

文章编号:1001-9081(2006)11-2550-04

基于超节点的结构化 P2P 路由算法的研究

王红玉, 董健全, 王孟孟, 钱小军
(上海大学 计算机工程与科学学院, 上海 200072)
(syswhy@tom.com)

摘要:在对经典的结构化 P2P 路由算法研究的基础上,提出了 BSNCCC (Based Super Node Cube-Connect-Cycle) 路由算法。该算法节点维护的信息为 $O(1)$, 查询步长为 $O(d)$ (节点个数 $N = d * 2^d$), 在充分考虑节点性能差异性的基础上,将性能好的节点作为路由过程中的主节点。模拟试验结果表明,在动态变化的 P2P 网络中,BSNCCC 路由算法的效率优于 Cycloid 等算法。

关键词:结构化 P2P; 路由算法; 超立方体

中图分类号: TP393.07 **文献标识码:** A

Research of routing algorithms in structured P2P network based on super nodes

WANG Hong-yu, DONG Jian-quan, WANG Meng-meng, QIAN Xiao-jun
(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

Abstract: Based on the research of classical routing algorithms in structured P2P network, a new routing algorithm, named Based Super Node Cube-Connect-Cycle (BSNCCC) was proposed. Based on $O(d)$ ($N = d * 2^d$) hops per lookup request by using $O(1)$ neighbors per node, the algorithm took advantage of the difference of nodes' capabilities in the network. The algorithm guaranteed that the nodes with the best capabilities served as the primary nodes. The simulation results show that BSNCCC routing algorithm has higher location efficiency than Cycloid in large scale and dynamic P2P networks that have frequent nodes arrival and departure.

Key words: structured P2P; routing algorithm; hyper-cube

0 引言

在 P2P 网络中节点共享所拥有的部分资源,节点之间可以直接访问共享资源而无需经过中间实体,突破了传统的 C/S (Customer/Server) 模式中用户节点被动接受资源的状况。这种资源边缘化的形式充分地利用网络中用户节点的资源,且克服了 C/S 模式容易遇到单点(服务器)失效和带宽瓶颈的缺陷。近年来,P2P 技术的成熟产品在 Internet 上得到广泛使用,如文件下载和在线通讯等。

P2P 网络路由算法根据其结构特性可以分为非结构化 P2P 路由算法和结构化 P2P 路由算法。在非结构化 P2P 如 Gnutella^[1] 中,节点之间连接任意,数据信息放置与对等网络拓扑结构无关。非结构化 P2P 路由算法采用基于完全随机图的洪泛发现或随机转发机制。这类路由算法容易实现,当前商业 P2P 产品都是非结构化的,但是采用洪泛、随机漫步或有选择转发算法,使得非结构化 P2P 路由直径不可控,可扩展性较差。结构化 P2P 路由算法拓扑结构有严格的控制,基于 DHT 数据存放的位置和查询算法都有很精确的定义或描述。结构化 P2P 网络中路由算法有全局搜索和信息冗余小的优点。在规模为 N 个节点的 P2P 网络中,通常每个节点维护的相关节点信息为 $O(\log N)$,每一次查询需要 $O(\log N)$ 步。

本文在对当前结构化 P2P 路由算法 Cycloid^[2,3] 等研究的基础上,提出了 BSNCCC (Based Super Node Cube-Connect-Cycle) 路由算法,此算法中每个节点维护信息为 $O(1)$, 查询步长为 $O(d)$ ($N = d * 2^d$)。充分考虑节点性能的差异性,选择性能好的

节点在路由过程中担任主节点,主节点不仅能及时处理繁忙的路由任务;而且由于其状态稳定使网络中信息量也大量减少。最后对该路由的具体算法和效果进行了分析和总结。

1 技术背景

1.1 经典结构化 P2P 路由算法性能比较

当前经典的结构化 P2P 路由算法有 Chord^[4], Pastry^[5], CAN^[6], Viceroy^[7], Koorde^[8] 和 Cycloid。在规模为 N 个节点的网络中它们的性能比较如表 1 所示(表中 d 为拓扑结构图的维数, $N = d * 2^d$)。

