第 34 卷 第 7 期 计 算 机 工 程 2008 年 4 月
Vol.34 No.7 Computer Engineering April 2008
·网络与通信· 文章编号：1000—3428(2008)07—0101—03 文献标识码：A 中图分类号：TP39

基于 DHT 的 P2P 覆盖网络的研究

李普聪[1]，魏文红[1,2]
（1. 江西财经大学软件学院，南昌 330013；2. 华南理工大学计算机学院，广州 510640）

摘 要：针对 P2P 覆盖网络具有易于构建、管理灵活、可扩展性强等特点，该文定义一种支持分组的 Cayley 图的网络模型 CayNET，阐述一个 DHT 协议——CayNET DHT 和其拓扑结构，分析节点的动态加入和退出过程。实验证明了该协议的有效性。
关键词：CayNET 协议；分组；P2P 网络；Cayley 图

Research on P2P Overlay Network Based on DHT

LI Pu-cong[1], WEI Wen-hong[1,2]
(1. School of Software, Jiangxi University of Finance & Economics, Nanchang 330013;
2. Dept. of Computer, South China University of Technology, Guangzhou 510640)

【Abstract】 Peer-to-Peer overlay networks have various features such as robust wide area routing, efficient search, and selection of nearby peers. This paper introduces a new network model——CayNET that supports grouping, and defines a DHT protocol——CayNET DHT in the network model, puts forward its topology, describes progress of adding and exiting in CayNET DHT. Experimental results show that the protocol is effective.
【Key words】 CayNET protocol; grouping; Peer-to-Peer(P2P) network; Cayley graph

目前，P2P技术已经被广泛用于Internet环境下的文件共享中。这些系统能够提供基于文件名称的数据共享功能，却难以实现类似Web搜索引擎提供的信息检索功能。大规模网络环境下的内容定位问题，是现有P2P系统深度信息共享面临的首要挑战。现在研究的P2P结构大都是基于分布式哈希(Distributed Hash Table, DHT)技术的，基于DHT的系统具有良好的搜索性能，但在大规模的动态P2P环境下，系统维持结构的代价很高[1]。另外，基于DHT的P2P系统只支持精确的对象键值搜索，缺乏模糊搜索能力，难以有效地支持基于内容的定位。为了弥补结构化P2P网络的不足，笔者提出了一种支持分组的Cayley图的网络模型(Cayley Network, CayNET)，基于CayNET设计了一套在动态系统中应用的P2P DHT协议——CayNET DHT，该协议可以支持文件浏览以及基于兴趣的分组等模仿人类社会的行为，具有很高的可用性。

1 符号与词汇说明

假定 x,y 是由数字或者“*”所组成的字符串。在 x,y 上定义以下操作：

$|x|$：字符串 x 的长度。

$x[i]$：字符串 x 的第 i 个字符。

$lock(x,y,i)$：$|x|<i \vee |y|< i$；或 $x[i]=$“*” $\vee y[i]=$“*”；或 $x[i]=y[i]$。例如：如果 $x=$“210”而 $y=$“2*11”，则 $lock(x, y, 0)$=True，$lock(x, y, 1)$=True，$lock(x, y, 3)$=True，但是 $lock(x, y, 2)$=False。

$lockbut(x, y, i)$：$\forall j\in N\wedge j\neq I, lock(x, y, j)$。

$lockall(x, y)$：$\forall j \in N, lock(x, y, j)$。

$AP(x, i, r)$：$i\in Z_r$，由$x[i], x[i+r], x[i+2r], \cdots$，组成的$x$的子串。比如$AP$(“010*2”, 1, 2)= “1*”。

$APlock(x, y, i, r)$：定义为 $lockall(AP(x, i, r), AP(y, i, r)$，例如：若 x = “010*2”, y = “110121”, i = 1, r = 2，则 $AP(x, i, r)$=“1*”, $AP(y, i, r)$=“111”，于是得到 $APlock(x, y, i, r)$。把 $APlock(x, y, i, r)$称为 x 在 i 维锁定 y。

