

文章编号:1001-9081(2007)04-0821-04

一种 P2P 系统节点聚类及信息检索算法

李章华,刘高嵩,刘柯萍

(中南大学 信息科学与工程学院,湖南 长沙 410075)

(lsjlyh@sohu.com)

摘要:提出了一种节点聚类及信息检索算法——NCSearch。NCSearch 利用 Hilbert 曲线的局部性特征保持能力,将有相似内容的节点聚类,形成若干个簇。搜索算法能快速定位到与查询最相关的簇,然后在簇内洪泛查找,返回的结果按相关度排序。模拟测试表明,NCSearch 稳定高效,相比传统算法在搜索效率方面有明显提高。

关键词:P2P;局部性;聚类;Hilbert 空间填充曲线;向量空间模型

中图分类号:TP393 **文献标识码:**A

A node clustering and information retrieval algorithm on P2P system

LI Zhang-hua, LIU Gao-song, LIU Ke-ping

(School of Information Science and Engineering, Central South University, Changsha Hunan 410075, China)

Abstract: An efficient node clustering and information retrieval algorithm named NCSearch was introduced. By using the Hilbert curve's good locality characteristic of preserving ability, NCSearch clustered the nodes that have similar content and form some clusters. The search algorithm can fast locate the cluster which is most relevant to query and then flood within the cluster, and all returned results were sorted by relevant degree. Simulation results show that, compared to the Gnutella algorithm, NCSearch improves search efficiency.

Key words: Peer-to-Peer (P2P); locality; clustering; Hilbert Space Fill Curve (HSFC); Vector Space Model (VSM)

0 引言

Peer-to-Peer (P2P) 系统以其自治性、可靠性、可伸缩性和低成本等优点,成为文件共享的理想平台。然而, P2P 系统也有其自身的一些缺点,比如参与节点的数目庞大,动态性强,缺乏中央的控制机制以及难以获得全局信息。这给文件共享带来了不小的困难,特别地,在 P2P 系统中查找相关文档(信息检索)的性能(查全率、查准率和响应时间)受到了很大的影响。在 P2P 系统中采用基于节点聚类的信息检索方法正受到日益广泛的关注^[1],节点簇的形成给 P2P 环境下进行信息检索带来了两个重要的好处:1)簇的成员能在簇中找到更有价值的文档;2)大多数的搜索工作可以被限定在簇中完成。这样,大大降低了网络开销。

本文提出了一种有效的节点聚类和信息检索方案——NCSearch。NCSearch 利用 Hilbert^[2,3]空间填充曲线的内容局部性特征保持能力,将具有相似内容的节点聚集成簇,搜索算法首先快速定位到与查询向量最相关的若干个簇,然后依次在这些簇内洪泛查找。模拟测试表明,NCSearch 在搜索效率上较 Gnutell^[4]有明显的改进。

1 相关工作

1.1 文档的特征向量表示

在经典的信息获取模型中,一个文档可以标识成一个具有字典维数的特征词向量(一个字典是一个文档集中所有唯一特征词列表)。设 P2P 网络系统中所有节点集合为 $P = \{p_i | 1 \leq i \leq N\}$,文档对象集合为 $A = \{a_j | 1 \leq j \leq M\}$,特征词集

合为 $T = \{t_k | 1 \leq k \leq L\}$, M 为文档对象的总数, L 为特征词的总数,每个节点 p_i 的文档对象集合记为 $A(p_i)$ 。根据向量空间模型(Vector Space Model, VSM),每个数据对象 a_j 可表示成一个特征词向量 $(d_{1j}, d_{2j}, \dots, d_{Lj})$, d_{kj} 为特征词 t_k 在 a_j 中的权值。计算特征词的权值的模式有很多种,其中最著名的就是 tf-idf,这种计算方法的缺陷在于它需要全局知识去计算特征词的权值,而获取全局知识在 P2P 环境中代价太大,因此使用一种“dampened”tf^[5]模式来计算单词的权值:首先构造临时文档向量 $a_j = (f_{1j}, f_{2j}, \dots, f_{Lj})$, f_{kj} 为特征词 t_k 在 a_j 中的词频(tf),然后令 $d_{kj} = 1 + \log f_{kj}$ 得到正式的文档向量。这种模式有两方面的优点:1)这种模式不要求任何全局信息;2)它产生高质量的文档簇^[5]。给定数据对象 $a_j = (d_{1j}, d_{2j}, \dots, d_{Lj})$ 和查询对象 $q = (d_1, d_2, \dots, d_L)$,两者之间的相关性度量定义为:

$$REL(a_j, q) = \frac{a_j \cdot q}{|a_j| \times |q|} = \frac{\sum_{k=1}^L d_{kj} d_k}{\sqrt{\sum_{k=1}^L d_{kj}^2} \sqrt{\sum_{k=1}^L d_k^2}} \quad (1)$$

