

文章编号:1001-9081(2009)03-0646-03

基于层次划分的 RP2P 路由算法

李 园¹, 陈世平^{1,2}

(1. 上海理工大学 计算机与电气工程学院, 上海 200093; 2. 上海理工大学 网络管理中心, 上海 200093)

(ivylee.china@gmail.com)

摘 要: RP2P 路由算法将用于非结构化 P2P 网络中的随机邻居选择策略与结构化的分布式哈希表 (DHT) 环相结合, 可在 d 跳内处理查询请求。但是, 由于网络中的主机在网络带宽、内存、CPU 等方面的能力差别很大, 那些能力较弱的节点势必会影响整个系统的效率。利用网络中节点性能的差异, 结合分层的概念, 提出基于层次的 RP2P 路由算法, 并对其性能进行了分析, 算法在一定程度上缓解了网络中一部分节点的频繁加入和退出所引起的系统震荡。模拟实验表明, 基于层次的 RP2P 路由算法有效提高了搜索的效率。

关键词: P2P; 分布式哈希表; 路由延迟; 随机技术; 超级节点

中图分类号: TP393 **文献标志码:** A

An efficient RP2P network based on hierarchical dividing

LI Yuan¹, CHEN Shi-ping^{1,2}

(1. School of Computer and Electrical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China;

2. Network Center, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: RP2P algorithm combines arbitrary neighbor selection, typically used only in unstructured P2P networks, with a Distributed Hash Table (DHT) ring. It is the first of its kind to resolve requests in d hops with a chosen probability of $1 - c$. However, the capacities of the hosts participating in the network, such as bandwidth, memory, CPU, are very different, which will affect the efficiency of the whole system. On the other hand, the shock caused by some of the nodes in the network frequent joining in/departing from the system is also one of the factors affecting the performance. This paper analyzed the capacities of the nodes and proposed an efficient RP2P network based on hierarchical dividing. It improves the efficiency of the system, and solves the problem of system shocks.

Key words: P2P; Distributed Hash Table (DHT); routing delay; randomized strategy; super node

0 引言

随着互联网技术的迅速发展, P2P 网络越来越广泛地应用于资源交换、数据管理、文件共享等领域。基于 P2P 技术的网络应用和服务已经成为互联网的重要组成部分。P2P 网络改变了传统 C/S 模式集中存储和处理资源的方式, 将网络上的资源有效地组织起来, 使资源提供者和资源接收者之间能够直接交换信息。P2P 网络的核心问题是如何在分布式环境下快速准确地进行资源搜索, 其中拓扑结构是决定资源搜索性能的一个重要因素。

按照资源组织与定位方法, P2P 网络可分为非结构化 P2P 网络和结构化 P2P 网络。非结构化 P2P 网络基于泛洪的搜索机制, 在大规模网络中会产生大量的通信负载, 可扩展性较差。基于分布式哈希表 (Distributed Hash Table, DHT) 的结构化 P2P 网络具有良好的可扩展性。在不需要服务器的情况下, 每个客户端负责一个小范围的路由, 并负责存储一小部分数据, 当一个查询请求发出, 请求将会根据特定的路由算法通过 P2P 网络路由到管理特定数据的节点, 从而实现整个 DHT 网络的寻址和存储。经典的查询路由协议有 Tapestry^[1], Pastry^[2], CAN^[3], Chord^[4] 等。其中在 N 个节点的 Tapestry 网络中, 每个节点有 $O(\log N)$ 个邻居, 所有路由路径

为 $O(\log N)$ 跳; 给定 N 个节点的 Pastry 覆盖网络中, 路由一个消息需要 $O(\log N)$ 步, 每节点需要维持 $O(\log N)$ 个入口; CAN 是一个具有良好可扩展性的系统, 给定 N 个节点, 系统维数为 d , 每节点状态信息与网络规模无关为 $O(d)$, 但其路由路径长度为 $O(kN^{1/k})$; 而 Chord 逻辑上将所有的节点看作一个按照节点的哈希值排列的环, 路由算法采用了类似二分查找的方法, 每次查找发送的消息数为 $O(\log N)$ 。

