软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(3):631−641 [doi: 10.13328/j.cnki.jos.004401]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

低占空比 WSN 中一种节点休眠调度算法*

陈良银[1], 王金磊[1], 张靖宇[1], 王 强[1], 刘 燕[2], 殷 锋[3], 罗 谦[4]

[1](四川大学 计算机学院,四川 成都 610064)

[2](北京大学 软件与微电子学院,北京 102600)

[3](西南民族大学 校园网络管理中心,四川 成都 610041)

[4](中国民用航空总局第二研究所 信息技术分公司,四川 成都 610042)

通讯作者: 陈良银, E-mail: chenliangyin@gmail.com

摘 要: 低占空比无线传感器网络(low-duty-cycle wireless sensor networks,简称 LDC-WSN)可以有效地延长网络生命周期.但是,现有的 LDC-WSN 中端到端的延迟非常大,并且现在很多关于 LDC-WSN 的算法没有充分考虑传输链路质量的问题.为了解决这两个问题,提出了一种基于链路质量和能量感知的节点休眠调度算法(link-quality and energy-aware based scheduling scheme,简称 LES).仿真实验结果表明,相比现在的典型算法,LES 算法能够在满足同样延迟要求的情况下很明显地节省能量,从而延长网络的工作寿命.

关键词: 低占空比;链路质量;延迟控制;节点休眠调度;能量感知

中图法分类号: TP393 **文献标识码**: A

中文引用格式: 陈良银,王金磊,张靖宇,王强,刘燕,殷锋,罗谦.低占空比 WSN 中一种节点休眠调度算法.软件学报,2014,25(3):631−641. http://www.jos.org.cn/1000-9825/4401.htm

英文引用格式: Chen LY, Wang JL, Zhang JY, Wang Q, Liu Y, Yin F, Luo Q. Scheduling scheme algorithm in low-duty-cycle WSN. Ruan Jian Xue Bao/Journal of Software, 2014,25(3):631−641 (in Chinese). http://www.jos.org.cn/1000-9825/4401.htm

Scheduling Scheme Algorithm in Low-Duty-Cycle WSN

CHEN Liang-Yin[1], WANG Jin-Lei[1], ZHANG Jing-Yu[1], WANG Qiang[1], LIU Yan[2], YIN Feng[3], LUO Qian[4]

[1](School of Computer Science, Sichuan University, Chengdu 610064, China)

[2](School of Software and Microelectronics, Peking University, Beijing 102600, China)

[3](Campus Network Management Center, Southwest University for Nationalities, Chengdu 610041, China)

[4](Information Technology Branch, The Second Research Institute of General Administration of Civil Aviation of China, Chengdu 610042)

Corresponding author: CHEN Liang-Yin, E-mail: chenliangyin@gmail.com

Abstract: Low-Duty-Cycle wireless sensor networks (LDC-WSN) can effectively increase the lifecycle of wireless sensor networks. However, in traditional LDC-WSN, not only is end-to-end delay very large, but the quality of link is also not considered. In order to solve these two problems, this paper proposes link-quality and energy-aware based scheduling scheme algorithm (LES). Simulation shows that, compared with traditional networks, LES can achieve significant performance improvement in terms of energy-saving and therefore increase life cycle.

Key words: low-duty-cycle; link-quality; delay control; node scheduling scheme; energy-aware

无线传感器网络(wireless sensor networks,简称 WSN)是一种自组织多跳的无线网络[1],其特征是当一个节

* 基金项目: 国家重点基础研究发展计划(973)(2011CB302902); 国家自然科学基金(61373091, 60933011, 11102124); 国家科技重大专项(2011ZX03005-002-02); 教育部新世纪优秀人才计划(NCET-10-0604); 四川大学青年教师科研启动基金(2011SCU11091)

收稿时间: 2012-09-16; 定稿时间: 2013-03-18

点需要向另外一个节点发送信息时,往往要借助于中间节点、以多跳的方式进行.由于传感器节点大多采用电池供电,而电池所能提供的能量往往十分有限,如何最大限度地延长整个网络的生存时间一直是研究的热点.低占空比无线传感器网络(low-duty-cycle wireless sensor,简称 LDC-WSN)正是在这种情况下产生的[2,3].低占空比无线传感器网络中,节点的占空比一般只有百分之几,极大地延长了网络的生命周期,但是同时也带来了新的问题,即邻居节点之间通信的延迟变大,进而影响到数据的及时有效传输.

对许多实时性要求高的无线传感器网络应用,比如军事应用、灾难预测系统等,需要有效地控制延迟,从而使终端用户能够更快速地获取所需的数据信息,更早地做出有效反应.这就意味着消息由源节点发出后,目标节点必须能够在指定的延迟内接收到.虽然现在已经有一些保证延迟的算法,但是他们没有考虑到链路质量对延迟的影响[4].传统方法对节点能量的动态变化以及低占空比下的操作没有深入地研究[5,6].

本文首先分析了链路质量在 LDC-WSN 中对传输延迟的影响,其次提出了 LES 算法.LES 算法主要由两部分组成:在考虑链路质量的基础上,使得网络能够在满足一定延迟要求的情况下所消耗的能量最小;同时加入能量感知技术,使得节点均匀地消耗能量,从而延长网络的工作时间.LES 算法是在不破坏原有网络结构的前提下,针对延迟问题提出的改进算法.最后,通过在不同环境和参数条件下对 LES 算法和其他节点休眠调度算法进行仿真实验,分析 LES 算法的性能.

1 相关工作

为了克服无线传感器网络的能量寿命问题,能量研究一直是传感器网络研究的重点.近年来,已经建立很多模型用来从节点周围的环境中收集能量,从而对其进行供电[5].然而,目前的研究大多数集中在硬件设计和电源管理方面[7],通过改变节点占空比技术来节省能量方面的研究还很少[8,9].近段时间,ESC 协议[10]通过引入一个透明的中间件来最大程度减小网络传输延迟,从而延长节点寿命.

