

## 顽健的无线传感器网络 $K$ 近邻查询处理算法

刘亮, 秦小麟, 刘亚丽, 李博涵

(南京航空航天大学 信息科学与技术学院, 江苏 南京 210016)

**摘要:** 提出了一种顽健的  $K$  近邻查询处理算法 ROC-KNN, 根据网络拓扑动态地将查询区域划分成若干子区域。每个子区域中选择一个簇头节点收集其他节点的感知数据, 并将其发送至下一个子区域的簇头节点, 直至遍历所有子区域。给出了 2 种分布式的启发式算法, 用于设置子区域大小和选择簇头节点, 以减少能量消耗。设计了一种利用子区域中非簇头节点恢复查询处理过程的算法, 降低了查询处理因簇头节点失效而中断的概率。实验结果表明, ROC-KNN 在能量消耗、查询成功率方面均优于现有的算法。

**关键词:** 无线传感器网络; 查询处理;  $K$  近邻查询; 顽健性; 节点失效

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2010)11-0171-09

## Robust $K$ nearest neighbor query processing algorithm in wireless sensor networks

LIU Liang, QIN Xiao-lin, LIU Ya-li, LI Bo-han

(College of Information Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

**Abstract:** A robust  $K$  nearest neighbor query processing algorithm called ROC-KNN was proposed. It divides the query region into several sub-regions according the network topology. Each sub-region has a cluster node which collects the sensory data in it, sends it to the cluster node in the next sub-region until traversing all the sub-regions. Two distributed heuristic sub-region size setting and cluster node election algorithms were proposed to reduce the energy consumption. A query processing recovery algorithm using the non-cluster nodes in each sub-region was designed to reduce the outage probability caused by cluster node failures. The experimental results show that the ROC-KNN outperforms the existing algorithms in terms of energy consumption and query success rate.

**Key words:** wireless sensor networks; query processing; KNN query; robustness; node failures

### 1 引言

传感器网络在国防军事、环境检测、医疗服务

和交通控制等领域有着广泛的应用前景, 近年来成为研究热点。传感器网络是以数据为中心的网络, 用户使用它的主要目的是查询其感知的数据或监控

收稿日期: 2010-07-15; 修回日期: 2010-10-07

**基金项目:** 国家自然科学基金资助项目(60673127); 国家高技术研究发展计划(“863”计划)基金资助项目(2007AA01Z404); 江苏省支撑计划基金资助项目(BE2008135); 工信部电子信息产业发展基金资助项目; 南京航空航天大学基本科研业务费专项科研基金资助项目(NS2010101); 江苏省普通高校研究生科研创新计划基金资助项目(CX10B\_112Z); 南京航空航天大学博士学位论文创新与创优基金资助项目(BCXJ10-07)

**Foundation Items:** The National Natural Science Foundation of China(60673127); The National High Technology Research Program of China (863 Program) (2007AA01Z404); The Jiangsu Province Science & Technology Pillar Program(BE2008135); The Electronic Development Fund of the Ministry of Information Industry; NUAA Research Funding(NS2010101); Funding of Jiangsu Innovation Program for Graduate Education(CX10B\_112Z); Funding for Outstanding Doctoral Dissertation in NUAA( BCXJ10-07)

的事件，常用的查询是  $K$  近邻查询。 $K$  近邻查询  $KNNQ(o, K)$  可以形式化为：给定一个包含有  $C$  个节点的传感器网络  $SN$  和查询点  $o$ ，找出一个含有  $K$  ( $K \leq C$ ) 个节点的子集  $SNK$ ，满足  $SNK \subseteq SN$  且对于  $\forall n_1 \in SNK$  和  $\forall n_2 \in SN - SNK$ ， $Dist(n_1, o) \leq Dist(n_2, o)$ ，其中  $Dist$  是欧几里得距离函数。 $K$  近邻查询用于获得距查询点最近的  $K$  个传感器节点的数据。当传感器网络监控区域有事件发生时，获得以事件发生地为查询点的  $K$  近邻查询结果，对分析该事件发生的原因和预测事件未来的发展趋势有着极其重要的作用。

在传统的空间数据库领域，对  $K$  近邻查询处理算法进行了深入的研究<sup>[1]</sup>。这些算法均假设所有数据存储在集中式数据库中。它们通常利用空间索引技术<sup>[2]</sup>减少磁盘 I/O 的访问次数，以提高处理查询的速度。由于传感器节点的能量十分有限，先将所有传感器节点的位置和感知数据收集至基站然后进行处理，势必造成大量的能量消耗。文献[3~10]利用传感器节点的计算和存储能力，在网内进行查询处理以减少能耗。但这些算法没有综合考虑网络拓扑变化、节点失效对查询处理的影响，导致查询处理过程极易中途被迫中止，查询结果频繁丢失，顽健性差。

为了避免这些问题，提出了一种顽健的传感器网络  $K$  近邻查询处理算法 ROC-KNN ( $K$  nearest neighbor query processing algorithm based on robust data collection protocol)。它根据网络拓扑动态地将查询区域划分成若干子区域。每个子区域中选择一个簇头节点收集其他节点的感知数据，并将其发送至下一个子区域的簇头节点，直至遍历所有子区域，生成最终的查询结果。当簇头节点失效时，该簇头节点所在子区域中的非簇头节点能够代替它继续查询处理过程，降低了算法因簇头节点失效而中断的概率。实验结果表明，在绝大多数情况下，ROC-KNN 在能量消耗、查询成功率方面均优于现有的算法。

## 2 相关工作

目前，在网内处理  $K$  近邻查询的算法按是否依赖网络拓扑结构可以分为以下 2 类。

1) 与拓扑结构相关的算法。文献[3,4]利用节点的位置信息将传感器网络组织成一棵分布式的  $R$  树<sup>[2]</sup>。节点收到查询消息后，利用  $R$  树判断以其为根的子树是否包含查询结果，这样无需访问不包含查询结果的子树，减少了能量的消耗。文献[5]首先利用位

