

# 基于 Chord 的动态多路由算法

陈沅涛, 吴海珍

(长沙理工大学计算机与通信工程学院, 长沙 410076)

**摘要:** 根据 Chord 路由协议中每一跳路由距离之间具有的聚类特性, 在不改变节点指针表的前提下, 提出一种基于 Chord 的动态多路由算法。结果证明, 当路由源节点和路由目的节点确定且 Chord 处于稳定状态时, 该算法能使节点转发消息引起的负载分布更均匀, 并提高消息路由的成功率。

**关键词:** 指针表; 跳数; 动态多路由; 路由协议

## Dynamic Multi-routing Algorithm Based on Chord

CHEN Yuan-tao, WU Hai-zhen

(College of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha 410076)

**【Abstract】** According to the clustering property of routing distance each hop in Chord routing protocol, this paper proposes a dynamic multi-routing algorithm based on Chord without modifying each pointer table of each peer. Result proves that this algorithm can make load more balancing caused by node forwarding message, and raises the message route's success ratio when the route source node and the route goal node have determined, and Chord is at the steady state in the situation.

**【Key words】** pointer table; hop; dynamic multi-routing; routing protocol

由于 Chord 协议中的每个节点只须维护少量相关节点的路由信息, 因此可以高效地对消息进行路由。所有的节点在 DHT 中的位置分布比较均匀, 当节点加入或离开时仅有少量节点的路由信息需要更新<sup>[1]</sup>。但是 Chord 协议的路由算法是固定的, 当路由源节点和路由目的节点确定且 Chord 处于稳定状态时(即没有节点加入或离开), 从路由源节点到路由目的节点之间的路由便确定了<sup>[2]</sup>。针对 Chord 协议, 提出一种基于 Chord 的路由改进算法。

### 1 问题描述

**定理 1** 假设在某条路由路径中, 从节点  $U$  到节点  $W$  为第  $i$  跳, 从节点  $W$  到节点  $Q$  为第  $i+1$  跳, 则有  $|W-Q| < |U-W|$ 。

证明:

$$|U-W| = |\text{successor}((U+2^{a-1}) \bmod 2^m) - U| = 2^{a-1} + \alpha (\alpha \in [1, m] \text{ 且 } \alpha \in \mathbb{Z})$$

从节点  $W$  到节点  $Q$  的路由距离为

$$|W-Q| = |\text{successor}((W+2^{b-1}) \bmod 2^m) - W| = 2^{b-1} + \beta (\beta \in [1, m] \text{ 且 } \beta \in \mathbb{Z})$$

(1) 当  $a=m$  时,  $|U-W| = 2^{m-1}$ , 由于  $|U-W| + |W-Q| = 2^{m-1} + 2^{b-1} + \beta > 2^{m-1}$ , 因此  $|W-Q| < 2^{m-1}$ , 此时  $|W-Q| < |U-W|$ 。

(2) 当  $1 \leq a < m-1$  时, 节点  $Q$  落在  $[(U+2^{a-1}) \bmod 2^m, (U+2^{m-1}) \bmod 2^m)$  范围之内。假设  $|W-Q| \geq |U-W|$ , 即  $|U-Q| \geq 2|U-W|$ , 可以得出  $2^{a-1} + \alpha + 2^{b-1} + \beta \geq 2(2^{a-1} + \alpha) = |\text{successor}((U+2^a) \bmod 2^m) - U|$ , 则节点  $U$  的第  $a$  个指针表所指的路由更加接近节点  $Q$ , 与 Chord 路由协议相矛盾, 假设不成立, 所以,  $|W-Q| < |U-W|$ 。

定理 1 成立, 证毕。

**定理 2** 在满环 Chord 中, 当从路由源节点  $U$  到路由目的节点  $W$  的路由跳数为  $h$  时, 在路由跳数不变的情况下, 总共

有  $h!$  条从节点  $U$  到节点  $W$  的路由路径。

证明: 在满环情况下, 根据 Chord 路由协议, 假设从路由源节点  $U$  到路由目的节点  $W$  路由的每一跳的路由距离分别为  $x_1, x_2, \dots, x_h$ , 其中,  $x_i (i \in [1, h])$  且  $i \in \mathbb{Z}$  可以表示为  $2^{a-1}$  的形式作为任意节点指针表中的路由距离。假设有  $x_1 > x_2 > \dots > x_h$ , 对路由距离  $x_1, x_2, \dots, x_h$  的顺序任意进行排列具有  $h!$  种组合, 其距离总长度保持不变, 可以对应  $h!$  条从节点  $U$  到节点  $W$  路由, 证毕。