对许多传感器网络来说,低占空比下的实时数据传输是影响节点能量的一个关键因素.目前,国内外的一些学者也针对节点剩余能量和链路质量等问题提出了一些解决方法.MMSPEED[11]提出了一种多路径多速度的路由协议用来在一定程度上保证网络的链路质量.PTW(pipelined tone wakeup)[8]介绍了一种在能耗节省和端到端延迟之间的平衡方法.Gu 等人提出了 SDC 算法[12],该算法试图使用一种通用发现的方法来减小网络的延迟.SDC 算法能够在特定条件下控制延迟,节省节点的能量消耗.但是,该算法仍然存在着一些不足之处,如并没有考虑链路质量可能引起的数据重传,从而导致节点间的传输延迟增大.

与以往增加额外设备、没有考虑链路质量的研究不同,本文提出了一种在 LDC-WSN 中基于链路质量并且融合能量感知的节点休眠调度方法.其核心思想是:在考虑链路质量,即更加接近于真实环境的条件下控制端到端的延迟,同时加入能量感知来延长网络的生存周期.本方法通过对节点插入苏醒时隙来减小端到端之间的延迟,并根据每个节点的剩余能量值进行能量感知,从而有效地避免节点过早死亡,使整个网络中各节点的能量相对均衡地消耗,延长网络的生命周期.与过去的相关工作比较,本文的难点在于:链路传输成功率并不是 100%情况下的重传以及与之同步进行的能量感知.

2 系统模型

2.1 网络模型

WSN 中的传感器节点通常只有:工作状态和休眠状态.当节点处于工作状态时,除了提供基本的感知等功能外,还必须提供数据发送和接收等基本通信功能,没有数据通信时节点将进行空闲侦听;而当节点处于休眠状态时,节点将会关闭除定时功能外的一切功能[13].

如果用Γ_i表示节点 i 工作调度表的一个周期,那么Γ_i就是由一系列有限的工作状态和休眠状态所组成的集合,T表示节点的一个周期所持续的时间.对于节点 i 来说,它第 j 次处于工作状态可以用数组(t_i^j,τ)来表示,其中,t_i^j 表示节点 i 第 j 次处于工作状态的开始时间;τ则表示第 j 次处于工作状态所持续的时间,即时隙个数,无论节点处于什么状态,时隙大小都是固定的.如果节点 i 一个周期中苏醒的次数为 n,那么可以得出节点 i 的一个周期

工作调度表,见公式(1).

$$\Gamma_i=\{(t_i^1,\tau),(t_i^2,\tau),...,(t_i^n,\tau)\} \tag{1}$$

节点占空比(duty cycle,简称 DC)是指传感器节点所有处于工作状态所持续时间之和与节点从开始到失效所持续时间的比值.对于周期性的节点来说,节点占空比就是在一个周期中处于工作状态的时间和总时间的比值.用 DC_i 表示节点 i 的占空比,根据公式(1)可以得出:

$$DC_i=\frac{\sum_{j=1}^{n}\tau}{T} \tag{2}$$

图 1 表示了节点 A 工作调度表的一个周期.图中用阴影填充的部分表示节点处于工作状态,空白方框则表示节点处于休眠状态,一个方框表示一个时隙.因此,可以计算出节点 A 在一个工作周期内的工作调度表为 Γ_A={(2,1),(3,1),(6,1),(8,1),(9,1),(10,1)},从而可以算出节点 A 的占空比为

$$DC_A=\frac{1+1+1+1+1+1}{10}=60\%.$$

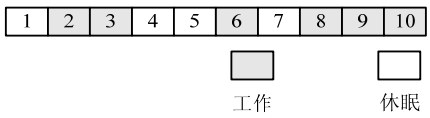


Fig.1 Scheduling table of node A

图 1 节点 A 的工作调度表

2.2 算法假设

本算法模型是建立在满足以下前提假设条件下的:

(1) 节点的工作调度表都是周期性的;

(2) 节点在部署前随机选择其工作调度表,在节点之间进行邻居发现后会共享其工作调度表.节点在更新其工作调度表之前会通知所有邻居节点,在确定其所有邻居节点都知道新的工作调度表后,该节点会在下一次苏醒时启动新的工作调度表;

(3) 节点间的链路质量基本保持不变.研究表明,随着时间的变化,节点间的链路质量变化很小,可以忽略不计[14];

(4) 网络中所有节点是时间同步的,本文中采用 FTSP 协议来保证同步[15];

(5) 节点间在进行数据传输过程中不考虑冲突问题[16,17].

2.3 传输延迟

在所有节点都一直处于工作状态的网络(always active network)中,邻居节点之间可以随时发送或接收信息.这时,节点之间数据传输延迟一般都是毫秒级别,可以忽略[18].但是在 LDC-WSN 中,发送节点必须要等到其邻居节点处于工作状态才能向其发送数据,而这个需要等待的时间可能是几秒甚至几十秒[13].

在本文中,我们将发送节点收到准备传输给其邻居节点的数据信息到邻居节点苏醒处于工作状态所持续的时间定义为休眠延迟.LDC-WSN 中,休眠延迟要比节点之间正常通信所产生的延迟大得多,因此,这里我们在考虑节点端到端的延迟时只考虑其休眠延迟,而忽略通信延迟[18,19].

不考虑链路质量的情况下,即认为数据每次传输都是 100%成功的,延迟计算非常容易,即为传输路径上所有节点之间的休眠延迟之和.但是引入了链路质量之后,数据传输并不是每次都 100%成功.需要引入数据重传,直到数据成功的传输到目的节点.