如果相关性高于某个阈值,那么 a_j 和 q 被认为是相关的。

1.2 节点向量和簇向量

为了在搜索算法中计算查询与节点,查询与簇之间的相关度,进一步提出了节点向量和簇向量的概念。

节点向量是一个节点的内容摘要,给定节点 p_i ,其节点向量 $p_i = (d_1, d_2, \dots, d_L)$ 的计算步骤如下:1)对每个文档 $a_j \in A(p_i)$, $j = 1, 2, \dots, n$, $n = |A(p_i)|$,构造临时文档向量 $a_j = (f_{1j}, f_{2j}, \dots, f_{Lj})$, f_{kj} 为特征词 t_k 在 a_j 中的词频;2)对每个特

收稿日期:2006-10-11;修订日期:2006-12-15 基金项目:湖南省自然科学基金资助项目(03JJY3111)

作者简介:李章华(1977-),男,湖南冷水江人,硕士研究生,主要研究方向:P2P网络、信息检索;刘高嵩(1948-),男,湖南长沙人,副教授,主要研究方向:软件工程、计算机网络技术;刘柯萍(1982-),女,江西上饶人,硕士研究生,主要研究方向:P2P网络、信息检索。

征词 t_k , 计算 $f_k = \sum_{j=1}^n f_{kj}, k = 1, 2, \dots, L; 3)$ 对每个 f_k , 令 $d_k = 1 + \log f_k$ 得到节点向量。

簇向量是对整个簇的内容的摘要, 其计算方式与节点向量有所不同, 直接由该簇的所有节点向量求和计算得出。节点向量和簇向量与查询的相似度计算公式与式(1)相同。

1.3 Hilbert 空间填充曲线

空间填充曲线已被用于改善 GIS、CAD、分子生物和多媒体数据库等高维数据空间的搜索, 常用的主要有三种空间填充曲线(SFC): Z 序(Z-Order)、格雷码(Gray Code)和 Hilbert 空间填充曲线(HSFC), HSFC 具有最好的保持局部位置特性的能力^[5]。实际应用中, 我们只对有限粒度的 L 维空间感兴趣, 因而只需构造 L 维 HSFC 的 t 阶近似, 对于 $t \geq 1, L \geq 2$, 令 H_t^L 表示 L 维 HSFC 的 t 阶近似, 将其 $[0, 2^{L \times t} - 1]$ 区间映射到 L 维整数空间 $[0, 2^t - 1]^L$ 。在 L 维向量空间中相关度高的节点向量将在一维数值空间中获得比较接近的 Hilbert 数字, 因此局部性特征得以保持。

假定节点向量空间为 A, 若构造 A 的 HSFC, 则 L 维 HSFC 的 t 阶曲线将 A 中每一维的取值范围区间离散化, 划分为 2^t 段, 从而将整个特征空间 A 划分为 $2^{L \times t}$ 个超子立方体。每个节点向量对应一个超子立方体, 并在 L 维 HSFC 上有一个序号, 称之为节点的 HilbertID。t 用于控制划分的粒度, 其值越小, 则划分的超子立方体越大, 两个节点向量获得同一 Hilbert 数字的可能性就越大, 获得的向量接近性的粒度就越粗。

2 系统设计

2.1 节点聚类模型

我们的设计考虑了节点能力异构性, 按节点的能力将节点分成两类: 超级节点和普通节点, 为了方便描述, 首先给出一组定义。

定义 1 簇中心节点, 指初始化并维护簇的超级节点。

定义 2 簇内节点, 就是指某个簇中心节点的所有孩子节点。

定义 3 簇内节点距离。令两个簇内节点 p_i 和 p_j , 则它们之间的距离 $DISC(p_i, p_j) = |p_i \cdot HilbertID - p_j \cdot HilbertID|$ 。

