

# 自适应的 P2P 网络路由方法研究

吴亚辉, 邓 苏, 黄宏斌

WU Ya-hui, DENG Su, HUANG Hong-bin

国防科技大学 信息系统与管理学院, 长沙 410073

College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China

E-mail: yahui\_wu@163.com

**WU Ya-hui, DENG Su, HUANG Hong-bin. Research of adaptive routing method on P2P network. Computer Engineering and Applications, 2009, 45(28): 120-122.**

**Abstract:** After analyzing a series of methods about resource discovery in P2P network, this paper proposes a query-based routing method in P2P network. In this algorithm, the node records the received query and the nodes that can answer the query very well, in these nodes, those that have longer distance are recorded firstly. Simulation experiments demonstrate that the method can enhance the searching efficiency very well with the number of queries becoming more and more.

**Key words:** Peer-to-Peer(P2P); semantic similarity; routing table; searching method; LightFlood

**摘 要:** 在分析一系列 P2P 网络资源发现方法的基础上, 提出了一种基于查询的 P2P 网络路由方法(Query-based Routing Method in P2P network, QRM 方法)。在该方法中, 节点会记录收到的查询以及满足该查询的节点, 在这些节点中, 优先存储那些距离较远的节点。仿真实验表明, 随着查询的增多, 该方法能有效地提高查询效率。

**关键词:** 对等网; 语义相似度; 路由表; 搜索方法; LightFlood

**DOI:** 10.3778/j.issn.1002-8331.2009.28.035 **文章编号:** 1002-8331(2009)28-0120-03 **文献标识码:** A **中图分类号:** TP393

## 1 引言

P2P(Peer-to-Peer)技术是近年研究的热点,它在文件共享、搜索引擎、分布计算等领域有着越来越广泛的应用。不同于传统的 C/S 模式, P2P 网络中的节点在功能上是对等的,既可以充当客户机,也可以充当服务器的角色,节点间可以建立连接进行直接交流,具有良好的自组织能力和可扩展性。由于 P2P 网络中的资源是分散地存储在各个终端节点上,缺乏集中的控制,如何进行资源的高效定位就成为 P2P 系统研究的核心问题。

最早出现的 P2P 网络是集中式索引结构,以 Napster 为代表,在该结构中,所有节点的索引信息都存储在一个中心服务器上,所有的查询请求都要发送到这个中心服务器,这很容易导致单点失效的问题。随后出现了完全分布式结构,主要分为分布结构化和分布非结构化两种。分布结构化网络(以 CAN<sup>[1]</sup>、Chord<sup>[2]</sup>、Pastry<sup>[3]</sup>和 tapestry<sup>[4]</sup>等为代表)主要利用分布式哈希 DHT 算法进行资源的发现和定位, DHT 的主要思想是每个节点通过 hash 函数映射为一个值 NodeID, 节点根据自己的 NodeID 维护一张路由表,查找时,将查询消息按照同样的 hash 函数映射为一个 ID,找到与 ID 值最近的节点,然后按照路由表中的内容进行选择性转发,从而在有限的步长内就可以定位到所需资源。非结构化网络以 Gnutella<sup>[5]</sup>为代表,它采用泛洪算法进行资源的发现与定位,虽然简单,但是却会带来大量的消息负担,为了提高非结构化 P2P 网络资源发现的效率,出现了许多改进算法。

文献[6]提出如果本地节点对另一个节点的某个资源感兴趣,则认为本地节点可能对那个节点的其他内容也感兴趣,所以本地节点就建立一条到那个节点的捷径(shortcut),以后通过捷径就可以直接找到所需资源。

文献[7]扩展了文献[6]的方法,提出 INGA 方法,它给每个节点增加一个推荐人节点表,记录那些从其他节点发起而经过它的查询以及查询发起节点信息,从而使捷径能够为其他节点所共享,但是在路由表中的信息很少时只能采取随机漫步方式进行查找,查找效率不高,而且,考虑到路由表的容量有限,它可能会存储大量的近距离节点,使得很多的远距离节点没有空间存储,只能采取泛洪算法进行查找,从而影响到查询效率。