综上所述, 大多数现存的 P2P 网络都需要 $O(kN^{1/k})$ 、 $O(\log N)$ 、 $O(\log N / \log k)$ 跳路由解决查询请求, 随着网络规模的扩大, 路由路径长度将会越来越大, 搜索效率越来越低。此外, 对于结构化 P2P 网络而言, 网络波动也增大了维护路由信息的网络开销。在一个缺少集中化服务器的动态环境下, 各个节点需要维持一致的网络拓扑信息。由于 P2P 的本质是一个多跳自组织的网络, 允许对等点任意地加入和离开, 大量路由信息需要更新, 节点的不稳定增大路由信息的维护成本, 增大了网络负载。另一方面, 网络波动后, DHT 算法需要多跳路由才能找到准确信息, 网络缺乏有效的自我恢复机制, 降低了 DHT 算法的效率。

目前, 在国内外对具有小路由特性的 P2P 网络的研究并不多见。文献[5]提出的算法可以在常数跳路由内解决查询请求, 但网络中每一个节点都必须维护一个巨大的路由表, 在节点频繁

收稿日期: 2008-09-22; 修回日期: 2008-12-15。

基金项目: 国家自然科学基金资助项目 (60573142); 上海市 (第三期) 重点学科项目 (S30504)。

作者简介: 李园 (1983-), 女, 吉林长春人, 硕士, 主要研究方向: P2P 计算; 陈世平 (1964-), 男, 浙江绍兴人, 教授, 博士, 主要研究方向: P2P 计算、计算机网络通信、数据库与知识库。

加入或离开的网络中,会引起相当大的维护负载。RP2P路由算法^[6]将主要用于非结构化P2P网络中的随机技术与结构化的Chord环相结合,能够达到在 $O(d)$ 的路由路径长度内解决查询请求的同时,时间复杂度仅为 $O((- \ln c)^{1/d} N^{1/d})$ 。

RP2P算法的思想简单,具有很好的时间复杂度和空间复杂度,但存在的问题是:在RP2P网络中的每个节点,无论其性能如何,都被赋予了相同的责任,包括查询、下载等。但是实际上,网络中的主机在网络带宽、内存、CPU等方面的能力往往差别很大。随着节点数量不断增加和网络规模的不断扩大,节点性能的差异会严重影响RP2P网络的效率,一些性能较低的节点可能成为系统的瓶颈。据资料统计,在现在的P2P系统中,仍有很多拨号上网用户和一些使用性能相对低下的PC机用户,这些节点使得系统的性能恶化。另一方面,网络中部分节点的频繁加入和退出所引起的系统震荡也影响了系统的性能。

我们通过基本的RP2P网络中引入分层的概念,充分利用节点性能的差异,在节点可以承受的范围内,给性能高的节点赋予更多的责任,不但提高了系统的效率,而且能够有效地处理系统震荡问题。

1 RP2P路由算法

RP2P路由算法通过哈希函数映射节点的IP地址、关键字为一个 m 位的标识符,这个ID空间被视为一个从0到 $2^m - 1$ 的环。设节点数为 N ,则 N 个节点将ID空间分为 N 个部分。节点 x 负责 x 的前驱节点与 x 之间的 id 所对应的位置信息,记为 $seg(x)$ 。如果一个关键字的 id 属于 $seg(x)$,那么它的位置信息会存储在节点 x 上。

在RP2P网络中,能够直接通信的节点之间互称为邻居。每一个节点都有两种类型的邻居:顺序邻居和随机邻居。节点 x 将它的一些前驱节点和后继节点作为其顺序邻居,记为 S_x ;将一些随机选择的邻居作为其随机邻居,记为 R_x 。 S_x 和 x 所负责的部分记为 $sup_seg(x)$ 。对于任意节点 x 的每一个顺序邻居 y , x 存储了 y 所负责的部分 $seg(y)$,同时 x 也得到了 $sup_seg(x)$;对于 x 的每一个随机邻居 z , x 存储了它的 $sup_seg(z)$ 。

我们用 $RP2P(d, c)$ 表示一个RP2P算法,其中, d 为拟定的完成一次查询的路由跳数, $1 - c$ 为 d 跳内完成查询的概率。当节点 x 收到一个查询请求 $request(id)$,如果 $id \in seg(x)$,则将查询结果返回给请求者,查询在0跳路由内完成;如果 $id \in seg(y)$ (其中 $y \in S_x$)则将查询请求转发给节点 y ,查询在1跳路由内完成;如果 $id \in sup_seg(z)$ (其中 $z \in R_x$)则将查询请求转发给节点 z ,查询在2跳路由内完成;否则,将查询请求在随机邻居中进行多播,查询请求在常数跳内到达。

