

一种修复网络拓扑的 Steiner 树移动控制算法

闫中江, 沈中, 常义林, 张颖, 代亮

(西安电子科技大学综合业务网理论及关键技术国家重点实验室, 710071, 西安)

摘要: 针对无线 Ad Hoc 网络中拓扑修复成功率低、节点移动开销大的问题, 提出了一种 Steiner 树移动控制算法(SMC). 采用三近似最少 Steiner 点算法建立一棵包含网络节点和 Steiner 点的 Steiner 树, 然后将引入的 Steiner 点作为节点移动的目的点, 选择并调度一些节点移动到这些 Steiner 点上, 最后更新网络拓扑, 迭代执行算法直到建立一个连通的网络拓扑. 仿真结果表明, 与基于分区最小生成树的移动控制算法相比, SMC 算法不仅修复网络拓扑的成功率可达到 100%, 而且还显著降低了节点移动开销, 其中节点移动总距离减小了 37%~45%, 节点移动总数减少了 9%~29%.

关键词: 无线 Ad Hoc 网络; 拓扑修复; 移动控制; Steiner 树

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2011)02-0039-05

Steiner Tree Movement Control Algorithm for Topology Recovery

YAN Zhongjiang, SHEN Zhong, CHANG Yilin, ZHANG Ying, DAI Liang

(State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China)

Abstract: A Steiner tree movement control algorithm (SMC) is proposed to improve low success rate and large movement cost in wireless ad hoc network topology recovery. The SMC computes a Steiner tree by calling 3STP-MSP algorithm, where the vertices contain all the nodes in the network and the introduced Steiner points. Then the Steiner points are regarded as movement destinations, and some nodes are selected and scheduled to move to these Steiner points. SMC runs iteratively until the network is connected. Simulation results show that the SMC not only reconnects the network with a success rate 100%, but also decreases the movement cost significantly, compared with the MST-based PMST-UV algorithm, and that the total movement distance is reduced by 37% to 45% and the number of total movement nodes is reduced by 9% to 29%.

Keywords: wireless ad hoc networks; topology recovery; movement control; Steiner tree

网络连通性是保障无线 Ad Hoc 网络通信和实现各种网络应用的基础^[1], 但是由于节点故障、网络被破坏或摧毁、电能耗尽等原因, 网络容易被分割成若干个不连通的子网. 对不连通的网络拓扑, 移动节点对于恢复网络连通、修复网络拓扑具有重要意义. 文献[2]调度非割节点移动到割点位置, 修复因割点失效而造成的拓扑分割, 文献[3]调度非支配节点移动到 2 个分区之间, 修复分区连通, 但它们只解决了拓扑修复问题中的部分特例. 文献[4]提出了一种调

度非割节点移动改善拓扑控制性能的 PMST-UV 算法, 该算法可修复网络拓扑恢复网络连通^[5], 但存在节点移动距离大, 算法可能失效等缺点.

本文引入 Steiner 树^[6-7]的概念, 提出了一种修复网络拓扑的移动控制算法(SMC). SMC 算法首先采用 3STP-MSP 算法^[8]建立一棵包含网络节点和 Steiner 点的 Steiner 树, 然后将引入的 Steiner 点作为节点移动的目的点, 选择并调度一些节点移动到这些 Steiner 点上, 最后更新网络拓扑, 迭代执行算

法直到建立一个连通的网络拓扑. 仿真结果表明, 与基于分区最小生成树的 PMST-UV 算法相比, SMC 算法不仅具有较高的网络拓扑修复成功率, 而且还极大地减小了节点移动开销.

1 系统模型

设无线 Ad Hoc 网络中共有 N 个具有移动和定位能力的无线节点, 任意节点 u 有唯一的节点 ID 和相同的通信半径 R , 定义 u 的重要度为 p_u . 如果节点 u, v 之间的距离 $d(u, v) < R$, 则它们之间存在一条无线链路, 使用边 $e(u, v)$ 表示, 形成的网络拓扑使用无向图 $G=(V, E)^{[6]}$ 表示, 其中 V 为网络中的无线节点集合, E 为节点间的无线链路集合. 令 $G=(V, E)=\bigcup_{i=1}^n G_i$ 为初始网络拓扑, 其中 G_i 为 G 的第 i 个连通分支称为分区变量.

