

无线传感器网络上的极值区域查询处理*

熊蜀光¹⁺, 李建中¹, 陈雷², 王新兵³

¹(哈尔滨工业大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

²(香港科技大学 计算机科学系, 香港)

³(上海交通大学 电气工程学院, 上海 200240)

Querying Peak Regions in Wireless Sensor Networks

XIONG Shu-Guang¹⁺, LI Jian-Zhong¹, CHEN Lei², WANG Xin-Bing³

¹(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

²(Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, China)

³(Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

+ Corresponding author: E-mail: n2xiong@gmail.com, http://db.cs.hit.edu.cn/p/shuguangxiong

Xiong SG, Li JZ, Chen L, Wang XB. Querying peak regions in wireless sensor networks. *Journal of Software*, 2010,21(8):1920–1935. <http://www.jos.org.cn/1000-9825/3710.htm>

Abstract: This paper proposes a query in wireless sensor networks: Peak Region Query (PRQ). Given the shape and size of the query region, i.e., a disk region with radius R , peak region query finds out a region with this shape in the network field, in which the aggregation value of the data of the sensors can be maximized. This paper first gives the definition of PRQ, and then proposes a centralized algorithm for the problem. Because the sensors have limited energy, a distributed approach EXQ (an algorithm for extreme value query processing) is proposed, which not only reduces the energy cost but also balances the workload of the sensors, so as to prolong the lifetime of the network. The basic idea is to divide the network field into overlapped sub-regions, compute a local result for each sub-region and aggregate these results to obtain the query answer. The paper compares the energy efficiency and load balance between EXQ and the centralized approach analytically and experimentally.

Key words: wireless sensor network; query processing; energy efficiency; extreme value; distributed algorithm

摘要: 提出了无线传感器网络上的一种查询——极值区域查询(peak region query,简称PRQ),即用户指定查询区域的大小和形状,例如半径为 R 的圆形区域,然后需要查询传感器网络中的某个区域,使得区域内传感器节点数据的某种聚集值最大.定义了极值区域查询的概念,并提出一种集中式算法以求解查询结果.由于传感器节点的能量有限,为了降低查询处理过程中的能耗,提出了分布式算法EXQ(an algorithm for extreme value query processing).与集中式算法相比,EXQ不但显著降低了能耗,而且使得每个传感器的能耗更加平均,从而延长了网络的使用寿命.EXQ的基本思想是,将整个网络划分为若干相互重叠的子区域,对每个子区域通过本地数据聚集得到一个本地结果,然后对这些结果再进行全局数据聚集从而得到查询结果.从理论和实验两方面分析和比较了集中式算法和EXQ的能耗

* Supported by the National Natural Science Foundation of China under Grant No.60533110 (国家自然科学基金); the National Basic Research Program of China under Grant No.2006CB303000 (国家重点基础研究发展计划(973))

Received 2008-09-26; Revised 2009-01-14, 2009-05-05; Accepted 2009-07-23

和节点负载分布.

关键词: 无线传感器网络;查询处理;能量有效;极值;分布式算法

中图法分类号: TP393 文献标识码: A

随着无线通信和计算机硬件技术的发展,无线传感器网络(wireless sensor network,简称 WSN)已被广泛应用于生产和生活,例如环境、健康、军事和交通状况监测等^[1-4].为了更好地发掘传感器网络的潜力和方便用户使用,学术界已经提出相关技术^[5,6],将传感器网络视为一个分布式数据库系统(data base management system,简称 DBMS),为数据管理提供逻辑意义上的抽象.用户只需提交 SQL 语言描述的查询即可得到结果,无须考虑数据如何产生和传输以及烧录在传感器节点上的程序如何工作.

按照查询中是否存在连接操作,传感器网络上的查询大致可以分为两类:一类是不带连接操作的查询,称为简单查询^[7-15];另一类查询中存在若干连接操作,称为复杂查询^[16-18].例如,图 1(a)所示的简单查询查找动物在矩形区域 (x_1, y_1, x_2, y_2) 内出现的平均次数,而图 1(b)所示的复杂查询^[18]查找下雨时动物是否有向某地聚集的事件发生.此时,每个传感器节点需要监测本地的空气湿度和动物数量,同时,传感器节点将自己的地理位置和数据获取时间添加到数据中.

| | |
|--|--|
| <pre>SELECT avg(A1.num) FROM Animal AS A1 WHERE A1.loc in (x1,y1,x2,y2)</pre> <p style="text-align: center;">(a)</p> | <pre>SELECT A1.humidity, A1.num, A1.time, A1.loc, A2.num, A2.time, A2.loc FROM Animal AS A1, Animal AS A2 WHERE A1.humidity>δ AND A2.num-A1.num>θ AND near(A1.loc,A2.loc,r) AND A2.time>A1.time AND A2.time-A1.time<h</pre> <p style="text-align: center;">(b)</p> |
|--|--|

Fig.1 Sample queries in WSN

图 1 传感器网络上的查询示例

这样,WSN 数据库中每一条数据具有如下的属性:(1) 地理位置;(2) 时间;(3) 空气湿度;(4) 动物数量.图 1(b)中的查询定义该事件为:如果数据库中存在两条数据 $A1$ 和 $A2$,满足如下的几个条件,则称下雨时,动物有从 $A1$ 向 $A2$ 聚集的事件发生:(1) $A1$ 的空气湿度大于 δ (即下雨时);(2) $A1$ 和 $A2$ 的地理位置相距不超过 r ;(3) $A2$ 的时间晚于 $A1$ 且位于 $A1$ 以后 h 分钟以内;(4) $A2$ 的动物数量比 $A1$ 的动物数量至少多 θ 个.

如上所述,与简单查询相比,复杂查询的优点之一是可以应用于 WSN 上的事件探测(event detection),其中,事件被定义为存在若干数据满足查询谓词(即满足给定的模式).当前的复杂查询需要用户指定查询变量.以文献[18]中的动物聚集事件查询为例,对于数据库中任意两条数据记录 $A1, A2$,查询谓词限定了两条数据的取值和时空相关性,且用户指定的查询变量集合 $S=\{A1, A2\}$.

然而在一些应用中,用户关心的是这样一类查询:是否存在若干数据 S 满足查询谓词,其中,查询谓词限定的是 S 中数据的地理位置位于某形状固定的区域内.注意,区域只是形状固定,而地理位置是不固定的.如图 2(a)所示(两个圆型区域,其取值分别为 3 和 1),限定 S 是半径为 R 的圆内的所有数据,则对 A 区域来说, S 中有 3 条数据,而 B 区域中 S 仅有 2 条数据.由于不同的区域导致查询变量集合 S 的大小可能产生变化,现有的针对复杂查询的处理方法不适用于这样的查询.下面列举该类查询的两个示例应用.

示例 1:动物学家利用传感器网络监测熊猫的活动^[19],需要探测整个被监测区域中熊猫出现最频繁的地点,以便在该处建立观察点进行观察和研究工作.假定观察点的观察区域是半径为 R 的圆,那么需要找到一个地点,使得在以该点为圆心的圆内,熊猫出现次数之和最大.由于此时查询谓词中限定的只是包含数据集合 S 的区域形状, S 的大小不能确定,因此现有方法不能处理此类查询.

