

基于骨干多边形的传感器网络分区双连通恢复算法^{*}

秦宁宁^{1,2}, 吴德恩¹, 余颖华¹

(1. 江南大学物联网工程学院, 江苏 无锡 214122;

2. 江南大学轻工过程先进控制教育部重点实验室, 江苏 无锡 214122)

摘 要:针对现有算法恢复分区连通性存在容错性差的问题,提出了分区双连通性恢复算法 DCRA。该算法旨在网络中心区域构建骨干多边形,分区以两条互不相交的路径与多边形连接,从而实现分区间的双连通。仿真实验表明,与现有一些双连通算法相比,所提算法不仅减少了部署中继节点的数量,而且算法的运行时间能够减少 60% 左右,可以快速确定部署的中继节点位置,从而快速恢复分区连通性。

关键词:双连通;容错性;骨干多边形;中继节点

中图分类号:TP393

文献标志码:A

doi:10.3969/j.issn.1007-130X.2017.04.009

A double connectivity recovery algorithm in partition based on backbone polygon in sensor networks

QIN Ning-ning^{1,2}, WU De-en¹, YU Ying-hua¹

(1. School of Internet of Things Engineering, Jiangnan University, Wuxi 214122;

2. Key Laboratory of Advanced Process Control for Light Industry of Ministry of Education, Jiangnan University, Wuxi 214122, China)

Abstract: In order to solve the problem that the existing algorithms have poor fault tolerance when recovering the partition connectivity, we propose a double connectivity recovery algorithm in partition (DCRA). The algorithm aims at building a backbone polygon in the center area of the network and connecting partitions with the polygon by two disjoint paths to realize double connectivity between partitions. Simulation experiments show that compared with some existing double connectivity algorithms, the proposed algorithm cannot only reduce the number of deployed relay nodes and the running time of the algorithm by about 60%, but also quickly determine the location of the relay nodes so as to quickly recovery partition connectivity.

Key words: double connectivity; fault tolerance; backbone polygon; relay node

1 引言

由大量传感器节点构成的无线传感器网络通过节点间的相互协作执行监测任务^[1-3],并广泛应用于恶劣的环境,因此易遭受大规模的破坏。当大量节点失效,网络随之分割成互不通信的分区,网

络的功能受到影响。为此,及时重建分区链路恢复网络功能十分重要^[4-6]。

通过部署中继节点,恢复分区连通性已成为研究的热点方向。在中继节点修复网络过程中,最关键的问题是如何部署更少的中继节点,同时确保分区能够通信^[7]。为了解决这个问题,许多启发式方法已经被提出。迟百川等人^[8]提出运用移动数据

^{*} 收稿日期:2015-12-28;修回日期:2016-03-04

基金项目:江苏省“六大人才高峰”第十一批高层次人才项目(DZXX-026);2014 年国家公派高级研究学者及访问学者(含博士后)项目;国家自然科学基金(61304264);江苏高校优势学科建设工程资助项目(PAPD);江苏省产学研联合创新资金前瞻性联合研究项目(BY2014023-31);中央高校基本科研业务费专项资金(JUSRP51510)

通信地址:214122 江苏省无锡市滨湖区江南大学物联网工程学院

Address: School of Internet of Things Engineering, Jiangnan University, Binhu District, Wuxi 214122, Jiangsu, P. R. China

收集器作为移动中继节点建立间歇性的通信链路,以恢复分区间通信,但该算法易造成数据传输的延迟。黄健文等人^[9]提出的连通自修复算法虽然减少了节点的移动距离和数量,但该算法需要节点具有一定的移动性,增加了网络的修复成本。Lee等人^[10]提出了一种基于最小斯坦纳树的算法。算法包括确定三角形斯坦纳点位置与部署中继节点两个阶段,但该算法复杂度较高,且形成的网络拓扑中存在大量割点,任意割点的失效会再次分割网络,破坏数据传输。为了减少割点,2C-SpiderWeb (2-Connected Spider Web)算法^[11]通过形成一个类似蜘蛛网的结构,部署中继节点实现网络的双连通,虽然形成的网络具有一定的容错性,但大量中继节点的布置降低了网络效率。具有容错性的分区恢复算法 CRAFT (Connectivity Restoration with Assured Fault Tolerance algorithm)^[12]也采用类似三角形斯坦纳点确定多边形,从而建立两条不相交路径,实现分区间的双连通。虽然该算法在一定程度上减少了中继节点数量,但该算法复杂度高,不适合在大规模网络中使用。