2.3.1 不考虑链路质量

在图 2 所示的一个简单网络中,节点 A 通过节点 B 发送数据到节点 C,节点 A,B 和 C 的工作周期都为 10 个时隙,即 10τ.节点 A 的工作周期可以用 Γ_A={(1,1)}表示,同理,Γ_B={(5,1)},Γ_C={(8,1)}.这里,我们假定 τ 的值为 1s.如果用 $d_{ij}(t)$ 表示发送节点 i 在时间 t 收到数据信息,然后将该数据信息发送给其邻居节点 j 的休眠延迟,$D_{ij}(t)$ 表示非相邻节点 i,j 的休眠延迟,那么可以计算出节点 A 和节点 B 之间的休眠延迟为 $d_{AB}(1)=(5-1)\cdot\tau=4\times1=4$s,同理可以计算出 $D_{AC}(1)=7$s.

为了描述方便,我们假设节点时隙都为 1s,节点 i 的工作调度表简化为 $\Gamma_i=\{t_i^1,t_i^2,...,t_i^n\}$.

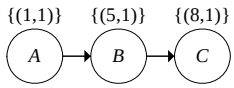


Fig.2 Network transmission model without considering link-quality

图 2 不考虑链路质量的网络传输模型

2.3.2 考虑链路质量

真实环境中,链路之间的数据传输成功率并不是 100%[19].用 p_{ij} 表示节点 i 到邻居节点 j 之间的链路质量,n_{ij} 表示节点 i 向节点 j 传输数据时重新传输的次数,$N_{\max}$ 表示节点的最大重传次数,重传的次数大于 $N_{\max}$ 则丢弃该数据包.

如图 3 所示,假设节点 A 到节点 B 之间的数据链路质量为 p_{AB}=0.8,节点 B 到节点 C 之间的数据链路质量为 p_{BC}=0.6,节点的工作周期 T 为 100s,$N_{\max}$=3,那么我们可以计算出这种情况下节点 A 到节点 C 之间端到端的休眠延迟 $D_{AC}(1)$的期望值:

$$\begin{aligned}E[D_{AC}(1)] &= \sum_{n_{AB}=0}^{2}\sum_{n_{BC}=0}^{2} 0.8\times(1-0.8)^{n_{AB}}\times 0.6\times(1-0.6)^{n_{BC}}\times(4+100\cdot n_{AB}+3+100\cdot n_{BC})\\ &= 0.8\times0.6\times(4+3)+0.8\times0.6\times0.4\times(4+3+100)+0.8\times0.6\times0.4^2\times(4+3+200)+\\ &\quad 0.8\times0.2\times0.6\times(4+100+3)+0.8\times0.2\times0.6\times0.4\times(4+100+3+100)+\\ &\quad 0.8\times0.2\times0.6\times0.4^2\times(4+100+3+200)+0.8\times0.2^2\times0.6\times(4+200+3)+\\ &\quad 0.8\times0.2^2\times0.6\times0.4\times(4+200+3+100)+0.8\times0.2^2\times0.6\times0.4^2\times(4+200+3+200)\\ &\approx 70.32(s).\end{aligned}$$

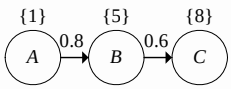


Fig.3 Network transmission model considering link-quality

图 3 考虑链路质量的网络传输模型

在考虑通信链路质量后,端到端的休眠延迟从 7s 增长到约 70.32s,休眠延迟大约是原来的 10 倍.而且,上述计算结果只是基于两跳路径计算的,随着网络中端到端之间跳数的增加,休眠延迟也就会变得更大.由此可见,网络中链路质量对端到端延迟的影响是巨大的.

在本文下面的例子中,都假设 $N_{\max}$ 为 3;T 是节点一个周期所持续的时间,令其为 100s.如果源节点 S 到目的节点 D 之间一条路径上经过的节点依次为 1,2,3,…,m–1,则该条路径在时刻 t 从源节点到目的节点进行数据传输的休眠延迟期望值为公式(3).

$$\begin{aligned}&E[D_{SD}(t)]=\\ &\sum_{n_{s1}=0}^{N_{\max}-1}\sum_{n_{12}=0}^{N_{\max}-1}\cdots\sum_{n_{(m-2)(m-1)}=0}^{N_{\max}-1}\sum_{n_{(m-1)D}=0}^{N_{\max}-1} p_{s1}(1-p_{s1})^{n_{s1}}\,p_{12}(1-p_{12})^{n_{12}}\cdots p_{(m-2)(m-1)}(1-p_{(m-2)(m-1)})^{n_{(m-2)(m-1)}}\,p_{(m-1)j}(1-p_{(n-1)D})^{n_{(m-1)D}}\times \quad (3)\\ &[d_{s1}(t)+d_{12}(t_1)+\ldots+d_{(m-2)(m-1)}(t_{m-2})+d_{(m-1)D}(t_{m-1})+T\times(n_{s1}+n_{12}+\ldots+n_{(m-2)(m-1)}+n_{(m-1)D})]\end{aligned}$$

3 算法设计

LES 算法主要解决的问题是:在链路质量不可靠的 LDC-WSN 中,如何能够在满足特定延迟要求的情况下,尽可能地减少能量消耗,从而最大化网络生命周期.LES 算法主要由以下几步构成:

(1) 节点休眠调度.针对该条路径进行节点休眠调度.节点休眠调度是通过增加节点苏醒时隙的次数,使链路在满足给定延迟要求的情况下,所消耗的能量值最小;

(2) 能量感知.这一步和上一步其实是一起进行的,在进行节点休眠调度的同时进行能量感知.这里,为了方便将它们分开描述.进行能量感知能够使得网络均匀地消耗能量,从而延长网络的工作时间.

3.1 节点休眠调度算法

由于链路质量不可靠的 LDC-WSN 中延迟大而不能够满足实际应用的要求,我们通过增加节点苏醒时隙来减少延迟.如图 4 所示,经过计算得出,在原始状态下,节点 *A* 向节点 *B* 传输的休眠延迟期望值约为 69.02s,节点 *B* 增加一次苏醒时隙后该值变为 11.55s,延迟减少了 83%左右.

