

# 基于分层象限空间的 P2P 超级节点查找技术

冯劲潇 陈贵海 谢俊元

(南京大学计算机软件新技术国家重点实验室 南京 210093)

**摘要** 在目前的非结构化超级节点查找中,采用简单的泛洪或随机游走消息传递方式,效率不高,同时系统没有将非结构模糊查询和结构化数据定位方式有效地结合。提出一种基于分层象限空间的新型超级节点结构 Quad,并在 Quad 上实现了两种非结构化超级节点查找方法:一是回溯扩展查找方法,该方法将泛洪和随机游走方式进行折衷,兼顾了网络流量和查询长度;二是利用 Bloom Filter 技术对回溯扩展查找进行改进。此外,Quad 同时支持结构化数据定位。模拟结果表明,Quad 查找在查询成功率、查找成本方面优于现有的超级节点查找方法。Bloom Filter 能有效提高 Quad 查询效率并降低查找长度,同时 Quad 具有高效的定位性能。

**关键词** 对等网络,超级节点,查找,分层象限空间

**中图分类号** TP393 **文献标识码** A

## P2P Super-peer Search Techniques Based on Hierarchical Quadrant Space

FENG Jin-xiao CHEN Gui-hai XIE Jun-yuan

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

**Abstract** Current super-peer search adopts flooding or random walk in routing method and their efficiencies are low. Meanwhile system does not combine the unstructured proximity search with structured data locating effectively. This paper proposed a new kind of super-peer topology based on hierarchical quadrant space called Quad and introduced two unstructured super-peer search methods. The first one is backtracking and expanding search (BES) technique which attains a tradeoff between flooding and random walk. BES considers both network traffic and search length. Second, bloom filter technique is adopted to improve the BES. Besides, Quad supports structured data locating. Simulations show that Quad super-peer search methods are better in search success ratio, search cost compared with current super-peer search techniques. Bloom Filter can effectively increase search efficiency and reduce search length. Meanwhile Quad is efficient in data locating.

**Keywords** P2P network, Super-peer, Search, Hierarchical quadrant space

近几年来,P2P 网络已经发展成为互联网最重要的基础架构之一。基于 P2P 结构的应用,如文件下载、视频通信网络流行迅速,研究人员也从路由寻址、流量监控及安全等多个角度展开了对 P2P 技术的研究。在这些应用和研究中,P2P 查找技术都是其中的核心问题。与传统的客户/服务器模型相比,P2P 网络中资源不再由中心服务器统一存放,而是分散在大量的网络边缘节点中,同时这些节点间没有统一的管理结构。因此,如何在高度动态并且节点自治的网络中准确高效地找到所需资源,成为 P2P 网络的难点与核心问题之一。

目前的 P2P 查找技术主要分为非结构化查找、结构化数据定位和超级节点查找 3 类方法。非结构化查找包括盲目查找和基于信息查找两种主要方法。在盲目查找<sup>[1-3]</sup>中,查询传递过程是完全随机的,在消息传递过程中不利用任何知识判断下一跳路由,因而查询效率偏低,并且易于产生过高的网络流量。信息查找<sup>[4-6]</sup>利用节点资源信息或历史查询结果等信

息建立索引,查询优先向命中率可能最高的节点进行传递。尽管和盲目查找相比,信息查找在查询命中率和查找成本上都有所改善,但总体来说,非结构化查找中查询跳数和查询结果都是不确定的,查询效率也不高。

在结构化的数据定位方式<sup>[7,8]</sup>中,系统利用分布式哈希表(DHT)机制实现资源到节点的映射,节点名称和资源名称都通过一致性哈希函数映射为一个特定的 ID 标识,资源存放在与其 ID 标识最接近的节点上。其优点在于性能较高,达到  $O(\log N)$ ,缺点在于查询时需要准确指定查询关键词,不支持模糊查询,并且在动态性强的环境中维护成本过高,因而结构化的数据定位方式不适应动态性强的广域网环境。

为了改进上述查询方法的不足,研究人员提出基于超级节点结构的查找技术<sup>[9,10]</sup>。在超级节点结构中,能力强的节点被选为超级节点,普通节点不参与路由,查询通过超级节点进行。由于减少了参与路由的节点数量,并且超级节点比普

