

基于可预测移动汇聚节点的无线传感网分簇算法研究

章 韵^{1,3} 巨德文¹ 陈 志^{1,2,3,4} 扈罗全⁵ 岳文静⁴

(南京邮电大学计算机学院 南京 210003)¹

(南京大学计算机软件新技术国家重点实验室 南京 210093)²

(江苏省无线传感网高技术研究重点实验室 南京 210003)³

(宽带无线通信与传感网技术教育部重点实验室 南京 210003)⁴

(苏州大学城市轨道交通学院 苏州 215104)⁵

摘 要 在汇聚节点移动可预测情况下,提出一种无线传感网分簇算法。该算法将 subsink 节点引入到 HEED 分簇算法中,以较快感知移动路径变化,快速形成分簇拓扑;采用 sink 节点注册机制,实现汇聚节点移动过程中的信息交互。实例分析表明,该算法能快速形成合理网络拓扑,延长无线传感网的生存期。

关键词 无线传感网,移动 sink,分簇算法

中图法分类号 TP393 **文献标识码** A

Research on Clustering Algorithm of Wireless Sensor Networks Based on Predictable Mobile Sink

ZHANG Yun^{1,3} JU De-wen¹ CHEN Zhi^{1,2,3,4} HU Luo-quan⁵ YUE Wen-jing⁴

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)¹

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)²

(Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China)³

(Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, Nanjing 210003, China)⁴

(School of Urban Rail Transportation, Soochow University, Suzhou 215104, China)⁵

Abstract A clustering algorithm for wireless sensor networks was presented under the predictable mobility condition of the sink node. The proposed algorithm introduces subsink node into the HEED clustering algorithm to achieve faster sensing of mobile track changes and quick formation of clustered topologies, and uses the sink node registration to achieve the information exchange in the sink node mobile process. Case analysis shows that the algorithm can quickly form a rational network topology and prolong the lifetime of wireless sensor networks.

Keywords Wireless sensor networks, Mobile sink, Clustering algorithm

1 引言

无线传感网由若干部署在监测区域内的传感器节点组成,节点将感知的数据收集起来以多跳的方式传输给汇聚节点 sink,最后通过汇聚节点将其发送给管理节点,以实现数据采集和任务监测。文献[1-3]指出离 sink 越近的传感器节点能量消耗也越快,这种现象被称为在 sink 周围形成的“能量洞”。文献[4-6]研究了支持移动 sink 节点的路由算法, sink 节点移动收集数据减少了对最后一跳节点的依赖,将数据传递任务分担到更多的节点以缓解“能量洞”问题。文献[4,5]

研究的路由算法基于平面拓扑结构,与基于分簇的拓扑结构相比有如下优点:可保持网络的拓扑结构相对稳定,降低节点移动带来的影响,由簇头和网关节点作为骨干节点进行通信,普通节点在没有数据时进入休眠状态节省能量。簇头选举有两类方法,即集中式与分布式,文献[6]指出基于综合因素的分布式选举簇头算法优化了簇头选举方式和簇头的分布。文献[7]指出 HEED 是一种能量高效、分布式的分簇算法。文献[8]指出 HEED 与 LEACH 等算法相比,其形成的簇大小均匀分布合理,但没有考虑 sink 移动的情况。本文研究在 sink 节点移动的情况下对 HEED 分簇算法进行改进,以形成

到稿日期:2011-07-21 返修日期:2011-11-02 本文受国家自然科学基金项目(60905040),江苏省基础研究计划(自然科学基金)项目(BK2011756),江苏高校优势学科建设工程资助项目(yx002001),江苏省博士后科研资助计划项目(1101006B,1102102C),江苏省高校自然科学基金计划(11KJB510018),南京邮电大学科研基金项目(NY210011,NY211009),江苏省出入境检验检疫局科研项目(2011KJ19),苏州市科研项目(SWG0913 SYJG0925)资助。