2 基于超级节点的RP2P路由算法

由于互联网的广泛应用,一个P2P系统的用户可以数以万计。主机在网络带宽、内存、CPU等方面的能力往往差别很大。据统计,仍有近40%的网民采用拨号的方式上网,另外也有使用性能很差的PC机的用户。这些节点在RP2P系统中也同样作为一个独立的节点存在,承担着与其他节点完全相同的责任,参与整个系统的运行。但是受自身资源的限制,它们势必会带来系统响应时间增加等问题,使得整个系统

的性能受到很大影响,这并不符合P2P计算的初衷。为了解决这一问题,我们考虑让处理能力强、网络带宽大的节点作为超级节点来维护一部分普通节点,超级节点负责管理那些性能较差的普通节点,并代替那些性能较差节点来完成它们的一部分工作。同时,将资源的搜索分解到更小的组中进行,而各个组中的超级节点拥有指向组内各节点的索引信息,使它的搜索更加快速、高效。

2.1 超级节点的形成

系统通过节点的CPU、存储容量、发送和接受能力以及网络带宽来确定超级节点。在网络形成时,系统会对每一个节点的信息进行分析,测试其CPU及网络带宽等,得到节点 x 的能力 C_x 。超级节点是发送和接受能力强、存储容量大、处理速度快的主机,负责管理所在组节点的加入、离开和更新。超级节点的性能决定了整个网络的性能。普通节点则是与其他主机联系较少,网络带宽较小的主机。

假设每个超级节点所负责的普通节点数为 k ,整个网络中的节点数为 N ,则RP2P主干网络中有 N/k 个节点,即网络中共有 N/k 个超级节点。在系统运行过程中,选择前 N/k 个 C_x 值最高的节点作为超级节点,而其他的节点作为普通节点。超级节点之间构成一个RP2P的主干网络,查询请求在超级节点之间进行转发。而每个普通节点都与超级节点相连接,由超级节点统一管理。RP2P系统中超级节点对所有文件进行分配。

每个超级节点 x 负责 id 落在超级节点中 x 的前驱节点到 x 之间的普通节点,这些普通节点的集合记作 O_x 。在各个超级节点负责的组内构造内层DHT,赋予每个节点一个 m 位的标志 o_id 。相应地,每个key值先通过对关键字一次哈希至超级节点,再通过二次哈希分配给超级节点所负责组的普通节点。

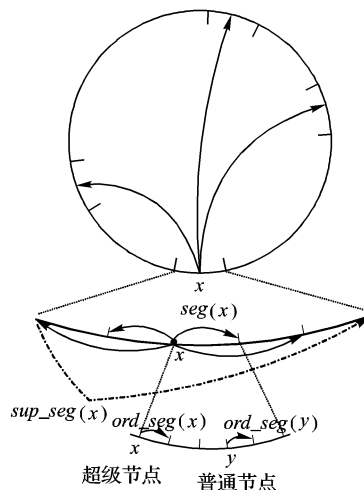


图1 基于超级节点的RP2P网络

因此, x 的路由表中只要存储 $(- \ln c)^{1/d} n^{1/d}$ 个顺序邻居的 $seg(x)$ 、 $(- \ln c)^{1/d} n^{1/d}$ 个随机邻居的 $sup_seg(x)$,其中 n 表示网络中超级节点的数量;以及存储它所负责的 $ord_seg(x)$ 即可。

例如,在图1所示的RP2P网络中,超级节点 x 有4个顺序邻居、3个随机邻居,并维护4个普通节点。

2.2 基于层次划分的RP2P路由算法

当超级节点 x 收到一个查询请求 $request(id)$,假定查询请求中包含了发出请求的节点的地址。查询请求首先在由超级节点构成的第一层网络中以 $RP2P_Routing_TTL(id)$ 算法

路由,找到负责的部分包含 id 的节点 z 。对 id 哈希得到 o_id ,然后查询请求在超级节点所负责的组构成的第二层网络中执行。如果 $o_id \in ord_seg(x)$,则将查询结果返回给请求者;如果 $id \in seg(x)$ 且 $id \in ord_seg(y)$ (其中 $y \in O_x$),则将查询请求转发给普通节点 y 。最终到达 $node(id)$, $node(id)$ 将查询结果返回给请求者。