文献[8]提出将查询分为两步,第一步用较小的步长,第二步在一个非 DHT 的环内进行查询,对于那些副本较多的资源通过第一步基本就能找到,而对于副本较少的非流行文件,经由第二步比盲目的泛洪可以大大减少消息的冗余。所以,该算法可以从一定程度上提高查询效率。

文献[9]提出 LightFlood 方法,和文献[8]类似,它也将查询分为两步,只是它将整个 P2P 网络中的节点组成一个个子树(FloodNet),第一步的 TTL 值也很小,在 TTL=0 时,以 TTL=0 的这些节点为种子(seeds),从这些节点开始进入第二步查询,第二步要在各个 FloodNet 中进行,大大减少了消息的冗余。

该文针对非结构化 P2P 网络,假设查询是完全随机的,结合文献[7]和文献[9]的优点,提出 QRM 方法。在该方法中,也将

**作者简介:** 吴亚辉(1983-),男,硕士研究生,研究方向:信息管理,智能决策,分布式计算;邓苏(1963-),男,教授,博士生导师,研究方向:信息管理,智能决策,数据仓库,分布式计算;黄宏斌(1975-),男,博士,副教授,研究方向:信息管理,智能决策,分布式计算与混合计算。

**收稿日期:** 2008-05-28 **修回日期:** 2008-09-26

网络中的节点组成多个 FloodNet, 需要泛洪时采取 LightFlood 方法, 同时每个节点也维护各自的路由表, 并且根据查询结果优先存储距离较远的节点, 因为每个节点的存储空间有限, 不可能与所有感兴趣的节点建立直接联系, 仿真实验表明该方法能在一定程度上降低消息冗余, 提高查询成功率。下面对算法进行具体描述。

## 2 算法描述

### 2.1 概述

首先进行一些定义:

起始节点: 发起查询的那个节点。

朋友节点: 满足查询条件的节点。

朋友节点列表(topic, ID, hop): 朋友节点列表存储的是关于查询主题的朋友节点信息, topic 代表查询主题, ID 代表节点的唯一标志符, hop 代表查询起始节点到该节点的距离。

中介节点表(topic, ID, hop): 如果从起始节点发起的查询途经某个节点, 则就将起始节点的信息存储到这个节点的中介节点表中, 表中各个属性的含义和朋友节点表中的一样。

在该方法中每个节点维护两张路由表: 朋友节点表和中介节点表, 这两个表都是优先存储那些距离较远的节点, 路由表的建立见 2.3 节。文中不考虑查询频率的影响, 假设查询是完全随机的。

### 2.2 语义相似度计算

为了判断一个节点的内容能否满足某个查询的需要, 将节点内容和查询主题进行一定的语义相似度计算, 查询起始节点可以根据实际情况规定一个值  $\theta$ , 相似度超过该值的节点, 就认为满足主题的需要。为了进行语义相似度计算, 将用到向量空间模型 VSM<sup>[10]</sup>, 利用该模型每个查询和节点中的每个资源都用一个向量来表示。假设节点基于同样的概念集, 向量的具体形式为:

$$P = \{(C_1, w_1), (C_2, w_2), (C_3, w_3), \dots, (C_n, w_n)\}$$

$C_i$  是某个查询或者资源中的一个关键词,  $w_i$  是该关键词在该查询或资源中的权重。所以为了比较某个查询和文档的相似度, 只需要比较它们的向量的相似度即可。

假设某个查询  $Q$  的向量表示为:

$$P_1 = \{(C_{11}, w_{11}), (C_{12}, w_{12}), (C_{13}, w_{13}), \dots, (C_{1n}, w_{1n})\}$$