置路由协议<sup>[11]</sup>将查询消息发送至距离查询点最近的节点（主节点）。到达主节点后，根据位置路由查询消息阶段收集到的节点信息估计出包含  $KNN$  查询结果的查询区域边界大小，然后将查询区域中的节点组织成树形拓扑结构，收集该区域中节点的感知数据，生成最终的查询结果。

2) 与拓扑结构无关的算法<sup>[6~10]</sup>。其核心思想是首先利用位置路由协议将查询消息发送至距查询点最近的节点，然后沿一条或多条以该节点为起点的路线，利用基于路线的数据收集协议<sup>[12]</sup>（为了讨论方便，本文将其命名为 IC 协议）收集距查询点最近的  $K$  个节点的数据。收集到距查询点最近的  $K$  个节点的数据后，利用位置路由协议将最终的查询结果返回至查询发起节点。

与拓扑结构无关的算法性能主要取决于路线的选择和数据收集算法。IKNN<sup>[6]</sup>设计了如图 1 所示的查询路线。算法沿该路线遍历完一圈后，判断已遍历的节点数目是否不小于  $K$ ，若是，则收集过程结束，查询路线到达的最后一个节点计算出离查询点最近的  $K$  个节点，并将这些节点感知的数据返回至查询发起节点；若否，则继续下一圈的遍历，直到遍历的节点数目不小于  $K$ 。IKNN 的优点是无需估计包含  $K$  近邻节点的区域边界且不用访问大量与查询结果无关的节点。但由于 IKNN 的查询处理过程是串行的，其时间延迟较大，且随着路线的变长，发送的数据量不断增大，带来了大量的能量消耗。针对该问题，文献[7~10]提出了采用多条数据收集路线的并行  $K$  近邻查询处理算法 DIKNN 和 PCIKNN。这些算法利用多条路线并行执行以减少时间延迟和能量消耗。

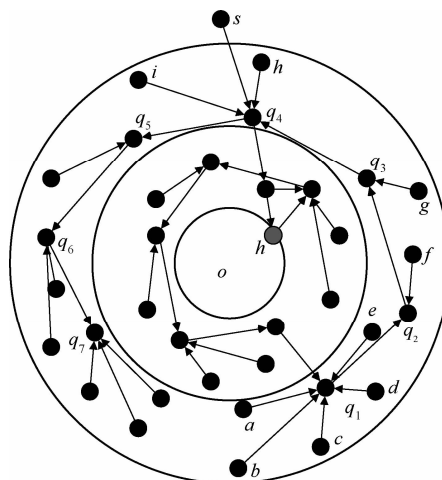


图 1 IKNN 算法

与拓扑结构无关的算法使用的基于路线的数据收集协议 IC 的核心思想如下: 查询路线上的节点(查询节点)首先将查询消息广播至其邻居节点(数据节点), 然后收集数据节点的感知数据, 最后将收集到的数据、本地的感知数据和从上一个查询节点接收到的感知数据集合发送给查询路线上的下一个节点, 如此继续, 直至遍历所有节点。

现有的与网络拓扑结构相关的算法缺点在于: 由于网络拓扑结构频繁发生变化, 维护拓扑会带来大量的能量消耗; 当节点失效时, 查询结果极易丢失, 导致查询结果不准确。与拓扑结构无关的算法根据网络拓扑信息动态地形成收集感知数据的路线, 因而能够降低网络拓扑变化对查询处理的影响。但该类算法还存在节点失效问题, 即当查询路线上的任意一个查询节点(例如图 1 中的节点  $q_2$ 、 $q_3$  等)失效时, 则该节点收集的感知数据全部丢失, 查询处理过程被迫中断, 且无法返回查询结果。

### 3 顽健的传感器网络 $K$ 近邻查询处理算法 ROC-KNN

#### 3.1 假设

传感器节点通信半径相等。节点能通过定位算法获得自己的位置信息。所有节点定时广播自己的位置信息, 因而各节点能获得其所有一跳邻居节点的位置信息。节点的感知数据存储于本地。由于传感器节点的能量十分有限,  $K$  近邻查询处理算法的优化目标在于使得查询处理过程中消耗的能量最小。

#### 3.2 算法基本思想

为了避免现有算法的问题, 提出了一种顽健的传感器网络  $K$  近邻查询处理算法。它分为以下 3 个阶段。

1) 位置路由查询消息阶段, 即利用位置路由协议<sup>[11]</sup>将查询消息发送至距离查询点最近的节点  $h$ 。

2) 分发查询和收集感知数据阶段, 即沿一条或多条以  $h$  为起点的查询路线, 利用 ROC (robust data collection protocol) 数据收集协议收集距查询点最近的  $K$  个节点的数据。

3) 返回最终查询结果阶段, 即利用 ROC 收集到这些数据后, 利用位置路由协议将最终的查询结果返回至查询发起节点。

基于 ROC 数据收集协议, 采用不同的查询路线对应于不同的  $K$  近邻查询处理算法。本文将基于

ROC 的  $K$  近邻查询处理算法统称为 ROC-KNN, 基于 ROC, 分别采用 IKNN<sup>[6]</sup>、DIKNN<sup>[8]</sup>、PCIKNN<sup>[9,10]</sup> 中的查询路线的  $K$  近邻查询处理算法命名为 ROC-IKNN、ROC-DIKNN 和 ROC-PCIKNN。

假设节点间的通信半径为  $R$ , 点  $p_1$  和点  $p_2$  之间的距离函数记为  $Dist(p_1, p_2)$ 。不失一般性, 以图 2 中的网络为例。以  $s$  为起始节点, 采用 ROC 协议收集圆环中所有节点的感知数据过程如下。

1)  $s$  收到查询消息后, 利用位置路由协议将查询消息发送至查询区域内的一个节点  $c_0$ 。

2)  $c_0$  收到查询消息后, 沿逆时针(或顺时针)方向收集圆环中节点的数据。假设线段  $Oc_0$  和圆环的内圆与外圆的分别相交于点  $A$  和点  $D$ 。节点  $c_0$  根据其维护的邻居节点位置信息计算出一个在圆环内的子区域  $NA(c_0)$ , 并在位于子区域  $NA(c_0)$  中的节点集合中选择一个簇头节点  $NC(c_0)$ , 用于收集子区域  $NA(c_0)$  中其他节点的数据。

