

基于域的分布式最小连通支配集的启发式算法^①

陈 勤, 朱 韬, 张 旻, 文小亮

(杭州电子科技大学 软件与智能技术研究所, 杭州 310018)

摘 要: 在规模较大且移动较频繁的 ad hoc 网络中, 针对构建树形连通支配集缓慢且网络开销大的问题, 提出了基于域的分布式最小连通支配集的启发式算法(ZBCDS)。ZBCDS 在求得极大独立集的基础上, 定义了节点阶势和候选节点的概念, 通过判断节点的阶势, 优化了域的生成和域边界上连接节点的调整, 达到 CDS 重构快速高效地实现的目的。实验结果表明, ZBCDS 算法能高效且快速的构建最小连通支配集, 且比同类算法生成的连通支配集更小, 时间复杂度有所降低。

关键词: ad hoc; 支配节点; 最小连通支配集; 分布式算法

Zone-Based Distributed Heuristic Approximation Algorithm for Minimum Connected Dominating Set

CHEN Qin, ZHU Tao, ZHANG Min, WEN Xiao-Liang

(Institute of Software and Intelligent Technology, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: With regard to the problem of slow progress and high overhead, when constructing a tree-like CDS in a fast-moving ad hoc networks, this paper puts forward a Zone-based distributed heuristic approximation algorithm for minimum connected dominating set(ZBCDS). ZBCDS defines the concept of potential-rank and candidate nodes, on the basis of calculating maximal independent set, optimizes the zone partition and the adjustment along the zone borders via nodes judging their own potential-rank, finally reaches the aim of rapidly and efficiently reconstructing CDS as topology change. Experimental results show that ZBCDS can rapidly and efficiently construct CDS, with smaller size of CDS and less time complexity.

Keywords: ad hoc; dominator node; minimum connected dominating set; distributed algorithm

1 引言

移动 ad hoc 网络不依赖固定基础设施, 是军事通信中一种重要的通信网络, 也被用于救援和野外通信等领域。设计有效的广播算法是 ad hoc 网络路由协议的重要内容, 通过“洪泛”进行广播易引起广播风暴问题, 为此通常设计最小连通支配集近似算法构造 ad hoc 网络的虚拟骨干网^[1]。这种策略的优势是只有较少的支配节点而不是所有节点去储存全网的拓扑信息, 减少了网络通信的开销, 可有效缓解广播风暴。

国外已提出许多计算网络最小连通支配集的分布式算法。文献[2]根据文献[3]的思想提出的分布式算法中要求每个节点都能获知自身是否具有最大的度数,

同时在连通支配集中的节点要求能获知其 2 跳范围内邻居的着色情况, 为此采用洪泛法传播各个节点的度数信息。文献[4]引入极大独立集概念, 根据节点在生成树中所处层次和节点标识确定相应的权值, 基于节点权值构造极大独立集(MIS)及支配树。以上算法构建的树形 CDS 拥有更少节点, 但分布式的簇头选举产生了巨大的网络开销, 大量消耗了在 ad hoc 网络中本来就稀缺的网络资源, 并且 CDS 的构建速度也会随着网络规模的增大而急剧变慢。

然而, 构建一个网状的 CDS 能减少网络开销并加速 CDS 的构建, 因此如何快速高效的构建网状 CDS 得到了广泛的关注^[5-7]。文献[5]中提出了 DCA-S(i), 在

^① 基金项目:现代通信国家重点实验室基金(9140c11020670c11);杭州电子科技大学校科学研究基金(KYF071506005)

收稿时间:2010-06-05;收到修改稿时间:2010-07-12

保证 CDS 连通的前提下, 通过剪断节点数量小于 i 的环路, 使得网状 CDS 变得稀疏。文献[6]中提出了一种 ad hoc 网络中的混合的虚拟骨干网路由架构。文献[7]中提出了域的概念, 将网络分成一系列不同的域, 在每个域中构建支配树并在域之间的边界上增加一些连接节点使得能形成最终的 CDS。但 Zone 算法过于宽泛地生成种子支配节点, 导致最终生成的 CDS 中环路过多。以上三种算法在相同条件下, 生成的网状 CDS 的大小会比树形 CDS 大一些。