本文在兼顾复杂度、效率、容错性等多种性能的同时,给出了一种分区双连通性恢复算法 DCRA (Double Connectivity Recovery Algorithm in partition)。DCRA 通过对比最小距离和,来确定最佳部署节点,基于此构造的骨干多边形,可以更快实现分区间的双连通,提高网络的容错性。

2 网络描述

2.1 网络模型

在给定的二维平面区域 Ω 内随机部署数量为 N 的传感器节点,组成初始静态同构网络 $S = \{S_i | i = 1, 2, \dots, N\}$ ^[13],网络内节点之间是端到端通信方式。图 1 给出了网络遭受大规模破坏后形成的分区示意,实心圆点代表失效节点,则作为正常节点的空心圆点组成了若干个独立分区,分区内可以通信,而分区间不能通信。其中, Seg_i 为所形成的 N_{seg} 个分区中的第 i 个分区, $i = 1, 2, \dots, N_{seg}$ 。

基于系统设计的集约化考虑,在追加部署中继节点、实现分区间双连通时,也应尽可能减少所需中继节点数目。为了便于分析,本文依旧假设中继节点与传感器节点具有相同的通信半径 R 和感知半径 r 。节点能够利用现有定位算法计算获知自身位置信息。

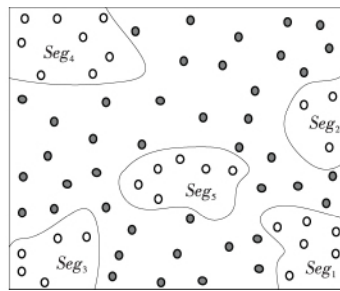


Figure 1 A separate partition established after the network is damaged

图 1 网络受破坏后,形成独立分区

2.2 问题描述

不失一般性,基于文献[7]的描述,将分区 Seg_i 抽象为点 P_i ,即 $Seg_i = P_i$,故存在集合 $P = \{P_i | i = 1, 2, \dots, N_{seg}\}$ 。此时,传感器网络分区连通性恢复问题可描述为:在空间 Ω 中,存在 N_{seg} 个不连通节点 P_i ,应追加放置尽可能少的中继节点 RN_k ,以重建节点 P_i 之间链路,实现分区间的双连通,从而使得恢复后的网络,不会因单个节点的失效再次陷入分区的不连通状态,即应满足下式:

$$N_{RN} = \min \sum_{i=1}^{N_{seg}} \left(\sum_{j=1, j \neq i}^{N_{seg}} Cycle(P_i, P_j) \right) \quad (1)$$

$$s. t. \begin{cases} RN_k^{i,j}, k = 1, \dots, Cycle(P_i, P_j) \\ RN_l^{i,j} \neq RN_m^{i,j}, \forall l, m \in \{1, \dots, Cycle(P_i, P_j)\} \end{cases} \quad (2)$$

其中, N_{RN} 为所部署的中继节点总数量, $Cycle(P_i, P_j)$ 为点 P_i 和 P_j 之间构成的闭合路径上部署的中继节点数量, $RN_k^{i,j}$ 代表闭合路径上的中继节点。式(1)描述了在式(2)的约束下,在分区网络中实现双连通所需最少中继节点数量^[12] N_{RN} 。鉴于该问题 NP 难的特征^[14],本文将采用启发式算法解决。

3 DCRA 算法

DCRA 算法分为生成骨干多边形、单连通部署、双连通部署三个阶段。在算法描述之前,需首先约定最佳部署节点的定义。

对于已经存在的基点集合 Q ,对其运行 Graham 扫描算法^[15]构建关于 Q 的凸多边形,满足不在其凸壳 $CH(Q)$ 上的节点均位于凸壳内。若节点 $Q_i \in CH(Q)$ 的最佳部署节点为 BN_i ,则 BN_i 应位于以节点 Q_i 为圆心、 R 为半径的圆周上,且 BN_i 与凸壳 $CH(Q)$ 剩余节点距离之和最小,即满足:

$$Dist(BN_i, Q_i) = R \quad (3)$$

$$\text{Min} \sum_{Q_j \in \underset{\text{凸包}}{CH(Q)}, j \neq i} Dist(BN_i, Q_j) \quad (4)$$

