论 文

无线传感器网络中一种基于分簇的节点调度算法

王雷^{①②*}, WEI RuiZhong^②, 田子红^{②③}

① 湖南大学软件学院, 长沙 410082

② Department of Computer Science, Lakehead University, Thunder Bay P7B 5E1, Canada

③ 河北师范大学数学所, 石家庄 050016

* 通信作者. E-mail: wanglei@hnu.cn

收稿日期: 2009-03-23; 接受日期: 2010-08-10

湖南大学中央高校基本科研业务费资助项目

摘要 本文研究 m 覆盖与连通的无线传感器网络中的节点调度问题, 提出了一种两跳簇的概念, 并在此基础上提出了一种新的节点调度算法. 该算法可在不知道节点位置信息的情况下, 将无线传感器网络中的所有节点较为均匀地指派到 k ($k \leq m$) 个不同的分组 $\{0, 1, \dots, k-1\}$ 之中, 且可同时保证每个分组中的节点具有全局连通性与较高的网络覆盖率. 理论分析与实验结果表明, 该算法具有比传统随机调度方法更好的节点调度性能, 可更加有效地延长无线传感器网络的生命周期.

关键词 节点调度 网络生命周期 无线传感器网络 两跳簇

1 引言

传感器节点由于体积微小, 通常只能携带能量十分有限的电池, 且很多时候在部署之后无法进行电池的充电或更换. 因此, 如何节能以延长整个网络的生命周期, 是目前无线传感器网络研究领域中的一大挑战. 由于在无线传感器网络中节点一般具有稠密分布的特性, 为此, 节点调度方法的研究在无线传感器网络节能技术的研究中占有重要地位^[1-4].

目前已有的节点调度方法大致可以分为基于轮次 (round-based) 的与基于分组 (group-based) 的两类. 其中在基于轮次的方法之中, 节点需要在每轮 (round) 的初始化时执行调度算法, 从而依据某种竞争方法使得一些节点在本轮中成为活跃 (active) 节点, 而其他节点则进入睡眠 (sleep) 状态^[5]. 显然, 该类方法需要每个节点在其生命周期之内多次执行调度算法. 而在基于分组的方法之中, 每个节点只需在部署之后执行一次调度算法, 以将所有节点分配到某个分组 (groups) 之中, 从而使得在接下来的时隙 (time slots) 之中, 各个分组可依据自己给定的时隙轮流开始工作^[6].

在节点调度算法的设计之中, 主要有以下两个问题需要考虑: 一是各个分组中节点的网络覆盖率, 二是各个分组中节点的全局连通性. 而在无线传感器网络中, 由于传感器节点的感知半径 (sensing range) 通常是与其通信半径 (transmission range) 完全独立的, 这样就导致了非常难于将上述两个问题同时解决^[6], 为此, 许多研究人员将注意力集中到了对单纯基于感知半径 (即: 侧重分组的网络覆盖率保持) 的调度算法研究^[7-9], 或对单纯基于网络连通性 (即: 侧重分组的全局连通性保持) 的调度

引用格式: 王雷, Wei R Z, 田子红. 无线传感器网络中一种基于分簇的节点调度算法. 中国科学: 信息科学, 2011, 41: 1013–1023

算法的研究^[5,10,11]之中,而很少有人关注同时保持各个分组中节点的网络覆盖率与全局连通性的联合调度算法的研究.特别地,由于在无线传感器网络中获得并维护每个节点的位置信息的开销非常巨大,因此,这也使得要在无法知道每个节点的位置信息的情况下去研究上述联合调度算法变得更为困难^[6,12,13].

假设在传感区域(sensory field)内的任意一个位置点(point)都至少处于 m 个不同的传感器节点的监控范围之内(即在这些节点的感知半径之内),则称该传感区域为一个 m -覆盖的区域.由于在无线传感器网络中节点一般具有稠密分布的特性,因此不失一般性,显然可假定给定的传感区域为一个 m -覆盖的区域^[14,15].基于该假定,本文将主要解决以下挑战性问题,即:假定给定的无线传感器网络为 m -覆盖且全局连通的(即:整个无线传感器网络中的所有节点构成一个连通图),在不知道节点位置信息的情况下,如何将网络中的所有传感器节点尽可能均匀地指派到 $k(k \leq m)$ 个不同的分组 $\{0, 1, \dots, k-1\}$ 之中,使得每个分组中的节点在具有全局连通性的同时还可以保持尽可能高的网络覆盖率.

为了解决上述问题,本文首先针对 m -覆盖与连通的无线传感器网络定义了一个称为两跳簇(two-hops-cluster)的新结构,然后在此基础上提出了一种新的分布式分组标号(group ID)指派算法,最后,为了保证每个分组中节点的全局连通性,提出了一种分布式的分组连通性维护算法.理论分析与实验结果表明,新算法具有良好的节点调度性能.

