

# 改进型自组织网络路由模型研究

王 蓉<sup>1</sup> 陈爱网<sup>1</sup> 杜 炜<sup>2</sup>

(空军工程大学电讯工程学院 西安 710077)<sup>1</sup> (空军驻西安地区军事代表 西安 710082)<sup>2</sup>

**摘 要** 自组织网络和对等网络的研究是下一代网络(NGN)研究领域的热点,而 P2P 技术作为四大技术之一必将改变互联网的未来。通过构建一个基于 P2P 计算模式的 P2P 覆盖网络的虚拟网络拓扑结构,可以有效地建立一个基于自组织网络路由模型的全分散的网络——层次聚集的自组织网络(HASN)。描述了 HASN 的目标和体系结构,同时对分散 P2P 的命名、路由发现和 HASN\_Scale 更新算法做了详细描述。最后仿真结果验证了 HASN 的良好性能。

**关键词** 对等网络,自组织网络,结构化覆盖网,层次路由表

**中图分类号** TP393 **文献标识码** A

## Study of Self-organizing Network Routing Model

WANG Rong<sup>1</sup> CHEN Ai-wang<sup>1</sup> DU Wei<sup>2</sup>

(Telecommunication Engineering Institute, Airforce Engineering University, Xi'an 710077, China)<sup>1</sup>

(Air Force Military Representative Office of Xian Area, Xi'an 710082, China)<sup>2</sup>

**Abstract** The study of self-organizing network is a hotspot in the next generation network (NGN) research domain. P2P as one of the four technologies will shape the Internet's future. By building P2P overlay network on top of internet's physical topology layer based on P2P computing mode, we can effectively build a full-decentralized internet based self-organizing network routing model-Hierarchical Aggregation Self-organizing Network (HASN). The target and architecture of HASN were described in this paper, as well as a detailed description of a P2P decentralized naming, route discovering and updating algorithm-HASN\_Scale. Simulation results testify the performance of HASN.

**Keywords** Peer-to-Peer network, Self-organizing network, Structured overlay network, Hierarchical routing table

## 1 引言

自组织网络是下一代网络(NGN)研究领域的热点<sup>[1,2]</sup>。自我组织过程亦即一个给定的所有节点可以自动建立一个网络,无需人工操作。这一过程可能在网络初始化期间或网络拓扑发生变化后发生。这意味着当网络拓扑结构发生变化后所有的节点都可以迅速地重组自己的网络。同时,对等网络是当前网络研究领域新兴的一种技术<sup>[3]</sup>。P2P 系统已经发展成为互联网最重要的应用系统之一。现在, P2P 系统几乎消耗了大量的互联网的带宽,它产生的网络流量已经超过了 HTTP 协议。

如何实现 P2P 技术到自组织网络技术,是我们研究的主要对象。在我们研究的过程中发现,自组织网络和 P2P 网络具有很多相似之处。在自我组织网络中,每个节点充当一个路由器,同时也是一台普通的主机,不存在网络管理中心这样的角色。每个节点都是相互对等的,路由协议通常是完全分散的,以保持网络的强大和灵活。所有的这些特征仅仅是 P2P 技术的主要特征。在我们正在进行的的研究中,我们发现,通过建立一个基于互联网物理基础设施上的被命名为 P2P 覆盖网络的虚拟网络拓扑,便可基于自组织网络有效地建立一个全分散的互联网。

目前, P2P 网络模型已被广泛使用,例如: Napster<sup>[4]</sup>, Freenet<sup>[5]</sup> 和 Gnutella<sup>[6]</sup>。然而,它们中的每一个都有一些公开问题。Napster 使用集中服务器管理目录,而 Napster 的性能被那些中央服务器限制,这也使得它很容易发生单点故障;而像 Gnutella 和 Freenet 的分散使用模型通过溢出机制发送查询消息,由于其没有集中的服务器,而导致溢出问题,并增加网络负载。因此,结构覆盖网络路由协议,如 CAN<sup>[7]</sup>, Chord<sup>[8]</sup>, Pastry<sup>[9]</sup> 和 Tapestry<sup>[10]</sup> 则建议提供一种机制,以提高网络的可扩展性。所有这些方法有一个共同点,它们都通过一个哈希函数分配数据和节点的键,并用这些键来存储和维护数据。但这些算法没有考虑到物理网络的拓扑结构,从而导致节点由于哈希函数的结果不得不长途跋涉,取得一个邻里节点的数据。这严重降低了路由效率。