由于传感器节点数据分布未知,事先固定 S 的大小而将此类查询转化为传统事件探测查询^[16-18]的方法仍

然无法满足需要.例如,当指定 $|S|=1$ 时,查询转化为求一个地点,使得该地点的传感器监测到的熊猫出现次数最大.由于熊猫的数量非常有限,且只有当熊猫出现在传感器节点附近时,传感器节点才能发现并记录,这导致每个节点获取的数据值即熊猫出现的次数可能会很小.如图 2(a)所示,在拥有 8 个传感器节点的网络中,熊猫在某地点的出现次数最多为 1,而这样的地点一共有 5 个.如果依据熊猫的最大出现次数来确定观察点位置,则可选的地点有很多,然而每个可选地点处,熊猫的出现次数却很少.因此,传感器监测的熊猫出现次数最大值不适合反映熊猫出现最频繁的区域,因此固定 S 大小的方法不可取.

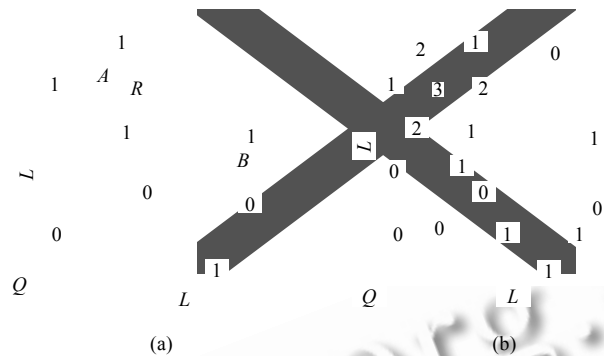


Fig.2 A sample sensor network

图 2 传感器网络示例

示例 2:利用传感器网络监测山体滑坡^[20],需要定期获得传感器节点的位移,如果位移超出允许值则标志山体滑坡事件的发生.然而,因为硬件的损坏、通信错误或者环境的干扰^[21],某些节点上可能会存在超出允许值的位移数据,而这些数据并非由山体滑坡事件产生.如果通过查询谓词约束查询中数据集合 S 产生的相对位置,例如半径为 R 的圆,则可以通过查看 S 中元素的均值是否超出允许值来判定山体滑坡事件的发生.由于用户可以灵活地指定 R 的大小以消除错误数据和外界干扰带来的影响(例如,假设过往汽车行驶的震动范围为 10m,则可指定 $R=100m$,这样,受震动影响而位移超过允许值的节点只占判定区域节点的很小一部分,所得平均位移将仍然低于允许值),利用查询谓词对 S 产生的相对位置进行约束是具有实用价值的.与示例 1 相似,事先固定 S 的大小而将此类查询转化为传统事件探测查询的方法仍然无法满足需要.

以上两个例子说明,在一些应用中存在着这样的查询,即用户指定感兴趣区域的大小和形状(例如半径为 R 的圆),但是不指定区域的地理位置.需要查询的是区域的地理位置,使得区域内传感器节点数据的某种聚集值最大,或者查询是否存在这样的区域,使得该聚集值大于给定阈值,我们称这样的查询为极值区域查询.以求和聚集为例,图 2(a)中以 A 点为圆心、 R 为半径的区域内有 3 个传感器,其读数均为 1,则区域 A 的聚集值为 3.因为在所有半径为 R 的圆中 A 的聚集值最大,所以 A 点即是一个所求位置.

与区域查询相比,极值区域查询给定的是用户关心的区域大小和形状,返回的是某个区域的地理位置及其上的数据聚集值,该区域的大小和形状与用户给定的相同.而区域查询给定的是大小、形状和地理位置都确定的某区域,返回的是该区域上的数据聚集值,无论是查询表达还是查询结果都有着本质的区别.

现有的极值查询处理方法^[7,8]同样不能适用于极值区域查询.这是因为极值查询返回的是所有传感器读数中的极值,而极值区域查询返回的是查询区域的地理位置以及此区域内数据的聚集值.即极值区域查询的基本对象从传感器数据变化到包含若干个传感器的固定形状区域,二者的查询结果完全不同.同样,现有的其他网内聚集技术^[9,12-14,22-25]也不适用于极值区域查询.

极值区域查询带来以下两个方面的挑战:首先,对于整个网络内的任何一个点 P ,都存在一个以 P 为圆心的用户指定区域,因此这样的区域会有无数多个.在 sink 能够获取整个网络的节点数据和地理位置的前提下,如何设计高效的算法以从这无数个区域的聚集值中找到极值成为一项挑战;其次,目前传感器节点通常由自身携带

的电池供电,并且在实际的环境中很难为电池充电^[1,2,8].因此,是否存在能量有效的分布式算法,能够返回极值区域查询的结果,同时尽可能地节省节点的能量以延长网络使用寿命^[8,12,18]成为另一项挑战.

针对无线传感器网络上的极值区域查询,本文首先提出了一种需要 $O(n^2 \log(n))$ 时间的集中式算法,其中, n 是传感器网络中的节点个数.为了节省查询处理过程中传感器节点的能耗,随后提出了基于分层处理机制的分布式算法 EXQ(an algorithm for extreme value query processing).本文的贡献列举如下:

(1) 首先提出了无线传感器网络上具有应用前景的极值区域查询概念,即用户事先指定查询区域的大小和形状,查询传感器网络中的某个区域,使得区域内传感器节点数据的某种聚集值最大.

(2) 将极值区域查询问题转化为一个覆盖问题,并提出一种 $O(n^2 \log(n))$ 时间的集中式算法,其中, n 是传感器网络中的节点个数,集中式算法需要将所有节点的数据和地理位置信息上传到 sink.

(3) 提出了高效的分布式算法 EXQ,其采用一种分层处理的机制:首先将整个网络划分为若干重叠的子区域,然后对每一个子区域执行一次本地数据聚集,最后执行全局数据聚集以得到查询结果.给定网络监测区域边长 L 和查询半径 $R(R \ll L)$ 且固定传感器密度时,与集中式算法相比,EXQ 所需的总能耗从 $O(L^3)$ 减少到 $O(L^2 R)$,单个传感器节点耗费的最大能量从 $O(L^2)$ 减少到 $O(R^2)$.

(4) 提出了近似最优子区域大小的计算方法以最小化 EXQ 所需能耗,并针对 3 种不同形状的子区域划分策略(正方形、三角形、六边形)进行分析.

(5) 运行大量的模拟实验,并统计网络的总能量开销和传感器节点的负载分布.实验结果验证了在不同的网络规模和参数下,EXQ 算法比集中式算法更节省能耗,且能够使得节点负载更加平衡.实验还验证了确定最佳子区域大小方法的有效性和不同子区域划分方式下网络的能耗和节点负载.

本文第 1 节简要回顾近年来的相关工作,第 2 节介绍网络模型和问题的形式化定义,第 3 节描述算法并给出理论分析和结果,第 4 节给出包括能量开销和负载分布的实验结果和相关分析,第 5 节总结全文并讨论本文的后续工作.