定义子区域  $NA(z)$  为节点  $z$  的后继子区域, 节点  $NC(z)$  为节点  $z$  的后继簇头节点, 子区域  $NA(z)$  中除节点  $NC(z)$  之外的节点为数据节点。后继子区域  $NA(z)$  和后继簇头节点  $NC(z)$  满足以下约束条件(子区域约束条件): 子区域  $NA(z)$  中的所有节点均是节点  $z$  的邻居节点; 子区域  $NA(z)$  中的所有节点均是节点  $NC(z)$  的邻居节点。

3) 节点  $c_0$  将查询消息及其感知数据发送至节点  $NC(c_0)$ 。由于无线通信的广播特性, 子区域  $NA(c_0)$  中的所有节点均能收到该消息。子区域  $NA(c_0)$  中的节点收到查询消息后, 该子区域内的数据节点按一定的次序依次将本地的感知数据发送至节点  $NC(c_0)$ 。此时, 节点  $NC(c_0)$  利用  $c_0$  发送的感知数据、自己的感知数据和子区域  $NA(c_0)$  中数据节点发送的感知数据, 能收集到子区域  $NA(c_0)$  中所有节点的感知数据。

4) 节点  $u=NC(c_0)$  收集子区域  $A_u=NA(c_0)$  中的数据结束后, 计算其后继簇头节点  $NC(u)$  以及后继子区域  $NA(u)$ , 并将查询消息和子区域  $NA(c_0)$  中的感知数据集合发送至节点  $NC(u)$ 。与步骤 2) 类似, 节点  $NC(u)$  收集子区域  $NA(u)$  中节点的感知数据, 并根据  $NA(c_0)$  中的感知数据集合(由节点  $NC(c_0)$  发送)和子区域  $NA(u)$  中节点发送的感知数据, 生成  $NA(c_0) \cup NA(u)$  中节点的感知数据集合。

5) 按步骤 3) 继续收集圆环中未遍历节点的数据, 直至遍历圆环中的所有节点, 计算出最终的查询结果。

由于受到节点软硬件故障、节点能量不足、节点休眠等因素的影响，传感器节点失效频繁发生。在 ROC 协议中，当簇头节点失效时，簇头节点所在子区域的数据节点代替簇头节点继续查询处理，以避免查询处理过程因节点失效而中断。以图 2 为例，考虑簇头节点  $c_1$  先收集子区域  $ABCD$  中的数据，然后将收集到的感知数据集合发送至子区域  $BEFC$  的簇头节点  $c_2$  这一过程。假设在节点集合  $\{a, b, c, c_0, c_1\}$  中仅  $a, c$  不能互相通信。在执行过程中，由于节点  $a, b, c_0$  能互相通信，根据无线通信的广播特性，节点  $a, b, c_0$  均能分别计算出子区域  $AB_1C_1D$  中节点  $a, b, c_0$  的感知数据集合。若簇头节点  $c_1$  发生失效，则子区域  $AB_1C_1D$  中的任意一个节点可以代替节点  $c_1$ ，继续收集圆环中除  $AB_1C_1D$  外剩余区域的感知数据。

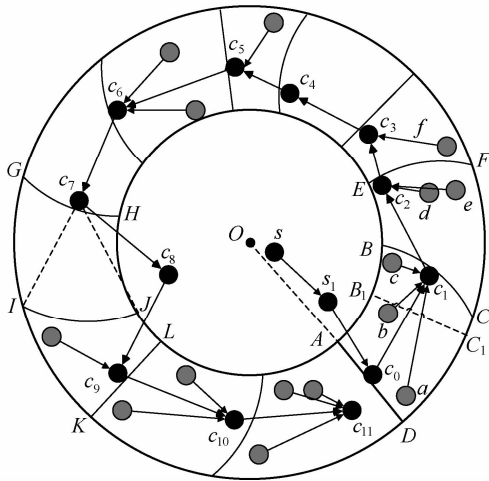


图 2 顽健的数据收集算法 ROC

综上所述，ROC 协议仅要求节点维护其邻居节点的位置信息，根据该信息动态地进行子区域划分和选择子区域的簇头节点，避免了网络拓扑结构变化对查询处理的影响。与此同时，当簇头节点失效时，ROC 协议利用备份节点恢复查询处理过程，降低了算法因节点失效而中断的概率。

### 3.3 算法能耗分析及优化

由于 ROC-KNN 算法的能量消耗取决于 ROC 数据收集协议，本节对 ROC 的能量消耗公式进行推导。为了便于讨论和理论分析，先作以下假设与定义。传感器网络表示为  $SN=\{s_1, s_2, \dots, s_C\}$ ，节点密度记为  $\rho$ 。对于区域  $SR$ ，函数  $N(SR)$  返回  $SR$  中的节点总数，函数  $Ns(SR)$  返回在区域  $SR$  中的节点集合，函数  $Area(SR)$  返回子区域  $SR$  的面积。节点通信半

径相等且为  $R$ 。平均每个节点有  $N_n$  个邻居节点。发送和接收 1byte 消耗的能量分别为  $E_t$  和  $E_r$ ，节点广播一个 1byte 网络消耗的能量为  $E=E_t+N_n \times E_r$ 。查询消息包的大小为  $S_q$ ，查询区域内各节点的感知数据消息包大小相等且为  $S_a$ 。  $Dist(p, q)$  表示点  $p$  与点  $q$  之间的距离。记以点  $o$  为圆心，内圆半径和外圆半径分别为  $R_l$  和  $R_h$  的圆环为  $Rg(o, R_l, R_h)$ 。  $Rg$  的宽度  $Rg.w=R_h-R_l$ ，  $Rg$  的面积  $A_r=2\pi[(R_h)^2-(R_l)^2]$ ，  $Rg$  中的节点数目  $N_r=N(Rg)$ 。点  $p$  到圆环  $Rg$  的圆心  $o$  之间的距离记为  $Dist(p, Rg)$ 。在圆环  $Rg$  中起始角度和终止角度分别为  $RA_s$  和  $RA_e$  的环扇区记作  $Rs(Rg, RA_s, RA_e)$ 。记以点  $o$  为圆心，半径为  $R_c$  的圆记为  $Circle(o, R_c)$ 。