章 韵(1963—),男,博士,教授,硕士生导师,主要研究方向为计算机在通信中的应用等,E-mail:zhy@njupt.edu.cn;巨德文(1987—),女,硕士生,主要研究方向为计算机应用技术等;陈 志(1978—),男,博士后,副教授,硕士生导师,主要研究方向为无线传感网、Agent、软件工程等;扈罗全(1972—)男,博士,高级工程师,主要研究方向为无线通信与电磁兼容等;岳文静(1982—),女,博士后,讲师,主要研究方向为认知无线电、无线传感网等。

更加合理有效的拓扑,通过引入 subsink 节点,并在其一跳范围内选取簇头节点,能够较快地适应移动情况,调整拓扑结构。

2 系统模型

在待检测区域内随机布设若干传感器节点,假设 sink 节点沿预先设置的路径周期移动。

2.1 节点移动路径

假设节点的分布区域为正方形^[9], R_1 为边长 L_1 的正方形, R_2 为外层边长 L_2 的正方形,数据监控中心点为 O ,如图 1 所示。在分布区域中,节点为均匀分布,设密度为 ρ ,单位数据的平均传输能量为 e ,则 R_1 内所有节点能量消耗为 $s_{R_1} \times \rho \times e$ 。如果要使总能量消耗最少,就要使 $s_{R_1} \times \rho \times e = (s_{R_2} - s_{R_1}) \times \rho \times e$,即 $L_1^2 = L_2^2 - L_1^2$,可解得 $L_1 = \sqrt{2} L_2 / 2$ 。故当 sink 节点沿检测中心为 O 、边长为 $\sqrt{2} L_2 / 2$ 的正方形移动时,所消耗的能量最少。

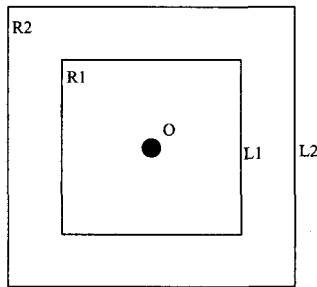


图 1 节点移动路径

2.2 节点移动模型

节点移动模型采用文献[11]中提出的模型并对其做改进,如图 2 所示。

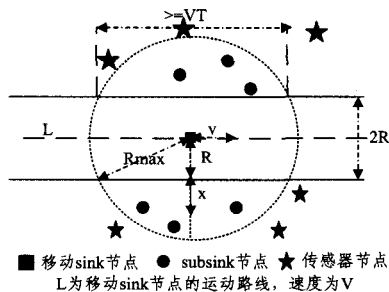


图 2 节点移动模型

节点的移动速度为 V ,发射半径为 R_{max} ,道路宽度为 $2R$ 。第一次周期运动时,将移动节点的发射半径、运动速度、进入时间 T_{start} 和离开时间 T_{end} 告知其发射半径 R_{max} 内的节点(将该类节点命名为 subsink 节点)。假设 subsink 节点将数据发送给 sink 节点所需的时间为 T ,为使数据能成功地接收,必须保证 sink 在 T 时间段内位于 subsink 的发射范围之内。从图 2 中得知,每个 subsink 节点向 sink 传输数据都有一个最大等待时间。

从图 2 中可计算出最大等待时间:

$$T_{max_wait} = \frac{2\sqrt{R_{max}^2 - (R+x)^2}}{V} - VT$$

式中, $R+x$ 为 subsink 节点距离 sink 的最短距离。

3 分簇算法

本文引入 subsink 节点并将其作为向 sink 节点转发数据

的最后一跳节点。选取 subsink 节点一跳范围内的节点为 subsink 所在簇的簇头,那么当 sink 路径发生变化时,subsink 节点选取随之改变,则簇头也会相应发生变化。这样能够快速地对移动路径的变化,及时调整拓扑,并且能够缩短簇头与 subsink 节点的通信距离,节省能量。