1 相关工作

无线传感器网络上的数据库原型系统 TinyDB^[5]和 Cougar^[6]等使得用户能够通过查询语句获取网络中节点的数据和相关结果,并且数据在网内的传输和处理均对用户透明.因为传感器节点的能量有限,查询处理过程中首要考虑的是算法的能量有效性.文献[26]采用统计建模技术使得查询结果更有意义,并且提出带概率保证的近似查询方法,从而显著降低了计算和通信开销.由于查询处理过程中是否存在连接(join)等较复杂的操作对网络能耗影响很大,据此可以大致将目前的查询处理算法分为两类:简单查询(不带 Join 操作的查询)^[7-9,10-15]和复杂查询(带 Join 操作的查询)^[16-18].

对简单查询的研究主要集中在数据聚集操作上,其定义为将数据从多个传感器节点上聚合到一起(包括求 sum, min/max, avg),并消除重复的数据传输以及向 sink 节点提供融合信息的处理过程^[10].定向扩散(directed diffusion)方法^[15]是较早关注高效聚集策略的研究之一.根据定向扩散方法,数据在节点本地进行聚集,并采用路径选择、本地数据缓存等策略来减小数据通信量.基于聚簇的聚集方法包括 LEACH-C^[11], HEED^[12]和文献[13]中提出的方法等,即将网络划分为若干簇,在每一簇的簇头节点处进行聚集.此类方法的不足之一在于簇头节点的选取受到较大的限制,以 HEED 为例,簇头节点离该簇中的其他节点仅有一跳的距离.与基于聚簇的方法相比,文献[23]提出了一种平衡节点负载的方法,通过计算每个节点的最优数据传输概率来延长网络寿命,避免了簇头节点的重复选取.针对稀疏网络中的数据聚集问题, Gao 等人提出了一种采用稀疏聚集树的策略^[24].

数据聚集中的极值查询已经受到广泛关注^[7-9]. Silberstein 等人^[8]针对连续极值查询提出一种策略,利用历史数据设置过滤器以过滤掉不可能成为极值的数据,从而减少传输开销. Olston 等人^[7]提出的设置缓存近似值精度的策略也可用于连续极值查询.极值查询问题的一个扩展即是在网络中查询前 k 个极值.文献[9]提出了针对该扩展问题的一种近似算法.然而这些方法仅限于处理极值查询,而不能应用到极值区域查询处理中.

带 Join 操作的复杂查询往往应用于事件探测^[16-18].用户在查询中给定变量集合和限定这些变量的谓词,如

果网络中存在满足谓词约束的数据集合,则可判定某类事件的发生.根据查询谓词与数据产生的地理位置是否相关,复杂查询也可分为两类.与现有复杂查询不同,极值区域查询需要用户在查询中给定查询区域形状和聚集函数,而不是变量集合及其谓词约束.由于在极值区域查询中查询区域内的数据集合大小并不固定,而现有复杂查询的谓词只能约束用户给定的变量集合,因而其不适用于极值区域查询.

针对传感器网络的覆盖问题,Chi-Fu Huang 等人^[27]提出了一种高效的算法以求解 k -UC 和 k -NC 等覆盖问题,其中, n 是传感器节点的数量. k -UC 问题即判定是否网络中的每一处都被至少 k 个传感器所覆盖(探测到). Anthony Man-Cho 等人^[28]针对 k -UC 问题提出了另一种基于 Voronoi 图的算法,其计算复杂度为 $O(n\log(n)+nk^2)$. 因为该算法与给定的 k 值相关,因此并不适用于转化为覆盖问题后的极值区域查询处理.

2 网络模型和问题描述

一个无线传感器网络 $\{s_1, s_2, \dots, s_n\}$ 由 n 个静止不动的传感器节点组成,其中, s_i 和 $v_i (1 \leq i \leq n)$ 分别表示一个传感器节点和它所感知的数据(读数).所有的传感器节点随机分布在边长为 L 的正方形监测区域内.整个网络是连通的,并且 sink 节点(即查询节点 Q)位于监测区域的左下角(如图 2 所示).此外,假定传感器网络有能力进行网内数据处理^[14,25],并且每个传感器的位置信息可以通过现有的定位算法^[4,29]获得.

2.1 问题描述

定义 1(圆型区域的取值). 以点 A 为圆心、 R 为半径的圆型区域 A 的取值 $V_A = f(v_{i1}, v_{i2}, \dots, v_{im})$. 其中 f 是一个聚集函数(例如求和), $v_{i1}, v_{i2}, \dots, v_{im}$ 是所有位于该区域内的 m 个传感器节点的读数.为便于表述,以 A 为圆心的圆型区域记为区域 A .以图 2(a)为例,区域 A 包含了 3 个传感器节点,其读数均为 1,则该区域的取值 $V_A=3$,而区域 B 包含 2 个读数 1 和 0,因此 $V_B=1$.

定义 2(极值区域查询). 给定查询半径 R ,极值区域查询将找出监测区域内所有半径为 R 的圆型区域取值中的最大/最小值,记极值区域查询的结果为 $Ext(R)$.例如,图 2(b)中的最大值区域查询结果是 3,也即圆心位于阴影区域的任意圆型区域的取值(传感器圆将整个网络平面划分为若干分割区域,每个分割区域的取值被标记出).

传统的极值查询也可以由与定义 2 相似的定义给出,只需将定义 2 中的“圆型区域取值”替换为“传感器读数”即可.事实上,极值查询是极值区域查询的特例.当查询半径 R 足够小,以至于在任意半径为 R 的圆型区域中最多只包含 1 个传感器节点时,极值区域查询的结果即是最大/最小的传感器读数,极值区域查询则退化为极值查询.

基于以上定义,本文研究的问题即是如何高效地获得极值区域查询结果,以下简称 PRQ 问题(peak region query problem).这里,“高效”有两重含义:一是查询处理过程所需能量开销相对较少;二是传感器的负载相对平衡,即缓解部分传感器承担繁重传输的状况以延长这些传感器节点的寿命,从而延长整个网络的使用寿命.

为方便起见,下文将以最大值代替极值进行讨论,针对最小值情况的讨论和策略与最大值情况将完全一致.下一节将给出衡量查询处理过程中传输代价的基本模型,以用于本文各种算法的分析.

2.2 能量开销估计

极值区域查询处理过程的能量开销由计算开销和通信开销组成.因为在无线通信上每传输一位字节的能量开销通常不低于执行 1 000 个 CPU 指令的能量开销^[2],并且极值区域查询处理的计算复杂度不会超过 $O(n^2 \log(n))$ (参见本文相关分析).其中, n 为传感器节点数,则即使在一个拥有 1 万个传感器节点的大规模网络中,节点的平均计算开销为 $O(n \log(n))$,即不超过 140 000 次基本操作,仅相当于传输 140bit 即 18 个字节,甚至不及一个数据包的有效负载.可见在实际应用中,计算所需的能量开销要远小于通信开销,因此本文将忽略计算开销,只考虑通信开销.

