

文章编号: 1001 - 9081 (2005) 09 - 1982 - 04

基于感知数据概率模型的无线传感器网络采样和通信调度算法

李建中, 石胜飞, 王朝坤

(哈尔滨工业大学 计算机科学与工程系, 黑龙江 哈尔滨 150001)

{ljzh, shengfei, chaokun}@hit.edu.cn

摘 要: 在无线传感器网络中, 如何动态地管理能量, 最大限度地延长网络的生命周期是一个关键的问题。文中提出了一种基于感知数据概率模型的传感器网络的采样和通信动态调度算法, 使传感器节点根据感知数据的概率模型来确定自己的采样和通信时机, 最小化采样频率和通信量, 减少传感器节点的能量消耗, 延长传感器网络的生命期。该算法是一种分布式算法, 适用于无线传感器网络。该算法采用了简单的概率模型, 资源需求量大, 适合于在目前普遍使用的资源受限的传感器节点上运行。模拟试验结果表明, 这种方法与其他方法相比, 具有很高的能量有效性。

关键词: 无线传感器网络; 感知数据概率模型; 动态调度; 能量管理

中图分类号: TP311 **文献标识码:** A

Sampling and scheduling algorithm for wireless sensor networks based on sample data probability model

LI Jian-zhong, SHI Sheng-fei, WANG Chao-kun

(Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin Heilongjiang 150001, China)

Abstract: A key challenge in wireless sensor networks is to achieve maximal network lifetime with dynamic power management on sensor nodes. In this paper an adaptive node sampling and scheduling strategy based on probability model of data generated by sensor nodes was investigated. Compared to previous works on dynamic power management, this method concentrated on the characters of data. The node can schedule its sampling and communication time according to the prediction value of the model. The results of simulation confirm the energy efficiency of the algorithms compared with previous algorithms.

Key words: wireless sensor networks; sample data probability model; dynamic scheduling; energy management

1 问题的提出

随着通信技术、嵌入式计算技术和传感器技术的飞速发展和日益成熟, 具有感知能力、计算能力和通信能力的微型传感器开始在世界范围内出现。由这些微型传感器构成的无线传感器网络引起了人们的极大关注。这种传感器网络综合了传感器技术、嵌入式计算技术、分布式信息处理技术和通信技术, 能够协作地实时监测、感知和采集网络分布区域内的各种环境或监测对象的信息, 并对这些信息进行处理, 获得详尽而准确的信息, 传送到需要这些信息的用户^[1,2]。在网络工作过程中节省能源, 最大化网络的生命周期, 是无线传感器网络需要解决的重要问题。为了节省能源, 需要研究如何尽量减少传感器节点间的通信量, 如何减少传感器节点的监听时间, 以及如何在满足系统监测精度的前提下最小化采样次数。

对某些特定区域或目标进行监测是无线传感器网络的主要用途^[3]。在许多实际应用中, 例如监测某区域内的温度、湿度和亮度等, 被监测区域要布置多个相同类型的传感器节点。它们通过合作的方式来完成同一监测任务。传感器节点根据用户的要求进行数据采样并通过多跳方式把数据传送给用户。为了减少传感器节点的能量消耗, 在满足用户的精度

要求的前提下, 动态调整节点采样的周期, 并减少数据的传输数量, 从而减少传感器节点的能量消耗。本文提出了一种基于感知数据概率模型的采样和通信的动态调度算法: 每个传感器节点根据感知数据的概率模型来确定自己的采样和通信时机, 最小化采样频率和通信量, 减少传感器节点的能量消耗, 以延长传感器网络的生命期。这是一种分布式算法, 适用于无线传感器网络, 采用了简单的概率模型, 资源需求量大, 适合于在目前普遍使用的资源受限的传感器节点上运行。

2 相关工作

文献[4]提出了数据为中心的调度算法 DCS (data-centric scheduling)。这个算法从无线通信的角度考虑能量节省问题, 其核心思想是在保证无线 Ad Hoc 通信丢包率较低的情况下, 尽量减少每个节点处于通信或者监听状态的时间, 以节省传感器网络的能量。在 DCS 算法中, 每个节点根据历史信息, 动态地预测下一个数据包到来的时间。此外, 每个节点还可以判断这些数据包到达的频率是否发生了变化, 并对预测的下一个数据包到来的时间做动态调整。DCS 算法的优点在于考虑了数据包产生频率的多样性, 并适合传感器节点分布密度较大的监测环境。文献[5]提出了 FPS 调度算法, 该算