(a) 原始状态 (b) 增加一次苏醒时隙

Fig.4 Reduce delay by increasing the wakeup slot

图 4 增加苏醒时隙减小延迟

一般地,对于邻居节点 *i* 和 *j*,当节点 *j* 需要增加一次苏醒时隙以减少休眠延迟,即节点 *j* 的工作周期由 $\Gamma_j=\{(t_j^1)\}$ 变为 $\Gamma_j=\{(t_i^1+1),(t_j^1)\}$ 时,如果 $t_j^1>t_i^1+1$,那么可以得出新增苏醒时隙后节点 *i* 和 *j* 之间的延迟期望值 $E[d_{ij}^1(t)]$,见公式(4).

$$E[d_{ij}^1(t)]=\sum_{n=0}^{N_{\max}-1}p_{ij}(1-p_{ij})^n\left\{1+\left\lfloor n/2+\frac{1}{2}\right\rfloor\times[t_j^1-(t_i^1+1)]+\lfloor n/2\rfloor\times[T-(t_j^1-(t_i^1+1))]\right\} \tag{4}$$

为了满足实际应用的延迟要求,就需要增加节点苏醒时隙次数.本算法就是要使得增加的苏醒时隙次数最小.增加的苏醒时隙次数最小,也就意味着其需要消耗的能量最少.下面介绍该节点调度算法的主要过程.

在本算法中,为了计算出节点 *i* 到节点 *j* 端到端的休眠延迟,我们用 $E[D_{ij}^{m,h}(t)]$ 表示当节点 *i* 在 *t* 时刻收到数据包并准备发送给节点 *j*,在该条路径上增加 *h* 次苏醒时隙所能达到的最小休眠延迟期望值.*m* 为该条路径上节点 *i* 到节点 *j* 所需要的跳数,*h*≤*m*.该算法分为以下两步:

第 1 步,根据初始的节点工作调度表,计算出从源节点 *i* 到目的节点 *j* 的延迟期望值 $E[D_{ij}^{m,0}(t)]$.如果该期望值小于等于实际应用要求的延迟值 *B*,即 $E[D_{ij}^{m,0}(t)]\leqslant B$,则认为已经满足了实际应用的要求,不需要额外增加节点的苏醒次数,该算法退出;否则,进入第 2 步;

第 2 步,从增加 1 次苏醒时隙开始,依次计算出增加 *h* 次苏醒时隙所能达到的最小延迟期望值 $E[D_{ij}^{m,h}(t)]$,直到满足不等式 $E[D_{ij}^{m,h}(t)]\leqslant B$ 或者 *h*=*m*.这时,*h* 的取值是最小的,所要增加的能量消耗也是最小的.这样就保证了从源节点到目的节点在满足限定延迟(*B*)的情况下,所消耗的能量是最小的.

对于邻居节点 *i* 和 *j*,如果不增加节点 *j* 的苏醒次数,则节点 *i* 和节点 *j* 的延迟为原来的值.在一个工作周期中,如果节点 *j* 的苏醒时刻不是比节点 *i* 的苏醒时刻大 1 个时隙的话,那么增加一次节点 *j* 的苏醒时隙次数将会减少节点 *i* 和节点 *j* 之间的休眠延迟.因此,可以得出两相邻节点的延迟期望值满足以下公式(5):

$$E[D_{ij}^{m,h}(t)]=\begin{cases}E[d_{ij}(t)], & m=1,h=0\\ E[d_{ij}^1(t)], & m=1,h=1\end{cases} \tag{5}$$

现在考虑一般的情况,节点 *i* 和节点 *j* 并不是邻居节点,节点 *i* 要向节点 *j* 发送数据信息必须经过某条路径上的其他 *m*–1 跳节点.这样,对于节点 *j* 来说,最后从源节点 *i* 发送过来的数据到节点 *j* 包括以下两种情况:

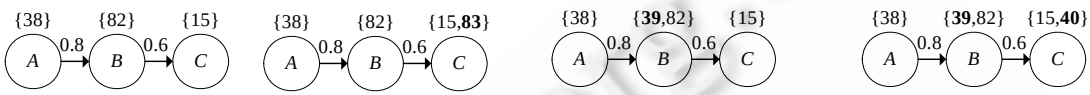
第一,数据包从源节点到目的节点的过程中,节点 1 并不增加苏醒时隙,而是保持原来的工作方式不变,然后在从节点 1 到节点 *j* 总共增加了 *h* 个苏醒时隙;

第二,数据包从源节点到其下一跳节点 1,为了减少休眠延迟期望值,节点 1 增加一次苏醒时隙,从节点 1 到节点 *j* 增加了 *h*–1 次苏醒时隙.

根据上面考虑的两种情况和公式(3),可以通过迭代计算出增加 *h* 次苏醒时隙后,节点 *i* 和节点 *j* 之间的端到端休眠延迟的期望值.因此,可以得到以下公式(6):

$$E[D_{ij}^{m,h}(t)]=\min\begin{cases}E[D_{ij}^{m,h}(t)]\\ E[d_{i1}(t)]\oplus E[D_{1j}^{m-1,h}(t')],\ h>0,m>0\\ E[d_{i1}^{1}(t)]\oplus E[D_{1j}^{m-1,h-1}(t')]\end{cases} \tag{6}$$

在公式(6)中, $E[d_{i1}(t)]\oplus E[D_{1j}^{m-1,h}(t')]$ 表示在节点 1 没有增加苏醒时隙,而从节点 1 到节点 j 增加了 h 次苏醒时隙次数的情况下,节点 i 到节点 j 的延迟期望值.同理, $E[d_{i1}^{1}(t)]\oplus E[D_{1j}^{m-1,h-1}(t')]$ 表示节点 1 增加了 1 次苏醒时隙,而从节点 1 到节点 j 增加了 h–1 次苏醒时隙后节点 i 到节点 j 的延迟期望值,t'表示节点 1 收到数据包的时刻.图 5 展示了整个节点休眠调度算法过程.