定义 4 与簇中心节点距离(加入距离)。任意节点 p_i 和簇中心节点 S_j , 它们之间的距离 $DISC_{join}(p_i, S_j) = |p_i \cdot HilbertID - S_j \cdot LogicID|$, S_j 的 LogicID 是 S_j 在顶层 Chord^[6] 环中的 ID。LogicID 与 HilbertID 的 ID 空间的取值范围相同。

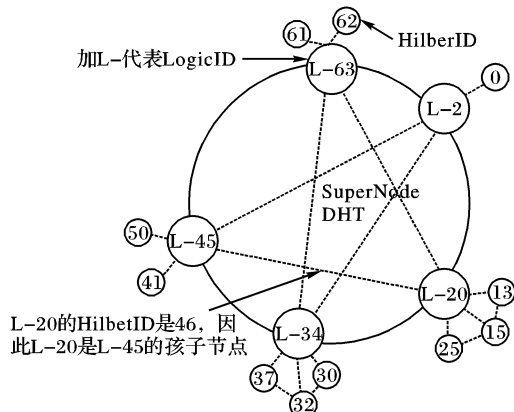


图1 节点聚类模型

整个模型分为两层结构, 如图 1 所示。顶层是超级节点

结构化层, 在超级节点间连接形成一个 Chord 环, 称之为 SuperNodeDHT, 底层的节点选择与自己的 $DISC_{join}$ 距离值最小的簇中心节点建立连接, 从而加入到簇中。每个超级节点有两个 ID: LogicID 和 HilbertID, LogicID 用于加入 Chord 环成为簇中心节点, HilbertID 用于加入 $DISC_{join}$ 值最近的簇成为簇内节点。

从图 1 可以看到, 整个网络形成若干个簇, 簇内每个节点最多与 K 个 DISC 距离最近的簇内节点建立连接(这 K 个节点由簇中心节点负责分配)。

2.2 节点加入、离开算法

普通节点 RP 的加入步骤: 1) 将其 HilbertID 作为 key 在 SuperNodeDHT 中路由查找到与自己具有最小 D_{join} 值的超级节点 SP; 2) 与 SP 建立连接, 并将节点向量上传给 SP, SP 给 RP 分配最多 K 个簇内距离最近的节点作为簇内邻居节点。

超级节点 SP 的加入步骤: 1) 调用 Chord 的加入算法加入到 SuperNodeDHT 中, 并初始化新簇; 2) 通知环中的前驱和后继超级节点, 前驱和后继超级节点各自计算其簇内节点与 SP 的 D_{join} 值, 若 D_{join} 小于簇内节点到自己的距离, 则将这些簇内节点转移至 SP 所维护的簇中; 3) SP 然后执行与普通节点相同的加入步骤。

如果是普通节点离开, 离开时只需通知超级节点; 如果是超级节点离开, 则还需将其维护的簇内节点转移至前驱或者后继节点。节点的加入、离开算法的伪代码如下所示。

簇内节点的连接根据 Hilbert 数字距离的远近建立, 实际上通过簇中心节点计算节点向量的相关度, 可以获得更为准确的语义邻居。但是这增加了簇中心节点的计算负荷, 尤其当更新周期较短时, 簇中心节点需要频繁计算节点向量的相关度, 导致负载过重。将这两种不同的连接方式称作 HLink 和 SLink。

算法 1 节点 n 加入算法

```

n.join(n') {
1: HilbertId = Normalize(n.Hilbertnum); //获取规范化 HilbertID
2: if n.capacity > Cap_threshold then // n 是超级节点
3:   LogicId = Normalize(Hash(IpAddress)); //获取规范化 LogicID
4:   n.joinChord(n', LogicId); //调用 Chord 的 join 函数加入到环中
5:   successor.supernodejoin_notify(n); //通知后继
6:   predecessor.supernodejoin_notify(n); //通知前驱
7: end if
8: s = n'.findsupernode(n, HilbertId); //找到 DISC_{join} 最近的簇中心节点
9: n.connectToSuperNode(s); //与簇中心节点建立连接
10: }
11: n.supernodejoin_notify(n') {
12: for i = 0 to childlist.size do
13:   child = childlist[i];
14: if DISC_{join}(child, n') < DISC_{join}(child, n) then //判断距离
15:   child.connectToSuperNode(n'); //与新的簇中心节点连接
16: end if
17: end for
18: }
19: n.connectToSuperNode(n') {
20: supernode = n';
21: supernode.addtochildlist(n);
22: uploadToSupermode(NodeVector); //上传节点向量给簇中心节点
23: }
24: n.addtochildlist(n') {
25: assignNeighbor(n')
//为 n' 分配 K 个簇内距离最近的簇内邻居;