其中, $Dist(BN_i, Q_j)$ 为 BN_i 与 Q_j 的二维欧氏距离。

式(3)与式(4)描述了凸壳上一点 Q_i 与其最佳部署节点 BN_i , 及其该部署点与其他凸壳节点间的约束关系。

3.1 生成骨干多边形

骨干多边形按轮计算得到。每轮中, 首先对基点集合运行 Graham 算法确定凸多边形, 然后为凸壳上的所有点确定其最佳部署节点; 再将凸壳内节点以及本轮新生成的最佳部署节点组成新基点集合, 进入下一轮凸多边形与最佳部署节点生成过程。直至所形成的凸多边形内不存在分区代表节点, 则此时由基点集合组成的凸壳即为骨干多边形。骨干多边形生成的伪代码如下:

```

1: 初始化基点集合  $Q=P$ , 轮数  $round=0$ , 凸壳节点集合  $U=\emptyset$ , 凸壳节点中分区节点集合  $T=\emptyset$ , 最佳部署节点集合  $BN=\emptyset$ ;
2: while  $P \neq \emptyset$ 
3:    $U=Graham(Q)$ ; 运行Graham算法, 得到Q凸包的节点集合U
4:    $Q=Q-U$ ; Q-落在凸包上的节点(U)
5:    $round=round+1$ ;
6:    $onPG(round)=U$ ; //第 round 个凸壳节点集合
7:    $T=FoundPartitionnodes(U)$  /* 从集合U中选取分区节点集合T */ 选取U中的representative node, T
8:    $P=P-T$ ; representative node里面删除T
9:   for  $j=1, length(T)$ 
10:     $z(T[j])=round$ ; 记录T中的节点落在第round个凸包上
11:   end for
12:   if  $P \neq \emptyset$ 
13:     $BN=FoundBN(U)$ ; /* 基于公式(3)和(4), 确定U中节点的最佳部署节点集合BN */
14:     $Q=Q \cup BN$ ; 对于U中的每个节点部署一个best relay node
15:    continue;
16:   else
17:    break;
18:   end if
19:    $BN=\emptyset$ ;
20: end while
```

上述伪代码阐述了骨干多边形的形成过程: 其中, $z(i)=0$ 表示分区节点 i 位于第几个凸壳。初始化后, 步骤 3~步骤 7 确定每轮凸多边形的凸壳, 步骤 8~步骤 11 确定分区节点所位于的凸壳, 步骤 12 判断是否依然有分区节点位于凸壳内部, 从而决定是否搜索最佳部署节点, 步骤 13~步骤

15 计算当前凸壳的最佳部署节点, 否则结束算法。当骨干多边形生成后, 算法进入第二阶段的单连通部署。

3.2 单连通部署

本阶段分为两步: 第一步, 在骨干多边形的每条边, 以间距 R 部署中继节点, 实现骨干多边形上所有点之间的连通; 第二步, 采用贪婪算法完成非骨干多边形上的分区节点与骨干多边形连通。部署伪代码如下:

```

1: for  $i=1:Nseg$ 
2:   if  $z(i)=round$ 
3:     continue;
4:   else for  $j=z(i)+1:round$ 
5:      $x=FoundMinDist(onPG[j,:], P_i)$ ; /* 在  $onPG[j,:]$  寻找距离  $P_i$  最近点  $x$  */
6:     连接点  $P_i$  与点  $x$ ;
7:     在线段  $\overline{P_i x}$  上, 以间距  $R$  部署中继节点;
8:      $P_i=x$ ;
9:   end for
10: end if
11: end for
```

