ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.4, April 2006, pp.794−805
DOI: 10.1360/jos170794

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

无线传感器网络环境下时-空查询处理方法*

郭龙江, 李建中+, 李贵林

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

Spatio-Temporal Query Processing Method in Wireless Sensor Networks

GUO Long-Jiang, LI Jian-Zhong+, LI Gui-Lin

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

+ Corresponding author: Phn: +86-451-86415827, E-mail: lijzh@hit.edu.cn, http://db.cs.hit.edu.cn

Guo LJ, Li JZ, Li GL. Spatio-Temporal query processing method in wireless sensor networks. *Journal of Software*, 2006,17(4):794−805. http://www.jos.org.cn/1000-9825/17/794.htm

Abstract: In wireless sensor networks, observers are interested in spatio-temporal information monitored by sensors. Observers are not interested in sensor itself or massive irrelevant readings from sensors. They often issue spatio-temporal queries such as "Which events did happen in region *R* from 10:00 to 12:00?". Since battery supply of sensors is limited, energy-efficient spatio-temporal query processing in sensor networks has become an important research problem. This paper presents a spatio-temporal query processing algorithm based on data-centric storage. The energy consumption of sensors in three storage strategies, namely external storage, local storage and data-centric storage, is analyzed and compared in this paper. The paper also studies the influence of the probability of an event occurring, node density, number of event types, number of queries, temporal window size and spatial area size in spatial-temporal query on energy consumption. Analytical and experimental results show that in most cases the spatio-temporal query processing algorithm proposed in this paper can save more energy than those algorithms based on the external storage and local storage strategies.

Key words: sensor network; spatio-temporal query; event; data-centric storage

摘 要: 在无线传感器网络环境中,观察者感兴趣的是由传感器网络监测得到的与时间-空间相关的事件,而不是传感器本身或者大量无关的观察数据.观察者会经常提出与事件相关的时-空查询,例如:"网络覆盖的某地理区域*R*中10:00~11:00发生了哪些事件?".由于每个传感器节点只有有限的能量,因此,研究能量有效性的时-空查询处理算法成为一个重要的研究课题.给出了一种以数据为中心的时-空查询处理算法.针对 3 种不同的存储策略:以数据为中心的存储、外部存储和节点本地存储,分析比较了节点的能量消耗.系统地研究了在 3 种不同的数据存储策略下,事件发生的概率,节点密度,事件类型数目,查询数目,时-空查询地理区域的大小以及时-空查询时间窗口的大小对节点能量消耗的影响.理论与实验结果表明,在多数情况下,这种以数据为中心的时-空查询处理算法的能量消耗少于基于外部存储和本地存储的时-空查询处理算法.

关键词: 传感器网络;时-空查询;事件;数据为中心的存储

中图法分类号: TP393 文献标识码: A

* Supported by Key Program of the National Natural Science Foundation of China under Grant No.60533110 (国家自然科学基金重点项目); the National Natural Science Foundation of China under Grant No.60473075 (国家自然科学基金); the Key Program of the Natural Science Foundation of Heilongjiang Province of China under Grant No.ZJG03-05 (黑龙江省自然科学基金重点项目)

Received 2005-06-15; Accepted 2005-10-11

近几年,无线传感器网络技术得到了飞速的发展.传感器网络是由一组传感器以 Ad Hoc 方式构成的有线或无线网络,其目的是协作地感知、采集和处理网络覆盖的地理区域中感知对象的信息,并发布给观察者[1].观察者感兴趣的是由传感器网络监测得到的与时间-空间相关的事件,而不是传感器本身或者大量无关的观察数据.例如:“上午 10:00 地理区域 R 中发生火灾”就是一个事件.用户不会提出这样的查询:“ID 为 70 的传感器监测的温度是多少?”,而常会提出一些时-空查询,如“网络覆盖区域中的哪些子区域 2:00~8:00 出现了霜冻?”或者“网络覆盖的子区域 R 中 1:00~12:00 发生了哪些事件?”.由于每个传感器具有有限的计算资源、存储资源以及能量,网络中的通信带宽也很有限,如何在资源有限的约束条件下有效地存储和管理与时-空相关的事件、更有效地回答观察者的时-空查询成为我们面临的一个非常紧迫的挑战.

