

基于压缩感知的传感器网络中概率负载均衡的数据路由协议

丁 旭^{1,2}, 黄 成¹, 吴晓蓓^{1†}, 徐志良¹

(1. 南京理工大学 自动化学院, 南京 210094; 2. 加州大学戴维斯分校 计算机系, 戴维斯 95616)

摘 要: 基于混合压缩感知(CS)理论, 提出一种负载有效的路由协议. 考虑分簇网络结构, 簇内节点传输原始数据到簇头, 簇头对数据进行压缩再通过最小生成树发送到 sink. 为防止簇头节点负载不均衡造成网络不能正常通信, 提出负载度的概念并设计基于 CS 的负载均衡策略; 然后, 研究概率负载均衡策略以均衡所有节点的负载流量; 最后, 提出分布式补偿算法构建分簇网络并实现数据汇聚功能. 仿真结果表明, 所提出方法在提高网络生存时间及能耗均衡方面均优于传统方法.

关键词: 无线传感器网络; 数据汇聚; 混合压缩感知; 负载度; 非均匀分簇

中图分类号: TP393.01

文献标志码: A

Probability load balance routing protocol based on compressive sensing in wireless sensor networks

DING Xu^{1,2}, HUANG Cheng¹, WU Xiao-bei^{1†}, XU Zhi-liang¹

(1. School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China; 2. School of Computer Science, University of California at Davis, Davis 95616, USA)

Abstract: A load-efficient data aggregation strategy in wireless sensor networks(WSNs) is proposed based on compressive sensing(CS). A cluster-based network is constructed, where all nodes send the raw data to their corresponding cluster heads(CHs) and the CHs send the compressed data to the sink through multi-hop path. To realize the load balance among all cluster heads in the networks, an uneven clustering strategy is proposed, and a CS-based load balance(CSLB) strategy is designed. Then a more efficient strategy of Probability-CSLB is proposed to balance the traffic load among all nodes in the network. Finally, a distributed compensation algorithm based on our model. Extensive experiments validate that our scheme can prolong the network lifetime and balance the energy consumption when compared with other schemes.

Keywords: wireless sensor networks; data aggregation; hybrid compressive sensing; load degree; uneven clustering

0 引 言

无线传感器网络(WSNs)是目前国际热点的研究领域^[1], 而设计能量有效的路由协议是能量受限的 WSNs 中最重要的操作之一, 并直接影响到网络生存周期.

已有许多路由协议^[1-3]基于数据聚合技术来降低传输包数以实现节能的目的. 但其存在许多缺点: 一些有损聚合策略(如 min/max/ave)只提取部分统计特征信息^[2], 而丢弃其他数据信息, 因此只适用于如火灾报警等特定应用场景; 此外, 基于信源编码^[3]的数据压缩策略需数据存在潜在的相关性结构^[4], 但由

于缺乏数据相关性的先验知识, 其并不完全适合实际应用.

新兴的压缩感知(CS)技术^[5]为 WSNs 数据传输开拓了新的方向. 与上述传统方法不同, CS 可实现无损压缩且无需数据相关, 只要信号在特定域中是稀疏的^[5], CS 即可实现对信号的无损压缩. 在实际中, 几乎所有的自然信号都可在特定域上进行稀疏表示^[6], 这使得 CS 能适用于 WSNs 数据收集的实际应用场景, 在能够恢复原始信息的基础上, 显著地减少数据传输量并延长网络生命周期.

在 CS 理论^[5]中, 设原始信号为 $\mathbf{X} = [x_1, \dots,$

收稿日期: 2017-03-10; 修回日期: 2017-07-17.

基金项目: 江苏高校优势学科建设工程项目; 国家留学基金委联合培养项目.

责任编辑: 俞立.

作者简介: 丁旭(1991—), 男, 博士生, 从事无线传感器网络的研究; 吴晓蓓(1958—), 女, 教授, 博士生导师, 从事无线传感器网络、智能控制等研究.

†通讯作者. E-mail: wuxb@njust.edu.cn

$x_N]^T$, 基向量 $\Psi = [\psi_{ij}]_{N \times N}$, 满足 $X = \Psi\alpha$, 其中 α 是 X 的稀疏系数, 当 α 仅有 k 个非零系数时, X 在基 Ψ 上是 k -稀疏的. 采用观测矩阵 $\Phi \in \mathbf{R}^{M \times N} (M \ll N)$ 将 X 压缩为 M 维信号 $y = \Phi X$. 最后, 当收到测量值 y 后, 通过求解 l_1 范数即可重构 X .

考虑到环境监测(气候、温度等)应用需要网络长时间工作, 为提高网络寿命, 文献[7]基于CS和随机投影设计了最小生成树投影方案(MSTP). 在WSNs数据收集应用中, 文献[4]研究了基于混合CS的数据汇集树路由协议以实现能耗最小化. 不同于直接考虑最小能耗, 文献[8]提出了基于混合CS的最小聚合树方案, 旨在研究数据传输数的最小化问题.