节点的某个资源  $R$  的向量表示为:

$$P_2 = \{(C_{21}, w_{21}), (C_{22}, w_{22}), (C_{23}, w_{23}), \dots, (C_{2n}, w_{2n})\}$$

则用文献[11]中的方法计算它们的相似度:

$$Sim(Q, R) = Sim(P_1, P_2) = \sum_{i=1}^{i \leq n} \sum_{j=1}^{j \leq m} [Sim(C_{1i}, C_{2j}) \times (w_{1i} \times w_{2j})]$$

$Sim(C_{1i}, C_{2j})$  表示任意两个概念之间的相似度, 用文献[12]中的方法来计算它的值, 在该方法中概念按其含义组成一个层次结构, 称为概念树, 计算公式如下:

$$Sim(C_{1i}, C_{2j}) = \begin{cases} e^{\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}, & \text{if } (C_{1i} \neq C_{2j}) \\ 1, & \text{if } (C_{1i} = C_{2j}) \end{cases}$$

其中  $l$  是概念  $C_{1i}$  和  $C_{2j}$  在概念树上的最小距离,  $h$  是两个概念在概念树上的最小深度,  $\alpha, \beta$  用来调整  $l$  和  $h$  对相似度的影响程度。

查询和路由表中记录 topic 属性的相似度也用同样的方式进行处理。

### 2.3 路由表的建立

根据起始节点到朋友节点的距离来决定是否将该节点加

入到起始节点的朋友节点表, 在朋友节点表未满足时, 直接将查到的所有朋友节点添加到起始节点的朋友节点表中, 如果朋友节点表已满, 则首先计算起始节点到新查到的朋友节点的距离 (NewHop), 然后找到朋友节点表中 hop 值最小的那条记录, 如果该记录的 hop 属性值小于 NewHop, 则用这个新的朋友节点替换该记录, 如果 hop 属性最小的记录有多个, 则随机选择其中一条进行替换, 否则, 丢弃这个新的朋友节点。对于查询过程经过的每一个节点都要更新自己的中介节点表, 在中介节点表中记录本次查询主题以及起始节点的信息, 更新的方法与朋友节点表相同。查询过程结束, 路由表中在本次查询中用到的记录, 如果不能返回满意的结果, 则删除。

随着查询过程来不断调整朋友节点表和中介节点表中的信息, 从而缩短与那些距离较远的节点之间的距离, 提高网络的查询效率。下面就具体介绍这两个表是如何随着查询过程不断更新的。

当关于某一主题的查询经过一个节点时, 该节点要记下查询消息此时的 TTL 值, 假设记为 RTTL, 称为该节点的 RTTL 值, 同时将起始节点加入到该节点的中介节点表中, 如果中介节点表已满, 则计算查询消息的 TTL 初值与 RTTL 值之差, 这个差值的绝对值就代表起始节点到该节点的距离, 然后就按照上面介绍的思想来更新中介节点表(即将该距离与中介节点表中 hop 属性的最小值进行比较, 如果距离大于 hop 属性的最小值, 则用本次查询的起始节点替换 hop 属性值最小的那条记录, 否则丢弃)。如果网络中的某个节点满足查询主题, 则它就向起始节点发送应答消息, 应答消息(response)应按照查询所经过的路径进行返回, 应答消息要将自己再收到查询消息时记下的查询消息的 TTL 值, 即 RTTL 返回, 接收到应答消息的节点, 首先检查应答节点能否满足查询主题, 如果满足, 就认为该应答节点是自己的朋友节点, 然后就更新自己的朋友节点表, 更新方法同上, 该节点到返回应答消息的节点的距离等于该节点的 RTTL 值与应答消息返回的 RTTL 之差的绝对值。

