

文章编号:1000-582X(2012)08-101-06

## 无线传感器网络中改进的 HEED 分簇算法

杨梦宁, 杨 丹, 黄 超

(重庆大学 软件学院, 重庆 400044)

**摘 要:**在无线传感器网络中,基于分簇的路由协议在能量消耗、拓扑控制以及数据融合等方面具有优势。采用完全分布式产生簇头的 HEED 协议具有分簇速度快、簇头均匀等优点。但是 HEED 协议没有考虑网络中节点的移动性,当邻居节点距离发生变化时,通过簇内最小平均功率 (AMRP) 决定节点属于哪个簇会导致簇头能量开销过大,网络生命周期减小等问题。针对 HEED 分簇算法存在的问题,提出了一种基于稳定性的 S-HEED 分簇算法。S-HEED 协议以稳定性作为参数来决定节点的所属簇,解决了因为移动性带来的簇内节点和簇头能量消耗过高问题。仿真实验证明,S-HEED 算法有效的减少了簇头节点的能耗,延长了网络寿命。

**关键词:**分簇算法;无线传感器网络;S-HEED;稳定性;网络寿命

**中图分类号:**TP393

**文献标志码:**A

## An improved HEED clustering algorithm for Wireless Sensor Network

YANG Mengning, YANG Dan, HUANG Chao

(College of Software, Chongqing University, Chongqing 400044, P. R. China)

**Abstract:** In wireless sensor network, routing protocols which based on clustering have the advantages of energy consumption, topology management and data fusion. The HEED protocol, which generates cluster heads based on distributed algorithm, drives up the rate of clustering and creates well-distributed cluster heads. However, it does not consider the mobility of nodes in the network. When the distance between neighbor nodes has changed, the AMRP method which decides the node belongs to different cluster heads would cause problems such as high energy consumption, short lifetime of network and so on. Responding to these problems, the paper proposes the S-HEED, a clustering algorithm based on stability, which chooses the stability as a parameter of nodes when choosing a cluster head. With S-HEED algorithm, the high energy consumption problem among cluster nodes and cluster heads caused by the mobility is tackled. The simulation experiment demonstrates that the S-HEED algorithm lower the energy consumption of cluster heads and prolongs the network lifetime.

**Key words:** clustering algorithm; wireless sensor network; S-HEED; stability; network lifetime

无线传感网络 (wireless sensor network, WSN) 是由部署在监测区域内大量的微型传感器节点组成,利用无线通信方式形成一个多跳的自组织网络系统,其目的是协同感知、采集和处理网络覆盖

区域中感知对象的信息,并发送给观察者<sup>[1]</sup>。

与现有的网络相比,由于无线传感器网络中节点能量有限,因此减少能量消耗、延长网络寿命是无线传感器网络路由协议重点考虑的问题。目前在

收稿日期:2012-02-20

基金项目:国家自然科学基金资助项目(60972105);重庆市自然科学基金资助项目(CSTC2009Bb2364)

作者简介:杨梦宁(1980-),男,重庆大学博士研究生,主要从事数据挖掘、物联网研究,(E-mail)mnyang@cqu.edu.cn。

WSN 路由协议中,主要通过分簇算法来解决。在分簇的拓扑管理机制下,网络中的节点可以划分为簇头节点和成员节点两类。在每个簇内,根据一定的机制算法选取某个节点作为簇头,用于管理或控制整个簇内成员节点、协调成员节点之间的工作、负责簇内信息的收集和融合处理以及簇间转发<sup>[2]</sup>。

针对现有的分簇算法未考虑节点移动性引起的能量消耗变化问题。文中提出了基于节点稳定性的 S-HEED(stable based HEED)分簇算法。节点在选择簇头时,首先计算节点加入簇头代价。若有多个候选簇头,再根据节点稳定性选出最优簇头候选集。最后选择加入平均通信功率最低的簇头。通过仿真实验验证了该算法在移动环境下的节点能耗、网络寿命等都优于传统 HEED 分簇算法。

## 1 经典分簇协议分析

LEACH (low-energy adaptive clustering hierarchy)是 WSN 中最早提出的分簇路由协议,其基本思想是通过等概率地随机循环选择簇头,将整个网络的能量负载平均分配到每个节点,从而达到延长网络生命的目的<sup>[3]</sup>。但是,LEACH 算法中当选簇头节点的阈值没有考虑能量因素且需要假设每个节点初始能量均等,与实际不符。

