

# 无线传感器网络邻近节点数据融合调度算法

许新琳, 郑 瑾, 王国军

(中南大学信息科学与工程学院, 长沙 410083)

**摘 要:** 现有的调度算法存在节点能耗大、数据收集时延长等问题。为此, 提出一种能量高效的邻近节点数据融合调度(CNDAS)算法。该算法通过计算竞争集, 产生无冲突的调度序列, 并为兄弟节点分配连续的传输时隙, 从而降低节点状态转换的频率, 节省节点能耗。模拟结果表明, 当网络规模较大时, CNDAS 算法的能量有效性明显优于 SCHDL 算法。

**关键词:** 无线传感器网络; 数据融合; 邻近节点数据融合调度; 状态转换; 时隙; 竞争集

## Contiguous Node Data Aggregation Scheduling Algorithm in Wireless Sensor Network

XU Xin-lin, ZHENG Jin, WANG Guo-jun

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

**【Abstract】** Existing scheduling algorithms have the problems of large energy consumption and high latency for data collection. So this paper proposes an energy-efficient algorithm called Contiguous Node Data Aggregation Scheduling(CNDAS), which generates a collision-free schedule by computing nodes' competitor sets. It assigns sibling nodes consecutive time slots to reduce the frequency of state transitions and to reduce the energy consumption of sensor nodes. Simulation results show that CNDAS outperforms SCHDL algorithm when the network scale is large.

**【Key words】** Wireless Sensor Network(WSN); data aggregation; Contiguous Node Data Aggregation Scheduling(CNDAS); state transition; time slot; competitor set

DOI: 10.3969/j.issn.1000-3428.2012.03.033

### 1 概述

无线传感器网络(Wireless Sensor Network, WSN)是由分布在监测区域内大量传感器节点组成的一类特殊的 Ad Hoc 网络, 它广泛用于完成环境监测、空间探索、战场监控等传统计算机网络难以胜任的特殊任务<sup>[1]</sup>。在这些无线传感器网络的应用中, 用户无需得到传感器节点监测到的全部原始数据, 而只对事件的监测结果感兴趣, 所以, 在数据收集的过程中需要采用数据融合技术减少通信数据量, 降低能耗<sup>[2]</sup>。

文献[3]证明了最小数据融合时间问题是一个 NP 难问题。文献[4]给出的融合调度算法虽然降低数据收集的延迟, 但很多情况下不能产生一个无冲突的调度, 而且以上 2 个调度算法都是集中式的。由于网络的拓扑结构经常改变, 如节点失效、新节点加入, sink 节点需要重新收集拓扑信息, 制定一个新的调度序列分发执行, 这个过程要消耗大量的能量。文献[5]给出了一个分布式的数据融合调度算法, 但是该算法数据收集的延迟较大, 并且能量效率不高。文献[6]提出的调度算法考虑在数据收集过程中所有节点的延迟之和, 使得有些节点会多次出现在调度序列中, 传输的数据量较大。上述算法都没有考虑节点在不同状态之间转换所消耗的能量, 因为频繁的状态转换会导致很大的额外的能量消耗。本文利用数据融合技术, 提出一种分布式的邻近节点数据融合调度(Contiguous Node Data Aggregation Scheduling, CNDAS)算法, 用以解决数据收集与融合过程中的冲突问题, 降低节点的能耗。

### 2 能量模型

网络节点集合为  $V$ , 由  $n$  个同构的传感器节点组成。节

点监测到数据之后, 经过多跳转发的方式将数据发送到 sink 节点, 中间转发节点需对多个子节点发来的数据进行融合。

本文采用的能量模型类似于文献[7], 假设传感器节点有 3 个状态: 活动状态, 睡眠状态和转换状态, 节点只有在活动状态才能进行数据的发送和接收。当采用 Mica2 尘埃节点和 CC1000 收发器时, 节点各种操作所需的时间和能耗如表 1 所示<sup>[8]</sup>。节点从睡眠状态到活动状态的开启过程包含: 初始化, 开启收发器, 发送/接收转换, 激活一个传感器节点所需的时间是 2.1 ms, 能量消耗为 22  $\mu$ J。

表 1 节点不同操作所需时间和能耗

操作	时间/ms	能耗/ $\mu$ J
睡眠	—	0.03
初始化	0.350	6.00
开启收发器	1.500	1.00
发送/接收转换	0.250	15.00
接收 1 Byte 数据	0.416	15.00
发送 1 Byte 数据	0.416	20.00

图 1 为本文采用的能量模型示意图。图 1(a)给出了传感器节点在开启和关闭过程中的时间和能量消耗; 图 1(b)表明, 如果能够合并这些活动时隙, 就能降低节点的开启频率, 从而节省状态转换所消耗的能量和时间。