3.1 簇的划分

HEED 分簇算法以当前簇头的剩余能量和簇内通信代价作为成簇因素。首先根据这两个因素来选举若干初始簇头,然后通过逐步迭代的方法来选举出最终簇头并形成簇。算法优点:产生的簇大小均匀,拓扑比较合理。缺点:算法在成簇阶段要广播大量消息,没有考虑网络之间的连通性问题。改进:引入连通性^[12]来确定初始簇头,能够提高簇内节点连通性;同时令簇头的选举概率每次以 4 倍增加来减少迭代次数,使得算法快速收敛,降低算法计算成本。

subsink 节点所在的簇是直接 sink 节点通信的最后一簇,因此需要特殊处理。在网络初始运行时各个节点能量相差无异,在选取初始簇头时除了要考虑算法成簇因素中的簇内通信代价和连通度,还要考虑初始簇头节点与 subsink 节点的连通性。其他不包含 subsink 节点的簇,则按照我们提出的算法进行分簇处理。综合考虑这 3 个因素可选取 subsink 节点一跳范围内满足要求的节点作为初始簇头。

算法用到的定义:

定义 1(簇头集合, S_{CH}) 邻居节点集合 $S_{nbr} = \{\text{节点编号 ID, 剩余能量 } E_{left}, \text{距离}\}$; subsink 节点具有特殊标识 sub_ID, 不作簇头,可减少收集簇内数据、融合数据等的能量消耗。

定义 2(簇头声明, cluster_head_msg, ID, selection status, cost) selection status 为选举状态,节点有两种状态: tentative_CH 和 final_CH, cost 为簇内通信代价。

定义 3 连通度 CPS 指成员节点之间的信号强度与两节点在一起时的最强信号强度的比值, $CPS(i) = (R - D(i, H(i))) / R$ 。其中, R 表示节点探测半径, $H(i)$ 表示节点 i 的簇头, $D(i, H(i))$ 表示连通可靠度,这里用距离来表示。当节点与其簇头在一起时, $D(i, H(i)) = 0$, $CPS(i) = 1$ 。在 subsink 一跳范围内的节点与 subsink 连通度 CPS 值较大,连通性好。

簇的划分步骤如下:

步骤 1 确定簇头

(1)节点初始状态为空闲。节点初始化,设置分簇时间为 T_c ,建立一个簇头节点的百分比 C_{prob} (假设为 5%)并计算自己成为簇头的百分比 $CH_{prob} = C_{prob} \times \frac{E_{left}}{E_{max}}$ 。节点使用低功率发射半径 r 来探测并记录直接邻居节点,高功率发射半径 R 用来建立簇头与簇头之间的联系。建立自己的邻居列表 S_{nbr} ,统计其直接邻居节点数目 N_{nb} 。用 subsink 的低功率发射半径 r 作为一跳范围来寻找此范围内的节点。

(2)节点比较自己邻居数和直接邻居的邻居数,选择具有最大邻居数的节点作为临时簇头,即如果节点在 subsink 节点的一跳范围之内,则选举自身为临时簇头,并加入初始簇头集合 S_{CH} 中。

(3)在初始的簇头集合中,产生了过多的簇头,簇平均规模小,并且有的节点周围存在多个临时簇头。因此要进一步筛选簇头,进入迭代过程,合理划分簇。临时簇头节点根据自身的剩余能量和连通度来计算权值 pri :

$$pri(i)=CH_{prob}(i)+Nc(i)*\sum cps(Nbs(i))$$

式中, $Nc(i)$ 是以 i 为簇头的节点个数。根据定义 3, 距离 subsink 节点越近的节点, 其连通度 CPS 越大, 权值也越大。那么在节点能量和簇内通信代价都相当的情况下, subsink 节点一跳范围内的临时簇头比一跳范围之外的相邻临时簇头成为簇头的概率要大, 这样就保证了一跳范围内的节点可以优先成为最终簇头, 即能够快速完成迭代形成合理拓扑。如果在 subsink 一跳范围内未找到合适的初始簇头, 由于簇内通信代价过大或者一跳范围的节点个数很少, 因此可适当放大 subsink 的发射功率来扩大一跳范围寻找合适节点。

迭代过程如下:

Step 1 如果节点 i 的当前簇头 $H(i)$ 为临时簇头, 且 $H(i)$ 的邻居节点中临时簇头数为 0 或 1, 这表明在 $H(i)$ 的邻居中只有自己是簇头或者只有一个邻居是簇头, 那么可以直接把自己标识为簇头 $final_CH$, 令 $CH_{prob}=1$, 退出迭代;