根据 Silberstein 等人的工作^[9],无线传感器网络中传输一个数据包的能量开销 $E_p = E_1 + xE_2$,其中, E_1 是固定的能量开销部分, E_2 是每传输 1 字节的有效数据所需的能量开销,而 x 是被传输的有效字节数.由于当前技术的限

制,每一个数据包传输的最大有效字节数为一个常数,因而我们可以将传输一个数据包的能量开销 E_p 看作是一个常数.这样,在极值区域查询处理过程中,总的能量开销可以被估计为查询处理过程中所传输数据包的个数.

基于以上讨论,从传感器节点 s_i 到 s_j 传输一个数据包的能量开销则可以被估计为 $h_{ij}E_p$,其中, h_{ij} 是将数据包从 s_i 路由到 s_j 的跳数.设 s_i 到 s_j 之间的距离为 d_{ij} ,则距离 d_{ij} 与跳数 h_{ij} 之间的关系满足 $h_{ij}=\alpha d_{ij}$,其中,系数 α 是与路由策略相关的一个常数,不同的路由策略对应不同的 α 值^[29].这样,从 s_i 节点向相距为 d_{ij} 的 s_j 节点传送一个数据包所需的能量可以被估计为

$$E_{ij}=\alpha d_{ij} \cdot E_p \quad (1)$$

3 极值区域查询处理算法

直观上看,因为一个半径为 R 的圆型区域的圆心可以位于传感器网络中的任意位置,所以这样的圆型区域将会有无数多个,从而无法一一列举这些区域的取值并从中找出极值.然而,我们可以采用以下方法将 PRQ 问题转化为一个覆盖问题.为方便起见,先给出几个定义.

定义 3(传感器圆). 以传感器节点 s_i 为圆心、以查询半径 R 为半径的圆称为一个传感器圆,记为圆 s_i .一个传感器圆的取值即是该传感器节点的读数 v_i .如图 2(b)所示的圆都是传感器圆.

定义 4(分割区域). 网络中的 n 个传感器圆将平面划分成的若干区域称为分割区域.一个分割区域的取值为 $f(v_{i_1}, v_{i_2}, \dots, v_{i_m})$.其中, f 是与定义 1 中的 f 相同的聚集函数, $v_{i_1}, v_{i_2}, \dots, v_{i_m}$ 是所有覆盖该分割区域的传感器圆的取值.如果没有任何传感器圆覆盖该分割区域,则该区域取值为 0.记所有分割区域取值中的最大值为 $Cut(R)$.例如,图 2(b)中的所有数字即表示分割区域的取值,这里,函数 f 为求和.不难看出, $Cut(R)=3$.

定义 5(边界区域). 传感器圆 s_i 的圆周被其他圆分割为若干段,则每一段圆周必然与两个分割区域相连,其中能够被圆 s_i 覆盖的区域称为圆 s_i 的内边界区域,不能被圆 s_i 覆盖的区域称为圆 s_i 的外边界区域.传感器圆 s_i 的所有内外边界区域的集合称为圆 s_i 的边界区域集合.例如,图 3(a)中的区域 $R_1 \sim R_5$ 均为圆 s_1 的内边界区域, $R_6 \sim R_{10}$ 均为圆 s_1 的外边界区域.因此,圆 s_1 的边界区域集合为 $\{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}\}$.

一个边界区域的取值为 $f(v_{i_1}, v_{i_2}, \dots, v_{i_m})$.其中, f 是与定义 1 中的 f 相同的聚集函数, $v_{i_1}, v_{i_2}, \dots, v_{i_m}$ 是所有覆盖该边界区域的传感器圆的取值.记所有边界区域取值中的最大值为 $Per(R)$.

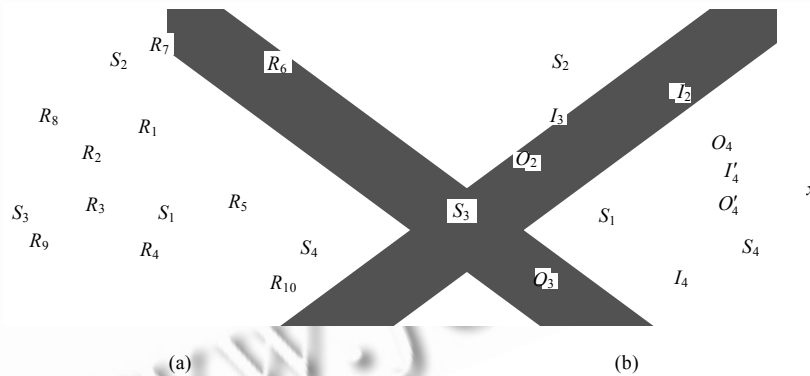


Fig.3 Calculating the values of the perimeter regions of a circle

图 3 计算一个传感器圆的边界区域取值

基于以上定义,可以得出如下结论:

结论 1. $Ext(R)=Cut(R)=Per(R)$.

结论 1 说明,最大值区域查询结果 $Ext(R)$ 即是所有分割区域的最大值 $Cut(R)$,同时也等于所有边界区域的最大值 $Per(R)$.因为所有的分割区域都是由传感器圆分割而成,任意的分割区域至少与 1 个传感器圆的圆周相连,因此一个分割区域必定是某圆的边界区域.同时,根据分割区域的定义,某圆的边界区域必然是分割区域,因此,

所有分割区域的集合等于所有边界区域的集合,故有 $Cut(R)=Per(R)$.

并且,当一个圆型区域 R_i 的圆心位于某个传感器圆 s_j 内时,表明 R_i 的圆心和传感器节点 s_j 之间的距离小于查询半径 R ,则 R_i 包含 s_j .同理,当 R_i 的圆心位于 m 个圆共同覆盖的分割区域内时,则说明 R_i 恰好包含这 m 个传感器节点,并且无论 R_i 的圆心在该分割区域内如何移动,它包含的传感器节点集合显然始终等于覆盖该分割区域的传感器圆对应的节点集合.因此,用以计算该圆型区域取值和该分割区域取值的传感器读数集合相同,因此,该分割区域的取值等于所有圆心位于该区域的圆型区域的取值,故有 $Ext(R)=Cut(R)$.

结论 1 表明,要得到极值区域查询结果,只需要求出所有分割区域的取值,并从中求出 $Per(R)$ 即可.因此,问题被转化为求被 n 个传感器圆划分的所有分割区域的取值.因为分割区域的取值只取决于覆盖它的传感器圆的读数,所以该问题可以被看作一个覆盖问题.

尽管转化后的覆盖问题与 k -UC 问题相似^[27],但是存在以下两点主要的不同,使得文献[27]中针对 k -UC 问题的算法不能直接应用于 PRQ 问题:首先,PRQ 问题是要找出所有圆型区域的最大/最小值,而 k -UC 问题是要回答监测区域的每一个点是否都能被 k -覆盖,两者所求的结果完全不同;其次, k -UC 问题中的感知半径固定不变,而 PRQ 问题中的查询半径 R 是可变的.因此,针对 PRQ 问题的分布式算法可以利用 R 来动态地划分网络,从而减小能耗.接下来将给出集中式算法和分布式的 EXQ 算法,并分析两种算法的计算开销、能量开销及负载平衡.