目前,关于传感器网络时-空查询的研究工作还很少.文献[2]给出了基于本地存储(local storage)的时-空查询处理算法.文献[3]给出了文献[2]中算法的能量消耗模型,并给出了 3 种处理算法:WinFlood,FullFlood 和 WinDepth.研究表明:WinFlood 更适合时-空查询;而 WinDepth 则不适合时-空查询;FullFlood 更适合查询地理区域较大的情况.该文并没有讨论查询的数目、事件发生的概率、节点密度、事件类型数目对节点能量的影响,也没有对其他存储策略进行研究.本文给出了一种以数据为中心的时-空查询处理算法.在第 3 节,理论上系统地分析比较了在外部存储方式、本地存储方式和以数据为中心的存储方式下,查询的数目、事件发生的概率、节点密度、事件类型数目对节点能量的影响.第 4 节通过实验验证了第 3 节的理论研究,并通过实验研究了时-空查询中的时间窗口大小以及空间区域大小对节点能量的影响.研究表明,本文提出的以数据为中心的时-空处理算法多数情况下优于 WinFlood.目前,国内在这方面的研究工作还很少.

由于事件信息需要事先保存起来,然后再通过查询把事件信息从网络中提取出来.因此,我们介绍一下目前传感器网络存储方面的研究工作.这方面的研究主要包括:文献[4,5]给出了以数据为中心的存储策略(data centric storage,简称 DCS).DCS 也存在一些问题,如网络中所有相同的事件都要发送给主节点,查询也要发送到主节点,因此主节点以及主节点周围的节点能量消耗最为严重;文献[6]提出一种基于环的索引(ring-based index)方法,解决了文献[4,5]的问题.文献[4,5]中存在的另一个问题是:没有考虑将事件类型为 ET 的事件保存到网络何处最省能量,本文在第 1.2 节给出了一个节省能量的方案.文献[4,5]中存在的第 3 个问题是:采用文献[4,5]中改进的 GPSR 路由算法,可能会出现目标节点不唯一的情况,即存在多个与路由目标距离相同且最近的节点.如果出现目标节点不唯一的情况,则同类型的事件信息可能会保存在多个节点上.因此,采用文献[4,5]中改进的 GPSR 路由算法从网络中往往只能获取部分事件信息.本文第 2.1 节给出了一种改进的路由算法,解决了这个问题;文献[7]提出了一个改进的 DCS 存储策略(resilient data-centric storage,简称 R-DCS).该策略降低了节点的平均存储代价以及获取数据的平均通信代价;文献[8,9]提出了一个称为维(dimensions)的层次体系存储结构,这种层次结构适合于下钻(drill down)操作;文献[10]给出了一种称为 DIFS(distributed index for features in sensor networks)的分布式索引方法,可以有效地处理区域查询(range query);文献[11]提出了一个称为 DIM(distributed index for multi-dimensional data)的能够支持多维属性区域查询的空间索引结构.

1 事件的生成与存储

在传感器网络环境下,用户更关心事件的发生.因此,不必研究所有观测数据的存储,而只需研究如何对观测数据进行预处理,计算观测数据是否满足事件发生的条件.如果满足事件发生的条件,传感器网络就把事件存储下来,而不保存大量无关的观测数据.保存事件信息的节点称为事件存储节点.

1.1 事件的生成

为讨论方便,首先给出如下相关定义:

定义 **1**(观察属性). 它是指预先指定传感器监测的某种指标,记为 $A_1,A_2,\ldots,A_n$.例如,温度、压强、湿度等.可以指定一个传感器监测一个观察属性,也可以监测多个观察属性.

定义 **2**(观察数据). 它是指观察属性的具体值.如果一个传感器监测一个观察属性,则观测数据就是一个实数.例如,23.5°C,15 帕斯卡等.

定义 3(工作周期). 每经过一段时间,传感器节点就采集一下观察数据,这段时间称为工作周期.

定义 4(事件类型). 时间类型是指人们感兴趣的事情.例如:森林火灾、霜冻、暴雨、地震等.事件类型可以看作多个观察属性组成的谓词条件.事件类型记为 $ET(A_1,A_2,\ldots,A_k)$,其中$(A_1,A_2,\ldots,A_k)$为 ET 的观察属性.