本文提供了一个层次聚集自组织网络(HASN),其有效地结合了基于 P2P 路由技术和自组织网络路由技术的结构化覆盖网络。HASN 模型是一种基于互联网基础设施的自组织网络路由模型。目的是为广域网建立一个有效负载均衡的和动态自组织网络模型。这个模型位于应用层,通过在互联网的物理拓扑结构顶部建立 P2P 覆盖网络和在它上面执行 P2P 路由算法,我们可以有效地建立一个基于自组织网络路由模型的全分散的网络。

本文受空军工程大学研究生创新基金项目(2011343)资助。

王 蓉(1977-),女,博士生,讲师,主要研究方向为自组织网络、信息安全。



### 3.1 节点命名机制

通过采用类似 Chord 的一致哈希算法, HASN 将分配给每个节点和键  $1m$ -bit 的标识符(NID)( $N \leq 2^m$ )。而 NID 是通过选择节点的 IP 地址来选择的, 关键标识符(NID)是通过选择键来产生。NID 可用于指定一个哈希环的节点的位置。当一个节点首次加入 HASN\_Scale, 通过哈希算法将自动分配给它一个范围从 0 到  $(2^m - 1)$  的 NID。每个  $m$ -bit 长 NID 值的计算如下:

$$\text{Valueof}(NID) = \sum_{i=1}^m NID_i \times 2^i \quad (1)$$

我们引进全球环区 (GRZ) 和个体环带 (IRZ) 的概念到 HASN\_Scale。GRZ 对应于整个  $[0, 2^m - 1]$  范围的数据。当一个新的节点加入系统时, 它要根据它的 NID 划分 GRS。节点的 IRZ 对应于分配给它的私人哈希环区。IRZ 可以被描述为 (NID\_Predecessor, NID), 其中 NID\_Predecessor 代表了当前节点的前身。

引进 GRZ 和 IRZ 的目的是为 HASN\_Scale 提供一个动态节点命名的机制, 而不是一个静态的。由于每个节点的 IRZ 的范围随着网络中节点的数目是可以改变的, 当一致性哈希算法足够对称并且网络状态稳定时, 一个节点可以估算出网络中的节点总数。另外, 只通过比较查询节点的 IRZ 与 KID 来确定节点是否为目标节点可以简化发现程序。

对于一个具有  $N$  个节点的网络, 每个节点 IRZ 满足式 (2):

$$\begin{aligned} IRZ_1 \cup IRZ_2 \cup IRZ_3 \dots \cup IRZ_N &= GRZ \\ IRZ_i \cap IRZ_j &= \varnothing, i \neq j, 1 \leq i, j \leq N \end{aligned} \quad (2)$$

### 3.2 节点的状态

在 HASN\_Scale 中每个节点保持  $H$  层次路由表 (HRT), 其中  $H$  是 NTL 的高度。如果在集群  $i$  中有  $n$  个节点, 集群  $i$  中的每个节点将保持  $O(\log N)$  个条目的级别为  $i$  的 HRT。HRT 是对 Chord 指针表的改善。不过, 由于这两种模式之间节点的命名机制和网络体系结构不同, HRT 的结构与指针表也有很大的不同。在一个  $m$ -bit 的空间中, 以下概念与节点的 HRT 相关。

- 当前集群水平 (CCL): 在当前 HRT 中一个节点所属的 NTL 的集群水平。

- 前身: 从当前节点起哈希环逆时针方向的第一个节点;
- 继任: 从当前节点起哈希环顺时针方向的第一个节点;
- HRTEntry[K].start: 涵盖了  $K_m$  HRT 项的环的变化范围的起始位置。

$$\text{HRTEntry}[K].start = (n + 2^{k-1}) \bmod 2^m \quad 1 \leq k \leq m \quad (3)$$

- 下一跳环区间 (NRI): 涵盖了  $K_m$  HRT 项的环的范围。

$$\text{NIR}(k) = [\text{HRTEntry}[K].start, \text{HRTEntry}[K+1].start), 1 \leq k \leq m \quad (4)$$