表 1 典型结构化 P2P 路由算法比较

系统名	拓扑结构	路由表大小	路由复杂度
Chord	cycle	$O(\log N)$	$O(\log N)$
Pastry	Plaxton trees	$O(\log N)$	$O(\log N)$
CAN	d-dim. torus	$O(d)$	$O(d \cdot N^{1/d})$
Viceroy	butterfly	$O(1)$	$O(\log N)$
Koorde/D2B	De Bruijn	$O(1)$	$O(\log N)$
Cycloid	Cube conn. cycle	$O(1)$	$O(d)$

Chord 基于一维的环形空间,每个节点维护的路由表大小为 $O(\log N)$, 路由的复杂度为 $O(\log N)$ 。Pastry 路由算法中每个节点维护的路由表大小及系统的路由步长和 Chord 数量级相同,其拓扑结构为 Plaxton 树。CAN 路由算法的拓扑结构为 d 维环形空间,CAN 和前两种路由算法相比,路由表数量

收稿日期:2006-05-10 基金项目:上海市科委发展基金资助项目(7A05722)

作者简介:王红玉(1976-),女,上海人,硕士研究生,主要研究方向:网络、数据库;董健全(1952-),女,上海人,副教授,主要研究方向:数据库、人工智能、网络;王孟孟(1976-),男,上海人,硕士研究生,主要研究方向:网络安全、数据库;钱小军(1971-),男,江苏泰兴人,高级工程师,主要研究方向:网络安全。

级为 $O(d)$, 比它们小, 但路由复杂度为 $O(d \cdot N^{1/d})$, 比它们大。Viceroy 和 Koorde 算法的路由表大小和路由复杂度在同一个数量级, 分别为 $O(1)$ 和 $O(\log N)$, 它们拓扑结构不同, 这两种算法较前三种算法在性能上有较大的改进, 路由复杂度在同一个数量级 $O(\log N)$, 但是路由表大小是常量级的, 和网络的规模无关, 这使得网络中的通信量大大减少。Cycloid 和前面的几种算法相比, 路由复杂度最小, 而路由表大小也是维护一个常数, 其性能是比较优异的。

1.2 Cycloid 的路由算法^[2,3]

Cycloid 的拓扑结构是带环超立方体 (cube-connect-cycle)。即一个 d 维的超立方体中, 每个顶点由 d 个点组成的环替代, 如图 1 所示。

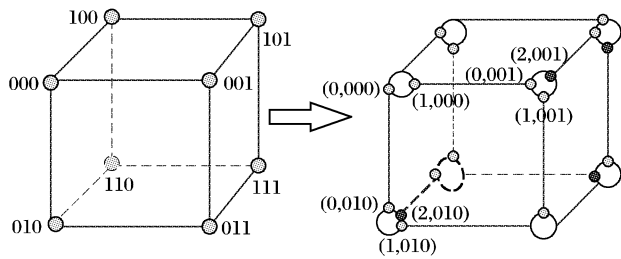


图 1 立方体到 Cycloid 拓扑结构

在 Cycloid 路由算法中, 节点和资源通过 DHT (Distribute hash Table) 得到范围在 $0 \sim d \cdot 2^d$ 的 ID 值。资源信息映射到 ID 值和自己 ID 相等或最接近的节点上。节点和资源在网络中的位置由环内序号 (k) 和超立方体序号 ($C_{d-1}C_{d-2} \cdots C_0$) 组成的序号对 ($k, C_{d-1}C_{d-2} \cdots C_0$) 确定。超立方体序号是 ID 值对维数 d 取整 ($C_{d-1}C_{d-2} \cdots C_0$ 的取值范围为 $[0, 2^d - 1]$), 环内序号是 ID 值对维数 d 取余 (k 的取值范围为 $[0, d - 1]$)。网络中的每个节点需维护 7 个相关节点信息如表 2 所示, 其中 3 个路由表邻居信息, 4 个叶节点信息, 以此实现网络的全局搜索。

