



分簇及局部优化的无线传感器网络拓扑控制算法

魏永红^{1,2}, 李孝忠¹

(1. 天津科技大学计算机科学与信息工程学院, 天津 300222; 2. 北京理工大学机电学院, 北京 100081)

摘要: 为保证网络连通性和覆盖度的情况下, 尽量合理、高效地使用网络能量, 延长网络生命周期, 提出一种基于分簇和局部优化的拓扑控制 (cluster and local optimization topology control, CLTC) 算法. 基于树型网络模型, 利用分簇思想将网络分割为不同的簇, 簇内运用最小生成树算法, 确定邻居节点关系, 降低节点通信碰撞; 簇间通过簇头连接, 形成优化的骨干网络拓扑. 仿真实验表明, 运行 CLTC 算法, 构建网络拓扑结构快速, 通信开销小, 可以有效降低节点平均能耗, 延长网络周期.

关键词: 无线传感器网络; 拓扑; 分簇; 最小生成树

中图分类号: TP393.02 文献标志码: A 文章编号: 1672-6510(2010)02-0068-04

Clustering and Local Optimization of Topology Control Algorithm for Wireless Sensor Network

WEI Yong-hong^{1,2}, LI Xiao-zhong¹

(1. College of Computer Science and Information Engineering, Tianjin University of Science & Technology, Tianjin 300222, China; 2. School of Mechatronic Engineering, Beijing Institute of Technology, Beijing 100081, China)

Abstract: To prolong the network lifecycle and utilize the network energy more reasonable and efficient in the condition of the network connectivity and coverage, an algorithm of cluster and local optimization topology control (CLTC) was presented. The idea of cluster was used to divide the network into different clusters based on the tree model. The minimum spanning tree algorithm was adopted to identify the relationship among the clusters and reduce the collision of node communication. Meanwhile, the cluster head took charge of connecting the clusters to form the optimal backbone of network topology. The simulation results show that using the CLTC can quickly construct the network topology with less communication consumption, efficiently reduce the average power consumption of the node and extend the network lifecycle.

Keywords: wireless sensor network; topology; cluster; minimum spanning tree

无线传感器网络中节点计算和通信能力有限, 而且采用供电电池能量也十分有限, 所以除了要设计能量高效的网络协议之外, 还要设计优化的网络拓扑控制^[1]. 良好的拓扑控制是一种重要的节省网络能耗的技术, 可保证网络覆盖质量和连通质量, 降低通信干扰, 提高路由协议和 MAC (Media access control) 协议的效率, 为数据融合提供拓扑基础. 网络拓扑控制对于自组织的无线传感器网络性能影响较大.

目前, 分布式拓扑控制方法主要集中在节点功率控制和层次拓扑控制两个方面. LEACH (Low-energy

adaptive clustering hierarchy)^[2]是典型的分层拓扑控制算法, LEACH 算法对节点分簇, 通过概率随机选取簇头, 相邻节点动态地组成簇. 簇内节点将数据发送给簇头, 簇头在进行数据融合后单跳发送给基站. LEACH 算法可使网络中节点等概率担任簇头, 节省节点能量, 延长网络寿命. 但是, 其簇头选举机制没有考虑节点的具体位置, 选举出的簇头分布不均匀, 可能导致网络内节点剩余能量不均衡, 将会增加其成为盲点的机会, 过多的盲点会降低网络的生命周期. LMST (Local minimum spanning tree)^[3]是较为典型的

功率控制算法之一,其基本思想是每个节点会收集它的邻节点的信息,节点间以通信能耗代价作为连接权重,通过全局调整局部独立构建的最小生成树,最终实现拓扑的双连通,能有效地降低维持全局连通的传输功率,减少网络中冗余链路,减少节点竞争通信,提高节点能量利用率,且生成拓扑中节点度的上限值为6,但算法忽略了生成拓扑的健壮性。