- 下一跳 IRZ (NIRZ): 在 NRI 顺时针方向的第一个活动节点  $G$  的 IRZ。

$$\text{HRTEntry}[K].start \in \text{IRZ}(G), 1 \leq k \leq m \quad (5)$$

HRT 的内容涉及了 CCL, HRTEntry[k].start 和 NIRZ,  $K$  为 HRT 的条目序号。HRT 的结构描述如图 4 所示。

Current Cluster Level (CCL)	HRTEntry[K].start	Next Hop IRZ (NIRZ)

图 4 HRT 结构

假设网络中都有两个层次集群结构, 表 1 和表 2 描述了节点  $B$  的 HRT, 即  $H = 1$ 。从表格中我们可以得出这样的结论: 节点  $A, B$  和  $D$  在  $CL_0$  中, 节点的  $A, B, C$  和  $D$  在  $CL_1$ , 而  $CL_1$  是完整的集群。

表 1 节点  $B$  的  $CL_0$

K	CCL	HRTEntry[K].start	NIRZ
1	1	100	$C = (011, 100]$
2	1	101	$D = (100, 000]$
3	1	111	$D = (100, 000]$

表 2 节点  $B$  的  $CL_1$

K	CCL	HRTEntry[K].start	NIRZ
1	0	100	$D = (011, 000]$
2	0	101	$D = (011, 000]$
3	0	111	$D = (011, 000]$

### 3.3 HASN\_Scale 路由发现算法

根据 HASN\_Scale 的节点命名机制, 为了建立 GRZ, 网络中每个节点都可以把它们 IRZ 与其他节点的相结合。通过持有继任者的地址信息, 每个节点都可以通过向它们的继任者传输信息来和其他节点通信, 然后和它们的继任者的继任者通信等等。虽然这个传递消息的方法非常简单, 但是它的效率非常低。消息可能需要经过  $O(N)$  跳才能从一个节点到达目标节点 ( $N$  为 HASN 中节点的数目)。为了解决这个问题, 提高 HASN\_Scale 路由发现效率, 根据路由发现算法我们为每个节点引入了 HRT 及它之后的 HRT。新的路由发现算法也是对 Chord 的改善, 主要区别在于我们已经向 HASN 引入多层次集群水平。

HASN\_Scale 路由发现算法可描述如下:

- 1) 当  $A$  启动了一个路由发现进程 (或  $A$  接收到一个传输的路由发现消息), 它将首先提取信息的查询 KID 并将它与其 IRZ 相比。如果 KID 属于 IRZ, 就构建一个查询成功的消息并返回到查询发起节点, 路由发现过程到此结束。否则进入第 2) 步。

- 2) 节点  $A$  根据其集群的水平在 HRT 中搜索 KID。在每一个 HRT 中,  $A$  会把 KID 与 HRT 的目录相比较并寻找与 KID 最接近的 NIRZ, 然后在相应的 HRT 目录中向下一跳节点传送信息。通过这种方法, 消息最终会传到 NID 最接近 KID 的节点。如果找到了目标节点, 就构造一个查询成功的消息并返回到查询发起节点, 路由发现过程就到此结束。如果在当前层次的 HRT 中查询过程失败, 发现过程将进入一个更高层次的 HRT。这个过程将一直进行, 直到找到目标节点或达到最高层次的 HRT。

- 3) 如果发现进程已经达到最高层次的 HRT 并且目标节点仍然没有找到, 路由发现过程就在这里结束并传回一个失败的消息。

### 3.4 节点加入和离开

对于一个自组织网络路由算法, HASN\_Scale 必须保持节点的动态加入和出发。当一个新的节点加入网络, HASN\_Scale 就会根据它的 NID 通过划分成员节点的 IRZ 的方法来分配给它一个 IRZ。当一个节点正常离开网络时, HASN\_Scale 就会通过把它与成员节点的 IRZ 相结合的方法回收它的 IRZ。

节点加入的过程描述如下:

- 1) 一个新的节点  $M$  首先根据 NTL 集群分组协议选择

它归属的 HA。

2)  $M$  节点选择一个成员节点  $H$  作为它的引导节点。在  $HASN\_Scale$  中,  $M$  的默认引导节点是它归属的 HA。然后在引导节点的帮助下  $M$  将寻找它的后继  $S$ 。然后根据  $M$  的 NID,  $S$  的 IRZ 将被分为两部分 (NID( $S$  的前驱), NID( $S$ ))。  $S$  将持有 (NID( $M$ ), NID( $S$ )) 的一部分, 而  $M$  将获得 (NID( $S$  的前驱), NID( $M$ )) 的另一部分作为它的 IRZ。

