一种快速构建 CAN 网络拓扑算法

戴长华, 张 昊
(国防科学技术大学 信息系统与管理学院,长沙 410073)

摘 要: 受二叉树思想的启发,在 P2P 网络拓扑管理协议 T-Man 和 Kademlia 网络快速构建算法的基础上,提出了从非结构化 P2P 网络快速构建 CAN 网络的算法。Kademlia 网络为二叉树拓扑结构,CAN 网络基于空间划分,由于已经提出了 Kademlia 网络快速构建算法,通过把 CAN 的空间划分方式强制定义为树图的空间划分方式,研究问题转换为由 Kademlia 网络的二叉树结构向 CAN 网络的树图结构转换及构建相应路由表的问题。实验表明,该算法能在对数的时间内构建出 CAN 网络。

关键词: 对等网络; 控制器局域网; 拓扑管理
中图分类号: TP393.01 **文献标志码**: A **文章编号**: 1001-3695(2010)03-1154-03
doi:10.3969/j.issn.1001-3695.2010.03.097

Fast algorithm for building CAN topology

DAI Chang-hua, ZHANG Hao
(*School of Information System & Management, National University of Defense Technology, Changsha* 410073, *China*)

Abstract: Inspired by binary tree, based on T-Man protocol and fast algorithm for building Kademlia, this paper proposed a fast algorithm for building CAN over unstructured P2P network. Kademlia is binary tree structure, and CAN bases on virtual multi-dimensional Cartesian coordinate space partition. Because the fast algorithm for building Kademlia has benn proposed, by defining CAN space partition as tree map, the key problem came to convert Kademlia binary tree structure to CAN tree map and built the routing table. Experiments demonstrate that algorithm builds CAN in a logarithmic number of steps.

Key words: P2P; CAN(controller area network); topology management

本文的出发点在于如何从非结构化 P2P 网络快速构建 CAN[1] 网络拓扑。M. Jelasity 等人[2] 提出的基于流言协议的分布式网络拓扑管理协议(topology management protocol, T-Man)的基本思想是从邻居节点中选择距离最近的节点进行交谈,交谈双方从对方得到信息,并与自身原有信息合并,通过多轮闲话,使网络拓扑向目标网络拓扑渐进演化。基于 T-Man 协议,文献[3]提出了 Chord[4] 快速构建算法,文献[5]提出了 Kademlia[6] 快速构建算法。

CAN[1] 是一种基于空间划分的结构化 P2P 网络。它的节点逻辑上存在于一个虚拟多维笛卡尔坐标系中,整个坐标空间被动态地分配给各个节点,每个节点负责自己所在的区域,区域相邻的节点称为邻居节点,各节点维护一个路由表,路由表包括节点各邻居的 IP 地址和相应的区域。节点间的路由机制是通过邻居关系,采用简单的贪婪算法向目标靠近。当一个新节点加入 CAN 网络时,首先利用哈希算法将该节点映射到坐标空间的一个点;然后将该点所在区域划分为两个区域,新加入节点和原有节点各占一个区域。节点退出或失败的处理采用加入过程的逆过程,即合并失败节点所负责的区域,当某节点发现它的邻居节点失败之后,它就主动合并该区域,从而保证了空间划分的连续性和一致性。

CAN 的平均路由路径长度为 $(d/4)\times(n^{1/d})$,各节点分别维护 $2\times d$ 个邻居,CAN 网络中节点数量的增加并不会带来节点路由表的增大,且路由路径长度的增加速度为 $O(n^{1/d})$。

CAN 具有伸缩性强、自组织、自修复等特点。由于 CAN 能面向多维坐标系,且邻居关系在空间上是连续的,相对于其他结构化 P2P 网络,如 Chord、Kademlia 等,CAN 网络在处理复杂查询如基于内容的查询、范围查询、事件订阅分发等方面具有独特的优势[7]。CAN 的多种优点,使对 CAN 网络拓扑的快速构建算法研究具有重要的研究价值。