$APlockbut(x, y, i, r)$：$\forall j\in Z_r \wedge j\neq i, APlock(x, y, j, r)$。

$APlockall(x, y, r)$：$\forall j\in Z_r, APlock(x, y, j, r)$。

若无特别说明，假设顶点、节点、对等点的意义相同；另外，还用顶点标识符或者对等点标识符表示顶点或者对等点本身。

2 CayNET 的定义

在 CayNET 中，对于任意一个$(c, p, r)\in G$(G 为群)，其中，c 为组标识符；p 为组内标识符；r 为地区标识符；(c, p, r)为顶点的标识符。CayNET 所依赖的 Cayley 图是基于群 G 的，下面定义用于生成 Caley 图的集合 S，分 3 步来说明这个集合中各个元素的作用：

(1)令 $S=\varnothing$，为了引进组内的“群居现象”，令

$$S_p=\{(0, p\underbrace{00\cdots0}_{k-1}, 0)\mid p\in Z_{r_p}\setminus\{0\}\}$$

$$S=S\cup S_p$$

(2)增加不同组之间的连接，得

$$S_c=\{(c\underbrace{00\cdots0}_{k-1}, 0, 0)\mid c\in Z_{r_c}\setminus\{0\}\}$$

$$S=S\cup S_c$$

(3)把不同地区的相应顶点连接起来，即

$$S_r=\{(0, 0, r)\mid r\in Z_k\setminus\{0\}\quad S=S\cup S_r$$

基于上述讨论，给出 CayNET 的代数定义：

定义 1 令$S = S_p \cup S_c \cup S_r$，则CayNET是定义在$G$上的Cayley图$Cay(G,S)$。

定义 2 对于 Cayley 图 $Cay(A,B)$和 B 的子集 C，定义由 C 导出的边集，即

基金项目：江西财经大学课题基金资助项目
作者简介：李普聪(1978—)，男，讲师、硕士，主研方向：网格计算；魏文红，讲师、在职博士研究生
收稿日期：2007-06-18 **E-mail：**gzcong@163.com

$$LinkBy(C)=\{e\mid e=(a,ac),a\in A\wedge c\in C\}$$

以及由 C 导出的 v 的邻居集合，即

$$Nbr(C,v)=\{v'\mid v'=vc,c\in C\}$$

定义 3 CayNET的边集可以分为 3 种类型，分别是：相同地区，相同组内的连接$Link_p$ =$LinkBy(S_p)$；相同地区不同组之间的连接$Link_c$ = $LinkBy(S_c)$；不同地区之间的连接$Link_r$ = $LinkBy(S_r)$。对于CayNET中的顶点v，定义 3 种邻居：$Nbr_p(v)$=$Nbr(S_p, v)$, $Nbr_c(v)$ = $Nbr(S_c, v)$, $Nbr_r(v)$ = $Nbr(S_r, v)$。

图 1 给出了一个简单的CayNET，r_c=2,r_p= 2, k = 2，图中顶点内是组内标识符，它们用$Link_p$连接；每 4 个顶点形成一个组，内部是组标识符，不同组之间的顶点用$Link_c$连接；白色背景的顶点的地区标识符为 0，而灰色背景的地区标识符为 1，不同地区之间对应的顶点用$Link_r$连接。注意，图中仅画出部分$Link_c$和$Link_r$。

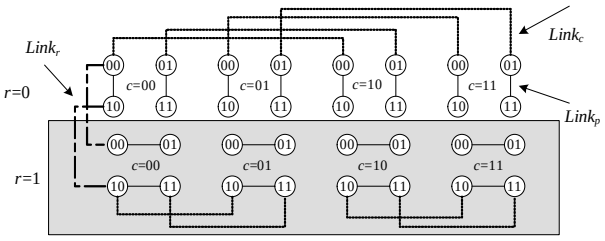

图 1　简单的 CayNET

3　CayNET DHT

CayNET 特指第 2 节提到的 CayNET 静态拓扑，而把基于 CayNET 静态拓扑的 P2P DHT 动态覆盖网络拓扑称为CayNET DHT。

3.1　标识符空间

在 CayNET DHT 中，每个对等点也是由一个称为对等点标识符的三元组(c, p, r)所唯一确定。下面的式子给出它们的定义：

$$c\in\{(c_0,c_1,\cdots,c_s)\mid -1\le s<l_c,c_i\in Z_2\cup\{*\}\}$$
$$p\in\{(p_0,p_1,\cdots,p_t)\mid -1\le t<\infty,p_i\in Z_2\}$$
$$r\in Z_k$$