DCHS 算法(deterministic cluster-head selection)改进了 LEACH 算法的不足,将能量因素考虑到了簇头选取的阈值函数中<sup>[4]</sup>。实验证明,DCHS 分簇算法在 LEACH 算法基础上有效提高网络生命周期 20%~30%。

Heinzelman 等人在 LEACH 算法的基础上提出了 LEACH-C(LEACH-centralized)算法,解决了 LEACH 算法中“节点根据随机数决定是否当选为簇首”以及“每轮产生的簇首没有确定的数量和位置”等问题,大大提高了簇的生成质量<sup>[5]</sup>。但由于每个节点都须向基站周期性地报告它们的能量和位置等信息,成簇开销较大,网络流量、时间延迟以及信号干扰的概率都会增加。

LEACH-F 算法(LEACH-fixed)也是集中式簇头产生算法,由基站负责挑选簇头<sup>[6]</sup>。相对于 LEACH-C 算法,LEACH-F 算法不需要每轮循环构造新的簇,减少了构造簇的开销。但是,由于 LEACH-F 算法不能动态处理节点的加入、失败和移动,并且增加了簇间信号干扰,所以不适合真实的网络应用。

### 1.1 HEED 协议

HEED (hybrid energy-efficient distributed clustering)协议<sup>[6]</sup>是由 Ossama 等人于 2004 年提出

的。簇头的选择主要依据主、次 2 个参数。主参数依赖于剩余能量,用于随机选取初始簇头集合。具有较多剩余能量的节点将有较大的概率暂时成为簇头,而最终该节点是否一定是簇头取决于剩余能量是否比周围节点多得多。考虑到分簇后簇内的通信开销,HEED 以 AMRP(簇内平均可达能量)作为衡量簇内通信代价的标准<sup>[7]</sup>。

HEED 的簇头选择算法具有以下特点:完全分布式的簇头产生方式;簇头产生在有限次迭代内完成;最小化控制报文开销;簇头分布均衡。实验结果表明,HEED 分簇速度更快,能产生分布更加均匀的簇头、更合理的网络拓扑<sup>[8]</sup>。

### 1.2 HEED 协议存在的问题

无线传感器节点模型如图 1 所示。

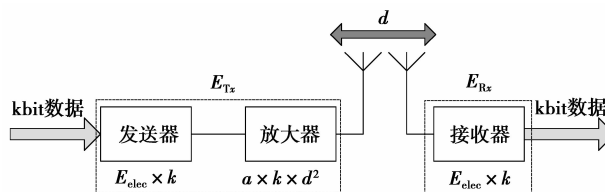


图 1 能量消耗模型

根据传感器节点的收发器能量模型<sup>[9-10]</sup>,1 个 kbit 的分组数据传输距离为  $d$ ,则发送的功耗为  $E_{Tx} = E_{elec} \times k + \alpha \times k \times d^2$ ,接收的功耗为  $E_{Rx} = E_{elec} \times k$ ,其中, $E_{elec}$ 表示发射和接收电路每发送和接收单位 bit 的能耗,单位为 J/bit; $\alpha$ 表示发射放大器将每 bit 传输单位平方米所耗的能量,单位为 J/(bit · m<sup>2</sup>)。可见,无线传感器节点发送的能耗随着发送距离的增加迅速增大。

HEED 协议中,当节点位于多个簇范围内时,选择加入最小平均功率的簇头,如公式(1)所示。

$$AMRP = \frac{\sum_{i=1}^M \min P_{wr_i}}{M}, \quad (1)$$

在实际的应用环境中,传感器节点有可能是移动的,而 HEED 协议中的最小平均功率忽略了节点与簇头以及簇头与 sink 节点之间的距离,会导致以下 2 个问题<sup>[12]</sup>:

1)某簇头距离 sink 节点较远而其最小平均功率较低,加入该簇的节点较多,导致该簇头处理的数据量较大,能耗开销较高。

2)节点采用最小平均功率来决定加入簇,只是针对当前时间点选择了最优的簇头节点,而未考虑节点后续的“移动趋势”。若节点正远离该簇头节点移动,根据能量模型可知数据发送过程中随着距离