每个节点的路由表信息初始化方法是: 首先将网络中超立方体序号相同的节点按环内序号从小到大排成一个小环, 所有小环按超立方体序号从小到大排成一个大环。序号为 ($k, C_{d-1}C_{d-2} \cdots C_0$) ($k \neq 0$) 节点路由表中维护的超立方体邻居为 ($k - 1, C_{d-1}C_{d-2} \cdots C_k \times \times \times$) \times 表示任意 0, 1 值。两个环上邻居序号对分别为 ($k - 1, C_{d-1}C_{d-2} \cdots C_k a_{k-1} \cdots a_0$) 和 ($k - 1, C_{d-1}C_{d-2} \cdots C_k b_{k-1} \cdots b_0$), 即超立方体序号大于 (或等于) 且最接近于本节点超立方体序号的节点和超立方体序号小于 (或等于) 且最接近于本节点超立方体序号的节点, 同时它们与本节点的最高不同位要低于 k 位的节点。环内序号为 0 的节点路由表为空, 超立方体序号为 0 的节点没有环上小邻居, 超立方体序号为 2^{d-1} 的节点没有环上大邻居。两个环内叶节点指的是与本节点处于同一小环的前驱和后继节点。两个超立方体叶节点指的是大环上本节点所在小环的前驱环和后继环上标号最大的节点 (称之为环内主节点)。

表 2 Cycloid 节点路由信息表

节点标号(3,1-0-011)	
路由表	
超立方体邻居(2,1-1- $\times \times \times$)	
环上大邻居(2,1-0-100)	
环上小邻居(2,1-0-010)	
小环叶节点集合(左小, 右大)	
(1,1-0-011)	(4,1-0-011)
大环叶节点集合(左小, 右大)	
(3,1-0-010)	(4,1-0-100)

假设一个维数为 5 的 Cycloid 系统中, 某一节点的序号对为 (3, 10011), 则它维护的路由信息表如表 2 所示。

Cycloid 的路由算法分三个阶段: 上升阶段、下降阶段及遍历环阶段, 并利用了弹性的互联模式。假设当前节点为 ($k, C_{d-1}C_{d-2} \cdots C_0$), 目标节点为 ($j, b_{d-1}b_{d-2} \cdots b_0$), MSDB (Most Significant Different Bit) 为当前节点与目标节点的超立方体序号最高不同位的位置, 则:

上升阶段 如果 $k < MSDB$, 则将查询请求转发给大环叶节点集合中离目标节点近的节点, 重复此过程, 直至找到环内序号 $k > MSDB$ 。

下降阶段 如果 $k \geq MSDB$, 若 $k = MSDB$, 则查询请求转发给超立方体邻居, 否则转发给环上邻居或小环叶节点集合中的节点, 取决于哪个节点更靠近目标节点。

遍历环阶段 如果目标节点已在叶节点集合中, 查询请求转发给离目标节点最近的节点上, 直到到达目标节点。

从 Cycloid 路由算法中可以看出, 系统中环内的节点承担的任务并不相等, 上升阶段通过主节点转发以期达到 $k > MSDB$, 每个小环的主节点较其他节点繁忙; 且主节点状态的改变 (上下线) 给网络带来的通讯量较普通节点多, 导致大环上此主节点所在的小环的前后的两个小环上的节点都要改变大环叶节点的设置。而在网络中节点的性能差异性很大, 在 Cycloid 中主节点是由节点 ID 随机决定, 没有考虑节点的性能差异。

2 BSNCCC 路由算法

2.1 BSNCCC 路由算法的基本思想

如果让性能好的节点担任主节点, 主节点的处理速度快、带宽大, 能够及时处理繁忙的路由任务; 主节点在线时间长, 稳定性好, 网络中因主节点状态改变带来的信息量就大大减少。