到稿日期:2009-04-24 返修日期:2009-07-08 本文受国家自然科学基金(60573131,60721002,60721002),国家“九七三”重点基础研究发展规划项目基金(2006CB303000)资助。

冯劲潇(1976-),男,博士生,主要研究方向为 P2P 网络等,E-mail: fjx\_mailbox@126.com;陈贵海(1963-),男,教授,博士生导师,主要研究方向为并行计算和无线网络等;谢俊元(1961-),男,教授,博士生导师,主要研究方向为人工智能、网络安全、网络计算等。

通节点具有更好的能力和稳定性,因此在非结构化查询和结构化定位中合理利用超级节点结构能够改善这些查询的不足。目前超级节点查找仍然存在两个问题:(1)超级节点之间只是采用简单的泛洪或随机游走等消息传递方式,效率不高。(2)没有将非结构模糊查询和结构化数据定位方式结合起来。

本文提出一种基于分层象限空间的超级节点结构,称为Quad,并在Quad上实现了3种超级节点查询方法:一是超级节点层的回溯扩展查找技术,对现有的盲目查找方法进行改善,兼顾了网络流量和查询长度;二是利用Bloom Filter技术降低回溯扩展查找技术的查询成本并提高其查询效率;三是在Quad上实现了结构化定位技术。模拟结果表明,Quad查找在查询成功率、查找成本上优于现有的超级节点查找方法,Bloom Filter能有效提高查找效率并降低查找长度,同时Quad具有高效的定位性能。

## 1 Quad网络模型

### 1.1 模型描述

Quad网络模型拓扑采用两层拓扑结构,从逻辑上分为超级节点层和叶节点层,如图1所示。其中超级节点是拓扑的核心,担负着系统的路由任务,并负责为叶节点建立和发布索引。叶节点通过超级节点进行查询。

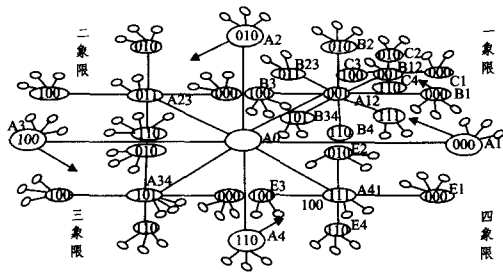


图1 Quad网络模型

超级节点层按分层象限空间进行组织。分层象限空间是象限空间的扩展。象限空间是一种二维平面空间,以原点为基准,横轴和纵轴将平面划分成4个象限。从原点的右轴开始,象限空间按逆时针方向分为一、二、三和四象限。象限空间的每个象限可继续划分为4个子象限,逐层划分,就形成分层象限空间。在图1中,象限中心点 $A_0$ 有 $A_1, A_{12}, A_2, A_{23}, A_3, A_{34}, A_4$ 和 $A_{41}$ 8个邻居,每个邻居节点包含一个方向,按逆时针顺序分别表示为二进制形式的(000, 001, 010, 011, 100, 101, 110, 111)。 $A_0, A_1, A_2, A_3$ 和 $A_4$ 5个超级节点互为同层邻居, $A_0$ 包含 $A_{12}, A_{23}, A_{34}, A_{41}$ 4个下层邻居,第一象限的 $A_1$ 包含 $B_1, B_2, B_3, B_4, A_{12}$ 5个下层邻居,其余3个象限类似。在Quad分层象限空间中,每个位置的标识值是唯一的,由 $A_0$ (不包含 $A_0$ )到该节点路径包含的节点方向构成,如 $A_{12}$ 位置标识为001, $B_1$ 位置标识为001000。 $A_0$ 可以设为一个特定的标识。

超级节点分为两种。象限横轴和纵轴的交点称为中心超级节点(Center Super-peer, CSP),如 $A_0, A_{12}, A_{23}$ 等;横轴和纵轴的4个端点构成4个端超级节点(Boundary Super-peer, BSP),如 $A_1-A_4, B_1-B_4$ 等。超级节点除了包含经过哈希函数映射后的ID标识,还具有位置标识。当超级节点离开后,位置标识分配给处于相同位置的新超级节点使用。超级节点按照能力和在线时间等综合因素动态地从普通叶节点中选拔