Step 2 将 S_{CH} 中节点的权值最大的节点 $most_pri(i)$ 选出作为 $my_cluster_head$, 如果 $CH_{prob}<1$, 则将节点状态标识为 $tentative_CH$, 进入下一步; 否则将节点状态标识为 $final_CH$, 退出迭代;

Step 3 $CH_{previous}=CH_{prob}$;

Step 4 $CH_{prob}=\min(CH_{prob}*4, 1)$; 继续迭代, 直到 $CH_{previous}=1$;

(4) 标识为 $final_CH$ 的节点被选举为簇头节点。

步骤 2 簇建立

(1) 簇头节点广播成簇命令 $cluster_head_msg$ 给在覆盖半径 r 内的节点, 节点接收到这个消息, 则给它的簇头节点发送确认消息 Ack。如果一个节点接收到多个成簇命令, 那么选择距离最近的作为簇头节点。

(2) 簇头节点可以从节点的加入消息中获得相邻簇首的消息, 然后簇头发送通知消息给它所有的邻近簇头, 这样所有簇头都知道彼此的存在。

(3) 每个簇可包含若干候选 subsink 节点, 但只能选择一个 subsink 作为转发节点。簇头根据 subsink 的剩余能量和任务量来选取转发节点, 平衡 subsink 的负载以延长寿命。

步骤 3 簇轮换

经过一轮簇运行时间和网络运行时间之后, 簇头将解散簇, 进入新一轮簇划分。

3.2 移动 sink 的簇注册机制

在 sink 移动的同时, 设定发送请求传输数据信号 $search_subs$, subsink 接收信号并计算 sink 在其范围内的时间 T_s ($T_s=T_{end}-T_{start}$), 发出 Request To Register(RTR) 消息向 sink 请求注册, 并将缓存的数据发送给 sink。sink 一旦接收到 RTR 消息就停止发送 $search_subs$ 消息包, 发送 Agree To Register(ATR) 给 RTR 中相匹配的 subsink, 并接收数据。如果 subsink 在 sink 进入时间 T_{start} 后等待了时间 T_{max_wait} 仍未接收到 $search_subs$ 信号, 则放弃本次传输数据。

4 实例分析

通过实例分析来论证本文提出的分簇算法。图 3 中示意分布若干节点在待检测区域, 图 4 选择 sink 移动的一侧节点分布进行分析, 每个节点均具有自己的编号, 并且可以得到邻