在本路由算法中节点性能主要考虑运算能力、带宽和在线时间三个方面的指标, 性能公式表示为: $P_{id} = \alpha C + \beta B + \gamma T$, 其中 α, β, γ 是常量系数 (满足 $\alpha + \beta + \gamma = 1$), 可以根据实际需要确定。C 表示运算能力, B 表示带宽, T 表示在线时间。在本文的模拟实验中, 侧重考虑节点的运算能力和带宽, 故选取 $\alpha = \beta = 0.4, \gamma = 0.2$ 。

BSNCCC 路由算法基本思想是: 选择环内性能最好的节点作为主节点; 环内序号动态确定, 主节点序号 $d - 1$, 普通节点的序号由主节点根据环内的负载情况分配; 小环上只要有节点在线, 环内序号为 $d - 1$ 的主节点就存在, 若某个节点是环内第一个上线节点, 则不考虑此节点的性能, 其环内序号为 $d - 1$, 这样减小了上升阶段的转发步长, 系统性能可以得到明显提高; 主节点维护环内所有在线节点信息, 环内定位阶段可以由主节点经过一个步长准确定位。

2.2 确定节点和资源在网络中的位置

BSNCCC 的拓扑结构亦为带环超立方体, 节点分布如图 2 所示。

节点序号对 ($k, C_{d-1}C_{d-2} \cdots C_0$) 确定节点在网络中的位置, 节点序号对确定的步骤如下: 第一步, 节点和资源信息根据 DHT 生成唯一的 ID 值。第二步, 超立方体序号 $C_{d-1}C_{d-2} \cdots C_0$ 是节点的 ID 除以 d 的商。第三步, 确定节点的环内序号。先判断网络中以 $C_{d-1}C_{d-2} \cdots C_0$ 为超立方体序号的小环是否存在, 如果此小环不存在, 则节点是小环内的第一个节点, 它就为这个环内的主节点, 它的环内序号值为 $d - 1$ 。如果节点的超立方体序号所代表的小环已存在, 则先比较节点和环内主节点的性能, 若节点性能比环内主节点的性能差, 环内序号由环内主节点根据

环内节点分布及负载均衡情况分配一个环内序号给节点 s , 使在线节点尽量在小环内均匀分布; 若节点性能比环内主节点的性能好, 则此节点取代环内主节点, 被取代的主节点环内序号由环内新主节点根据环内节点负载均衡情况分配一个环内序号 s 给节点。流程如图 3 所示。

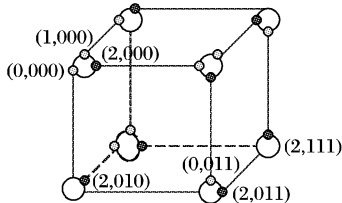


图2 BSNCCC 拓扑结构

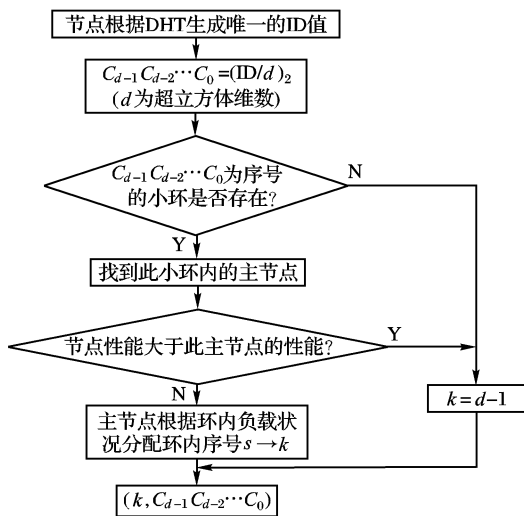


图3 节点网络中序号确定流程

资源信息的超立方体序号是 ID 值对维数 d 取整 ($a_{d-1}a_{d-2}...a_0$ 的取值范围为 $[0, 2^d - 1]$), 环内序号是 ID 值对维数 d 取余 (k 的取值范围为 $[0, d - 1]$)。资源信息在网络中放置在序号对 $(k, C_{d-1}C_{d-2}...C_0)$ 值和自己相等或最接近的节点上, 定位时, 首先查找立方体标号与 $a_{d-1}a_{d-2}...a_0$ 最接近的小环, 然后在此小环上查找其 ID 值和自己最接近的节点。