应用改进后的算法,查询请求仍然能够在常数跳内解决,而且由于在顶层 RP2P 网络中的节点数量远远少于网络中的节点数量,查询效率更高。

2.3 节点的加入和离开

P2P 系统是一个动态系统,任意时刻都会有节点加入和离开。为了保持节点路由表的正确性,当有节点加入或离开时需要对相关节点的路由表进行更新。为了防止超级节点失效的问题,我们采用超级节点备份的方法。在一个超级节点所负责的普通节点中,寻找性能最好的节点,将超级节点的信息备份在这个点上,每隔一定的时间就执行一次备份操作。一旦超级节点失效,就由这个备份的升级成为超级节点,保持查询请求可以正常执行。

2.3.1 节点的加入

在基于超级节点的 RP2P 网络中,当一个节点 x 通过节点 y 加入到网络时,它首先找到它所属的超级节点,然后 x 与该超级节点进行通信,成为其成员。在超级节点所维护的部分内部,相关节点更新其前驱-后继的关系,方式与 Chord 类似。这个操作在超级节点所维护的部分内部执行,对外是不可见的。超级节点更新与之相关的少量路由表。这将使得节点加入频繁的情况下 RP2P 网络的路由并不会受到太大影响,从而提高了系统的效率。

2.3.2 节点的退出

在基于超级节点的 RP2P 网络中,如果一个超级节点所维护的部分内部节点退出系统不会影响外部节点,超级节点就只需维护少量的路由表信息。超级节点的退出可以通过超级节点路由表备份的机制有效地保证了 RP2P 节点频繁退出时系统的稳定性。

只有当节点数量加倍时,备份节点全部升级为超级节点,系统重新为所有的超级节点寻找新的备份的节点。

2.4 性能分析

一个查询请求在基于超级节点的 RP2P 网络中经过的最大跳数是 $(d+1)$,基于层次划分的 RP2P 路由算法对原算法的时间复杂度并没有影响,其时间复杂度仍为 $O(d)$ 。

实质上,基于层次划分的 RP2P 路由算法只是将能力弱的节点维护邻居的责任分担给了能力较强的超级节点。每个超级节点的随机邻居、顺序邻居数为 $2(-\ln c)^{1/d} n^{1/d}$,其中 n 表示网络中超级节点的数量;除此之外,每个超级节点还要维护它所负责的普通节点。一旦确定了网络中每个超级节点所负责的普通节点数,那么这个值将是一个常数。因此其空间复杂度为 $O((-\ln c)^{1/d} n^{1/d})$ 。

3 模拟实验及分析

我们分别模拟了 10000 个节点的 RP2P(3, c) 网络和基于层次划分的 RP2P(3, c) 网络。对于基于层次划分的 RP2P(3, c) 网络,选取平均每个超级节点维护 10 个普通节点来进行模拟实验。

本模拟实验程序用 C++ 语言编写,针对不同的 c 值, $c = 1.0E-1 \sim 1.0E-7$,反复进行实验。模拟了原始 RP2P 算法和改进后的基于层次划分的 RP2P 路由算法,对比系统

不能在 3 跳以内解决查询请求的概率 P_3 ,以对性能进行比较分析。每个节点平均保存着 100 个文件,分别对这些网络进行测试得到各自网络的 P_3 值,对比基于层次划分的 RP2P 网络和原始 RP2P 网络的性能。为得到更为准确的仿真结果,我们对每个值的获取都重复进行 30 次实验。