收稿日期: 2005 - 06 - 26 基金项目: 国家自然科学基金资助项目 (60473075)

作者简介: 李建中 (1950 -), 男, 黑龙江哈尔滨人, 教授, 博士生导师, 主要研究方向: 数据库、数据仓库、并行计算、无线传感器网络; 石胜飞 (1972 -), 黑龙江哈尔滨人, 讲师, 博士研究生, 主要研究方向: 移动计算、无线传感器网络; 王朝坤 (1976 -), 江苏东台人, 博士研究生, 主要研究方向: 音乐数字图书馆、P2P。

法适应于网络中通信需求动态变化的特点。FPS算法适合于基于树状的拓扑结构,目的是调度每个节点收发信息的时间片,减少通信的冲突和丢包。文献 [6]提出的算法使节点仅在被需要时才从睡眠状态转换到通信状态,以节省能量。文献 [7]中提出了一种分布式节点调度协议 LORD。该协议在保证无线传感器网络的覆盖性和连通性基础上,调度每个节点进入活动或者睡眠状态。上述算法都没有考虑数据本身的特性,没有很好地解决能量节省问题。

文献 [8~10]提出一类预测查询处理方法。这些方法考虑了感知数据本身的特性,试图解决上述方法的问题。他们首先对传感器网络进行建模,然后根据概率模型进行未来感知数据的预测并进行查询处理。但是,文献 [8~10]提出的方法是集中式算法,需要把所有感知数据传送到中心处理节点,要求大量的网络通信,导致高能量消耗,仅适合与所有传感器节点距离中心处理节点都很近的情况,不适合于一般的无线传感器网络。

本文提出了一种基于感知数据概率模型的传感器节点采样和通信的分布式动态调度算法,既考虑了感知数据本身的特性,克服了文献 [4~7]提出的方法的弱点,也克服了文献 [8~10]提出的集中式方法存在的缺点。

3 基于简单概率模型的采样与通信调度算法

3.1 感知数据的建模算法

许多无线传感器网络的应用采用基于“簇”的体系结构。根据应用的要求,对于一个监测目标,传感器网络中一般具有多个传感器节点对其进行监测。这些节点构成了一个“簇”,其中一个节点被选为簇头(Cluster head)。簇的成员节点在簇头的调度下协同完成采样和建模任务。

由于传感器节点可以使用的资源很有限,因此在节点上执行的建模算法要尽量简单,使用的内存也要小。为此,使用了比较简单的 $AR(P)$ 模型进行感知数据建模。我们首先介绍感知数据建模的方法。

由于传感器节点采样的数据是一个时间序列,可以使用概率模型来表示这个时间序列。由于传感器节点的资源有限性,我们使用较容易在传感器节点上实现的 $AR(P)$ 模型^[11]来表示传感器节点的采样数据。 $AR(P)$ 模型可以具有如下形式:

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \dots + \alpha_P X_{t-P} + \varepsilon_t(t-P), \quad \varepsilon_t \sim N(0, \sigma^2) \tag{1}$$

其中, X_t 是在时刻 t 采样得到的数据, α_i 是回归系数, ε_t 是白噪声序列,服从标准正态分布。 P 阶模型(记作 $AR(P)$)用前 P 个采样数据的一个线性组合加上一个随机白噪声来预测时刻 t 的数据。

$AR(P)$ 模型可以用于预测。例如,对于 $AR(1)$ 模型,在时刻 t 能以 95% 的置信度向前进行 k 步预测:

$$X_t(k) \pm 1.96 \sigma \sqrt{1 + \alpha_1^2 + \alpha_1^4 + \dots + \alpha_1^{2(k-1)}}$$