```

```
26: childlist.add(n');
27: }
```

2.3 资源定位过程描述

1) 查询发起节点 P_0 首先将查询请求发送至其所属簇的中心节点 SP_0 ;

2) SP_0 计算查询向量与所有簇向量的相关性,返回与查询最相关的 m 个簇的序列 $C_1, C_2, C_3, \dots, C_m$ 给 P_0 ;

算法2 节点 n 离开算法

```
n.leave(){
1: if n 是普通节点 then
2:   notify(supermode);           //通知簇中心节点
3: else                             // n 是超级节点
4:   notify(supermode);           //通知簇中心节点
5: for i = 0 to childlist.size do
6:   childlist = childlist[i];
7: if DISCjoin(child, predecessor) < DISCjoin(child, successor) then
8:   将孩子节点转移至前驱节点簇;
9: else
10:  将孩子节点转移至后继节点簇;
11: end if
12: end for
13: end if
14: }
```

3) P_0 依次向这些目标簇发起查询,将查询发往簇 C_i 的中心节点 $SP_i, i = 1, 2, \dots, m$;

4) SP_i 在收到查询后,计算簇内所有节点向量与查询的相关度,选择与查询最相关的 S 个簇内节点转发请求;

5) 收到请求的簇内节点在本地进行查找,并返回与查询相关的文档给 P_0 ,然后转发请求给自己邻居表内的所有簇内邻居节点;

6) P_0 在收到簇 C_i 返回的所有搜索结果后,判断是否搜索结果数已达到最大结果数阈值 $MAX_RESPONSES$,如果达到则转7),否则重复步骤3)到6)的过程继续发送请求至下一个簇 C_{i+1} 继续查找;

7) 在收到所有搜索结果后, P_0 将结果按相关度进行排序得到最终结果。

为了控制簇内洪泛所引起的消息冗余,查询发起节点给查询分配一个唯一的标识符 GUID,在洪泛过程中,一旦节点接收到已经访问过该节点的带相同 GUID 的查询,则将其丢弃。

2.4 节点向量与簇向量的更新策略

节点的文档内容集合经常会发生改变,因此它的节点向量也会发生变化,节点需要周期性地计算其节点向量的 HilbertID 以及与簇中心节点的距离,如果发现与中心节点距离已超出某一阈值,则重新执行一次节点加入算法,加入到正确的簇中;否则,节点会将节点向量上传给簇中心节点,为了减少带宽的消耗,仅传送节点向量的“ultra delta”变化量。

簇内节点的节点向量变化,则簇向量也会发生变化,超级节点周期性地计算簇向量,并将簇向量的“ultra delta”变化量在 SuperNodeDHT 环中广播。广播采用文献[7]提出的有效 Chord 广播算法,该算法保证环内所有节点只收到消息一次,没有冗余信息产生,如环内超级节点数为 N ,则总共产生 $N - 1$ 条消息。

2.5 故障容忍与语义邻居维持

在 P2P 这样的高度动态环境中,为了保证在簇中心节点发生故障情况下,其簇内节点仍能找到相应的簇加入,采用了

一种“Lazy Repair”的方式。由于簇内节点 CP 要周期性向簇中心节点 SP 发送节点向量的变化量消息,因此可以在该消息上附加一个“Keep alive”消息,SP 必须回复一个“Keep alive”消息,并重新为 CP 分配 K 个簇内距离最近的邻居节点,将这些节点的 HilbertID 及 IP 地址附加在回复消息上返回给 CP,收到消息的 CP 会重新建立与新的 K 个邻居节点的连接;如果 CP 没有收到 SP 返回的“Keep alive”消息,则认为 SP 已失效,并重新执行一次节点加入算法,加入到正确的簇中。同样,SP 如果在周期 T 内未收到 CP 发来的消息,则它认为该节点已失效,因此会移去所维护的 CP 的节点向量及其他相关信息。通过这种方式,节点总能加入离自己最近的簇,并获得与自己簇内距离最近的 K 个邻居节点的连接。

3 实验仿真

3.1 实验设置

模拟实验在一台配置为 CPU P42.0GHz,内存 1GB,操作系统为 Windows 2000 的 PC 机上进行。模拟程序用 Java 语言编写,其中 NCSearch 模拟程序建立在开源模拟器 Open Chord 的基础上。为了验证本文提出方法的有效性,比较了 NCSearch 与 Gnutella(随机走步)的搜索性能,并且比较了 NCSearch 采用二种不同策略(HLink 和 SLink)时的搜索性能。