3) 在引导节点的帮助下  $M$  将寻找它的前驱  $P$ 。然后  $M$  就向  $P$  和  $S$  发送一个加入请求消息,  $P$  和  $S$  收到消息后, 它们就会更新自己的后继和前驱, 其中  $P$  将作为  $M$  的新的后继,  $S$  作为  $M$  的新的前驱。直到现在, 哈希环的完整性得到了保证。

4)  $M$  在它的后继  $S$  的帮助下, 根据 HRT 的每一条目通过寻找下一跳节点来初始化它的 HRT。

5) 根据  $M$  分配到的 IRZ, 相关的 KID 将会从  $S$  转移到新的节点  $M$ 。

6) 路由更新过程将被触发去更新网络中其他节点的 HRT (路由更新的详细过程将在下一部分讨论)。

节点离开的过程描述如下:

1) 节点  $Q$  将首先向它的前驱  $P$  和后继  $S$  发出离开的消息。  $P$  和  $S$  收到消息后, 它们将更新自己的后继和前驱, 其中  $P$  将更新  $Q$  的后继作为它的后继, 更新  $Q$  的前驱作为它的前驱。直到现在, 哈希环的完整性得到保证。

2)  $S$  将通过比较它的 IRZ (NID( $Q$ ), NID( $S$ )) 与  $Q$  的 IRZ (NID( $P$ ), NID( $Q$ )) 把它的 IRZ 更新到 (NID( $P$ ), NID( $S$ ))。

3) 根据  $S$  的 IRZ 的更新, 属于  $Q$  相关的 KID 将被转移到  $S$ 。

4) 路由更新过程  $B$  将被触发去更新网络中的其他节点的 HRT。

### 3.5 路由更新机制

对于像  $HASN\_Scale$  这样的自组织路由算法, 路由更新机制是由网络本身触发的自恢复机制。路由更新机制的主要功能是当网络拓扑结构变化并转变为收敛的网络时更新网络中所有节点的 HTR。当前普遍使用的路由更新机制可以概括为 3 种类型: 定期触发, 事件触发路由更新机制和前两个机制的结合。对于一个大型的网络, 在不断变化的网络拓扑中采取定期触发更新机制去更新所有节点的 HRT 的方法, 将会消耗巨大成本 (如带宽, CPU 时间和功率), 尤其是在高度动态网络环境中。虽然这种机制保证了路由维护的成本与 HRT 的规模成正比, 但只有当网络是一个小型网络时它才是可行的。当网络是一个大型网络时, 为了保持网络收敛, 它的成本将会倍增。例如, 为了匹配一个有 1000 个目录要更新的路由表, 该节点不得不每秒发送 200 个探测信息, 但是探测过程每隔 50s 才发生一次。由于自组织网络的拓扑结构频繁变化,  $HASN\_Scale$  决定采取事件触发的路由更新机制。

一般来说, 在  $HASN\_Scale$  中有两种类型的触发事件更新路由。当一个节点正常加入和退出系统 (例如, 用户关闭了 P2P 应用程序) 时, 该节点将在网络中向相关节点发送一个离开的消息。这个离开的消息就将充当触发事件。但是, 当一个节点意外退出系统 (例如, 计算机已关闭或崩溃) 时,  $HASN\_Scale$  将不得不采取发现及广播机制来通知网络中的其他节

点, 就像在一跳覆盖网络中所采取的机制一样<sup>[11]</sup>。而在这时, 广播信息将充当触发事件。为了处理意外节点的退出,  $HASN\_Scale$  要求系统中的每个节点应定期向它的前驱发送存活消息, 以发现它的前驱是否还活着。一旦发现前驱偶然退出, 它的后继将启动路由更新过程。

由新节点加入网络引起路由更新过程可以描述如下:

1) 路由更新过程是由事件触发并由新的节点  $M$  开启的。路由更新过程以检索的方式进行。检索按反时针方向工作并且检索的步骤为  $2^{i-1}$ ,  $i=1, 2$ 。在每一步, 路由更新过程将比较和更新 HRT 的第  $i_n$  项。