其中, $X_t(k)$ 可以由公式 (1) (令 $P = 1$) 递归计算出来。

使用 P 阶 $AR(P)$ 模型进行建模,簇内的节点需要 N 个采样值: X_1, X_2, \dots, X_N ,然后通过最小二乘法求出系数 $\alpha_1, \alpha_2, \dots, \alpha_P$ 。令:

$$Y = (X_{p+1}, X_{p+2}, \dots, X_N)^T$$

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_P \\ X_p & X_{p+1} & \dots & X_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ X_1 & X_2 & \dots & X_{N-P} \end{pmatrix}^T$$

则公式 (1) 能够表示成 $Y = A \alpha$, 系数矩阵 A 可以通过最小二乘法计算:

$$\alpha = (A^T A)^{-1} A^T Y \tag{2}$$

把计算公式 (2) 的任务划分为多个子任务,每个节点执行其中的一个。令 $A_t = (X_t, X_{t+1}, \dots, X_{t+p-1})^T$,表示截止到时刻 t 长度为 p 的采样序列。注意到在时刻 $t+1, A_{t+1} = \text{Push}(D \text{rop}(A_t), X_{t+1})$,即首先去掉 A_t 中最开始的采样数据 X_{t-p+1} ,然后加入新的采样数据 X_{t+1} 。于是矩阵 A 的转制矩阵 A^T 可以表示成:

$$A^T = (A_p, A_{p+1}, \dots, A_{N-1})$$

为了计算式 (2),分别进行如下计算:

$$A^T A = \begin{pmatrix} N-1 \\ k=p \\ N-1 \end{pmatrix} A_k A_k^T \tag{3}$$

$$A^T Y = \begin{pmatrix} N-1 \\ k=p \end{pmatrix} X_{k+1} A_k \tag{4}$$

公式 (3) 和 (4) 将任务 (2) 中大的矩阵运算分解为长度为 p 的向量运算,而且不占用很大的存储空间,在实际应用中, p 通常的取值不会很大,例如 $p = 5$ 就可以拟合很多实际的序列^[11]。

为了保持传感器网络各节点的能量均衡,把上述计算任务按照簇中每个节点剩余能量的多少分成若干个子任务。每个节点所要完成的子任务包括:在任务规定的时间范围内进行采样,对采样数据计算公式 (3) 和 (4) 中的部分和,将上述两个部分和返回簇头,最后由簇头节点完成公式 (2) 的计算。任务的划分以及调度由簇头节点来完成。

由于 $AR(P)$ 模型假设时间序列的均值为 0,算法首先需要估计时间序列的均值。然后,每个节点把采样值减去均值的估计值,进行 0 均值化处理。

可以根据上述建模方法,给出感知数据的建模算法。在给出感知数据建模算法之前,首先给出算法中使用的符号。如下所示:

- M : 一个 cluster 内部参加计算的节点数目。
 - $Power_i$: 节点 i 剩余能量。
 - N : 采样总数。
 - N_i : 分配给节点 i 采样的个数。
 - L_i : 指定节点 i 开始采样的时间。
 - μ_t : 时刻 t 计算得到的样本均值。
 - T_m : Cluster head 节点计算初始均值所需的采样时间。
 - X_j : 时刻 j 的采样值。
 - P : $AR(P)$ 模型的阶数。
 - ARA_i : 节点 i 计算的 $A^T A$ 部分值 (公式 3)。
 - ARY_i : 节点 i 计算的 $A^T Y$ 部分值 (公式 4)。
 - A_i : 节点 i 计算公式 (3) 和 (4) 中的向量 A_k 。
 - S_i : 用于计算均值的各个节点采样值的和。
- 感知数据的建模算法定义如下:
- 算法 1 分布式传感器网内建模算法

输入: N 个采样数据

输出: $AR(P)$ 模型的数矩阵, 白噪声序列的方差²

在 Cluster head 节点执行如下计算:

$$(1) \mu_t = \left(\sum_{j=1}^{N_i} X_j \right) / \left\{ X_j \mid t - T_m < j < t \right\};$$