相比于上述路由树结构, 集簇方法在网络故障容差及负载均衡等方面具有较多优势^[9]. 针对周期性环境监测和数据收集, 文献[9]提出了基于混合CS的分簇方案, 通过研究最优簇数实现传输数的最小化, 然而利用均匀分簇结构会造成网络负载不均衡并导致“能量空洞”的问题^[10]. 针对上述问题, 文献[11]研究了基于CS的非均匀分簇策略, 其只考虑簇头间负载均衡, 在多轮传输后, 其他节点的不均衡性同样会造成网络性能的降低.

上述研究一定程度上仍存在局部区域能耗过多, 从而造成能量空洞问题并降低网络连通性^[10]. 为解决此类问题而重构路由结构(尤其是树结构)会消耗大量的能量, 因此, 在路由构造阶段考虑负载均衡可以有效避免能量空洞并保证网络连通, 并且无需在传输期间频繁重构路由, 降低了能耗, 从而保证WSNs进行长时间正常的数据传输.

基于这一思想, 本文设计了负载有效的路由协议来平衡网络节点负载. 混合CS方案相比传统CS方案可进一步降低数据传输数^[4], 因此本文设计基于混合CS的数据传输模型. 首先构建分簇路由协议的框架; 为实现负载均衡, 定义了“负载度”的概念并提出两个非均匀分簇策略; 然后基于CS的负载均衡策略(CSLB)和概率-CSLB策略(P-CSLB)设计分布式补偿算法来构建最终簇结构; 最后, 通过理论分析本文分簇算法和补偿算法的复杂度, 以及仿真实验验证了本文方法的优越性.

1 网络模型及问题描述

本文研究的对象假设如下:

1) N 个节点随机布撒在区域 S 中, 节点密度 $\rho = N/S$; 节点通信半径为 d_S , 各节点维护邻居表, 其到 sink 的跳数已知, sink 的处理存储等能力远大于普通节点;

2) 节点组织成簇, 各簇头知道簇内节点的投影向量(Φ 的子矩阵), 这在实际系统中可利用伪随机数生成器构造^[12].

本文协议包括分簇路由构建阶段和数据传输阶段. 首先依据混合CS传输的特性, 将网络分簇并生成传输路由; 网络构建完成后, 普通节点将原始数据经最短路径路由^[9,13]传给簇头, 当收到原始数据包后, 簇头基于CS压缩数据并通过最小生成树路由将压缩包传给 sink.

网络建模为 $G = \langle V, E \rangle$, $V = V_{CH} \cup V_{NN}$. 其中: V 为节点集, V_{CH} 为簇头节点集, V_{NN} 为普通节点集; E 为链路集, 当节点 $i, j \in V$ 的距离小于 d_S 时存在通信链路 $e_{ij} \in E$. 图1为本文路由协议模型图, 令 CH_i 表示簇头节点 $i (i \in V_{CH})$, x^{CH_i} 表示簇 CH_i 内所有节点的原始数据向量, Φ^{CH_i} 表示观测矩阵 Φ 在簇 CH_i 内对应的子矩阵.

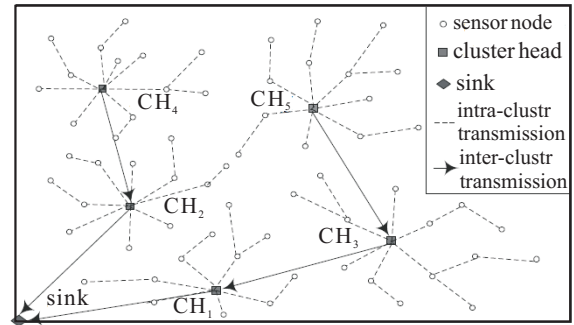


图1 基于混合CS的数据路由协议

考虑多轮次传输的应用场景^[11], 每轮传输中各节点发送原始数据包到簇头 CH_i , 当收到簇内数据 x^{CH_i} 后, CH_i 在 Φ^{CH_i} 上计算数据的投影 $\Phi^{CH_i} x^{CH_i}$; 然后, CH_i 将生成的 M 组压缩包传输到 sink. 如图1所示, 网络划分为5个簇 $CH_i (i = 1, 2, \dots, 5)$, 有 $X = [x^{CH_i}]_{5 \times 1}$, $\Phi = [\Phi^{CH_i}]_{5 \times 1}$, 最终, sink 收集到 M 维数据

$$y = \Phi X = \sum_{i=1}^5 \Phi^{CH_i} x^{CH_i}. \quad (1)$$

2 协议描述

本节首先介绍路由构造的具体过程, 即如何实现网络分簇以均衡节点负载; 其次, 在分簇网络下执行基于混合CS的数据传输.

2.1 临时簇的建立

定义指示函数

$$I(x) = \begin{cases} 1, & x \geq 1; \\ x, & \text{others;} \end{cases} \quad \text{dom}\{I(x)\} \in \mathbf{R}^+. \quad (2)$$

其中 $\text{dom}\{I(x)\}$ 为 $I(x)$ 的定义域.

初始时,所有节点参与簇头竞争;各节点*i*生成一个随机数 R_i ,如果 R_i 小于簇头选举指标 T_i ,则节点*i*当选为临时簇头TCH. T_i 定义如下:

$$T_i = P \cdot I\left(\frac{E_i}{E_{\text{threshold}}}\right), P = \frac{N_{\text{opt}}}{N}. \quad (3)$$

其中: E_i 为节点*i*的能量, $E_{\text{threshold}}$ 为指定能量阈值, N_{opt} 为采用文献[9]研究的最小化传输数的最优簇头数理论结果, $N_{\text{opt}} \ll N$.