实验所采用的文档数据来自 Reuters^[8],Reuters 是广泛用于 IR 检索领域的文档测试集,其中包含了大量带有作者字段的新闻文档,从 Reuters 集随机选取了 1384 个作者的文档,从每个作者的文档中再抽取子集,这样共抽取了 24750 篇文档构造文档集,这些文档覆盖了经济、体育、战争等多个主题。实验用一个作者模拟一个节点,与该作者有关的文档存放在这个节点。对文档集中的每篇文档进行预处理:保留词干,去除文档中的停用词和高频词汇。然后从预处理过的文档集中选取 IDF 反转频率最高的 100 个词条作为特征词,用于构造文档向量。随机选择一个文档,并随机从该文档的特征词向量中抽取 3 个特征词来构造一个查询,通过这种方式共生成了有 50 个查询的查询集。为了便于实验,这里假定与一个查询相关的文档是那些包含查询中所有特征词的文档。根据 Gnutella Profile,节点按能力划分为 $1 \times, 10 \times, 100 \times, 1000 \times, 10000 \times$ 五个等级,各占节点总数的百分比分别为 20%, 45%, 30%, 4.9%, 0.1%,因此将 $1000 \times$ 设为超级节点能力阈值。实验中的部分参数设置见表 1。

表1 仿真实验参数说明和参数设定

参数名称	参数符号表示	值
节点数目	N	1384
超级节点能力阈值	$Cap_threshold$	$1000 \times$
文档对象数目	M	24750
特征词数目/向量维数	L	100
最大簇内邻居数目	K	8
超级节点转发查询数目	S	4
HSFC 近似阶数	t	5

实验用查全率和查询处理开销来评价 NCSearch 的性能,分别说明如下。

1) 查全率:用于量化搜索结果的质量,定义为搜索结果中相关文档的数量除以系统文档集中所有相关文档的数量。

$$recall = \frac{\sum_{retrieved} Answer}{\sum_{total} Answer} \quad (2)$$

2) 查询处理开销:是指分布式查询过程中路由和处理查询消息的节点数目百分比,为消耗系统资源的决定因素。低的查询处理开销增加了系统的扩展性。

3.2 实验结果及分析

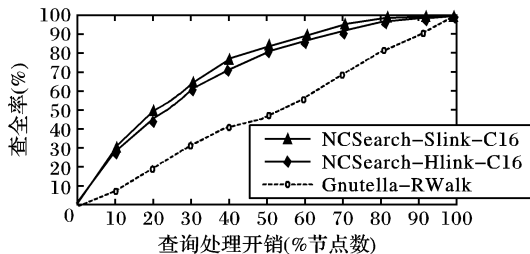
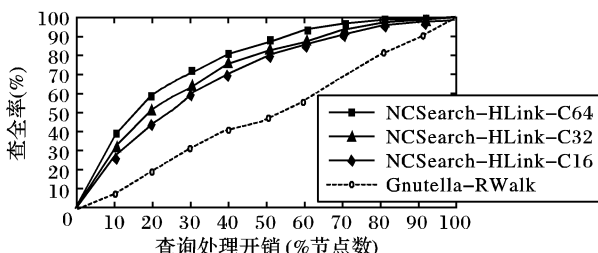
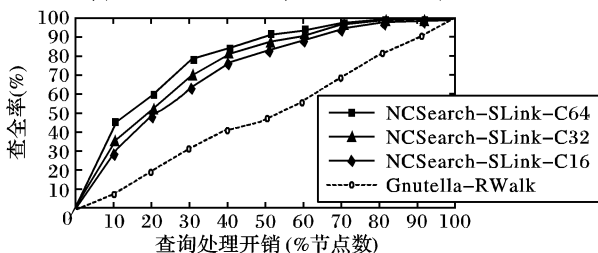


图2 查询性能比较(簇数目=16)



(a) NCSearch-HLink (簇数目=16,32,64)



(b) NCSearch-SLink (簇数目=16,32,64)