(a) 原始状态 (b) 增加 1 次苏醒时隙 (c) 增加 2 次苏醒时隙

Fig.5 Scheduling scheme algorithm

图 5 节点休眠调度算法示意图

在原始状态下,节点 A 到节点 C 的休眠延迟期望值约为 135.32s,如果原始状态不满足实际延迟要求 B,就需要增加一次苏醒时隙.增加一次苏醒时隙后能够取得的最小值为 72.15s,如果 B 仍然比 72.15s 小,就需要再增加苏醒时隙,增加两次苏醒时隙的值为 38.22s.

3.2 能量感知

在实际 WSN 应用中,由于节点的初始能量值不同等许多因素的影响,会导致节点的剩余能量存在很大的差异[20],如果继续过多地使用低能量值节点,将会导致其能量提早耗尽而被废弃.这样的节点个数的增多将会降低网络的性能,进而减小网络生命周期[21].基于以上考虑,本文在节点休眠调度算法的基础上加入能量感知部分,通过能量感知,可以使得剩余能量值低的节点不增加苏醒次数,使其尽可能地处于休眠状态以节省能量.这样虽然可能会增加网络的延迟,但是可以使得网络中的节点均匀地消耗能量,从而会增加整个网络的工作时间.如果用 E_{avg} 表示网络平均能量,节点 j 自身的剩余能量用 E_{res}^{j} 表示.具体算法见公式(7).

$$E[D_{ij}^{m,h}(t)]=\min\begin{cases}E[D_{ij}^{m,h}(t)]\\ E[d_{i1}(t)]\oplus E[D_{1j}^{m-1,h}(t')],\quad h>0,m>0\\ E[d_{i1}^{1}(t)]\oplus E[D_{1j}^{m-1,h-1}(t')],\ E_{res}^{1}>\alpha E_{avg}\end{cases} \tag{7}$$

公式(7)中,α为动态权重值.与进行能量感知前的公式(6)相比较可以发现,在节点 1 试图通过增加苏醒次数来减少延迟之前,会将自己的剩余能量和网络平均能量进行比较.节点 1 要想增加苏醒时隙,必须满足条件 $E_{res}^{1}>\alpha E_{avg}$,只有当其剩余能量大于$\alpha$倍的网络平均能量时,节点 1 才会增加苏醒时隙,否则按照原来的工作调度表工作.经过大量的仿真实验我们发现,当α大概为 1.2 时,能够在较少影响原来节点休眠调度算法性能的基础上延长网络的生存时间.能量感知算法如图 6 所示.

(a) 未考虑节点剩余能量情况 (b) 考虑节点剩余能量情况

Fig.6 Energy-aware process

图 6 能量感知过程

从图 6 可以看出,在没有考虑节点剩余能量的情况下,节点 A 到节点 C 的休眠延迟期望值约为 $E[D_{AC}^{2,1}(23)]\approx$ 45.8s.考虑节点剩余能量后,假如节点 B 剩余能量不足,即 $E_{res}^{B}\leqslant 1.2\times E_{avg}$,节点 B 不增加苏醒时隙次数.而此时,节点 C 满足条件 $E_{res}^{C}>1.2\times E_{avg}$,节点 C 可以增加苏醒时隙,节点 A 到节点 C 的休眠延迟期望值约为

$E[D_{AC}^{2,1}(23)] \approx 80.68$ s.加入了能量感知后,与原来节点休眠调度相比,网络中端到端的延迟期望值变大,但是网络生存周期却得到延长.这也是为了保护剩余能量较少的节点所必须付出的代价.

4 仿真实验与性能评价

本节通过仿真实验验证 LES 算法的性能,并根据不同仿真指标对 LES 算法和其他算法进行对比和分析.

4.1 评价方法

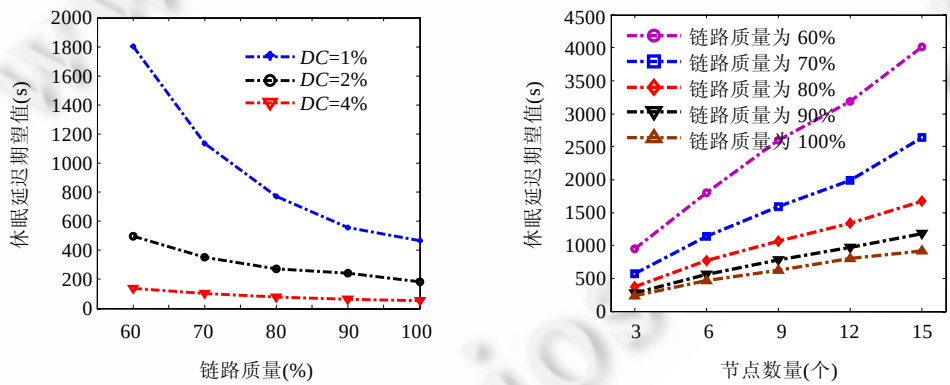
对于 LES 算法的性能分析,主要从以下几个方面进行衡量:

(1) 增加苏醒时隙次数的最小值:在满足实际指定延迟要求的情况下,需要增加苏醒时隙次数的最小值;

(2) 网络的生命周期:从仿真实验开始到因能量耗尽而被废弃的节点数量超过 30%(因为在静态的网络中,当失效的节点数量超过 30%时,网络的性能变得非常差)所持续的时间.

在本节中,将在不同节点占空比、不同节点数量和不同链路质量条件下对休眠延迟期望值进行仿真实验.休眠延迟期望值是评价 LES 算法是否满足实际延迟要求的标准.

