

# 分簇无线多媒体传感器网络中的能耗控制策略

李兰英, 蒋维成, 何 勇, 朱 晋

(成都理工大学 工程技术学院, 四川 乐山 614000)

**摘 要:**无线多媒体传感器网络中的能量十分宝贵,而控制无线多媒体传感器节点的能量消耗具有非常重要的意义。为此,在建模分析传感器节点能量消耗状况并进行分簇的基础上,提出了一种能耗控制策略,并给出了适合无线多媒体传感器网络的能耗速率色带表示方法及其计算方式,有效地简化了传感器节点的信息处理和计算操作,减少了能量消耗。所提出的能耗控制方法能够协调各节点能量消耗快慢,保持能耗水平趋于相同,使节点能量在网络有效生存期内得以充分利用,减少了无用残存能量的现象,延长了网络生存期。仿真实验结果表明,相对于 LEACH 算法,所提出的方法具有更长的网络存活期,更迟的节点死亡出现时间,死亡百分之五、百分之十和百分之十五的时间均要晚得多,有效地验证了所提出方法的正确性。

**关键词:**无线多媒体传感器网络;网络生存有效期;分簇;能耗速率色带

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2017)07-0174-06

**doi:**10.3969/j.issn.1673-629X.2017.07.039

## Energy Consumption Control Strategy in Wireless Multimedia Sensor Networks with Clustering

LI Lan-ying, JIANG Wei-cheng, HE Yong, ZHU Jin

(Engineering & Technical College, Chengdu University of Technology, Leshan 614000, China)

**Abstract:**Energy is a valuable resource in wireless multimedia sensor networks. It is of great significance to control the energy consumption of wireless multimedia sensor nodes. Through analyzing and processing the energy consumption of each node, a new energy consumption control strategy with clustering has been proposed, in which the method of the energy consumption rate ribbon representation and calculation for wireless multimedia sensor networks has been given. This strategy has simplified the information processing and computing as well as the decrease of energy consumption. This proposed method can coordinate the speed of energy consumption of each node and make full use of node energy in the network lifetime, which reduces unnecessary residual energy and prolongs the network lifetime. Simulation results show that compared with the LEACH algorithm, the method proposed has a longer network survival time and later node death time and that the deaths time of five percent, ten percent and fifteen percent nodes are much later having verified the correctness of this method.

**Key words:**wireless multimedia sensor networks; network lifetime; clustering; energy consumption rate ribbon

## 0 引言

无线传感器网络中,传感器常布置在无人值守的野外环境,进行环境监测、数据采集等<sup>[1-3]</sup>。普通传感器价格不高,区域内存在冗余节点。然而对无线多媒体传感器网络来说,多媒体传感器价格较高,往往仅在重要区域才进行布置。一个无线多媒体传感器常负责某一具体区域(或角度)的监视<sup>[4-6]</sup>,一次性布置完成,中途不再更换。因此,区域内某一无线多媒体传感器节点的过早死亡,势必影响到对该区域的监测。如何

防止区域内无线多媒体传感器节点提早死亡,保持各无线多媒体传感器节点的能耗一致具有重要意义。

文献[7]为解决分簇不均导致小簇簇头能耗多而过早死亡的问题,提出采用邻近极大极小簇协作,均衡了网络能耗,提高了通信可靠性。文献[8]提出非均匀布置传感器节点,利用冗余节点,延长了网络生命周期。文献[9]提出了基于环扇交错的非均匀分簇路由算法,簇头选举考虑节点的剩余能量及与其他节点的距离,均衡了网络能耗,延长了网络生存期。但上述算

收稿日期:2016-08-15

修回日期:2016-11-23

网络出版时间:2017-04-28

基金项目:四川省教育资助项目(16ZB0404);成都理工大学工程技术学院基金资助项目(C122015008)