步骤 2 判断分区代表节点是否位于骨干多边形上。对未位于骨干多边形上的分区节点, 通过步骤 4~步骤 9 部署中继节点与骨干多边形连通。

3.3 双连通部署

单连通后的网络虽然能够执行监测任务, 但是分区与骨干多边形连通路径存在大量割点, 任意割点的失效, 均可能导致网络再次陷入分区。为了提高网络对于单连通网络中割点的容错性, 各个分区节点应从骨干多边形中选择一个点通过中继节点与之相连, 确保距离该点尽可能最近的同时与已有的单连通网络路径无共属路径段。以此路径作为分区节点到骨干多边的替代路径, 从而构成双连通网络。

4 仿真分析

本文采用 Matlab 软件作为仿真平台研究 DCRA 的性能, 与 2C-SpiderWeb 和 CRAFT 两个算法, 在算法效率、运行时间、稳定性等几个方面进行了对比实验。实验设定目标区域为 $\Omega=1\ 000\text{ m} \times 1\ 000\text{ m}$ 的正方形区域。为验证网络随机有效性, 分区数量 $Nseg$ 取值变化从 4 到 7, 通信半径 R 以 20 m 为间隔, 从 40 m 到 120 m 取值变化, 共计 20 种场景。考虑到网络模型与情景设置的随机

性,仿真分析结果均为 30 次随机实验的均值。

4.1 中继节点数量

图 2a~图 2d 以图 1 中网络为案例,阐述 DCRA 实现分区双连通的具体操作步骤。5 个分区代表节点随机存在于目标区域 Ω 中,图 2a 为第一轮节点布局;算法运行到第三轮即确定骨干多边形,如图 2b 所示;图 2c 和图 2d 分别为分区实现单连通和双连通部署后的网络拓扑。其中,单连通拓扑中,追加部署了 22 个中继节点,双连通拓扑中再追加部署 11 个中继节点。

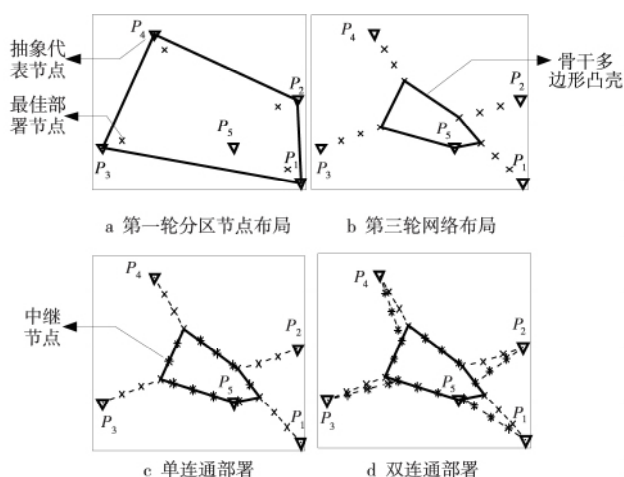


Figure 2 Recovering partition connectivity by the DCRA algorithm

图 2 DCRA 算法恢复分区连通性

为了验证 DCRA 算法的有效性,本文统计了 DCRA 与 2C-SpiderWeb、CRAFT 三种算法在不同通信半径和分区数量情况下,所需部署的中继节点数量。如图 3 所示,给定通信半径 R 和分区数量 N_{seg} ,DCRA、CRAFT 算法所需中继节点数量均小于 2C-SpiderWeb 算法,DCRA 算法部署的中继节点最少,这主要是由于三种算法以不同的方式在中心区域构建多边形或环形造成的。其中,2C-SpiderWeb 算法构造环形时只考虑了部分分区,CRAFT 算法通过局部最优方法计算斯坦纳点构建多边形,而 DCRA 算法通过全局最优方法计算最佳部署节点构建多边形,最大化地减少了中继节点的数量。当给定分区数量 N_{seg} ,随着通信半径 R 的增加,相对于 2C-SpiderWeb、CRAFT,在 DCRA 中由路径长度减少带来的高效优势不再明显,在图 3 中则体现为:随着 R 取值的增大,DCRA 与 2C-SpiderWeb、CRAFT 所需中继节点数量越来越接近。