2) 在检索的第  $i_n$  步, 路由更新过程将把新节点的 NID 与 HRT 中检索节点的第  $i_n$  个目录的下一跳的 NID 相比较。如果前者较小, HRT 中第  $i_n$  个目录的下一跳节点将被新的节点替换。然后路由更新过程将在哈希逆时针方向环中继续进行, 直到它遇到 HRT 中的第  $i_n$  个目录优于  $n$ 。

3) 在此之后, 路由更新过程将继续检索第  $(i+1)$  步的节点, 直到  $i$  的值达到  $m$ 。

由节点离开引起的路由更新过程与新的节点加入网络引起的过程相似, 但有两个关键的不同:

- 路由更新过程是由离开节点或离开节点的后继所引起的。

- 路由更新过程也可以以检索的方式进行。但在检索的每一个步, 路由更新过程将只是比较离开节点的 NID 是否与检索条目的第  $i_n$  个节点的下一跳节点的 HRT 相等。如果它们是相等的, HRT 中第  $i_n$  个目录的下一跳节点将被离开节点的后继节点代替。

## 4 实验评价

### 4.1 系统环境

在英特尔 P4、CPU 的主频为 1.6G 和 Linux RedHat 8.0 环境下, 我们利用网络模拟器 NS2 模拟实验。通过 OTCL 和 C++ 编程, 我们分别实现了  $HASN$  和 Chord 的路由算法。GT-ITM 的拓扑生成器用来生成一个模拟的网络拓扑结构, 在该拓扑结构中过境存根 (TS) 模型能够对网络的实际状况做出很好的描述。在此同时, 我们通过调整参数值和交通拥堵来模拟网络中的动态干扰流动。图 5 描述了用 GT-ITM 生成的网络拓扑图。

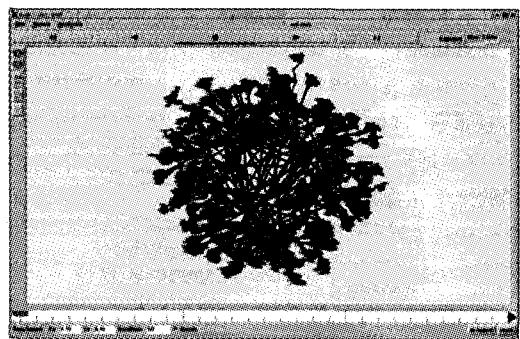


图 5 网络拓扑图

### 4.2 基本性能评价

1) 消息传输的平均时延 (AMTD)

消息传输平均时延 (AMTD) 是评估路由算法性能的一个

(下转第 158 页)

TECTION ALGORITHMS [EB/OL]. [http://www.ross.net/crc/download/crc\\_v3.txt](http://www.ross.net/crc/download/crc_v3.txt), 2010-04

- [4] 毕占坤,张羿猛,黄芝平,等. 基于逻辑设计的高速 CRC 并行算法研究及其 FPGA 实现[J]. 兵工学报, 2007, 28(12): 2243-2249
- [5] 刘畅,章建雄,王玉艳. RapidIO 控制器的 CRC 模块设计[J]. 计算机工程, 2011, 37(15): 238-239
- [6] Campobello G, Patane G, Russo M. Parallel CRC realization [J]. IEEE Transactions on Computers, 2003, 52(10): 1312
- [7] Peterson W W, Brown D T. Cyclic Codes for Error Detection

[J]. Proceedings of the IRE, 1961, 49(1): 228-235

- [8] Wikimedia Foundation, Inc. Linear feedback shift register [EB/OL]. [http://en.wikipedia.org/wiki/Linear\\_feedback\\_shift\\_register](http://en.wikipedia.org/wiki/Linear_feedback_shift_register), 2011-06-30
- [9] Shieh M D, Sheu M H, Chen C H H. A Systematic Approach for Parallel CRC Computations [J]. Journal of Information Science and Engineering, 2001, 17: 445-461
- [10] 阳璞琼,何怡刚,谭阳红,等. 超高频 RFID 系统 CRC 电路设计 [J]. 电路与系统学报, 2009, 14(2): 18-21

(上接第 135 页)