作者简介:李兰英(1979-),女,硕士,讲师,研究方向为计算机网络。

网络出版地址:<http://kns.cnki.net/kcms/detail/61.1450.TP.20170428.1704.098.html>

法都没有对网络中的残存能耗问题进行讨论。文献[10]考虑簇头节点和协作节点的剩余能量和位置信息,提出了均匀成簇协作通信方法。但成簇时需要 Sink 节点多次参与,系统开销大。文献[11]中两次执行 LEACH 成簇路由算法,分别选出簇头节点和协作节点,有效减少了无线传感器网络的能量消耗,延长了网络生命周期,但成簇时容易出现簇的大小不一,从而导致整个网络的能量消耗不均衡。文献[12]提出了基于区域划分集合的思想,将所有传感器按区域划分,在一定程度上解决了网络能量均衡问题,但区域划分计算过于复杂,难以进行扩展。

网络生存有效期并不是指从网络开始布置到网络中最后一个节点死亡这一段时间。网络内传回的监视数据要有一定利用价值。随着能量不断消耗,节点开始死亡。当死亡数量达到一定值之后,将存在较多区域无法监视,这时传回的数据毫无用处。此刻虽有传感器节点存活,已不属于网络生存有效期的范畴。网络生存有效期之外还存活的无线多媒体传感器已经没有利用价值,其保留的剩余能量也是一种浪费。

文献[13]指出,当传感器网络中死亡数量超过20%时,网络生存有效期就结束了。网络生存有效期结束后还活着的这20%节点,如果能在网络生存有效期还没有结束之前,帮助协作其他节点,减少能耗,势必延长网络生存有效期,充分地利用网络中所有节点的能量。为此,必须考虑无线多媒体传感器网络中的能耗控制。

为了协调无线多媒体传感器各节点能量消耗速率的快慢,使各节点的能耗保持一致,提出了协调节点能耗速率的一种方法。主要做了如下方面的工作:

(1) 构建了算法模型,对网络节点的能量消耗进行实时处理,发布相应的能耗控制消息,节点根据消息,进行相应的耗能调节,保持能量消耗一致。

(2) 给出了能耗速率色带表示方法及相应的处理方式,简化了操作。

提出算法旨在于协调节点的能耗速率,实现下列目标:协调各节点能量消耗快慢,使节点能量在网络生存有效期内充分利用,减少网络死亡后仍存活的传感器剩余能量;防止节点过早死亡,避免相关区域缺乏监视;保持各节点能耗均衡,延长网络生存有效期。

## 1 算法模型

根据节点的地理位置,通信半径和监测任务要求,将网络划分成若干虚拟监测子区域( $a_1, a_2, \dots, a_n$ ),各子区域构成一个簇。每个簇内有一个簇头节点和若干成员节点。簇头节点负责管理本簇内事务,协调各节点工作,接收簇内成员发来的监测数据,进行数据融

合与处理,以及数据转发等。成员节点负责监测和采集数据,并将采集数据发送给簇头节点。为了保证各节点通信有效进行,簇内的任意两个节点都在有效通信半径内,相邻两簇中的节点也能直接进行通信,节点可以获得整个区域的相关信息。

由于簇头节点负责接收簇内所有成员节点采集的数据,并与其进行通信,簇头就可以获得想要的信息。为了能够掌握一段时间里簇内无线多媒体传感器节点的能耗状态,成员节点把自身能耗信息附加在数据帧中传送给簇头。簇头通过提取其中的能耗信息,对所有成员的能耗信息进行建模分析,簇头就可以掌握簇内的能耗状况,进行相关的决策和控制,实现对成员的能耗管理。并且这些相关信息在簇头轮换时,可以转交给下一轮簇头节点继续保存和使用。

为了实现对簇内能耗的管理和控制,在簇头节点增加三个功能模块,分别是能耗信息收集模块、能耗信息处理模块和能耗控制模块。成员节点增加能耗信息模块和能耗调整模块。提出算法的功能模块和工作流程如图1所示。