2 随机节点调度算法

为了解决在上一节中所提到的联合节点调度问题,Liu^[6]等人提出了一种可同时保持分组网络覆盖率与全局连通性的节点调度算法RSGC(randomized scheduling scheme with guaranteed connectivity),该算法的前提条件如下:(1)无法提供无线传感器网络中的节点位置信息;(2)传感器节点的感知半径与通信半径无关;(3)无法提供额外的无线传感器网络路由协议;(4)所有的计算都是分布式进行的;(5)无线传感器网络中的时钟无法精确同步.

基于上述假设,RSGC的实现步骤具体可描述如下:

步骤 1 子集的随机选择过程.在初始化过程中,每个传感器节点首先随机生成一个0到 $k-1$ 之间的随机数 i ,然后将自己指派到第 i 个分组/子集(subset).

步骤 2 最小跳数计数的传播过程.该过程由Sink节点发起,通过邻节点之间的信息交换,从而使每个节点可最终得到自己到Sink节点的最小跳数计数(the minimum hop count to the sink).

步骤 3 邻节点之间的信息交换过程.在该过程中,邻节点之间通过信息交换以获得自己的所有上游节点(upstream node)标号及其所属子网标号等信息.其中,某个节点的上游节点是指在该节点的邻节点中到Sink节点的跳数要比该节点到Sink节点的跳数小1的那些节点.反之,在该节点的邻节点中到Sink节点的跳数要比该节点到Sink节点的跳数大1的那些节点,则称之为该节点的下游节点(downstream node).

步骤 4 分组连通性保持的实现过程.在该过程中,为了保障各个分组中节点的全局连通性,每个节点将首先基于一种称为extra-on的规则以及由步骤3得到的相关信息来决定自己的额外工作间隙,并据此更新自己的工作时间表.然后再将更新后的工作时间表广播给自己的所有邻节点.其中,extra-on规则定义如下:若节点 A 有一个下游节点 B ,且 B 对应的工作间隙为 i ,若 B 的所有上游节点的工作间隙均不为 i ,则规定节点 A 要在间隙 i 保持活跃状态.

步骤 5 分组轮流工作的过程. 在该过程中, 每个分组依在规定的工作时隙轮流保持活跃状态, 而其余节点则进入休眠状态, 从而达到延长网络生命周期的目的.

本文的主要目的就是在上述文献 [6] 中所给出的 RSGC 算法相同的前提条件下, 提出一个具有更好性能的基于分组的节点调度算法, 以进一步延长无线传感器网络的生命周期.

3 网络模型与相关定义

本文主要考虑部署在二维平面区域的静态无线传感器网络, 并假设传感器节点是随机独立地部署在整个传感区域之中, 传感器节点的感知半径用 r_s 表示, 而传感器节点的通信半径则用 r_c 表示, 显然, 可假定 r_c 大于等于 r_s [16].

另外, 由于在大规模的无线传感器网络之中难于实现节点之间的精确的时间同步, 为此, 与文献 [6] 类似, 本文亦不假定无线传感器网络之中存在精确的时间同步机制.

最后, 假定在节点部署之前, 所有节点均已被按照 n - 维超立方体 (hyper-cube) 的编码方式进行了编码 [17], 且所有非 Sink 节点的编码范围为从 $\underbrace{00 \cdots 001}_n$ 到 $\underbrace{11 \cdots 111}_n$, 而 Sink 节点的编码为 $\underbrace{00 \cdots 000}_n$. 由此一来, 在节点部署之后, 每个节点将具有一个预先从 $\{\underbrace{00 \cdots 001}_n, \dots, \underbrace{11 \cdots 111}_n\}$ 之中随机选取得到的唯一节点标号 (node ID), 而 Sink 节点的 ID 即为 $\underbrace{00 \cdots 000}_n$.

接下来, 再给出一些本文中后面需要用到的一些定义如下:

定义 1 对任意两个传感器节点 p_1 和 p_2 , 若两者之间的距离小于等于它们的通信半径 r_c , 则称这两个节点互为邻节点 (neighboring nodes).

定义 2 对一个 m - 覆盖与连通的无线传感器网络中的任意节点 p_1 , 假定 $p_2, \dots, p_t$ 为其所有的邻节点, 且对 $\{p_1, p_2, \dots, p_t\}$ 中的任意节点 p_i , 假定其所有的邻节点为 $p_{i1}, p_{i2}, \dots, p_{it_i}$. 假定在 $\bigcup_{i=1}^t \{p_{i1}, p_{i2}, \dots, p_{it_i}\}$ 之中所有不同的节点分别为 $\{p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_s\}$, 其中, $s \geq t$, 则称节点集 $\{p_1, p_2, \dots, p_s\}$ 为节点 p_1 的两跳簇 (two-hops-cluster).