3.1 集中式算法

集中式算法需要全部传感器的地理位置和读数信息,并且只运行于 sink 节点上.为求出 $Per(R)$,集中式算法将求出每一个传感器圆的所有边界区域取值,然后在这些取值中再求最大/最小值,所得结果即是 $Per(R)$.

为了求出一个传感器圆 s_i 的所有边界区域取值,可采用如下算法:首先,求出其他传感器圆与该圆的所有交点,获得交点集合.如果某个传感器圆 s_j 与 s_i 相交,则交点必然有两个,规定按逆时针方向进入相交区域的交点为入点,记为 I_{ij} ,按逆时针方向离开相交区域的交点为出点,记为 O_{ij} .图 3(b)中(将圆周上的所有入点和出点按角度升序排列,然后计算所有相邻两节点对应的边界区域的取值),圆 s_1 的交点集合为 $\{I_2, I_3, I_4, O_2, O_3, O_4\}$.

然后计算出每一个交点的角度,并将交点集合按照交点的角度升序排列(如果角度相同,则入点位于出点之前).交点 I_{ij} 的夹角的起始边是以圆心 s_i 为起点、沿水平向右方向的射线,终止边是射线 $s_i I_{ij}$.所有角度的取值范围均是 $[0, 2\pi]$,并且为使接下来的计算方便,如果圆 s_j 在圆 s_i 上入点角度大于出点角度,则新增加一个角度为 0 的入点和角度为 2π 的出点.例如,图 3(b)中,圆 s_4 在圆 s_1 上的交点为 I_4 和 O_4 ,因为 I_4 的角度大于 O_4 的角度,则增加一个角度为 0 的入点 I'_4 和一个角度为 2π 的出点 O'_4 .这样,圆 s_1 圆周上的所有交点按角度升序排列后为 $\{I'_4, O_4, I_2, I_3, O_2, O_3, I_4, O'_4\}$.

接着,遍历排序后的交点集合,在遍历到交点 P_k 时维护变量 m, RIN_k 和 $ROUT_k$.记圆 s_i 的圆周上从 P_k 逆时针方向到 P_{k+1} 的部分为 $\widehat{P_k P_{k+1}}$,则变量 m 的含义是除圆 s_i 外,覆盖 $\widehat{P_k P_{k+1}}$ 的圆的个数. RIN_k 和 $ROUT_k$ 分别是与 $\widehat{P_k P_{k+1}}$ 相连的圆 s_j 的内边界区域和外边界区域的取值.如果 P_k 是某个圆 s_j 在圆 s_i 上的入点,则有 $m=m+1, RIN_k=(RIN_{k-1} \cdot m + v_j)/(m+1)$ 以及 $ROUT_k=[ROUT_{k-1} \cdot (m-1) + v_j]/m$,其中, v_j 是圆 s_j 对应的传感器读数.如果 P_k 是某个圆 s_j 在圆 s_i 上的出点,则有 $m=m-1, RIN_k=[RIN_{k-1} \cdot (m+2) - v_j]/(m+1)$ 以及 $ROUT_k=[ROUT_{k-1} \cdot (m+1) - v_j]/m$.

假定函数 f 的计算结果是输入数据的平均值,并且查询所求的是最大值,我们有以下算法:

极值区域查询集中式算法.

Input: 查询半径 R, n 个传感器圆 C_i 的圆心坐标 (x_i, y_i) 和读数 $v_i (1 \leq i \leq n)$.

Output: 边界区域取值的最大值 $Per(R)$.

- 1: $Per(R)=0$; //初始化 $Per(R)$
- 2: for $i=1$ to n do
- 3: for $j=1$ to n do
- 4: if 传感器圆 C_i 和 C_j 相交 and $i \neq j$ then
- 5: 求出 C_j 在 C_i 圆周上的入点 I_{ij} 和出点 O_{ij} 及其对应角度;
- 6: $S_i=S_i \cup \{I_{ij}, O_{ij}\}$; //将 I_{ij} 和 O_{ij} 添加到交点集合 S_i

```

7:           if  $I_{ij}$  的角度  $> O_{ij}$  的角度 then
8:                $S_i = S_i \cup \{ I'_{ij}, O'_{ij} \}$ , 其中,  $I'_{ij}$  点的角度为 0,  $O'_{ij}$  点的角度为  $2\pi$ ;
9:           endif
10:        endif
11:    next
12:    将  $S_i$  中元素按角度升序排序;
13:    令  $m=0$ ,  $RIN=v_i$ ,  $ROUT=0$ ; //  $m$  为当前参与计算的其他传感器圆的个数
14:    for  $k=1$  to  $|S_i|$  do
15:        获得  $S_i$  中第  $k$  个元素  $P_{ij}$ ;
16:        if  $P_{ij}$  是圆  $C_j$  的入点 then
17:             $m=m+1$ ;
18:             $RIN=(RIN \cdot m + v_j)/(m+1)$ ;
19:             $ROUT=[ROUT \cdot (m-1) + v_j]/m$ ;
20:        else
21:             $m=m-1$ ;
22:             $RIN=[RIN \cdot (m+2) - v_j]/(m+1)$ ;
23:             $ROUT=[ROUT \cdot (m+1) - v_j]/m$ ;
24:        endif
25:        if  $RIN > Per(R)$  or  $ROUT > Per(R)$  then
26:             $Per(R) = \max\{RIN, ROUT\}$ ; //更新  $Per(R)$ 
27:        endif
28:    next
29: next

```

计算复杂性分析. 因为该算法是集中式算法, 所以只考虑在 sink 上进行计算的时间复杂度. 第 1 行计算所需时间为 $O(1)$; 从第 2 行到第 29 行需要进行 $O(n)$ 次循环, 以计算每一个传感器圆 s_i 的所有边界区域取值; 对每一个圆 s_i , 从第 3 行到第 11 行需要 $O(n)$ 次循环以计算其他圆在 s_i 上的所有交点, 第 12 行的排序需要 $O(n \log(n))$ 次计算, 从第 14 行到第 28 行需要 $O(n)$ 次循环, 以计算圆 s_i 每个边界区域的取值. 因此, 该算法的时间复杂度为 $O(1) + O(n)[O(n) + O(n \log(n)) + O(n)] = O(n^2 \log(n))$.

能量开销分析. 考虑平面上长为 dx 宽为 dy 的矩形区域, 以 sink 为原点, 设该区域的中心坐标为 (x, y) , 则当 dx 和 dy 足够小时, 可认为该区域上的传感器到 sink 的距离均为 $\sqrt{x^2 + y^2}$. 并且因为传感器的位置服从随机分布, 可设网络中的传感器密度为 ρ , 则该区域上的传感器数量为 $\rho dx dy$. 根据前面的能量开销估计可知, 该区域中的一个传感器向 sink 发送一个数据包的能量开销为 $\alpha \sqrt{x^2 + y^2}$, 则该区域中所有传感器都向 sink 发送一个数据包的能量开销为 $\rho \alpha \sqrt{x^2 + y^2} dx dy$. 因为传感器网络上一个数据包能够携带的用户数据通常为 20 字节~40 字节, 所以集中式算法所需的每个传感器节点的位置和读数信息完全能够被封装在一个数据包中, 则针对平面上长为 dx 、宽为 dy 的矩形区域, 集中式算法的能量开销也为 $\rho \alpha \sqrt{x^2 + y^2} dx dy$. 整个网络的能量开销如下式所示:

$$E_c = \iint_{\Gamma} \rho \alpha \sqrt{x^2 + y^2} dx dy = \frac{1}{3} \left[\ln(\sqrt{2} + 1) + \sqrt{2} \right] \rho \alpha L^3 \quad (2)$$

其中, 被积区域 Γ 即是边长为 L 的正方形区域. 公式(2)说明, 在传感器密度和路由算法确定的情况下, 集中式算法的能量开销与监测区域边长的三次方成正比.