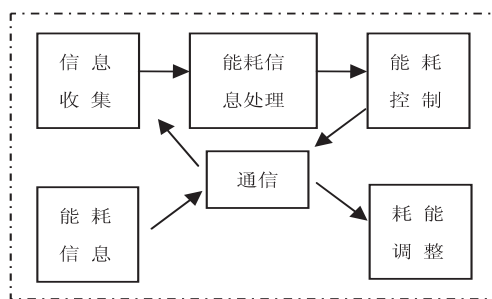


图1 算法模块结构

簇头节点的信息收集模块负责收集簇内各成员节点的能耗信息。通过能耗信息处理模块进行计算处理。能耗控制模块根据处理结果进行相应的决策控制发布相应的能耗控制信息。为了减轻簇头的处理负担,成员节点的能耗信息模块负责该节点的能耗信息计算处理,并把处理好的能耗相关信息发送给簇头。成员节点收到簇头发送的能耗控制信息后,根据节点能耗情况,结合所处的环境及具体任务要求,采取相应的能耗调节措施。为了更好地讨论能耗问题,定义如下参数:

定义1(能耗速率):指单位时间里节点能量减少量,用 $R_i$ 表示。它用来衡量一个节点能量消耗的快慢,能耗速率较大的节点,将在较短时间内耗尽能量,较早死亡。设( $t_1, t_2$ )这段时间节点 $i$ 的能量由 $E_1$ 变为 $E_2$ ,那么它的能耗速率为:

$$R_i = \frac{|E_2 - E_1|}{t_2 - t_1} \quad (1)$$

定义2(能耗相对速率):指某节点能耗速率与其他节点的能耗速率比值,用 $\beta$ 表示。它是反应节点相

对其他节点死亡时间快慢的参考指标。设节点  $i$  的能耗速率为  $R_i$ , 节点  $j$  的能耗速率为  $R_j$ , 则有:

$$\beta = \frac{R_i}{R_j} \quad (2)$$

当  $\beta > 1$  时, 节点  $i$  的能耗速率要大于节点  $j$ , 对于剩余能量相同的无线多媒体传感器节点来说, 节点  $i$  的死亡时间将早于节点  $j$ 。而  $\beta < 1$  时, 死亡时间迟于节点  $j$ 。

定义 3(能耗平均速率): 指区域内节点能耗速率的平均值, 用  $A$  表示。它反映区域内节点能量消耗快慢的整体状况。对于有  $n$  个节点的区域, 其能耗平均速率表示为:

$$A = \frac{\sum R_i}{n} \quad (3)$$

定义 4(能耗相对平均速率): 指节点  $i$  能耗速率与域内能耗平均速率的比值, 用  $\gamma$  表示。

$$\gamma = \frac{R_i}{A} \quad (4)$$

当  $\gamma > 1$ , 表示节点  $i$  的能耗速率要快于区域内平均能耗速率, 该节点的能量消耗比域内平均水平要多, 有可能提前死亡, 因而该节点应采取措施减小能量消耗。而对于  $\gamma < 1$  的节点, 它距离死亡时间较远, 剩余能量比其他节点要多, 能够为网络贡献更多的能量。

## 2 能耗速率色带表示及计算

簇头收集一段时间内簇内成员能耗速率进行计算处理, 并将簇内能耗平均速率信息发布给簇内的无线多媒体传感器节点, 成员节点根据收到的能耗平均速率信息, 将该信息保存下来, 与自身能耗速率进行比较, 判断上一时段是否大于能耗平均速率, 进而采取相应的能耗调整措施。

为了减轻簇头的负担, 能耗速率计算处理的任务分担到簇内每个成员节点, 各成员节点负责计算处理一段时间内自身的能耗速率信息, 并将处理好的能耗速率信息发送给簇头。

由于能耗速率是实时变化的, 节点数量较多时, 处理起来十分繁琐复杂。为了降低处理的复杂性, 简化计算, 同时考虑无线多媒体传感器网络特点和要求, 采用色带来表示能耗速率。每种色带表示一定范围的能耗速率大小。范围的大小可以根据具体应用环境进行设置。采用八种色带表示能耗速率大小, 每一种色带的范围值为 10。