的增加,能量开销会逐渐增大。

针对 HEED 等算法中节点根据最小平均功率原则选择加入簇不能保证负载均衡和能量有效的问题,EECS 算法<sup>[12]</sup>提出一种新的通信代价公式来决定节点加入哪个簇。实验结果表明,EECS 的控制报文开销比 HEED 少且网络生命周期较 LEACH 提高 135%。但 EECS 仍未考虑因节点的移动性引起的能量消耗变化问题。

### 1.3 改进 S-HEED 协议

针对以上问题,笔者提出基于节点稳定性的 S-HEED(stable based HEED)分簇算法。节点在选择簇头时,首先计算节点加入簇头代价。若有多个候选簇头,再根据节点稳定性选出最优簇头候选集。最后选择加入平均通信功率最低的簇头。

## 2 基于稳定度的 S-HEED 分簇算法

### 2.1 相关定义

**定义 1** 时间距离( $t_d$ ):若节点 A、B 通信,A 发送信息到 B 收到所间隔的时间,称为 A 到 B 的时间距离。 $t_d$  可以通过周期性的邻居节点 HELLO 消息交互来完成,其计算方法如下<sup>[12]</sup>:

假设 A、B 节点通信,A 在  $t_{1s}$  时刻给 B 发送信息,B 在  $t_{1r}$  时刻收到,B 在  $t_{2s}$  时刻给 A 发送信息,A 在  $t_{2r}$  时刻收到信息,则

$$t_d = \frac{(t_{1r} + t_{2r}) - (t_{1s} + t_{2s})}{2} \quad (2)$$

**定义 2** 稳定性( $S_l$ ):若两节点的时间距离  $t_d$  呈减小趋势,则稳定性增加, $S_l$  为“+”;否则为“-”。

### 2.2 S-HEED 分簇算法

S-HEED 算法考虑到节点的移动性,在整个分簇过程中通过 HELLO 机制实现邻居路由表信息交换。邻居路由表包含信息如表 1 所示。

表 1 邻居路由表结构

nodeID	cost	$S_l$	energy	power
节点 ID	加入簇代价	节点稳定性	剩余能量	平均功率

#### 2.2.1 初始化阶段

网络中节点部署后,各个节点根据公式(3)设置自己成为簇头的概率  $CH_p$ :

$$CH_p = \max(C_p \times E_r / E_m, P_{\min}), \quad (3)$$

其中: $C_p$  为网络中初始簇头比例; $E_r$  为节点当前剩余能量; $E_m$  为节点初始能量。为防止簇头选举时迭代收敛速度过慢,设置  $C_p$  最小值为  $P_{\min}$ 。

#### 2.2.2 簇生成阶段

在簇生成阶段,每个节点设置自己的初始值为  $CH_p$ 。

若  $CH_p < 1$  且节点没有收到其它节点的广播簇头消息,则将  $CH_p$  乘以 2 进入下一个迭代过程;若收到其它节点的广播簇头消息则进入加入簇的算法流程。

若  $CH_p \geq 1$  则迭代终止。节点广播自己的信息给邻居节点,邻居节点根据加入簇代价、稳定性以及簇头平均功率来判断是否加入该簇。

根据不同的  $CH_p$  值,实质上是把网络中节点根据剩余能量分为了 8 个不同等级,等级越高的节点因为剩余能量比其它节点多得多,所以优先成为簇头节点。如表 2 所示。

表 2 节点能量等级

节点等级	$CH_p$
1	[0.5, 1]
2	[0.25, 0.5)
3	[0.125, 0.25)
4	[0.0625, 0.125)
5	[0.03125, 0.0625)
6	[0.015625, 0.03125)
7	[0.0078125, 0.015625)
8	[0, 0.0078125)

当节点收到多个簇头的广播消息时,需要选择加入其中一个簇。根据表 2 可知,同时发来广播消息的簇头肯定是属于同一能量等级的节点,即节点的剩余能量相差不大。因此,节点选择簇头时分别根据加入簇代价 cost、节点与簇头之间的稳定性以及平均通信功率 3 个条件来考虑,且优先级依次递减。其加入簇的具体算法流程如下。