负载均衡分析. 在集中式算法中, 越靠近 sink 的传感器节点担负着越重的数据转发任务, 因此消耗的能量就越多. 设传感器的通信半径为 r , 则距离 sink 1 跳距离内的传感器个数大约为 $1/4 \rho \pi r^2$. 因为这些传感器将转发整

个网络产生的 n 个数据包,那么距离 sink 1 跳距离内的传感器平均转发数据包的个数为 $n/(1/4\rho\pi r^2)=4L^2/(\pi r^2)$. 可见它们的能量开销与网络区域边长 L 的平方成正比.

3.2 分布式算法EXQ

总的来说,EXQ 分为 3 个阶段:子区域划分、本地数据聚集和全局数据聚集.在子区域划分阶段,整个监测区域被划分为若干边长为 $2C$ 的方格,每个方格都与四周的方格相连, C 为与 R 相关的一个常数,如何确定 C 的值以减小 EXQ 的能量开销将在后文中讨论.图 4(a)以虚线显示了划分出的每一个方格,图中灰色区域是方格划分的一个子区域,由一个方格和围绕它的宽为 R 的带状区域组成.该带状区域包括方格四周的 4 个长为 $2C$ 、宽为 R 的矩形以及方格的四角上的 4 个半径为 R 的 $1/4$ 圆.

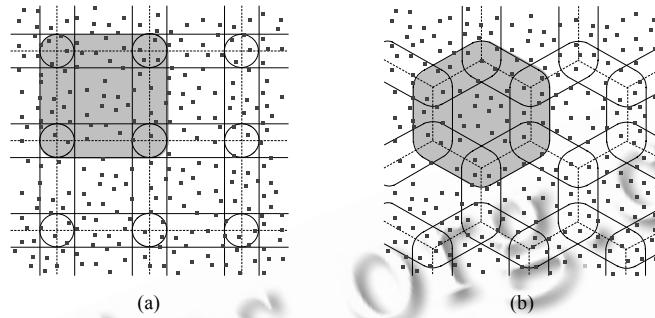


Fig.4 Square division and hexagon division for the network

图 4 对网络的方格划分和六边形划分

因为方格与方格之间是紧密相连的,所以子区域之间是互相重叠的.事实上,除了这种方格划分方式以外还有其他划分方式,本节将以方格划分为例进行讨论和分析,其他划分方式将在下一节加以讨论.

然后,对于每一个子区域,指定离区域中心最近的一个传感器节点为该区域的头节点.如果 sink 节点知道网络中所有节点的地理位置信息,它就可以通过方格的大小直接计算出每个子区域的头节点是哪一个传感器节点,然后,sink 将发送一个指定消息给每一个头节点,指定消息里携带该子区域的大小(包括方格边长 $2C$ 和查询半径 R)和位置等相关信息.

在本地数据聚集阶段,当一个头节点收到指定消息后,它立刻广播该消息给邻居节点.当一个传感器(非头节点)收到一条指定消息时,它同样广播该消息给它的邻居节点,并且判断自己是否位于此指定消息对应的子区域内.如果是,则向该指定消息对应的头节点传回自己的传感器读数.这样,每一个头节点都将获得它对应子区域内所有节点的传感器读数.接下来,头节点将进行本地数据聚集操作,即根据这些读数计算出该子区域的本地查询结果.本地数据聚集所用的算法与集中式算法相同,但需要注意的是,本地数据聚集只考虑在该子区域对应方格内的传感器圆,而不是在该子区域内的所有传感器圆.

在全局数据聚集阶段,每个头节点都将自己的本地查询结果发送给 sink,然后由 sink 计算出这些查询结果的极值,即最后的查询结果 $Ext(R)$.为了节省能量开销,如果一个节点在此阶段收到多个本地查询结果,则只需转发这些结果的极值即可.

EXQ 的正确性由以下定理保证:

定理 1. EXQ 的计算结果是极值区域查询的结果 $Ext(R)$.

证明:我们首先证明对于每一个子区域,头节点能够计算出圆心位于该方格内的所有传感器圆的边界区域的极值.考察任意圆心位于该方格内的传感器圆 s_i ,如果另一个传感器圆 s_j 与圆 s_i 相交,那么点 s_j 必定位于该方格对应的子区域内.否则, s_j 位于子区域外,同时,圆 s_j 的一部分位于方格内.任取圆 s_j 在方格内的一点 P ,则 P 到 s_j 的距离大于 R ,这与圆 s_j 的半径为 R 相矛盾.因此, s_j 必定位于该子区域内.而该子区域所有节点的信息必定被头节点获得,因此,头节点能够正确计算出所有圆心位于该方格内的所有传感器圆的边界区域取值,从而计算出这

些边界区域的极值.

其次,因为 EXQ 将整个监测区域分为若干方格,所以每个传感器节点必定位于某个方格(如果恰好位于某些方格的分界线上,则可以采用固定策略加以区分),因而任何传感器圆边界区域的取值都将被至少 1 个头节点计算得到.因此,从所有头节点计算出的本地查询结果中选出的最大/最小值必然是整个网络中边界区域的最大/最小值.故在全局数据聚集阶段结束后,EXQ 得到的值是整个网络中所有边界区域的极值 $Per(R)$,也即极值区域查询的结果 $Ext(R)$. \square

能量开销分析.因为每个方格的边长为 $2C$,方格和子区域的个数均为 $[L/(2C)]^2$,则 EXQ 在本地数据聚集阶段的能量开销为

$$E_d = \frac{L^2}{C^2} \iint_A \rho \alpha \sqrt{x^2 + y^2} dx dy \quad (3)$$

公式(3)中,被积区域 A 是 $1/4$ 个子区域,如图 5(b)中的灰色区域所示,坐标原点即是子区域的中心.对公式(3)求解,并令 $F = \sqrt{2C^2 + 2CR + R^2}$,有

$$E_d = \frac{\rho \alpha L^2}{3C^2} \left[(C+R)^3 \ln \left(\frac{C+F}{C+R} \right) + F(C+R)C - \left(2 \arccos \frac{C+R}{F} - \frac{\pi}{2} \right) F^3 \right] \quad (4)$$