给定欧氏平面上的点集 P , 在允许添加额外点集 S 的条件下构造连接 P 中所有节点的树, 其中具有最短总距离的树称为最小 Steiner 树 T , 点集 S 中的点 s 称为 Steiner 点^[6-7]. 使用最少的 Steiner 点连通给定节点集, 并且节点间距离不大于给定常数 R 的问题称为 STP-MSP 问题, 已被证明为 NP-hard 问题^[9]. Cheng 等人^[8]给出了一个以近似系数为 3 构造 Steiner 树的 3STP-MSP 算法: 假设所有的节点均不连通, 任意 2 个节点之间存在一条虚拟边, 对于边长小于 R 的边, 如果所连接的 2 个节点不在同一个连通分支中, 则将这 2 个连通分支合并; 检查每一个节点组合 (a, b, c) , a, b, c 分别属于 3 个不同的连通分支, 如果存在一个 Steiner 点 s 能够连通该组合, 则添加 s 和链路 $\{e(s, a), e(s, b), e(s, c)\}$, 并将这 3 个连通分支合并; 按序添加 $d(u, v) > R$ 的边 $e(u, v)$, 在 $e(u, v)$ 的方向上相隔距离 $\frac{d(u, v)}{R}$ 放置 $\lfloor \frac{d(u, v)}{R} \rfloor$ 个 Steiner 点, 并将距离不大于 R 的节点 u, v 以及 Steiner 点连接, 合并边 $e(u, v)$ 所连接的 2 个连通分支; 依次添加、连接 Steiner 点, 合并连通分支, 直到所有节点连通.

2 SMC 算法

基于 Steiner 树的拓扑修复移动控制算法——SMC 算法的设计目的是使用较小的移动距离修复存在多个分区的不连通网络.

2.1 算法描述

图 1 给出了 SMC 算法的伪代码, 主要包括 3 个

部分: 首先采用 3STP-MSP 算法^[9] 建立 Steiner 树 T , 再将 T 中引入的 Steiner 点集 S 作为节点移动的目的点(步骤 1~2); 然后逐一确定网络中各节点的重要度(步骤 3~10); 最后将网络中重要度为 0 的节点移动到 Steiner 点上(步骤 11~16), 当网络拓扑重新连通时算法结束, 否则迭代执行算法(步骤 17~20).

```

1. Call 3STP-MSP in [8] to build a Steiner tree  $T$  of  $G$ , with
   node set  $V$  and transmission radius  $R$ ;
2. Let  $S$  denote the Steiner point set in  $T$ ;
3. Determine all the cut nodes in each  $G_i$  of  $G$  by DFS[6];
4. for each node  $u$ , determine  $u$ 's important degree  $p_u$ 
5.    $p_u = 0$ ;
6.   if  $u$  is a cut node
7.      $p_u = p_u + 1$ ;
8.    $p_u = p_u + m$ ,  $m$  is the neighbor Steiner points number of  $u$  in  $T$ .
9.   if  $u$  is an isolate node in  $G$ 
10.     $p_u = p_u - 1$ ;
11. for each network partition  $G_i$ 
12.   get the matched nodes pair:  $(u_i, s_i) = \min \{d(v_i, s_j) \mid d(v_i, s_j)$ 
      where  $v_i \in G_i, p_{v_i} = 0$ , and  $s_j \in S\}$ ;
13. for each  $s \in S$ 
14.    $u = \min \{d(u_i, s_i) \mid \text{where } u_i \in G_i, \text{ and } s_i \text{ is matched with } u_i\}$ ;
15.   if node  $u$  exists
16.     move  $u$  to  $s$ ;
17. if  $G$  is connected
18.   return;
19. else
20.   turn to step 1;
  