从图 7(a)中可以看出:随着节点占空比的增大,端到端的休眠延迟期望值会显著变小,期望值变化曲线也越来越平滑.这是因为随着节点占空比的增大,节点间由于传输失败而进行重传需要等待的时间变短;随着节点间链路传输成功率的减小,端到端的休眠期望值会增大.这是因为链路质量越差,传输失败的概率也会越大.例如,在节点占空比为 1%的情况下,链路质量为 100%的休眠延迟期望值为 465s,链路质量为 60%的休眠延迟期望值约为 1801.33s,增加了 3 倍左右.图 7(b)中,随着链路上节点数量的增多,休眠延迟期望值也会增大.在链路质量为 100%的情况下,节点数量为 3 个的时候端到端的休眠延迟期望值为 232s,传输路径上节点数量为 15 个的时候休眠延迟期望值为 918s.



(a) 链路质量和休眠延迟期望值 (b) 节点数量和休眠延迟期望值

Fig.7 Impact of different conditions on expected sleep delay

图 7 不同条件对休眠延迟期望值的影响

4.2 LES算法性能仿真

在这一节中,我们在不同的延迟要求下对 LES 算法所能满足对应要求延迟的概率进行仿真实验,平台参数的默认值见表 1.平台中,节点在不同状态下所消耗的能量是根据 Deborah Estrin 在 2002 年 Mobicom 会议上提出的耗能模型取的近似值.

根据 LES 算法,随着节点苏醒时隙次数的增加,网络中休眠延迟的期望值会随之减少.但该延迟期望值只是理论值,并不是在实际传输过程中所产生的真正延迟值.为了更真实地模拟 LES 算法性能,我们在 0~1 之间产生一个随机数,如果该值不小于链路质量,则表示该次传输成功;否则,表示传输失败.对每次增加的苏醒时隙次数的仿真次数为 50 次,仿真结果如图 8 所示.

Table 1 Default parameter information of simulation platform

表 1 仿真平台默认参数信息

平台参数(单位)	取值
节点通信、侦听半径(米)	10
节点数量(个)	100
节点占空比	1%
节点每个时隙持续时间(秒)	1
节点工作周期(秒)	100
链路质量	60%~90%
最大重传次数(次)	10
节点发送数据消耗能量(单位能量)	1
节点接受数据消耗能量(单位能量)	0.8
节点空闲状态消耗能量(单位能量)	0.75
节点初始携带能量(单位能量)	20 000~50 000
平均消息产生时间间隔(秒)	1 200

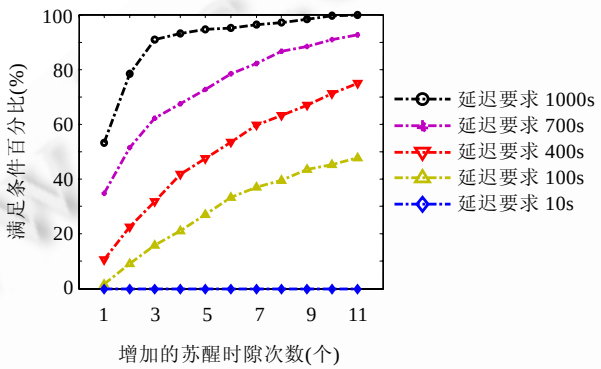


Fig.8 Ratio of LES to meet the conditions under different demanded delay

图 8 LES 算法在不同延迟要求下满足条件百分比

根据图 8 的仿真结果,在不同延迟要求情况下得到的满足条件百分比仍然具有一定的特点.在某一给定延迟要求的情况下,随着节点苏醒时隙次数的增多,满足条件的百分比也会越来越高,这也就说明仿真得到的实际休眠延迟小于或者等于该要求延迟的次数越来越多.例如:当延迟要求为 1000s 的时候,增加 1 次苏醒时隙次数从而满足条件的百分比约为 53.23%;而当增加 3 次苏醒时隙时,该概率则已经达到 90.87%.

4.3 不同参数对算法性能的影响

本节对不同条件下的 LES 算法和 TOSS 算法[22]就前面指出的标准进行仿真实验.仿真中,每次只改变一个影响参数,其余参数均保持默认值.TOSS 算法是由 Cao 等人在单一汇聚节点情况下提出的一种使节点端到端延迟最小的线性节点休眠调度算法.

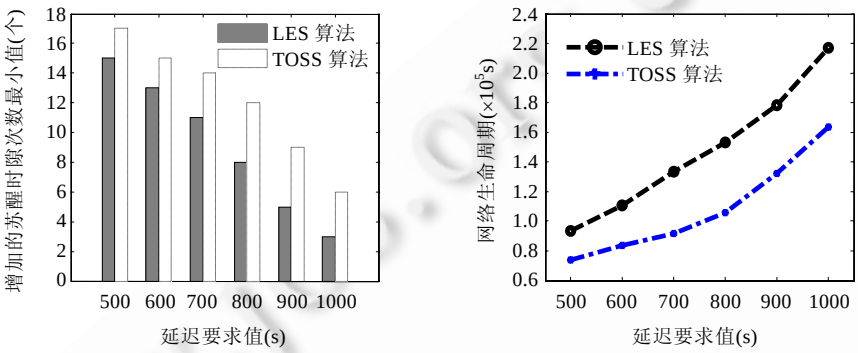
4.3.1 延迟要求的影响

延迟要求是 LES 算法要满足的延迟值,要满足不同的延迟,就需要增加不同的苏醒时隙次数.增加的苏醒时隙次数的不同,会影响网络的生命周期.具体仿真结果如图 9 所示.

从图 9(a)中可以看出,随着延迟要求的增大,为满足该要求值而要增加的苏醒时隙次数最小值会随之减小.这是因为随着要求延迟增大,只需要增加很少的几次苏醒时隙就能满足.