重要指标。图 6 描绘了在 Chord, HASN, 以及优化的 HASN 中, AMTD 与  $N$ (节点数)的关系。我们可以看到,虽然每个路由算法中的 AMTD 演示了与  $N$  的一种  $O(\log N)$  关系,但是 HASN 和优化的 HASN 的性能比 Chord 的好。这是因为 HASN 把临近网络的性质考虑在内。

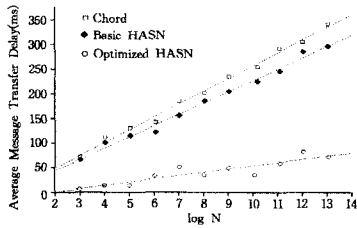


图 6 AMTD 与节点数  $N$  的关系

### 2) 节点加入时发送消息的数目

图 7 示出了节点加入时发送消息的数目与  $N$  的关系。

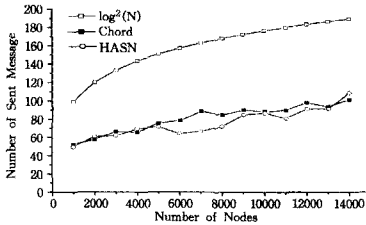


图 7 节点加入时发送消息的数目与  $N$  的关系

### 3) 节点离开时发送消息的数目

节点离开时发送消息的数目,也是评价一个自组织网络性能的重要指标。图 8 描绘了 Chord 和 HASN 中节点离开时发送消息的数目与  $N$  的关系。我们可以看到,Chord 和 HASN 在用路由算法演示  $N$  与  $O(\log^2 N)$  的关系时,彼此的性能是相似的。

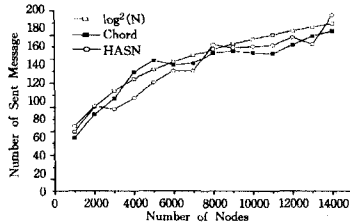


图 8 节点离开时发送消息的数目与  $N$  的关系

**结束语** 在自组织网络研究领域,移动特设网络(MANET)始终是一个热点,吸引了许多科学家的关注。与此相反,很少有人关注基于自组织网络的网络领域。HASN 是一个基于互联网基础架构的自组织网络路由模型。基于路由算法的 DHT,它继承了结构化 P2P 覆盖网络的所有优点并保

证了网络的可扩展性。通过引入集聚分组机制和层次结构, HASN 充分利用互联网的网络邻近特性和网络局部特性改善了路由性能。

进一步的研究将侧重于把 P2P 计算模式延伸到 MANET。我们的目标是通过路由协议在一个逻辑上的命名空间进行操作和通过 MANET 路由协议在物理上的名称空间进行操作,从而提出一种新的多跳路由协议以传统的形式进行无缝集成。

## 参考文献

- [1] Èapkun S, Buttyán L, Hubaux J P. Self-organized public-key management for ad hoc networks[J]. IEEE Transactions on Mobile Computing, 2003, 2(1)
- [2] Perkins C, Belding-Royer E M, Das S R. Ad hoc on-demand distance vector (AODV) Routing[Z]. Network Working Group RFC 3561, July 2003
- [3] Parameswaran M, Susarla A. P2P Networking: An Information-Sharing Alternative[R]. IEEE Computing Practices, July 2001
- [4] C—NET NEWS. Napster among fastest-growing Net technologies [EB/OL]. <http://news.com.com/2100-1023-246648.html>. October 2000
- [5] Clarke I, Sandberg O, Wiley B, et al. Freenet: A distributed anonymous information storage and retrieval system [R]. Workshop on Design Issues in Anonymity and Unobservability, July 2000
- [6] Gnutella [EB/OL]. <http://gnutella.wego.com>
- [7] Ratnasamy S, Francis P, Handley M, et al. A Scalable Content-Addressable Network[C]// Annual conference of the Special Interest Group on Data Communication (SIGCOMM 2001). August 2001
- [8] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for Internet applications[C]// Annual conference of the Special Interest Group on Data Communication (SIGCOMM 2001). August 2001
- [9] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems[C]// International Conference on Distributed Systems Platforms (Middleware 2001). November 2001
- [10] Zhao Ben, Kubiatowicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing[R]. UCB/CSD-01-1141. Computer Science Division, U. C. Berkeley, April 2001
- [11] Jain S, Mahajan R, Wetherall D. A Study of the Performance Potential of DHT-based Overlays[C]// 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003). March 2003