**Step1.** 节点  $CH_p \geq 1$  后,簇头节点通过 HELLO 机制获得自己与 sink 节点的距离  $d$ ,然后广播报文声明自己为簇头。

**Step2.** 若收到报文的邻居节点也是簇头节点且剩余能量高于源节点,则返回步骤 1 发送广播报文;若节点是非簇头节点且收到 1 个广播报文,则加入该簇头,完成分簇过程;若节点是非簇头节点且收到多个广播报文,则进入步骤 3。

**Step3.** 邻居节点收到多个广播报文后,根据公式(3)~(5)计算自己加入每个簇头各自的通信代价<sup>[12]</sup>:

$$\text{cost}(j, i) = w \times f(d(P_j, CH_i)) + (1 - w) \times g(d(CH_i, BS)), \quad (4)$$

$$f = \frac{d(P_j, CH_i)}{\sum_{i=1}^n d(P_j, CH_i)} \quad (5)$$

$$g = \frac{d(CH_i, BS)}{\sum_{i=1}^m d(CH_i, BS)} \quad (6)$$

其中,公式(4)表示节点  $j$  加入簇头  $i$  的代价占所有可达簇头总代价的比值;公式(6)表示簇头  $i$  到达 sink 节点的距离代价与节点  $j$  的所有可达簇头到达 sink 节点的距离代价比值。权值  $w$  根据具体应用设置不同值,其目的是平衡成员节点与簇头的能耗。

通过计算最小通信代价,避免了邻居节点选择距离 sink 节点较远的簇头,从而使得这些簇头有较少的簇内节点,降低了簇内能量开销,弥补簇头与 sink 节点传输数据的高能耗开销。

**Step4.** 邻居节点  $j$  将最小通信代价的簇头  $i$  加入候选簇头集合  $ClusterHead(N_j) = \{N_j | N_j, ClusterHead=N_i\}$ 。同时,邻居节点将其  $cost_{min}$  与其它簇头的  $cost_n$  根据公式(6)进行比较。 $\Delta L$  用于衡量节点加入 2 个簇头的代价  $cost$  差值是否在可接受范围内,其值根据具体网络环境设置:

$$|cost_{min} - cost_n| \leq \Delta L, \quad (7)$$

若差值小于  $\Delta L$ ,则将该簇头加入到候选簇头集  $ClusterHead(N_j)$ 中;若其值大于  $\Delta L$ ,则放弃加入该簇。计算结束后,若  $ClusterHead(N_j)$  只有 1 个候选簇头,则节点加入该簇;若有多个簇头,则进入步骤 5。

**Step5.** 依次遍历候选簇头集  $ClusterHead(N_j)$  中的簇头,若  $N_i \cdot S_j = '+'$ ,则表示该簇头与节点  $j$  距离逐渐缩短,稳定性呈增加趋势,能耗开销逐渐减少,保留此簇头信息在候选集中;若  $N_i \cdot S_j = '-'$ ,表示该簇头与节点  $j$  距离逐渐增加,能耗开销会逐渐增大,将其从候选集用删除。

遍历结束后,若候选簇头稳定性均为‘-’,则返回步骤 4 选择最小通信代价的节点作为簇头;若候选簇头只有 1 个稳定性为‘+’,则选择节点  $j$  加入该簇,完成分簇过程;若候选簇头中有多个稳定性为‘+’,则进入步骤 6。