居节点的距离。

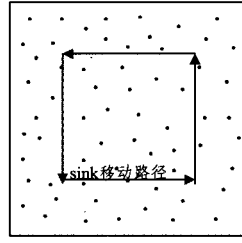


图 3 节点分布示意图

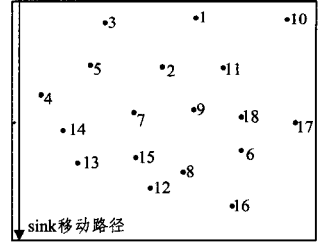


图 4 移动路径一侧分布图

图 5 和图 6 给出节点沿预设路径移动时原 HEED 算法与改进算法的分簇结果。

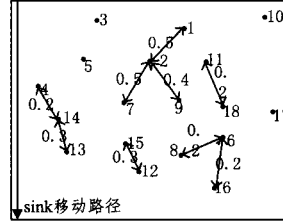


图 5 HEED 分簇图

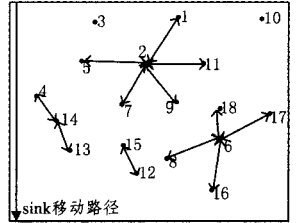


图 6 改进的 HEED 分簇图

表 1 中详细计算了两种算法分簇所需的各项参数值。HEED 和改进后的算法相比: 首先前者初始簇头平均个数为 1, 后者为 0.5, 有效缩小了簇头的选择范围, 使簇头在一个合理的范围内选择, 因为改进之后的分簇算法在初始分簇时引入了连通度, 故初始簇头选择比较合理; 原算法形成的簇个数过多, 簇结构比较零散, 而改进的算法形成的簇个数为 6, 形成的簇结构大小合理, 网络连通性好; 其次在选举 $final$ 簇头时令簇头的选举概率每次以 4 倍增加, 虽然相比原算法的 2 倍概率增加提高了选举误差, 但是由于事先引入连通度保证了 $final$ 簇头是在合理范围内选举, 故准确度不会降低, 并且该算法可以经过较少次数的迭代选举出最终簇头, 前者的平均迭代次数为 5.89, 后者为 1.83。算法得以快速收敛。

在图 5 中, 距离 sink 最近的簇头为节点 14。我们选择比较有代表性的最短路径路由选择算法来确定各个簇头节点的下一跳, 那么所有的簇头节点都会将节点 14 作为向 sink 发送数据的最后一跳。图 6 中, 节点 4 和节点 13 是距离移动路径最近的两个节点, 根据节点移动模型的定义, 它们被选作 subsink 节点, 共同作为向 sink 发送数据的最后一跳。各簇头节点选择下一跳时, 将 subsink 节点的剩余能量和任务量作为标准, 均衡两者的负载。最后一跳节点的生存期可决定整个网络的生存期, 故计算第一轮数据传输过程中节点 14 与节点 4、13 的能量消耗, 以此作为网络生存期的判断依据。根据文献[13]中的模型, 发送 l 比特数据经过距离 d , 节点所消耗的能量和接收 l 比特数据所消耗的能量分别为 $E_t(l, d)=E_{elect}+\epsilon_{amp}d^2$, $E_r(l)=lE_{dec}$, 其中 ϵ_{amp} 为传输放大器的功率能耗, 典型值为 $10\text{pJ}/(\text{bit}\cdot\text{m}^{-2})$, E_{dec} 为发送或接收 1 比特所消耗的能量, 典型值为 $90\text{pJ}/\text{bit}$ 。假设图 4 中每个节点产生 1 比特数据, 节点 4、13 均发送数据, 那么节点 14 和节点 4、13 的能量消耗为:

$$E_{14}=E_t(18, 0.4)+E_r(18)=3241.6$$

$$E_4=E_t(9, 0.1)+E_r(9)=1620.1$$

$$E_{13}=E_t(9, 0.2)+E_r(9)=1620.4$$

表1 算法分簇过程参数值

节点 ID	直接邻居节点	HEED		
		初始簇头	迭代次数	最终簇头
1	2 11	1	6	2
2	1 5 7 9 11	2	5	2
3	5	3	7	3
4	5 14	4	6	14
5	2 3 4	5	6	5
6	8 16 17 18	6	5	6
7	2 9 14 15	7	6	2
8	6 9 12 16	8	6	6
9	2 7 8 11 18	9	6	2
10	无	10	7	10
11	1 2 9 18	11	6	18
12	8 15 16	12	6	15
13	14 15	13	6	14
14	4 7 13 15	14	5	14
15	7 12 13 14	15	5	15
16	6 8 12	16	6	6
17	6 18	17	6	17
18	6 9 11 17	18	6	18
平均	2.95	1	5.89	0.5

节点 ID	改进后的 HEED			
	初始簇头	权值	迭代次数	最终簇头
1	2	4.85	2	2
2	2	4.85	1	2
3	3	0.05	1	3
4	14	3.05	2	14
5	5	0.05	2	2
6	6	1.65	1	6
7	2	4.85	3	2
8	6	1.65	2	6
9	2	4.85	2	2
10	10	0.05	1	10
11	18	0.85	2	2
12	15	0.75	2	15
13	14	3.05	2	14
14	14	3.05	1	14
15	15	0.75	1	15
16	6	1.65	2	6
17	17	0.05	2	6
18	18	0.85	4	6
平均	0.5	2.05	1.83	0.33

subsink 节点 4、13 共同作为最后一跳发送数据,可以减少节点的能量消耗速度,延长存活时间,进而延长网络寿命。如图 7 所示,根据本文提出的算法来计算网络中出现死亡节点的百分比和网络的运行时间(以网络运行轮数为单位)。