定义 5(观察区域). 逻辑上把传感器网络覆盖的区域划分成 $m\times n$ 个网格,每个网格称为一个观察区域.

定义 6(事件). 事件类型的一个实例称为一个事件,记为 $e(ET,g,t)$.ET 表示事件类型;g 表示事件 e 发生的观察区域编号;t 表示事件 e 发生在第 t 个工作周期.

定义 7(时-空查询). 查询 $Q\{[t_1,t_2],Rect\}$:给定起止工作周期$[t_1,t_2]$以及网络中的矩形地理区域 $Rect=[x_1,y_1,x_2,y_2]$,找出所有从第 t_1 个工作周期到第 t_2 个工作周期内在 $Rect$ 内发生的事件.其中,(x_1,y_1)是矩形地理区域的左上角坐标;(x_2,y_2)是矩形地理区域的右下角坐标;$[t_1,t_2]$确定了一个时间窗口.

传感器节点监测观察属性有两种情况:(1) 一个传感器节点负责监测一个观察属性;(2) 一个传感器节点负责监测一个事件类型所涉及的所有观察属性.如果一个事件类型所包含的观察属性比较多,则情况(2)需要向传感器节点里配置较多的感知部件,这样会提高节点的成本.因此,我们主要考虑情况(1).

负责监测观察属性的节点称为观察节点.若要完成事件类型 $ET_i(A_{i1},A_{i2},\ldots,A_{ik_i})$的监测,需要 k_i 个分别负责监测观察属性 $A_{i1},A_{i2},\ldots,A_{ik_i}$ 的观察节点,简称事件类型 ET_i 的观察节点为 $A_{i1},A_{i2},\ldots,A_{ik_i}$.每个观察节点都知道自己监测的观察属性所属的事件类型.假设观察者关心 W 个事件类型 $ET_1,ET_2,\ldots,ET_w$,每个观察区域内都需要监测这 W 个事件类型,因此观察区域内需要布置足够的观察节点.

为了捕获事件类型为 ET_i 的事件,需要从 ET_i 的观察节点 $A_{i1},A_{i2},\ldots,A_{ik_i}$ 中选择一个节点作为 ET_i 的事件合成节点.假设选择 A_{i1} 为 ET_i 的事件合成节点,则每经过一个工作周期,ET_i 的观察节点就将观察数据路由到 A_{i1}.A_{i1} 接收到所有观察节点发来的观察数据后,就按照事件类型 ET_i 的谓词条件进行计算,判断事件是否发生.如果产生事件,事件合成节点就把事件信息路由到一个指定节点,该节点保存整个网络区域内事件类型为 ET_i 所有事件,该节点称为 ET_i 的事件存储节点.

为使传输观察数据所消耗的能量均匀地分摊到 ET_i 的观察节点 $A_{i1},A_{i2},\ldots,A_{ik_i}$ 上,ET_i 的事件合成节点应由 $A_{i1},A_{i2},\ldots,A_{ik_i}$ 轮番担任.具体的选择办法是:假设当前为第 k 个工作周期,$j=(k \bmod k_i)+1$,则 $A_{i,j}$ 为第 k 个工作周期中 ET_i 的事件合成节点.为了节省能量,观察节点观测到数据后需要判断一下:如果新观察数据较以前的数据发生很大变化,才将观察数据路由到事件合成节点.这需要观察节点增量维护观察数据的均值和方差.假设观察节点前 k 个工作周期所采集的观察数据为 $x_1,x_2,\ldots,x_k$,则观察数据的均值 $\bar{x}=\frac{1}{k}\sum_{i=1}^{k}x_i$,观察数据的方差 $s^2=\frac{1}{k-1}\left(\sum_{i=1}^{k}x_i^2-k\bar{x}\right)$.在第 k+1 个工作周期采集的数据为 x_{k+1},则

$$\bar{x}_{new}=\frac{1}{k+1}(k\bar{x}+x_{k+1}),\ s^2_{new}=\frac{1}{k}[(k-1)s^2+k\bar{x}+x^2_{k+1}-(k+1)\bar{x}_{new}].$$

如果 $s^2_{new}>s^2$,则将新观察数据路由到事件合成节点,否则不发送.