```

图 1 SMC 算法伪代码

为了定量描述节点 u 对拓扑连通的重要性, 以及删除或移动节点 u 对网络连通性的影响程度, 定义了节点的重要度. 在采用 Steiner 树连通的网络拓扑中, 有两类节点对连通性至关重要. 第一类是割点: 在网络分区内部, 如果节点 u 为割点, 那么移动或删除该节点将使得分区进一步分割. 第二类是与 Steiner 点相邻的节点: 在网络分区之间, 如果节点 u 是某个 Steiner 点 s 的邻节点, 那么移动节点 u 之后, 即使有其他节点移动到该 Steiner 点 s 的位置, 也不能与节点 u 所在的分区连通. 此外, 还有一类节点对网络连通无重要性, 即孤立节点. 当且仅当孤立节点 u 为 Steiner 树的叶子节点(u 仅与一个 Steiner 点相邻), 且对于分区内部连通无重要性时, $p_u = 0$. 在 SMC 算法中, 对任意节点 u , 采用下述方法计算其重要度 p_u : 对于第一类重要节点, 如果 u 为割点则 $p_u = p_u + 1$; 对于第二类重要节点, 如果与节点 u 相邻 Steiner 点的个数为 m , 则 $p_u = p_u + m$; 对于孤立节点, 如果节点 u 为孤立节点, 则 $p_u = p_u - 1$.

节点重要度描述了删除或移动单个节点对拓扑连通性的影响,但是同一分区中如果有多个节点同时移动,仍可能造成网络分区分割^[6]. 为避免这种情况,每个分区在调度节点移动时,只选择一个重要度为 0 的节点移动. 对每个网络分区,从中只选取一个重要度为 0、移动距离最小的节点与 Steiner 点 s 匹配(见图 1 中算法步骤 11~12),在每一个 Steiner 点 s 处,选择已经被匹配的移动距离最小的节点移动到 s 的位置(步骤 13~16). 由上述描述可知,当网络中所有节点均为孤立节点时,重要度为 0 的节点为 Steiner 树 T 的叶子节点,SMC 算法将选择这部分孤立节点移动,从而重建网络连通.

当节点被调度移动之后,检查网络是否已经连通(见图 1 算法步骤 17~20),如果尚未连通,则转到步骤 1,迭代执行算法,否则 SMC 算法结束.

2.2 算法分析

下面对 SMC 算法的计算复杂度进行分析. 采用 3STP-MSP 算法建立 Steiner 树的计算复杂度为 $O(n^3)$ ^[8],使用 DFS 算法确定每个分区 G_i 的割点,计算复杂度为 $O(n^2)$ ^[6],确定每个节点的重要度,计算复杂度为 $O(n)$,遍历每个分区和 Steiner 点,匹配移动节点和移动目的节点集,最坏情况下计算复杂度为 $O(n^2)$,因此算法总的计算复杂度为 $O(n^3)$.

图 2 给出了使用 PMST-UV 算法和 SMC 算法进行网络拓扑修复的示例. 图 2a 中初始网络拓扑含有 5 个网络分区,即 $\{1,2,3,4\}$ 、 $\{5,6\}$ 、 $\{7\}$ 、 $\{8,9\}$ 和 $\{10\}$. 为修复网络拓扑,PMST-UV 算法首先通过建立分区最小生成树,在分区间添加尽可能少的增补边^[4] $e(3,9)$ 、 $e(4,7)$ 、 $e(4,6)$ 和 $e(7,10)$ 使得网络连通,如图 2b 所示. 然后,确定各分区中的割点(即节点 $\{2,3\}$),移动非割节点到增补边建立网络连通,即调度非割节点 $\{8,5\}$ 分别移动到 $\{8',5'\}$. 由图 2b 可以看到,孤立节点 10 作为增补边 $e(7,10)$ 的网关节点^[4] 没有被调度移动,而由图 2c 可以看到,SMC 算

法则已调度节点 10 移动. 当网络中不存在可被调度的非割节点,或所有节点均为孤立节点时,PMST-UV 算法将失效. 此外,PMST-UV 算法只考虑了 2 个分区间的位置关系,本文所提出的 SMC 算法则联合考虑了多个分区的位置,这样可进一步减少移动节点数量,减小节点移动距离.