优秀叶节点作为候选超级节点。候选超级节点对父超级节点路由表和资源索引进行备份,并定期和超级节点交换信息,以便在超级节点失效时能接替其位置。每个叶节点通过一个超级节点加入到网络中。

### 1.2 路由表

Quad中超级节点的路由表包括两类:一类是邻居路由表(NR),记录超级节点与同层、下层和上层邻居超级节点之间的路由关系;另一类是象限路由表(QR),支持顶层象限不相同的超级节点之间定位。路由表如图2所示。

路由类型	同层路由	下层路由	上层路由		
索引号	0 2 4 6	1 3 5 7	8 9		
邻居路由表	$A_{12}$ (CSP)	$B_1 B_2 B_3 B_4$	$A_0 A_1$		
索引号	0 2 4 6	1 3 5 7 8	9		
	$B_1$ (BSP)	$A_{12} B_2 B_3 B_4$	$C_1 C_2 C_3 C_4 B_{12}$		
象限路由表	象限号	第一象限	第二象限	第三象限	第四象限
	CSP, BSP	$S_0 S_1$	$S_2 S_3$	$S_4 S_5$	$S_6 S_7$

图2 路由表

以图1为例,中心超级节点 $A_{12}$ 的NR包括同层NR( $B_1, B_2, B_3, B_4$ ),对应索引(0, 2, 4, 6)、下层NR( $B_{12}, B_{23}, B_{34}, B_{41}$ ),对应索引(1, 3, 5, 7)、上层NR( $A_0, A_1$ ),对应索引(8, 9)。端超级节点 $B_1$ 的NR包括同层NR( $A_{12}, B_2, B_3, B_4$ ),对应索引(0, 2, 4, 6),其中该BSP节点自身方向对应的索引用来存储CSP,下层NR( $C_1, C_2, C_3, C_4, B_{12}$ ),对应索引(1, 3, 5, 7)、上层NR( $A_1$ ),对应索引(9)。在Quad中,超级节点NR中节点数上限为10,不随网络规模增大而同步变化。容易看出,CSP的4个同层BSP和4个下层CSP的索引号与这些节点的方向是一致的,所以超级节点的位置标识(PI)由距离其最近的CSP位置标识加节点自身所包含的方向构成。

QR记录超级节点所在顶层象限之外的其他3个顶层象限内节点,因为顶层象限只有4个,所以超级节点QR只需记录其他3个顶层象限内的节点即可,其中每个象限取两个超级节点(可以为空)。假设超级节点为 $S$ ,某个象限的路由表中两个节点为 $S_1, S_2$ ,则三者之间的层次关系满足 $(L_{s_1} < L_s) \wedge (L_{s_2} \leq L_s)$ ,即象限路由表中的两个超级节点处于不同层并且层次都不超过该超级节点的层次,因此象限路由表最多包含6项。Quad对于QR的节点要求并不严格,因为只需要知道特定象限的一个节点就可以定位到该象限。取不同层并且层次与本节点相近的两个节点是为了降低超级节点之间路由项的重复率。另外,将每个象限的路由表项数设为2,是为了增加系统的容错性。在同一顶层象限内,超级节点利用邻居路由表进行定位。

## 2 Quad非结构化查询

### 2.1 回溯扩散查找方法

非结构化查询包括泛洪和随机游走两种主要查找方法。两种方法的缺点都比较明显。前者消息呈指数级增长,后者查询路径过长。我们在Quad中将两种查找方法进行折衷,提出回溯扩展查找(BackTracking and Expanding Search, BES)方法。超级节点收到叶节点查询请求后,先通过邻居路由表按随机游走的方法沿一条路径到达象限的最底层。如果没有命中目标,则向上回溯一层,然后向同层其他所有邻居超级节点进行查询,每个超级节点收到查询后,在本象限内利用泛洪方法进行扩展查询。如果本次扩展没有匹配的查询结果,则