在无线传感器网络中存在节点分布不均匀的情况,在某些局部区域内,节点数量多、密度大。如果节点都以最大发送功率通信,将导致局部通信竞争激烈,网络吞吐量急剧下降。因此需要减少邻居节点间发生冲突的概率,提高节点能量利用率。

本文综合 LEACH 算法和 LMST 算法的优点,提出一种基于分簇和局部优化的拓扑控制(cluster and local optimization topology control, CLTC)算法。利用分簇方法将网络按一定的规则划分不同的簇;簇内运用 LMST 算法,确定邻居节点关系,并调整节点发送功率,减少节点通信碰撞;簇间通过簇头连接,形成优化的骨干网络拓扑,减少冗余链路和节点间干扰,降低网络能耗,延长网络寿命。

1 网络模型

假设无线传感器网络模型为树型结构如图 1 所示。汇聚节点为根节点,节点非均匀分布,所有节点都有唯一 ID 和相同的初始能量,节点的发射功率可根据距离调整大小。节点间通信链路对称,无线传播无障碍,通信采用自由空间模型,收发数据的无线通信模型^[2]分别为

$$E_{rx}(h,d) = he_1 + he_2d^2$$

$$E_{tx}(h,d) = he_1$$

式中: e_1 为无线收发电路能耗; e_2 为自由空间模型放大器的能耗; d 为数据传送距离; h 为要发送或接收数据的比特位数。

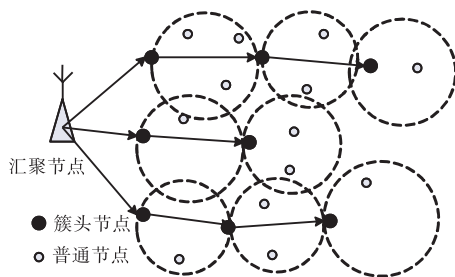


图 1 树型结构网络模型

Fig.1 Tree structure network model

2 CLTC 算法

2.1 相关定义

(1) 某簇的后继簇数由式(1)确定^[4]。

$$n_{\text{CluNode}} = \begin{cases} 2 & 0 < n \leq 10 \\ \lceil n/10 \rceil & 10 < n < 100 \\ 10 & n \geq 100 \end{cases} \quad (1)$$

式中, n 为某簇内节点数。

(2) 邻节点消息 (Hello 消息) 是簇头节点确定单跳邻居节点集合时首先广播的消息, 包含源节点 ID、坐标位置等信息。

(3) 邻节点确认消息 (Neighbor_Node_ACK 消息) 是所有收到 Hello 消息的节点立即发送的反馈消息, 包含源节点 ID、本节点 ID、剩余能量和距离等。

(4) 簇头消息 (ClusterHead 消息) 是已确定的簇头节点向新选出的簇头节点发送的消息, 包含新簇头节点 ID、前继簇头节点 ID 以及树层次。

(5) 簇内消息 (Cluster 消息) 是由簇头发送给簇内节点的消息, 告知簇内节点当前簇头的 ID 及簇内各节点间邻居关系, 收到此消息的节点记录自身所归的簇。

(6) 相邻节点间链接通信的综合权值参数^[5]为

$$w_{ij} = d(i,j)^a / (e_i e_j / \sqrt{e_i + e_j})^b \quad (2)$$

式中: $d(i,j)$ 为两节点之间的距离; e_i 、 e_j 分别为传感器节点的剩余能量; a 、 b 为预设指数。权值由 $d(i,j)$ 和 $e_i e_j / \sqrt{e_i + e_j}$ 两部分决定, 分别反映两节点之间的功耗大小和链接的健壮情况。

2.2 算法的思想

该算法的基本思想是, 首先汇聚节点 (Sink) 发起簇头选取过程, 发送建立邻节点消息, 在确定其邻节点集合后, 汇聚节点依据相应的规则从邻节点集合中选择部分节点作为源簇头 (指由 Sink 节点发起并确定的簇头节点), 并向这些源簇头发送建立簇头消息, 同时设置好前后继关系; 源簇头并行向网络发送建立邻节点消息, 确定各自的邻节点集合后, 按照与源簇头相同的选取原则确定新的簇头; 新选举的簇头重复上面的过程, 直至完成网络分簇的过程。此算法保证网络的覆盖性和连通性。