下面结合 SMC 算法伪代码和图 2c,详细阐述 SMC 算法修复拓扑的步骤. ①根据图 1 中步骤 1 建立连通所有节点的 Steiner 树,确定节点移动的目的点,即 Steiner 点 $\{5',7',10'\}$. ②根据图 1 中步骤 3 对每一个分区执行 DFS 算法,即确定分区中的割点 $\{2,3\}$. ③为了使得节点的移动不破坏原网络分区的连通性和整个网络的连通性,在图 1 中步骤 4~10 采用计算节点重要度的方法确定每一个节点的重要度. 表 1 给出了在每一步计算过程中各节点的重要度. 可以看出,最终可以移动的节点为 $\{1,5,8,10\}$. ④根据图 1 中步骤 11~16,匹配重要度为 0 的节点和 Steiner 点,并调度节点 $\{5,8,10\}$ 移动到 Steiner 点 $\{5',7',10'\}$. 在 SMC 算法中,只有成为 Steiner 树叶子节点的孤立节点的重要度才为 0(如节点 10),其他孤立节点的重要度不为 0(如节点 7). 从图 2c 可以看出,SMC 算法只确定了 3 个移动目的点,调度 3 个节点移动即可重新连通网络,与 PMST-UV 算法相比,节点移动数减少了,移动距离也减小了.

表 1 图 2c 中各节点重要度的变化情况

步骤	p_u									
	1	2	3	4	5	6	7	8	9	10
4~5	0	0	0	0	0	0	0	0	0	0
6~7	0	1	1	0	0	0	0	0	0	0
8	0	1	2	1	0	1	2	0	1	1
9~10	0	1	2	1	0	1	1	0	1	0

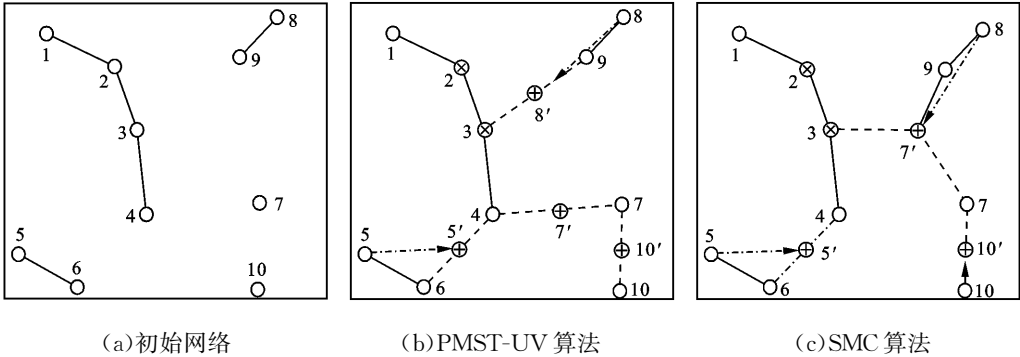


图 2 PMST-UV 算法和 SMC 算法网络拓扑修复示例

3 仿真结果

成功率和迭代次数是评价拓扑修复算法的重要性能参数. 成功率越高说明算法越有效, 迭代次数少不仅表明算法收敛速度快, 还表明算法执行时间短. 在能够保证重新建立网络连通的前提下, 节点移动距离小则说明节点在移动过程中所消耗的能量少, 重新建立连通后网络生存时间长. 因此, 算法的性能验证主要集中在算法的有效性和节点移动开销上.