**基金项目:** 国家自然科学基金资助项目(61073037); 湖南省杰出青年科学基金资助项目(07JJ1010)

**作者简介:** 许新琳(1986—), 男, 硕士研究生, 主研方向: 无线传感器网络; 郑 瑾, 副教授、博士; 王国军, 教授、博士生导师

**收稿日期:** 2011-07-24 **E-mail:** xxl464@gmail.com



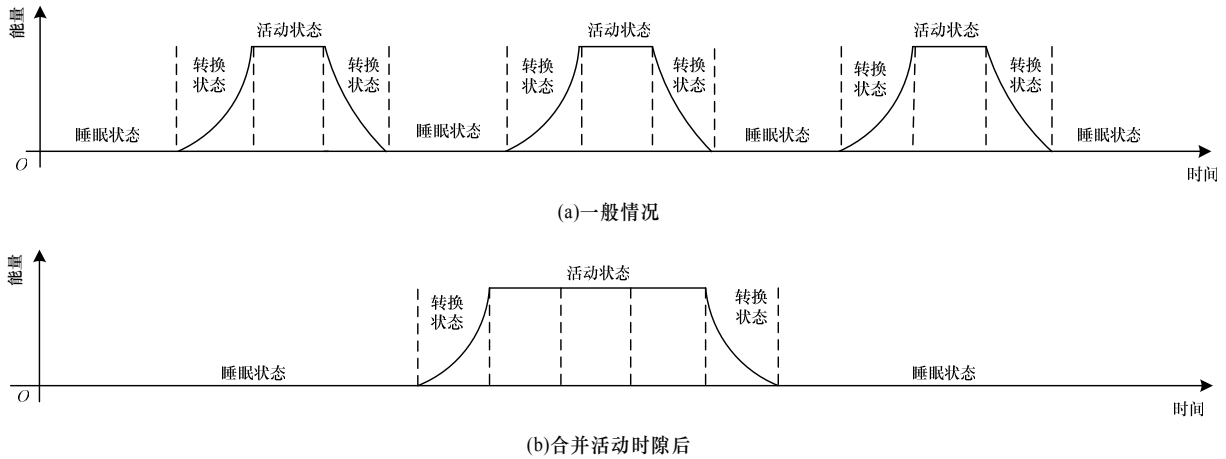


图1 能量模型示意图

### 3 邻近数据融合调度问题描述

假设集合  $A, B \subset V$  且  $A \cap B = \emptyset$ ，在某个时隙中，集合  $A$  的节点同时发送数据，而集合  $B$  的节点正确收到了所有数据，则称数据从集合  $A$  融合到集合  $B$ ，集合  $A$  为发送集。数据融合调度就是要找这样一个发送集序列  $S_1, S_2, \dots, S_l$ ，其满足以下条件：(1)  $S_i \cap S_j = \emptyset, \forall i \neq j$ ；(2)  $\cup_{i=1}^m S_i = V - \{n_s\}$ ；(3)在时隙  $k (k=1, 2, \dots, m)$  中，数据从  $S_k$  融合到  $V - \cup_{i=1}^m S_i$ ，经过  $m$  个时隙后，数据就融合到 sink 节点  $n_s$ 。

图 2 给出了非邻近节点调度与邻近节点调度的比较结果，其中，图 2(a)是以节点 1 为根的数据融合树，为了简化，假设任意 2 个节点同时发送数据都会产生冲突；图 2(b)为非邻近节点调度结果，可以看到，每个节点需开启多次才能接收所有子节点发送来的数据；图 2(c)是数据融合邻近节点调度结果，这时，每个节点仅开启一次便能接收完所有数据。