如式(3)所示,各节点*i*的簇头选举指标 T_i 考虑了其能量 E_i 的作用.当 $E_i < E_{\text{threshold}}$ 时,有不等式 $I(E_i/E_{\text{threshold}}) = E_i/E_{\text{threshold}} < 1$,此时有 $T_i < P$,表明能量较低的节点当选为临时簇头的概率小于能量充足的节点.

临时簇头TCH当选后,广播其当选消息IM包,包中包括节点ID及能量、跳数*H*及TTL(Time-To-Live)等信息,各IM包的跳数初始为0.当节点收到任一IM包后,提取包中跳数信息*H*并与节点储存的TCH的跳数作对比:若*H*小于已储存的信息,则更新储存的跳数信息及此包的发送节点作为待选父节点,同时,节点更新此IM包中的*H*及TTL信息并继续向其邻节点广播此包;否则,节点直接丢弃此包.当IM包中TTL为0时,节点停止转发此包.

当上述操作完成后,节点根据储存的到达各TCH的跳数选择最近的TCH作为自己的簇头,同时向对应父节点回复ACK消息(包含其ID、位置及能量信息),最终沿着IM包传输路径的反向路径形成各节点到所属的TCH的路由树.此操作过后,各临时簇头TCH得到其真实簇尺寸ACS的值,定义如下.

定义1(真实簇尺寸ACS) 各簇内实际存在的节点总数称为各簇的真实簇尺寸.

此后,各TCH建立簇间传输路径,选择上游TCH作为转发节点.本文采用最小生成树^[11](MST)思想构建连接TCH到sink的骨干路由树^[9]:基于上述IM包广播过程,各TCH已知与它相近的TCH集合的位置信息以及到它们的跳数,则从上游TCH集合中选取到该TCH具有最小跳数的节点作为父TCH节点,最终形成一颗骨干路由树.

2.2 负载度及分簇策略

当任一TCH有更多的簇内节点及子TCHs时,相比于其他TCH,其将传输更多的数据包并消耗更多能量.因此,本节设计两种优化模型计算各簇期望簇尺寸RCS以均衡各簇负载量:簇头节点负载均衡(CSLB)策略和概率-CSLB策略(P-CSLB).

定义2(期望簇尺寸RCS) 为满足负载均衡的要求,各簇内期望存在的节点数称为各簇的期望簇

尺寸.

定义3(负载度) 在构造的MST树中,各簇头节点的负载度LD是其所有一跳子簇头节点数与其自身的和.

由第1节网络模型可知,设簇 CH_i 的 $\Phi^{\text{CH}_i} \in \mathbf{R}^{M \times n_i}$,簇内节点数为 n_i ,则其 $\mathbf{x}^{\text{CH}_i} \in \mathbf{R}^{n_i \times 1}$,故 CH_i 生成的压缩数据包向量为 $\Phi^{\text{CH}_i} \mathbf{x}^{\text{CH}_i} \in \mathbf{R}^{M \times 1}$,表明各簇头生成*M*个压缩包.在簇间通信中,各簇头的簇间传输数包括其自身压缩包数以及其转发的子簇头的压缩包数,对于负载度为 LD_i 的簇头 CH_i ,其自身压缩包数为*M*且其转发的下游簇头压缩包总数为 $(\text{LD}_i - 1)M$,则其传输的簇间压缩包总数为 $\text{LD}_i \cdot M$.不同的LD会造成不同传输数,为满足节点的负载均衡,一个策略是改变节点的簇间传输数.然而,考虑到网络稳定性的要求,频繁地改变簇间传输数(改变MST结构)来满足负载均衡是不可取的,因此本文通过调节各簇的簇内传输数均衡各簇负载,即通过不均匀分簇满足各簇负载均衡.

2.2.1 簇头节点负载均衡策略CSLB

基于CS的簇头负载均衡策略(CSLB)通过非均匀分簇保证每轮传输中簇头的负载均衡.设 $\text{TCH}_i (i = 1, 2, \dots, N_{\text{opt}})$ 的负载度为 LD_i ,其簇内传输数为 s_i ,则总传输数 $T_i = s_i + \text{LD}_i \cdot M$.为保证负载均衡,网络中各簇头需满足 $T_i = T_j$,其中*i*和*j*为任意两个簇头节点标号.

由于网络拓扑未知, s_i (或 s_j)难以测量,CSLB通过分配不同节点数到不同的TCHs且采用簇内节点数 n_i 近似 s_i ,建立如下优化问题:

$$\begin{aligned} & \min \Delta T. \\ & \text{s.t.} \begin{cases} \Delta T = \max_{i \neq j} |T_i - T_j|; \\ T_i = n_i + \text{LD}_i \cdot M; \\ T_j = n_j + \text{LD}_j \cdot M; \\ \sum_{a=1}^{N_{\text{opt}}} n_a = N, n_a \in \mathbf{Z}^+. \end{cases} \quad (4) \end{aligned}$$

此时得到的 n_i 即为各簇 TCH_i 的期望簇尺寸RCS.其中: n_i 为 CH_i 的期望节点数RCS, \mathbf{Z}^+ 表示正整数.第1个约束是寻找网络中各簇头间传输数的最大差值,第4个约束是所有簇内的节点总数为*N*、优化目标为最小化各簇头间总传输数的最大差值,使得各簇头的负载近似相同.