不失一般性，考虑采用 ROC 协议收集在圆环  $Rg(o, R_l, R_h)$  (其中，  $Rg.w \leq R$ ) 内所有感知数据的情况。进行数据收集的起始节点  $SRC_0$  位于圆环内。ROC 协议将 Ring 划分成  $\Phi$  个子区域  $SRS=\{SR_1, SR_2, \dots, SR_\Phi\}$ ，  $SRS$  对应的簇头节点集合为  $SRCs=\{SRC_1, SRC_2, \dots, SRC_\Phi\}$ 。记区域  $\bigcup_{j=1}^{j=i} SR_j$  中所有节点的感知数据集合为  $DS(SR_i)$  (其中，  $1 \leq i \leq \Phi$ )。

采用 ROC 协议收集圆环内所有节点的数据消耗的能量为

$$E_{Ring}(ROC) = \sum_{i=1}^{i=\Phi} [E_q(SR_i) + E_d(SR_i) + E_c(SR_i)] \quad (1)$$

其中，  $E_q(SR_i)$  表示簇头节点  $SRC_{i-1}$  将查询消息发送至簇头节点  $SRC_i$  消耗的能量，  $E_d(SR_i)$  表示簇头节点  $SRC_{i-1}$  将感知数据集合  $DS(SR_{i-1})$  发送至簇头节点  $SRC_i$  消耗的能量，  $E_c(SR_i)$  表示簇头节点  $SRC_i$  收集子区域  $SR_i$  中节点的感知数据集合消耗的能量。由 ROC 协议的执行过程可知：

$$E_q(SR_i) = S_q E \quad (2)$$

$$E_d(SR_i) = \sum_{j=1}^{j=i-1} N(SR_j) S_a E \quad (3)$$

$$E_c(SR_i) = [N(SR_i) - 1] S_a E \quad (4)$$

根据式(2)~式(4)，式(1)化简后可得：

$$E_{Ring}(ROC) = [\Phi S_q + N_r S_a - \Phi S_a + \sum_{i=1}^{i=\Phi} (\sum_{j=1}^{j=i-1} N(SR_j) S_a)] E \quad (5)$$

为了减少 ROC 协议的能量消耗，等价于求解优化问题：

Minimize  $E_{Ring}(ROC)$  subject to.

$$\sum_{i=1}^{\Phi} Area(SR_i) = A_r$$

$$Dist(SRC_{i-1}, n) \leq R, \forall n \in Ns(SR_i), 1 \leq i \leq \Phi$$

$$Dist(n, SRC_i) \leq R, \forall n \in Ns(SR_i), 1 \leq i \leq \Phi \quad (6)$$

其中, 式(6)的约束条件为子区域约束条件。显然, 子区域的数目、划分方法以及子区域簇头节点的选择策略决定了  $E_{Ring}(ROC)$  的大小。为了求出式(6)的最优解, 需要知道圆环内所有未失效节点的位置信息。但是收集这些信息会带来大量的能量消耗, 另外, 由于受到节点移动、节点失效和节点休眠等因素的影响, 节点的状态和网络拓扑频繁发生变化。因而即使获得了当前时刻圆环内所有未失效节点的位置信息, 并不能保证根据这些信息求出的最优解在网络状态发生变化后依然是最优的。

基于上述分析, 给出一种求解式(6)的近似方法。假设子区域的平均面积为  $A_1$ , 可以近似认为:  $\Phi \approx A_r/A_1$ 。可以得到:

$$E_{Ring}(ROC) \approx [\Phi S_q + N_r S_a - \Phi S_a + (\Phi - 1)\Phi \rho A_1 S_a / 2] E$$

化简后得到:

$$E_{Ring}(ROC) \approx \left[ \frac{A_r}{A_1} S_q + N_r S_a - \frac{A_r}{A_1} S_a + \left( \frac{A_r}{A_1} - 1 \right) N_r S_a / 2 \right] E \quad (7)$$

在式(7)中, 仅  $A_1$  是变量, 其余参数均为常量。易知,  $E_{Ring}$  的值近似与子区域的平均面积成反比。因此, 在满足子区域约束条件的前提下, 使各子区域的面积尽可能大, 可以增大子区域的平均面积, 从而达到减小算法能量消耗的目的。

### 3.4 算法设计

由 ROC 协议的执行过程可知, 簇头节点  $SRC_i(0 \leq i \leq \Phi - 1)$  将收集到的感知数据集合  $DS(SR_i)$  以及查询消息 (包含圆环  $Rg$ 、子区域  $SR_{i+1}$  等信息) 广播至子区域  $SR_{i+1}$  的簇头节点  $SRC_{i+1}$  后, 簇头节点  $SRC_i$  的邻居节点收到该消息需要判断是否在子区域  $SR_{i+1}$  中, 以决定是否发送其感知数据至节点  $SRC_{i+1}$ 。由于传感器节点具有有限的计算能力, 子区域形状过于复杂会导致传感器节点计算是否在子区域内的代价太大, 同时使得查询消息过大, 增大了分发查询消息的能量消耗。基于上述分析, 给

出如下 4 种子区域表示方法 (令  $C_b, C_e$  为圆心在圆环  $Rg$  中的圆,  $\alpha_b, \alpha_e$  为角度值)。

1)  $SR(Rg, C_b, C_e)$ 。该子区域包含圆环  $Rg$  中在圆  $C_e$  内但不在圆  $C_b$  内的点。例如图 3 中的子区域  $GHIJ$ 。