2 ZBCDS 算法

2.1 概述

本文借鉴了文献[8]采用代表短程场且具有良好的数学性质的高斯势函数描述节点间相互作用的思想, 定义阶(rank)来表示节点的属性, 节点在势场中影响力大小为阶势, 采用节点通过邻居节点的信息及其行为, 来判断本节点的阶势, 避免节点索取全网的拓扑信息所带来的巨大网络开销。ZBCDS 算法分为域的生成部分和域边界上连接节点的调整部分。域的生成: 通过判断节点的阶势来生成种子支配节点, 从种子支配节点开始生成一个树形 CDS 网络; 域边界上连接节点的调整: 在域之间的边界上, 增加一部分连接节点使得域之间连通。以上过程避免了 Zone^[7]中过于宽泛地生成种子支配节点(两个种子支配节点之间的距离最小为 2)所带来的密集环路。

2.2 符号说明和算法假设

(1) $N_1(v)$: $\{v+u \mid \text{Hop_count}(u,v) \leq 1\}$, 表示节点 v 的 1 跳之内邻居节点集合的闭集。

(2) $\text{level}(v)$: 表示节点 v 在其所在域中的层数 (level)。

(3) rank_1 : 定义节点 v 的初始状态阶(rank_1)为有序对 $\langle \delta(v), \text{id}(v) \rangle$, 这里 $\delta(v)$ 表示节点 v 的度, $\text{id}(v)$ 表示节点 v 的 ID。若两个节点分别是 v 和 u , 当满足以下任一条件时 $\text{rank}_1(v) > \text{rank}_1(u)$: (1) $\delta(v) > \delta(u)$; (2) $\delta(v) = \delta(u)$ 且 $\text{id}(v) < \text{id}(u)$ 。

(4) rank_2 : 定义节点 v 的非初始状态阶(rank_2)为有序对 $\langle \langle \delta(v), \text{level}(v) \rangle, \text{id}(v) \rangle$, 若两个节点分别是 v 和 u , 当满足以下任一条件时 $\text{rank}_2(v) > \text{rank}_2(u)$: (1) $\delta(v) > \delta(u)$; (2) $\delta(v) = \delta(u)$ 且 $\text{level}(v) < \text{level}(u)$; (3) $\delta(v) = \delta(u)$, $\text{level}(v) = \text{level}(u)$ 且 $\text{id}(v) < \text{id}(u)$ 。

(5) select 消息: 若节点 v 在 $N_1(v)$ 中初始状态阶 (rank_1)不是最大的, 则向 $N_1(v)$ 中 rank_1 最大的节点发送 select 消息。

(6) Dominator 消息: 表示节点 v 宣布自己为支配节点, 包含自己的域 ID、level 信息和选择的连接节点 ID。

(7) Dominatee 消息: 表示节点 v 宣布成为被支配节点, 包含自己的域 ID、level 信息和选择的支配节点 ID。

算法应满足以下假设: (1)在网络初始时, 每个节点可通过交换控制信息知道其邻居节点的信息; (2)一个节点发出的消息能被其所有的邻居节点在短时间内正确接收。

2.3 预备概念介绍

(1) 候选节点

初始状态的节点 v 成为候选节点, 需满足以下两个条件之一: (1)节点 v 在 $N_1(v)$ 中初始状态阶(rank_1)最大; (2)节点 v 的 $N_1(v)$ 中 rank_1 比 v 大的节点都成为了被支配节点。令满足条件(1)的节点为种子候选节点; 满足条件(2)的节点为非种子候选节点。

由于传统的通过构建极大独立集 MIS 的方法来构建连通支配集 CDS 的过程中, MIS 集合中的节点必然是极小的支配集, 即属于 CDS 集合中, 但很多情况下, MIS 集合中的节点不拥有其所能支配的节点。因此, 本文引入候选节点的概念, 以突破传统的极大独立集 MIS 和连通支配集 CDS 的关系, 使得候选节点为 MIS 集合中的节点, 但不支配任何节点, 所以候选节点不属于最终的 CDS 集合。

(2) 阶势

节点 v 通过 $N_1(v)$ 中的邻居节点的 rank_1 及其行为 (发送 select 消息和选择支配节点), 感知 2 跳范围内 rank_1 的相对大小, 体现了一种势的概念。阶势分为三个等级: 绝对强势、相对强势和弱势。

绝对强势: 一个候选节点 v , 如果收到所有邻居节点的 select 消息, 可知自己 rank_1 在其 2 跳范围内是最大的, 一旦节点 v 成为种子支配节点, 其邻居节点都将成为节点 v 所支配的节点。

相对强势: 一个候选节点 v , 如果从收到的 Dominatee(u)消息中得知自己被节点 u 选为支配节点, 可知相对于 2 跳范围内的一些节点处于相对强势。