// 计算在时间段 T_m 内采样值的均值, 初始化 μ_t

$$(2) N_i = \text{Max}(N * Power_i / \sum_{j=1}^m Power_j, P + 1);$$

// 根据节点剩余能量分配要采样数据的个数, 最少 $P + 1$ 个

$$(3) L_i = \left(\sum_{j=1}^{i-1} N_j \right) - P + 1, \text{ 其中 } 1 < i \leq m, L_1 = 1;$$

// 计算每个节点开始采样的时间

(4) Cluster head 节点将上述步骤的结果通知给 cluster 内部其他参与计算的节点;

进入睡眠状态;

(5) 按照调度计划, cluster head 在每个成员返回子任务的结果时转入工作状态, 接收结果。当所有子任务结束后, 执行如下计算:

$$(6) A^r A = \sum_{i=1}^m ARA_i \quad // \text{计算公式 (3)}$$

$$(7) A^r Y = \sum_{i=1}^m ARY_i \quad // \text{计算公式 (4)}$$

$$(8) = (A^r A)^{-1} A^r Y$$

$$(9) \mu = \left(\sum_{i=1}^m S_i \right) / N$$

// 根据其他节点返回的部分和计算均值

(10) Cluster head 节点将参数 μ 通知给其他节点, 并且接收从其他节点返回的残差值² 以及 $Power_i$;

$$(11) \sigma^2 = \left(\sum_{i=1}^m \epsilon_i^2 \right) / m$$

// 计算 $AR(P)$ 模型中白噪声序列的方差

在 Cluster 成员节点:

根据 cluster head 节点的调度计划, 在时间区间 $t_i = [L_i,$

$L_i + N_i]$, 节点 i 执行如下的步骤:

(1) 计算分配给自己的建模子任务:

a) $ARA_i = ARY_i = 0;$

b) While($t_i \leq L_i + p - 1$)

c) Begin

d) $A_i[t_i] = X_{t_i} - \mu;$

// 对最初的 P 个采样数据进行零均值处理

e) $t_i = t_i + 1;$

f) End

g) $S_i = 0;$

h) While($t_i \leq L_i + N_i$)

i) Begin

j) $ARA_i = ARA_i + A_{t_i-1} A_{t_i-1}^r;$ // 计算公式 3

k) $ARY_i = ARY_i + (X_{t_i} - \mu) * A_{t_i-1}^r;$ // 计算公式 4

l) $A_{t_i} = \text{Push}(D \text{ rop}(A_{t_i-1}), X_{t_i});$

m) $t_i = t_i + 1;$

n) $S_i = S_i + X_{t_i}^2;$

o) End

(2) 将 ARA_i , ARY_i 和 S_i 返回 cluster head 节点。转入睡眠状态;

(3) 根据调度计划, 在 cluster head 发送参数矩阵的时

候, 唤醒;

(4) 计算并返回 μ_i 以及 $Power_i$;

$$L_i + N_i - p$$

$$\mu_i = \sum_{k=L_i+N_i-p}^{L_i+N_i-1} X_k$$

3.2 采样和通信调度算法

通过算法 1, 每个簇对所监测目标的采样数据建立了 $AR(P)$ 模型。我们可以使用这个 $AR(P)$ 模型以及用户指定的精度, 动态唤醒节点、执行采样以及传送数据, 从而尽量减少能量的消耗。我们的采样和调度算法的基本思想是: 利用 $AR(P)$ 模型的预测功能, 根据历史数据来预测将来的数据。同时, 根据用户的精度要求, 动态调整采样时间间隔。如果预测的值满足精度要求, 则不需要启动传感器节点进行实际采样和数据传送, 适当延长节点的睡眠时间。如果预测值误差超过了精度要求, 则一方面要将真实的采样和数据传送, 另一方面需要对 $AR(P)$ 模型进行适当的调整。

对于 $AR(P)$ 模型, 在时刻 t 的 t 步前向预测值可以用以下公式计算:

$$X_t(l) = \sum_{k=1}^p G_k X_{t-k}$$

对应的 $1 - \alpha$ 置信区间可以为:

$$X_t(l) \pm z_{\alpha/2} (1 + G_1^2 + \dots + G_{t-1}^2)^{1/2} \quad (5)$$

其中, $(z_{\alpha/2} = 1 - \alpha/2)$, $G_j = \sum_{i=1}^j G_{j-i}$, $G_0 = 1$, 为标准正态分布函数, G_j 称为格林函数 (Green Function)^[11]。