定义 3 对任意节点 p_1 , 假定其两跳簇为 C_1 , 若一个属于 C_1 的子簇 C 中包含有 t 个元素, 则称该子簇 C 为 p_1 的一个 t - 子簇.

定义 4 对任意节点 p_i , 假定 p_j 为其一个邻节点, 且 p_i, p_j 到 Sink 的最小跳数计数分别为 H_i 和 H_j . 若 $H_i = H_j + 1$, 则称 p_j 为 p_i 的一个上游节点, 而 p_i 为 p_j 的一个下游节点. 若 $H_i = H_j$, 则称节点 p_j 为 p_i 的一个兄弟节点 (brother node), 同时, 若还有 $p_j > p_i$, 则称 p_j 为 p_i 的兄节点 (older brother node), 而 p_j 为 p_i 的弟节点 (younger brother node).

4 基于两跳簇的节点调度算法

4.1 两跳簇内的分组标号指派算法

由假设 $r_c \geq r_s$ 易知下述定理 1 成立:

定理 1 (两跳簇) 假定传感区域中的给定位置点 p 可被 t 个节点 $p_1, p_2, \dots, p_t$ 所覆盖, 则对 $\{p_1, p_2, \dots, p_t\}$ 中的任意节点 p_i , 节点集 $\{p_1, p_2, \dots, p_t\}$ 必属于 p_i 的两跳簇之内.

基于上述定理 1, 针对一个 m -覆盖与连通的无线传感器网络, 我们可以为一个两跳簇内的节点给出一个分布式的分组标号 GIDs(group IDs) 指派算法如下:

算法 1 GANC: GID assignment for nodes in a two-hops-cluster.

假定无线传感器网络为 m -覆盖与连通的, 对任意一个由 $t(t \geq m)$ 个传感器节点 $p_1, p_2, \dots, p_t$ 构成的两跳簇, 假定各节点的节点 ID 分别为 $P_1, P_2, \dots, P_t$, 则各个节点可依据以下方法被均匀地指派到 $k(k \leq m)$ 个不同的分组 $\{0, 1, \dots, k-1\}$ 之中.

步骤 1 不失一般性, 假定 $P_1 = \min\{P_1, P_2, \dots, P_t\}$, 则 p_1 宣布自己成为两跳簇 $\{p_1, p_2, \dots, p_t\}$ 的簇头 (cluster head). p_1 将首先收集其两跳簇 $\{p_1, p_2, \dots, p_t\}$ 内的所有簇成员 (cluster members) 的分组标号 (GIDs). 若所有簇成员均已被分配有 GIDs, 则算法终止. 否则, 跳转到步骤 2.

步骤 2 假定 $\{p_1, p_2, \dots, p_t\}$ 中的前 $a(a < t)$ 个节点 $\{p_1, p_2, \dots, p_a\}$ 已被分别分配有分组标号 $\{g'_1, g'_2, \dots, g'_a\}$, 且 $\{g'_1, g'_2, \dots, g'_a\}$ 中所有不同的 GIDs 为 $\{g_1, g_2, \dots, g_b\}$, 其中, $b \leq a$.

情形 1 若 $b = k$, 则对任意节点 $p_j \in \{p_{a+1}, p_{a+2}, \dots, p_t\}$, p_1 随机选择 $j \in \{0, 1, \dots, k-1\}$, 并将分组标号 j 指派给节点 p_j .

情形 2 若 $b < k$, 则令 $U = \{0, 1, \dots, k-1\} \setminus \{g_1, g_2, \dots, g_b\} = \{u_0, u_1, \dots, u_{k-b-1}\}$.

情形 A 若 $t - a \geq k - b$, 则 p_1 从 $\{0, 1, \dots, k-1\}$ 之中随机选择 $t - a - k + b$ 个不同的分组标号 $V = \{v_0, v_1, \dots, v_{t-a-k+b-1}\}$, 并将 $U \cup V$ 随机指派给节点 $\{p_{a+1}, p_{a+2}, \dots, p_t\}$.

情形 B 若 $t - a < k - b$, 则 p_1 从 U 中随机选择 $t - a$ 个不同的分组标号 $\{v_0, v_1, \dots, v_{t-a-1}\}$, 并将这些分组标号随机指派给节点 $\{p_{a+1}, p_{a+2}, \dots, p_t\}$.

4.2 m -覆盖的无线传感器网络中的分组标号指派算法

基于上述算法 1, 本节将进一步给出在整个无线传感器网络中的分组标号指派算法如下:

算法 2 DGAS: distributed GIDs assignment scheme.