2)  $SR(Rg, C_b, \alpha_e)$ 。该子区域包含圆环  $Rg$  中角度小于  $\alpha_e$  且不在圆  $C_b$  内的点。例如图 3 中的子区域  $IJKL$ 。

3)  $SR(Rg, \alpha_b, \alpha_e)$ 。该子区域为环扇区  $Rs(Rg, \alpha_b, \alpha_e)$ 。例如图 3 中的子区域  $KLMN$ 。

4)  $SR(Rg, \alpha_b, C_e)$ 。该子区域包含圆环中角度大于  $\alpha_b$  且在圆  $C_e$  内的点。例如图 3 中的子区域  $MNPQ$ 。

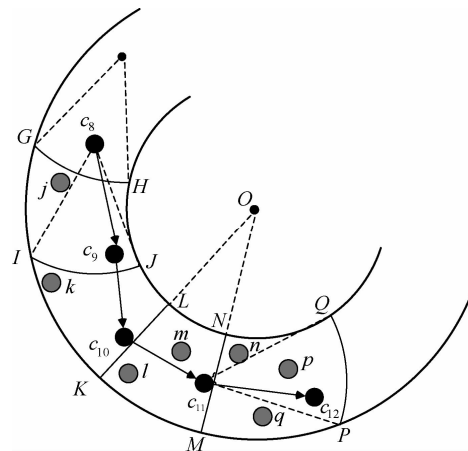
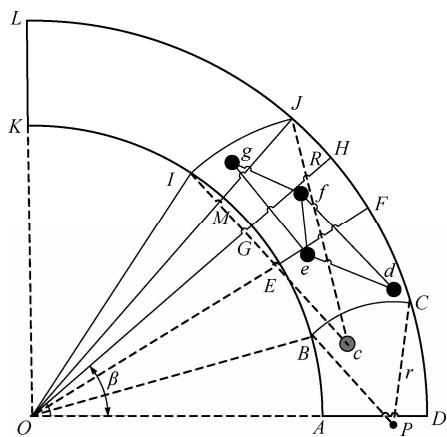


图 3 子区域形状

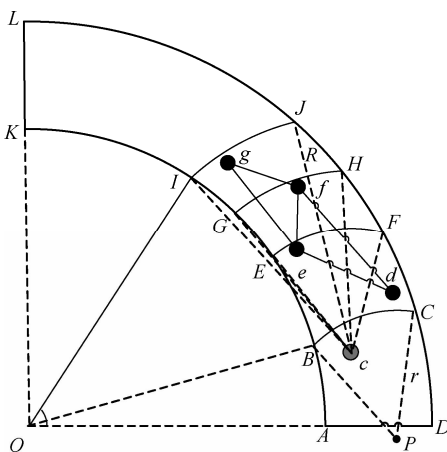
显然, 若子区域采用上述 4 种形状表示, 则传感器节点判断是否在子区域的计算量很小, 且查询消息包中用于表示子区域的字节数较少。

下面根据 3.3 节给出的降低 ROC 能耗的方法, 提出簇头节点选择和子区域大小设置算法。不失一般性, 以图 4 为例。以点  $P$  为圆心  $r (r \leq R)$  为半径的圆与圆环的内圆和外圆的交点分别为点  $B$  和点  $C$ 。以  $c$  为圆心  $R$  为半径的圆与圆环的内圆和外圆的交点分别为点  $I$  和点  $J$ 。区域  $BIJC$  中的节点  $d, e, f$  均在彼此的通信半径内, 节点  $e, f, g$  均在彼此的通信半径内。子区域  $ABCD$  的簇头节点为  $c$ 。假设簇头节点  $c$  在区域  $BIJC$  中的邻居节点集合记为  $N_c$ 。函数  $Ns(x)$  表示在区域  $x$  中节点集合。对于节点  $n$  和节点集合  $s$ , 若节点  $n$  能与节点集合  $s$  中的所有节点通信, 则函数  $f(n, s)$  返回真, 若否, 则返回假。簇头节点  $c$  收集完子区域  $ABCD$  中的感知数据后, 继续收集区域  $BKLC$  中的感知数据。簇头节点  $c$  的

后继子区域和后继簇头节点分别记为  $NA_c$  和  $NC_c$ 。提出 2 种根据节点  $c$  的邻居节点信息计算  $NA_c$  和  $NC_c$  的算法。



(a) ABA 算法



(b) DBA 算法

图 4 子区域大小设置及簇头节点选择算法示意图

1) 基于角度的算法(ABA, angle based algorithm)。其核心思想是：对  $N_c$  中的所有节点  $n_c$  按角度  $\angle n_c OA$  从小到大排序，排序后的节点列表记为  $AN_c$ 。在  $AN_c$  中查找节点  $AN_c[i]$  (其中,  $1 \leq i \leq |AN_c|$ )，满足条件：节点集合  $\{AN_c[0], \dots, AN_c[i-1]\}$  中的所有节点能与  $AN_c[i]$  通信。簇头节点  $c$  的后继簇头节点设置为满足上述条件且  $i$  取值最大的节点  $x$ ，后继子区域设置为  $SR(Rg, Circle(P,r), \angle xOA)$ 。以图 4(a) 为例，若采用 ABA 算法，簇头节点  $c$  的后继簇头节点为  $f$ ，后继子区域为  $SR(Rg, Circle(P,r), \angle fOA)$ ，即子区域  $BGHC$ 。ABA 的伪代码详见算法 1。

2) 基于距离的算法(DBA, distance based algorithm)。其核心思想是：对  $N_c$  中的所有节点按其到节点  $c$  的距离从小到大排序，排序后的节点列表