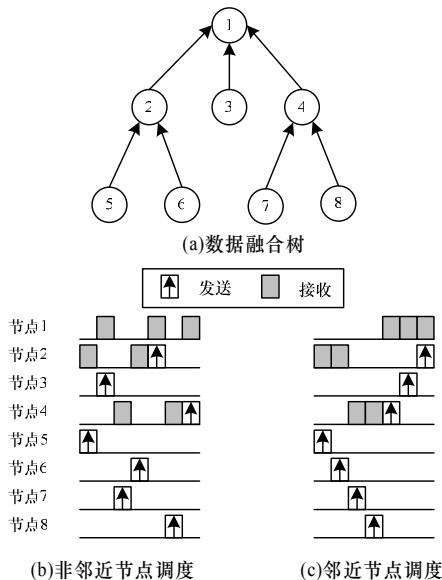


图2 非邻近节点调度与邻近节点调度比较

### 4 邻近节点调度算法

本文针对邻近节点数据融合调度问题提出了一个分布式的调度算法 CNDAS。假设已在网络中构造好一棵以 sink 节点为根的融合树。

#### 4.1 冲突问题描述

如果 2 个节点需要在某一时隙中同时发送数据，而接收节点在它们的重叠传输范围内，那么这个节点将不能正确接

收到它们发送的数据，即发生了冲突。

**定义 1** 对于任意节点  $u$ ，如果某一节点与  $u$  同时发送数据会产生冲突，则称该节点为  $u$  的竞争者。

**定义 2**  $u$  的所有竞争者的集合构成了  $u$  的竞争集  $C(u)$ 。

假设在某一时隙节点  $u$  向其父节点  $p$  发送数据，同时节点  $v$  向其父节点  $w$  发送数据，那么这时冲突可能发生在节点  $p$  或节点  $w$ ，这 2 种情况如图 3 所示。在图 3(a)中，节点  $p$  在节点  $v$  的传输范围中，它会接收到  $u$  和  $v$  等节点发来的多份数据，所以， $p$  的一跳邻居都是  $u$  的竞争者。这里要将  $p$  的父节点  $t$  从  $p$  的一跳邻居中删除，因为  $t$  在没有收到所有后代节点的数据之前不会发送数据，即  $t$  不会和  $u$  同时发送数据。在图 3(b)中，节点  $w$  在节点  $u$  的传输范围中，它也会接收到  $u, v$  或其他子节点发来的多份数据，所以， $u$  的一跳邻居的子节点都是  $u$  的竞争者。这里要将  $u$  的父节点  $p$  和  $u$  的所有子节点从  $u$  的一跳邻居中删除，因为节点  $p$  的子节点已经包含在图 3(a)中，而  $u$  的孙子节点不会和  $u$  同时传输。

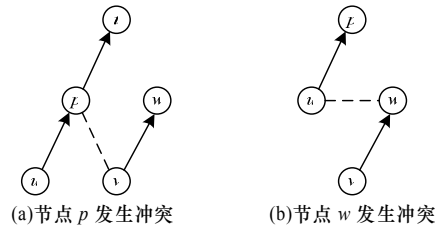


图3 传输干扰示例

#### 4.2 调度算法

数据融合调度是把时间分成一个个时隙，然后分配给传感器节点。节点在它所属的时隙中才能发送数据，当节点不发送也不接收数据时，应该关闭其收发器进入睡眠状态。为了得到一个无冲突的调度，最简单的方法是分配给每个节点一个传输时隙，这样 sink 节点需要  $n$  个时隙才能接收完所有数据。很明显，这增加了延迟。但多个无冲突的数据传输可以同时进行，由此可以减少所需的时隙数。

当分配给节点  $u$  一个传输时隙后， $u$  的父节点  $p$  开启接收从  $u$  发送来的数据，在这个时隙之后，节点  $u$  和  $p$  都应该关闭收发器然后进入睡眠状态。如前文所述，频繁的节点开启会导致额外的时间和能量消耗，所以，本文提出的邻近节点数据融合调度算法为兄弟节点分配连续的传输时隙，从而降低状态转换的频率，这样，节点可能仅需开启一次便能接收完所有子节点发来的数据，降低了能耗和延迟。邻近节点数据融合调度算法具体步骤如下：

Input: aggregation tree  $T$  in wireless sensor network

Output: a data aggregation schedule  $S_1, S_2, \dots, S_m$

Procedure: CNDAS( $G, T$ )

```

1. Initially time slot  $TS = 0$ 
2. Set all the leaf nodes as ready to be scheduled
3. while the sink node  $ns$  is not ready do
4.    $TS = TS + 1$ 
5.   for each ready node  $u$  do
6.     if it is the brother of nodes scheduled in time slot  $TS - 1$ 
without collision then
7.       assign time slot  $TS$  to the node
8.     end if
9.   end for
10.  if no node assigned time slot  $TS$  then
11.  assign ready nodes whose ID is larger than all nodes in  $C(u)$ 
12.  end if
13.  for each scheduled node  $u$  do
14.    transmit data to its parent, inform all nodes in  $C(u)$  and turn
to sleep state
15.  end for
16.  if all children of a node complete the schedule then
17.    change the node ready to be scheduled
18.  end if
19. end while

```

开始时所有叶子节点进入可调度状态, 如果某个叶子节点的 ID 大于其竞争集  $C(u)$  中所有节点的 ID, 则在当前时隙调度该节点。如果某个节点的所有子节点完成调度, 则该节点进入可调度状态。在之后的时隙里, 选择前一时隙被调度节点的兄弟节点进行调度, 直到所有节点完成调度。由此减少了父节点接收数据的状态转换次数, 从而降低了能耗和时延。