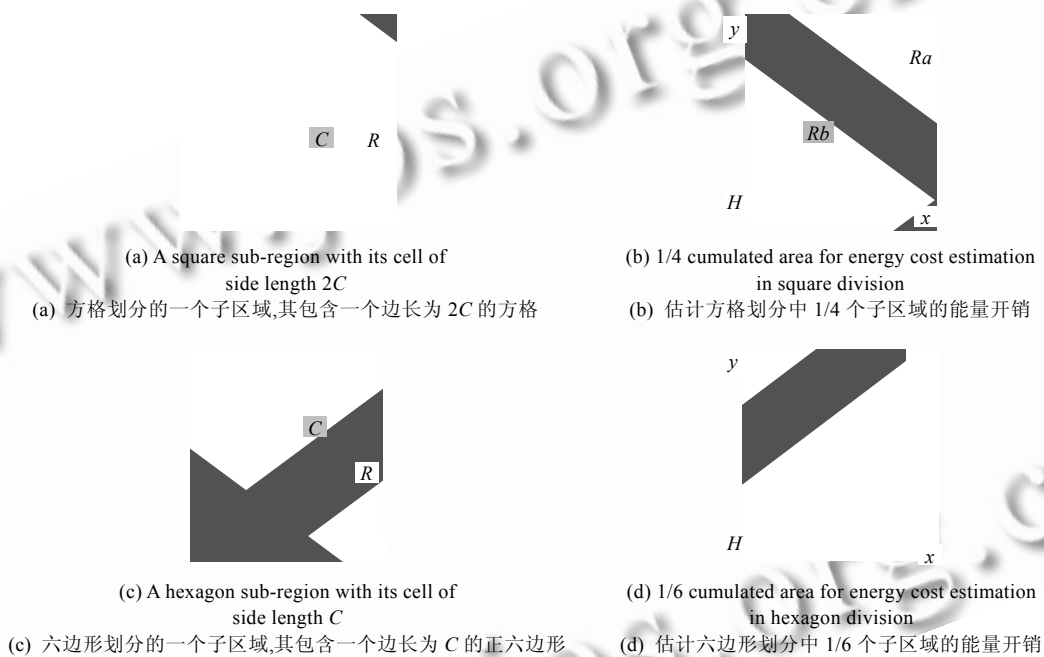


Fig.5 Energy cost estimation for square and hexagon division

图 5 估计方格划分和六边形划分的能量开销

EXQ 在子区域划分阶段需要 sink 向每个头节点发送一个指定消息包,在全局数据聚集阶段需要每个头节点向 sink 传回一个本地结果包,都只有头节点和 sink 之间通过多跳路由进行通信.因此,参与通信的节点个数均远小于本地数据聚集阶段,而每个节点产生的数据量与本地数据聚集阶段相同,因此,EXQ 在子区域划分阶段和全局数据聚集阶段的能量开销远小于本地数据聚集阶段的能量开销,因而完全可以利用公式(4)来估计 EXQ 在整个查询处理过程中的能量开销.

由于公式(4)中的 EXQ 能量开销与方格的边长 $2C$ 密切相关,下面将讨论如何确定参数 C 的取值,以使 EXQ 的能量开销最小.将公式(4)看作是 C 的函数,令 $d(E_d)/d(C)=0$,则可求出使 E_d 取最小值的 C .然而,由于对公式(4)右部求导后的表达式过于复杂,我们将采用另一个函数来近似 E_d .如图 4(b)所示,令边长为 $(C+R)$ 的正方形区域

的面积为 R_s , 白色区域的面积为 R_a , 灰色区域的面积为 R_b , 则在 EXQ 查询处理中, R_b 区域上的传感器节点的能量开销为 $E_{Rb} = E_{Rs} - E_{Ra}$.

令 $\delta_1 = \left[\ln(\sqrt{2} + 1) + \sqrt{2} \right] / 3$, $\delta_2 = \sqrt{2}(1 - \pi/4)$, 因为 R_a 区域中的任意点到原点 H 的距离在 $\left[\sqrt{2}C + R, \sqrt{2}(C + R) \right]$ 范围内, 故有 $E_{Rb} > \delta_1 \rho \alpha (C + R)^3 - \delta_2 \rho \alpha (C + R) R^2$. 因此, E_d 满足:

$$E_d > \delta_1 \rho \alpha L^2 (C + R)^3 C^{-2} - \delta_2 \rho \alpha L^2 R^2 (C + R) C^{-2}.$$

令 $\delta = \frac{1}{3} \left[\ln(\sqrt{2} + 1) + \sqrt{2} \right] \rho \alpha L^2$, 定义 E'_d 如下:

$$E'_d = \delta (C + R) \left(1 + \frac{2R}{C} \right) \quad (5)$$

令 $d(E'_d)/d(C) = 0$, 则不难求出 $C = \sqrt{2}R$. 在此情况下, $E'_d = 6.31 \rho \alpha L^2 R$, 而 $E_d = 6.74 \rho \alpha L^2 R$. 因为对于任意的 C , $E_d > E'_d$, 故 E_d 的下界是 $6.31 \rho \alpha L^2 R$, 这表示当 $C = \sqrt{2}R$ 时, EXQ 的能量开销不会超过最小能量开销的 106.79%, 因此可以说, 选择 $C = \sqrt{2}R$ 是近似最优的. 因为 R 通常远小于网络的边长 L , EXQ 将查询处理过程中的能量开销从 $O(L^3)$ 降低到 $O(L^2 R)$, 从而极大地减少了能量开销.

负载均衡分析. 在查询处理过程中, 因为监测区域中的每一个点最多被 $k=4$ 个子区域覆盖 (在六边形划分中 $k=3$), 每个节点需要转发的数据均来自最多 k 个区域. 而根据 $C = \sqrt{2}R$ 可知, 在最优划分下每个子区域包含的节点数与子区域面积即 R^2 成正比, 因此每个节点的负载上界为 $O(R^2)$. 考虑到在集中式算法中 sink 附近任务最重的节点负载为 $O(L^2)$, EXQ 比集中式算法具有更好的负载均衡特性.

时间复杂度分析. EXQ 的本地查询结果由 m 个头节点进行计算, 显然, 用于每个头节点计算的传感器读数个数为 $O(n/m)$, 则每个头节点的计算复杂度为 $O((n/m)^2 \log(n/m))$. 当 $C = \sqrt{2}R$ 时, 有 $m = L^2 / (8R^2)$, 又因为 $n = L^2 / \rho$, 故每个头节点的计算复杂度为 $O(8R^2 / \rho \log(8R^2 / \rho))$, 即 $O(R^2 \log(R))$, 则 EXQ 在本地数据聚集阶段总共需要时间为 $O(n^2 / m \log(n/m)) = O(L^2 R^2 \log(R))$. 此外, EXQ 在子区域划分和全局数据聚集阶段均需要 $O(m) = O(L^2 / R^2)$ 次计算, 故 EXQ 的时间复杂度为 $O(L^2 R^2 \log(R)) + O(L^2 / R^2) = O(L^2 R^2 \log(R))$. 与集中式算法的时间复杂度 $O(n^2 \log(n))$ 即 $O(L^4 \log(L))$ 相比, 当 $R \ll L$ 时, 明显优于集中式算法.

3.3 子区域的其他划分方式

除方格划分外, 还有其他两种通常的划分方式: 六边形划分和三角形划分. 它们都可以将整个网络划分为大小相同的正六边形或者正三角形, 然后以此为基础构成具有新形状的子区域, 如图 5(c) 所示.