设红 (red)、橙 (orange)、黄 (yellow)、绿 (green)、青 (indigo)、蓝 (blue)、紫 (purple)、棕 (brown) 分别对应的能耗速率范围为 10 ~ 19, 20 ~ 29, ..., 80 ~ 89。这样八种不同色带的表示仅需 3 bit。

为了简化计算, 提高处理效率, 采用如下方式进行计算。对于一个有  $n$  个无线多媒体传感器的簇, 簇头保存上一时段的能耗平均速率色带信息 ( $C_o$ ), 计算处理时, 仅考虑那些与色带  $C_o$  不相同的节点数量和色带值。设能耗速率色带  $C_i$  的节点数量为  $K_{ij}$ , 那么簇内能耗平均速率色带值可以根据式(5)进行计算。

$$C_1 = \begin{cases} C_o + \lfloor \sum K_{ij} \cdot (C_i - C_o) / n \rfloor, & (C_{\min} \leq C_1 \leq C_{\max}) \\ C_o, & (C_1 < C_{\min} \text{ 或 } C_1 > C_{\max}) \end{cases} \quad (5)$$

其中,  $C_{\min}$  为色带最小值;  $C_{\max}$  为色带最大值。当计算结果超过色带范围值时, 取原来的色带值。 $\lfloor \cdot \rfloor$  表示取整数部分, 这里取整数一方面是为了使计算结果转化为相应的色带值; 另一方面使能耗平均速率尽量停留在原来的色带上, 减少部分节点意外情况产生能耗速率的变化而影响簇内平均能耗速率的变化。

由式(5)可以看出, 免去了与能耗平均速率色带相同的那部分节点的计算处理, 而这些节点数量占大多数, 大大减少了计算量。能耗速率色带与能耗平均速率色带不同的节点数量较少, 这样计算量就小。另外式(5)比较简单, 计算并不复杂。因而只需要消耗极少的能量就可以得出这一时段的能耗平均速率。

各节点保存本簇的能耗平均速率色带信息, 为了减少通信量, 当成员计算得到自身能耗速率色带与簇内能耗平均速率色带相同时, 不发送能耗速率色带消息到簇头。在能耗速率色带与簇内能耗平均速率色带不相同, 才将其能耗速率色带信息附加在数据帧中传给簇头。另外, 簇头计算得到这一时段的能耗平均速率色带与上一时段相同时, 不发送能耗控制消息。在色带改变时, 才将相应的能耗控制色带信息发送给簇内成员。

计算能耗速率是在已知上一时段的能耗速率基础上进行的, 在网络运行初始时, 不存在上一时段的能耗速率。网络初始总能量是可以知道的, 结合网络生存期的预期值, 就可以对能耗速率进行初始化。

为了协调相邻簇的能耗速率, 簇头与相邻簇头之间可以进行通信, 获取相邻簇的能耗速率色带信息。设有  $m$  个相邻的簇, 根据式(6)来设置下一周期本簇的能耗平均速率色带。

$$C_i = k \left( \sum C_m / m \right) + (1 - k) \cdot C_1 \quad (6)$$

其中,  $C_m (i = 1, 2, \dots, m)$  为相邻簇的能耗速率色带值;  $C_1$  为式(5)计算得到的本簇能耗平均速率色带;  $k (0 < k < 1)$  为贡献因子, 表示相邻簇能耗速率对本簇能耗速率的影响大小。

取相邻簇能耗色带信息作为本簇下一时段能耗速



率设置的一个参考因素,实现与相邻簇能耗速率保持一致。

为了方便处理,减少通信,约定一个特殊时间段内进行能耗信息的传递和处理工作,非此时段不进行这类处理。需要发送能耗速率色带信息的节点,仅在一个周期的最后一个时间片里进行。簇头的能耗控制信息,也仅在一个周期的第一个时间片里进行。设  $t_e$  表示一个周期的最后一个时间片,  $t_b$  表示一个周期的第一个时间片。簇头节点的 type 值为 cluster, 别的为成员节点。  $C_o$  为上一周期的能耗速率色带,  $C_i$  表示待求的这一周期的能耗速率色带。flag 为数据帧中是否包含能耗信息的标识位。