超级节点再向上回溯一层,然后按照同样的方法向其他同层超级节点转发查询,这些超级节点再在本象限内进行扩展查询。这样每回溯一次,会在更大的范围内进行查询,其作用相当于泛洪方法的一跳。与泛洪方法相比,BES能有效控制TTL值,因为只有前一次回溯扩展没有命中的情况下才进行下一次回溯。同时由于Quad的分层特征,在回溯扩展的过程中没有重复消息。与扩展环查找技术<sup>[9]</sup>相比,BES不要预设查询跳数,效率更高。

## 2.2 利用 Bloom Filter 改进 BES 查询

在 Quad BES 中,超级节点在每次回溯扩展的过程中要向所有下层节点转发查询。回溯的层次越高,向下的消息量就越大,从而降低效率。如果在每一次向下传递时,能够通过索引判断一条路径是否包含所查的资源,则可以消除不必要的传递,使路由效率大大提高。基于这个思路,我们采用 Bloom Filter<sup>[11]</sup>技术对 Quad BES 查找方法进行扩展。

### 2.2.1 Bloom Filter 压缩技术

Bloom Filter(BF)是 Burton Bloom 提出的一种压缩数据的优秀结构,它采用一个长度为  $m$  的位向量来表示一个数据集。一个 BF 利用  $K$  个相互独立的 Hash 函数( $h_1, h_2, \dots, h_k$ )对元素集( $e_1, e_2, \dots, e_n$ )求 Hash 值,然后对  $m$  取余,其表示的值域为  $0..m-1$ 。再将位向量对应位置 1,图 3 显示了一个简易数据集( $e_1, e_2$ )和 4 个独立哈希函数得到的位图。

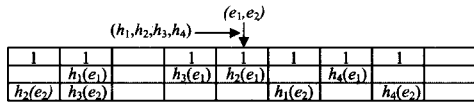


图 3 Bloom Filter 示例

查询某个数据对象  $e_i$  是否是某个集合  $E$  的成员时,采用同样的哈希函数计算对象的 Hash 值。经过对  $m$  取余后,检测对应的位向量值是否全为 1,如果不全为 1,则该数据对象肯定不属于集合  $S$ ;如果全为 1,则存在一定的伪阳率(False Positive)的可能性。这是因为由于位向量空间的限制,不同数据对象的哈希值可能会映到同一个位向量位置。

### 2.2.2 用 Bloom Filter 建立索引

在 Quad 中,叶节点通过超级节点加入网络。在与超级节点建立连接后,叶节点将共享资源的非结构化索引传递给超级节点,超级节点将叶节点中各个资源对象的名称加入本地 BF 向量。为了记录邻居超级节点的 BF 向量,超级节点维护一个  $m \times n$  的二维 BF 向量表  $A$ ,其中行  $m$  代表超级节点方向,列  $n$  代表距离本节点的层数。在 Quad 中,超级节点分为中心超级节点和端超级节点。对中心超级节点来说, $m$  为 4,表示 4 个象限, $A_{ij}$  表示第  $i+1$  象限距离本节点  $j$  层的超级节点 BF 向量。如  $A_{00}$  表示第 1 象限的同层端超级节点向量, $A_{01}$  表示第 1 象限距离本节点 1 层的经过合并的子节点向量, $A_{02}$  表示第 1 象限距离本节点 2 层的经过合并的子节点 BF 向量。不同层次间的超级节点二维 BF 向量表  $A$  是有关联的,如超级节点  $A_{12}$  等于下层子节点的  $A_{11}$ 。对端超级节点来说,下层象限节点包含 5 个子节点,因而  $m$  为 5。 $A$  表其他内容与中心超级节点类似。

超级节点定期将 BF 数据传递给同层和上层邻居超级节点。向同层邻居超级节点传送的 BF 数据只包括自身 BF 向量,向上层邻居超级节点传送的 BF 数据包括自身和一定层次的子超级节点 BF 向量。在处理子超级节点 BF 向量时,将

处于同一层的所有子超级节点 BF 向量合并成一个 BF 向量。例如当超级节点  $A$  表列数为 3 时,超级节点向上层传递自身和下面 2 层子超级节点的合并向量(如果下层超级节点不足 2 层,相应的 BF 向量为空)。相应地,上层超级节点若保持下面 3 层范围内子超级节点 BF 向量,则填入  $A$  中相应位置。