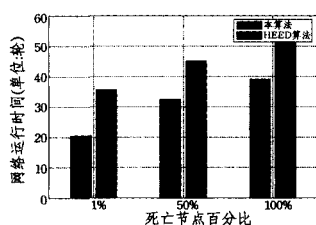


图7 死亡节点百分比图

起初各个节点的能量是充足的,随着网络运行轮数的增加,死亡节点的百分比急剧增加。但是本算法比 HEED 算法延长了网络生存期,这是因为本算法采用了 subsink 节点来共同分担数据转发的任务,避免了簇头节点作为网络的骨干节点既是数据收集和汇聚节点,又是数据转发节点的高任务量而导致的能量过快消耗。subsink 节点作为向 sink 节点转发数据的最后一跳节点,其平衡了网络流量和负载,延长了网络生存期。

结束语 本文将 subsink 概念引入到移动 sink 环境中,通过 subsink 节点感知移动 sink 路径变化,从而使得网络分簇拓扑能够产生相应的变化,分簇算法可以适应移动环境;同

时在已有成熟算法 HEED 的基础上改进连通度,获取了较好的分簇效果。本文给出实例分析,结果发现利用改进的分簇算法能够使算法快速收敛形成合理的拓扑,降低了算法分簇成本,subsink 节点作为数据转发节点,使网络生存时间更长。

参考文献

- [1] Olariu S, Stojmenovic I. Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting[C]// Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM 2006). New York: IEEE Communications Society, 2006:1-12
- [2] Lian J, Naik K, Agnew G. Data capacity improvement of wireless sensor networks using non-uniform sensor distribution[J]. Journal of Distributed Sensor Networks, 2006, 2(2): 121-145
- [3] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An application-specific protocol architecture for wireless micro sensor networks[J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 660-670
- [4] Jea D, Somasundara A, Srivastava M. Multiple controlled mobile elements(data mules) for data collection in sensor networks[C]// Proceedings of the First IEEE international conference on Distributed Computing in Sensor Systems. Heidelberg: Springer-Verlag Berlin, 2005: 244-257
- [5] Gao S, Niu Y C, Huo H W, et al. An energy efficient communication protocol based on data equilibrium in mobile wireless sensor network[C]// Proceedings of the 3rd international conference on Mobile ad-hoc and sensor networks. Heidelberg: Springer-Verlag Berlin, 2007: 433-444
- [6] 成鹭, 成庚民. 无线传感器网络中基于综合因素的分布式路由算法[J]. 计算机科学, 2009, 36(9): 1-2
- [7] Younis O, Fahmy S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks[J]. IEEE Transaction on Mobile Computing, 2004, 1(3): 366-379
- [8] Youssef M A, Younis M F, Aisha K A. A Constrained shortest-Path Energy-Aware Routing Algorithm for Wireless Sensor Networks[C]// Proceedings of 2002 IEEE Wireless Communications and Networking Conference. 2002(2): 794-799
- [9] 周新莲. 基于分簇技术的移动无线传感器网络数据收集协议研究[D]. 长沙: 中南大学, 2010
- [10] Bi Y Z, Niu J W, Sun L M, et al. Moving schemes for mobile sinks in wireless sensor networks[C]// Proceedings of the 26th IEEE International Performance Computing and Communications Conference(IPCCC2007). Louisiana: IEEE Computer Society, 2007: 101-108
- [11] Chakrabarti A, Sabharwal A, Aazhang B. Using predictable observer mobility for power efficient design of sensor networks[C]// Proceedings of the 2nd international conference on Information processing in sensor networks. Heidelberg: Springer-Verlag Berlin, 2003: 129-145
- [12] 于继明. 无线传感器网络基于分簇的多路径路由算法研究[D]. 南京: 南京理工大学, 2008
- [13] Manish B, Timothy G, Anantha P C. Upper bounds on the lifetime of sensor networks[C]// Proceedings of 2001 IEEE International Conference on Communications (ICC 2001). Helsinki: MIT Press, 2001: 785-790