弱势: 当节点 v 在其 2 跳范围内处于弱势, 需满

足以下两个条件之一：(1)没有满足成为候选支配节点的条件；(2)成为候选支配节点，但没有所能支配的被支配节点。

2.4 算法描述

本文的算法分为两个部分：(a)域的生成；(b)域边界上连接节点的调整。

(1) 域的生成

若节点 v 在其 $N_1(v)$ 中 $rank_1$ 最大，则确定自己为种子候选节点，否则向自己邻居中 $rank_1$ 最大的节点发送 *select* 消息。当节点 v 确定自己为种子候选节点且阶势为绝对强势，则声明自己为种子支配节点，广播 *Dominator* 消息。当节点 v 收到 *Dominator* 消息，若为初始状态，则节点 v 选择该节点为自己的支配节点，声明自己为被支配节点，广播 *Dominatee* 消息。节点 v 成为支配节点，首先必须先成为候选节点，然后满足下面三个条件之一：(1)节点 v 阶势为绝对强势。(2)节点 v 阶势为相对强势。(3)节点 v 得知自己在边界上且需要连接到别的域。

令满足条件(1)的支配节点为种子支配节点，满足条件(2)、(3)的为非种子支配节点。为了划分网络并同时在域内生成一个树形的 CDS，令种子支配节点的 ID 为其所在域的域 ID， $level$ 初始值为 0，然后将域 ID 和 $level$ 信息加入 *Dominator* 消息中。当一个节点收到 *Dominator* 消息，就可得到自己的域 ID、 $level_1$ 值(将 *Dominator* 消息中的 $level$ 值加 1)。每个被支配节点也将自己的域 ID、 $level_1$ 信息加入 *Dominatee* 消息。

当节点 v 收到 *Dominatee* 消息，若节点 v 为种子候选节点或者节点 v 为初始状态且 $N_1(v)$ 中 $rank_1$ 比 v 大的节点都成为被支配节点，则从邻居节点中选择非初始状态阶($rank_2$)最大的被支配节点作为连接节点，并从被挑选的节点的 *Dominatee* 消息中得到自己的域 ID、 $level_1$ 值，声明自己为非种子候选节点，广播 *Dominator* 消息。然后等待邻居节点选择自己作为其支配节点，根据阶势判断是否成为非种子支配节点。拥有相同域 ID 的节点组成一个域，网络将被划分为不同的域，域内为树形结构。

(2) 域边界上连接节点的调整

为了生成整个网络的 CDS，需要在域边界上选择一些被支配节点作为连接节点。若支配节点 v 在其 3 跳内有不同域的支配节点，则支配节点 v 是边

界上的支配节点，记做 *Domb*；若被支配节点邻居中有不同域的节点，则节点是边界上的被支配节点，记做 *Deeb*。

为了使支配节点得到 3 跳内的不同域的支配节点信息，*Deeb* 需要向自己的支配节点发送 *Neighboring-dom* 消息，该消息包含 *Deeb* 邻居中的不同域的节点 $rank_2$ 信息、域 ID。*Domb* 可以根据已有的节点信息，选择邻居中 $rank_2$ 最大的链路上的被支配节点作为连接节点。若一个候选节点得知自己在边界上且需要连接到相邻域，则令自己为非种子支配节点。最后为了防止冲突和减少通信开销，只有拥有更小域 ID 的 *Domb* 才能进行该域边界上连接节点的调整。

本文算法最终生成的 CDS 中节点包含种子支配节点、非种子支配节点和连接节点。

2.5 图例

(1) 初始，所有节点都是初始状态(图 1(a))。

(2) 节点 2、5、10 和 15 确定自己为种子候选节点，等待邻居节点发送 *select* 消息，其余节点都向自己邻居中 $rank_1$ 最高的节点发送 *select* 消息。箭头指向表示节点发送的 *select* 消息的去向(图 1(b))。

(3) 节点 2 和 10 宣布自己是种子支配节点，广播 *Dominator* 消息，由于它们已经是种子候选节点且阶势为绝对强势。这样所有节点将被划分为以这两个节点为种子支配节点的两个域。节点 4、6、11、13、14、18、21、22 和 23 收到 *Dominator* 消息之后，宣布自己为被支配节点，广播 *Dominatee* 消息(图 1(c))。

(4) 在收到 *Dominatee* 消息之后，节点 15 和 24 都选取了邻居中 $rank_2$ 最大的被支配节点 22 作为连接节点，广播 *Dominator* 消息。因为节点 15 已经是种子候选节点，而节点 24 则满足成为非种子候选节点的条件(图 1(d))。