随着网络要求延迟值的减少,LES 算法和 TOSS 算法所需增加的苏醒时隙次数最小值都会增大.但是在相同延迟要求的情况下,LES 算法比 TOSS 算法要增加的苏醒时隙次数少.比如说延迟要求值在 1000s 的情况下,LES 算法要增加的苏醒时隙次数最小值为 3 次,而 TOSS 算法要增加的苏醒时隙次数最小值为 6 次,LES 算法较 TOSS 算法要增加的苏醒时隙次数最小值较少了 50%.随着要求的延迟值越来越小,网络中要增加苏醒时隙次数

的节点也会越来越多.当路径上的所有节点都增加苏醒时隙后,LES 算法就等同于 TOSS 算法,这也是对每个节点都增加一次苏醒时隙所能取得的最小延迟值.在要求的延迟值为 500s 的情况下,TOSS 算法要增加的苏醒时隙次数最小值为 17 次,而 LES 算法要增加的苏醒时隙次数为 15 次,LES 算法仍比 TOSS 要增加的时隙小.LES 算法比 TOSS 算法要增加的苏醒时隙次数少,就表明 LES 算法使节点在更多的时间内将会处于休眠状态,从而可以节省节点能量消耗,延长网络的工作寿命.从图 9(b)中可以看出,随着要求的延迟值减小,在 LES 算法和 TOSS 算法下都会延长网络的生命周期.在要求的延迟值从 500s~1000s 的过程中,LES 算法比 TOSS 算法在最好的情况下将延长 46%的生命周期,在最坏的情况下也会延长 27%的生命周期.



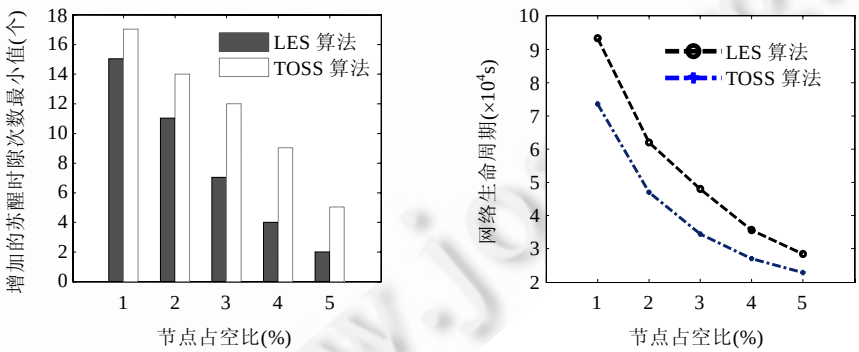
(a) 延迟要求和增加苏醒时隙次数最小值 (b) 延迟要求和网络生命周期

Fig.9 Impact of node's demanded delay

图 9 延迟要求的影响

4.3.2 节点占空比的影响

节点占空比对休眠延迟期望值和节点能量消耗都会产生明显的影响,也将会影响要增加的苏醒时隙次数最小值和网络生命周期.本次仿真在固定延迟要求条件下,考察节点占空比对 LES 算法性能的影响(如图 10 所示).



(a) 占空比和增加苏醒时隙次数最小值 (b) 节点占空比和网络生命周期

Fig.10 Impact of node's duty cycle

图 10 节点占空比的影响

从图 10(a)的仿真结果可以看出,随着节点占空比越来越高,要满足给定延迟要求所需增加的苏醒时隙最小值也会减少.在同一占空比下,LES 算法比 TOSS 算法要增加的苏醒时隙次数最小值要小.在占空比为 2%的情况下,LES 算法要增加的次数最小值为 12 次,而 TOSS 算法则需要 15 次.LES 算法和 TOSS 就需要增加的次数最小值相比,根据仿真结果,在最好的情况下,LES 算法将比 TOSS 少 60%左右.随着节点占空比越来越高,网络的生命周期会显著下降,这是因为节点处于空闲状态的时间会越来越多.在节点占空比从 1%~5%变化的过程中,使用 LES 算法比 TOSS 算法更能够延长网络的工作时间.在最好的情况下延长近 42%,在最坏的情形下也会比 TOSS 算法延长 21%的生命周期.

5 结束语

本文针对低占空比无线传感器网络中链路质量不高、延迟大等特点,提出了 LES 算法.该算法在考虑链路质量的基础上,使得网络能够在满足一定延迟要求的情况下所消耗的能量最小;同时加入能量感知技术,使得节点能够均匀地消耗能量,从而延长网络的工作时间.LES 算法可以为限定延迟下的节点休眠调度算法提供研究基础,同时,在通过能量感知来延长网络生命周期方面提供思路.

本文针对链路质量不可靠的 LDC-WSN 所提出的 LES 算法,在满足相同给定延迟限制的情况下,虽然较其他算法更能延长网络的生命周期,但仍然存在一定的局限性.以下是几点不足和需要改进的地方,也是我们今后研究工作的重点:

(1) LES 算法计算出的延迟值是单向延迟值,并没有考虑双向情况.LES 算法计算延迟的方式是按照从源节点到目的节点多跳进行的,这样,从源节点到目的节点的延迟是满足要求的,但是从目的节点到源节点这个方向上的延迟并不一定是最优的.而无线传感器网络本身就是一个双向网络,因此,如何能够使网络双向延迟值在满足一定要求的条件下,其所消耗的能量最小,这也是一项非常有意义的研究内容;

(2) LES 算法为了减少节点间的休眠延迟采用的是使接收节点只增加一次苏醒时隙的方式,当每个节点都增加一次苏醒时隙时,是 LES 算法所能达到的最小延迟.但是在链路质量不可靠的网络中,有时每个节点只增加一次苏醒时隙并不能满足实际的应用要求,能否让接收节点多苏醒几次,从而在减少休眠延迟的同时使能量消耗达到最小,这也需要进一步的验证理论计算和仿真实验.

致谢 在此,我们向对本课题研究给予支持和指导的明尼苏达大学计算机科学与工程系 Tian He 博士表示衷心的感谢.

References:

[1] Sun LM, Li JZ, Chen Y, Zhu SH. Wireless Sensor Networks. Beijing: Tsinghua University Press, 2005. (in Chinese).

