

农业传感网中的分布式节能调度算法

戚啸¹, 赵春江², 王跃宣¹, 王永才¹, 陈锦标¹

(1. 清华大学 理论计算机科学研究中心, 北京 100084; 2. 国家农业信息化工程技术研究中心, 北京 100097)

摘 要: 能量管理是无线传感器网络最重要的问题之一, 尤其对于大规模的, 需要长时期、稳定可靠运行的农业无线传感器网络, 如何设计节能调度算法, 使得网络既能稳定正常地采集数据, 又能长年节能地运行, 就变得至关重要。针对此, 提出了一种分布式的能量有效的阶梯式调度算法, 该算法基于相邻层的反向时间同步与阶梯式协调工作与休眠, 保证了网络的节能运行与可靠多跳通信。在 TinyOS 上实现了这种算法, 运行在 60 个 IRIS 节点的实际系统中, 并仿真测试了算法性能。这种调度算法需要网络中节点之间严格的时间同步, 因此也提供了一种实现节点之间时间同步的解决方案。

关键词: 时间同步; 分布式节能调度; 能量管理

中图分类号: TP212.9

文献标识码: A

文章编号: 1000-436X(2010)9A-0104-08

Distributed energy efficient scheduling in wireless sensor network for agriculture

QI Xiao¹, ZHAO Chun-jiang², WANG Yue-xuan¹, WANG Yong-cai¹, CHEN Jin-biao¹

(1. Institute for Theoretical Computer Science, Tsinghua University, Beijing 100084, China;

2. Research Center for Information Technology for Agriculture, Beijing 100097, China)

Abstract: Energy management is one of the most important problems in wireless sensor network(WSN), especially in WSN of agriculture which need to work stably in large-scale farmland for long term. It is therefore very important to design an energy efficient scheduling algorithm which can make sure that the sensor network could sample data and work for long term. A distributed energy efficient stair scheduling algorithm based on reverse time synchronization and stair working and sleeping was proposed to solve this problem. This algorithm guaranteed energy-efficient motion and reliable multi-hop communication in sensor network. The algorithm was implemented in TinyOS and its performance was demonstrated in a real system based on 60 IRIS nodes. This algorithm requires strict time synchronization, so a solution for time synchronization during nodes was also proposed.

Key words: time synchronization; distributed energy efficient scheduling; energy management

1 引言

大规模的农业无线传感器网络是由大量形体

较小、能量受限、具备计算和通信能力的传感器的节点组成的无结构网络。这些节点被部署在广域农田内, 每个节点通过自身配备的土壤温湿度、CO₂、

收稿日期: 2010-07-15

基金项目: 国家高技术研究发展计划(“863”计划)基金资助项目(2006AA10Z216); 国家自然科学基金资助项目(60604033); 国家重点基础研究发展计划(“973”计划)基金资助项目(2007CB807900, 2007CB807901)

Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2006AA10Z216); The National Natural Science Foundation of China (60604033); The National Basic Research Program of China (973 Program) (2007CB807900, 2007CB807901)

光照等农业传感器，检测邻域的土壤和农作物的信息，并通过无线通信形成自组织多跳网络，以多跳通信的方式将采集的数据汇聚到远方的农业信息监测基站。

在前期研究工作中，设计并开发了如图1所示的农业传感器网络信息监测系统^[1]。系统由部署在农田中的大量无线传感器网络和布置在机房中的信息服务平台组成。其中，为覆盖广域农田，无线传感器网络部分采用了传感器节点和3G网关的2层网络体系结构。大量下层的低成本无线传感器自组织形成簇，由3G网关担当簇头，整个农业信息监测网络可以包含多个簇。

在每簇的下层，大量的低成本、低功耗无线传感器节点检测农田中的农作物和土壤信息，并通过无线通信自组织形成多跳的网络将监测信息向簇头汇聚。簇头通过3G网络将数据向远程的信息服务平台汇聚。在整个系统的研究过程中，农业传感器网必须在户外能够长时间稳定地运行，例如，玉米、水稻的生长信息监测均需要半年的时间。而由于传感器节点都是依靠电池供电，能量非常有限，如何最大限度地节省下层节点的能量使得传感器网络可以长期稳定地运行是一个关键性的问题。

除去硬件因素，从协议和算法上解决传感器网络的节能问题最有效的方法是调度节点的工作状态，使节点尽可能多的工作在休眠状态，从而达到节能的目的^[2]。以IRIS节点为例，节点在数据分组发送、信道监听时的工作电流为17~22mA，而在休眠模式的工作电流仅为8μA，相差2000倍。节能调度是延长节点寿命的主要手段。