2.2.2 概率-簇头节点负载均衡策略P-CSLB

CSLB可保证在每轮数据传输中各簇头节点的负载均衡.由式(4)可知,LD较大的簇头被分配的节

点数相对较少, LD较小的簇头被分配的节点数较多. 这导致了另一个问题: 由于簇内节点轮流成为簇头, 这样较小簇内的节点成为簇头的概率将大于较大簇内的节点. 因此, 即使CSLB可以保证每轮数据传输中簇头的负载均衡, 但是较小簇内的节点由于更频繁地当选为簇头, 在多轮次传输后, 节点能耗相对更多.

设网络参数 $N = 20, N_{\text{opt}} = 3, M = 0.1N, LD_1 = 3, LD_2 = LD_3 = 1$, 总轮次 $R = 1000$. 基于CSLB, 各簇内节点数分别为4、8、8, 因此在每轮传输中, 所有簇头的 T_i 均为10. 然而, 假设簇内节点当选为簇头的概率服从均匀分布, 即每轮传输中簇 i 内节点 j 被选为簇头的概率为 $p_{ij} = 1/n_i$, 在总的 R 轮传输中, 节点 j 被选为簇头的次数为 Rp_{ij} , 则当其被选为簇头后总的传输数为 $Rp_{ij}T_i$. 基于上述分析, 各簇内节点在经过 R 轮传输后, 对应的总传输数分别为2500、1250、1250, 各簇节点传输数差异较大, 表明CSLB不能均衡多轮传输后的节点负载.

针对这一问题, 考虑单轮传输中簇头的负载均衡和节点成为簇头的概率之间的权衡, 提出概率-CSLB策略(P-CSLB)

$$\begin{aligned} & \min \Delta R. \\ & \text{s.t.} \begin{cases} \Delta R = \max_{i \neq j} \left| \frac{T_i}{n_i} - \frac{T_j}{n_j} \right|; \\ T_i = n_i + LD_i \cdot M; \\ T_j = n_j + LD_j \cdot M; \\ \sum_{a=1}^{N_{\text{opt}}} n_a = N, n_a \in Z^+. \end{cases} \end{aligned} \quad (5)$$

由于 $T_i = n_i + LD_i M$, P-CSLB可转化为P-CSLB₁, 即

$$\begin{aligned} & \min \Delta R_1. \\ & \text{s.t.} \begin{cases} \Delta R_1 = \max_{i \neq j} \left| \frac{LD_i}{n_i} - \frac{LD_j}{n_j} \right|; \\ \sum_{a=1}^{N_{\text{opt}}} n_a = N, n_a \in Z^+. \end{cases} \end{aligned} \quad (6)$$

P-CSLB₁ 分配给各簇 CH_i 的节点数 n_i 与其 LD_i 相关, LD_i 越大, 分配的节点数越多. 这是与CSLB相反的结论, 表明两者所要达到的均衡效果不同. 由于P-CSLB₁ 中只需LD的知识而无需每次计算 T_i , 其比P-CSLB更为简洁. 算法1是P-CSLB₁的伪代码, 其中 $\Delta LD_{i,j} = LD_i - LD_j, \Delta n_{i,j} = n_i - n_j$.

算法1 分簇策略P-CSLB₁.

输入: $N_{\text{opt}}, N, M, t_{\text{max}}, LD_i$;

输出: 各簇 CH_i 的 $n_i, i = 1, 2, \dots, N_{\text{opt}}$.

1) 初始化 $n_i = (N - N_{\text{opt}})/N_{\text{opt}}, i = 1, 2, \dots,$

N_{opt} ;

2) For $t = 1$ to t_{max}

3) 计算 $LD_i/n_i, i = 1, 2, \dots, N_{\text{opt}}$;

4) 选择 $\{i_{\text{max}}\} = \arg \max_i (LD_i/n_i), \{j_{\text{min}}\} = \arg \min_j (LD_j/n_j)$,

5) If $(\text{Num}\{i_{\text{max}}\} \geq 2) \parallel (\text{Num}\{j_{\text{min}}\} \geq 2)$,

6) 从 $\{i_{\text{max}}\}$ 或 $\{j_{\text{min}}\}$ 中随机选择一个 i_{max} 或 j_{min} ;

7) End If

8) 更新 $\Delta R_1 = LD_{i_{\text{max}}}/n_{i_{\text{max}}} - LD_{j_{\text{min}}}/n_{j_{\text{min}}}$;

9) If $\Delta R_1 \leq \Delta R_{\text{threshold}}$,

10) Break;

11) End If

12) If $\Delta LD_{i_{\text{max}}, j_{\text{min}}} \neq 0$,

13) $n_{j_{\text{min}}} = n_{j_{\text{min}}} - \Delta LD_{i_{\text{max}}, j_{\text{min}}} \cdot M/2$,