### 2.4 搜索方法

首先按照 LightFlood 方法, 在整个 P2P 空间形成了一系列 FloodNet, 它们是一个逻辑结构, 文献[9]对其形成过程进行了详细的叙述, 每个 FloodNet 的结构如图 1 所示, 当在 FloodNet 中进行查询时, 每个节点会将查询消息转发到在该 FloodNet 中与它直接相连的那些节点上, 比如和节点 E 直接相连的节点就是节点 B、H、I, 从节点 E 转发的消息就会转发到节点 B、H、I 中的一些节点上。

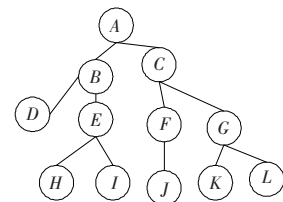


图 1 FloodNet

该文的查询也按照 LightFlood 方法分为两步: 假设从节点 A 发起一次查询主题为 topic 的查询  $Q$ , 表示为  $Q(topic, ID, FirstTTL, secondTTL)$ , FirstTTL 和 secondTTL 分别表示第一步和第二部的步长, 这样  $Q$  的 TTL 值就是 FirstTTL 和 secondTTL 之和。首先进行第一步查询: 节点 A 首先查看本地内容能否满足要求, 如果不满足, 且消息的  $TTL > 0$ , 则选择  $K$  个节点将消息转发出去, 同时消息的 FirstTTL 值减 1。这  $K$  个节点的选择遵

循的原则就是:优先选择朋友节点表中语义相似度与查询主题达到一定阈值的节点,假设存在  $x$  个,如果  $x < K$ ,则从中介节点表中选择满足查询主题的添加进来,假设找到  $y$  个,如果  $(x+y) < K$ ,则从底层物理连接中随机选择  $K-x-y$  个添加进来,如果与节点  $A$  直接相连的节点个数  $num < (K-x-y)$ ,也只能选择  $num$  个。整个查询过程接收到消息的每个节点都进行和节点  $A$  同样的处理。如果当  $FirstTTL=0$  时,还没有查到所需节点,则从这些  $FirstTTL=0$  的节点开始进行第二步处理,这一步基本和第一步一样,只是消息每转发一次  $secondTTL$  值减 1,而且当消息需要转发时,如果朋友节点表和中介节点表中满足查询主题的节点达不到  $K$  个,这时就不像第一步那样直接从底层物理连接中选择,而是从这些节点所在的 FloodNet 中进行选择,如图 1 所示,节点  $B$  就会从节点  $A、D、E$  中进行选择。

### 3 实验及算法分析

为了检验 QRM 算法的有效性,用 Java 编写仿真程序进行验证,网络拓扑结构用 BRUTE 生成,共生成 8 000 个节点,进行 10 000 次查询,这些查询是随机的,同时假设每个节点的朋友节点表最多只能容纳 500 条记录,中介节点表最多容纳 100 条记录,查询仿照 LightFlood 算法分为两步,同时设第一步的  $TTL$  值  $< 4$ 。主要从网络流量和查询成功率两个方面与 LightFlood 和 INGA 算法进行比较,网络流量规定为平均每条查询所产生的消息数目。依据该算法,在查询比较少的时候,由于各个节点朋友节点表和中介节点表中的记录非常少,几乎只能采取 LightFlood 算法进行查询,网络流量应该和 LightFlood 算法相近,随着查询的发展,路由表中的内容逐渐多起来,这样很多查询就可以通过路由表中的信息直接找到,很多查询只需要一步就能定位到所需资源,所以网络流量会大大减少。和 INGA 算法比较接近,随着查询的进一步发展,当路由表的容量满的时候,由于存储的主要是那些距离较远的节点,就会比 INGA 算法的网络流量更小,这通过实验结果图 2 就可以看出。