节点的调度方式可以分为2种：集中式和分布式。在小范围的传感器网络中，可以采用集中式调

度，由网关(Gateway)节点统一调度网络中各个底层节点(LN, low-tier energy-limited nodes)的工作和休眠时间。在大规模的传感器网络中，集中式调度的开销较大，效率较低，因此采用分布式调度是解决问题的一个可行性方案。

在大规模网络的分布式调度中，如何保证节点既可以尽可能地休眠，又可以在工作时隙协同工作完成数据采集、处理与传输的协作任务是研究中的难点。因为在分布式调度中，节点的休眠和工作时隙自行决策。以多跳数据通信为例，如果分布式调度的结果是某个节点醒来发送数据时，其父节点仍处于休眠状态，则数据发送将失败。这在农业网络中将造成检测数据丢失。所以解决节点的协同工作与节能调度的矛盾是研究的重点和难点。

针对上述问题，本文提出一种分布式的能量有效的阶梯式调度算法，其核心思想如下。

1) 基于层次树的多跳路由与全网时间同步。

基于Gateway广播的分层路由树构造与全网时间同步，是节能调度的基础。

2) 各层节点阶梯式的协同工作与休眠。

在分层和时间同步基础上，设计了阶梯状节能调度算法和在线的反向时间同步机制，保证节点节能与可靠多跳协作通信。

3) 同层节点基于TDMA的通信时隙调度。

在同层节点间，设计了基于TDMA的通信时隙调度，避免通信冲突，并增强节能效果。

通过理论分析和仿真研究了算法的性能。将上述所提算法在Tinyos上实现，在由60个IRIS节点组成的农业传感器网络中进行了实验测试。所提方法相比免调度方法节能近10倍，相比随机节能调度方法，显著提高多跳可靠性，并可以减少通信冲

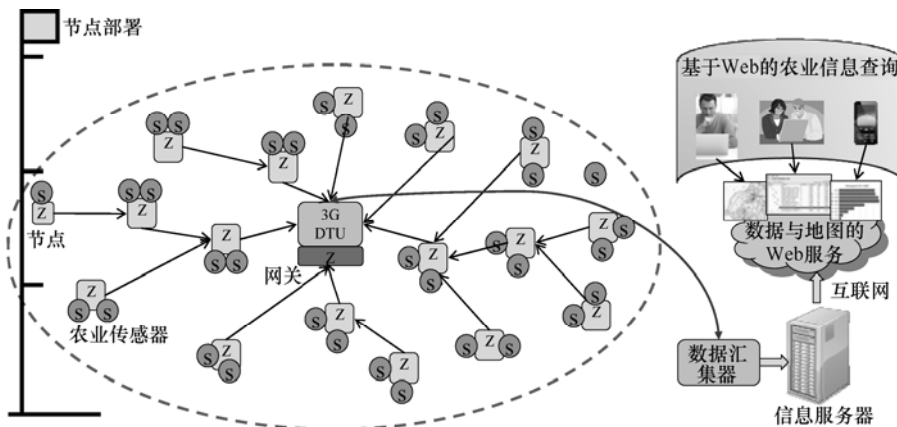


图1 广域农业传感器网络信息监控系统

突和分组丢失。

本文第 2 节介绍相关研究工作；第 3 节具体描述分布式节能调度问题；第 4 节介绍阶梯式节能调度算法；第 5 节介绍分析与实验结果；第 6 节是结束语。

2 相关工作

“休眠—工作”的调度方式是无线传感器网络降低能耗的最主要的解决方案。针对不同的网络模型，采用的具体调度算法也不同，最主要的调度方式有 2 种：集中式调度和分布式调度。集中式调度在实际系统的应用中很少用到，因为使用这种调度方式的通信代价太高。一般在实际系统中都采用分布式的调度方式。例如在一个单跳的网络中，所有的节点通过一跳将数据传到每一簇的簇头 Gateway 节点，而不需要转发其他节点的消息，这样每个底层的节点都可以完全独立地调度自己的工作休眠时间^[2~5]，在文献[5]中，针对在一个簇中部署比较密集的底层节点，根据底层节点到 Gateway 节点的直线距离来做休眠调度，当节点距离 Gateway 节点的距离比较远时，节点会以比较高的概率进入休眠状态。在文献[4]中，针对高密度节点部署中冗余节点较多的特点，每个工作周期只选择小部分的节点处于工作状态，其余的进入休眠状态。在文献[3]中，底层节点的调度则考虑到了能量的冗余和节点感应范围的重叠等问题。