1.2 事件的存储

根据事件存储节点在网络中的位置,可以把传感器网络数据存储方式分成 3 类:外部存储、本地存储和以数据为中心的存储.

外部存储(external storage)[12]:基站(sink)节点是事件存储节点,事件合成节点生成的事件都被路由到基站节点.同时,时-空查询也在基站节点上执行;

本地存储(local storage)[13]:事件合成节点生成的事件信息均保存在事件合成节点本地.当用户发出时-空查询时,查询以泛洪(flood)方式或窗口泛洪(WinFlood)[2]方式在网络中传播.

以数据为中心的存储(data-centric storage)[4,5,7]:该存储策略用散列函数把事件类型 ET_i 映射到传感器网络中某个地理位置(x_i,y_i),距离(x_i,y_i)最近的节点充当 ET_i 的事件存储节点.当事件合成节点生成事件时,事件信息被路由到距离(x_i,y_i)最近的节点上.目前的文献还没有讨论把事件类型为 ET_i 的事件存储到网络中的何处最节省

能量的问题.

本文基于数据为中心的存储策略,提出了一种节省能量的策略:网络中的 $m\times n$ 个观察区域被编码,用散列函数将事件类型 ET_i 映射到网络中心的观察区域 Z_j,在观察区域 Z_j 内距离网络中心最近的节点被指定为 ET_i 的事件存储节点.观察区域的编码方案是:对于第 i 行第 j 列的观察区域编码为 k,这里 $k=(i-1)\times n+j$,$i=1,2,\ldots,m$;$j=1,2,\ldots,n$.本文的存储策略称为数据为中心的中心映射存储策略(center mapping data centric storage,简称 CM-DCS).为说明我们的映射策略最节省能量,首先给出如下定理:

引理 **1**. 在无线传感器网络中,事件信息从事件合成节点路由到事件存储节点所消耗的总能量,与事件合成节点到事件存储节点的距离成正比.

证明:事件合成节点到事件存储节点的距离 D 可以近似表示为 $D=x\times d$,其中:x 为从事件合成节点到事件存储节点的最小跳数;d 为每一跳的平均距离.设每个节点接收并转发事件信息所消耗的能量为 v,从事件合成节点到事件存储节点需要转发 x 次信息,有 x 个节点因转发事件信息而消耗了能量.把一个事件信息从事件合成节点路由到事件存储节点所消耗的能量:$V=x\times v=(D/d)\times v$.当 D 增大时,V 增大,证毕. □

引理 **2**. 若观察节点均匀地分布在观察区域内,则事件合成节点的坐标可用所在观察区域的中心近似代替.

证明:因为观察节点均匀地分布在观察区域内,因此,观察节点的坐标(X,Y)可以看作是服从二维均匀分布 $U(a,c,b,d)$的二维随机变量,其中:(a,b)是观察区域的左下角坐标;(c,d)是观察区域的右上角坐标.事件合成节点由监测同一事件类型的观察节点轮番担当,因此,事件合成节点的坐标(CX,CY)也可以看作是服从总体 $U(a,c,b,d)$的二维随机变量.计算 CX 和 CY 的数学期望:$E(CX)=0.5\times(a+c)$,$E(CY)=0.5\times(b+d)$.$(E(CX),E(CY))$是观察区域的中心,由概率论知识可知:当随机变量取值为数学期望时,随机变量的方差最小.因此,观察区域内的事件合成节点的坐标可用观察区域的中心近似代替,证毕. □

定理 **1**. 若观察节点均匀地分布在观察区域内,指定距离网络中心最近的节点作为事件存储节点,可使传输事件所消耗的能量最少.

证明:由引理 2 可知,每个观察区域的中心可以近似看作 ET_i 的事件合成节点的坐标.设 $m\times n$ 个观察区域的中心坐标为$(X_1,Y_1),(X_2,Y_2),\ldots,(X_{m\times n},Y_{m\times n})$,$ET_i$ 的事件存储节点的坐标为(X,Y).各个观察区域内,ET_i 的事件合成节点都要将产生的事件路由到(X,Y).由引理 1 知,传输事件的能量消耗与传输的距离成正比,因此,ET_i 的事件存储节点的位置选择问题转化成:找(X,Y),使得 $\min\sum_{i=1}^{m\times n}(X_i-X)^2+(Y_i-Y)^2$.对 X 与 Y 分别求偏导数得 0,解得 $X=\frac{1}{m\times n}\sum_{i=1}^{m\times n}X_i$,$Y=\frac{1}{m\times n}\sum_{i=1}^{m\times n}Y_i$.$(X,Y)$近似等于网络中心坐标,因此,指定距离网络中心最近的节点作为事件存储节点,可使传输事件所消耗的能量最少,证毕. □