假设给定的 m -覆盖与连通的无线传感器网络中包含 n 个不同的传感器节点, 且每个节点维护有一个信息表 IT(information table), 其中, IT 中包含的信息有 IDNN(邻节点的节点标号, ID of its neighboring nodes), NNLN(所有邻节点的邻节点表, neighboring nodes list of its neighbors), 以及 $IDNN \cap NNLN$. 则这 n 个节点可依据以下步骤被分布式地指派到 $k(k \leq m)$ 个不同的分组 $\{0, 1, \dots, k-1\}$ 之中.

步骤 1 对任意节点 p_1 , 令 $\{p_2, p_3, \dots, p_s\}$ 为其所有的活跃邻节点, $\{P_1, P_2, \dots, P_s\}, \{N_1, N_2, \dots, N_s\}$ 分别表示这些活跃邻节点的节点标号 (node IDs) 与 NNLN. 则依据定义 2, 节点 p_1 可基于 $\{P_1, P_2, \dots, P_s\}$ 与 $\{N_1, N_2, \dots, N_s\}$ 计算出其两跳簇.

步骤 2 节点 p_1 计算出其两跳簇中的所有 t -子簇, 其中, $t \geq m$, 且每个 t -子簇均应包含有节点 p_1 .

步骤 3 对节点 p_1 计算出来的任意 t -子簇, p_1 依据算法 1 为该 t -子簇中的所有节点指派分组标号.

由上述算法 2 的步骤描述易证以下定理 2 成立.

定理 2 假定给定的无线传感器网络为 m -覆盖与连通的, 则在执行完算法 2 之后, 网络中的每个节点均将被指派有一个 $\{0, 1, \dots, k-1\}$ 之中的单一分组标号.

4.3 分布式分组连通维护算法

在前述章节中, 已经给出了如何将一个 m -覆盖与连通的无线传感器网络中的所有节点均匀地指派到 k ($k \leq m$) 个不同分组之中的有关算法, 接下来, 将进一步考虑如何确保各个分组之中节点的全局连通性问题. 基于第 3 节中给出的定义 4, 本小节将给出一个分布式的分组连通维护算法如下:

算法 3 CMEG: connectivity maintenance for each group.

假定整个无线传感器网络是连通的, 且每个节点维护有一个包含有其所有上游节点与兄弟节点标号的列表. 在完成了分组标号指派步骤之后, 每个节点将会依据以下步骤来对其分组标号列表 (GID list) 进行维护与更新, 以保障各个分组中节点的全局连通性. 其中, 节点的初始分组标号列表中显然只包含有一个分组标号, 即: 依据算法 2 得到的那个单一分组标号.

步骤 1 对任意节点 p_i , 令 L_i 表示其分组标号列表. 对任意 $g \in L_i$.

步骤 1.1 若不存在 p_i 的上游节点或兄弟节点的分组标号列表中包含有 g , 则 p_i 将首先选择一个自己的上游节点, 不妨设为 p_j , 其中, p_j 具有最短的分组标号列表长度 (当然, 这里也可同时综合考虑节点的剩余能量等因素), 然后再发送一个 AGAC (appended GID application for connectivity) 包给 p_j .

步骤 1.2 若 p_i 没有分组标号列表中包含 g 的上游节点, 但却有分组标号列表中包含 g 的兄弟节点, 则 p_i 将首先选择一个自己的兄弟节点, 不妨设为 p_j , 其中, p_j 具有最短的分组标号列表长度 (当然, 这里也可同时综合考虑节点的剩余能量等因素), 然后再发送一个 AGAC 包给 p_j .

步骤 2 在完成了分组标号指派步骤之后, 每个节点维护一个回退时间定时器 (back-off timer).

步骤 2.1 若节点 p_j 在定时器超时之前收到了一个来自自己的下游节点或弟节点 p_i 的 AGAC 包, 则 p_j 将根据 AGAC 包中的内容更新自己的分组标号列表.

步骤 2.2 若节点 p_j 在定时器超时之前收到了一个来自自己的兄节点 p_i 的 AGAC 包, 且 p_j 发现在自己的所有上游节点的分组标号列表之中均不包含 g , 则 p_j 将首先选择自己的一个上游节点, 不妨设为 p_x , 其中, p_x 具有最短的分组标号列表长度 (当然, 这里也可同时综合考虑节点的剩余能量等因素), 然后再发送一个 AGAC 包给 p_x .

步骤 2.3 当定时器超时之后, 节点 p_j 将广播更新之后的分组标号列表给自己的所有邻节点.

依据上述算法过程的描述, 易证算法 3 具有以下有用的性质:

定理 3 假定整个无线传感器网络是连通的, 则在执行算法 3 之后, 每个节点将维护有一条到 Sink 节点的最短跳数路径, 且该路径上的所有节点均具有一组相同的分组标号.

4.4 连通性与覆盖率保持的节点调度算法

在前面几节的基础之上, 本小节将最终给出本文提出的基于分簇的可同时保持分组全局连通性与高网络覆盖率的分布式节点调度算法如下:

算法 4 THCNS: two-hops-cluster based node scheduling.

