

基于分层成簇改进无线传感器网络 iHEED 协议

尹安 汪秉文 胡晓娅

(华中科技大学控制科学与工程系 武汉 430074)

摘要 分簇思想是设计能量高效的无线传感器网络路由协议的一种重要方法。在分析 iHEED 协议的基础上,通过仿真实验和理论分析,指出 iHEED 协议可能出现“成簇父节点丢失”现象。针对该现象提出并在 TinyOS 上实现了 iHEED-CHLevel 分簇路由协议,引入“分层成簇”思想对 iHEED 协议进行了改进。TOSSIM 模拟仿真表明,iHEED-CHLevel 协议是有效的,它实现了层次分簇,保证了簇间多跳数据传输链路的建立。

关键词 无线传感器网络,iHEED,iHEED-CHLevel,分层成簇,TOSSIM

Improvement of iHEED Protocol in Wireless Sensor Network Based on Clustering by Node Level

YIN An WANG Bing-wen HU Xiao-ya

(Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract Clustering is an important method to design the energy efficient routing protocols in wireless sensor networks. This paper analyzed the principles of the iHEED protocol and pointed out the “parent node loss after clustering” phenomenon by simulation and theoretical analysis. For improving the iHEED protocol, the “clustering by node level” was introduced and implemented on TinyOS named iHEED-CHLevel protocol. From simulation using TOSSIM, it shows that the iHEED-CHLevel protocol is effective on clustering by node level and guarantees the setup of routing tree among the cluster heads for aggregated data transfer.

Keywords Wireless sensor networks, iHEED, iHEED-CHLevel, Clustering by node level, TOSSIM

无线传感器网络(Wireless Sensor Network, WSN)通常是由部署在监测环境内的传感器节点以自组织的方式形成的一种无线网络,可应用于安全监控、环境监测、结构健康监测、军事等领域,具有广阔的应用前景。与传统有线网络相比,WSN中的节点能量有限,且在部署后不易进行再充电,因此高效利用节点已有能量,延长网络寿命,成为了WSN路由协议设计的重点^[1,2]。分簇机制是设计能量高效的WSN路由协议的一种重要技术。在分簇路由协议中,网络被划分为多个簇,每个簇由一个簇头节点和若干个簇成员节点组成。簇头节点负责收集簇成员节点得到的数据信息,经过处理(如聚合操作等)发送给其他簇头或汇聚节点(sink节点,亦称为基站),典型代表有LEACH^[3], HEED^[4,5], PEGASIS^[6]等,以及改进的协议如LEACH-C^[7], MELEACH^[8], MECH^[9]等。但这些协议除HEED外并没有转化为可部署于传感器节点中的代码进行实际测试和应用,而仅在理论上对WSN路由进行了优化研究。Ossama Younis将HEED分簇算法集成到TinyOS的多跳路由协议中,实现了iHEED^[10]路由协议。根据目前相关文献资料,iHEED是首个在TinyOS^[11]操作系统上实现的并可部署于Mica节点中进行实际应用的分簇路由协议,对无线传感器网络的工程应用具有很好的指导意义。

本文在分析iHEED路由协议的工作原理后,在TOSSIM

仿真平台下发现在多跳网络中iHEED分簇算法有可能出现“成簇父节点丢失”现象,并对此开展了研究,提出了改进的iHEED-CHLevel路由协议。本文第1节分析了HEED分簇算法的工作原理;第2节论述了iHEED在TinyOS上的实现;第3节分析了iHEED分簇算法中可能出现的“成簇父节点丢失”现象并提出了改进协议iHEED-CHLevel;第4节通过实验仿真验证了所提iHEED-CHLevel协议的性能;最后给出小结。

1 HEED分簇算法

HEED采用“轮”的概念,在每轮中使用主、次两个参数来选择簇头。其中,主参数依赖于节点的剩余能量,次参数依赖于簇内通信代价,HEED使用簇内平均可达能量(AMRP)作为衡量簇内通信代价的标准,在簇重叠区域中的节点根据次参数选择最终加入哪个簇。具体算法如下^[4]:

算法1 HEED protocol pseudo-code

- (1) Initialize
 1. $S_{nbr} \leftarrow \{v: v \text{ lies within my cluster range}\}$
 2. Compute and broadcast cost to $\in S_{nbr}$
 3. $CH_{prob} \leftarrow \max(C_{prob} * E_{residual} / E_{max}, P_{min})$
 4. $is_final_CH \leftarrow FALSE$
- (2) Repeat

到稿日期:2009-08-14 返修日期:2009-10-28 本文受国家自然科学基金项目(60773190,60802002)资助。

尹安(1980—),博士生,主要研究方向为无线传感器网络和应用技术,E-mail:ibm_hust@yahoo.com.cn;汪秉文(1946—),教授,博士生导师,主要研究方向为无线网络技术、生产过程综合自动化系统集成技术;胡晓娅(1974—),副教授,主要研究方向为无线传感器网络、网络化控制系统。

```

1. If ((SCH ← {v: v is a cluster head}) ≠ ∅)
2.   my_cluster_head ← least_cost(SCH)
3.   If my_cluster_head == NodeID
4.     If (CHprob == 1)
5.       Cluster_head_msg (NodeID, final_CH, cost)
6.       is_final_CH ← TRUE
7.     Else
8.       Cluster_head_msg(NodeID, tentative_CH, cost)
9.   ElseIf (CHprob == 1)
10.    Cluster_head_msg(NodeID, final_CH, cost)
11.    is_final_CH ← TRUE
12.  ElseIf Random(0,1) ≤ CHprob
13.    Cluster_head_msg (NodeID, tentative_CH, cost)
14.    CHprevious ← CHprob
15.    CHprob ← min(2 * CHprob, 1)
16.  UNTIL CHprevious = 1
(3) Finalize
1. If (is_final_CH == FALSE)
2.   If ((SCH ← {v: v is a final CH}) ≠ ∅)
3.     my_cluster_head ← least_cost(SCH)
4.     join_cluster(cluster_head_ID, NodeID)
5.   Else Cluster_head_msg((NodeID, final_CH, cost)
6. Else Cluster_head_msg((NodeID, final_CH, cost)

```

在每轮开始时,节点首先初始化 CH_{prob} , 设置节点成为簇头的初始概率 $CH_{prob} = \max(C_{prob} * E_{residual} / E_{max}, P_{min})$, 其中 $E_{residual}$ 为节点当前剩余能量, E_{max} 为节点初始能量, C_{prob} 用来限制初始簇头申明的数目。规定 CH_{prob} 最小值为 P_{min} , 防止簇头选举迭代收敛速度太慢。

在每轮的迭代过程中,各节点从本地候选簇头集合 S_{CH} 中寻找最低通信代价的节点作为簇头,候选簇头集合 S_{CH} 为第 $i-1$ 次迭代后选举产生的簇头加上第 i 次迭代产生的簇头,若节点候选簇头集合为空,则依照概率 CH_{prob} 选举自身成为簇头。在每个迭代过程结束后, $CH_{prob} = \min(2 * CH_{prob}, 1)$, 进入下一迭代过程,当 CH_{prob} 等于 1 时,中止迭代。

在最终迭代阶段结束后,节点选择成为簇头(节点簇头标志 $is_final_CH = TRUE$)或者根据接收到的簇头申明(节点簇头标志 $is_final_CH = FALSE$),在候选簇头节点中选择最低通信代价的节点作为簇头,加入该簇;如果最终迭代过程结束后,节点的候选簇头集合为空,则申明节点自身作为簇头节点。

2 iHEED 在 TinyOS 中的实现

Ossama Younis 将 HEED 算法集成在 TinyOS 中的多跳路由协议中,实现了基于 TinyOS 操作系统的可部署于 Mica 系列节点的分簇路由协议 iHEED。iHEED 的模块框图如图 1 所示^[10]。