记为  $DN_c$ 。在  $DN_c$  中查找节点  $DN_c[i]$  (其中,  $1 \leq i \leq |DN_c|$ )，满足条件：节点集合  $\{DN_c[1], \dots, DN_c[i-1]\}$  中的所有节点能与  $DN_c[i]$  通信。簇头节点  $c$  的后继簇头节点设置为满足上述条件且  $i$  取值最大的节点  $x$ ，后继子区域设置为  $SR(Rg, Circle(P,r), Circle(c, Dist(c, x)))$ 。以图 4(b) 为例，若采用 DBA 算法，簇头节点  $c$  的后继簇头节点为  $f$ ，后继子区域为  $SR(Rg, Circle(P,r), Circle(c, Dist(c, f)))$ ，即子区域  $BGHC$ 。DBA 算法的伪代码与 ABA 类似，在此省略。

**算法 1 ABA**

输入：簇头节点  $c$ ,  $N_c$

输出： $c$  的后继子区域  $NA_c$ ,

$c$  的后继簇头节点  $NC_c$

- 1)  $NC_c = NULL; NA_c = NULL;$
- 2) IF  $|N_c| == 0; RETURN; END IF$
- 3) //对  $N_c$  中的所有节点  $n_c$  按角度  $\angle n_c OA$  从小到大排序
- 4)  $AN_c = sortByAngle(N_c);$
- 5)  $NC_c = AN_c[i]; i = |DN_c|;$
- 6) WHILE  $i > 1$
- 7) // 判断  $AN_c[i]$  是否能收到节点集合  $\{AN_c[1], \dots, AN_c[i-1]\}$
- 8) 所有节点发送的消息
- 9)  $NC_c = AN_c[i]; BREAK;$
- 10) ELSE
- 11)  $i = i - 1; CONTINUE;$
- 12) END IF
- 13) END WHILE
- 14)  $NA_c = SR(Rg, Circle(P, r), \angle NC_c OA)$

**4 实验**

由于 ROC-KNN 与现有的 IKNN、DIKNN、PCIKNN 算法的区别在于：IKNN 等算法采用了 IC 数据收集协议，而 ROC-KNN 使用 ROC 数据收集协议，且这些算法的性能由数据收集协议决定，因此仅对 IC 和 ROC 数据收集协议进行比较。在 ROC 协议中，可采用 ABA 或 DBA 算法作为选择后继簇头节点和设置后继子区域大小的算法。为了描述方便，将分别采用 ABA 和 DBA 算法的 ROC 协议命名为 AROC (robust data collection protocol based on ABA) 和 DROC (robust data collection protocol based on DBA)。

在文献[13]的仿真器上实现了 AROC、DROC

和现有的 IC 协议。实验的硬件环境为 P4(3.0GHz) CPU, 512MB 内存; 软件环境为 Ubuntu 操作系统、Eclipse 开发工具。对 3 种协议在不同节点密度、失效节点数目条件下的能量消耗和查询成功率进行了实验。查询成功率定义为:  $sr=se/te$ , 其中,  $te$  表示算法运行的总次数,  $se$  表示算法成功返回查询结果至查询发起节点的次数。

实验的参数选择如下: 根据文献[14], 无线通信电路发送和接收 1byte 的能量消耗公式为  $E_t=\alpha+\gamma \times d^e$ ,  $E_r=\beta$ 。其中,  $\alpha$  为通信发送电路消耗的能量,  $\gamma$  为传输放大器消耗的能量,  $d$  为传输距离;  $e$  为路径损失因子(path loss factor),  $\beta$  为通信接收电路消耗的能量。采用文献[13]中的参数:  $\gamma=10 \text{ pJ/bit/m}^2$ ,  $\alpha=45 \text{ nJ/bit}$ ,  $\beta=135 \text{ nJ/bit}$ ,  $e=2$ 。其他参数如表 1 所示。

参数名	参数值
网络覆盖区域	450m×450m
网络节点分布	随机均匀分布
节点通信半径	52m
节点数	400
感知数据消息大小	35byte
查询消息大小	10byte
圆环小圆半径	80m
圆环宽度	45m
实验次数	20
失效节点数目	8

### 4.1 查询成功率

图 5 显示了网络节点密度对簇头节点备份节点数目的影响。可见, IC 协议中的查询节点没有任何

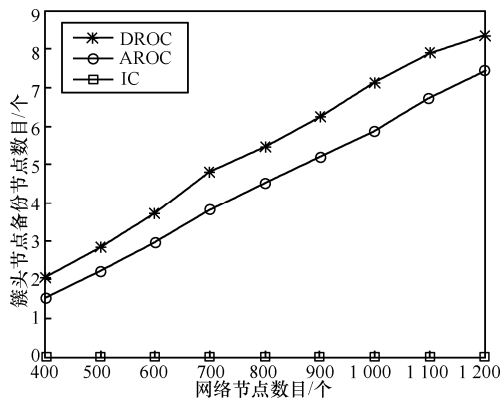


图 5 网络节点密度对备份节点数目的影响

备份节点, 而 DROC 和 AROC 协议的子区域中存在一些数据节点可作为簇头节点的备份节点, 且 DROC 协议的平均备份节点数目大于 AROC 协议。另外, 随着节点密度的增大, 子区域中的数据节点数目随之增大, 因而 DROC 和 AROC 协议的备份节点数目逐渐增大。

图 6 显示了失效节点数目为 24 时网络节点密度对查询成功率的影响。可见, DROC 和 AROC 协议的查询成功率均优于 IC 协议, 且随着网络节点密度的增大, IC 协议的查询成功率基本不变, 而 DROC 和 AROC 协议的查询成功率有较大的提高。当节点密度较小时, 因为 DROC 协议的备份节点数目大于 AROC 协议, 所以其查询成功率优于 AROC 协议; 当节点密度较大时, 2 种算法的查询成功率近似相等且趋近于 1。可以看出, 当网络节点密度变大时, DROC 和 AROC 协议簇头节点的备份节点数目随之增大, 有效地减小了节点失效对查询处理的影响。