实验结果如图 1 所示,图中分别显示了原始的 RP2P 算法和改进后的基于层次划分的 RP2P 路由算法下不能在 3 跳以内解决查询请求的概率 P_3 。

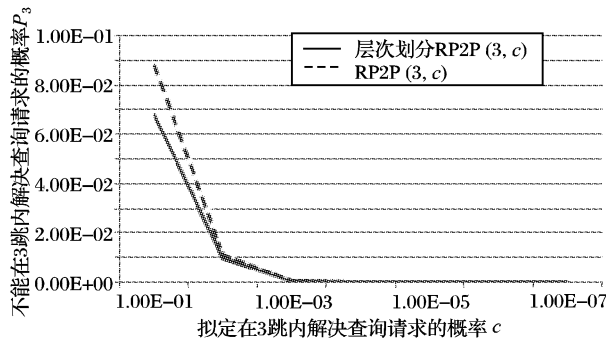


图1 两种算法的模拟实验比较结果

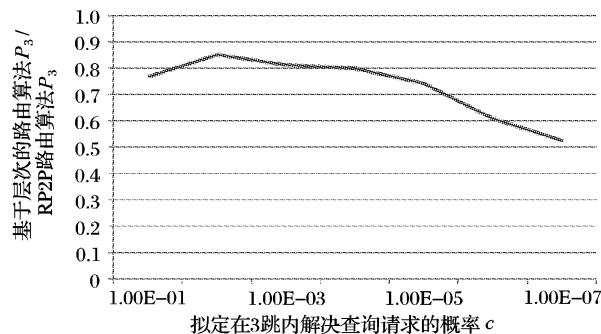


图2 P_3 值的改进结果

在网络规模相同的情况下, P_3 值越小表明查询效率越高。从图 1 可知,当 c 值较大时,基于层次的划分的 RP2P 网络不能在 3 跳内处理查询请求的概率远远小于原始的 RP2P 算法,随着 c 值的增大,这种差别逐渐缩小。但是从改进的程度上来看,如图 2 所示,随着 c 值的增大, P_3 改进的程度会略有上升。当 $c = 1.0E-1$ 时, P_3 大约会减小 30%;当 $c = 1.0E-7$ 时, P_3 大约会减小 50%。这表明随着网络规模的增大,基于层次划分的 RP2P 路由算法比原始的 RP2P 路由算法查询效率更优。

在我们的系统中,每个超级节点维护 10 个普通节点,那么由超级节点形成的 DHT 环比原始的 DHT 环要小得多,我们的路由算法所维护的邻居数大约为原始 RP2P 路由算法的 50%,这必然大降低网络的维护负载,提高系统效率。

4 结语

本文在具有小路由延迟特性的 RP2P 路由算法的基础上,分析了节点性能差异对系统效率产生的影响,提出了基于层次划分的 RP2P 路由算法。实验表明,该算法不但提高了系统的查询效率,也在一定程度上保证了系统的稳定性。

每个超级节点维护多少个普通节点也是影响系统性能的一个重要因素。极端情况下每个超级节点都不维护任何的普通节点,系统性能接近原始的 RP2P 网络;若一个超级节点维护的普通节点过多,超级节点反而会成为系统性能的瓶颈,从而也会带来单点失效或超级节点负载过重等问题。因此,如何分配超级节点维护的普通节点数量,以使每个节点在网络中充分的发挥能力是下一步工作的重点。

(下转第 651 页)