2.3 节点路由表的维护

表3 普通节点路由信息表

节点序号(3, 1-0-011)
路由表
超立方体邻居(2, 1-1-×××)
环上大邻居(2, 1-0-100)
环上小邻居(2, 1-0-010)
小环主节点
(4, 1-0-011)
大环叶节点集合(左小, 右大)
(4, 1-0-010) (4, 1-0-100)

表4 主节点路由信息表

主节点序号(4, 1-0011)
路由表
超立方体邻居(3, 0-××××)
环上大邻居(3, 1-0100)
环上小邻居(3, 1-0001)
在线环内节点
Node LeafNode[5]
大环叶节点集合(左小, 右大)
(4, 1-0010) (4, 1-0100)

BSNCCC 路由算法中主节点和普通节点维护的路由表内容是不相同的。普通节点维护超立方体邻居、环上的大小邻居、环叶节点及主节点 6 个节点的信息, 如表 3 所示。主节点维护超立方体邻居、环上的大小邻居、大环叶节点及环内在线普通节点的信息, 如表 4 所示, 其中环内节点信息根据节点的在线情况而动态变化。在线环内节点组从左到右表示环内序号 0~5 的节点的信息, 如果某节点在线, 则其所属的小环主节点维护的在线环内节点数组中, 下标等于某节点环内序号的元素值为该节点的信息。主节点和普通节点的路由表中邻

居节点和叶节点的初始化和 Cycloid 的方法相同。

由表 3、表 4 可见, 普通节点维护的 2 个大环叶节点信息, 主节点已经维护, 普通节点可以通过主节点得到, 即普通节点维护 4 个节点信息就可满足路由要求, 可以使网络中的信息量得以减少。本算法中普通节点维护了 2 个大环叶节点信息, 是为了减轻主节点的负载。

2.4 BSNCCC 路由算法步骤

在 BSNCCC 路由算法中, 节点收到查询任务后, 首先比较待查询资源的立方体序号和该节点的立方体序号的最高不同位 $MSDB$, 接着比较 $MSDB$ 和该节点的环内序号的大小, 根据比较结果分别进入三个不同的阶段:

上升阶段 如果 $k < MSDB$, 则将查询请求转发给大环叶节点集合中离目标节点近的节点, 经过一个步长即可达到环内序号 $k \geq MSDB$ 的节点。

下降阶段 如果 $k \geq MSDB$, 当 $k = MSDB$, 则查询请求转发给超立方体邻居, 如果超立方体邻居不存在, 则转发到离目标节点最近的大环外叶节点; 当 $k > MSDB$, 转发给更靠近目标节点的某个环上邻居或大环叶节点上。

环内定位阶段 如果目标节点的超立方体序号和叶节点集合中的节点的或本节点所在内环的超立方体序号相同, 但环内序号不同, 则查询请求转发给超立方体序号相同的主节点, 主节点从所记载的在线普通节点中找到满足要求的节点, 此阶段只需经过一个步长即可完成。BSNCCC 路由算法流程如图 4 所示。