而在大规模的传感器网络中，节点不能采用完全独立的分布式调度算法，分布式调度需要满足协同工作的约束。大规模网络中将所有节点分成若干簇，每簇的节点通过多跳将数据传到簇头，这种网络架构在^[6,7]中做了详细的描述，是比较适合大规模传感器网络的一种架构模式。比如在文献[8]中，采用贪婪算法将每一簇的子网生成一棵以 Gateway 节点为根节点的路由树，整个网络就变成了由若干棵路由树构成的森林。在这种多跳的网络中，底层节点的调度不再是完全独立的，因为节点不但要发送自己采集的数据，还要转发子节点的数据分组，因此底层节点在采用分布式调度算法时，要充分考虑到和周围邻居节点的协同，保证多跳通信的质量。在文献[6]中，提到了一种根据节点到 Gateway 节点的跳数分配不同的工作概率来实现分布式的时间调度，但是该调度算法如何与多跳通信协同工作却没有提到。

3 分布式协同调度问题描述

节能调度是研究如图 1 所示的传感器网络的每簇的成员节点的状态调度与协同工作的问题，包括如下 2 个关键问题。

1) 分布式的节点工作休眠状态切换决策。

在实际系统中，只有节点的休眠和工作时隙调度由每个节点自行独立计算、决策和执行，才能避免集中式算法的规模约束和信息交互的通信代价。

2) 基于分布式状态切换的协同工作。

在分布式状态调度的同时，节点必须协同工作以完成多跳数据通信等任务。

研究在广域农业传感器网络中的一个簇内部节点的节能调度问题。如下网络模型，Gateway 节点和 LN 节点的通信半径都为 R_c ，LN 节点的监测半径为 R_s 。每一簇是一个子网，当路由建立之后，每个子网就形成一个以 Gateway 节点为根节点的路由树。树的高度为 h ，即从 Gateway 节点到最远端的节点的跳数为 h 。根据跳数的不同将路由树分成 h 层，形成一个基于层的拓扑路由树^[6,9]，每层有一个编号 $L_i (1 \leq i \leq h)$ 。设簇内的节点总数为 $N+1$ ，节点编号为 $j (0 \leq j \leq N)$ 。

在层次路由数结构中，分布式节能调度的 2 个子问题可以通过图 2 给出的模型关联起来。对于任选的第 k 层的某个节点 A，其父节点 C 和子节点 B，A、B、C 节点均分布式地自主进行休眠→醒来监听→数据采集→发送数据→休眠的循环状态切换。为进行协同工作，A 的子节点 B 在进行状态调度时，需将其发送数据时隙与节点 A 的监听时隙对齐。对于 A 的父节点 C，需要将其监听时隙与 A 的发送时隙对齐。这样在每个节点分布式状态调度的基础上，实现网络的协同工作。

因此，设 LN 节点的采样周期为 T ，即节点每隔一段时间 T 都要采集一次数据并将数据传到 sink 节点，每个采集周期包括采集数据、接收消息、发送消息和休眠等工作状态。将一个采样周期分为 M 个长度相等的时隙，设接收消息和发送消息可以各在一个时隙中完成，在余下的时隙中，节点进入休眠状态。设 LN 节点在一个采样周期内的时隙集合为 $S = \{S_1, S_2, \dots, S_M\}$ ，调度算法就是要决定 LN 节点在每个时隙中所处的任务状态。