### 2.2.3 基于 Bloom Filter 技术的查询

在有了 BF 向量索引后,查询仍然采用回溯扩展查找方法的传递路径。但在节点的选择上不再向所有节点进行传递,而是按照 BF 匹配情况向下进行传递。当超级节点收到叶节点查询请求后,首先按照 BF 索引选择下一层满足匹配的一个超级节点进行传递。如果没有匹配项,就选择任意一条路径向下。如果传递过程中有匹配项,则将查询结果反馈给发起查询的超级节点,否则一直到达最低层超级节点。如果仍然没有匹配项,则将请求回溯到上一层超级节点,该超级节点将请求转发给同层超级节点。每个超级节点在向下扩展时,不再向所有下层超级节点进行扩展,而是先通过 BF 向量表进行判断,如果沿某个方向的一定层范围内有匹配项,则沿该路径向下传递,否则就放弃该路径。按照 BF 性质,在满足 BF 向量位图匹配的路径中,存在一定的伪阳率,但该方法可以减少无效的传递路径。

## 3 Quad 结构化定位

Quad 结构化路由通过查询路由表来逐步实现定位。超级节点(设为  $S$ )接到叶节点资源发布或查询请求后,首先判断自身的位置标识( $PI_s$ )与资源 ID(设为  $K$ )是否属于同一顶层象限。如果两者不一致,则通过象限路由表将请求传递给与  $K$  顶层象限相同的超级节点。如果两者一致,则先定位到象限前缀匹配值最长的超级节点,再从该节点按照邻居路由表逐层向下定位到目标节点。

### 算法 1 Procedure Routing( $PI_s, K$ )

(1)  $S$  判断 ID 标识  $K$  与自身的位置标识  $PI_s$  是否在同一顶层象限。方法是将两者的第一个三位二进制组的前二位进行比较。

(2) 如果两者不在同一顶层象限,则  $S$  通过象限路由表查找到与  $K$  顶层象限相同并且前缀匹配值最长的一个节点  $S$ ,然后转第(3)步。

(3) 如果两者在同一象限,则  $\bar{S}=S$ ,计算  $S$  位置标识  $PI_s$  与  $K$  的最长象限前缀值  $m$ ,并将请求从当前节点向上传递  $L_s - m$  层,定位到与  $K$  象限前缀匹配值最长的节点,然后按下列过程向下层定位:

① 如果该节点不是 CSP,则从 BSP 将请求传递给对应的 CSP(设为  $S_2$ )。

② 取  $K$  的第  $m$  个三位二进制组,从  $S_2$  的邻居路由表中取该二进制组对应的下层 CSP。如果下层 CSP 存在,则取  $K$  的下一个三位二进制组,重复本步骤。如果下层 CSP 不存在,转第③步。

③ 从  $S_2$  邻居路由表中取该二进制组对应的同层 BSP(设为  $S_3$ )。如果该节点存在, $S_3$  就是目标节点,否则转第④步。

④ 超级节点下层 CSP 和同层 BSP 都不存在,目标节点为超级节点自身。

## 4 模拟验证

在 PeerSim<sup>[12]</sup> 基础上用 Java 分别实现了 Quad 协议和类似于 Gnutella 0.6 版的两层非结构化超级节点拓扑(简称为 Gnu)模拟程序。在 Quad 协议中,实现了 Quad 回溯扩展查询(Quad BES)、基于 Bloom Filter 的改进查询(Quad BF)以及 Quad 定位。在 Gnu 协议中,实现了随机游走查询方式(Gnu RW)。为了和 Gnu RW 查询方式的跳数相比较,在计算 Quad 查询跳数时,以查询传递的超级节点数来计算跳数。模拟中两种协议的节点构造方式、随机函数都保持一致,节点容量服从 Power-Law 规则,分布指数为 1.8,具体参数如表 1 所列。模拟比较的内容包括查找成功率、查找成本、查找长度等方面。