**Step6.** 遍历候选簇头集  $ClusterHead(N_j)$  中的簇头,选出平均通信功率最小的节点作为最终簇头。

图 2 描述了簇生成算法的实现。

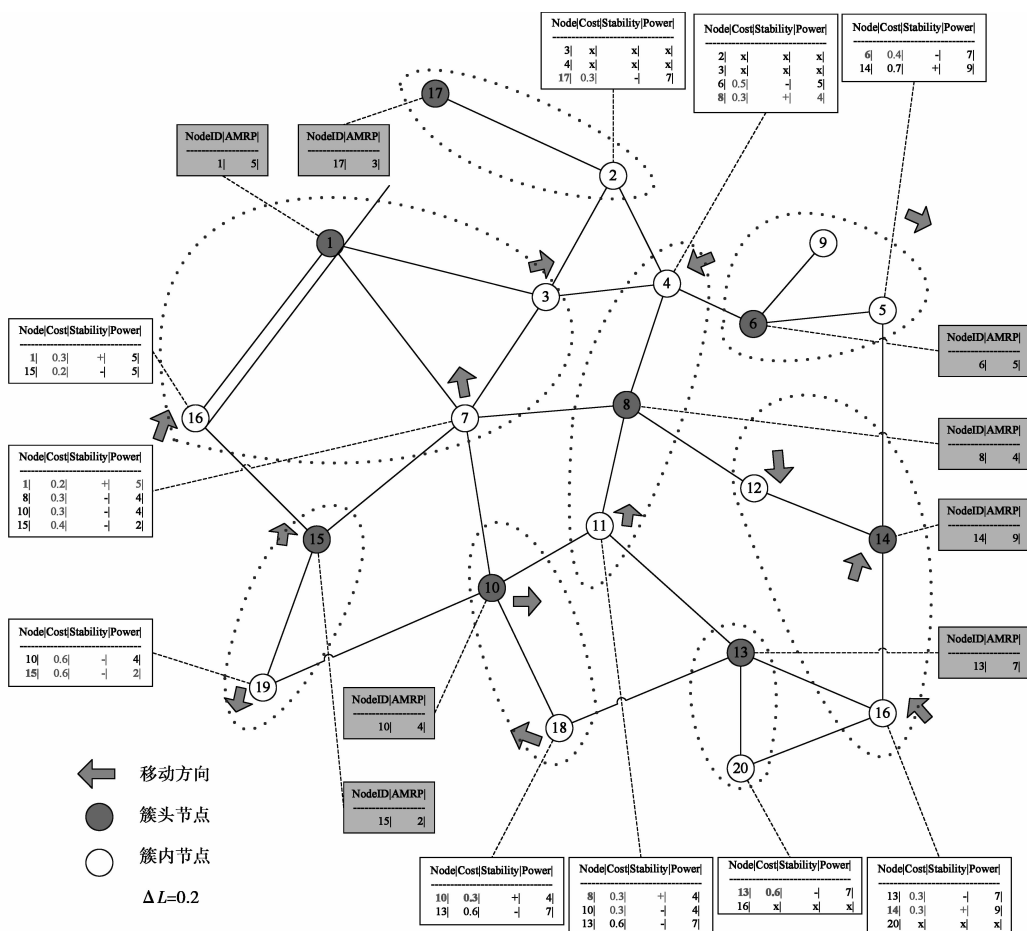


图 2 簇生成算法的实现

### 2.2.3 簇维护与异常处理

由于簇头的产生采用了分布式选举,节点在本地计算自己剩余能量以及成为簇头的概率  $CH_p$ ,所以无需交换控制报文来获取邻居节点信息,减少了额外的能量开销。

若节点在数据传输期间因能量耗尽,则判断是否是簇头节点。

1)若为簇头节点,则该簇头通过广播消息告诉簇内成员节点“解散”簇的消息。成员节点收到该报文后,在自己的邻居表内从簇生成算法步骤 3 开始,重新选择新簇头加入。

2)若为簇内成员节点,则节点发送消息给簇头,簇头将其从自己的簇内成员列表中删除。

## 3 实验仿真

文中采用 NS2 仿真工具对 S-HEED 协议进行了仿真实验,与 LEACH、HEED 2 种分簇算法在网络生命周期、簇头能量消耗和数据包投递率等方面进行比较。实验参数见表 3。

表 3 NS2 仿真实验参数

参数	值
网络范围/m	500 × 500
节点数量	1~200
Sink 节点坐标	(50,50)
初始能量/J	2~4
簇有效半径/m	20
$E_{elec}/(nJ \cdot bit^{-1})$	50
$C_p/\%$	5
发射功率/dBm	1
移动速度/( $m \cdot s^{-1}$ )	0~5
运行时间/s	300
MAC 层	IEEE 802.11

### 3.1 网络生命周期

定义网络生命周期为网络部署到网络中第一个节点死亡的时间,因为在 S-HEED 算法中簇头是轮流产生,所以用簇头选举的轮数(round)来衡量网络生命周期。观察在不同移动速率和节点数量下网络生命周期的变化。如图 2 和图 3 所示。

