

# 物联网中 LEACH 算法的研究与改进

张冬悦, 徐四委, 高 辉

(辽宁大学信息学院, 沈阳 110036)

**摘 要:**针对物联网节点异构的特点,从时效性要求不同的角度出发,提出一种改进的 LEACH 路由分簇算法。在传送数据之前,要求节点发送代表数据缓急的脉冲,根据脉冲长度及频率得到节点时效性要求参数,结合参数的大小和传输的数据量,动态分配每轮数据传输的时隙,使得高时效性要求节点分配较长时隙,在相对较短的轮次内完成数据传输。仿真结果表明,高时效性节点优先得到资源,传送时延较小,特别在节点数目众多的条件下,节点的平均传送时延得到降低。

**关键词:**物联网;成簇;时效性;时隙;平均时延

**中图分类号:**TP393

**文献标识码:**A

物联网把传统的信息通信网络延伸到了更为广泛的物理世界,是信息化发展的一大趋势<sup>[1]</sup>。物联网的底层可以认为是由不同功能的传感器网络交织在一起,因此物联网既具有无线传感器网络的特点,同时又具有新的异构特性。节点在异构的环境中如何更好的相互协作传递信息,成为新的研究热点,本文重点针对物联网中的路由分簇算法进行改进。

现有的分簇算法大多是基于 LEACH 算法<sup>[2]</sup>,LEACH 分簇算法每轮随机选择簇首,簇首通过均衡时隙分配对成员节点进行数据融合,平衡了节点间的能量消耗,延长整个网络生命期<sup>[3]</sup>。但在实际物联网的应用中,异构的节点执行不同的任务,不同的任务需求对传送数据的缓急程度的要求是不同的,LEACH 算法的均衡平等的时隙分配,不能针对不同节点对时效性的不同要求进行动态时隙分配。因此,本文从物联网中对时效性要求异构的特点出发,提出了适于物联网特点的改进的 LEACH 算法。

## 1 LEACH (Low Energy Adaptive Clustering Hierarchy) 算法

LEACH 算法是一种基于分布式的平等随机的自组织自适应分簇算法,目标是平衡负载,减少能量消耗,延长网络寿命。在一个周期中节点轮流充当簇首,每轮分

为簇首选举和稳定工作两个阶段。

在簇首选举阶段,先为没有担任过簇首的节点  $n$  生成一个  $0 \sim 1$  之间的随机数,如果随机数值小于阈值  $T(n)$ <sup>[4]</sup>,则节点在这一轮成为簇首。阈值  $T$  为:

$$T(n) = \begin{cases} \frac{P}{1 - P \cdot \left( r \bmod \frac{1}{P} \right)} & n \in G \\ 0 & \end{cases} \quad (1)$$

其中,  $P$  是网络中簇首节点占有所有节点的比例,  $r$  是正在经历的轮数为第  $r$  轮,  $G$  是最近  $1/P$  轮中没有成为簇首的节点的集合。建立了第  $r$  轮节点成为簇首的概率与网络中簇头节点的比例之间的关系。节点成为簇首后,向周围节点广播信息,而其它节点根据收到信号的强弱,选择一个簇加入,并回复加入消息。簇首收到所有成员的加入消息后,以时分多址 (Time Division Multiple Access, TDMA) 方式为成员节点均衡分配时隙<sup>[5]</sup>,然后进入稳定工作阶段。

其中,簇首节点需要融合成员节点的数据并向全局站点传输数据,因此成为高耗能节点,LEACH 算法的随机簇首选择,平衡了节点间的能量消耗,防止单一节点过早死亡。同时,在这样的分层簇结构中,每个成员节点也只需要知道自己所属簇的簇头信息,簇头节点也只需要维持很小的路由表,减少了大量数据的长距离传输<sup>[6]</sup>。但是,在物联网环境中,它也有一些不足:

收稿日期:2012-02-05

基金项目:国家自然科学基金资助项目(60703068);辽宁省高等学校科研资助项目(2008257)

作者简介:张冬悦(1986-),女,山西运城人,硕士生,主要从事物联网方面的研究,(E-mail)xx\_zdy@163.com

(1) 式(1)中阈值  $T(n)$  的确定依赖于随机生成数  $n$ , 及簇首节点所占比例  $P$ 。 $n$  值和  $P$  值的确定均假设所有节点地位平等, 因而,  $T(n)$  值是在这种平等前提下进行的简单比例运算。在物联网环境中, 节点异构, 地位不平等, 不适于等比值计算。

(2) 在稳定工作阶段, 采用 TDMA 均衡分配, 忽略了节点间的差异性, 有悖于物联网的节点物理结构差异和节点的功能性不同的特点, 不能体现节点时效性要求方面的差异。