算法 1: 能耗速率色带处理。

输入: 成员节点能耗速率色带, 相邻簇能耗速率色带

输出: 簇内能耗速率色带设置值

过程:

```
while(  $T == t_e$  )
{
  if( type == cluster )
  {
    检查数据帧的 flag 位;
    while( flag = 1 )
    {
      提取色带信息;
      根据式(5)计算本簇能耗速率色带;
       $C_i \leftarrow$  根据式(6)计算设置的能耗速率色带;
    }
    if( type  $\neq$  cluster )
    {
       $C_i \leftarrow$  计算节点能耗速率色带;
      if(  $C_i \neq C_o$  )
      {
        flag 位置 1;
         $C_i$  封装在数据帧中发送到簇头;
      }
    }
  }
  while(  $T == t_b$  )
  {
    if( type == cluster )
    {
      if(  $C_i \neq C_o$  )
      {
        确认帧 flag 置 1;
        能耗速率色带信息封装在确认帧中发送给成员;
      }
      if( type  $\neq$  cluster )
      {
        检查收到 ACK 帧的 flag 位;
        if( flag == 1 )
        {
           $C_i \leftarrow$  提取色带信息;
          if(  $C_i \neq C_o$  )
          {
            采取能耗调节措施;
          }
        }
      }
    }
  }
  While(  $T \neq t_e \&\& T \neq t_b$  )
  {
    if( type == cluster )
    {
      簇头正常工作;
    }
    if( type  $\neq$  cluster )
  }
```

{ 监视和采集数据; }

}

### 3 能耗调节和节能措施

簇头对能耗信息进行处理后,根据需要对成员节点发送能耗控制信息。成员收到簇头的能耗控制信息后,提取出这一周期内的能耗速率色带信息  $C_i$ , 与过去一周期里自身能耗速率色带信息  $C_o$  进行比较。采取不同的处理措施。

(1)  $C_i$  与  $C_o$  相等的节点,上一周期能耗速率色带与能耗控制信息中的能耗速率色带相同,符合能量消耗标准,不进行能耗调节。

(2)  $C_i$  大于  $C_o$  的节点,能耗速率色带值大于控制信息色带,在上一周期消耗的能量比能耗标准要多,在这一周期内需要降低能量消耗。可以求助其他节点,缩小监视范围或监视时间来降低能量消耗的速率,也可以通过降低采样频率,减少数据处理和发送量以降低能耗速率。根据不同应用对数据质量的要求和节点所处的环境采取相应措施。

(3)  $C_i$  小于  $C_o$  的节点,能耗速率色带值小于控制信息色带,在上一周期消耗的能量比能耗标准要少,有能量剩余,在这一周期里可以贡献更多的能量。可以扩大监视范围,增加监视的时间帮助协作其他节点,合理利用过剩的能量。

无线多媒体传感器网络中节点的能耗主要包括节点的通信能耗(  $E_t$  )、发送数据能耗(  $E_s$  )、接收数据能耗(  $E_r$  )、数据采集能耗(  $E_c$  )。

为了节省能量,减少处理和存储上的开销,仅在一个周期的两个时间片里进行能耗速率色带的计算和处理,增加 4 bit 的处理开销,其中 1 bit 表示是否含有能耗速率色带信息,3 bit 表示 8 种不同能耗速率色带信息。消耗的时间和能量是很少的,在可接收的范围之内。另外,色带信息是附加在数据帧中的,没有增加额外的通信能耗。

文献[14]给出了能量模型,节点在发送数据过程中消耗的能量采用式(7)进行计算。

$$E_s = \begin{cases} (E_o + E_{fs} d^2) S, d < d_0 \\ (E_o + E_{mp} d^4) S, d > d_0 \end{cases} \quad (7)$$

其中,  $E_o$  为每发送或接收 1 bit 数据所消耗的能量;  $E_{mp}$  和  $E_{fs}$  是通信能量常数;  $S$  为发送数据块长度;  $d$  为发送节点到接收节点之间的距离。