其中，l_c是一个整型参数。

对等点标识符与CayNET中的顶点标识符存在 2 个区别：(1)CayNET DHT的c和p都是可变长的串；(2)这里的c和p不再是r_c和r_p进制的字符串，而是由“ * ”、“ 0 ”和“ 1 ”组成的字符串，$AP(c, i, k)$对应于CayNET中的$c[i]$，同理$AP(p, i, k)$对应于CayNET中的$p[i]$。比如，若CayNET中的c=“ 32 ”, k =2，则在CayNET DHT中的c表示为“ 1110 ”。由于c和p都是不定长的字符串，因此在CayNET DHT中r_c和r_p不再固定，随着网络规模的变化而变化。

3.2　分布式散列表

在CayNET DHT中数据文件名以及搜索关键词都会被映射到一个三元组(α, β, γ)，其中，α和β是定长的二进制字符串($|\alpha|=l_c,|\beta|>>|p|$)，而$\gamma\in Z_k$。标识符为($c$, p, r)的对等点负责所有满足$lockall(c,\alpha)$, $lockall(p,\beta)$, r=k的散列表表项。比如对等点(0*1, 11001, 3)负责键值为(0111, 1100101, 3)的表项，同理，它也负责(0010, 1100100, 3)。在不发生混淆的情况下，同样把α称为组标识符，把β称为组内标识符。把所有满足$lockall(c,\alpha)$的α组成的标识符空间称为该对等点所负责的组区域，而把所有满足$lockall(p,\beta)$的β组成的标识符空间称为该对等点所负责的对等点区域。事实上，需要把文件按照它的元信息、内容、名称等分成不同的组，不同的组拥有不同的组标识符，它们与对等点组标识符一一对应。至于如何把文件分组，则涉及信息分类和机器学习，不在此探讨。

3.3　CayNET DHT 拓扑

CayNET DHT拓扑是CayNET拓扑在动态网络的一次模拟。在CayNET DHT中，(c_1, p_1, r_1)与(c_2, p_2, r_2)相邻，当且仅当它们满足下面 3 个条件之一即可：

(1)$r_1=r_2\wedge APlockall(c_1,c_2,k)\wedge APlockbut(p_1,p_2,r_1,k)$，它们之间的连接对应于$Link_p$。

(2)$r_1=r_2\wedge APlockall(p_1,p_2,k)\wedge APlockbut(c_1,c_2,r_1,k)$，它们之间的连接对应于$Link_c$。