$n_{i_{\text{max}}} = n_{i_{\text{max}}} + \Delta LD_{i_{\text{max}}, j_{\text{min}}} \cdot M/2$;

14) Else

15) $n_{j_{\text{min}}} = n_{j_{\text{min}}} - \Delta n_{i_{\text{max}}, j_{\text{min}}}/2, n_{i_{\text{max}}} =$

$n_{i_{\text{max}}} + \Delta n_{i_{\text{max}}, j_{\text{min}}}/2$;

16) End If

17) End For

初始时, 设定各簇内节点数相同; 计算各簇头的 LD_i/n_i 并找出具有最大 LD_i/n_i (最小 LD_i/n_i) 的 $CH_{i_{\text{max}}}$ ($CH_{j_{\text{min}}}$); 若存在两个以上的簇头同时具有最大 (或最小) 的 LD_i/n_i , 则从中随机选取一个作为当前的 $CH_{i_{\text{max}}}$ (或 $CH_{j_{\text{min}}}$); 更新 ΔR_1 的值, 若其小于阈值 $\Delta R_{\text{threshold}}$, 则跳出 For 循环并返回结果, 否则执行分配策略; 为减少 ΔR_1 的值, 只需将簇 $CH_{j_{\text{min}}}$ 内的节点分配给 $CH_{i_{\text{max}}}$, 此时 $n_{j_{\text{min}}}$ 减少而 $n_{i_{\text{max}}}$ 将会增加, 这样 ΔR_1 便会降低; 根据不同的 $\Delta LD_{i_{\text{max}}, j_{\text{min}}}$ 条件, 给出不同的分配策略; 此后继续循环直到终止条件发生.

由于算法实现需要所有簇的LD知识, 各簇头将自身LD信息封装为LD包经由2.1节建立的路由树传往 sink, 此时最坏情况下各节点通信复杂度为 $O(N_{\text{opt}})$, 由 sink 执行算法1后再将得到的RCS封装成RCS包沿反向路由树传往各簇头.

2.3 最终簇的建立

TCH通过上述分簇策略得到RCS后即可构建最终簇结构. 结合已有分簇网络构建步骤^[11,14], 基于本文ACS和RCS, 各TCH并行执行补偿算法.

算法2 补偿算法.

输入: $N_{\text{opt}}, N, RCS_i, ACS_i, i = 1, 2, \dots, N_{\text{opt}}$;

输出: 最终簇的构建.

- 1) If $ACS_i > RCS_i$,
- 2) $\Delta n_i = ACS_i - RCS_i$;
- 3) For $j = 1$ to Nei_i
- 4) If $RCS_j > ACS_j$,
- 5) $CPS_i = CPS_i \cup \{TCH_j\}$;
- 6) $\Delta n_i(j) = RCS_j - ACS_j$;
- 7) End If
- 8) End For
- 9) $\Delta n_i' = \sum_{k=1}^{|CPS_i|} \Delta n_i(k)$;
- 10) For $j = 1$ to Nei_i ,
- 11) $\Delta \tilde{n}_i(j) = \Delta n_i \times \Delta n_i(j) / \Delta n_i'$
- 12) COMPENSATE_PROCESS($i, j, \Delta \tilde{n}_i(j)$);
- 13) End For
- 14) End If

当满足 $ACS_i > RCS_i$ 时, TCH_i 计算其可以补偿给其他 TCH 的节点数 Δn_i ; 如果 TCH_i 的任意邻居 TCH_j 的 ACS_j 小于 RCS_j , 则 TCH_i 保存 TCH_j 的 ID 信息并计算其所需要被补偿的节点数 $\Delta n_i(j)$; 上述操作后, TCH_i 可得其所有邻居簇所需的节点总数 $\Delta n_i'$; 此后, 根据 $\Delta n_i, \Delta n_i(j)$ 及 $\Delta n_i'$, TCH_i 计算出其实际可以补偿给邻居簇 TCH_j 的节点数 $\Delta \tilde{n}_i(j)$; 最后, 执行补偿算法, 根据计算的结果, TCH_i 将部分簇内节点补偿给需要的簇: 对于邻居簇 TCH_j , 计算距离 TCH_j 最近的 $\Delta \tilde{n}_i(j)$ 个节点, 将其补偿给 TCH_j ; TCH_i 向其簇内广播公告消息 BM, BM 消息包含可补偿节点的 ID 及对应的新簇头 TCH_j ; 簇内节点收到消息后, 判断自身是否为可补偿节点, 若是, 则从 TCH_i 内断开并加入新的簇 TCH_j 内 (即从储存信息中选择簇头为 TCH_j 的父节点, 这一储存信息在 2.1 节构建临时簇时得到); 最后, 最终簇即可建立.

分簇网络建立好后, 则进入数据传输阶段. 各 CH 根据簇内节点数分配时隙, 并保留额外的时隙进行数据融合和簇间路由; 时隙分配完成后, 各 CH 创建 TDMA 广播消息并采用随机退避机制广播此消息; 成员节点收到 TDMA 消息后, 根据时隙分配表设置自己的时隙, 簇内节点在对应时隙发送数据包到所属簇头; 簇头节点压缩收到的数据并经由 MST 树传输到 sink, 完成数据的最终传输工作.