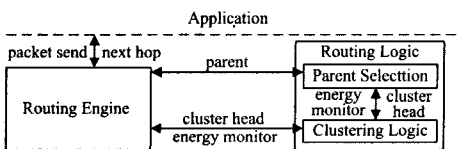


图 1 iHEED 协议框图

(1) Routing Engine 模块:簇头负责监听簇内节点发送的数据,完成聚集操作,再将数据包发送到下一跳目的节点。

(2) Routing Logic 模块:负责提供路由算法,包含两个子模块:1) Clustering Logic:实现分簇算法,并在簇头间构建路由树,在分簇中使用 CREP^[10]方法估算节点每轮剩余能量。分簇算法代码设计的核心是使用两个定时器触发分簇操作,其中 Timer2 为分簇触发定时器,Timer3 为每轮成簇过程中的迭代触发定时器;2) Parent Selection:节点基于 LEPS(Link Estimation and Parent Selection)^[12]算法选择下一跳目的节点,在 iHEED 协议中使用 Timer1 触发发送路由更新包和新路由的计算。

3 iHEED-CHLevel 分簇路由协议

3.1 相关定义和网络模型

在描述 iHEED-CHLevel 分簇路由协议之前,首先给出以下定义。

定义 1 节点 v_i 所在网络层次 v_i , $Level = v_i$, $LocalHop$, 即节点 v_i 所在网络层次定义为节点 v_i 到 sink 节点的跳数(本文所提的跳数都是指到目的节点的最小跳数)。

定义 2 第 i 层节点集合 S_i 定义为 $S_i = \{v_i | v_i, LocalHop = i, i = 1, 2, 3, \dots\}$ 。

定义 3 簇头节点 v_i 的簇成员节点集合 $ClusterMember(v_i) = \{v_k | v_k, ClusterHead = v_i\}$ 。

本文研究的 WSN 网络抽象为一个有向图模型 $G(V, E)$, 其中 V 表示节点的集合, E 表示节点间链路的集合,网络具有以下特点:

(1) 传感器节点部署后处于静止状态。

(2) 传感器节点间为双向链路。

(3) 传感器节点传送功率可调。传感器节点以功率 P_i 对应的发射半径 $R_{inter-cluster}$ 进行簇内通信,簇头节点以功率 P_i 对应的发射半径 R_{th} 进行簇间通信($R_{th} > R_{inter-cluster}$)。在 P_i 对应的发射半径内, $v_i \in S_i$ 可以与 S_{i-1} ($i = 2, 3, 4, \dots$) 集合内的任一节点进行通信, S_1 集合内的节点可以与 sink 节点通信。

3.2 iHEED 中的“成簇父节点丢失”现象

iHEED 协议将 HEED 算法加入 TinyOS 操作系统下的多跳路由协议中,实现了网络分簇及簇间多跳传输。但在 iHEED 协议中有可能出现“成簇父节点丢失”现象,导致网络断裂。下面将给出该现象在本文所研究的网络模型下产生的一个场景并加以证明。

若 $\forall v_i \in S_1, v_j \in S_2, \exists (v_i, v_j) \in E$, 那么在迭代过程结束后,选举的最终簇头 $v_k \in S_2$, 则出现“成簇父节点丢失”现象。

证明:因为 S_1, S_2 两节点集合中的任一节点都可以通信,最终簇头 $v_k \in S_2$, 则有 $\forall v_i \in S_1, v_i \in ClusterMember(v_k)$ 。

又因为 $v_k, parent \in S_1$, 但 $\forall v_i \in S_1, v_i \in ClusterMember(v_k)$, 故 v_k 父节点丢失,聚合数据无法通过下一跳节点 $v_i \in S_1$ 传送回 sink 节点,网络断裂。

本文使用 TOSSIM 仿真工作验证了 iHEED 中“成簇父节点丢失”现象的存在。使用文献[10]中的网络拓扑结构, $S_1 = \{1, 2, 3\}$, $S_2 = \{4, 5, 6\}$, S_1 中的节点可与 sink 节点和 S_2 中的任一节点进行通信, S_2 中的节点只能与 S_1 中的节点进行直接通信。 S_2 中的数据必须以 S_1 中的节点为中继,以多跳的形式传送回 sink 节点。仿真使用 seed=10000 作为程序运行的随机种子数。当第三轮成簇时,选举的簇头为节点 6,这时网络通信拓扑图如图 2 所示。因为节点 6 的父节点为其簇内