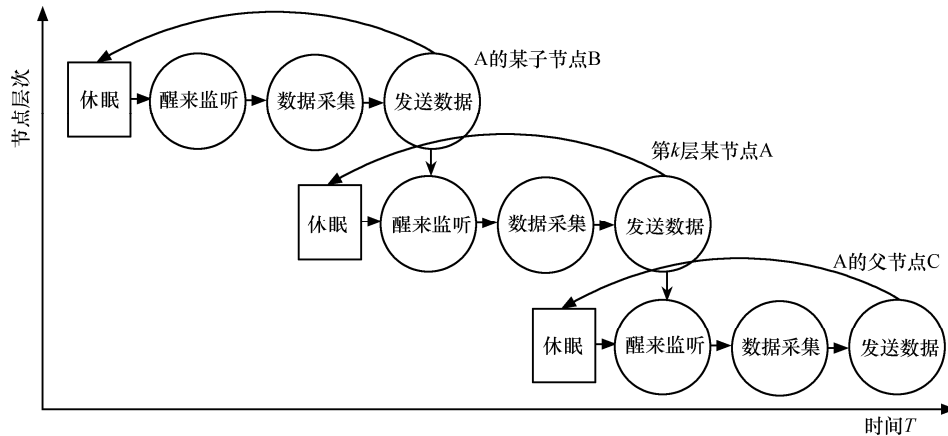


图 2 层次网络中分布式节能调度与协同通信问题

4 分布式能量有效调度算法

针对上述问题，本文提出基于层次分布式阶梯节能调度算法。

4.1 算法描述

算法的主要思想是对每个 LN 节点工作和休眠时隙做调度使其同相邻层的 LN 节点可以协同工作，并支持多跳通信。算法包含 2 个基本的机制。

1) 阶梯状的时间调度机制。在 LN 节点开始调度之前，整个子网由 Gateway 节点发出组网命令，用洪泛的方式在整个子网中生成基于层的拓扑路由树^[6]，当路由建立完成之后，所有节点开始调度。在调度过程中，LN 节点的每个采样周期包含 3 个工作时隙：接收消息时隙(R-Slot)、发送消息时隙(T-Slot)和时间同步时隙(Sync-Slot)。LN 在接受消息时隙接收子节点发送的消息；在发送消息时隙将自身采集的数据打包成一个消息，和接收到的下层子节点的消息一起转发给上层父节点；在时间同步时隙监听父节点发到更上一层节点的消息并作时间同步。

2) 反向时间同步机制。根据前面提到的阶梯状时间调度，当某个 LN 节点在 T-Slot 向父节点发送消息时，该 LN 节点的子节点此时正处于 Sync-Slot，这些子节点会监听到这些消息，并根据这些消息中包含的全局时间信息做时间同步。如果某个子网所形成的路由树的高度为 h ，则经过 h 个采样周期后，距离 Gateway 节点 h 跳之内的所有节点就会和 Gateway 节点的时间达到同步，即整个子网的所有节点的时间会达到和 Gateway 节点同步。

4.2 基于线性回归的初始化时间同步

本文介绍的分布式调度算法的关键是要做到使子网中所有 LN 节点和 Gateway 节点的时间同步，

在系统实现中采用了线性回归的方法，由 LN 节点通过不断的采样 Gateway 节点的时间和自己的本地时间，计算出 2 个时间之间的斜率值和偏移值。采样时间的方法如图 3 所示。通过图 3 可以看到，sender 在发送数据分组之前会经过 CPU Interrupt, Channel access 和 Encode 3 个阶段，其中 Channel access 所需的时间是随机的，因此 sender 在信道获取完成之后，编码之前将自己的全局时间(如果是 Gateway 节点，直接记录本地的系统时间；如果是 LN 节点，则要通过自己的本地时间计算出全局时间)写入要发送的消息中，receiver 在收到消息后，首先对消息解码，解码完成后记录本地时间，同时读出消息中的全局时间。为了使得时间同步更加精确，sender 需要多次给 receiver 发送消息，假设发送的消息数为 n ， x_i 为第 i 次发送时 sender 记录的全局时间， y_i 为第 i 次 receiver 节点记录的自己的本地时间。通过线性回归可以计算出全局时间和本地时间之间的斜率值和偏移值。计算公式如式(1)：

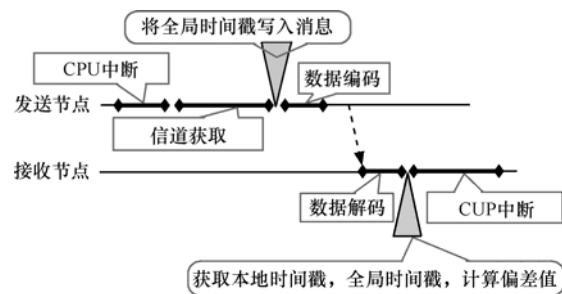


图 3 时间采样

$$slope = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad (1)$$

$$bias = \frac{\sum_{i=1}^n y_i - slope \sum_{i=1}^n x_i}{n} \quad (2)$$

Receiver 通过计算得到斜率和偏移值，这样就可以根据本地时间计算出当前的全局时间。在实际系统中采用这种算法进行时间同步，经过试验发现，IRIS 节点 1ms 的节拍(Click)数为 384，2 个节点做时间同步时，误差在 10 个节拍数左右，我们分别在不同的网络拓扑结构中做了试验，图 4 所示为一个 10 跳的线性网络，包含 11 个节点，1 号节点到 11 号节点传递消息要经过 10 跳，网络中的节点做完时间同步后，经过测试发现 1 号和 11 号节点的时间误差在 150 个节拍数左右，不到 0.5ms。