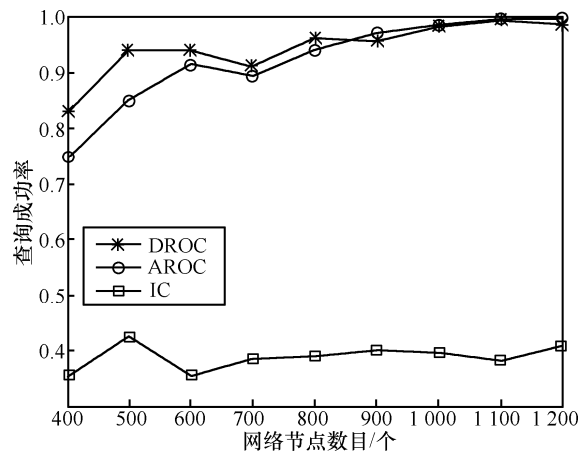


图 6 网络节点密度对查询成功率的影响

图 7 显示了网络失效节点数目对查询成功率的影响。可见, 随着节点失效概率的增大, 3 种算法

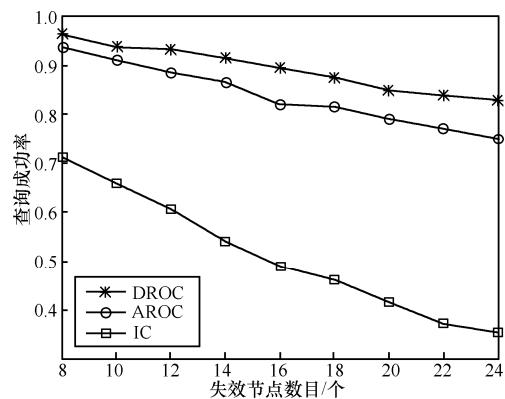


图 7 失效节点数目对查询成功率的影响

的查询成功率均逐渐降低。相比 IC 协议，由于 DROC 和 AROC 对簇头节点进行了冗余，其查询成功率的下降幅度较小。另外，由于 DROC 协议的平均备份节点数目大于 AROC 协议，其查询成功率优于 AROC 协议。

### 4.2 能量消耗

由式(7)可知，可以近似认为，DROC 和 AROC 协议的能量消耗与簇头节点数目成反比。图 8 显示了网络节点密度对簇头节点数目的影响。可见，采用 DBA 算法划分子区域和选择簇头节点的 DROC 协议的簇头节点数目少于基于 ABA 的 AROC 协议。另外，由于 DROC 和 AROC 协议在选择簇头节点时，需要满足子区域约束条件，而 IC 协议没有该约束，因而 IC 协议的查询节点数目少于 DROC 和 AROC 协议的簇头节点数目。

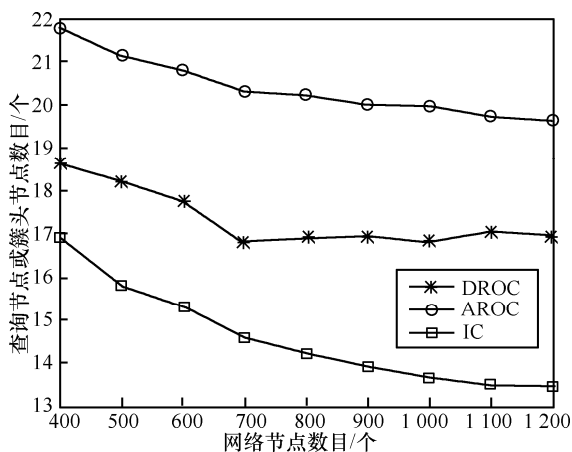


图 8 网络节点密度对簇头节点数目的影响

图 9 显示了节点不失效时网络节点密度对能量消耗的影响。可见，AROC 协议消耗的能量大于 IC 协议，且当网络节点数目  $n \geq 900$  时，DROC 协议消耗的能量大于 IC 协议。这是因为：AROC 和 DROC 协议的簇头节点数目大于 IC 协议的查询节点数目，使得圆环中节点的感知数据发送至最后一个簇头节点的路径变长，增大了能量消耗。当  $n < 900$  时，DROC 协议消耗的能量少于 IC 协议。这是因为：当网络节点密度较小时，DROC 协议的簇头节点数目与 IC 协议的查询节点数目近似相等。另外，在 IC 协议中，绝大多数圆环中的节点（例如图 1 中节点 *d*、*e*、*f* 等）需要先将感知数据发送至查询路线反方向上的查询节点，然后由查询节点将其发送至查询路线正方向上的最后一个查询节点，而在 DROC 协议中，节点的感知数据先发送至与查询路

线同向的簇头节点，再由簇头节点转发至查询路线正方向上的最后一个簇头节点，减少了转发感知数据的能量消耗。

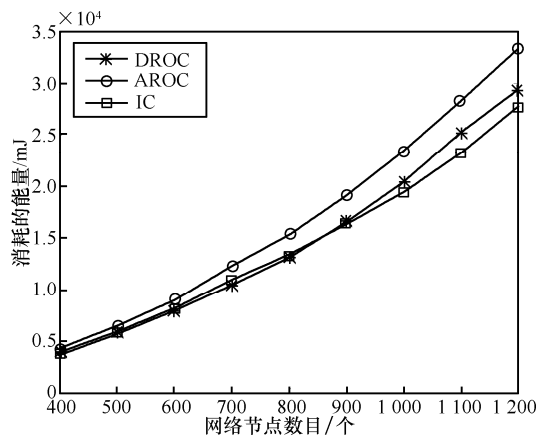


图 9 节点不失效时网络节点密度对能量消耗的影响

图 10 显示了失效节点数目对能量消耗的影响。对于 IC 协议，随着失效节点数目的增大，IC 协议的查询处理过程因查询节点失效而中断的概率增大。为了获得正确的查询结果，IC 协议需要反复重新执行，带来了大量的能量消耗。因而，IC 协议的能量消耗随着失效节点数目变大而急剧增大。对于 DROC 和 AROC 协议，当簇头节点失效时，这 2 种算法利用簇头节点的备份节点恢复查询处理过程，且只要簇头节点的所有备份节点不同时全部失效，查询处理过程能够继续进行，保证了算法的顽健性，因而随着失效节点数目的增大，DROC 和 AROC 协议消耗的能量较缓慢地增大且小于 IC 协议。另外，由于 DROC 协议的查询成功率高于 AROC 协议，且 DROC 协议的簇头节点数目少于 AROC 协议，根据式(7)可知，DROC 协议的能量消耗少于 AROC 协议。