在我们的采样和通信调度算法中, 当一个节点被唤醒后, 它首先进行采样, 然后检查使用公式 (5) 计算的预测值与当前采样值的误差, 根据误差决定下次采样的时间、是否需要传送数据以及模型调整等。

下边, 我们给出采样和时间调度算法。下面给出了算法中使用的符号的定义:

T : 初始的采样周期。

t_0 : 节点开始采样的时间。

len : 指定一个节点采样的次数。

t : 当前的时间。

T_1 : 调整后的采样周期。

X_t : 时刻 t 的预测值。

x_t : 时刻 t 的采样值。

$error_bound$: 误差允许的阈值。

n_error : 系统制定的采样周期内误差超出阈值的次数。

算法 2 无线传感器网络采样和通信调度算法

1) $T_1 = T$; // 节点初始的采样周期

2) 每隔 T_1 的时间间隔, 节点被唤醒, 设当前时刻为 t , 执行下列操作:

3) If ($t < t_0 + len * T$)

// 当前时间仍处于指定采样时间范围内

4) Begin

5) If ($|X_t - x_t| > error_bound$)

6) Begin

7) $n_error++;$

8) if $n_error > threshold$ of system

9) $T_1 = \max(T, T_1 - T/2);$

// 缩短采样的周期, 获得更多的精确采样数据

10) return x_t ; // 返回真实的采样数据

11) End

```
12)      Else
13)       $T_1 = T_1 + T/2;$            //增加采样的周期
14)      End
15)      Else           //指定的采样任务已经完成
16)      Go to the sleep mode
```

4 性能分析

主要从以下几个方面考察上述算法在不同精度要求下的能量有效性: 实际传送的数据量以及节点处于工作状态的时间。下面列出了模拟系统的主要参数配置:

时间序列: 生成的 $AR(3)$ 和 $ARMA(4, 3)$ 序列, MOTE^[12] 模拟生成的序列

节点数目: 30
cluster数目: 5

节点实际传送的数据量如图 1 所示。从图中可以看出, 采用了满足不同置信度的概率模型之后, 需要传送的数据量大大减少。图中实时监测表示不对数据进行建模, 而是返回所有采样的数据。对于采用预测模型的方法, 随着置信度的增加, 由于预测数据与实际采样数据的误差增加, 因此需要传送的数据也随之增加。但是, 从图 1 我们可以得出结论, 在满足应用精度要求的基础上, 适当调整预测数据的置信度可以有效减少传感器网络中数据的传输量, 从而有效减少网络能量的开销。