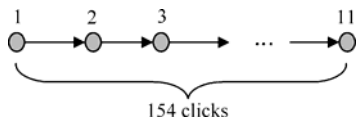


图 4 线性网络的时间同步

在实际系统中，主要通过 2 种方式来应用这种基于线性回归的节点间的时间同步算法，第一是正向的洪泛时间同步方式^[10]，即从 Gateway 节点向外层层的广播，各个节点根据收到的全局时间来计算斜率和偏移值，从而实现时间同步，这种时间同步的方式发生在初始化的子网路由建立阶段；第二是反向的实时的时间同步，这种方式会在下文做详细介绍。

4.3 分布式阶梯状的时间调度

阶梯状的时间调度是针对 LN 节点的，如图 5 所示为一个 LN 节点在一个采样周期中的任务状态转换情况。

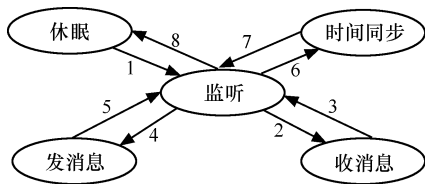


图 5 LN 节点的任务状态转换

调度算法的主要目标就是在一个采样周期内为 LN 节点的这几个任务状态分配时隙。在一个子网中，根据 LN 节点所在的层，计算出该层节点所分配的时隙，使得各层的节点可以协同工作，将数据通过多跳的方式发给 Gateway 节点。在做时间调度之前，Gateway 节点必须对所在的子网做初始化，形成以 Gateway 节点为根节点的路由树，同时以

Gateway 节点的本地时间为标准全局时间，采用 4.1 节中提到的正向的洪泛时间同步方式对子网中的所有节点进行时间同步。

1) 阶梯状时间调度实现细节。网络经过初始化之后形成了以 Gateway 节点为根节点的基于层的路由树，假设路由树的最大层数为 h ，一个采样周期可分的工作时隙数为 M ，并且 $M \geq h + 3$ 。

算法的工作流程如图 6 所示，最外层 h 层的 LN 节点在采样周期的第 $M-h-3$ 个时隙进入工作状态；第 $M-h-2$ 个时隙开始发送消息，此时第 $h-1$ 层的节点正好进入工作状态，开始接受 h 层节点的消息； h 层在第 $M-h-1$ 个时隙进入同步时隙，此时第 $h-1$ 层的节点开始发送消息，第 $h-2$ 层节点开始接受消息，同时第 h 层节点也会监听到 $h-1$ 层节点发送的消息，并根据消息中所带的由第 $h-1$ 层节点计算的全局时间信息作同步，从而实现该调度算法的另一种时间同步方式：反向同步。当接收消息时隙、发送消息时隙和时间同步时隙结束后，该层节点进入休眠状态，等待下一个采样周期的到来。在实际系统实现中，各层节点的采样周期必须是同时开始的，这样才能保证阶梯状调度算法的正常工作。在实际系统中具体的实现方式是这样的，当基于层的路由树建好之后，Gateway 节点会向外广播一个带有时间戳的消息，该时间戳是一个未来的全局时间值，各层的 LN 节点收到该消息后，会计算出这个时间戳和当前全局时间的差值，即该 LN 节点还需要等待多长时间才开始第一个采样周期。这样就可以保证所有节点的采样周期都在同一时刻开始。