(3)$APlockall(c_1,c_2,k)\wedge APlockall(p_1,p_2,k)$，它们之间的连接对应于$Link_r$。

图 2 是一个含有 10 个对等点的CayNET DHT例子，k = 2, l_c = 4。

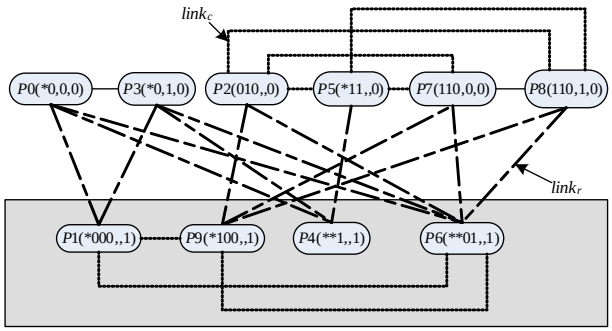

图 2　CayNET DHT 拓扑例子

3.4　加入 CayNET DHT

在节点加入CayNET DHT的过程中，一个新的对等点$P1$的加入包括“ 寻找网络位置 ”、“ 更新标识符 ”、“ 更新路由表 ”、“ 更新分布式散列表 ”4 个步骤，寻找网络位置是为了确定新的对等点在现有CayNET DHT中的位置，它首先使用既定的规则生成一个预期的标识符(expected$C$, expectdP, $r1$)，但此标识符并不是它最后所使用的标识符，然后寻找现有网络中负责该标识符的对等点$P3$=(c_3, p_3, r_3)。“ 更新标识符 ”最终生成$P1$ 和修改$P3$ 标识符。标识符的生成包括两种类型：组分割和组内分割。算法首先检查$P1$ 和$P3$ 的expected$C$，也就是期望的组标识符是否完全相同，若不相同则进行组分割，把$P3$ 原来负责的组区域平分为两部分，其中一个由$P3$，另一个由$P1$ 负责；若相同则进行组内分割，把$P3$ 原来负责的对等点区域平分为两部分，分由$P3$ 和$P1$ 负责。区域分割算法采用的是组分割优先，这是为了尽快地使对等点只负责它所期望负责的组，而不是由许多组组成的组区域。加入CayNET DHT的最后两步是更新路由表和更新分布式散列表。更新路由表是为了形成新的CayNET DHT拓扑。而更新分布式散列表是为了使那些不适应$P3$ 的新的标识符的散列表表项迁移到新的对等点$P1$ 上。由于$P1$ 和新的$P3$ 所负责的组区域和对等点区域都分别是原来$P3$ 所负责的组区域和对等点区域的子集，因此$P1$ 和新的$P3$ 的邻居集以及散列表都分别是原有$P3$ 的邻居集和散列表的子集。于是，仅需要循环检查$P3$ 原来的路由表和散列表即可构造$P1$ 和$P3$ 的新的路由表和散列表。最后$P1$ 和$P3$ 仍有机会相邻，若它们相邻则还需要把对方放到自身的路由表。CayNET DHT网络的构建过程如图 3 所示，其中，k = 2，l_c = 4。Time 0 初始状态，网络初始状态必须至少部署k个对等点，它们具有不同的地区标

识符，分布在不同的地区。Time 1：*P*2= (0101, 11010001, 0) 加入后；Time 2：*P*3 = (0000, 10011000, 0) 加入后；Time 3：*P*4 = (0011, 00000100, 1) 加入后；Time 4：P5 = (0111, 00010101, 0) 加入后；Time 5：*P*6 = (0001, 101011 11, 1) 加入后。

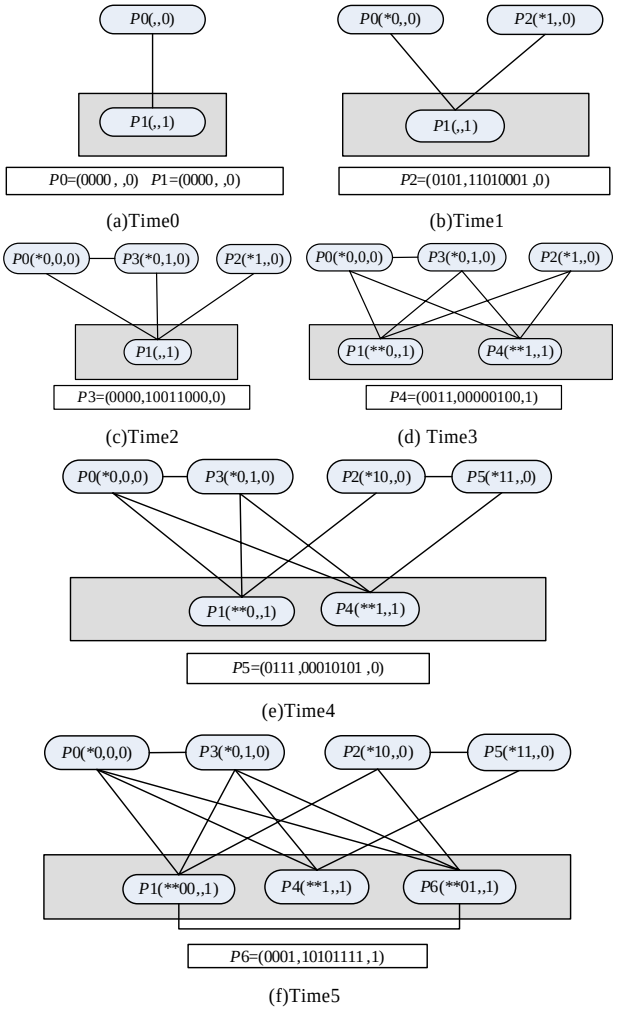

图 3　CayNET DHT 网络构建过程

3.5　退出 CayNET DHT

如果对等点离开CayNET DHT前，完成所有的清理工作，则称该次离开为正常离开，否则称为非正常离开。非正常离开比正常离开多一个失败检测阶段：失败检测策略包括向邻居定期地发送Hello和异步检测机制[2]。检测到错误后，该节点帮助非正常离开的节点完成清理工作。