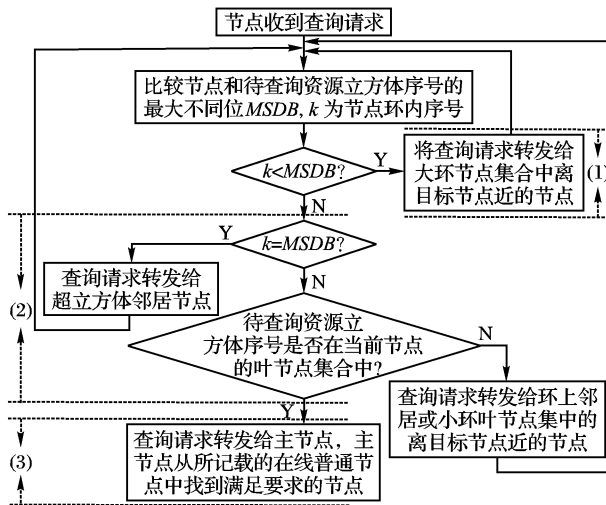


图4 资源查询流程

2.5 BSNCCC 路由算法示例

图 5 是节点 (2, 11000) 到目标节点 (3, 00111) 的路由示例。节点 (2, 11000) 和目标节点 (3, 00111) 的 $MSDB$ 等于 4。(2, 11000) 的环内序号 $k = 2$ 。即 $k < MSDB$, 所以进入上升阶段, 在 (2, 11000) 有路由表中的外叶节点集合中选择一个离目标节点近的节点, 外叶节点 (4, 11000) 被选中, 查询任务转发到节点 (4, 11000) 处理。节点 (4, 11000) 和目标节点 (3, 00111) 的 $MSDB$ 等于 4。即 $k = MSDB$, 进入下降阶段, 且节点 (4, 11000) 的超立方体邻居存在, 查询任务转发到其超立方体邻居 (3, 00110) 处理。节点 (3, 00110) 和目标节点 (3, 00111) 的 $MSDB$ 是 0, $k > MSDB$ 且路由表中有外叶节点 (4, 00111) 的立方体序号和目标节点的立方体序号相同, 查询任务转发到节点 (4, 00111) 处理, 进入环内定位阶段。(4, 00111) 是环内主节点, 到在线环内节点数组中查看下标为 3 的数组元素是否为空, 不为空, 即查找到了目标节点。为便于

表达,图 5 中主节点路由表中的在线环内节点数组中的元素



图 5 BSNCCC 路由算法示例

2.6 模拟实验

本模拟实验程序运用 Java 语言编写,在带环超立方体维数为 $d = 3 \sim 8$ (节点个数为 $24 \sim 2024$) 中,模拟了 BSNCCC 和 Cycloid 两种路由算法,使每个在线节点处理 $n/4$ 个查询请求的任务,模拟实验结果如图 6 所示,图中分别显示了 BSNCCC 和 Cycloid 算法的平均查找步长。

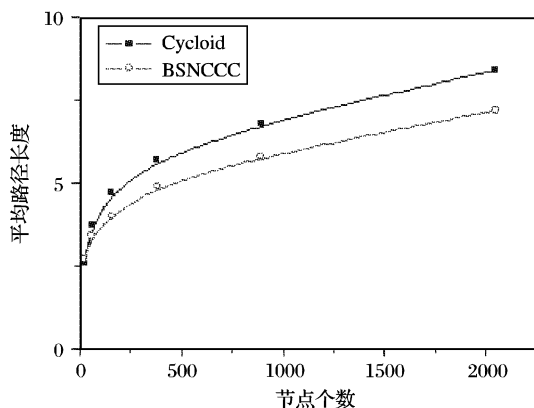


图 6 两种路由算法模拟实验结果

在相同的网络规模下,路由算法查询的步长越短,查询的效率就越高。从图 6 中可知,当网络节点规模较小时,两种算法的步长没有明显差别,如带环立方体的维数 $d = 3$ 时,Cycloid 的路由平均步长为 2.6,BSNCCC 的路由平均步长为 2.7;当 $d = 4$ 时,Cycloid 的路由平均步长为 3.7,BSNCCC 的路由平均步长为 3.4;当 $d = 6$ 时,Cycloid 的路由平均步长为 5.7,BSNCCC 的路由平均步长为 4.9;当 $d = 8$ 时,Cycloid 的路由平均步长为 8.4,BSNCCC 的路由平均步长为 7.2。这表明随着网络规模的增大,BSNCCC 比 Cycloid 路由算法查询效率更优。

在节点的负载均衡问题上,两种算法都是将资源信息映

的值简单表示为 0 或 1,0 表示不在线,1 表示节点在线。

射到 ID 值与自己相等或最接近的节点上,但是 BSNCCC 的节点在环内序号是上线时由主节点根据节点分布情况动态分配的,主节点可以根据节点负载情况实时分配上线节点的环内序号,Cycloid 中节点的环内序号是通过 DHT 生成的 ID 值确定,负载均衡由上线节点序号的分布情况决定,不能根据实际情况调整。显然,BSNCCC 节点负载比 Cycloid 更均衡。