图3 HLink 和 SLink 在不同簇数目下的查全率

图2给出了在簇数目为16(即16个超级节点)的条件下查全率随查询处理开销的变化情况。从图2可以看出,在相同查询处理开销情况下,NCSearch的两种策略均明显优于Gnutella,在查询处理开销50%的情况下,SLink和HLink的查全率较Gnutella均高出35%以上,其中Slink甚至高出近40%。这是因为NCSearch快速定位至与查询最相关的簇,然后将搜索限定在簇范围内,从而在查询处理开销较小的情况下获得了较高的查全率。从图2中还可以看到,SLink的搜索性能略优于HLink,这是因为在SLink中,簇内节点的连接相

关度由簇中心节点进行向量相关度计算得出,因此比HLink更准确。

图3(a)和图3(b)分别给出了HLink和SLink在不同簇数目条件下的查全率随查询处理开销的变化情况。从图3中可以看出,簇数越高,获得的簇的粒度越细,因而搜索的效率越高。

4 结语

本文提出了一种有效的P2P系统节点聚类及信息检索算法,主要目标是提高P2P信息检索的效率。通过利用Hilbert空间填充曲线的局部位置特征保持能力快速地将节点聚类成若干个簇,查询能快速定位至与查询最相关的若干个簇,并在簇内进行洪泛搜索。模拟测试表明系统是有效的,但该算法的有效性依赖于节点向量及簇向量的及时更新,更新周期太长会影响搜索质量和效率,太短会加重系统节点的负担,因此更新周期的合理设定是下一步需要研究的问题。此外,引入有效的向量降维机制也是我们下一步研究的目标。

参考文献:

- [1] KLAMPANOS IA, JOSE JM. An architecture for information retrieval over semi-collaborating Peer-to-Peer networks [A]. SAC 2004 [C]. Nicosia: ACM, 2004. 1078 - 1083.
- [2] MOON B, JAGADISH HV, FALOUTSOS C. Analysis of the clustering properties of the hilbert space-filling curve [J]. IEEE TKDE, 2001, 13(1): 124 - 141.
- [3] BOHM C. Search in high - dimensional spaces - index structures for improving the performance of multimedia databases [J]. ACM Computing Surveys, 2001, 33(3): 322 - 373.
- [4] Gnutella [EB/OL]. <http://gnutella.wego.com>, 2003.
- [5] SCHUTZE H, SILVERSTEIN C. A comparison of projections for efficient document clustering [A]. Proceedings of ACM SIGIR [C]. Philadelphia, PA: ACM Press, 1997. 74 - 81.
- [6] STOICA I, MORRIS R, KARGERD. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications [A]. Proceedings of SIGCOMM 2001 ACM [C]. San Diego, CA: ACM Press, 2001. 149 - 160.
- [7] EL-ANSARY S, ALIMA LO, BRAND P, *et al.* Efficient Broadcast in Structured P2P Networks [A]. Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03) [C]. Berkeley, CA, USA, 2003.
- [8] ROSE T, STEVENSON M, WHITEHEAD M. The reuters corpus volume1 - from yesterday's news to tomorrow's language resources [A]. Proceedings of the 3rd International Conference on Language Resources and Evaluation [C]. Las Palmas: ELRA, 2002. 29 - 31.

(上接第820页)

频和音频数据,目的节点往往固定且数据量大,所以 ODS-CDMA 能够发挥比较好的性能。

参考文献:

- [1] 田甜,孙献璞,王子奋. Ad hoc 网络的扩频多址接入协议的研究与展望 [J]. 电子科技, 2005, (9): 60 - 64.
- [2] QIANG G, LIU Z, ISHIHARA S. CDMA-based carrier sense multiple access protocol for wireless LAN [A]. IEEE Conference on VTC 2000 [C]. 2001. 1164 - 1168.
- [3] MUQATTASH A, KRUNZ M. CDMA-based MAC protocol for wireless AdHoc networks [A]. Proceedings of the 4th ACM International

Symposium on Mobile Ad Hoc networking and computing [C]. New York: ACM Press, 2003. 153 - 164.

- [4] TANG ZY, GARCIA-LUNA-ACEVES JJ. Hop-reservation multiple access (HRMA) for Ad hoc networks [A]. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99) [C]. New York: IEEE, 1999, 1.
- [5] 周晓东,李建东,杨军. 一种新的移动 Ad Hoc 网络的单址音码分多址协议 [J]. 西安电子科技大学学报, 2005, 32(4): 587 - 592.
- [6] NARDIS LD, BALDI P, BENEDETTO MD. UWB Ad hoc networks [A]. IEEE Conference UWBST [C]. 2002.