成员节点,不参与数据转发,节点6无法从 S_1 中找到下一跳 $CH_type = FINAL_CH$ 的簇头节点中转数据,出现“成簇父节点丢失”现象,网络断裂。

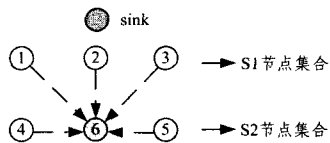


图2 “成簇父节点丢失”场景

针对iHEED中的“成簇父节点丢失”现象,本文提出了iHEED-CHLevel算法,使用“分层成簇”机制解决iHEED多跳路由中的“成簇父节点丢失”现象。下面将描述iHEED-CHLevel算法的具体实现。

3.3 iHEED-CHLevel 算法描述

iHEED-CHLevel协议的核心思想是“分层成簇”,即根据节点所在层次进行簇划分,保证在多跳路由中上一层存在簇头节点参与数据转发。下面对其核心模块进行描述。

网络中的节点在网络拓扑形成后,记录节点到sink节点跳数LocalHop,即该节点所在网络的层次。初始网络拓扑的形成需要各个节点彼此交换路由信息,并根据LEPS机制形成多跳路由树。这个过程需要一段时间,且随网络中节点数目的增加而增加。在实际应用中设置ROUTE_TREE_SET个路由由更新包后,各节点记录其LocalHop。其算法如下。

算法2 iHEED-CHLevel set localHop value

```

1. for i=1 to ROUTE_TREE_SET
2.   SendRouteMsg()
3.   Event on Receive RouteMsg
4.     UpdateNeighborInfo()
5.   End Event
6. End for
7. for  $v_i \in \text{NeighborInfo}$ 
8.   If  $v_i.id == 0$ 
9.     localHop  $\leftarrow 1$ 
10.    break;
11.  else
12.    localHop  $\leftarrow \min(v_i.hop) + 1$ 
13. end for

```

节点 v_i 在发送簇头消息包 $Cluster_head_msg$ 时,包含节点跳数信息 $v_i.LocalHop$ 。邻居节点 v_j 在接收到节点 v_i 发送的 $Cluster_head_msg$ 时,若 $v_j.LocalHop = v_i.LocalHop$,说明节点在同一层次,则节点 v_j 更新本地候选簇头信息 S_{CH} ,即各节点只保留同层次的节点作为候选簇头。在迭代过程中,其他步骤与HEED算法相同。在迭代过程结束后,若节点的候选簇头集合 S_{CH} 不为空,则从候选簇头集合 S_{CH} 中选择最低通信代价的节点作为簇头;若候选簇头集合 S_{CH} 为空,则申明自己为簇头节点。

在每轮分层成簇选举产生的簇头后,各簇头根据LEPS机制选择其下一跳目的簇头节点,在簇间构造多跳路由树。若簇头节点 $v_k \in S_1$,则其下一跳目的节点为sink节点。簇头节点 v_k 在聚合簇内成员节点数据后再转发给下一层次簇头节点,最终将数据传送回sink节点。

根据iHEED-CHLevel算法,可知在本文所研究的网络模型下,iHEED-CHLevel分簇后的簇间链路必连通。

证明:根据iHEED-CHLevel簇头选择算法, $\forall S_i, \exists v_i$

为簇头节点, $v_i \in S_i$ 。

又因为根据所研究的网络模型, v_i 在功率 P_i 对应的通信半径下,可以与 S_{i-1} 中的任一节点进行通信。

所以 v_i 可以找到至少一个 v_{i-1} 为簇头节点, $v_{i-1} \in S_{i-1}$ ($i=2,3,\dots,\maxHop$),作为下一跳中继节点转发数据。当 $v_i \in S_1$ 时,其下一跳目的节点为sink节点,所以iHEED-CHLevel算法下网络必连通。