接收数据的能耗为:

$$E_r = E_o \times S \quad (8)$$

在分簇路由协议 LEACH 中,采用单跳的方式将数据传输给 Sink,当簇头到 Sink 的距离大于  $d_0$  时,传输的能耗  $E_s$  与距离  $d^4$  相关,距离小于  $d_0$  时,传输能耗

与  $d^2$  相关。可见传输距离超过  $d_0$  时所花费的能量显著增加。此外,随着无线传输距离的增加,信号传输误码率也将增加。为了节省能耗,对于距离超过  $d_0$  的数据均采用多跳的方式进行,使得各跳距离在  $d_0$  范围内。

文献[15]指出,发送数据、接收数据、空闲状态、休眠的能耗比为 2 000 : 400 : 400 : 1。传感器节点传输信息时要比执行计算更消耗电能,传输 1 bit 信号到相距 100 m 的其他节点大约需要的能量相当于执行 3 000 条计算指令消耗的能量。为了提高传输效率,减少通信开销,对时延要求不高的数据,采用如下措施减少能耗。

(1)成员节点采集的数据不是立即组装成帧发送,而是将数据存储起来,达到一定长度之后再传输给簇头,这样可以减少发送次数和通信开销。对于四个长度为  $L$  的数据帧,分四次发送消耗的能量为  $4E_s + 4E_t$ ,其中  $E_s$  为传输能量消耗,  $E_t$  为通信开销。而组装成一个长度为  $4L$  的数据帧连续发送,消耗能量为  $4E_s + E_t$ ,减少了  $3E_t$  通信能量消耗。此外,这种方式还有助于减少碰撞的发生,从而减少因碰撞造成双方能量损失。

(2)采集的数据不是直接传输给簇头,而是在本地经过压缩后再传输给簇头,减少了传输数据的量,节省了能量。

4 仿真结果

以 Matlab 和 OPNET 作为仿真平台,模拟比较提出算法和 LEACH 算法的性能。实验参数设置如表 1 所示。

表 1 实验参数设置

参数名称	参数值
初始能量/J	0.4
Sink 位置	(100,50)
$E_0$ /(J/bit)	$50 \times 10^{-9}$
$E_{fs}$ /(J/bit)	$10 \times 10^{-12}$
$E_{mp}$ /(J/bit)	$0.001\ 3 \times 10^{-12}$