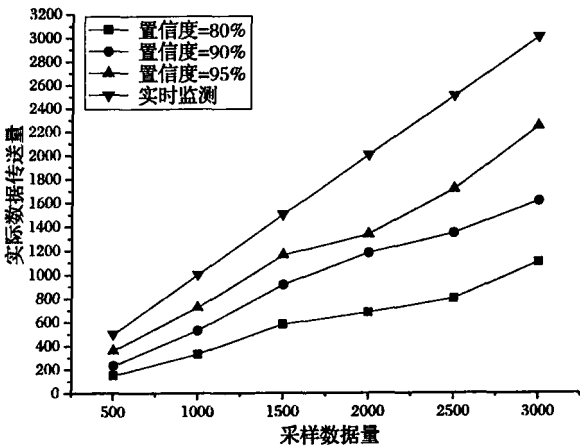


图 1 实际传送的数据量与全部采样数据的关系

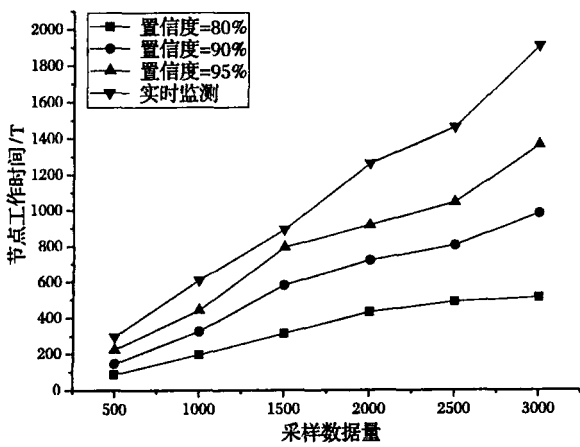


图 2 节点处于工作状态的时间

节点消耗能量的工作状态主要是通讯、采样和计算。通过基于模型的调度策略, 可以减少结点处于上述工作状态的时间。图 2 显示了节点在不同置信度要求下, 处于工作状态

的时间。为了进行统一的表示, 节点的工作时间以采样周期长度 T 为单位。从图 2 可以看出, 基于概率预测模型的调度策略有效地减少了节点的工作时间, 因为在适当的选择置信度后, 如果预测模型能够较好满足查询的需要, 那么节点可以减少通信和采样的时间。另一方面, 建模所需的计算与通信相比, 消耗的能量远小于后者。图 2 与图 1 在趋势和形状上相似, 这主要是因为这类应用中, 数据的无线传输时间占了很大的比例, 因此工作时间大致上与传输的数据量呈正比。

5 结语

本文提出了一种基于概率模型的无线传感器网络采样和通信动态调度算法。与已有工作不同的是它是构建于传感器网络节点上的分布式算法, 因此节点可以根据数据本身的特性动态调整自身采样与通信等操作的时间。模拟试验结果显示这种策略可以有效减少网络通信开销, 缩短节点处于耗能工作状态的时间。

参考文献:

[1] 李建中, 李金宝, 石胜飞. 传感器网络及其数据管理的概念、问题与进展 [J]. 软件学报, 2003, 14 (10): 1717 - 1726.

[2] TLAK S, ABU-GHAZALEH NB, HENZELMAN W. A taxonomy of wireless micro-sensor network models[J]. Mobile Computing and Communications Review, 2002, 1 (2): 1 - 8.

[3] AKYLDIZ LF, SU WL, SANKARASUBRAMANIAM Y, et al. A survey on sensor networks [J]. IEEE Communications Magazine, 2002, 40 (8): 102 - 114.

[4] MA C, MA M, YANG YY. Data-Centric Energy Efficient Scheduling for Densely Deployed Sensor Networks [A]. Proceedings of IEEE International Conference on Communications [C]. Paris, France. IEEE Computer Society, United States, 2004. 3652 - 3656.

[5] HOHLT B, DOHERTY L, BREWER E. Flexible Power Scheduling for Sensor Networks [A]. Proceedings of IPSN 2004 [C]. Berkeley, CA., United States, ACM, 2004. 205 - 214.

[6] LU HF, CHANG YC, HU HH, et al. Power-Efficient Scheduling Method in Sensor Networks [A]. IEEE International Conference on Systems, Man and Cybernetics [C]. The Hague, Netherlands, 2004. 4705 - 4710.

[7] GHOSH A, GNARGIS T. LORD: A Localized, Reactive and Distributed Protocol for Node Scheduling in Wireless Sensor Networks [A]. Proceedings of the conference on Design, Automation and Test in Europe [C]. IEEE Computer Society, 2005. 190 - 195.

[8] QU QR, PEDRAM M. Dynamic Power Management Based on Continuous - Time Markov Decision Processes [A]. Proc. Design Automation Conference [C]. IEEE Computer Society, New Orleans, LA, USA, 1999. 555 - 561.

[9] LAZAR D ISL. Capturing Sensor - Generated Time Series with Quality Guarantees [A]. Proceedings of IEEE International Conference on Data Engineering [C]. Bangalore: IEEE Computer Society, 2003. 429 - 440.

[10] DESHPANDE A, GUESTIN C, MADDEN SR, et al. Model-Driven Data Acquisition in Sensor Networks [A]. Proceedings of the 30th VLDB Conference [C]. Toronto: Morgan Kaufmann, 2004. 588 - 599.

[11] SUDHAKAR M. Pandit. Time Series and System Analysis with applications [M]. New York: John Wiley and Sons, 1983.

[12] Http://www.xbow.com/index.htm [EB/OL].