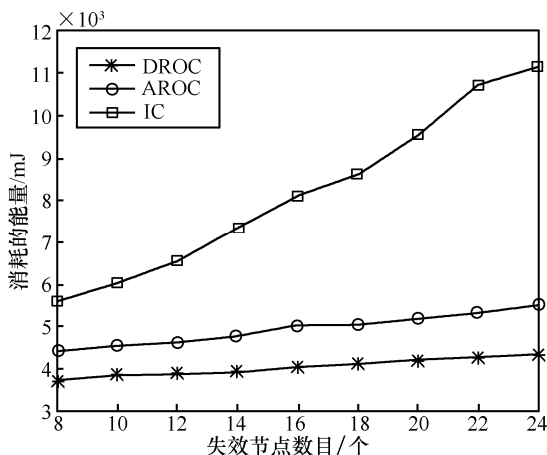


图 10 失效节点数目对能量消耗的影响



## 5 结束语

现有传感器网络环境下的  $K$  近邻查询处理算法没有综合考虑网络拓扑变化、节点失效等对查询处理的影响, 导致查询处理过程极易中断, 查询结果频繁丢失。本文提出了一种顽健的  $K$  近邻查询处理算法 ROC-KNN。它具有如下优点: 1) 当簇头节点失效时, 利用无线通信的广播特性, 簇头节点所在子区域中的一些节点可作为簇头节点的备份节点, 代替簇头节点继续查询处理过程, 避免了因节点失效带来的查询处理过程中断问题, 且保证了已计算出的部分查询结果无需重新计算, 减少了算法的能量消耗; 2) 通过合理地划分子区域和选择簇头节点, 降低了算法的能量消耗。实验结果表明, 在绝大多数情况下, ROC-KNN 在能量消耗、查询成功率方面均优于现有的算法。

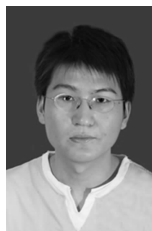
### 参考文献:

- [1] ROUSSOPOULOS N, KELLEY S, VINCENT F. Nearest neighbor queries[A]. Proc of the 1984 ACM SIGMOD International Conference on Management of data[C]. New York, 1995. 71-79
- [2] GUTTMAN A. R-trees: a dynamic index structure for spatial searching[A]. Proc of the 1984 ACM SIGMOD International Conference on Management of Data[C]. New Yorks, 1984.47-57.
- [3] DEMIRBAS M, FERHATOSMANOGLU H. Peer-to-peer spatial queries in sensor networks[A]. Proc of the IEEE International Conference on Peer-to-Peer Computing Peer-to-Peer Computing[C]. Washington, 2003. 32-39.
- [4] GOLDIN D, SONG M, KUTLU A, *et al.* Georouting and delta-gathering: efficient data propagation techniques for geosensor networks[A]. Proc of GeoSensor Networks Workshop[C]. Portland, Maine, 2003.
- [5] WINTER J, LEE W C. KPT: a dynamic knn query processing algorithm for location-aware sensor networks[A]. Proc of the 1st International Workshop on Data Management for Sensor Networks in Conjunction with VLDB 2004[C]. New York, 2004. 119 -124.
- [6] XU Y Q, FU T Y, LEE W C, *et al.* Itinerary-based techniques for processing  $K$  nearest neighbor queries in location-aware sensor networks[J]. Signal Processing, 2007, 87(12): 2861-2881.
- [7] YAO Y X, TANG X Y, LIM E P. Localized monitoring of KNN queries in wireless sensor networks[J]. The Very Large Database Journal, 2009, 18(1):99-117.
- [8] WU S H, CHUANG K T, CHEN C M, *et al.* DIKNN: an itinerary-based KNN query processing algorithm for mobile sensor networks[A]. Proc of the 23rd International Conference on Data

Engineering[C]. Washington, 2007. 456-465.

- [9] WU S H, CHUANG K T, CHEN C M, *et al.* Toward the optimal itinerary-based KNN query processing in mobile sensor networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(12): 1655-1668.
- [10] FU T Y, PENG W C, LEE W C. Optimizing parallel itineraries for knn query processing in wireless sensor networks[A]. Proc of the Sixteenth ACM Conference on Conference on Information and Knowledge Management[C]. New York, 2007. 391-400.
- [11] KARP B, KUNG H T. GPSR: greedy perimeter stateless routing for wireless networks[A]. Proc of the 6th annual International Conference on Mobile Computing and Networking[C]. New York, 2000. 243-254.
- [12] XU Y Q, LEE W C, XU J L, *et al.* Processing window queries in wireless sensor networks[A]. Proc of the 22rd International Conference on Data Engineering[C]. Washington, 2006. 70.
- [13] COMAN A, SANDER J, NASCIMENTO M A. Adaptive processing of historical spatial range queries in peer-to-peer sensor networks[J]. Distributed and Parallel Databases, 2007, 22(2-3):133-163.
- [14] RAPPAPORT T. Wireless Communications: Principles and Practice[M]. New Jersey, Prentice-Hall Press, 1996.

### 作者简介:



刘亮 (1985-), 男, 江西景德镇人, 南京航空航天大学博士生, 主要研究方向为传感器网络、流数据库。



秦小麟 (1953-), 男, 江苏南京人, 南京航空航天大学教授、博士生导师, 主要研究方向为安全数据库、时空数据库、分布式环境数据管理与安全等。



刘亚丽 (1980-), 女, 江苏徐州人, 南京航空航天大学博士生, 主要研究方向为物联网、信息安全。

李博涵 (1979-), 男, 黑龙江哈尔滨人, 南京航空航天大学博士后, 主要研究方向为传感器网络数据管理。