4 实验仿真

本文在TinyOS平台上使用nesC^[13]语言实现了iHEED-CHLevel算法,并利用TOSSIM验证了算法的可靠性。TOSSIM是一个专门用来模拟TinyOS应用程序的仿真器,可以将要烧录进节点的应用程序编译为可在用户PC机上运行的代码,供用户在节点实际部署前在本机模拟网络环境,测试代码性能^[14]。

在仿真中使用nss文件定义图2所示的网络拓扑结构,测试协议分簇多跳路由性能。仿真参数如表1所列。

表1 仿真参数设置

参数名	参数值
路由数据包发送间隔	10000 ms
分簇间隔	250000ms
每轮分簇迭代间隔	11000ms
Cprob	0.03
ROUTE_TREE_SET	8

选取随机种子数 $seed = 10000$, S_1 的簇头节点完成其他各节点发送数据融合后,再发送到sink节点。统计sink节点收到的数据包数,如图3所示。

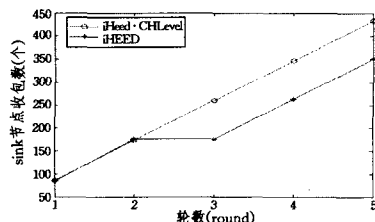


图3 sink节点收包数统计

从图3可以看出,iHEED在第3轮中sink节点没有收到数据包,此轮中网络出现断裂现象,数据包无法传送回sink节点。在iHEED-CHLevel协议下sink节点都能收到簇头发送的数据包,在每轮中网络都始终处于连通状态。

随机选取不同的种子数进行5组仿真。从每组中抽取5轮,统计簇间连通成功率(分簇后各簇聚合数据是否能传送回sink节点)。簇间连通成功率如图4所示。

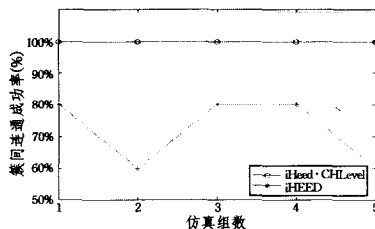


图4 簇间连通成功率

由图4可知,在iHEED-CHLevel协议下,网络在任何一次分簇情况下都能保证每层存在簇头节点,簇头间可以构建到sink节点的传送路径,将数据传送回sink节点,这与算法

理论分析相符。iHEED 在多跳情况下,可能出现“分簇父节点丢失”现象,导致网络断裂,聚合数据无法传回 sink 节点,在 5 组仿真情况下,其分簇后网络连通的成功率最高只有 80%。通过仿真实验,证明了“分簇父节点丢失”现象的存在。

结束语 本文通过理论分析和实验仿真说明了 iHEED 协议中可能出现的“分簇父节点丢失”现象,提出了改进协议 iHEED-CHLevel。iHEED-CHLevel 采用分层成簇算法,保证了簇间在多跳情况下的连通,并通过在 TOSSIM 上模拟运行,验证了 iHEED-CHLevel 算法的可靠性。本文所实现的 iHEED-CHLevel 协议可集成于 TinyOS 下其他多跳路由协议中,对实际工程应用具有一定的指导意义。

参考文献

- [1] Akyildiz I F, Su W, Sankarasubramanian Y, et al. Wireless sensor networks: a survey [J]. *Computer Networks*, 2002; 393-422
- [2] 任丰原, 林闯. 无线传感器网络[J]. *软件学报*, 2003, 14(07): 1282-1291
- [3] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks [C]//*Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. Maui, HI, 2000; 1-10
- [4] Younis O, Fahmy S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks[J]. *IEEE Transactions on Mobile Computing*, 2004, 3(4): 660-669
- [5] Younis O, Fahmy S. Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-efficient Approach[C]// *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*. INFOCOM, 2004; 629-640
- [6] Lindsey S, Raghavendra C, Sivalingam K M. Data gathering algorithms in sensor networks using energy metrics [J]. *IEEE*