由于文献[2-3]只是本文所研究问题的特例, 不具有一般性, 因此本文在仿真实验中未与它们进行比较, 而是与 PMST-UV 算法^[4]进行比较. 设无线 Ad Hoc 网络部署在一个 $L \times L$ 的平坦、无障碍的矩形区域内, 节点通信半径 $R=50$ m, 网络中节点总数为 N , 则平均邻节点度 $D=N\pi R^2/L^2$. 文献[10]表明, 当 $D<6$ 时, 网络不连通的概率大于 90%. 为研究算法修复不连通网络的性能, 令 $1 \leq D \leq 5$, 对任意固定的 D 值, 令 $50 \leq N \leq 150$. 为减小随机误差, 对每一个确定组合 (D, N) , 随机生成 100 个网络场景, 以平均值作为仿真结果.

表 2 给出了初始网络和分别使用 SMC 算法与 PMST-UV 算法进行节点移动控制后的网络性能. 从表 2 可以看出, 在平均邻节点度比较低($D=1$) 时, PMST-UV 算法的成功率不高, 而 SMC 算法在所有场景中均可修复网络拓扑(成功率 100%). 此

外, SMC 算法的迭代次数均小于 PMST-UV 算法. 在保证能够修复网络拓扑的情况下, 2 种算法在节点的最大通信半径方面相当, 而 SMC 算法的平均通信半径则较大, 这表明 SMC 算法不仅能够保证修复网络拓扑, 还减小了节点的移动开销.

图 3 给出了 2 种算法在不同 D 值下, 节点移动总数和节点移动总距离的比较. 可以看出, SMC 算法所使用的节点移动总数明显比 PMST-UV 算法少, 同时节点移动总距离比 PMST-UV 算法也小, 与理论分析一致,

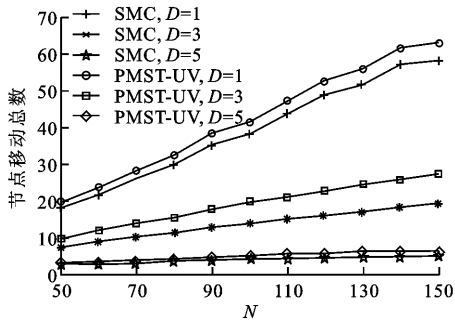
图 4 比较了 2 种算法的节点平均移动距离和节点最大移动距离. 图 4a 表明, SMC 算法中节点的平均移动距离较小, 图 4b 表明, SMC 算法中节点的最大移动距离远小于 PMST-UV 算法, 从而对每个被移动的节点都不会造成太大的移动开销.

4 结束语

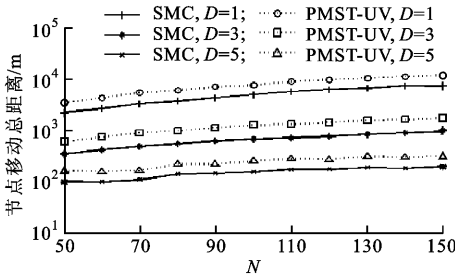
在无线 Ad Hoc 网络中, 网络连通是保障网络通信、实现各种网络应用的基础. 本文研究了无线 Ad Hoc 网络分割后的拓扑修复问题, 提出了一种 Steiner 树移动控制算法——SMC. 该算法迭代调度节点向 Steiner 点移动, 直到网络重新连通. 仿真结果表明, 与基于分区最小生成树的 PMST-UV 算法相比, SMC 算法不仅具有较高的网络拓扑修复成功率, 而且还极大地减小了节点移动开销.