(5) 节点 5 和节点 15 相同，收到 *Dominatee* 消息之后广播 *Dominator*。通过节点 23 的 *Dominatee* 消息，候选节点 24 得知自己在域的边界上。通过节点 23 的 *Neighboring-dom* 消息，支配节点 10 得知自己在两个域的边界上，选择节点 23 为连接节点。当节点 24 得知节点 10 通过连接节点 23 连接上自己所的域，令自己为非种子支配节点。其余节点的过程和以上类似。(图 1(e))。

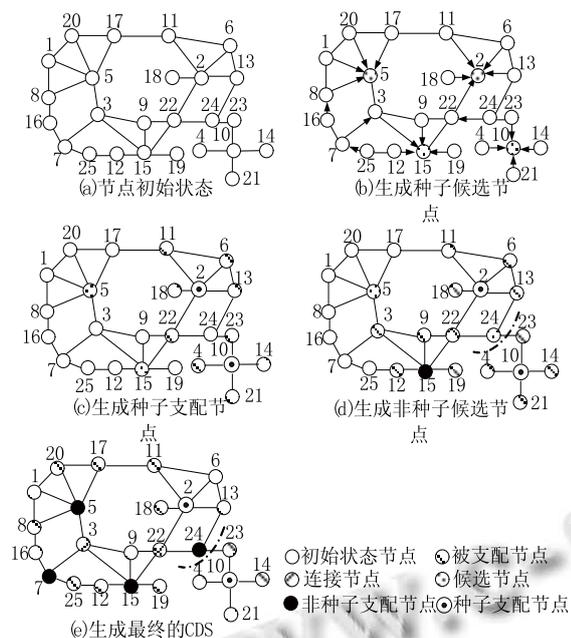


图 1 算法图例。(a)初始状态(b)生成种子候选节点(c)生成种子支配节点(d)生成非种子候选节点(e)生成最终的 CDS。

4 算法分析

定理 1^[9]. 图 G 的一个顶点集 D 是一个独立支配集当且仅当 D 是一个极大独立集。

推论 1^[9]. 图 G 的每个极大独立集是一个极小支配集。

定理 2. 给定一个连通的简单无向图 $G=(V,E)$,由本文算法得到的 D (包含种子支配节点、非种子支配节点和候选节点)构成了一个极小支配集。

证明: 只需证明 D 是极大独立集,依据推论 1 就可说明它也是极小支配集。先说明 D 是独立集,即 D 中的任两个支配点不可能是邻居。用反证法,令 v 和 u 是相邻的支配点,不失一般性可假定,则依据候选节点定义, v 不能成为候选节点,因为它的邻居中至少已存在一个支配点 u , 那么不可能成为支配节点了,与假设矛盾,所以两个支配点不可能相邻,故 D 是独立集。再说明 D 的极大性,也用反证法,即算法结束后,得到的 D 是独立集但不是极大独立集,这说明至少还存在有某受支配节点 u 周围邻居中没有任何支配点,则节点 u 肯定要确定自己为候选节点,这与假设矛盾,所以 D 是极大独立集。依据推论 1 就可说明 D 也是极小独立支配集。证毕。

引理 1^[10] 在单位圆图 $G=(V,E)$ 中,,其中, $MIS(G)$ 表示图 G 的极大独立集的节点总数。

定理 3 给定一个单位圆图 $G=(V,E)$, 最优的情况下 ZBCDS 算法生成连通支配集大小为。

证明: 假设 ZBCDS 算法生成的候选节点都成为了支配节点, 则当每个域中不止一个支配节点时, 每个域中的非种子支配节点都连接到一个 2 跳远的支配节点, 并且由引理 1 中可知, 因而 ZBCDS 算法生成的 CDS 大小最大为。证毕。

5 仿真结果

通过 NS-2 仿真, 比较本文的算法、文献[7]中的 Zone 算法和文献[11]中的 DCA-S(4)算法产生的 CDS 大小以及时间开销。实验所用的参数如表 1 所示。

表 1 实验参数

参数	值	描述
L	200 m	区域的长度
W	200 m	区域的宽度
N	50,100,...,300	节点个数
R	30 m	无线传输距离

测试所用的随机图使用文献[12]中的方法产生,令节点数从 50 变化到 300, 对每种规模均随机产生 50 个网络拓扑图, 然后测试算法在这些网络中的性能并求其平均。本文对算法产生的结果进行归一化处理, 即求算法产生的连通支配集的大小与网络节点数之比。显然, 算法求得的连通支配集越小, 性能越好。算法在随机网络中的测试结果如图 2 所示。