1 算法基本思想

如图 1 和 2 所示,Kademlia 为二叉树结构,而 CAN 基于空间划分。二叉树也可表示为树图,如果把构建出的 CAN 网络拓扑的空间划分定义为树图划分的结构,这样从 Kademlia 转换到 CAN 的过程即为二叉树转换为树图,同时建立 CAN 网络的路由表的过程。从非结构化的 P2P 网络构建 Kademlia 的快速构建算法基于 T-Man 协议,针对 Kademlia 的特点作了相应算法的设计,可在对数的步数内构建出 Kademlia,详请参见文献[5]。

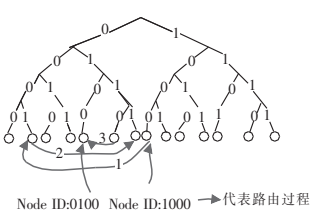


图 1 Kademlia 原理图

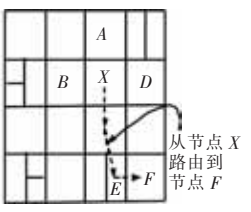


图 2 CAN 原理图

快速构建 CAN 的步骤为:a)首先运用 Kademlia 构建算法快

收稿日期: 2009-07-08; **修回日期**: 2009-08-18

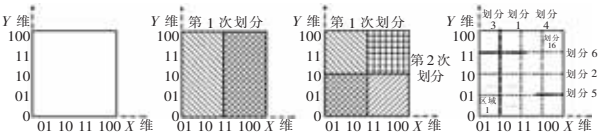
作者简介:戴长华(1962-),男,副教授,主要研究方向为信息资源管理、数据库技术;张昊(1983-),博士研究生,主要研究方向为对等计算、信息可视化(zhang._hao@163.com).

速构建出 Kademlia;b)将 Kademlia 转换为 CAN。步骤 a)对应的算法已发表于文献[5],本文主要解决步骤 b)面临的问题。

2　算法详细设计

由于 CAN 采用了多维坐标系,且节点的路由表为它所负责区域相邻的节点列表,这种邻居关系与节点的 ID 耦合性较 Chord 要松散些。坐标空间的划分是一个自顶向下的过程,因此在很短的时间内为大规模节点各分配一个空间区域成为一个难题,这导致了 CAN 的快速构建相比 Chord 和 Kademlia 更有难度。

从二叉树和树图得到启发,可以把整个坐标空间划分全过程规定为确定的步骤,使坐标空间划分为足够小的区域(使一个区域内存在两个以上节点的可能性非常小或节点 ID 所能表示的最小精度)。各坐标轴按固定的次序逐次划分坐标空间,重复这个过程直到划分后的各区域足够小。如图 3 所示,对于二维空间,先沿 X 轴对坐标空间进行二等分;沿 Y 轴对已划分区域二等分,再沿 X 轴对已划分区域进行二等分;最后形成足够小的区域。图 3 描述了以上定义的划分过程:从图3(a)~(d),观图即可形象地理解以上所述的划分过程,需要注意的一点是,这种划分只是概念上的,并不需要执行实际操作。



(a) 坐标空间未划分　(b) 坐标空间沿 X 维第一次划分　(c) 坐标空间沿 Y 维第二次划分　(d) 坐标空间划分完

图 3　坐标空间划分过程

如图 4 所示,为了便于说明原理,假设节点 ID 用 4 位二进制串表示。若某节点 ID 为 $abcd$,则在坐标空间中,它的 X 轴坐标为 ac,Y 轴坐标为 bd。如此处理后,各节点在空间中各有一个位置,各节点负责其所在位置的右上区域,节点 0000、1011、0011、0110、1100、0111、1101、1111 分别负责各区域,如图 4 中各种不同填充图案的矩形所示。

图 5 为图 4 所示的空间分配情况的二叉树表示,黑色节点表示已有节点负责的区域,白色节点表示没有节点负责的区域。图 4 所示的空间分配是不连续的,有部分区域并没有节点负责,所以构建 CAN 网络的关键问题是合并掉那些没有节点负责的区域,并保持合并后的区域仍为矩形。