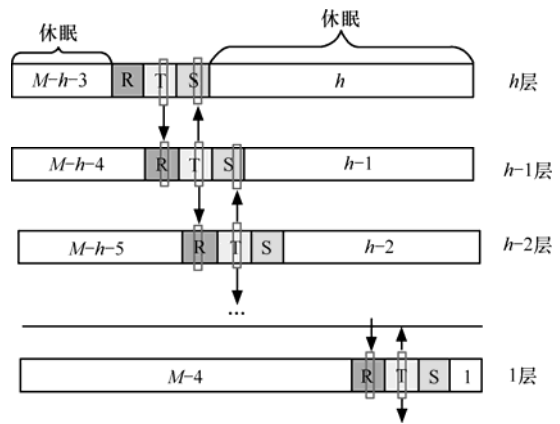


图 6 分布式阶梯状时间调度

通过图 6 可知每层节点的休眠时间是固定的，都是 $M-3$ 个时隙，但是在实际系统的实现中，休眠时间的长度并不是固定的，IRIS 节点的 CPU 对于

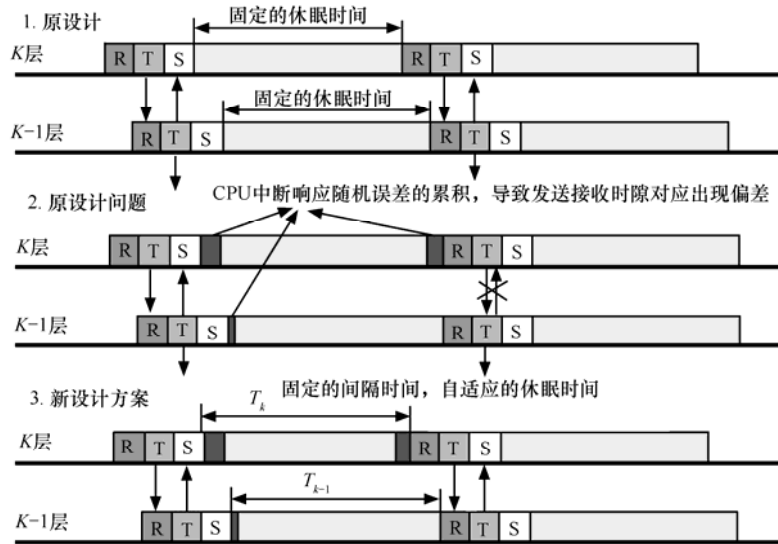


图7 固定休眠时间和自适应休眠时间

事件中断响应会存在一个随机等待的时间，例如在调度算法中 CPU 每次针对休眠事件的中断响应都会有一个随机延迟，随着采样周期的增加，这种延迟带来的负面影响就会越来越明显，最终使得各层节点不能协同工作。因此我们对原算法的休眠时间做了修正，使每层节点的休眠时间由固定时间改为自适应的时间。图7给出了原算法和新算法的比较。

算法1 分布式能量有效调度算法

- ① LocalLevel=getLevel(); //得到 LN 节点所在的层数
 - ② //通过收到的消息，提取开始工作的时间戳
 - ③ Begin time = getStartWorkTime(Receive Msg);
 - ④ //计算距离开始工作的时间间隔
 - ⑤ InterVal = Begin time – getGlobalTime();
 - ⑥ Set startTimer(InterVal); //设置开始工作的定时器
 - ⑦ When startTimer end
 - ⑧ Begin sample circle.
 - ⑨ Set ReceiveTimer, Set SendTimer; Set SyncTimer.
 - ⑩ When SyncTimer end
 - ⑪ Begin sleep;
 - ⑫ Set SleepTimer(adaptive sleep time);
 - ⑬ When SleepTimer end
 - ⑭ Go to step ⑧
- 2) 反向实时的时间同步。容易得到，经过 h

个采样周期，最外层的节点就会和 Gateway 节点的时间达到同步。例如在第一个采样周期，第一层的节点在第 $M-1-1$ 个时隙收到 Gateway 节点的时间同步信息，使得第一层节点和 Gateway 节点作了时间同步。第二个采样周期，第二层的节点在 $M-2-1$ 个时隙监听到第一层的消息，使得第二层和第一层的节点时间达到同步，而第一层的节点在第一个采样周期已经和 Gateway 节点的时间达到同步，由此可以得到第二层的节点也和 Gateway 节点的时间达到同步。以此类推，经过 h 个采样周期，网内所有 LN 节点都会和 Gateway 节点达到时间同步。

3) 同层节点采用 TDMA 方式避免冲突。从图6中可以看到同层节点都在同一个发送时隙发送消息，为了避免冲突，对发送时隙做进一步的划分，划分成 L_{max} 个相等的时间片，同层中的任意一个节点都可以在一个时间片中将消息发送出去，同时假设每层的节点数不能超过 L_{max} ，这样每层的节点可以根据自己的节点号和 L_{max} 值来计算出自己在发送时隙中的哪个时间片发送数据，从而避免同层冲突，具体方法是当 LN 本身的节点号小于 L_{max} 时，时间片索引号就是 LN 的节点号，反之，时间片的索引号为 $L_{max} - LN$ 的节点号。因为在前文假设中做了这样的规定，子网中的节点号各不相同，并且外层节点号要大于内层节点号，因此在 L_{max} 个时间片内，任意一层的所有节点都可以无冲突的将消息发送出去。