正常离开CayNET DHT的算法类似于文献[2-3]提出的算法。若对等点 *P*1 需要离开网络，首先联系伙伴对等点 *P*3，若 *P*3 没有被组分割或组内分割(即没有新的对等点加入 *P*3 所负责的区域)，则 *P*1 和 *P*3 执行加入算法的逆过程：*P*1 把分布式散列表迁移到 *P*3、构建的网络拓扑，*P*3 生成新的对等点标识符。当然 *P*3 也可能在 *P*1 离开前被分割了，此时将在 *P*3 负责的区域中使用深度优先算法搜索一对最小的对等点 *P*4 和 *P*5，并且用它们其中之一，如：*P*4 代替 *P*1 的位置，然后 *P*5 合并它自己和 *P*4 所负责的区域。与文献[5]不同的是，若 *P*1 或 *P*3 加入网络时执行了组分割或 *P*3 的任何一个子区域被组分割，则 *P*4 和 *P*5 的组标识符与 *P*1 的组标识符将会不一致。此时，仍然让 *P*4 负责 *P*1 的区域，并且当有对等点需要成为 *P*4 的伙伴对等点时，*P*4 把它的路由表和分布式散列表的所有项迁到该新的对等点，而它自己再执行一次加入操作。注意，虽然此操作比较繁琐，但出现这种情况的概率是很低的。

4　结束语

本文设计了一种的支持分组的Cayley图的网络模型——CayNET，该网络模型在某种程度上模拟人类的社交网络，非常适合 P2P 网络。另外在 CayNET 模型中，本文又提出了一个新的 P2P 网络拓扑——CayNET DHT，它是以 DHT 为基础的 P2P 覆盖网络，提供了结构化的分组方式，许多特性优于其他流行协议。在基于分组结构优化网络使用率的基础上，可以在文件共享型的 P2P 网络中添加实用的浏览功能。

参考文献

[1] Sripanidkulchai K, Maggs B, Zhang H. Effcient Content Location Using Interest-based Locality in Peer-to-Peer Systems[C]//Proc. of IEEE InfoCay'03. San Francisco, USA: [s. n.], 2003: 2166-2176.

[2] Kumar S, Merugu J. Ulysses: A Robust, Low-diameter, Low-latency Peer-to-Peer Network[J]. European Transaction on Telecommunications, 2004, 15(6): 571-587.

[3] Xu Jun. On the Fundamental Tradeoffs Between Routing Table Size and Network Diameter in Peer-to-Peer Networks[C]//Proc. of IEEE InfoCay'03. San Francisco, USA: [s. n.], 2003: 2177-2187.

[4] Watts D J, Strogatz S H. Collective Dynamics of Small-world Networks[J]. Nature, 1998, 393(6684): 440-442.

[5] Aberer K, Alima L O. The Essence of P2P: A Reference Architecture for Overlay Networks[C]//Proc. of the 5th IEEE International Conference on Peer-to-Peer Computing. Galway, Ireland: [s. n.], 2005.

（上接第 86 页）

参考文献

[1] Gustavson F G. High-performance Linear Algebra Algorithms Using New Generalized Data Structures for Matrices[J]. IBM J. RES. & DEV., 2003, 47(1).

[2] Goto K. Anatomy of High-Performance Matrix Multiplication[J]. ACM Transactions on Mathematical Software, 2007, 34(3): 1-24.

[3] 蒋孟奇，张云泉，宋 刚，等. 综合递归分块技术及其在数值计算中的应用[C]//全国高性能计算学术年会会议论文集. 中国，北京: [出版社不祥], 2006.

[4] Robert A. van de Geijn Enrique S. Quintana-Ort´ I. The Science of Programming Matrix Computations[M]. [S. 1.]: MIT Press, 2006.

[5] Herrero J R, Navarro J J. Building Libraries for Small Matrix Kernels[EB/OL]. (2007-02-20). www.citeseer.ist.psu.edu/703531.html.

[6] 张云泉，孙家昶，迟学斌，等. 数值计算程序的存储复杂性分析[J]. 计算机学报, 2000, 23(4): 363-373.

[7] 张云泉. 面向高性能数值计算的并行计算模型 DRAM(h)[J]. 计算机学报, 2003, 26(12): 1660-1670.