3.3 端到端延时

网络编码能使吞吐量和能耗的性能提高,但付出的代价是端到端的延时和节点存储计算量的增大。由于网络编码是

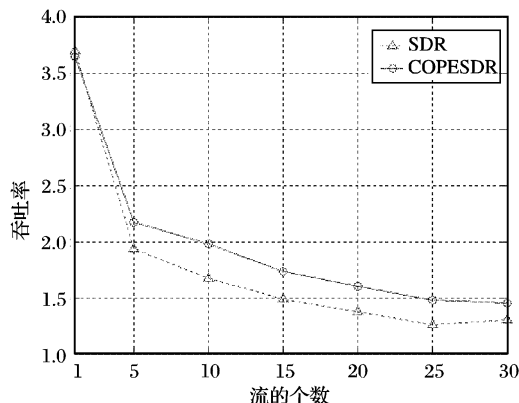


图3 吞吐率与流的个数

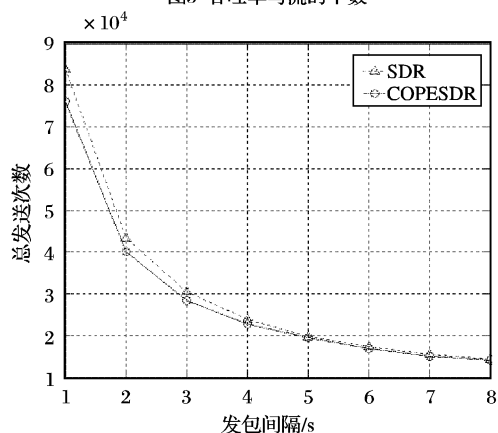


图5 总发送次数与发间隔

两个以上的数据包进行线性或非线性运算,必须等待能编码的包到达同一个节点,这就带来了端到端延时和存储计算量的增大。

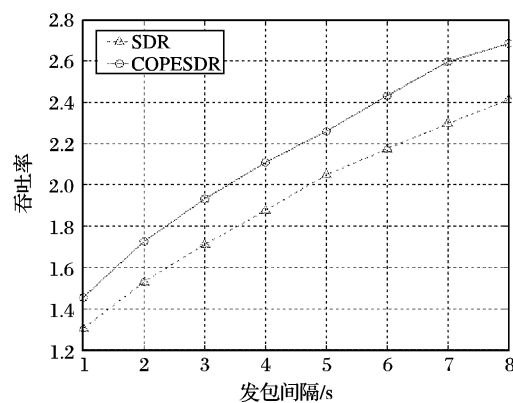


图4 吞吐率与发间隔

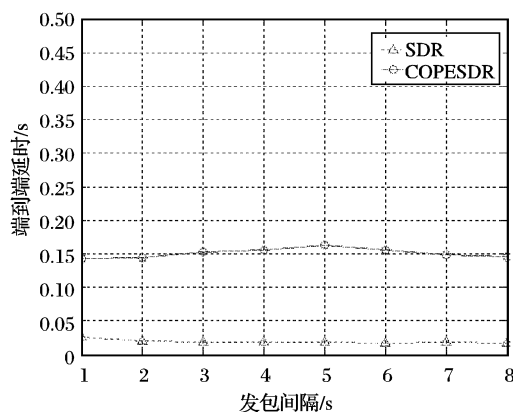


图6 端到端延时与发间隔

4 结语

当前网络编码大多处于理论研究阶段,本文将源定向中继的思想与机会网络编码结合起来,定义了蝶形结构和链状结构,并提出一种新的环形结构,大大地降低了网络编码的实现难度,从而使网络编码在无线环境下的应用成了现实。在NS2平台上的仿真结果表明,进行机会网络编码后,系统的吞吐量获得了较大的提高,同时具有高能量效率。

参考文献:

- [1] AHLSTEDT R, CAI N, LI S-Y R, *et al.* Network information flow [J]. IEEE Transactions on Information Theory, 2000, 46(4): 1204 - 1216.
- [2] KATTI S, KATABI D, HU W-J, *et al.* The importance of being opportunistic: Practical network coding for wireless environments[EB/OL]. [2008-09-01]. <http://www.cl.cam.ac.uk/~wh214/>

research/papers/allerton05. pdf.

- [3] MIL-STD-188-220C[S/OL]. [2008-09-01]. http://www.cnrwg.itsi.disa.mil/docs/MS188_220C_Final_2.pdf.
- [4] KOETTER R, MEDARD M. An algebraic approach to network coding [J]. IEEE/ACM Transactions on Networking, 2003, 11(5): 782 - 795.
- [5] BISWAS S, MORRIS R. Opportunistic routing in multi-hop wireless networks[J]. ACM SIGCOMM Computer Communication Review, 2004, 34(1): 69 - 74.
- [6] BLETSAS A, KHISTI A, REED D P. A simple cooperative diversity method based on network path selection[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(3): 659 - 672.
- [7] LUN D S, RATNAKAR N, KOETTER R, *et al.* Achieving minimum-cost multicast: A decentralized approach based on network code[C]// Proceedings of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Miami, Florida, USA: IEEE, 2005, 3: 1607 - 1617.

(上接第648页)

参考文献:

- [1] ZHAO B, KUBIATOWICZ J, JOSEPH A. UCB/CSD-01-1141, Tapestry: An infrastructure for wide-area location and routing [R]. Berkeley, CA: University of California at Berkeley, Computer Science Department, 2001.
- [2] DRUSCHEL P, BOWSTRON A. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems [C]// Proceedings of 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001). New York: ACM, 2001.
- [3] RATNASAMY S, FRANCIS P, HANDLEY M, *et al.* A scalable content-addressable network [C]// Proceedings of ACM SIG-

COMM'01. New York: ACM, 2001: 161 - 172.

- [4] STOICA I, MORRIS R, KARGER D, *et al.* Chord: A scalable peer-to-peer lookup service for Internet applications [C]// Proceedings of ACM SIGCOMM'01. New York: ACM, 2001: 149 - 160.
- [5] GUPTA A, LISKOV B, RODRIGUES R. Efficient routing for peer-to-peer overlays [C]// Proceedings of the First Symposium on Networked Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2004: 9 - 9.
- [6] CHEN S. Building a scalable P2P network with small routing delay [C]// Proceedings of Asia-Pacific Web Conference, LNCS 4976. Berlin: Springer, 2008: 456 - 467.