根据定理 1 可以构造散列函数,把事件类型 ET_k 映射到网络中心地带的观察区域 Z_r.网络被划分成 $m\times n$ 个观察区域,对于正整数 m,n,存在尽可能小的正整数 m_1 和 n_1,满足 m_1 的奇偶性与 m 相同,n_1 的奇偶性与 n 相同,且 $m_1<<m,n_1<<n,W\le m_1\times n_1$.将 W 个事件类型映射到由 $m_1\times n_1$ 个观察区域构成的网络中心地带.对于第 k 个事件类型 ET_k 映射到网络的中心区域 Z_r,这里,$k=(i-1)\times n_1+j$,$r=[(m-m_1)\times n]/2+(n-n_1)/2+(i-1)\times n+j$,$i=1,2,\ldots,m_1$;$j=1,2,\ldots,n_1$.图 1 中,网络监测 4 个事件类型:$ET_1,ET_2,ET_3,ET_4$.网络被划分成 4×4 个观察区域,编码为$\{Z_1,Z_2,\ldots,Z_{16}\}$.构造映射:$\{ET_1\to Z_6;ET_2\to Z_7;ET_3\to Z_{10};ET_4\to Z_{11}\}$.$Z_6,Z_7,Z_{10},Z_{11}$ 是中心区域,标为阴影区域,这里,$m=4$,$n=4$,$m_1=2$,$n_1=2$,$W=4$.深色节点是各个阴影区域内距离网络中心最近的节点被指定为事件存储节点.Z_6,Z_7,Z_{10},Z_{11} 中的深色节点分别为 ET_1,ET_2,ET_3,ET_4 的事件存储节点.

如果事件类型 ET_i 的事件发生较频繁,则 ET_i 的事件存储节点周围的节点由于要转发较多的事件,其能量消耗会较快些.当 Z_j 内的事件存储节点 A 的剩余能量低于一个阈值时,它将向 Sink 提出调整事件类型到观察区域映射的申请.Sink 节点接到 A 的申请后,根据本地维护的集合 G,构造新的映射.G 是观察区域的下标集合,若 $l\in G$,则有两种可能:(1) 曾经存在一个映射 $ET_p\to Z_l$,由于 ET_p 的事件发生较频繁,导致 Z_l 内的节点能量消耗较多,映射被调整过;(2) 当前存在一个映射 $ET_q\to Z_l$.

如果原映射为 $ET_i \to Z_j$,则应选择新的观察区域 Z_k,构造新映射 $ET_i \to Z_k$.Z_k 的中心坐标为

$$X_k = \frac{1}{|m \times n| - |G|} \sum_{i \notin G} X_i \ , Y_k = \frac{1}{|m \times n| - |G|} \sum_{i \notin G} Y_i \ .$$

Sink 将调整的映射在网络中泛洪,通知每个节点.在如图 1 所示的例子中,事件类型为 ET_1 的事件发生较频繁,则 ET_1 事件存储节点周围的节点能量消耗较快,因此,网络工作一段时间后可以重新构造一个映射:$ET_1 \to Z_2$,构造新映射前,G={6,7,10,11}.图 2 是调整映射后的情形,Z_6 的邻居观察区域 Z_2 被指定为 ET_1 的映射区域.观察区域 Z_2 中距离网络中心最近的节点成为 ET_1 的新的事件存储节点.调整映射后,G={2,6,7,10,11}.网络工作一段时间后,经过上述调度,网络内各个观察区域内的节点能量消耗会达到均衡.