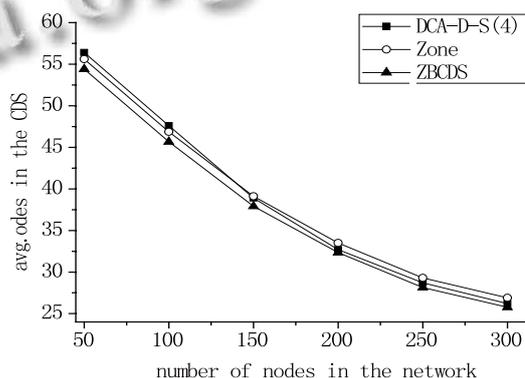


图 2 算法的性能比较

由图 2 可以看出, ZBCDS 算法、Zone^[7]和 DCA-D-S(4)^[11]算法的性能都随网络规模的增大而提高, 而 ZBCDS 算法的性能优于后两者, 当节点数达

到 250 以上时, ZBCDS 算法求得的连通支配集的大小不到网络节点数的 30%。

在 ad hoc 网络中快速构建 CDS, 即算法执行的时间开销是评价算法的另一个重要指标。仿真结果如图 3 所示。

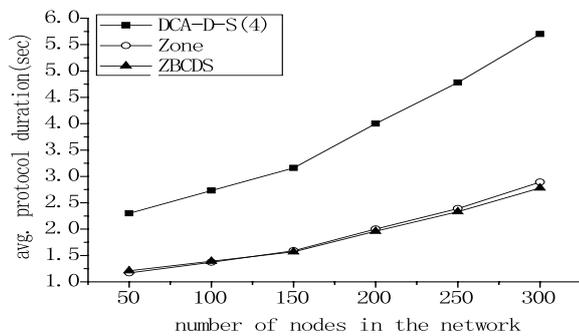


图 3 时间开销比较

随着节点数的增加, 节点的密集程度相应增大, 节点之间的环路相应增多。三种算法的时间开销都随节点密集程度的增大而变大, 而算法 DCA-D-S(4) 的执行时间明显比另两种算法长。当节点数量大于 150 时, ZBCDS 算法优于文献[7]的算法。

6 总结

Ad hoc 网络中的虚拟骨干网及其生成算法具有重要的应用价值。本文在基于域的算法的基础上, 定义了阶势和候选节点概念, 提出了 ZBCDS 算法, 与 Zone^[7]和 DCA-D-S(4)^[11]算法相比, 本算法在生成的连通支配集大小、时间复杂度等方面更具优势。

参考文献

1 陈勤, 范文涛, 张旻. 新的分布式最小连通支配集启发式算法. 计算机工程, 2009, 35(9): 92-94.

- 2 Das B, Bharghavan V. Routing in Ad hoc networks using minimum connected dominating sets. 1997 IEEE of International Conference on Communications. Montreal, 1997. 376-380.
- 3 Guha S, Khuller S. Approximation algorithms for connected dominating sets. Proc. of the 4th annual European symposium on Algorithms. 1996.
- 4 Wan PJ, Alzoubi K, Frieder O. Distributed construction on connected dominating set in wireless ad hoc networks. Proc. of the 21 Annual Joint Conference of the IEEE Computer and Communications Societies. 2002, 3: 1597-1604.
- 5 Basagni S, Mastrogiovanni M, Petrioli C. A Performance comparison of protocols for clustering and backbone formation in large scale ad hoc network. Proc. of the First IEEE International Conference on Mobile Ad Hoc and Sensor Systems. 2004. 70-79.
- 6 Liang B, Haas ZJ. Hybrid routing in ad hoc networks with a dynamic virtual backbone. IEEE Trans. on Wireless Communications, 2006, 5(6): 1392-1405.
- 7 Han B. Zone-based virtual backbone formation in wireless ad-hoc networks. Ad-hoc Networks, 2009, 7: 183-200.
- 8 淦文燕, 赫南, 李德毅, 王建民. 一种基于拓扑势的网络社区发现方法. 软件学报, 2009, 20(8): 2241-2254.
- 9 殷剑宏, 吴开亚. 图论及其算法. 合肥: 中国科学技术出版社, 2004.
- 10 Wu WL, Du HW, Jia XH, Li YS, Huang SCH. Minimum connected dominating set and maximal independent set and in unit disk graphs. Theoretical Computer Science, 2006, (3)52: 1-7.
- 11 Basagni S, Mastrogiovanni M, Panconesi A, Petrioli C. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. IEEE Trans. on Parallel and Distributed Systems, 2006, 17(4).
- 12 阎新芳, 孙雨耕, 胡华东. 基于极大权的最小连通支配集启发式算法. 电子学报, 2004, 32(11): 1774-1777.