由此可见, 物联网最大的特点之一就是节点多元化, 本文着重分析物联网节点对时效性要求不同的特点, 采用动态分配时隙的方法对经典 LEACH 算法进行改进。

## 2 基于时效性要求的 LEACH 算法的改进

针对物联网节点对时效性要求不同的新特点, 对 LEACH 算法的改进策略如下: 在传送数据之前, 节点发送表示时效性缓急程度的脉冲, 结合脉冲长度和节点固有的频率, 得到时效性缓急参数, 通过参数的大小体现时效性要求。参数大的节点分配相对较长的时间间隙, 使其在较少的轮次内完成传输, 达到短时间完成数据传输的目的。复杂度的增加对簇首能力提出了更高的要求, 本文采取对节点进行智能分级, 使高智能程度的节点以高概率成为簇首, 从而整体上提升簇首的处理数据速度, 适应了时效性要求。改进的算法在一个周期内同样分为簇首选举阶段和稳定传输阶段。

### 2.1 簇首选举阶段

根据物联网中节点异构的特点, 对节点进行智能程度分级, 把智能程度高的节点看作多个低智能程度节点的聚合体<sup>[7]</sup>, 从而增加了高智能程度节点成为簇首的概率。设定节点智能程度在  $[W, \alpha_{max}W]$  区间内随机分布, 其中  $\alpha_{max}$  为最高智能程度节点与最低智能程度节点的倍数比, 对任意节点  $s_i$ , 它的智能程度为  $\alpha_i W$ , 因而总智能程度为:

$$W_{total} = \sum_{j=1}^N \alpha_j W \quad (2)$$

所以, 对簇首节点占所有节点的比例  $P$  进行优化为:

$$P_i = \frac{P \cdot N \alpha_i}{\sum_{j=1}^N \alpha_j} \quad (3)$$

其中  $N$  代表节点个数,  $\alpha_i$  代表节点的智能级别。将公式(3)中的  $P_i$  代入公式(1)中的  $P$ , 即可得各节点在第  $r$  轮成为簇首的概率。可以看出高智能程度节点成为簇头的轮次要小于普通节点, 使得高智能程度节点作为簇头的机会大于普通节点, 从而高效利用了资源, 提高了簇首

处理数据的能力。

### 2.2 稳定传输阶段

在物联网的异构节点环境中, 已知  $n$  个节点固有发射频率为  $\{f_1, f_2, f_3, \dots, f_n\}$ , 在节点发送数据之前, 先发送一组消息, 表示其数据传输的缓急程度, 发送脉冲消息长度为  $\{m_1, m_2, m_3, \dots, m_n\}$ , 则节点时效性要求强弱  $r = m/f$ , 即节点的发射频率越小, 发送缓急程度的脉冲越长, 节点的时效性要求越高。

已知传输信道的容量  $C$ , 在时间  $t$  内, 发送的数据量为  $Ct$ 。簇内  $n$  个成员节点, 记为  $\{s_1, s_2, \dots, s_n\}$ , 初始状态下节点时隙分别为  $\{t_1, t_2, \dots, t_n\}$  且  $t_1 = t_2 = \dots = t_n = \alpha$ , 其中  $\alpha$  为最短时隙长度。对于任意节点  $s_i$ , 时隙  $t_i$ , 在  $t_i$  时隙内发送数据量为  $Ct_i$ , 其发射频率为  $f_i$ , 脉冲长度  $m_i$ , 则节点时效性要求参数  $r_i = m_i/f_i$ , 需要传输的数为  $l_i$ 。

经过一轮数据传输后, 剩余数据量为  $l - Ct_i$ , 如果  $l - Ct_i > 0$ , 说明  $s_i$  的数据在时隙  $t_i$  内没有发送完毕。若希望数据在下一轮发送完毕, 则需要分配时隙为  $(l - Ct_i)/C$ 。从时效性方面考虑, 仅考虑时效性强弱, 则希望下一轮分配的时隙为  $Kr$ , 其中  $K$  为固定常数 ( $K \neq 0$ )。为了保证成员节点间的公平性, 设定分配时隙的上限为  $\beta$ , 防止拥有长数据量的高时效性节点长时间占用资源。

由上述分析可知影响下一轮时隙长度分配的因素见表 1。

表 1 影响下一轮时隙长度分配的因素

符号	含义
$\alpha$	最短时隙长度
$\beta$	最长时隙长度
$(l - Ct_i)/C$	从剩余数据量角度考虑的希望时隙值
$Kr$	从时效性要求角度考虑的希望时隙值