(1)节点死亡数比较。无线多媒体传感器节点随着能量的不断消耗,出现死亡。随着死亡数的增加,整个网络将死亡。网络节点死亡数的比较如图 2 所示。

应用文中算法时的节点出现死亡的时间晚于 LEACH 算法,网络中节点的死亡数要明显少于 LEACH 算法。表明提出的能耗控制策略能有效减少节点的死亡。

(2)节点存活数比较。存活的节点越多,网络生存期就越长。随着网络生存时间的增加,节点存活数

不断减少。网络节点存活数的比较如图 3 所示。

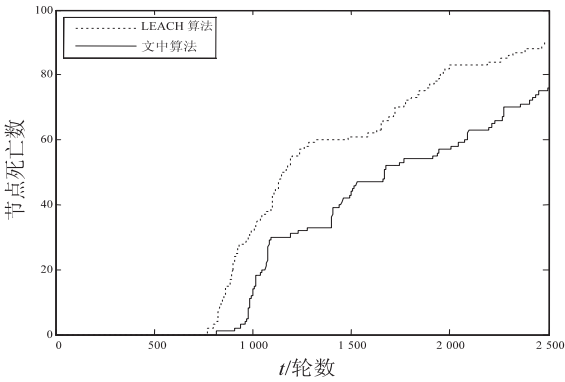


图 2 死亡数

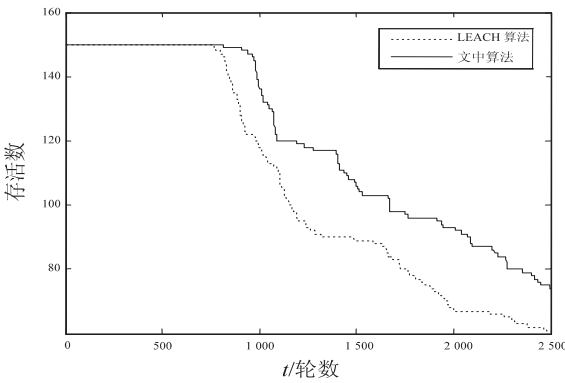


图 3 存活数

采用提出算法的节点存活数要远多于 LEACH 算法。表明提出算法网络生存时间比 LEACH 算法长。

(3)节点死亡百分比的比较。节点死亡数超过一定百分比后,一些区域将无法监视,监控效果也不断下降。图 4 是每轮中随机 60 个节点发送数据情况下,死亡数占总数百分之五、百分之十和百分之十五的比较。

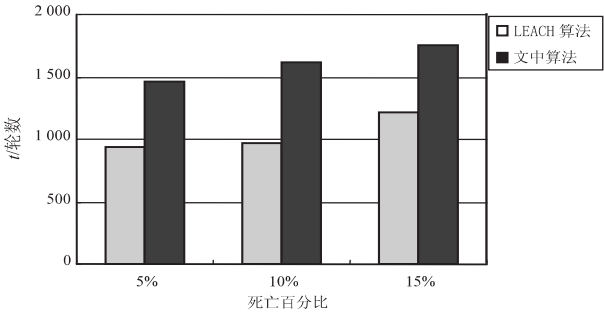


图 4 死亡百分比

从图上可以看出,提出算法在这三种情况下都比 LEACH 算法要晚得多。LEACH 算法出现百分之十五的死亡数在第 1 221 轮,而提出算法是在第 1 457 轮。表明提出算法在能耗控制上的优势,有效地延长了网络存活有效时间。

5 结束语

从能耗速率的角度,提出了一种无线多媒体传感器网络中的能耗控制方法,给出了适合无线多媒体传

传感器网络能耗速率的色带表示及计算处理方法,简化了计算。通过对簇内成员的能耗速率色带信息进行分析处理,为能耗决策控制提供了依据,节点根据控制消息采取相应的方式进行耗能调节。该方法能保持节点的能耗速率一致,避免节点的过早死亡而影响对该区域内的监视,有助于保持网络节点能耗均衡化,减少各节点剩余能量的差异,使节点能量在网络死亡之前充分利用起来,减少无用残余能量存在,保证了网络内无线多媒体传感器节点能量的高效利用,有效延长网络存活时间,实验结果也验证了这一点。

参考文献:

[1] Ma H D,Dan T. Multimedia sensor network and its research progresses[J]. Journal of Software, 2006, 17 ( 9 ) : 2013 – 2028.

[2] Shen H, Bai G, Tang Z, et al. QMOR: QoS-aware multi-sink opportunistic routing for wireless multimedia sensor networks[J]. Wireless Personal Communications, 2014, 75 ( 2 ) : 1307–1330.

[3] 周 灵,王建新. 无线多媒体传感器网络路由协议研究[J]. 电子学报, 2011, 39 ( 1 ) : 149–156.

[4] 赵 静,曾建潮. 无线多媒体传感器网络感知模型与数量估计[J]. 软件学报, 2012, 23 ( 8 ) : 2104–2114.

[5] 张军强,王汝传,黄海平. 基于分簇的无线多媒体传感器网络数据聚合方案研究[J]. 电子与信息学报, 2014, 36 ( 1 ) : 8–14.

[6] 张乾燕,黄海平,沙 超. 无线多媒体传感器网络中一种自适应的休眠调度策略[J]. 科学技术与工程, 2012, 12 ( 28 ) :

7226–7230.

[7] 黄舜尧,黄生叶. 成簇无线传感网中能量高效的协作通信[J]. 计算机工程与应用, 2015, 51 ( 15 ) : 75–79.