在六边形划分方式中, 一个正六边形对应的子区域由该六边形以及围绕它的 6 个长 C 宽 R 的矩形和 6 个半径为 R 的 $1/6$ 圆组成. 令 C 为六边形的边长, 则子区域的面积要小于半径为 $C + 2R/\sqrt{3}$ 的圆的面积, 如图 5(d) 所示. 因此, 当 $C = 4R/\sqrt{3}$ 时, 一个子区域的能量开销 $E_h < 2\pi \rho \alpha L^2 R$. 另一方面, 子区域的面积要大于半径为 $\sqrt{3}C/2 + R$ 的内切圆的面积 (如图 5(d) 所示), 故当 $C = 2R/(3 - \sqrt{3})$ 时, $E_h > \sqrt{3}/(3 - \sqrt{3}) \rho \alpha L^2 R$.

因为对于任意的 C , E_h 的上界均小于 E_d 的下界, 因此六边形划分要优于方格划分. 通过计算可知, 节省的能量开销在方格划分能量开销的 6.7%~36.29% 之间.

与前两种划分方式相比, 三角形划分是不能节省能量开销的. 在此划分方式中, 一个子区域由一个正三角形以及围绕它的 3 个长 C 宽 R 的矩形和 3 个半径为 R 的 $1/3$ 圆组成. 令 C 为三角形的边长, 则子区域的面积比半径为 $\sqrt{3}C/6 + R$ 的内切圆面积要大. 而因为该划分方式的能量开销 E_t 在 $C = 2\sqrt{3}R/(3\sqrt{3} - 1)$ 时取得最小值, 且 E_t 满足 $E_t > 18\pi/(3\sqrt{3} - 1) \rho \alpha L^2 R = 13.48 \rho \alpha L^2 R$, 明显大于方格划分和六边形划分的能量开销上界, 因此, 三角形划分的能量开销比前两种划分方式都要大得多.

需要说明的是, 本文采用的网络能耗模型适用于传感器节点位置均匀分布的情况. 当传感器节点位置分布不均匀时, 需要针对特定的分布进行网络能耗估计. 而由定理 1 可知, EXQ 算法在传感器节点位置任意分布的情

况下,都能得到极值区域查询的结果 $Ext(R)$.

4 模拟实验

模拟实验测试了针对极值区域查询处理的集中式算法和 EXQ 算法的性能,对于 EXQ 算法,实验设置了方格划分和六边形划分这两种划分方式,因为三角形划分方式的能量开销明显大于前两种划分,因而实验中并没有考虑三角形划分方式.实验中的主要参数有:查询半径 R 、监测区域边长 L 、传感器数目 n 和传感器通信半径 r .这几种参数都出现在各种能量开销的表达式中,因此,它们对于实验结果至关重要.针对不同的参数,实验比较了集中式算法和 EXQ 算法的能量开销以及传感器节点的能耗分布情况.实验采用 TOSSIM 模拟器^[30]来模拟网络传输和统计传输数据量,实验中使用的网络拓扑由固定通信半径 r 的形式随机产生.为消除不确定因素,以下每一组实验都在 10 种不同的网络拓扑上进行,并且对于每一种网络拓扑,实验都重复进行 10 次,最后取这 100 种情况的平均值作为实验结果.

实验中采用的路由方式是目前比较普通和可靠的树状路由方式.实验中的路由树以类似于 BFS 的机制建立.具体来说,尚未加入确定路由节点集合 S 的一个传感器节点 u 将选择 S 中离它最近的邻居节点作为它在路由树中的父节点,然后, u 将广播一条消息给它的邻居节点,告知它已经加入到 S 中.在本地聚集中, S 被初始化为一个子区域的头节点;而在全局聚集中, S 被初始化为 sink 节点.图 6 显示了集中式算法、以方格划分的 EXQ 算法以及以六边形划分的 EXQ 算法的拓扑建立示例.其中,参数设定为 $L=300\text{m}$, $n=900$, $r=20\text{m}$ 以及 $R=20\text{m}$.

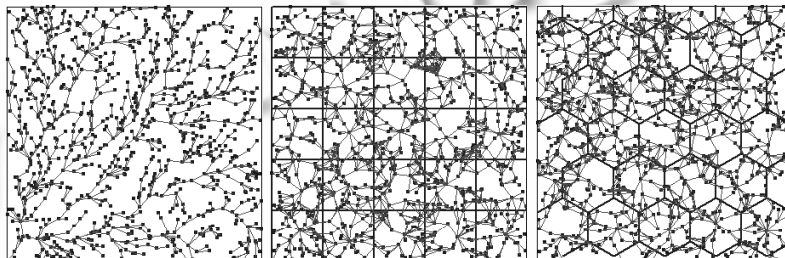


Fig.6 Three sample topologies for the three approaches, which are the centralized approach, EXQ with square division and EXQ with hexagon division from the left to the right

图 6 从左至右分别为集中式算法、基于方格划分的 EXQ 以及基于六边形划分的 EXQ 的拓扑示例

第 1 个模拟实验用于验证子区域大小与能量开销之间的关系,实验设置参数 $L=500\text{m}$, $R=20\text{m}$, $r=20\text{m}$.方格和六边形的边长从 $r/2=10\text{m}$ 逐步增加到 500m ,区内包含 $2\sqrt{2}R=56.56\text{m}$.由图 7(b)可以看出,随着子区域大小的不断增加,方格划分和六边形划分的能量开销均呈现出先减小后增加的 U 型曲线,这是与公式(4)完全符合的.在方格划分中,当最小的能量开销(15 930 个数据包)出现在方格边长为 125m 时,整个监测区域共被划分为 $4\times 4=16$ 个子区域.与之相比,理论分析结果显示,在这种情况下,能量开销是 17 409 个数据包,且方格边长为 57m ,方格数为 64 个.实验结果仅比理论分析结果多出 9.28%.在六边形划分下,最小的能量开销出现在六边形边长为 100m 时,为 14 040 个数据包,优于方格划分大约 13%.

图 7(a)显示了各种策略的能量开销随网络规模的变化情况.此时,网络边长以 50m 的步长从 100m 递增到 $1\ 000\text{m}$,而其他参数设置如下: $R=20\text{m}$, $r=20\text{m}$, $n=L^2/100$.其中, n 的取值意味着网络密度 $\rho=0.01\text{m}^{-2}$ 是固定不变的.随着网络规模的增加,集中式算法的能量开销迅速增大,而 EXQ 的两种策略的能量开销虽然也在不断增大,但增加幅度远远小于集中式算法.特别是当监测区域的边长达到 $1\ 000\text{m}$ 时,EXQ 的两种策略所需能量仅为集中式算法的 15%左右.此外,尽管六边形划分比方格划分更节省能量,然而区别并不明显.从图上还可以看出,对集中式算法和方格划分的 EXQ 算法的理论分析结果与实验结果符合得很好.在理论分析中,我们取系数 $a=16$,这意味着平均每跳的传输距离是 16m ,它小于通信半径 $r=20\text{m}$,就实际情况来看也是合理的.