在数据传输阶段, 若簇头死亡, 整个簇的数据包都将丢失, 故需设计周期性簇头轮转机制及时更换新簇头. 在设定周期内, 各 CH 检查其剩余能量, 若剩余能量低于阈值 $E_{\text{threshold}}$, 则利用式 (3) 重新选举 CH. 选举完成后新的 CH 广播更新消息, 簇内节点收到消

息后更新 CH, 再进行如上数据传输操作.

2.4 算法分析

定理 1 在最坏情况下, 算法 1 的计算复杂度为 $O(t_{\text{max}} N_{\text{opt}})$, 通信复杂度为 $O(N_{\text{opt}})$; 算法 2 的计算复杂度为 $O(N_{\text{opt}} n \log n)$, 各节点通信复杂度为 $O(\Delta)$, 其中 $n (n \leq N)$ 表示簇内节点数, $\Delta (\Delta \ll N)$ 表示簇内实际补偿节点数.

定理 2 在本文网络场景中, 簇头节点的负载度 LD 满足 $LD \in [1, LD_{\text{max}}]$, 其中 $LD_{\text{max}} = N_{\text{opt}}$.

引理 1 若各 CH_i 的 LD_i 均为 1, 则网络可实现均匀分簇, 且 CSLB 与 P-CSLB 的负载均衡效果相同.

此处, 定理 1、定理 2 和引理 1 的证明不再赘述.

设网络规模为 $L \times L$, sink 位于 (L_0, L_0) 处, 任一簇头的簇内节点数为 n , 负载度为 LD, 数据包大小为 l , 基于节点能耗模型^[15-16], 簇头接收簇内数据和其他簇头数据的总能耗为 $E_r = (n + (LD - 1)M)lE_{\text{elec}}$, 发送数据包的能耗为 $E_t = LD \cdot Ml(E_{\text{elec}} + \varepsilon d^\alpha)$, 其中 $\alpha = 2 \sim 4$, d 为簇头与下一跳的传输距离, E_{elec} 表示电路驱动能耗. 因此, 一般场景下簇头在一轮传输中的总能耗 $E = E_t + E_r$. 基于引理 1 可得以下特殊场景下簇头的能耗结论.

引理 2 若各簇头的 LD 均为 1, 则簇头总能耗 $E = E_t' + E_r'$, 其中 $E_r' = NlE_{\text{elec}}/N_{\text{opt}}$, $E_t' = Ml(E_{\text{elec}} + \varepsilon d^\alpha)$, d 满足

$$E[d^2] = \frac{2}{L} \left[\frac{(L - L_0)^3}{3} + \frac{(L_0)^3}{3} \right].$$

证明 LD 均为 1 时, 由引理 1 知网络均匀分簇且各簇头直接连接 sink, 有 $n = N/N_{\text{opt}}$, 故 $E_r = E_r', E_t = E_t'$ 成立; 参考文献 [17], 单跳距离 d 的平方的期望

$$E[d^2] = \int \int_0^L [(x - L_0)^2 + (y - L_0)^2] \rho(x, y) dx dy = \frac{1}{L^2} \int \int_0^L [(x - L_0)^2 + (y - L_0)^2] dx dy = \frac{2}{L} \left[\frac{(L - L_0)^3}{3} + \frac{(L_0)^3}{3} \right]. \quad \square$$

考虑到节点的通信半径为 d_s , 因此在上述能耗的理论分析中, 当簇头节点到下一跳的距离 $d = d_s$ 时, 节点的能耗最大.

3 仿真与结果分析

本节对比分析了相关协议的数据传输数, 网络生命周期和节点剩余能量等指标. 传感器节点随机分布在 $50 \text{ m} \times 100 \text{ m}$ 的区域内, 基站位于 $(0, 0)$ 处. 节点的初始能量为 2 J, 各包大小定义如下: 数据包长度为 500 bytes; ACK 长度为 10 bytes; IM 包长度为 18 bytes (包括保留位 2 bytes^[11]); LD 包、RCS 包和 BM 包长度均为 16 bytes. 节点能耗模型参考文献 [15-16], 为保证

公平性,节点能耗包括构造分簇网络的控制开销及多轮次数据传输的总能耗,在此基础上研究各协议的生命周期和剩余能量.令 $N = 100 \sim 500$, $d_S = 32$ m, $M = 0.1N$, N_{opt} 的计算、CS测量矩阵 Φ 和变换基 Ψ 均取自文献[9].为减轻随机性,取10次独立实验平均值和95%置信区间,比较以下协议的数据传输数.