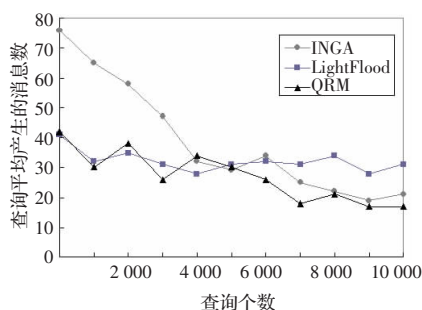


图2 网络流量

等查询基本稳定,才开始计算查询成功率,因为在查询较少时路由表中的信息也很少,大部分查询还只能采取 LightFlood 方法进行泛洪,其查询成功率几乎和 LightFlood 方法相同,对于 LightFlood 方法,因为它不记录以前查询的任何历史信息,也不采取任何方式指引它向可能返回满意信息的节点转发信息,可以说还是采取盲目泛洪的方式,在很多情况下需要较大的步长才能找到满意的结果,所以在路由表中信息较少时不能充分体现算法的优越性,为此统计查询到达 6 000 条以后的成功率,从实验结果图 3 可以看出在消息的  $TTL$  值很小时也能获得很高的查询成功率,所以该算法在一定程度上提高了查询成功率。

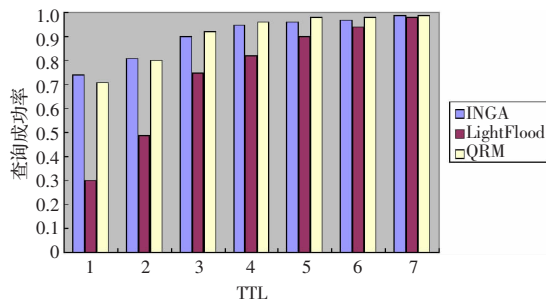


图3 查询成功率

### 4 结束语

提出了一种根据节点到查询起始节点的距离来建立和维护路由表的方式,仿真实验表明该算法有一定的优越性,在今后的工作中,要进一步考虑访问频率的影响,在实际的 P2P 网络中来验证。

### 参考文献:

- [1] Ratnasamy S, Francis P, Handley M, et al. A scalable content-addressable network[C]//proc ACM SIGCOMM, San Diego, CA, 2001: 161-172.
- [2] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for Internet applications[C]//Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, San Diego, USA, August 2001.
- [3] Druschel P, Rowstron A. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems[C]//Proc 18th I-FIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), November 2001.
- [4] Zhao B Y, Kubiawicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing[R]. University of California at Berkeley, 2001.
- [5] Bordignon F, Tolosa G, Gnutella. Distributed system for information storage and searching model description[EB/OL]. (2002). [http://sise.ttu.ee/it/vorgutarkvara/wav4101/paper\\_final\\_gnutella\\_english.pdf](http://sise.ttu.ee/it/vorgutarkvara/wav4101/paper_final_gnutella_english.pdf).
- [6] Sripanidkulchai K, Maggs B, Zhang H. Efficient content location using interest-based locality in peer-to-peer systems[C]//22nd Int Conf of the IEEE Computer and Communications Societies (INFOCOM), 2003.
- [7] L'oser A, Staab S, Tempich C. Semantic social overlay networks[J]. IEEE Journal on Selected Areas in Communications, 2007, 25(1): 5-14.
- [8] Chen Shi-ping, Zhang Zhan, Chen Shi-gang, et al. Efficient file search in non-DHT P2P networks[J]. Computer Communications, 2008, 31: 304-317.
- [9] Jiang S, Guo L, Zhang X. LightFlood: An efficient flooding scheme for file search in unstructured peer-to-peer systems[C]//Proceedings of ICPP, October 2003.
- [10] Berry M, Drmac Z, Jessup E. Matrices, vector spaces, and information retrieval[J]. SIAM Review, 1999, 41(2): 335-362.
- [11] 陈汉华, 金海, 宁小敏, 等. SemreX: 一种基于语义相似度的 P2P 覆盖网络[J]. 软件学报, 2006, 17(5): 1170-1181.
- [12] Yuhua L, Bandar Z A, McLean D. An approach for measuring semantic similarity between words using multiple information sources[J]. IEEE Trans on Knowledge and Data Engineering, 2003, 15(4): 871-882.