表 1 模拟参数

Parameter	Value	Default
Network Size	10000~40000	10000
Peer Capacity	20~80	—
Peer Load	20~80	—
Resource Number	5~30	15
BF Bits Vector length	100~1000	1000
Hops	5~30	10
Search Number	100~3200	200
Hash Function Number	8	—
Gnu Neighbour Number	20	—

### 4.1 查找成功率

图 4 比较了 Quad BES, Quad BF 和 Gnu RW 3 种查询方式在不同查询跳数下的成功率。从图中可以看出, Quad BES 查询方法和 Gnu RW 在不同的跳数限制下查询成功率比较接近。如在 10 跳限制下, Quad BES 查询成功率为 29.5%, 而 Gnu RW 是 32.5%; 在 30 跳限制下, Quad BES 查询成功率为 60.5%, 而 Gnu RW 是 57.5%。与这两者相比, Quad BF 查询成功率要明显高出这两种方法。例如在 10 跳和 30 跳下, 它的成功率分别达到 48.5% 和 97.5%。从图中可以看出, 当查询跳数在 20 以上时, Quad BF 的查询成功率都保持在 80% 之上, 查询具有较高的命中率。

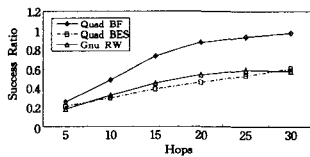


图 4 查询成功率 vs 查找跳数

### 4.2 查找成本

我们用超级节点在查询过程中处理的消息数来表示查找成本。图 5 是 Quad BES, Quad BF 和 Gnu RW 3 种查询方式在不同查询次数下的总消息处理成本。模拟中 Gnu RW 的最大跳数限制设置为 32, 该值与 Quad BES 的平均查找跳数相等。

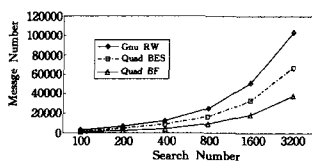


图 5 查找总成本比较

从图 5 中可以看出, 在 3 种查找中, Gnu RW 的查询成本是最高的, Quad BES 次之, Quad BF 查询成本最低, 例如当查

询次数为 800 时, Gnu RW 的消息处理成本为 25263, Quad BES 的消息处理成本为 16407, Quad BF 的消息处理成本为 9338; 当查询次数为 3200 次, 3 种查询方法的消息处理总成本分别为 103883, 66393 和 38225, Quad BF 的查询总成本只占 Quad BES 的 50%、Gnu RW 的 30% 左右。

### 4.3 查找长度

图 6 显示了节点包含的资源数量变化对 Quad BES 和 Quad BF 查找长度的影响。可以看出, 当每个节点的最大资源数从 5 提高到 30 之后, Quad BES 查询最大跳数从 143 降到 53, 查询平均跳数从 63 降到 9, 显示了节点资源数量对 Quad BES 查找长度具有较大的影响。此外, 图中数据显示了节点资源数变化对 Quad BES 查找长度的影响并不一致。例如当单个节点最大资源数从 5 提高到 20 时, 查询的最大跳数和平均跳数下降比较明显。但当节点的资源数继续增加时, 查询跳数变化不大。与 Quad BES 比较, 节点资源数的变化对 Quad BF 查找长度影响比较小。例如当节点最大资源数从 5 提高到 30 时, 它的最大跳数在 25 到 51 之间波动, 不随资源数增加而同步减少。从查找长度平均值来看, 随着节点资源的增加, Quad BF 查询跳数保持较为缓慢的下降, 其值从 12 降到 6, 体现了 Quad BF 查询对节点资源变化具有的良好适应性。

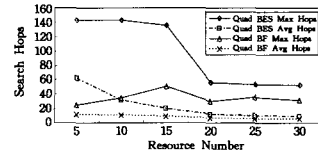


图 6 节点资源数对查找长度的影响

### 4.4 位向量长度对 Quad BF 的影响

在 Bloom Filter 压缩索引方法中, 不同的元素可能映射到 BF 位向量的同一个数据位, 产生一定的伪阳率。伪阳率与位向量的长度是紧密相关的, 位向量的长度越大, 伪阳率就越低, BF 向量位匹配时命中率也越高。图 7 显示了不同的 BF 位向量长度对 Quad BF 查询的平均跳数和最大跳数的影响。从图中还可以看出, 随着 BF 位向量长度的增加, 查询的平均跳数和最大跳数会减少。例如当位向量长度从 100 位上升到 1250 位时, 平均查询跳数从 29 降到 10, 最大查询跳数从 125 降到 31, 查找跳数降低了 60% 以上。另外, 从图中还可以看出, 在位向量长度增加的初期, 查找跳数提升比较明显, 但是随着位向量长度增加到一定程度后, 位向量长度变化对跳数的影响很小。