2.3 分簇过程

(1) 源簇头确定。节点部署完以后, 在网络初始化阶段, Sink 节点向网络广播 Hello 消息, 发起寻找邻居节点过程, 收到 Hello 消息的节点都向 Sink 节点回

复 Neighbor_Node_ACK 消息, Sink 节点接收到其他节点回复的确认消息, 统计自身邻居节点集, 形成一个以 Sink 节点为簇首的簇, 再发送 Cluster 消息, 所有收到簇内消息的节点记录自身所在簇的簇头 ID, 表示自己成为簇内成员; Sink 节点利用最小生成树算法对其管辖的簇进行局部拓扑优化, 在优化的簇中, 依据式(1)确定新簇头数量, 即源簇头数量, 新簇头的多少取决于其前继簇的规模大小; 根据各节点与源簇头的链路综合权值 w_{ij} 以及簇头数量确定具体的源簇头, 并将选取的源簇头消息反馈给 Sink 节点, 然后, Sink 节点向源簇头节点发送建立 ClusterHead 消息, 收到 ClusterHead 消息的源簇头将自己设置为簇头, 设置前继关系, 同时将节点的层次属性设置为 1.

(2) 簇头建立. 每个源簇头分别向网络广播一个建立邻居节点集 Hello 消息, 已经是簇头节点和前继簇的成员拒绝接收 Hello 消息, 当某节点收到来自多个簇头节点的 Hello 消息时, 加入距离最近簇. 新簇头的确定过程与源簇头确定过程相同. 新的簇头确定后, 设置前继关系, 并将节点的层次设置为前继簇层次的值加 1.

(3) 新簇头重复执行(2)过程, 直到到达网络的边界, 即建立树型网络拓扑结构.

2.4 簇内拓扑控制

每个簇看作一个连通网, 即 $N = \{V, E\}$, 其中 V 为连通网顶点集合, E 为边的集合, T_E 是 N 上最小生成树中边的集合. 利用普里姆算法, 从 $U = \{\text{簇头节点 } u_0\}$, $T_E = \{\}$ 开始, 重复下面操作: 在所有 $u \in U$, $v \in V - U$ 的边 $(u, v) \in E$ 中找一条代价最小的边 (u_0, v_0) 并入集合 T_E , 同时 v_0 并入 U , 直至 $U = V$ 为止, 则 $T = (V, T_E)$ 为簇 N 的最小生成树. 簇头 u_0 按到达最近邻居节点的发射功率向簇内节点发送簇内优化消息, 收到消息的节点确定自身邻节点关系, 调整各自发射功率为到达最近邻节点的功率. 簇内优化后, 删除邻节点之间冗余通信链路, 减少通信的碰撞与干扰, 降低了能耗.

2.5 拓扑维护阶段

簇头节点在运行中由于数据处理、转发频繁, 能量消耗较快, 因此, 需设置一个能量门限值 M , 当簇头能量降到 M 时该簇头发起簇头重选工作. 簇头首先检查自己最近邻居节点和其前驱后继的可达性. 当可达性满足时, 则选此节点为新簇头节点并改变前驱后继信息. 若不满足可达性, 则选次近邻居节点, 依次类推直到找到合适的节点或邻节点集合选择完毕. 最坏的情况是, 簇头的邻节点集中没有与前后继

簇头可通信的节点, 此簇将形成孤立簇, 当此种情况达到一定限度后 Sink 发起全网簇头重建工作.