## 5 模拟与结果分析

本文使用 C++ 对 CNDAS 算法与文献[5]的 SCHDL 算法进行模拟实验, 把节点的平均状态转换次数和调度所需的总时隙数作为比较的性能指标。在模拟中, 传感器节点随机均匀地分布在一个  $200\text{ m} \times 200\text{ m}$  的方形区域内, 所有传感器节点的通信半径均为  $25\text{ m}$ 。

图 4 给出在数据融合调度过程中节点的平均状态转换次数, 随着网络规模的增大, 节点的平均状态转换次数增加。CNDAS 算法的状态转换次数为  $2 \sim 4$ , 而在 SCHDL 算法中, 由于很多节点需要开启多次, 这与其子节点的数目有关, 因此它的状态转换次数远多于 CNDAS 算法。由于状态转换会导致额外的能量消耗, 因此在能量有效性方面, CNDAS 算法比 SCHDL 算法更有效。

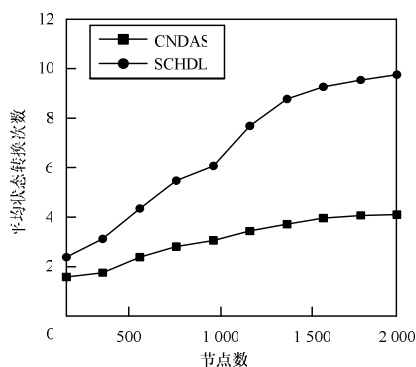


图 4 节点平均状态转换次数比较

图 5 给出数据融合调度所需的总时隙数, 随着网络规模

的增大, 调度所需的总时隙数增加, 这是因为当传感器节点的数目增多时, 发生冲突的节点数目也会增多。但 CNDAS 算法所需的总时隙数只略高于 SCHDL 算法, 特别当网络规模较大时, 因为网络规模越大, 下一时隙可供选择的兄弟节点就越多, 所以两者在时隙上的差距会逐渐变小。而 SCHDL 算法中节点需要的状态转换次数更多, 使得总的时间延迟更大, 同时状态转换也消耗了更多的能量。综上, CNDAS 算法在不明显增加调度时隙的前提下降低了节点能量。

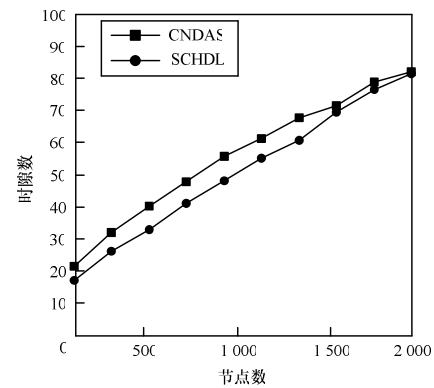


图 5 调度所需时隙数比较

## 6 结束语

数据融合调度能提高 WSN 的能量有效性、降低时延。为此, 本文提出一种无冲突的数据融合调度算法, 通过分配给兄弟节点连续的传输时隙, 降低节点状态转换的频率。节点可能只需开启一次便能接收完一个数据收集周期内所有子节点发送的数据, 从而降低了网络的能量消耗。

## 参考文献

- [1] Akyildiz I, Weilian S, Sankarasubramaniam Y, et al. A Survey on Sensor Networks[J]. IEEE Communications Magazine, 2002, 40(8): 102-114.
- [2] 周 琴, 戴佳筑, 蒋 红. 无线传感器网络数据融合路由算法的改进[J]. 计算机工程, 2010, 36(19): 148-150.
- [3] Chen Xujin, Hu Xiaodong, Zhu Jianming. Minimum Data Aggregation Time Problem in Wireless Sensor Networks[C]//Proc. of the 1st Int'l Conference on Mobile Ad Hoc and Sensor Network. Wuhan, China: [s. n.], 2005.
- [4] Huang Scott, Wan Pengjun, Vu Chinh, et al. Nearly Constant Approximation for Data Aggregation Scheduling in Wireless Sensor Networks[C]//Proc. of INFOCOM'07. Hong Kong, China: IEEE Press, 2007.
- [5] Yu Bo, Li Jianzhong, Li Yingshu. Distributed Data Aggregation Scheduling in Wireless Sensor Networks[C]//Proc. of INFOCOM'09. Harbin, China: IEEE Press, 2009.
- [6] Joo C, Choi J, Shroff N. Delay Performance of Scheduling with Data Aggregation in Wireless Sensor Networks[C]//Proc. of INFOCOM'10. Columbus, USA: IEEE Press, 2010.
- [7] Cui Shuguang, Goldsmith A J, Bahai A. Energy-constrained Modulation Optimization[J]. IEEE Transactions on Wireless Communications, 2005, 4(5): 2349-2360.
- [8] Polastre J, Hill J, Culler D. Versatile Low Power Media Access for Wireless Sensor Networks[C]//Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems. New York, USA: ACM Press, 2004.