(1)  $(l - Ct_i)/C < Kr$  表示在满足时效性要求的前提下, 可以完成数据发送。

①  $(l - Ct_i)/C \leq \alpha$ , 选择时隙长度为  $\alpha$ , 即发送完全部数据所需的时隙长度小于最短时隙, 取下届值。

②  $\alpha < (l - Ct_i)/C < \beta$ , 选择时隙长度为  $(l - Ct_i)/C$ , 即在时隙内数据发送完毕。

③  $(l - Ct_i)/C \geq \beta$ , 选择时隙长度为  $\beta$ , 即发送完全部数据所需的时隙长度大于最长时隙, 取上届值。

(2)  $(l - Ct_i)/C > Kr$  表示若按照时效性要求分配时隙, 则数据不能全部发送, 若按照剩余数据量分配时隙, 则会打破时效性要求。在数据量和时效性两者冲突时, 优先考虑时效性要求。

①  $Kr \leq \alpha$ , 选择时隙长度为  $\alpha$ , 即按照时效性要求确定的时隙长度小于最短时隙, 取下届值。

②  $\alpha < Kr < \beta$ , 选择时隙长度为  $Ks$ , 即优先满足时效性发送要求。

③  $Kr \geq \beta$ , 选择时隙长度为  $\beta$ , 即按照时效性要求确定的时隙长度大于最长时隙, 取上届值。

通过上述算法动态自主进行时隙分配, 优先考虑成员节点对时效性的要求, 同时兼顾了剩余数据量, 平衡了节点间的公平性, 在一定程度上适应了对时效性要求不同的物联网环境。

### 3 仿真实验

模仿物联网节点异构的环境, 对 LEACH 和改进的算法进行仿真实验, 监测区域大小为  $200 \times 200$  m。

条件一: 在监测区域安放 200 个节点, 随机生成节点分布。为简化实验, 设定这 200 个节点分为五类, 分别为:

A: 初始能量 3 J, 节点频率 10 Hz, 表示缓急程度的脉冲长度为 30。

B: 初始能量 3 J, 节点频率 10 Hz, 表示缓急程度的脉冲长度为 60。

C: 初始能量 4 J, 节点频率 20 Hz, 表示缓急程度的脉冲长度为 20。

D: 初始能量 4 J, 节点频率 30 Hz, 表示缓急程度的脉冲长度为 60。

E: 初始能量 4 J, 节点频率 10 Hz, 表示缓急程度的脉冲长度为 80。

数据包长度在 500 字节到 1000 字节随机分布。

在上述条件下, 检测下一轮分配的时隙因子  $K$  与数据包的平均时延的关系。

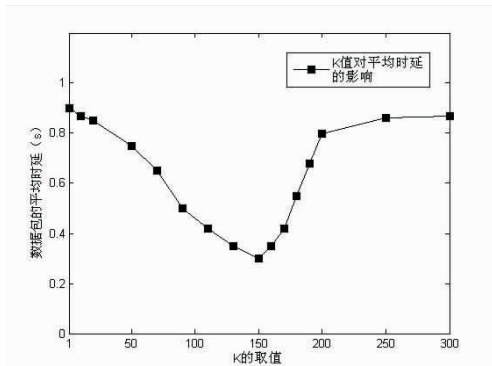


图 1 平均时延随 K 值的变化

由图 1 可以看出, 在实验条件下, 平均时延随 K 值的增加成 V 字形趋势, 当 K 值足够小时,  $Kr < (l - Ct_i)/C$ , 即单纯考虑时效性要求, 对数据长度不断分成小片进行传递, 时效性要求不同的差异性不能体现。当 K 值足够大时,  $Kr > (l - Ct_i)/C$ , 即单纯考虑数据包长

度, 对数据包不进行分割, 没有考虑异构节点对时效性要求的差异性, 因此当 K 值较小或较大的时候, 时延大, 在实验条件下, K 取值在 150 附近时, 时延最小。

条件二: 在监测区安放 N 个节点, 随机生成节点分布。节点同样分为上述 A、B、C、D、E 五类。取  $K = 100$ , 数据包长度在 500 字节到 1000 字节随机分布。