**定理 3** 在非满环 Chord 中, 当从路由源节点  $U$  不能直接到路由目的节点  $W$  时(即节点  $U$  的指针表中没有直接指向节点  $W$  的路由信息), 从节点  $U$  到节点  $W$  的路由跳数是不确定的。

证明: 在非满环情况下, 根据 Chord 路由协议, 假设从路由源节点  $U$  到路由目的节点  $W$  的每一跳的路由距离分别为  $x_1, x_2, \dots, x_i$ , 其中  $x_i$  落在  $[2^{a-1}, 2^m - 1]$  范围内, 当  $\sum_{i=1}^h x_i = |U-W|$  时,  $x_1, x_2, \dots, x_h$  为每一跳路由的增量,  $h$  为路由跳数, 由于  $x_i$  取值范围的不确定, 因此  $h$  不确定。由此可知, 从节点  $U$  到节点  $W$  的路由跳数是不确定的, 证毕。

### 2 2 种动态多路由算法

#### 2.1 满环动态多路由算法的路由协议

当路由源节点  $U$  和路由目的节点  $W$  固定且 Chord 处于稳

**基金项目:** 湖南省自然科学基金资助项目(06JJ50109); 湖南省科技计划基金资助项目(06FJ3161); 湖南省教育厅科学研究基金资助项目(07C081)

**作者简介:** 陈沅涛(1980 -), 男, 讲师、硕士, 主研方向: 计算机网络与通信, P2P 计算; 吴海珍, 讲师、硕士

**收稿日期:** 2009-06-11 **E-mail:** yufeng8552@yeah.net

定状态时,为了在不增加跳数的情况下克服 Chord 协议中负载均衡和路由失败率的问题,在定理 2 的基础上,满环动态多路由算法的路由协议描述如下:

路由源节点  $U$  计算从自己出发到路由目的节点  $W$  每一跳的路由距离  $x_1, x_2, \dots, x_h$ , 其中,  $x_i \neq x_j (i \neq j \text{ 且 } i, j \in [1, h])$ ,  $\sum_{i=1}^h x_i = |U-W|$ 。节点  $U$  对路由距离所组成的向量  $x = [x_1, x_2, \dots, x_h]$  有一个对应的评价值向量  $e = [e_1, e_2, \dots, e_h]$ , 初始化值都为 1, 设  $e_{\max}$  和  $e_{\min}$  分别为向量  $e$  中的最大和最小的 2 个元素, 将  $e$  归一化到  $[0, 1]$  的范围内, 归一化后的第  $i$  个元素  $E_i$  为

$$E_i = \begin{cases} \frac{e_i - e_{\min}}{e_{\max} - e_{\min}}, & e_{\max} \neq e_{\min} \\ 1, & e_{\max} = e_{\min} \end{cases} \quad (1)$$

节点  $U$  以每次概率  $P_i$  选择第  $i$  个路由距离进行第 1 跳路由:  
 $P_i = E_i / \sum_{i=1}^h E_i$  (2)

对于中间路由节点, 则依次在剩下的路由距离中随机选择一个路由距离进行路由。当从节点  $U$  到节点的某次路由成功时, 节点  $U$  将此次选择的路由距离对应的评价值加 1, 否则将该评价值减 1。

## 2.2 非满环动态多路由算法的路由协议

在非满环情况下, 根据定理 3 中路由跳数的不确定性, 路由源节点无法实现计算从路由源节点  $U$  到路由目的节点  $W$  之间每一跳的路由距离, 借鉴满环动态多路由算法的路由协议, 非满环动态多路由算法的路由协议描述如下:

设路由源节点  $U$  删除指针表中长度相同及长度大于  $|U-W|$  的路由距离后剩下的路由距离组成的向量为  $z = [z_1, z_2, \dots, z_h]$ , 节点  $U$  对向量  $z$  中的路由距离有一个对应的评价值向量  $e = [e_1, e_2, \dots, e_h]$ , 初始化值为 1, 采用式(1)对  $e$  进行归一化, 设归一化后的向量为  $E = [E_1, E_2, \dots, E_h]$ , 节点  $U$  以式(2)计算得到的概率选择第  $i$  个路由距离进行第 1 跳路由。对于中间路由节点, 根据 Chord 协议选择顺时针方向距离节点  $W$  最近的路由距离。

在路由源节点  $U$  和路由目的节点  $W$  固定且 Chord 协议处于稳定状态时, 非满环动态多路由算法的路由协议具有负载均衡和降低路由失败率的特性。