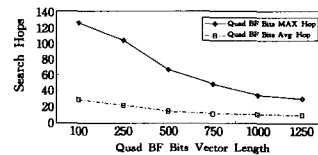


图 7 不同向量位长度对查找跳数的影响

### 4.5 Quad 结构化定位

图 8 反映了 Quad 结构化定位性能。模拟中系统的节点数从 10000 到 45000, 比较的数据包括 Quad 平均查找长度和最大查找长度。从图中可以看出, 在不同的网络规模下, 系统查找的平均和最大查找长度均保持一个缓慢的增长。例如, 当网络规模从 10000 变化到 45000 时, 平均查找长度从 3 跳增加到 5 跳, Quad 的最大查找长度从 6 跳增长到 8 跳, 显示 Quad 数据定位具有很高的效率。

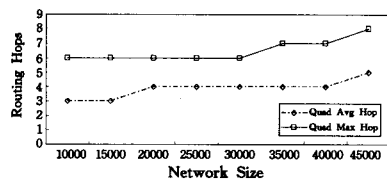


图8 Quad 结构化定位性能

**结束语** 本文提出一种基于分层象限空间的超级节点拓扑结构 Quad, Quad 可以同时支持非结构化查找和结构化数据定位。模拟结果表明, Quad 查找比现有的超级节点查找技术具有更高的查询成功率、更小的查询成本。Bloom Filter 技术能有效提高 Quad 查询效率, 降低 Quad 查找长度, 同时 Quad 具有高效的定位性能。

### 参考文献

[1] The Gnutella Protocol Specification v0.4[EB/OL]. [http://www9.limewire.com/developer/gnutella\\_protocol\\_0.4.pdf](http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf)

[2] Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D. A local search mechanism for peer-to-peer networks[C]//The ACM 11th Int'l Conf. Information and Knowledge Management (CIKM'02). Virginia, USA, 2002; 300-307

[3] Gkantsidis C, Mihail M, Saberi A. Random walks in peer-to-peer networks[C]//Proceedings of the IEEE INFOCOM 2004. New York: IEEE Press, 2004; 120-130

[4] Yang B, Garcia-Molina H. Improving search in peer-to-peer net-

works[C]//Rodrigues L E T, Raynal M, Chen W S E, eds. Proceedings of the 22nd Int'l Conf. on Distributed Computing Systems. Washington: IEEE Computer Society, 2002

[5] Crespo A, Garcia-Molina H. Routing indices for peer-to-peer systems[R]. Computer Science Department, Stanford University, 2002

[6] Tsoumakos D, Roussopoulos N. Adaptive probabilistic search (APS) for peer-to-peer networks[R]. CS-TR-4451. 2003

[7] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for Internet applications[C]//Cruz R, Varghese G, eds. Proceeding of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SigComm). New York: ACM Press, 2001; 149-160

[8] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems[C]//Guerraoui R, ed. Proceedings of the IFIP/ACM Int'l Middleware Conf. London: Springer-Verlag, 2001; 329-350

[9] Gnutella protocol spec. v. 0.6 [EB/OL]. <http://rfc-gnutella.sourceforge.net/src/rfc-0.6-draft.html>

[10] 夏启志, 谢高岗, 闵应骅, 等. IS-P2P: 一种基于索引的结构化 P2P 网络模型[J]. 计算机学报, 2006, 29(4): 602-610

[11] 张一鸣, 卢锡城, 郑倩冰, 等. 一种面向大规模 P2P 系统的快速搜索算法[J]. 软件学报, 2008, 19(6): 1473-1480

[12] PeerSim Simulator[EB/OL]. <http://peersim.sourceforge.net/>

(上接第 51 页)