Transactions on Parallel and Distributed Systems, 2002, 13(9): 924-935

- [7] Heinzelman W, Chandrakasan A, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks [J]. *IEEE Transactions on Wireless Communications*, 2002, 1(4): 660-670
- [8] 陈静, 沈鸿. MELEACH 一个高效节能的 WSN 路由协议[J]. *传感技术学报*, 2007(12): 2089-2094
- [9] Chang Ruay-shiung, Kuo Chia-jou. An Energy Efficient Routing Mechanism for Wireless Sensor Networks[C]// *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*. 2006; 308-312
- [10] Younis O, Fahmy S. An experimental study of routing and data aggregation in sensor networks[C]// *Proceedings of the International Workshop on Localized Communication and Topology Protocols for Ad hoc Networks*. Washington, DC, 2005; 49-57
- [11] Levis P, Madden S, Polastre J, et al. TinyOS: An operating system for wireless sensor networks[C]// *Ambient Intelligence*. New York, NY: Springer-Verlag, 2005(12): 115-148
- [12] 颜庭莘, 孙利民. TinyOS 路由协议原理及性能评估[J]. *计算机工程*, 2007(1): 112-114
- [13] Gay D, Levis P, Behren R, et al. The nesC language: A holistic approach to networked embedded systems[C]// *Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*. New York, USA: ACM Press, 2003; 1-11
- [14] Levis P, Lee N, Welsh M, et al. TOSSIM: Accurate and scalable simulation of entire tinyos applications [C] // *Proc. of the 1st International Conference on Embedded Networked Sensor System*. ACM Press, 2003; 126-137

(上接第 10 页)

- [7] Vaughn R B Jr, Henning R, Siraj A. Information Assurance Measures and Metrics-State of Practice and Proposed Taxonomy [C]// *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)*. Big Island, Hawaii, 2003
- [8] Seddigh N, Piedad P. Current Trends and Advances in Information Assurance Metrics[C]// *Proceedings of the Second Annual Conference on Privacy, Security and Trust*. Privacy, Security, and Trust, 2004; 197-205
- [9] Lu Xin, Ma Zhi. Information Assurance Evaluation for Network Information Systems[C]// *Computational Intelligence and Security*. Guangzhou, China: Springer, 2006; 869-877
- [10] 吕欣. 信息系统安全保障理论与评价指标体系[J]. *微电子学与计算机*, 2006, 23(10): 10-12
- [11] 郭振民, 胡学龙, 姜会亮. 网络与信息系统安全性评估及其指标体系的研究[J]. *现代电子技术*, 2003(9): 9-11
- [12] 魏忠. 从定性到定量的系统性信息安全综合集成评估体系[J].

系统工程理论方法应用, 2004, 13(5): 478-479

- [13] 肖道举, 杨素娟. 网络安全评估模型研究[J]. *华中科技大学学报*, 2002, 30(4): 37-39
- [14] 黄丽民, 王华. 网络安全多级模糊综合评价方法[J]. *辽宁工程技术大学学报*, 2004, 23(4): 510-513
- [15] 李雄伟, 杨义先, 等. Fuzzy-AHP 法在网络攻击效果评估中的应用[J]. *北京邮电大学学报*, 2006, 29(1): 124-127
- [16] 章文辉, 杜百川, 杨盈昀. 模糊层次分析法在广播电视信息安全保障评价指标体系中的应用研究[J]. *电子学报*, 2008(10): 2060-2064
- [17] 程学东. 电信网络安全评估指标体系研究[J]. *现代电信科技*, 2005(8): 10-13
- [18] 王剑, 彭越, 吴志军. 航空信息系统安全评价指标体系[J]. *信息安全与保密通信*, 2008(1): 24-26
- [19] 中国工业和信息化部. 中国信息安全产品评测指标体系, 技术标准[S]. 深圳: 工业和信息化部旗下中国软件评测中心, 2008; 1-4
- [20] 国家信息中心. 信息安全保障评价指标体系[R]. 2005; 1-20