):995-1002, 2003.
- [61] Wouhaybi RH and Campbell AT. Phenix: Supporting resilient low-diameter peer-to-peer topologies. *Proceedings of the Conference on Computer Communications (IEEE INFOCOM'04)*, volume 23, 108-119. Hong Kong, China, 2004.
- [62] Cohen E, Fiat A and Kaplan H. A case for associative peer-to-peer overlays. *ACM SIGCOMM Computer Communication Review*, 33(1):95-100, 2003.
- [63] Ramanathan MK, Kalogeraki V and Pruyne J. Finding good peers in peer-to-peer networks. *Proceedings of the 16<sup>th</sup> International Parallel and Distributed Processing Symposium (IPDPS'02)*, 232-239. Fort Lauderdale, FL, USA: IEEE Computer Society, 2002.
- [64] Sripanidkulchai K, Maggs B and Zhang H. Efficient content location using interest-based locality in peer-to-peer systems. *Proceedings of the Conference on Computer Communications (IEEE INFOCOM'03)*, volume 22, 2166-2176. San Francisco, CA, USA, 2003.
- [65] 黄道颖, 黄建华, 庄雷, 李祖鹏. 基于主动网络的分布式P2P网络模型. *软件学报*, 15(7):1081-1089, 2004.
- [66] Peer-To-Peer technology company P. FastTrack, 2001 URL <http://www.fasttrack.nu>
- [67] LLC LW. Rfc-Gnutella 0.6, 2002 URL <http://rfc-gnutella.sourceforge.net/development>
- [68] Padmanabhan VN and Subramanian L. An investigation of geographic mapping techniques for internet hosts. *Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 173-185. San

- diego, CA, USA, 2001
- [69] Krishnamurthy B and Wang J. Topology modeling via cluster graphs. Proceedings of the 1<sup>st</sup> ACM SIGCOMM Workshop on Internet Measurement (IMW'01), 19-23. San Francisco, California, USA: ACM Press, 2001.
  - [70] E. Adarand, B. A. Huberman. Free-riding on Gnutella. First Monday (Peer-reviewed Journal on the Internet), 5(10):8, 2000.10.
  - [71] P. Resnick, R. Zeckhauser: Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation Systems, Working Paper for the NBER workshop on empirical studies of electronic commerce, Jan 2001.
  - [72] Kalogeraki V, Gunopulos D and Zeinalipour-Yazti D. A local search mechanism for peer-to-peer networks. Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, 300-307. McLean, Virginia, USA, 2002.
  - [73] Zhuang Z, Liu Y, Xiao L and Ni L M. Hybrid periodical flooding in unstructured peer-to-peer networks. Proceedings of the 32nd International Conference on Parallel Processing (ICPP'03), 171-178. Kaohsiung, Taiwan: IEEE Computer Society, 2003.
  - [74] Yang B and Garcia-Molina H. Improving search in peer-to-peer networks. Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), 5-14. Vienna, Austria: IEEE Computer Society, 2002.
  - [75] Andy Oram: Peer to Peer: Harnessing the power of disruptive technologies. ISBN 0-596-00110-X, 2001
  - [76] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems*, 2005.
  - [77] VeriSign, Inc. 487 East Middlefield Road, Mountain View, CA 94043, [www.verisign.com](http://www.verisign.com)
  - [78] D. Mazieres, M. Kaminsky, M.F. Kaashoek, E. Witchel, Separating Key Management from File System Security, 17th SOSP, 1999, pp. 124-139.
  - [79] 周晋, 路海明, 李衍达. 用Small-world设计无组织P2P系统的路由算法. 软件学报, 15(6):915-923, 2004.
  - [80] J. Douceur. The Sybil attack. In IPTPS, 2002.
  - [81] L. von Ahn, M. Blum, N.J. Hopper, and J.Langford. CAPTCHA: Telling humans and computers apart. In Eurocrypt, 2003.
  - [82] R. Bazzi and G. Konjevod. On the establishment of distinct identifies in overlay networks. In ACM PODC, 2005.
  - [83] H.Yu, M. Kaminsky, P.B.Gibbons, and A. Flaxman. SybilGuard: Defending against Sybil attacks via social networks. Technical Report IRP-TR-06-01, Intel Research Pittsburgh, June 2006.
  - [84] Adar E and Huberman BA. Free riding on Gnutella. Technical Report, Xerox PARC, 2000.
  - [85] Lv Q, Cao P, Cohen E, Li K and Shenker S. Search and replication in unstructured peer-to-peer networks. Proceedings of the 16th Annual International Conference on