4.2 平均计算时间

图 4 统计了 DCRA 与 2C-SpiderWeb、CRAFT

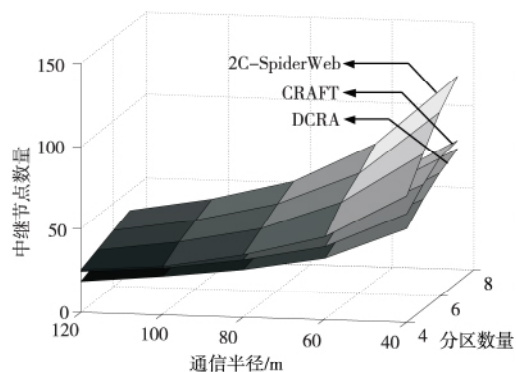


Figure 3 Analysis of the influence of R and N_{seg} on the number of relay nodes

图 3 中继节点数量受 R 和 N_{seg} 的影响分析

三种算法在不同通信半径和分区数量情况下,系统运行所需的平均时间。从实验结果可知,给定通信半径 R 和分区数量 N_{seg} ,DCRA、2C-SpiderWeb 算法的平均计算时间相近,且都小于 200 s,而 CRAFT 算法的平均计算时间都在 500 s 左右,故 DCRA 与 CRAFT 相比,运行时间减少 60% 左右。可见,CRAFT 算法的运行效率最低,而 DCRA 算法具有较高的运行效率,主要是因为两种算法确定多边形的方式不同造成的。DCRA 算法通过最佳部署节点方式确定多边形,大大降低了算法的复杂度,减少了计算时间,从而迅速确定中继节点的位置,达到快速恢复分区连通的目的。

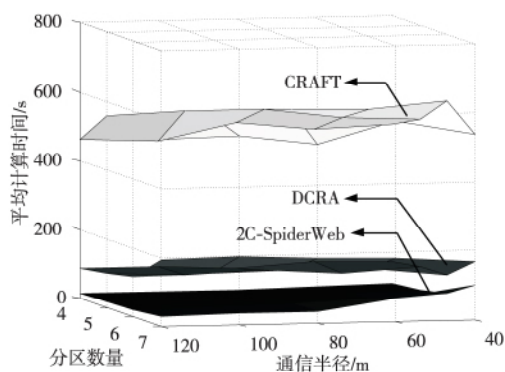


Figure 4 Analysis of the influence of R and N_{seg} on average computation time

图 4 平均计算时间受 R 和 N_{seg} 的影响分析

4.3 平均节点度

图 5 统计了 DCRA 与 2C-SpiderWeb、CRAFT 三种算法在不同通信半径和分区数量情况下的平均节点度。从实验结果可知,给定通信半径 R 和分区数量 N_{seg} ,2C-SpiderWeb 比 DCRA、CRAFT 算法有更高的平均节点度,这意味着网络具有更好的连通性。由于 2C-SpiderWeb 算法以大量中继节点形成网状结构恢复分区连通性,许多节点成为邻居节点,增加了网络中的链路,因此该

算法的平均节点度最高。而 DCRA 算法的平均节点度始终高于 CRAFT 算法,即 DCRA 算法中平均每个节点连接其它节点的个数多于 CRAFT 算法,说明 DCRA 算法形成的网络拓扑的鲁棒性比 CRAFT 算法要好。

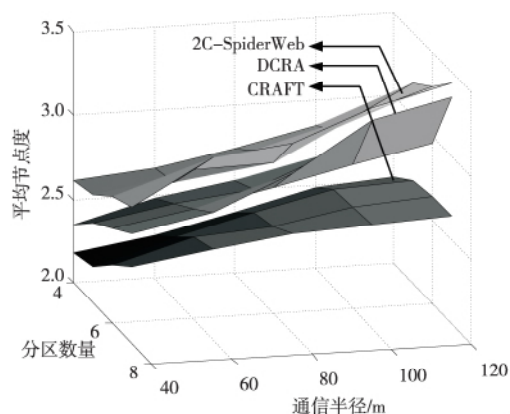


Figure 5 Analysis of the influence of R and N_{seg} on average node degree

图5 平均节点度受 R 和 N_{seg} 的影响分析

5 结束语

本文提出基于骨干多边形的分区双连通性恢复算法(DCRA),应用于遭受破坏的网络,使恢复的网络具有一定的容错性。该算法引入最佳部署节点确定骨干多边形,降低算法复杂度;通过贪婪算法实现分区与多边形的双连通,减少部署中继节点数量。实验结果表明,与当前一些算法相比,该算法能够快速确定部署中继节点的位置,及时恢复分区连通性。在部署较少中继节点的情况下,网络的平均节点度更高,鲁棒性更好。