[2] Wang F, Liu JC. On reliable broadcast in low duty-cycle wireless sensor networks. IEEE Trans. on Mobile Computing, 2012,11(5): 767−779. [doi: 10.1109/TMC.2011.94]

[3] Li ZJ, Li M, Liu JL, Tang SJ. Understanding the flooding in low-duty-cycle wireless sensor networks. In: Proc. of the Parallel Processing (ICPP). IEEE Computer Society, 2011. 673−682. [doi: 10.1109/ICPP.2011.56]

[4] Su JS, Hu QL, Zhao BK, Peng W. Routing techniques on delay/disruption tolerant networks. Ruan Jian Xue Bao/Journal of Software, 2010,21(1):119−132 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3689.htm [doi: 10.3724/SP.J.1001.2010.03689]

[5] Dutta P, Hui J, Jeong J, Kim SK, Sharp C, Taneja J, Tolle G, Whitehouse K, Culler D. Trio: Enabling sustainable and scalable outdoor wireless sensor network deployments. In: Proc. of the IPSN 2006. 2006.

[6] Rahimi M, Shah H, Sukhatme GS, Heideman J, Estrin D. Studying the feasibility of energy harvesting in a mobile sensor network. In: Proc. of the ICRA 2003. 2003. 19−24. [doi: 10.1109/ROBOT.2003.1241567]

[7] Kansal A, Potter D, Srivastava MB. Performance aware tasking for environmentally powered sensor networks. In: Proc. of the SIGMETRICS 2004. 2004. 223−234. [doi: 10.1145/1005686.1005714]

[8] Yang X, Vaidya NH. A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay. In: Proc. of the RTAS 2004. 2004.

[9] Zhao T, Guo TD, Yang WG. Energy balancing routing model and its algorithm in wireless sensor networks. Ruan Jian Xue Bao/Journal of Software, 2009,20(11):3023−3033 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3384.htm [doi: 10.3724/SP.J.1001.2009.03384]

[10] Gu Y, Zhu T, He T. ESC: Energy synchronized communication in sustainable sensor networks. In: Proc. of the ICNP 2009. 2009. 52−62.

[11] Felemban E, Lee CG, Ekici E, Boder R, Vural S. Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. In: Proc. of the INFOCOM 2005. 2005. 2646−2657.

[12] Gu Y, He T, Lin M, Xu J. Spatiotemporal delay control for low-duty-cycle sensor networks. In: Proc. of the RTSS 2009. 2009.

[13] Dutta P, Culler D. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In: Proc. of the SenSys 2008. 2008. 71−83.

[14] Lin S, Zhang JB, Zhou G, Gu L, He T, Stankovic JA. ATPC: Adaptive transmission power control for wireless sensor networks. In: Proc. of the SenSys 2006. 2006.

[15] Maroti M, Kusy B, Simon G, Ledeczi A. The flooding time synchronization protocol. In: Proc. of the 2nd Int'l Conf. on Embedded Networked Sensor Systems (SenSys 2004). 2004.

[16] Gu Y, He T. Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks. IEEE Trans. on Mobile Computing, 2011,10(12):1741−1754. [doi: 10.1109/TMC.2010.266]

[17] Su L, Liu CL, Song H. Routing in intermittently connected sensor networks. In: Proc. of the ICNP 2008. 2008. 278−287.

[18] Jackson AW, Sterbenz JPG, Condell MN, Hain RR. Active network monitoring and control: The SENCOMM architecture and implementation. IEEE Trans. on Mobile Computing, 2002,1:70−80.

[19] Boano CA, Zuniga MA, Voigt T, Willig A, Romer K. The triangle metric: Fast link quality estimation for mobile wireless sensor networks. In: Proc. of the ICCCN. IEEE, 2010. 1−7.

[20] Lachenmann A, Marron PJ, Minder D, Rothermel K. Meeting lifetime goals with energy levels. In: Proc. of the SenSys 2007. 2007. [doi: 10.1145/1322263.1322277]

[21] Gu Y, He T. Bounding communication delay in energy harvesting sensor networks. In: Proc. of the ICDCS. 2010. 837−847. [doi: 10.1109/ICDCS.2010.41]

[22] Cao Q, Abdelzaher T, He T, Stankovic J. Towards optimal sleep scheduling in sensor networks for rare event detection. In: Proc. of the 4th Int'l Symp. on Information Processing in Sensor Networks (IPSN 2005). 2005.

附中文参考文献:

[1] 孙利民,李建中,陈渝,朱红松.无线传感器网络.北京:清华大学出版社,2005.8−9.

[4] 苏金树,胡乔林,赵宝康,彭伟.容延容断网络路由技术.软件学报,2010,21(1):119−132. http://www.jos.org.cn/1000-9825/3689.htm [doi: 10.3724/SP.J.1001.2010.03689]

[9] 赵彤,郭田德,杨文国.无线传感器网络能耗均衡路由模型及算法.软件学报,2009,20(11):3023−3033. http://www.jos.org.cn/1000-9825/20/3023.htm [doi: 10.3724/SP.J.1001.2009.03384]

陈良银(1968－),男,重庆人,博士,副教授,主要研究领域为无线传感器网络.
E-mail: chenliangyin@gmail.com

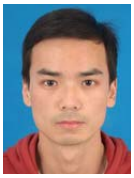
刘燕(1971－),女,博士,副教授,主要研究领域为计算机网络,软件工程.
E-mail: ly@ss.pku.edu.cn

王金磊(1988－),男,硕士,主要研究领域为无线传感器网络.
E-mail: muyishuihan@163.com

殷锋(1972－),男,博士,教授,CCF 会员,主要研究领域为分布式计算.
E-mail: Yf_eagle@swun.cn

张靖宇(1976－),男,博士,讲师,主要研究领域为移动电子商务,通讯软件架构.
E-mail: zhangjingyu@scu.edu.cn

罗谦(1975－),男,博士,高级工程师,主要研究领域为数据挖掘,进化计算和民航机场信息化.
E-mail: caacsri_luoqian@163.com



王强(1986－),男,硕士,主要研究领域为无线传感器网络.
E-mail: Wq1988128@yahoo.com.cn