参考文献:

- [1] <http://www.Gnutella.com> [EB/OL], 2006.
- [2] SHEN H, XU C, CHEN G. Cycloid: A constant degree and lookup-efficient p2p overlay network [A]. International parallel and Distributed Processing Symposium (IPDPS2004) [C]. Santa Fe, New Mexico, 2004.
- [3] 陈贵海, 须成忠, 沈海英, 等. 一种新的常数度数的 P2P 覆盖网络 [J]. 计算机学报, 2005, 28(7): 1084 - 1095.
- [4] STOICA I, MORRIS R, KARGER D, et al. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications [A]. Proceedings of the ACM SIGCOMM [C]. San Diego, CA, USA, 2001.
- [5] ROWSTRON A, DRUSCHEL P. Pastry: Scalable, Distributed Object Location and Routing for Large scale Peer-to-Peer Systems [EB/OL]. <http://research.microsoft.com/ant/past/pastry.pdf>, 2001.
- [6] RATNASAMY S, FRANCIS P, HANDLEY M, et al. A scalable content-addressable network [A]. Proceedings of ACM SIGCOMM [C], 2001. 329 - 350.
- [7] MALKHI D, NAOR M, RATAJZAK D. Viceroy: A scalable and dynamic emulation of the butterfly [A]. 21st ACM Symposium on Principles of Distributed Computing [C]. Monterey, California, USA, 2002.
- [8] KLEIS M, KLEIS, LUA EK, ZHOU X. Hierarchical Peer-to-Peer Networks using Lightweight SuperPeer Topologies [A]. ISCC2005 [C], 2005.
- [9] CHEN G. P2P Overlay Networks of Constant Degree [A]. GCC2003, LNCS 3032 [C], 2004. 412 - 419.

(上接第 2549 页)

所示)。但用户的信干比随着它与基站距离的增加而减小(如图 6 所示)是不公平的。所以 λ_i 取固定值时功率控制效果明显不如 $\lambda_i = k \cdot h_i$ 时的效果。同时从图 1, 图 3, 图 5 可以看出,传统分布式功率控制算法的收敛速度(30 次迭代达到收敛)明显比本文所提出的算法(15 次迭代达到收敛)慢。

参考文献:

- [1] 吴伟陵. 移动通信中的关键技术 [M]. 北京: 北京邮电大学出版社, 2000.
- [2] FOSCHINI GJ, MILJANIC J. A Simple Distributed Autonomous Power Control Algorithm and its Convergence [J]. IEEE Transactions on Vehicular Technology, 1993, 42(4): 641 - 646.
- [3] GOODMAN DJ, MANDAYAM NB. Power control for wireless data [J]. IEEE Personal Communications Magazine, 2000, 7(4): 48 - 54.
- [4] SHAH V, MANDAYAM N, GOODMAN D. Power Control for Wire-

- less Data based on Utility and Pricing [A]. Proceedings of Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'98) [C]. 1998, 3. 1427 - 1432.
- [5] FUDENBERG D, TIROLE J. Game theory [M]. Cambridge: MIT Press, 1991.
- [6] NASH J. Equilibrium Points in n-Person Games [A]. Proceedings of the National Academy of Sciences [C], 1950. 48 - 49.
- [7] (美) MATHEWS JH, FINK KD. 数值方法 (Matlab 版) [M]. 第 3 版. 陈渝, 周璐, 钱方, 等译. 北京: 电子工业出版社, 2004.
- [8] YATES R. A framework for uplink power control in cellular radio systems [J]. IEEE Journal on Selected Areas in Communications, 1995, 13(7): 1341 - 1347.
- [9] LEUNG KK, SUNG CW, WONG WS, et al. Convergence theory for a general class of power control algorithms [A]. Proceedings of IEEE International Conference on Communications [C], 2001, 3. 811 - 815.