在上述条件下, 检测节点数量与平均时延的关系。

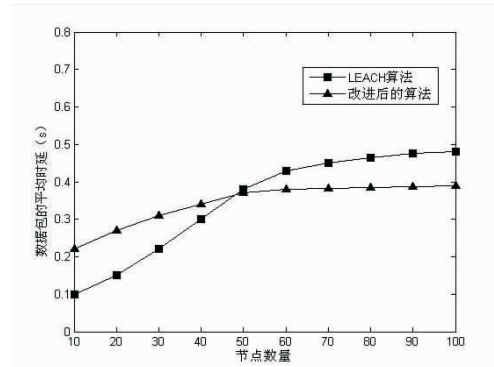


图 2 平均时延随节点数量的变化

由图 2 可以看出, 改进的算法随着节点数目的增多, 降低节点平均时延的效果越好。在节点较少时, 改进后的算法, 增加了簇首节点计算选择的复杂度, 增加了冗余, 时延高于 LEACH 算法。随着节点数目的增多, 传统的 LEACH 算法采用对每个节点分配相等时隙方式, 将导致所有节点轮转一次周期增长, 而节点数据的发送, 往往需要多个周期才能完成, 因此节点等待时间增长, 平均时延加大。改进的算法, 随着节点数目的增多, 不均衡的动态时隙分配优势更突出, 从仿真结果也可以看出, 改进后的算法当节点数目增长到一定程度时平均时延低于 LEACH 算法。

条件三: 在条件二的基础上, 取节点个数为 200, 分别对 LEACH 算法, 改进后的 A、B、C、D、E 五类节点分别进行时延统计。

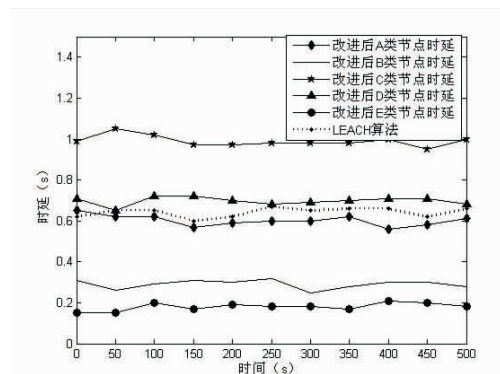


图 3 各类节点时延随时间的变化

由图 3 可以看出, 改进后的算法, 对五类不同时效

性要求的异构节点,能够从时延方面得到很好的区分,高时效性节点在相对较短的时延完成传输,达到了预期效果。对于 LEACH 算法,由于采用均衡时隙分配的方法,不能够对不同时效性要求的节点进行区分,因此在一定程度上不满足物联网的环境。

#### 4 结束语

针对物联网节点对时效性要求不同的特性,对传统的 LEACH 算法进行改进,采用了不均衡的动态时隙分配方法。改进后的算法,能够优先满足高时效性节点的数据传输要求,很大程度上适应了物联网环境的要求。在时效性的创新中,增加了簇首节点的计算复杂度,通过这种方式换取节点对时效性要求区分。经仿真结果分析,算法能够很好的表现异构节点的差异性。在簇首选举时如何评判其智能程度,本文没有进行深入探讨,还需进一步研究、解决。

#### 参考文献:

- [1] Luigi Atzori, Antonio Lera, Giacomo Morabito. The Internet of Things: A survey[J]. Computer Networks, 2010(10):2787-2805.
- [2] 黄河清,沈杰,马奎,等.无线传感器网基于梯度的非均匀分簇[J].光学精密工程,2009,17(18):2053-2059.
- [3] 陈闻杰,陈迅.无线传感器网络成簇算法研究[J].小型微型计算机系统,2008(2):219-225.
- [4] Manjeshwar A, Agrawal D P. Apte: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks[C]. Proceedings of International Parallel and Distributed Processing Symposium, 2002, 195-202.
- [5] 吴臻,金心宇.无线传感器网络的 LEACH 算法的改进[J].传感技术学报,2006,19(1):34-36.
- [6] Smaragdakis G, Matta I, Bestavros A. SEP: a stable election protocol for clustered heterogeneous wireless sensor networks[C]. Proc. of the Int'l Workshop on SANPA, 2004.
- [7] Ranjjan V, Obraczka K, Garcia-Hna-aceves J. Energy efficient collision free medium access control for wireless sensor networks[A]. The First ACM Conference on Embedded Networked Sensor Systems [C]. Los Angeles: ACM, 2003, 181-192.

## Research and Improvement on LEACH Algorithm for Internet of Things

ZHANG Dong-yue, XU Si-wei, GAO Hui

(College of Information, Liaoning University, Shenyang 110036, China)

**Abstract:** According to the characteristics of heterogeneous nodes over Internet of Things, herein LEACH algorithm is proposed based on the different aspects of timeliness. First, the nodes send pulse that the more urgent it is the more pulse it sends. Then according to the length of pulse and frequency of the notes, the head node gets the factor of timeliness. At last, combined with the parameters and amount of data, the head node allocates slot of each round dynamically so that the most Emergency nodes are assigned a longer time slot to complete the data transmission in less times of round. The simulation results show that the Emergency nodes get resources priority, especially in the conditions of the large number of nodes, the average transmission delay of the node is reduced.

**Key words:** Internet of Things(IOT); clustering; timeliness; slot; average delay