假定整个无线传感器网络是 m -覆盖与连通的, 则整个网络中的所有节点可依据以下步骤被均匀地调度到 k 个不同的分组 $\{0, 1, \dots, k-1\}$ 之中, 其中, $k \leq m$.

阶段 (1) 分组标号指派阶段.

步骤 1 每个节点 p_i 维护有一个信息表 IT_i , 其中 IT_i 中包含以下信息: IDNN, NNLN, IUN (上游节点的节点标号, identification of upstream node), TSHS (到 Sink 的最小跳数计数, the shortest hops to the sink), 以及参数 k . 在初始化时, IT_i 为空.

Sink 节点广播包含有参数 k 的 Hello 消息包给它的邻节点. 每个收到该 Hello 消息包的 Sink 的邻节点将记录 k 并设置自己的信息表中的 TSHS 字段的值为 1.

步骤 2 每个具有非空 TSHS 值的节点广播包含有自己的节点标号, 参数 k , 以及 TSHS 值的 Hello 消息包给自己的所有邻节点. 对任意节点 p_i , 若其从节点 p_j 处收到一个包含有 TSHS $_j$ 与参数 k 的 Hello 消息包, 则 p_i 将首先比较 TSHS $_j$ 与自己所维护的消息表 IT $_i$ 中记录的 TSHS $_i$ 的大小, 若 TSHS $_i$ 的值为空, 则 p_i 将记录 p_j 的节点标号, TSHS $_j+1$, 以及参数 k 到自己的消息表的相应字段之中. 若 TSHS $_i$ 的值非空, 则 p_i 将检查是否有 TSHS $_i = \text{TSHS}_j + 1$. 若上述等式成立, 则 p_j 的节点标号将被记录到消息表中对应的 IUN 字段之中 (这里假定先到达 p_i 的 Hello 消息包中的 TSHS 值不会大于后到达的 Hello 消息包中的 TSHS 值), 另外, 在收到来自邻节点的 Hello 消息包之后, p_i 也会相应更新其消息表中的 IDNN 字段.

步骤 3 每个节点生成并广播一个包含有自己的节点标号与 IDNN 的消息包给自己的所有邻节点, 如此一来, 在经过了邻节点之间的信息交换之后, 每个节点将可获知自己的活跃邻节点信息, 并由此可填充自己的消息表中的 NNLN 字段.

步骤 4 执行算法 2 进行分组标号的指派.

阶段 (2) 分组全局连通性维护阶段.

步骤 5 执行算法 3 进行分组连通性的维护.

阶段 (3) 分组轮流工作阶段.

步骤 6 在该阶段, 各组中的所有节点依据其指定的时隙轮流工作. 在时隙 t , 若 $t \equiv g \pmod k$, 则具有分组标号为 g 的所有节点保持活跃, 而其他节点则进入休眠状态.

5 算法性能分析与实验评价

5.1 算法性能分析

引理 1 假设随机选用 k 种不同颜色填充 $t(k \leq t)$ 个不同的点, 则这些点中恰有 $j(j \leq k)$ 种不同颜色的概率为

$$\frac{1}{k^t} \sum_{i=0}^{j-1} (-1)^i \binom{k}{j} \binom{j}{i} (j-i)^t.$$

证明 显然, 总共有 k^t 种不同方式去用 k 种不同的颜色来填充 t 个不同的点. 另外, 易知从一个具有 t 个元素的集合到一个具有 j 个元素的集合之间的映射函数 (onto functions) 的个数为

$$\sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j-i)^t.$$

此外, 还可易知总共有 $\binom{k}{j}$ 种不同方式来从 k 中颜色之中选取 j 种颜色. 由此可知引理 1 的结论成立.

定义

$$P(t, k, j) = \frac{1}{k^t} \sum_{i=0}^{j-1} (-1)^i \binom{k}{j} \binom{j}{i} (j-i)^t. \quad (1)$$

则易证下述引理 2 成立.

引理 2

$$\sum_{j=1}^{\min\{k,t\}} P(t,k,j) = 1.$$

基于下述第二类 Stirling 数 (the second kind Stirling number)

$$S(n,j) = \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j-i)^n,$$

可将上述公式 (1) 改写为

$$P(t,k,j) = \frac{j!}{k^t} \binom{k}{j} S(t,j).$$

定理 4 假定无线传感器网络是 m -覆盖与连通的, 且已经使用了算法 GANC 将网络中的所有节点指派到了 $k(k \leq m)$ 个不同的分组之中. 若传感区域中的一个给定位置点 p 可被 $t(t \geq m)$ 个不同的传感器节点覆盖, 则在这 t 个不同的传感器节点之中至少存在 k 个不同的传感器节点具有不同的分组标号的概率为

$$P_{\text{THCNS}} = \begin{cases} 1, & \text{当 } t_2 \geq k; \\ \frac{1}{k^{t_1}} \sum_{j=(k-t_2)}^{\min\{k,t_1\}} \sum_{i=0}^{j-1} (-1)^i \binom{k}{j} \binom{j}{i} (j-i)^{t_1}, & \text{当 } t_2 < k. \end{cases}$$

其中, $t_1 + t_2 = t$, 且 t_1 为在由这 t 个不同的传感器节点所构成的 t -子簇中执行算法 GANC 之前就已经被指派了分组标号的传感器节点的数目.