- Supercomputing (ICS'02), 84-95. New York City, NY, USA, 2002
- [86] LLC L W. Rfc-Gnutella 0.6, 2002 URL <http://rfc-gnutella.sourceforge.net/Development>
- [87] Clarke I, Sandberg O, Wiley B and Hong T W. Freenet: A distributed anonymous information storage and retrieval system. Proceedings of the ICSI Workshop On Design Issues in Anonymity and Unobservability, 46-66. Berkeley, CA, USA: Springer-Verlag, 2000.
- [88] Crespo A and Garcia-Molina H. Routing indices for peer-to-peer systems. Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), 23-34. Vienna, Austria: IEEE Computer Society, 2002.
- [89] Adamic L A, Lukose R M, Puniyani A R and Huberman B A. Search in power-law networks. Physical Review E, 64, 2001.
- [90] Menasce D and Kanchanapalli L. Probabilistic Scalable P2P resource location services. SIGMETRICS Perf. Eval. Review, Volume 30, 48-58. 2002
- [91] Tsournakos D and Roussopoulos N. Adaptive probabilistic search for peer-to-peer networks. Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P'03), 102-109. Linköping, Sweden: IEEE Computer Society, 2003.
- [92] Kumar A, Xu J and Zegura E W. Efficient and scalable query routing for unstructured Peer-to-Peer networks. Proceedings of the Conference on Computer Communications (IEEE INFOCOM'05). Miami, USA: IEEE Computer Society, 2005.
- [93] Chawathe Y, Ratnasamy S, Breslau L, Lanham N and Shenker S. Making Gnutella-like P2P systems scalable. Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, 407-418. Karlsruhe, Germany: ACM Press, 2003.
- [94] David Karger, Eric Lehman, Tom Leighton, Matthew Levine, Daniel Lewin, Rina Panigrahy. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 654-663.
- [95] FIPS 180-1. Secure Hash Standard. U.S. Department of Commerce/NIST, National Technical Information Service, Springfield, VA, Apr. 1995.
- [96] C. Greg Plaxton, Rajmohan, and Andrea W. Richa. Accessing nearby Copies of Replicated Objects in a Distributed Environment. Proceedings of ACM SPAA. ACM, June 1997.
- [97] F. Cornelli, E. Damiani, S. D.C. di Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servants in a P2P network. In Proceedings of the 11<sup>th</sup> World Wide Web Conference, 2002.
- [98] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In IEEE Conference on E-Commerce (CEC'03), 2003.
- [99] S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina, The Eigen Trust algorithm for reputation management in P2P networks. WWW2003; May 20-24, 2003; Budapest, Hungary.

- [100] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks. *Proceedings of ACM SIGCOMM*, pp. 175-186.
- [101] M. Li, W.-C. Lee and A. Sivasubramaniam. Semantic Small World: An Overlay Network for Peer-to-Peer Search. Technical Report CSE 04-016. Pennsylvania State University, 2004.
- [102] Klemm A, Lindemann C, Vernon MK and Waldhorst OP. Characterizing the query behavior in peer-to-peer file sharing systems. *Proceedings of the 4<sup>th</sup> ACM SIGCOMM Conference on Internet Measurement*, 55-67. Taormina, Sicily, Italy: ACM Press, 2004.
- [103] Pandurangan G, Raghavan P and Upfal E. Building P2P networks with good topological properties. *Technical Report*, Brown University, 2001.
- [104] P. Ganesan, G. S. Mankuy. Optimal Routing in Chord. In 15th ACM SIAM Symposium on Discrete Algorithm (SODA), 2004.
- [105] Beverly Yang, Hector Garcia-Molina. Designing a super-peer network. 19th International Conference on Data Engineering, IEEE Computer Society. March 5-March 8.
- [106] Brite, a network topology generator. <http://www.cs.bu.edu/brite/>
- [107] M.S. Khambatti, \* K.D. Ryu,\* P. Dasgupta\*, Efficient Discovery of Implicitly formed Peer-to-Peer Communities, *International Journal of Parallel and Distributed Systems and Networks*, Vol. 5, No. 4, 2002
- [108] Diego Gambetta.,Can We Trust? In, *Trust: Making and Breaking Cooperative Relations*, Gambetta, D (ed.). Basil Blackwell. Oxford, 1990, pp. 213-237
- [109] Abdul-Rahman A, Hailes S. A distributed trust model: In *Proceedings of the 1997 New security Paradigms Workshop*, Cambrian, ACM Press, 1998.
- [110] Srinivasan, T.V.; Vincelette, C.J. Dasgupta, D. Overlay network applications for network modernization and positioning for the future. 4th IEEE Region 10th International Conference-TENCON '89 Nov pp.22-24.
- [111] M. Li, W. C. Lee and A. Sivasubramaniam. Neighborhood signatures for searching P2P networks. In *Proceedings of International Database Engineering and Application Symposium (IDEAS)*, pages 149-158, July 2003.
- [112] FALOUTSOS C, K2I D LIN. Fastmap: A fast algorithm for indexing datamining and visualization of traditional and multimedia datasets. In *Proceedings of ACM SIGMOD*, pages 163-174, 1995.
- [113] Cheng XueQi, L Jianming and Zhou Zhaotao. P2P Full Text Information Retrieval Based on Centroid Method. *Journal of Computer Research and Development (in Chinese)*, 2004, 41(12):2148-2153.
- [114] M. W. Berry, Z. Drmac and E. R. Jessup. Matrices, Vector spaces, and Information Retrieval. *Society for Industrial and Applied Mathematics Review*, 41(2):335-362, 1999.
- [115] 宋建涛, 沙朝锋, 杨智应, 朱洪. 语义对等网构造及搜索机制研究. *计算机研究与发展*, 41(4):645-652, 2004.
- [116] M. Bawn, G. S. Manku and P. Raghavan. SETS: Search enhanced by topic segmentation. In