3 算法的仿真及分析

采用 Matlab7.0 仿真软件, 在 $200\text{ m} \times 200\text{ m}$ 区域内随机部署 150 个节点, 进行算法仿真. 汇聚节点和随机部署的节点同质, 且通信半径都为 60 m. 仿真中节点初始能量为 0.5 J, 节点收发电路 (ETX/ERX) 能耗均为 50 nJ/bit, 发射放大器能耗为 10 pJ/(bit·m²), 数据聚合能耗为 5 nJ/bit.

3.1 拓扑分析

为了对比不同算法的性能, 在同一场景分别运行不同的拓扑控制算法. 图 2 是没有使用拓扑控制算法的拓扑, 坐标原点为汇聚节点, 所有节点都以最大发送功率通信时形成的拓扑结构, 网络中链路数量多且相互重叠, 节点间通信碰撞概率大, 能耗高.

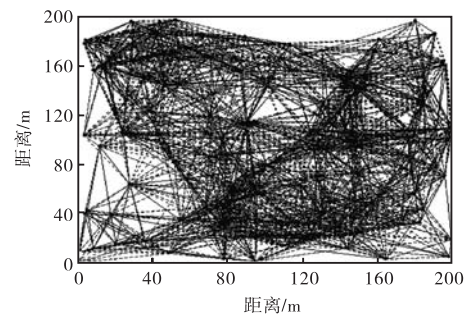


图 2 无拓扑控制的网络拓扑

Fig.2 Network topodgy without topology control

图 3 是运行分簇拓扑算法后形成的拓扑结构. 坐标原点为汇聚节点, 由它发起建立树状网络拓扑, 圆圈为簇头节点, 与簇头连接的圆点为簇内成员节点.

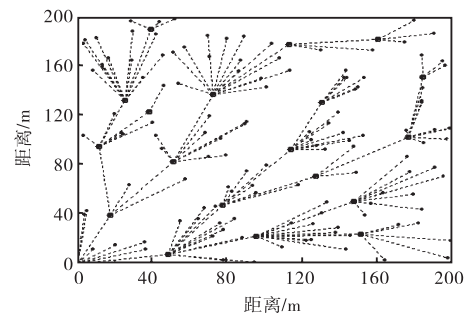


图 3 未局部优化的网络拓扑

Fig.3 No local optimization of network topology

图 4 是分簇后, 每个簇头在簇内执行最小生成树算法形成拓扑结构, 保证网络的连通性和覆盖性, 链路数量明显减少, 降低了节点间通信干扰, 减少能耗, 延长网络生命.

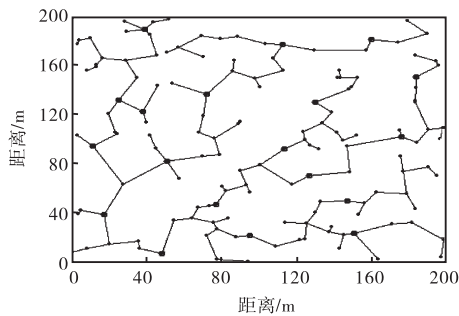


图4 簇内优化后的网络拓扑

Fig.4 Local optimization of network topology

3.2 性能分析

从网络节点平均能耗和簇头能耗两个方面去评价算法的性能. 图5反映了CLTC算法节点平均能耗低于LEACH节点能耗,原因是CLTC簇头选取简单、快捷,不需要全局信息;每个簇头建立簇时,不向全网广播Hello消息,区域内的其他节点接收到的冗余Hello消息较少,耗能也就少,相应簇头也不因长距离传输Hello消息而消耗大量能量;重要的是每个簇内进行了局部拓扑优化,去掉了冗余通信链路,减少节点间的通信碰撞和干扰,从而大大降低了能耗.

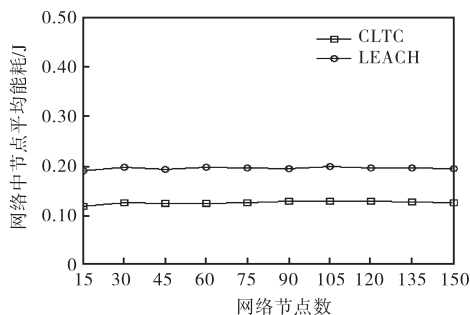


图5 网络节点平均能耗对比

Fig.5 Average energy consumption contrast of nodes

图6为选取了200次循环仿真中,两种拓扑算法每轮循环中簇头节点的总能耗对比.