表 2 SMC 算法与 PMST-UV 算法的性能比较

D	N	分区数	算法成功率/%		迭代次数		最大通信半径/m			平均通信半径/m		
			PMST-UV	SMC	PMST-UV	SMC	初始网络	PMST-UV	SMC	初始网络	PMST-UV	SMC
1	60	37	78	100	18.3	5.2	143.1	38.7	38.6	61.1	29.5	38.6
1	100	61	84	100	21.4	6.5	144.6	41.9	42.7	60.5	31.6	42.7
1	140	85	90	100	24.5	7.2	147.6	44.8	45.0	60.1	33.8	45.0
3	60	13	100	100	3.3	1.1	83.3	49.5	49.4	35.3	31.8	49.4
3	100	21	100	100	4.5	1.2	86.4	49.5	49.6	35.1	31.6	49.6
3	140	27	100	100	4.9	1.2	85.9	49.7	49.8	34.7	31.4	49.8
5	60	4	99	100	1.3	1.0	62.5	47.8	48.1	27.6	26.6	48.1
5	100	6	100	100	1.6	1.0	63.5	48.8	49.0	27.2	26.6	49.0
5	140	7	100	100	2.0	1.0	65.4	49.1	49.2	26.8	26.2	49.2

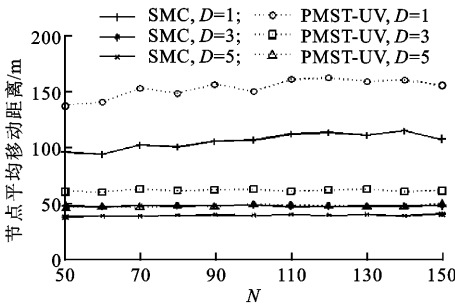


(a)节点移动总数

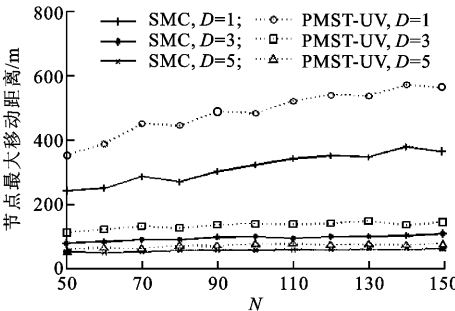


(b)节点移动总距离

图 3 节点移动总数和节点移动总距离比较



(a)节点平均移动距离



(b)节点最大移动距离

图 4 节点平均移动距离和节点最大移动距离比较

参考文献:

[1] SESAY S, YANG Zongkai, HE Jianhua. A survey on mobile ad hoc wireless network [J]. Information Technology Journal, 2004, 3(2): 168-175.

[2] ABBASI A A, AKKAYA K, YOUNIS M. A distributed connectivity restoration algorithm in wireless sensor and actor networks [C] // Proceedings of 32nd IEEE Conference on LCN. Piscataway, NJ, USA; IEEE, 2007: 496-503.

[3] AKKAYA K, SENEL F. Detecting and connecting disjoint sub-networks in wireless sensor and actor networks [J]. Elsevier Ad Hoc Networks, 2009, 7 (7): 1330-1346.

[4] 公维宾, 常义林, 沈中. 传感器网络中实现传输功率均衡的移动控制算法 [J]. 西安电子科技大学学报, 2008, 35(3): 816-822.

GONG Weibing, CHANG Yilin, SHEN Zhong. Movement control algorithms for realizing the balance of transmission power for sensor networks [J]. Journal of Xidian University, 2008, 35(3): 816-822.

[5] 公维宾. 无线 Ad Hoc 网络节点移动技术研究 [D]. 西安: 西安电子科技大学通信工程学院, 2009.

[6] DIESTEL R. Graph theory [M]. 3rded. Heidelberg, Germany: Springer-Verlag, 2006.

[7] GILBERT E N, POLLAK H O. Steiner minimal trees [J]. SIAM Journal on Applied Mathematics, 1968, 16 (1): 1-29.

[8] CHENG Xiuzhen, DU Dingzhu, WANG Lusheng, et al. Relay sensor placement in wireless sensor networks [J]. Wireless Networks, 2008, 14(3): 347-355.

[9] CHEN Donghui, DU Dingzhu, HU Xiaodong, et al. Approximations for Steiner trees with minimum number of Steiner points [J]. Elsevier Theoretical Computer Science, 2001, 262(1/2): 83-99.

[10] BETTSTETTER C. On the minimum node degree and connectivity of a wireless multihop network [C] // Proceeding of 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing. New York, NY, USA: ACM Press, 2002: 80-91.

(编辑 刘杨)