渐减弱后发起切换。

图 6(a) 显示 MT 移出 WLAN 网络后, 3G 流量开始增加, 此时 MT 不再使用 WLAN 提供的网络服务, 并且 MT 所获得的带宽资源也减少了。图 6(b) 则说明了与 PFL-VHO 算法相比, 使用平均值预测与坡度预测法后, 从 WLAN 网络切换到 3G 网络过程中, 切换延迟降低, 从而使得由切换引起的丢包率也大为减少, 平均为 30%, 切换效率得到了提高。图 6(c) 显示了 10 次 MT 从 WLAN 切换到 3G 网络的切换延迟的比较。MV-SLP 策略由于计算复杂度低, 并进行了有效的预测从而减少了切换延迟。切换过程的延迟平均为 6s, 而 PFL-VHO 平均切换延迟为 10s。可见本文所提算法有效地减少了切换延迟, 保证数据能够被高效可靠地传输。

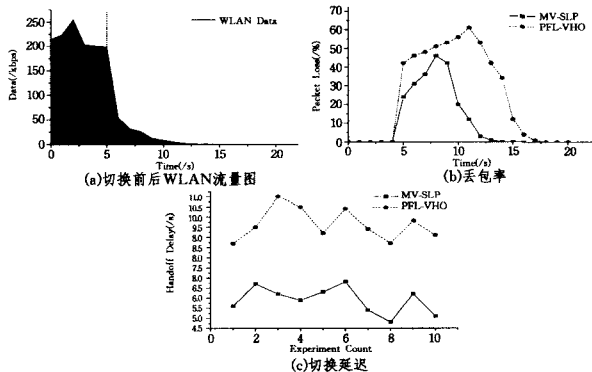


图6

**结束语** 本文针对 3G 网络与 WLAN 网络的切换进行了研究。首先分析了 3G 到 WLAN 网络切换的过程, 提出了对现有技术的改进措施, 并设计了基于背景扫描的可以提高切换效率的切换策略。然后分析了 WLAN 到 3G 网络的切

换过程, 并提出了基于 beacon 信号预测的切换发起改进策略, 在 MT 断开与 WLAN 连接之前提前与 3G 节点进行通信并建立连接, 从而缩短由于 WLAN 连接断开而 3G 连接未建立所带来的无连接状态的时间。实验结果显示提出的两个改进策略在实际网络环境中取得了很好的应用效果, 在 3G 切换与 WLAN 的过程中, 尽可能地达到无缝切换的效果, 最大程度地缩短了切换延迟。

### 参考文献

[1] Newman P. In search of the all-IP mobile network[J]. IEEE Communications Magazine, 2004, 42(12): 53-58

[2] Chen W T, Liu J C, Huang H K. An adaptive scheme for vertical handoff in wireless overlay networks[C]//Proceedings of the Tenth International Conference on Parallel and Distributed Systems (ICPADS'04). Newport Beach, California, USA, 2004; 541-548

[3] 刘侠, 蒋铃鸽, 何晨. 一种基于模糊逻辑的预判辅助垂直切换算法[J]. 电子学报, 2007, 35(10): 1989-1993

[4] Siddiqui F, Zeadally S, Fowler S. A Novel Architecture for Roaming between 3G and Wireless LANs[C]//Multimedia Services Access Networks (MSAN'05). Orlando, Florida, USA, 2005; 101-105

[5] Vriendt D J, Laine P, Lerouge C, et al. Mobile Network Evolution: A Revolution on the Move [J]. IEEE Communications Magazine, 2002, 40(4): 104-111

[6] Chen J C, Chan S H, He J, et al. Mixed-Mode WLAN: The Integration of Ad Hoc Mode with Wireless LAN Infrastructure[C]//IEEE Global Telecommunications Conference (GLOBECOM'03). San Francisco, California, USA, 2003; 231-235

[7] Wang S-W. A Mobile IPv6 based Fast Handoff Scheme for 3G-WLAN Interworking [D]. Taiwan: Chaoyang University of Technology, etd-0702107-121352. June 2007