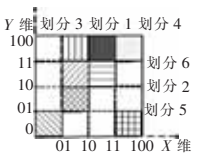


图 4　节点根据自己 ID 负责相应区域

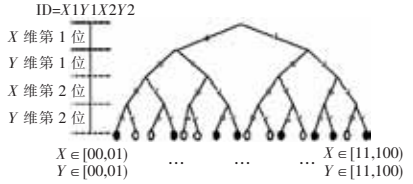


图 5　图 4 的二叉树表示形式

从图 5 可发现,区域合并分为以下三种情况:a)两兄弟节点都为黑色节点,则它们无须合并;b)两兄弟节点一个为白色节点,另一为黑色节点,则该黑色节点合并白色节点并代替它们原来的父亲节点;c)两兄弟全为白色节点,则合并之,产生一个白色节点代替它们的父节点。重复以上合并过程直到不能再合并。对图 5 执行以上合并过程得图 6。可以看出,图 6 中仍存在一个白色节点,算法的目标是合并掉所有白色节点。

为此,再采取以下策略:白色节点的兄弟节点必然不是叶节点(否则可以继续合并),以白色节点的兄弟节点为根节点,存在一棵子树,对其进行深度优先搜索,直到找到两个互为兄弟的叶节点,让其中一个叶节点接管白色节点所代表区域,代替原来的白色节点,另一叶节点接管两兄弟节点负责区域的并,代替它们的父节点。图 6 采用以上策略后得到图 7,这时,图中的所有叶节点均为黑色节点。

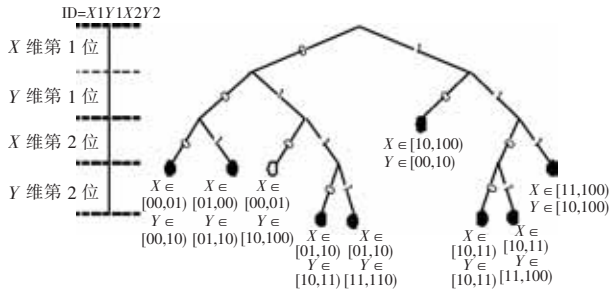


图 6　对图 5 节点合并后

将图 7 所示的二叉树映射到原空间坐标系中得图 8,这时,整个坐标空间被连续、完全、不重叠地划分完毕,每个节点各负责一个相应区域。

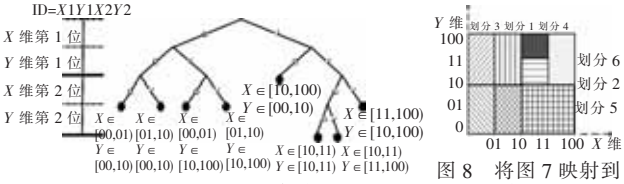


图 7　对图 6 再次合并　　图 8　将图 7 映射到坐标空间

综合以上分析,可以先快速构建出 Kademlia,形成一棵二叉树,然后各个节点运行空间合并算法,使整个坐标空间被连续、完全、不重叠地分配给各个节点;之后通过 Kademlia 路由机制,各节点找到对应的邻居节点,建立 CAN 的路由表,最终形成 CAN 网络拓扑。

以上分析中的二叉树和树图只是概念上的,任何一个节点都不需要存在这样的全局二叉树结构,各个节点根据自己的 ID 可以知道自己在二叉树中的逻辑位置。当所有节点构成 Kademlia 后,每个节点拥有了 Kademlia 路由表,即 K 桶。节点从近到远依次检查路由表的各个 K 桶,当发现某一 K 桶为空时,即表明全局二叉树中对应的某棵子树的叶节点都是白色节点,该节点与自己所在的兄弟子树中的节点用 Kademlia 路由机制传递消息进行通信,当有多个可用黑色节点时,则选举出一个黑色节点接管那棵子树;当只有自身一个黑色节点时,则合并那棵子树,该黑色节点接替这两棵子树的父节点。针对图 5 所示的 Kademlia,最终构建出的 CAN 网络拓扑如图 7 所示。

空间区域合并完毕后,整个空间已经被完全、连续、不重叠地分配给各有节点,下一阶段的任务是让各个节点建立 CAN 的路由表,即各节点通过 Kademlia 路由机制查找到 CAN 中的邻居节点。如图 9 所示,对于二维 CAN 网络来说,各节点需要找到它的四个邻居节点,为了减少通信开销,各节点分别找到两个指定方向上(如 X、Y 维正向)的邻居,并把自身信息通知给邻居,这样所有节点都能知道自己的四个邻居。假定各节点分别找到各维正方向上的邻居,正方向的邻居也就知道了向它发消息的节点,从而各维的负方向邻居关系就自动建立起来,这样整个 CAN 网络拓扑就建立起来了。