Case 1: 本文基于CSLB分簇的路由协议;

Case 2: 概率分簇模型^[16],类似本文的临时簇头选举模型,但所有节点采取单跳传输模式;

Case 3: 概率分簇&CS,基于Case 2,簇头节点采用CS压缩数据后传输到sink;

Case 4: SPT^[13],最短路径树路由协议;

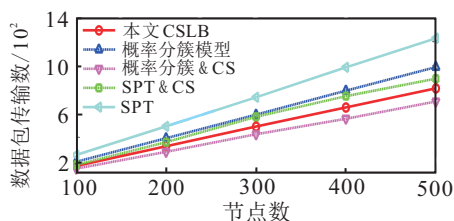
Case 5: SPT&CS,基于Case 4,当节点传包数大于 M 时,此节点利用CS压缩数据后再传;

Case 6: 本文基于P-CSLB分簇的路由协议;

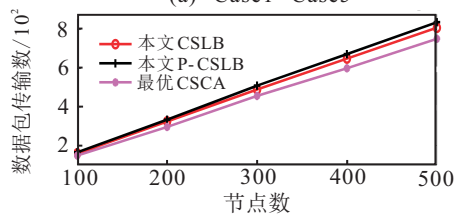
Case 7: 最优CSCA协议^[9],基于CS的均匀分簇协议,集中式算法旨在最小化数据包传输数,本文最优簇头数选自文献[9].

3.1 数据传输数对比分析

图2显示了不同协议传输数之间的对比.由图2(a)可知,本文CSLB的传输数相比概率分簇模型减少了约17%,即使后者采取了单跳传输方式;在结合CS后,概率分簇&CS模型的传输数略小于本文CSLB,这是因为其采用了单跳传输和数据压缩技术.由后面的分析可以看到,即使其传输数最小,但由于单跳传输耗能极大,其网络生命周期仍小于本文协议;此外,CSLB的传输数小于SPT协议,且随着节点数增加,两者间的传输数差距增大.SPT&CS的传输数略大于CSLB算法,验证了本文在采用最优分簇和CS技术后能有效降低传输数.



(a) Case1~Case5



(b) Case1, Case6, Case7

图2 不同协议传输对比

如图2(b)所示,考虑不同分簇方式的P-CSLB的传输数略大于CSLB协议,但相差极小(3%左右).最

优CSCA算法的传输数最小,因为其通过分析设计最优簇数来最小化网络中传输数.本文主要研究网络负载均衡问题,但传输数也仅仅增加了约7%左右,这远小于其他协议.

3.2 网络生命周期对比分析

为分析不同协议的生命周期,本文选取第1个节点死亡时的轮数作为网络生命周期,当节点剩余能量小于 $0.01E_0$ 时定义节点死亡.

图3分析了不同协议的生命周期对比.可以看出,随着节点数的增加,网络生命周期呈下降趋势.P-CSLB存活时间最长,CSLB次之,这是因为P-CSLB能更好地均衡节点负载.此外,尽管概率分簇&CS的传输数小于本文协议,但生命周期也小于本文协议,这是因为采用的单跳传输方式相比多跳传输耗能极大;最后,尽管采取分簇结构,概率分簇模型的生命周期最小,且随节点数增加而大幅降低.这表明即使本文协议在初步分簇后还需额外开销执行算法1和算法2的最终簇构建过程,但于控制包远小于数据包大小且只在初始构建网络时存在,同时本文协议中节点传输更少的数据包,节点总能耗仍会小于已有协议,而节点负载均衡的特性进一步延长了网络生命周期.

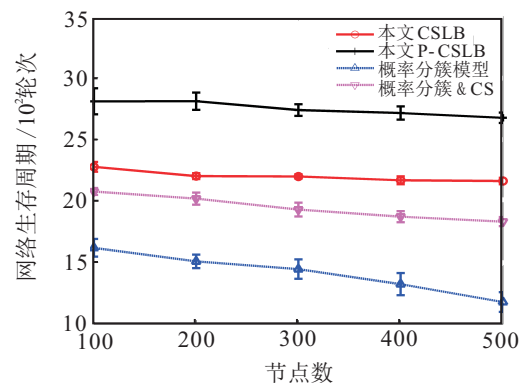


图3 不同协议的网络生命周期对比

3.3 剩余能量对比

本节主要研究CSLB及P-CSLB均衡节点负载的效果.图4为第1个节点死亡时,网络中各节点的剩余

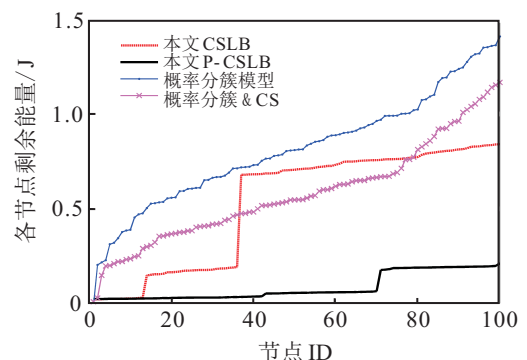


图4 节点剩余能量对比

能量(由小到大排序). 此时, 概率分簇模型及概率分簇&CS协议的剩余能量仍然很多, 可见能耗不均衡, 节点最大剩余能量分别有1.4J和1.2J, 这意味着节点能量不能充分利用.

分析节点剩余能量的均值和方差, CSLB的平均剩余能量比概率分簇模型和概率分簇&CS分别少了0.29J和0.054J, 但各节点剩余能量波动很大, 仍未保证整个网络的均衡. P-CSLB的平均剩余能量只有0.084J, 且各节点剩余能量波动很小, 仅为0.005, 说明节点能量得到充分利用, 整个网络负载均衡.