图 3 的网络中共有 100 个节点,节点速率从 0 到 5 m/s 逐渐增加。从图中可以看出,S-HEED 相对于 LEACH 和 HEED 在生命周期上有所提升。当节点不移动时,S-HEED 和 HEED 协议在生命周

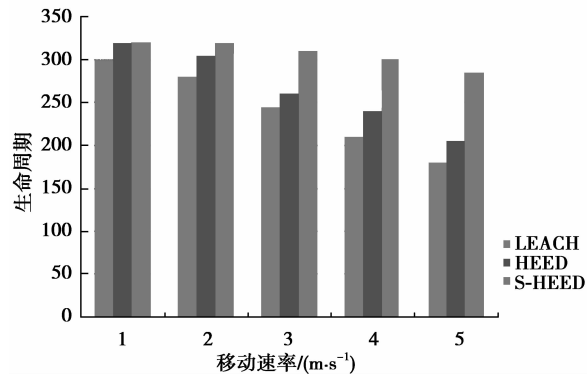


图 3 网络生命周期与节点移动速率

期上几乎差不多。随着节点移动速度的增加,采用 HEED 和 LEACH 算法因为节点选择簇头时未考虑到传输距离,能耗开销较大,其网络生命周期明显降低。而 S-HEED 协议因为考虑了节点的移动性,将节点传输距离和稳定性作为选择簇头的参数,所以生命周期仍然高于 LEACH 和 HEED 算法且缩小幅度不大。

图 4 显示了节点数量与网络生命周期的关系,在仿真实验中,节点保持 5 m/s 的平均速度移动,节点数量从 50 个逐渐增加到 200 个。从图中可以看出,S-HEED 协议在生命周期上仍然高于 LEACH 和 HEED 协议。

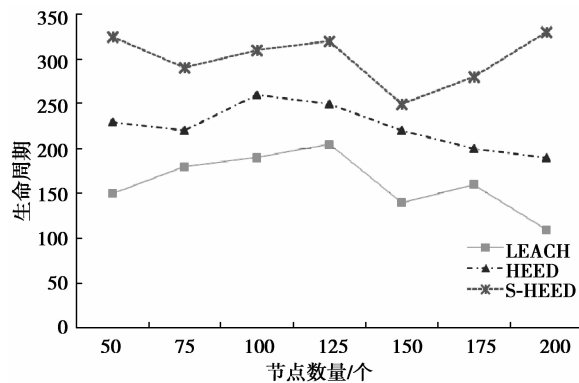


图 4 网络生命周期与节点数量

### 3.2 簇头能量消耗

每轮的数据传输阶段,簇头的能耗开销是整个网络中最大的。因此,通过观察前 100 轮循环中每 10 轮里所有簇头的能量消耗总和来评估整个网络的能耗情况。实验中,节点的数量为 100 个且均以 5 m/s 的平均速度移动如图 5 所示。

在前 100 轮循环中,由于节点保持 5 m/s 的速率移动,基于稳定性和最小通信代价的 S-HEED 协议中的簇头消耗能量总和低于 LEACH 和 HEED 协议。

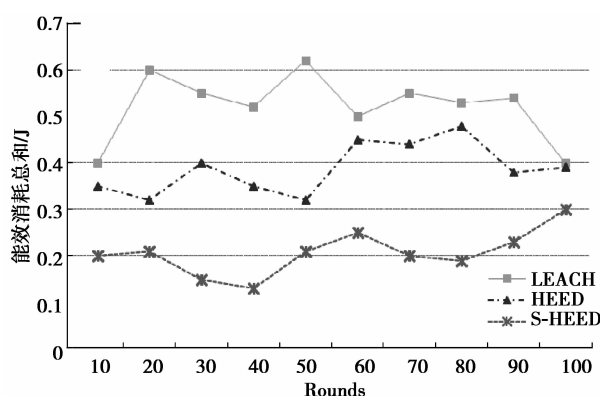


图 5 每轮簇头能效消耗总和

### 3.3 数据包投递率

数据包投递率为目的节点接收到的数据包的个数与理论上应该收到的数据包个数的比值,它反应了簇结构和网络链路的质量。

图 6 显示了不同速率下,3 种算法的数据包投递率情况。在速率低于 2 m/s 时,LEACH、HEED 和 S-HEED 算法的投递率几乎相同。随着节点移动速率的增加,3 种算法的数据包投递率均有所下降,但是 S-HEED 始终优于 LEACH 和 HEED 算法。