5 性能分析和模拟结果

在大规模的传感器网络中，考虑到成本因素，不可能做到冗余部署，因此一般采用“1-coverage”的覆盖模式来部署，首先采用仿真的方法对算法性能进行分析。采用文献[11]的分析结果，LN 节点的部署密度为

$$\lambda = -\frac{\ln(1 - P_{1\text{-cover}})}{\pi r_s^2} \quad (3)$$

其中， $P_{1\text{-cover}}$ 是“1-coverage”的概率。当 Gateway 节点部署好之后，LN 节点部署在 Gateway 节点的周围，则 LN 的节点数 $N = \lambda \pi R^2$ ，其中， R 为整个簇的半径。

为了分析该算法的能量消耗，采用了文献[6,12,13]中提到的能量消耗模型。发送节点能量消耗公式为

$$E_{TX}(l, d) = l \times E_{elec} + l \times \varepsilon_{amp} \times d^\gamma \quad (4)$$

其中， l 为发送数据的比特数， d 为发送节点和接受节点之间的距离， E_{elec} 是发送一个比特数据需要消耗的能量， ε_{amp} 是放大器的能量， γ 是路径丢失因子，通常 $2 \leq \gamma \leq 4$ ，这里取 $\gamma = 2$ 。接收节点的能量消耗公式为

$$E_{RX}(l) = l \times E_{elec} \quad (5)$$

在以下的分析中，发送节点和接收节点之间的距离用跳数来表示，因此，采用文献[6]中提到的在传感器网络中 LN 节点到 CH 节点所需平均跳数的模型：

$$r_h = \sqrt{3\lambda} \int_0^{r_c} x^2 e^{\frac{\pi}{3}\lambda(x^2 - r_c^2)} dx \quad (6)$$

其中， r_c 为节点的传输半径， λ 是节点的部署密度。我们还可以得到整个簇的层数 $h = R/r_c$ ，平均每层的节点数为

$$N_i = \pi \lambda (2i - 1) r_h^2 \quad (7)$$

其中， $1 \leq i \leq h$ ，由此很容易得到第 i 层每个节点的平均子节点数

$$C_i = N_{i+1} / N_i \quad (8)$$

在所介绍的分布式能量有效调度算法中，处于最外层的节点只在“T-Slot”和“Sync-Slot”消耗能量，即

$$E(e_h) = l \times E_{elec} + l \times \varepsilon_{amp} \times (r_h)^\gamma + l \times E_{elec} \quad (9)$$

其余各层节点需要在“R-Slot”、“T-Slot”和“Sync-Slot”消耗能量，即

$$E(e_i) = (2 + C_i) \times l \times E_{elec} + l \times \varepsilon_{amp} \times (r_h)^\gamma \quad (10)$$

令： $E_{elec} = \varepsilon_{amp} = 10^{-10}$ ， $l = 100\text{bit}$ ， $r_c = 200\text{m}$ ， $P_{1\text{-cover}} = 0.999$ ， $h = 10$ ，得到的结果如图 8 所示。

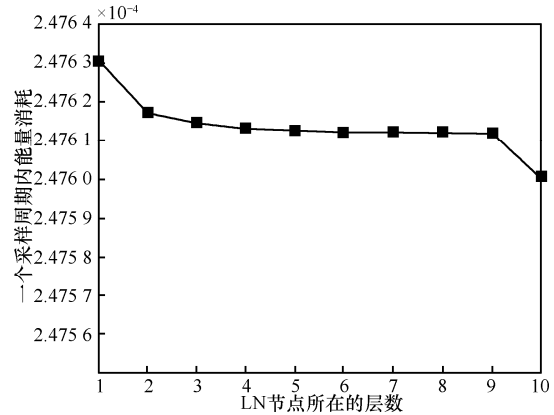


图 8 能量消耗

可以看到，各层节点消耗的能量大体相等，通过计算可以得到最内层节点比最外层节点消耗的能量仅仅多了 0.001 2%。网络中的 LN 节点基本上可以做到能量消耗的平衡。

在实际的传感器网络系统中也做过该分布式能量有效调度算法的性能测试，整个网络包含 60 个 IRIS 节点，其中一个作为 Gateway 节点，其余 59 个为 LN 节点，网内最大跳数为 7。IRIS 节点在工作模式下平均电流为 16mA，在休眠模式下平均电流小于 8μA。如果每个节点使用 2 节 300mA 时的电池供电，当节点一直处于工作模式的时候，可以工作 11 天。采用分布式的能量有效调度算法，将系统的采样周期设为 1min，每个时隙是 1s，测得节点的平均电流小于 1mA，在这种情况下节点可以连续工作超过 100 天。