[8] 袁辉勇,刘伟群,刘永逸,等. 传感器网络中一种非均匀的节点部署算法[J]. 传感技术学报, 2008, 21 ( 6 ) : 1039 – 1042.

[9] 李法平,芮素娟,廖玉霞. 基于环扇交错的无线传感器网络分簇算法[J]. 科学技术与工程, 2013, 13 ( 7 ) : 1975–1978.

[10] Qin Yongming, Tang Qiuling, Liang Ye, et al. An energy-efficient cooperative MIMO scheme for wireless sensor networks based on clustering[C]//14th IEEE international conference on computational science and engineering. [ s. l. ] : IEEE, 2011 : 471–474.

[11] Kong H Y. Energy efficient cooperative LEACH protocol for wireless sensor networks[J]. Journal of Communications and Networks, 2010, 12 ( 4 ) : 358–365.

[12] Przydatek B, Song D, Peering A. Secure information aggregation in sensor networks[C]//Proceedings of the first ACM conference on embedded networked systems. [ s. l. ] : ACM, 2003 : 75–79.

[13] Maleki M, Dantu K, Pedram M. Power-aware source routing protocol for mobile Ad hoc networks[C]//Proceedings of the 2002 international symposium on low power electronics and design. [ s. l. ] : [ s. n. ], 2002 : 72–75.

[14] 孙 毅,刘浩程,陆 俊,等. 基于两步前向区域空洞预测的 WMSNs 路由算法[J]. 传感技术学报, 2015, 28 ( 1 ) : 132 – 136.

[15] 杨博雄,倪玉华. 无线传感网络[M]. 北京:人民邮电出版社, 2015.

(上接第 173 页)

Android application[J]. Journal of Computer and Communications, 2014, 2 ( 8 ) : 54–63.

[2] 赵 亮,张 维. 基于 Android 技术的界面设计与研究[J]. 电脑知识与技术, 2009, 5 ( 29 ) : 8183–8185.

[3] Shukran M A M, Sharif W S S B. Android augmented reality system in malaysia military operations – unit positions [ J ]. Australian Journal of Basic and Applied Sciences, 2012, 6 ( 8 ) : 79–82.

[4] 彭春华,刘岳峰,晏 磊,等. 基于 Java 的移动地理信息服务关键技术研究[J]. 计算机工程与应用, 2007, 43 ( 11 ) : 190–192.

[5] Jaradat A, Mohamad N A, Asadullah A, et al. Issues in location based marketing; a review of literature [ J ]. International Journal of Scientific and Research Publications, 2015, 5 ( 1 ) : 1–4.

[6] 吕志平,赵冬青,徐爱民,等. 位置服务系统 ( LBS ) 的构建[J]. 测绘科学, 2005, 30 ( 2 ) : 92–94.

[7] 赵志弘,李志林,余 萌. 涉及位置的信息服务:关键趋势和商务模式[J]. 地理信息世界, 2003, 1 ( 3 ) : 14–18.

[8] Shah P, Gadgil R, Tamhankar N. Location based reminder using GPS for mobile ( Android ) [ J ]. ARPN Journal of Science and Technology, 2012, 2 ( 4 ) : 377–380.

[9] 刘 正,陈 强. 基于 Android 的 GPS 应用中的研究与开发[J]. 中国西部科技, 2010, 9 ( 32 ) : 15–16.

[10] Niknejad A. A quality evaluation of an Android smartphone application [ D ]. Michigan: Eastern Michigan University, 2011.

[11] 姚昱旻,刘卫国. Android 的架构与应用开发研究[J]. 计算机系统应用, 2008 ( 11 ) : 110–112.

[12] 侯春良,张义文. 基于 Android 平台的地理信息系统工具设计与开发[J]. 地理空间信息, 2012, 10 ( 5 ) : 14–16.

[13] 孙晓宇. Android 手机界面管理系统的设计与实现[D]. 北京:北京邮电大学, 2009.

[14] 公 磊,周 聪. 基于 Android 的移动终端应用程序开发与研究[J]. 计算机与现代化, 2008 ( 8 ) : 85–89.