以上算法构建出的网络拓扑综合了 Kademlia 网络拓扑与 CAN 网络拓扑。Kademlia 的路由效率为 $O(\log N)$,一般情况下快于 CAN 的路由效率 $O((d/4)\times n^{1/d})$,但 Kademlia 不能支持复杂多维查询,所以同时保持 Kademlia 网络拓扑与 CAN 网络拓扑有重要作用,并且,它们共同存在并相互补充提高了系

统的鲁棒性。当然,维护网络拓扑的开销不可避免地增加了,可以在保证一定的系统鲁棒性基础上,各节点适度减少主动维护网络拓扑的动作。文献[8]提出了一种根据节点在线时间来减少结构化 P2P 网络拓扑维护开销的算法,可以用来缓解本章提出算法带来的维护网络拓扑开销增加的问题。

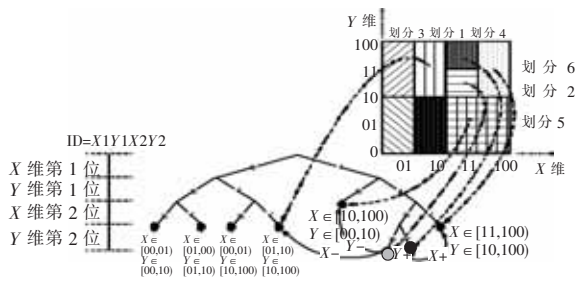


图 9　建立 CAN 网络的邻居关系

节点加入和退出该网络拓扑的处理办法可以沿用 Kademlia 网络和 CAN 网络的处理办法。当然,构建出 CAN 网络拓扑后,完全可以丢掉之前构建的 Kademlia 网络拓扑,这可以视具体情况的要求而定。

3　算法实验分析

Kademlia 网络拓扑的构建可以在对数的轮数内完成,详细讨论与实验见文献[5]。因此,本文对 CAN 网络拓扑构建算法的实验分析集中于从 Kademlia 网络拓扑构建 CAN 网络拓扑过程的开销。

从 Kademlia 网络拓扑构建 CAN 网络拓扑过程的开销主要是节点间的通信开销。从 Kademlia 网络拓扑构建 CAN 网络拓扑,节点间通信主要发生在两个阶段:第一个阶段是节点所负责区域的合并过程,第二阶段是节点建立 CAN 网络路由表的过程。

从 Kademlia 网络拓扑构建 CAN 网络拓扑的第一个阶段是节点所负责区域的合并过程,如上文所述,区域合并采用自底向上的方式,该阶段的通信开销主要是选举节点去接管相应区域,对这一阶段的评估采用统计节点需要参加的选举次数。第二阶段为建立 CAN 网络路由表的过程,各节点通过 Kademlia 网络找到自己各维正方向上的邻居。在这一阶段,对于 k 维坐标系,各节点需要查询 k 个邻居,查找次数是固定的。

实验分析主要考察从 Kademlia 网络拓扑构建 CAN 网络拓扑过程的第一阶段的通信开销。实验采用 peerSim 作为模拟平台,对于所有节点,采用均匀随机函数为各节点产生一个 ID。

第一组实验考察不同网络规模下,所有节点需要参加的选举次数。节点 ID 长度为 60 bit,实验结果如图 10 所示。可以看出,总选举次数与网络模型成线性关系,斜率约为 0.6,即在不同的网络规模下,节点平均需参加约 0.6 次选举。

第二组实验考察不同 ID 长度下,所有节点需要参加的选举次数。网络中共有 10 000 个节点,实验结果如图 11 所示,可以看出,ID 的长度对选举次数几乎不产生影响。

实验发现,节点参加选举的次数分布如图 12 所示,当网络规模为 10 000 个节点,ID 长度为 60 bit 时,节点最多参加选举的次数为 3 次,从而不会出现一个节点需要参加太多次选举的情况,网络中所有节点的选举过程都可以在很短时间内完成,从而保证了算法的实用性。