6 结束语

本文介绍了这种分布式的能量有效的调度算法，首先在网络的初始化阶段，由 Gateway 节点发起组网命令，用洪泛的方式在整个网络中生成基于层的拓扑路由树，同时对网络中的节点作时间同步。初始化结束后，网络进入采样工作阶段，在每个采样周期，LN 节点根据自己所在的层和全局时间形成阶梯状的时间调度，这个过程是完全分布式的，LN 节点只需要知道自己有限的信息就可以完

成该调度, 同时为了避免同层节点的冲突, LN 节点根据已知的 L_{\max} 值计算出所在的发送时间片索引, 从而可以无冲突地将消息发送出去。整个网络通过各层节点的协同调度工作最终形成一个多跳的可靠的无线网络, 同时由于每个节点在一个采样周期中大部分时隙都处于休眠状态, 使得整个网络的能耗也大大降低。

下一步的工作是要降低一个采样周期中工作时隙(R-Slot、T-Slot 和 Sync-Slot)的长度, 使得同层节点在尽量少的时隙中完成接收、发送和时间同步, 同时不会产生冲突。

参考文献:

- [1] WANG Y C, WANG Y X. PAIMS: precision agriculture information monitoring system[A]. Proceedings of CASES'09[C]. Paris, 2009.
- [2] DEMIRBAS M, CHOW K Y, WAN C S. INSIGHT: internet-sensor integration for habitat monitoring[A]. Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM' 06)[C]. Buffalo, NY, USA, 2006.
- [3] WU X L, CHO J S, BRIAN J, *et al.* Sleep nodes scheduling in cluster-based heterogeneous sensor networks using AHP[A]. ICSS 2007[C]. Daegu, Korea, 2007.
- [4] TIAN D, GEORGANAS N D. A coverage preserving node scheduling scheme for large wireless sensor networks[A]. Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications[C]. 2002.
- [5] DENG J, HAN Y S, HEINZELMAN W B, *et al.* Varshney: scheduling sleeping nodes in high density cluster-based sensor networks[J]. ACM/Kluwer Mobile Networks and Applications, 2005, 10(6): 825-835.
- [6] WANG Y, WANG D. Hops-based sleep scheduling algorithm for enhancing lifetime of wireless sensor networks[A]. Proc of MASS[C]. Canada, 2006. 709-714.
- [7] RAMESH G, EDDIE K, DEBORAH E, *et al.* Tenet: an architecture for tiered embedded networks[A]. Proceedings of the 4th International Conference on Embedded Networked Sensor Systems[C]. Boulder, Colorado, USA, 153-166
- [8] SUBHASIS B, NABANITA D. Distributed data gathering scheduling in multihop wireless sensor networks for improved lifetime[A]. ICCTA[C]. Alexandria, Egypt, 2007.
- [9] WOOGUIL P, CHOI J G, BAHK S. Tier based anycast to achieve maximum lifetime by duty cycle control in wireless sensor networks[A]. Wireless Communications and Mobile Computing Conference[C]. 2008. 123-128.
- [10] MIKLOS M, BRANISLAV K, GYULA S, *et al.* Robust multi-hop time synchronization in sensor networks[A]. Proceedings of the International Conference on Wireless Networks (ICWN'04)[C]. San Diego, California, USA, 2004. 454-460.
- [11] WANG Y C, ZHAO Q C, ZHENG D Z. Clusters partition and sensors configuration in wireless sensor networks[A]. Proceedings of ICSS2004[C]. Hangzhou, China, 2004. 237-242.
- [12] HEINZELMAN W B, CHANDRAKASAN A P, BALKRISHNAN H. An application-specific protocol architecture for wireless microsensor networks[J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 660-670.
- [13] ZHAO L, HONG X, LIANG Q. Energy-efficient self organization for wireless sensor network: a fully distributed approach[A]. IEEE Proceedings of Global Telecommunications Conference (GLOBECOM)[C]. Dallas, Texas, USA, 2004.

作者简介:



戚啸 (1983-), 男, 山东淄博人, 清华大学硕士生, 主要研究方向为无线传感器网络技术与应用。

赵春江 (1964-), 男, 河北定兴人, 博士, 国家农业信息化工程技术研究中心研究员, 主要研究方向为农业信息化技术。

王跃宣 (1975-), 女, 浙江金华人, 博士后, 清华大学教授, 主要研究方向为无线传感器网络和网格计算。

王永才 (1978-), 男, 吉林省吉林市人, 清华大学博士后, 主要研究方向为无线传感器网络和网络优化。

陈锦标 (1986-), 男, 广东湛江人, 清华大学硕士生, 主要研究方向为无线传感器网络技术和应用。