- Proceedings of ACM SIGIR, pages 306-313, July 2003.
- [117] A. Ianmitchi, M. Ripeanu and I. T. Foster. Locating data in peer-to-peer scientific collaborations. In Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS), pages 232-241, March 2002.
  - [118] C. H. Ng, K. C. Sia and C.H. Chang. Advanced peer clustering and firework query model in the peer-to-peer network. In Proceedings of International World Wide Web Conference (WWW), May 2003, Poster.
  - [119] Hai Jin, Yijiao Yu. SemreX: a Semantic Peer-to-Peer Scientific Reference Sharing System. IEEE Computer Society Washington. DC. USA, pages 97,2006.

# 致 谢

时光荏苒，在科大的学习生活就要结束。几年来，恩师母校的培养和教诲，同学之间朝夕相处，情同手足都牢牢地印在我的脑海中，并将在我今后的工作和生活中产生深远的影响。值此论文完成之际，谨向多年来所有关心和帮助过我的人表达最衷心的感谢！

首先真诚感谢我的导师杨寿保教授，在攻读博士学位期间，杨老师给我提供了宽松和谐的学习和研究环境，在研究方向的把握、研究工作的开展、研究思路的选取给予我悉心指导。杨老师深厚渊博的学识、严谨务实的学风、平易近人的长者风范，永远是我学习的榜样。同时，杨老师幽默开朗的性格、乐观积极的生活态度、勤奋忘我的工作作风更值得我终生学习。在此，谨向杨老师表示衷心的感谢。

感谢网络中心的所有老师在我的学习生活中给予的帮助。感谢孙适老师，他的精神使我感到自我的懒惰；感谢张焕杰老师，让我了解了许多实践知识；感谢徐为民老师、汝梅老师、封蕾老师在生活中的关心和帮助。

感谢一直帮助和鼓励我的网格组和P2P组的所有成员：郭磊涛、周金洋、王莉苹、彭晓燕、房向明、陈颖、申凯、刘鹏展、裴艳琴、刘亮、张然美、姜燕、韦东、路卫娜、郭良敏、刘玉枚，你们认真的研究态度影响着我，积极的生活态度感染着我，每次与你们的交流都很都能从中得到启发。特别要感谢的郭磊涛和周金洋，对我平时学习中的帮助，总之，我的博士论文的完成是与你们的帮助和协作分不开的。

感谢实验室的其他成员：王绍林、孙伟峰、张蕾、董阔、王大鹏、胡云、徐钊、史凌志、武斌、滕达、陈为、赵叶红等众多的兄弟姐妹以及郑明春老师，谢谢你们给予我在生活当中的关心和帮助。

感谢我的父母，他们不但多年来一直默默支持我在学业上不断前行，而且以自身的辛劳、正直和宽容教给我做人道理，他们在精神和物质上给了我巨大的帮助和支持，让我有条件 and 勇气完成学业；感谢我的男朋友高鹰，是他无私的支持和信任给予我完成学业的动力。