对于 k 维坐标系,由于第二阶段各节点需要查询 k 个邻居,查找次数为 k,第一阶段平均每个节点需参加约 0.6 次选举,从而使平均每个节点的新增通信开销为常数,不随网络规模而变化。结合文献[5]提出的快速构建 Kademlia 网络拓扑算法的性能,本文提出的 CAN 网络拓扑快速构建算法能在对数的时间内构建出 CAN,且各节点新增通信开销为常数。

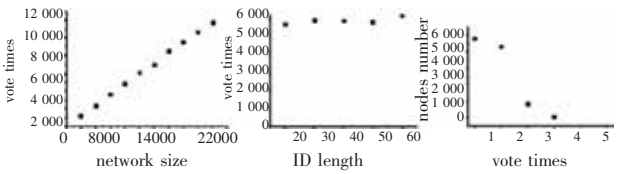


图 10　ID 长度为 60 bit 时,网络规模与选举次数关系图　　图 11　网络规模为 10 000 时,ID 长度与选举次数关系图　　图 12　ID 长度为 60,网络规模为 10 000 个节点时,各节点选举次数分布图

如图 6 所示,只有当节点存在兄弟节点且其父节点的兄弟节点为空白节点时,才需要进行选举,节点的选举次数与其 K 桶中的信息有关,用 L 表示 ID 的长度,N 表示网络中节点的个数,则当节点 ID 完全随机时,第 i 个 K 桶为空的概率为 $P_i = (1 - N/2^L)^{2^i}$,则第 i 个 K 桶不为空的概率 $\overline{P_i} = 1 - (1 - N/2^L)^{2^i}$,节点需要参加的选举次数 $V = \sum_{i=0}^{N-1} P_{i+1} \times \overline{P_i}$,当 $N = 10\,000$,$L = 60$ 时,$V \approx 0.584\,96$。通过对 N、L 代入不同的值(保证 N 远小于 2^L)可以发现,对于 N、L 的不同取值,$V \approx 0.58$ 始终成立,与实验结果一致(图 10)。

4　结束语

在快速构建 Kademlia 网络拓扑的基础上,本文提出了一种快速构建 CAN 网络拓扑的算法,阐述了算法的思想及过程,最后进行了实验。实验表明,本算法能在对数的时间内构建出 CAN,且平均每个节点的新增通信开销为常数,不随网络规模而变化,从而使算法拥有较好的伸缩性。算法如何应用于复杂的网络环境是下一步研究的重点。

参考文献:

[1] RATNASAMY S, FRANCIS P, HANDLEY M, *et al*. A scalable content-addressable network[C]//Proc of International Conference on Applications Technologies, Architectures, and Protocols for Computer Communication. New York: ACM Press, 2001: 161-172.

[2] JELASITY M, BABAOGLU O. T-Man: gossip-based overlay topology management[C]//Proc of the 3rd International Workshop on Engineering Self-Organizing Applications. Berlin: Spring, 2006: 1-15.

[3] MONTRESOR A, JELASITY M, BABAOGLU O. Chord on demand[C]//Proc of the 5th IEEE International Conference on Peer-to-Peer Computing. Washington DC: IEEE Computer Society, 2005: 87-94.

[4] STOICA I, MORRIS R, KARGER D, *et al*. Chord: a scalable peer-to-peer lookup service for Internet applications[C]//Proc of International Conference on Applications Technologies, Architectures, and Protocols for Computer Communication. New York: ACM Press, 2001: 149-160.

[5] 张昊, 戴长华, 张翀. 一种构建 Kademlia 网络拓扑的高效算法[J]. 计算机应用研究, 2009, 26(2): 534-536.

[6] MAYMOUNKOV P, MAZIERES D. Kademlia: P2P information system based on the XOR metric[C]//Proc of IPTPS'02. Berlin: Springer, 2002: 53-65.

[7] LUA E K, CROWCORFT J, PIAS M, *et al*. A survey and comparison of peer-to-peer network schemes[J]. Journal of IEEE Communications Survey and Tutorial, 2005, 7(2): 72-93.

[8] XU Zhi-yong, MIN Rui, HU Yi-ming. Reducing maintenance overhead in DHT based peer-to-peer algorithms[C]//Proc of the 3rd International Conference on Peer-to-Peer Computing. Washington DC: IEEE Computer Society, 2003: 218-219.