证明 由于位置点 p 被 t 个不同的传感器节点覆盖, 因此由定理 1 可知, 这 t 个节点形成了一个 t -子簇. 当在该 t -子簇中执行算法 GANC(算法 1) 时, 簇头将为簇内的每个节点指派分组标号. 令 t_1 表示在该 t -子簇中执行算法 1 之前就已经被指派有分组标号的传感器节点数目.

若 $t_2 \geq k$, 则依据算法 1, 簇头将从这 t_2 个尚未被预先指派有分组标号的传感器节点之中随机选择 k 个不同的传感器节点, 并将分组标号 $\{0, 1, \dots, k-1\}$ 随机指派给这 k 个传感器节点. 因此, 在该情形下, 在该 t -子簇中存在至少 k 个传感器节点具有不同的分组标号的概率显然为 1.

若 $t_2 < k$, 由引理 2 可知, 在这 t_1 个已被预先指派有分组标号的传感器节点中存在 j 个不同分组标号的概率为

$$\frac{1}{k^{t_1}} \cdot \sum_{i=0}^{j-1} (-1)^i \binom{k}{j} \binom{j}{i} (j-i)^{t_1}.$$

令

$$P(t_1, k, j) = \frac{1}{k^{t_1}} \sum_{i=0}^{j-1} (-1)^i \binom{k}{j} \binom{j}{i} (j-i)^{t_1},$$

则依据算法 1 的步骤 2 可知, 由于所有的 t_2 个尚未被预先指派有分组标号的传感器节点将会被指派不同的分组标号, 因此, 在所有的 t 个传感器节点之中存在 k 个节点具有不同的分组标号的概率为

$$\sum_{j=(k-t_2)}^{\min\{k,t_1\}} P(t_1, k, j).$$

在文献 [6] 给出的 RSGC 算法中, 每个传感器节点将会从 $\{0, 1, \dots, k-1\}$ 之中随机选择一个数字作为其分组标号. 假定在一个 t -子簇 ($t \geq k$) 中, 令 P_{RSGC} 表示执行 RSGC 调度算法之后存在 k 个不同的分组标号的概率, 则基于下述有关 Stirling 数的公式 (2):

$$S(n+1, j) = S(n, j-1) + jS(n, j). \quad (2)$$

易证下述定理 5 成立.

定理 5 假定无线传感器网络为 m -覆盖与连通的, 且网络中的所有节点都被指派到了 k ($k \leq m$) 个不同的分组 $\{0, 1, \dots, k-1\}$ 之中, 则本文提出的节点调度算法 THCNS 具有比算法 RSGC 更好的分组覆盖率保持性能, 且

$$P_{\text{THCNS}} - P_{\text{RSGC}} = \begin{cases} \sum_{j=1}^{k-1} P(t, k, j), & \text{或 } t_2 \geq k; \\ \sum_{i=0}^{t_2-1} \frac{1}{k^{t-i}} \left((k-i-1)(k-i-1)! \binom{k}{k-i-1} S(t-i-1, k-i-1) \right), & \text{其他.} \end{cases}$$

为了更直观地说明上述定理 5 的结论, 图 1 给出了算法 THCNS 比算法 RSGC 实际提高的分组覆盖率, 其中, 我们假设给定的位置点 p 可被 t 个不同的传感器节点所覆盖, 且无线传感器网络中的所有节点均被分配到了 k 个不同的分组之中.

由以上图 1 易知, 算法 THCNS 的分组覆盖率要远远高于算法 RSGC 的分组覆盖率, 特别地, 算法 THCNS 的分组覆盖率在一些情形下可达到 100%, 而算法 RSGC 的分组覆盖率则非但不能达到 100%, 而且还非常低, 仅有大约 20%–30% 左右.

接下来, 再进一步评估算法的通信开销 (communication overhead), 在评估中将主要以每个传感器节点所需要的平均广播次数为依据. 显然, 在算法 4 的步骤 1 中, 除了 Sink 节点之外, 其余的所有节点均不需要进行广播. 在步骤 2 中, 由于每个节点需要通过信息交换来确认自己的活跃邻节点, 因此需要一轮广播. 在步骤 3 中, 由于每个节点需要广播一个包含有自己的节点标号与 IDNN 的消息包给自己的所有邻节点, 因此也需要一轮广播. 在步骤 4 中, 由于在执行算法 2 时需要一轮广播, 因此在该步骤中也需要一轮广播. 在步骤 5 中, 于在执行算法 3 时需要一轮广播, 因此在该步骤中也需要一轮广播. 而在步骤 6 中, 各组节点轮流工作, 显然无需进行任何广播. 由此可知, 算法 THCNS 中的每个节点总共需要 4 轮广播.