最后，谨向百忙之中抽出宝贵时间评阅论文的老师表示深深的谢意。

王 菁

二〇〇七年四月于中国科大



## 攻读博士学位期间发表的论文

- [1] 王菁, 杨寿保, 郭磊涛, 周金洋: 基于延迟敏感的双向查询Chord系统. 《小型微型计算机系统》, Vol.28, No.4, P604~608, 2007.4.
- [2] Jing Wang, Shoubao Yang, Leitao Guo: A Bidirectional Query Chord System Based on Latency-Sensitivity. In the Fifth International Conference on Grid and Cooperative Computing (GCC2006). IEEE Computer Society Order Number P2694, pp.164-167, 2006.
- [3] Jing Wang, Shoubao Yang, Ying Gao, Leitao Guo: FCAN: A Structured P2P System Based on Content Query. In the Fifth International Conference on Grid and Cooperative Computing (GCC2006). IEEE Computer Society Order Number P2694, pp.113-120, 2006.
- [4] Jing Wang, Shoubao Yang: Content-based Clustered P2P Search Model Depending on Set Distance. In the 2006 IEEE/WIC/ACM International Workshop on P2P Computing and Autonomous Agent. IEEE Computer Society, pp.471-476. 2006.
- [5] 王菁, 张焕杰, 杨寿保, 高鹰: 利用集合差异度实现基于内容聚类的P2P搜索模型. 《中国科学院研究生院学报》, Vol.24, No.2, P241~247, 2007.
- [6] 王菁, 杨寿保, 高鹰, 郭磊涛. FCAN: 一种基于快速映射的内容访问网络. 《系统仿真学报》. 已录用.
- [7] 姚远, 王菁, 刘永: 交换机时间同步管理系统及其实现. 《电信科学》, Vol.20, No.3, P68~71, 2004.
- [8] Leitao Guo, Shoubao Yang, Jing Wang, and Jinyang Zhou. Trust Model Based on Similarity Measure of Vectors in P2P Networks. In the Fourth International Conference on Grid and Cooperative Computing (GCC2005), LNCS 3795, pp.836-847, 2005. (SCI 索引, IDS Number: BDQ17) (EI Accession Number: 06229908957)
- [9] 郭磊涛, 杨寿保, 王菁, 周金洋: P2P网络中基于矢量空间的分布式信任模型. 《计算机研究与发展》, Vol.43, No.9, P1564-1570, 2006. (EI Accession Number: 064610239630)
- [10] Jinyang Zhou, Shoubao Yang, Leitao Guo, Jing Wang, Ying Chen: Towards Reputation-Aware Resource Discovery in Peer-to-Peer Networks. In the Fourth International Conference on Grid and Cooperative Computing (GCC2005), LNCS

- 3795, pp.930-940, 2005. (SCI 索引, IDS Number: BDQ17)(EI Accession Number: 06229908964)
- [11] 周金洋, 杨寿保, 郭磊涛, 王菁, 陈颖: P2P网络中一种基于信任感知的资源发现算法. 《小型微型计算机系统》, Vol.27, No.10, P1798-1802, 2006.
- [12] 周金洋, 杨寿保, 郭磊涛, 王菁: Gnutella环境下基于兴趣和信任的资源发现算法. 华中科技大学学报 (自然科学版), Vol.34, No.9, 2006
- [13] **Jing Wang**, Shoubao Yang, Jun Zhu, Weina Lu: Sybil Defense Mechanism Based on Small World. In the Sixth International Conference on Grid and Cooperative Computing (GCC2007). (Submitted)

# 攻读博士学位期间参加的科研项目

- [1] 国家自然科学基金项目, 基于计算市场模型的网格资源管理研究(60273041)
- [2] 国家自然科学基金项目, 网络计算环境中信任感知的资源交易模型(60673172)
- [3] 国家863项目, 网格节点--合肥网格节点的建设及若干典型网格应用的研制(2002AA104560)

# P2P系统中资源管理机制的研究

作者: [王菁](#)  
学位授予单位: [中国科学技术大学](#)  
被引用次数: 3次

## 本文读者也读过(3条)

1. [孙亮](#) [P2P网络中资源管理机制的研究](#)[学位论文]2006
2. [乐利锋](#) [P2P系统中网络优化与资源共享机制研究](#)[学位论文]2008
3. [任浩](#) [P2P覆盖网拓扑优化技术研究](#)[学位论文]2007

## 引证文献(3条)

1. [李继容](#), [张先勇](#) [一种基于P2P技术的网络管理系统的设计](#)[期刊论文]-[光通信研究](#) 2009(4)
2. [李静](#), [张雪东](#) [基于P2P技术的农产品供应链云计算信息平台构建](#)[期刊论文]-[湖北农业科学](#) 2012(7)
3. [王杨](#), [严远亭](#) [基于P2P技术的云计算教育服务平台设计与实现](#)[期刊论文]-[计算机教育](#) 2010(16)

引用本文格式: [王菁](#) [P2P系统中资源管理机制的研究](#)[学位论文]博士 2007