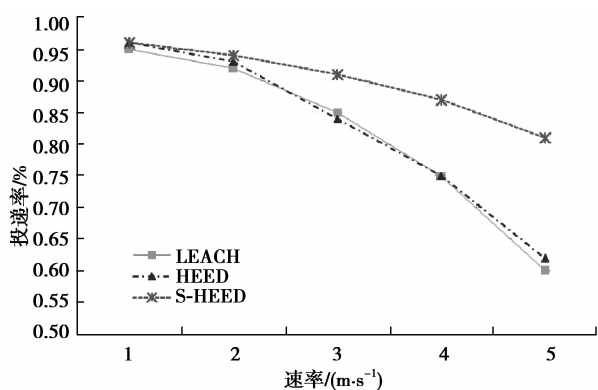


图 6 数据包投递率

## 4 结束语

针对 HEED 分簇算法的不足,提出了基于稳定性的 S-HEED 分簇算法。该算法将节点间的最小通信代价和稳定性作为加入簇的条件,解决了因移动性带来的高能耗问题。仿真实验显示,S-HEED 算法在移动性较强的网络中表现出良好性能,在网络生命周期、能量消耗和数据包投递率方面均有所提高。

### 参考文献:

[1] 孙利民,李建中,陈渝. 无线传感器网络[M]. 北京:清华大学出版社,2005.  
[2] Lifetime analysis of random event-driven clustered

wireless sensor networks[J]. IEEE transactions on mobile computing, 2011, 10(10):10.

- [3] Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy-Efficient communication protocol for wireless microsensor networks[C]// Proc. of the Hawaii Int'l Conf on System Sciences. San Francisco: IEEE, Computer Society, 2000, 3005-3014.
- [4] Handy M J, Haase M, Timmermann D. Low energy adaptive clustering hierarchy with deterministic cluster-head selection[C]// In: Proc of the 4th IEEE Conf on Mobile and Wireless Communications Networks. Stockholm: IEEE Communications Society, 2002. 368-372.
- [5] Heinzelman W. Application-Specific protocol architectures for wireless networks [D]. Boston: Massachusetts Institute of Technology, 2000.
- [6] Younis O, Fahmy S. Heed: A hybrid energy-efficient distributed clustering approach for ad-hoc sensor networks [J]. IEEE Trans on Mobile Computing, 2004, 3(4): 660-669.
- [7] Hoda T, Peyman N. Improving on HEED protocol of wireless sensor networks using non probabilistic approach and fuzzy logic[C]//2010 5th International Symposium on Telecommunications.
- [8] Rafał Pa, Bartosz W, Maciej N. New simplified HEED algorithm for wireless sensor networks [J]. Communications in Computer and Information Science, 2010, 79: 332-341.
- [9] 阎新芳,孙雨耕,赵承利. 无线传感器网络中基于能量的分级簇算法[J]. 天津大学学报, 2005, 38(12): 1106-1110.  
YAN Xinfang, SUN Yugeng, ZHAO Chengli. Energy-aware hierarchical clustering algorithm for wireless sensor networks[J]. Journal of Tianjin University, 2005, 38(12):1106-1110.
- [10] 刘刚,周兴社. EEPC: 一种新的无线智能传感器网络组织结构与协议[J]. 计算机工程, 2005, 31(18):57-59.  
LIU Gang, ZHOU Xingshe. EEPC: A new protocol for wireless sensor network [J]. Computer Engineering, 2005, 31(18):57-59.
- [11] Ye M, Li C F, Chen G H, et al. EECS: An energy efficient clustering scheme in wireless sensor networks [C] // In: Proc. of the IEEE Int'l Performance Computing and Communications Cone. New York: IEEE Press, 2005. 535-540.
- [12] Xu C N, Zhao L, Xu Y J, et al. Broadcast time synchronization algorithm for wireless sensor networks [C]//Proceedings of the 1th International Conference on Sensing, Computing and Automation. Chongqing, 2006: 2366-237. (编辑 陈移峰)