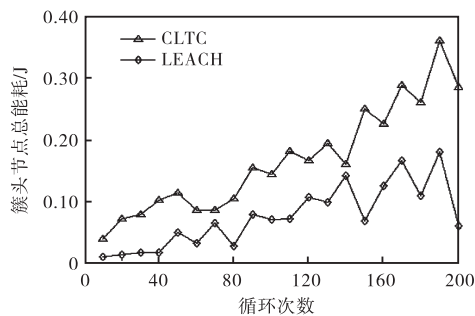


图6 簇头能耗对比

Fig.6 Energy consumption contrast of cluster head

从图中可以看出,随着循环的次数增加,CLTC簇头耗能多于LEACH.原因是CLTC算法采用分簇机制,汇聚点和簇头以多跳的方式进行通信,每个簇头不仅处理来自簇内的数据,同时还向汇聚节点转发来自其他簇的数据,特别是CLTC网络模型是树型结

构,导致靠近汇聚节点的簇头随着仿真时间的增加耗能剧增;而LEACH簇头与汇聚节点以单跳的方式通信,簇头不转发来自其他簇的数据.

4 结语

通过现有拓扑控制算法的研究,提出CLTC算法,该算法利用分簇思想划分网络,簇内通过局部优化调整节点相互关系,建立合适的网络拓扑.仿真实验证明,运行CLTC算法,构建网络拓扑结构快速、通信开销小、有效降低节点平均能耗;同时,该算法在构建拓扑结构时引入相应的路由策略,即建立簇头后设置簇头间的关系,为开发基于此拓扑结构的路由协议奠定了基础;另外,当簇头失效时,仅根据其邻节点就可以重新选取新的簇头,拓扑维护开销较少.

目前,算法在稀疏网络中会存在单成员的簇,选择簇头算法简单,需要进一步改进和完善;而且,靠近汇聚节点的簇头能耗快,容易出现“热区”.针对CLTC算法存在的问题,可以借鉴文献[6]中论述的关于“热区”解决策略、文献[7-8]中关于设计不等规模合理簇结构方法,从非均匀分簇、网络负载能耗均衡以及“热区”避免三个方面来完善改进算法,使CLTC算法能够成为能耗最少、健壮的拓扑控制算法.

参考文献:

- [1] Akyildiz I F, Su W, Sankarasubramanian Y, et al. A survey on sensor networks[J]. IEEE Communications Magazine, 2002, 40(8): 102-105.
- [2] Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks[C]//Proceedings of the Hawaii International Conference on System Sciences. Piscataway: IEEE, 2000: 223.
- [3] Li N, Hou J C, Sha L. Design and analysis of an MST-based topology control algorithm[J]. IEEE Transactions on Wireless Communications, 2005, 4(3): 1195-1206.
- [4] 黄海平, 王汝传, 孙力娟, 等. 基于父亲树的无线传感器网络路由协议[J]. 计算机技术与发展, 2008, 18(8): 4-7.
- [5] 郝晓辰, 翟明, 刘彬, 等. 负载均衡的无线传感器网络拓扑控制算法[J]. 计算机工程, 2008, 35(5): 84-86.
- [6] 吴小兵, 陈贵海. 无线传感器网络中节点非均匀分布能量空洞问题[J]. 计算机学报, 2008, 31(2): 253-260.
- [7] 张擎, 柴乔林, 刘玲. 无线传感器网络不等规模节能分簇路由算法[J]. 计算机工程, 2008, 34(23): 98-101.
- [8] 杨军, 张德运. 非均匀分簇的无线传感器网络数据传送机制[J]. 西安交通大学学报, 2009, 43(4): 14-17.