## 3 性能指标分析

为了分析动态多路由算法的路由协议的路由跳数、负载均衡, 以及路由成功率相对于 Chord 路由协议的性能<sup>[3]</sup>, 在 Windows Server 2003 操作系统上, 基于 Bamboo<sup>[4]</sup> 使用 Java 实现了 Chord 路由协议和动态多路由算法的路由协议, 所有实验均基于步进行。

### 3.1 路由跳数

每步随机选择 2 个不同的在线节点分别作为路由源节点和路由目的节点, 然后路由源节点分别采用 Chord 路由协议和动态多路由算法的路由协议向路由目的节点发送一条消息, 最后对每次路由的跳数进行统计。

Chord 路由协议中 ID 的位数  $m$  为 7~11, 非满环时的节点数目为满环时的 70%, 在 100 步之后计算路由总跳数。满环动态多路由算法的跳数与满环 Chord 的跳数相同, 非满环动态多路由算法的跳数相对于非满环 Chord 的跳数增加不到 11%。当  $m=10$ , 非满环时的节点数目占满环时节点数目的 10%~90%, 非满环动态多路由算法的跳数相对于非满环

Chord 的跳数增加不到 11%。

### 3.2 负载均衡

为了比较动态多路由算法相对于 Chord 在路由源节点和路由目的节点确定且 Chord 处于稳定状态时负载均衡的性能, 在  $m=10$  的情况下, 选择 2 个路由距离最远的 ID 为 0 的路由源节点和 ID 为 1 022 的路由目的节点, 分别在满环和非满环(占满环节点数目的 70%)时节点 0 向节点 1 022 发送 1 000 条消息, 然后统计每个节点(不包括节点 0 和节点 1 022)需要转发的消息数量。

在实验中可以发现, 在满环 Chord 中, 节点 0~节点 1 022 中间的 7 个中间路由节点分别需要转发 1 000 条消息, 而在满环动态多路由算法中, 节点 0 向节点 1 022 发送的 1 000 条消息分别由大约一半的节点 ( $2^{m-1}$ ) 承担, 每个节点转发的消息数量基本保持在 30 条以下, 除了个别少量的节点需要转发 100 条左右的消息。在非满环 Chord 中, 从节点 0 到节点 1 022 中间的 6 个中间路由节点分别需要承担 1 000 条消息的负荷, 而在非满环动态多路由算法中, 尽管负载均衡性能没有满环动态多路由算法的优越, 但是较非满环 Chord 其有更多的节点去承担消息量。

### 3.3 路由成功率

假设在  $m=10$  的 Chord 中有  $p\%$  的恶意节点, 每个恶意节点以  $q\%$  的概率拒绝提供消息转发服务。每步中随机选择一个路由源节点和路由目的节点, 然后路由源节点向路由目的节点发送 200 条消息, 记录所有路由的最后一条消息成功与失败的次数, 经过 1 000 步之后对路由成功率进行统计。

在实验中可以发现, 满环动态多路由算法具有较高成功率, 当有 60% 的恶意节点拒绝转发消息时, 路由成功率大于 0.5, 而相同条件下满环 Chord 的路由成功率不到 0.05。非满环动态多路由算法的路由成功率也高于非满环 Chord。满环动态多路由算法的路由成功率保持在 0.85 以上, 当  $q > 0.7$  时, 非满环动态多路由算法的路由成功率高于 0.4, 而满环和非满环的路由成功率低于 0.3。因此, 本文算法明显提高了路由成功率。

## 4 仿真结果分析

### 4.1 平均查询延迟时间

比较 Chord 协议与本文算法协议的平均查询延迟时间的差异。平均延迟时间定义为考虑到每跳的平均物理网络延迟的资源定位时间。平均查询延迟时间比较如图 1 所示。

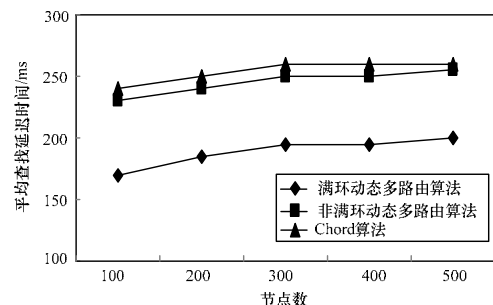


图 1 平均查询延迟时间比较

由图 1 可以看出, 满环动态多路由算法的路由协议能在充分实现查询的情况下更有效地降低查询延迟时间。

### 4.2 平均查询路径长度

在相同的模拟实验环境下, 比较平均查询路径长度的差 (下转第 102 页)