此外, 再进一步考虑算法 THCNS 的存储开销 (storage overhead) 如下: 首先, 由于每个节点维护的信息表 (IT) 中包含以下信息: IDNN, NNLN, IUNN, TSHS, 以及参数 k . 对网络中的任意传感器节点, 令 $N_{\max}$ 表示最大的邻节点数目, $H_{\max}$ 表示到 Sink 节点的最大跳数计数, N 表示网络中传感器节点的个数, 由于传感器节点均被按照 n -维超立方体的编码方式进行了编码, 因此, 显然每个节点的节点标号的长度为 $\lg N$. 由此一来, 每个节点维护信息表 (IT) 的存储开销将不超过 $[N_{\max} \cdot \lg N + N_{\max}^2 \cdot \lg N + \lg H_{\max} + \lg k]$. 其次, 每个节点在被指派了分组标号之后还需维护分组标号列表, 由于分组标号总共有 k 个, 因此易知维护分组标号列表的存储开销将不超过 $[k \cdot \lg k]$. 另外, 由于 $k \leq m$, 且 $m \leq N_{\max} < N$, 因此, 算法 THCNS 的总存储开销可估算为 $O(N_{\max}^2 \cdot \lg N)$.

5.2 实验评价

为了进一步评价算法 THCNS 的实际性能, 本小节基于 Java 实现了一个实验平台, 并基于以下指标通过仿真实验比较了算法 THCNS 与 RSGC 的性能.

图 1 算法 THCNS 与 RSGC 的分组覆盖率比较

Figure 1 Comparison between THCNS and the randomized scheduling scheme RSGC

(a) $t = 10$ and $k = 6$; (b) $t = 15$ and $k = 8$

图 2 算法 THCNS 与 RSGC 的分组覆盖率比较

Figure 2 Coverage ratio between THCNS and the randomized scheduling scheme RSGC

(a) Network density vs coverage ratio; (b) k vs coverage ratio

图 3 算法 THCNS 与 RSGC 的网络生命周期比较

Figure 3 Network lifetime between THCNS and the randomized scheduling scheme RSGC

(a) Network density vs network lifetime; (b) k vs network lifetime

(1) 分组覆盖率 (coverage ratio): 主要比较了在不同的节点分布密度与不同的分组数目 (即不同的 k 值) 情况下, 基于算法 THCNS 与 RSGC 的分组覆盖率大小, 其中, 分组覆盖率是指每个分组中的节点对整个传感区域的覆盖率。

(2) 网络生命周期 (lifecycle): 主要比较了在不同的节点分布密度与不同的分组数目 (即不同的 k 值) 情况下, 基于算法 THCNS 与 RSGC 的网络生命周期长短, 其中对任意分组, 若分组中的节点不再全局连通或分组中的死亡节点比率达到或超过 5%, 则认为该分组已经死亡。

依据以上指标, 实验结果分别如下图 2 和 3 所示. 特别地, 为了简化实验过程, 在图 3 中仅考虑了传感器节点在工作阶段的能量消耗, 而忽略了有关的计算能量开销。

图 2(a) 给出了当 $k = 5$ 时算法 THCNS 与 RSGC 在不同节点分布密度下的分组覆盖率比较. 由图 2(a) 易知, 算法 THCNS 的分组覆盖率要远远优于 RSGC. 图 2(b) 给出了当 $r_s = 30$ 时算法 THCNS 与 RSGC 在不同分组数目下的分组覆盖率比较. 由图 2(b) 也易知, 当分组数目 k 变大时, 算法 RSGC 的分组覆盖率将大幅降低, 而算法 THCNS 则依然可以保持较高的分组覆盖率.

图 3(a) 给出了当 $k = 5$ 时算法 THCNS 与 RSGC 在不同节点分布密度下的网络生命周期比较. 由图 3(a) 易知, 算法 THCNS 的网络生命周期要远远长于 RSGC. 图 3(b) 则给出了当 $r_s = 30$ 时算法 THCNS 与 RSGC 在不同分组数目下的网络生命周期比较. 图 3(b) 也证明了算法 THCNS 比 RSGC 能更加有效地延长无线传感器网络的生命周期.