参考文献:

- [1] Senel F, Younis M. Recovery from multiple simultaneous failures in wireless sensor networks using minimum steiner tree [J]. Journal of Parallel and Distributed Computing, 2010, 70(5): 525-536.
- [2] Chen Hong-sheng, Shi Ke. Quadrilateral steiner tree based connectivity restoration for wireless sensor networks [J]. Journal of Computers, 2014, 37(2): 457-468. (in Chinese)
- [3] Senel F, Younis M, Akkaya K. A robust relay node placement heuristic for structurally damaged wireless sensor networks [C] // Proc of the 34th Conference on the Local Computer Networks, 2009: 633-640.
- [4] Cheng X Z, Du D Z, Wang L S, et al. Relay sensor placement in wireless sensor networks [J]. Wireless Networks, 2008, 14(3): 347-355.
- [5] Senel F, Younis M. Relay node placement in structurally damaged wireless sensor networks via triangular steiner tree

approximation [J]. Elsevier Computer Communications, 2011, 34(16): 1932-1941.

- [6] Yang Hong, Xu Li, Zhang Jing. On constructing fault-tolerant virtual backbone algorithm in wireless sensor networks [J]. Journal of Chinese Computer Systems, 2014, 35(12): 2612-2616. (in Chinese)
- [7] Lee S, Younis M. Optimized relay placement to federate segments in wireless sensor networks [J]. IEEE Journal on Selected Area in Communications, Special Issue on Mission Critical Networking, 2010, 28(5): 742-752.
- [8] Chi Bai-chuan, Cao Jiang-tao, Zhang Yi, et al. Research on restoring technology of network connectivity by mobility-assisted method [J]. Journal of Dalian University of Technology, 2015, 55(1): 97-102. (in Chinese)
- [9] Huang Jian-wen, Ni Wei-ming. An improved self-healing algorithm for connectivity restoration in wireless sensor networks [J]. Information and Electronic Engineering, 2012, 10(6): 659-663. (in Chinese)
- [10] Lee S, Younis M. Optimized relay node placement for connecting disjoint wireless sensor networks [J]. Computer Networks, 2012, 56(12): 2788-2804.
- [11] Senel F, Younis M, Akkaya K. Bio-inspired relay node placement heuristics for repairing damaged wireless sensor networks [J]. IEEE Transactions on Vehicular Technology, 2011, 60(4): 1835-1848.
- [12] Lee S, Younis M, Lee M. Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance [J]. Ad Hoc Networks, 2015, 24(PA): 1-19.
- [13] Senel F, Younis M. Optimized relay node placement for establishing connectivity in sensor networks [C] // Proc of Global Communications Conference (GLOBECOM), 2012: 512-517.
- [14] Lin G H, Xue G. Steiner tree problem with minimum number of Steiner points and bounded edge-length [J]. Information Processing Letters, 1999, 69(2): 53-57.
- [15] Graham R L. An efficient algorithm for determining the convex hull of a finite planar set [J]. Information Process Letters, 1972, 1(4): 132-133.

附中文参考文献:

- [2] 陈洪生, 石柯. 基于四边形斯坦纳树的无线传感器网络连通恢复 [J]. 计算机学报, 2014, 37(2): 457-468.
- [6] 杨洪, 许力, 章静. 无线传感器网络中容错虚拟骨干网构造算法 [J]. 小型微型计算机系统, 2014, 35(12): 2612-2616.
- [8] 迟百川, 曹江涛, 张一, 等. 基于移动协助方式的网络连通性恢复技术研究 [J]. 大连理工大学学报, 2015, 55(1): 97-102.
- [9] 黄健文, 倪卫明. 一种改进的无线传感网连通性自修复算法 [J]. 信息与电子工程, 2013, 10(6): 659-663.

作者简介:



秦宁宁(1980-),女,黑龙江虎林人,博士,副教授,研究方向为无线传感器网络覆盖。E-mail: ningning801108@163.com

QIN Ning-ning, born in 1980, PhD, associate professor, her research interest includes coverage in wireless sensor network.