图5是节点按簇归类后剩余能量的显示图, 可以看出: CSLB保证了各簇内节点能耗相对均匀, 但各簇间能耗差别仍然很大; 而P-CSLB不仅保证簇内节点能耗均衡, 相邻簇的能耗也能保持相对均衡, 因此数据传输可以充分利用所有节点能量, 网络生命周期得到极大的延长(如图3).

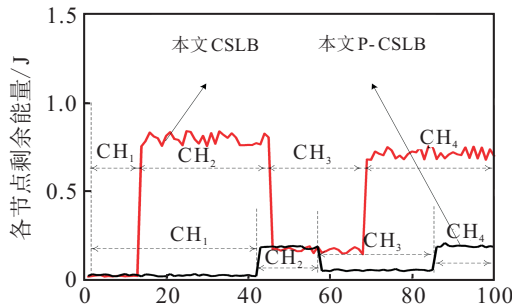


图5 CSLB及P-CSLB剩余能量对比(按簇划分)

4 结论

为充分利用网络各节点能量, 本文提出了基于非均匀分簇的混合CS路由协议. 簇内节点通过最短路径向簇头传输原始数据, 簇头压缩数据并经由MST传输到sink. 针对不同的负载均衡要求, 提出了CSLB和P-CSLB两种分簇策略并构建相应的优化模型. 基于模型, 本文通过补偿算法构建了非均匀的分簇结构. 仿真结果表明CSLB和P-CSLB仅比最优结果增加了极少的传输数, 相比于已有算法, 提高了网络的生命周期, 且P-CSLB可以充分均衡利用各节点能量, 进一步提高网络性能.

参考文献(References)

[1] Ding X, Sun X, Huang C, et al. Cluster-level based link redundancy with network coding in duty cycled relay wireless sensor networks[J]. *Computer Networks*, 2016, 99(4): 15-36.

[2] Cheng S Y, Li J Z, Ren Q Q, et al. Bernoulli sampling based (ϵ, δ) -approximate aggregation in large-scale sensor networks[C]. *Proc of IEEE Int Conf on Computer Communications*. San Diego: IEEE Press, 2010: 1-9.

[3] Slepian D, Wolf J. Noiseless coding of correlated information sources[J]. *IEEE Trans on Information*

Theory, 1973, 19(4): 471-480.

- [4] Liu X, Jun L, Athanasios V. Compressed data aggregation for energy efficient wireless sensor networks[C]. *Proc of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. Utah: IEEE Press, 2011: 46-54.
- [5] Donoho D L. Compressed sensing[J]. *IEEE Trans on Information Theory*, 2006, 52(4): 1289-1306.
- [6] Tropp J, Gilbert A C. Signal recovery from partial information via orthogonal matching pursuit[J]. *IEEE Trans on Information Theory*, 2007, 53(12): 4655-4666.
- [7] Ebrahimi D, Assi C. Compressive data gathering using random projection for energy efficient wireless sensor networks[J]. *Ad Hoc Networks*, 2014, 16: 105-119.
- [8] Xie R T, Jia X H. Minimum transmission data gathering trees for compressive sensing in wireless sensor networks[C]. *Proc of IEEE Global Communications Conference*. Houston: IEEE Press, 2011: 1-5.
- [9] Xie R T, Jia X H. Transmission-efficient clustering method for wireless sensor networks using compressive sensing[J]. *IEEE Trans on Parallel and Distributed Systems*, 2014, 25(3): 806-815.
- [10] Salarian H, Chin K W, Naghdy F. An energy-efficient mobile-sink path selection strategy for wireless sensor networks[J]. *IEEE Trans on Vehicular Technology*, 2014, 63(5): 2407-2419.
- [11] Ding X, Wu X B, Wang H Y, et al. Compressed data collection for clustering-based sensor networks[C]. *Proc of the 34th Chinese Control Conf*. Hangzhou: IEEE Press, 2015: 7703-7708.
- [12] Haupt J, Bajwa W U, Rabbat M, et al. Compressed sensing for networked data[J]. *IEEE Signal Processing Magazine*, 2008, 25(2): 92-101.
- [13] Augustine J, Han Q, Loden P, et al. Tight analysis of shortest path convergecast in wireless sensor networks[J]. *Int J of Foundations of Computer Science*, 2013, 24(1): 31-50.
- [14] Zhao X, Zhou Z, Jiang T, et al. A multi-hop routing protocol based on grid optimization for wireless sensor networks[C]. *Proc of IEEE GLOBECOM Workshops*. Anaheim: IEEE Press, 2013: 1405-1409.
- [15] 侯贵升, 吴晓蓓, 黄成, 等. 混合型异构传感器网络中移动数据收集器的路径优化[J]. *控制与决策*, 2016, 31(7): 1285-1290.
- (Hou G S, Wu X B, Huang C, et al. Optimizing path selection for mobile data collector in hybrid heterogeneous sensor networks[J]. *Control and Decision*, 2016, 31(7): 1285-1290.)
- [16] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks[J]. *IEEE Trans on Wireless Communications*, 2002, 1(4): 660-670.
- [17] Nguyen M T, Teague K A. Compressive sensing based random walk routing in wireless sensor networks[J]. *Ad Hoc Networks*, 2017, 54: 99-110.

(责任编辑: 孙艺红)