6 结论

在目前已有的无线传感器网络节点调度技术的研究之中, 在不知道节点的位置信息情况下, 传统方法一般难于同时解决分组覆盖率与全局连通性保持的问题, 本文提出了一种新的节点调度算法, 不但可以在不知道节点的位置信息情况下, 将 m -覆盖与连通的无线传感器网络中的所有节点分布式地指派到 $k(k \leq m)$ 个不同的分组 $\{0, 1, \dots, k-1\}$ 之中, 而且还可同时保证调度之后的每个分组中的节点不但具有全局连通性, 而且还具有较高的网络覆盖率. 理论分析与实验结果表明, 新算法具有比已有传统随机节点调度算法更优的节点调度性能, 是一种适合无线传感器网络特性且具有良好可行性的联合节点调度方法.

参考文献

- 1 Akyildiz I F, Su W, Sankarasubramaniam Y, et al. A survey on sensor networks. *IEEE Commun Mag*, 2002, 40: 102–114
- 2 Wang C, Yin Q Y, Wang W J. An efficient ranging method based on Chinese remainder theorem for RIPS measurement. *Sci China Inf Sci*, 2010, 53: 1233–1241
- 3 Wang L, Wei R Z. Reputation model based pair-wise key establishment scheme for sensor networks. *Ad Hoc Sensor Wirel Netw*, 2010, 9: 163–177
- 4 Tang Q L, Yang L Q, Qin T F, et al. Energy-saving PPM schemes for WSNs. *Sci China Ser F-Inf Sci*, 2008, 51: 571–585
- 5 Chen B J, Jamieson K, Balakrishnan H, et al. Span: an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wirel Netw*, 2002, 8: 481–494
- 6 Liu C, Wu K, Xiao Y, et al. Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks. *IEEE Trans Parall Distr*, 2006, 17: 562–575
- 7 Li X Y, Wan P J, Frieder O. Coverage in wireless ad hoc sensor networks. *IEEE Trans Comput*, 2003, 52: 753–763
- 8 TSAI Y R. Coverage preserving routing protocols for randomly distributed wireless sensor networks. *IEEE Trans Wirel Commun*, 2007, 6: 1240–1245
- 9 Cortes J, Martinez S, Karatas T, et al. Coverage control for mobile sensing networks. *IEEE Trans Robot Autom*, 2004, 20: 243–255
- 10 Yong D, Chen W, Li X. A connectivity based partition approach for node scheduling in sensor networks. *Lect Notes Comput Sci*, 2007, 4549: 354–367
- 11 Godfrey P B, Ratajczak D. Naps: scalable, robust topology management in wireless ad hoc networks. In: *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. Berkeley: ACM, 2004. 443–451
- 12 Wang X, Xing G, Zhang Y, et al. Integrated coverage and connectivity configuration in wireless sensor networks. In: *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems(SenSys)*. Los Angeles:

- ACM, 2003. 28–39
- 13 Tian D, Georganas D. Connectivity maintenance and coverage preservation in wireless sensor networks. *Ad Hoc Netw*, 2005, 3: 744–761
 - 14 Kumar S, Lai T H, Balogh J. On k -coverage in a mostly sleeping sensor network. In: *Proceedings of the ACM International Conference on Mobile Computing and Networking(MobiCom)*. New York: ACM, 2004. 144–158
 - 15 Wu Y W, Li X Y, Liu Y H, et al. Energy-efficient wake-up scheduling for data collection and aggregation. *IEEE Trans Parall Distr*, 2010, 21: 275–287
 - 16 Lu J, Bao L C, Suda T. Probabilistic self-scheduling for coverage configuration in wireless ad-hoc sensor networks. *Int J Perv Comput Commun*, 2008, 4: 26–39
 - 17 Wang L, Lin Y P, Liu Y H. Key distribution for group-based sensor deployment using a novel interconnection graph. In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM)*. Helsinki: IEEE, 2007. 1–9

Cluster based node scheduling method for wireless sensor networks

WANG Lei^{1,2*}, WEI RuiZhong² & TIAN ZiHong^{2,3}

¹ College of Software, Hunan University, Changsha 410082, China;

² Department of Computer Science, Lakehead University, Thunder Bay P7B 5E1, Canada;

³ Mathematics and Information Science College, Hebei Normal University, Shijiazhuang 050016, China

*E-mail: wanglei@hnu.cn

Abstract By researching on the node scheduling problem of m -covered and connected sensor networks, a new concept of two-hops-cluster is proposed in this paper, and based on it, a new distributed node scheduling algorithm THCNS for allocating all nodes in the sensor network into k ($k \leq m$) different groups $\{0, 1, \dots, k-1\}$ is designed, without requiring location information. Our algorithm guarantees that each group to be connected and maintains the coverage ratio with high possibility. Theoretical analysis and simulation results show that it has better performance than previous randomized scheduling scheme, and can prolong the lifetime of the sensor network effectively.

Keywords node scheduling, lifetime, wireless sensor network, two-hops-cluster