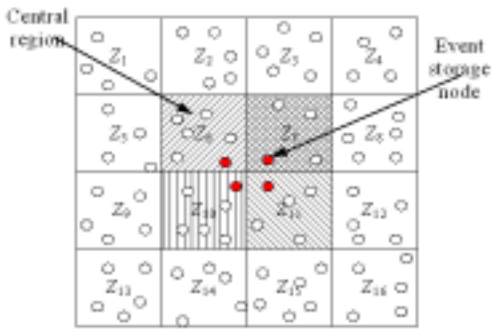

Fig.1 Distribution of event storage nodes

图 1 事件存储节点的分布

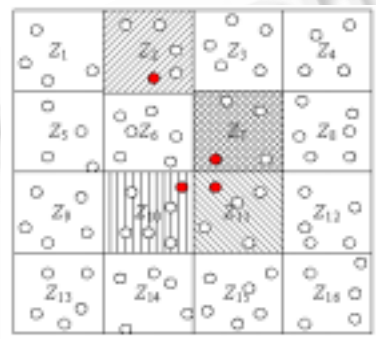

Fig.2 Distribution of event storage nodes after adjusting

图 2 调整后的事件存储节点的分布

1.3 事件存储节点的存储策略

事件存储节点需要保存网络中长时间内发生的同一类型事件信息,事件信息是数据流.理论上,无论是 Sink 节点,还是一般的传感器节点,都无法保存下所有的事件信息.为了更有效地回答时-空查询,就需要研究单节点的存储策略,本文给出了带有时间印的位向量集合存储策略.

带有时间印的位向量集合可以表示为 $B(ET_i)$={$(I,t)|I$ 是含有 $m \times n$ 个位的位向量,t 是工作周期编号,$t \geq 0$}.如果 I 的第 $j_1,j_2,\ldots,j_k$ 位为 1,其他位为 0,则表示在第 t 个工作周期内,观察区域 $Z_{j_1},Z_{j_2},\ldots,Z_{j_k}$ 内发生了事件类型为 ET_i 的事件.若 $ET_i \to Z_j$,则 $B(ET_i)$被保存在 Z_j 内距离网络中心最近的节点上.

假设事件存储节点最多能够保存 V 个位向量,$B(ET_i)$中的位向量被划分成 x 个互不相交的位向量子集合 $SB_0,SB_1,\ldots,SB_{x-1}$, $B(ET_i) = \bigcup_{i=1}^{x-1} SB_i$.划分的原则是:(a) 每个子集合最多保存 $l(l>0)$个位向量,$x \times l \leq V$;(b) 若$(I,t) \in SB_i$,则满足 t 能够被 2^i 整除;(c) 对于$\forall (I,t)$,若 t 既能被 2^i 整除又能被 2^j 整除且 $j>i$,则$(I,t) \in SB_j$.

(I) 位向量的插入:系统初始时,$B(ET_i)$为空集.如果事件存储节点接收到事件信息 $e(ET_i,g,t)$,根据 t 计算 $e(ET_i,g,t)$所属的子集合 SB_j.如果已经存在一个$(I,t) \in SB_j$,则把位向量 I 的第 g 位置 1;否则,建立一个新的位向量 (I_{new},t),位向量 I_{new} 的第 g 位置 1,其余位置为 0.

(II) 位向量的删除:若新产生的位向量$(I_{new},t) \in SB_i$,且 SB_i 中的位向量数目已经等于 l,则应先将 SB_i 中某个位向量删除,再将(I_{new},t)插入 SB_i.删除策略有两种:(a) 删除 SB_i 中时间最老的位向量;(b) 删除 SB_i 中包含事件信息最少的位向量.

当事件合成节点生成了一个事件,就需要将事件信息路由到事件存储节点上.事件类型为 ET_i 的事件合成节点通过映射 $ET_i \to Z_j$.根据第 1.2 节的讨论,可以将 Z_j 的 4 个顶点坐标中距离网络中心最近的顶点坐标作为路由目标[4],然后采用本文第 2.1 节的路由算法,可以将事件路由到事件存储节点.

2 时-空查询处理

本节研究基于 CM-DCS 的时-空查询处理算法回答时-空查询 Q[t_1,t_2],$Rect$}.网络中可能会出现 ET_i 的事件

存储节点不唯一的情况.如图 3 所示,在观察区域 Z_6 中,有 3 个与网络中心距离相同的事件存储节点.由于发送查询时预先不知道同类型的事件信息保存在哪些节点上,采用文献[4,5]中的 GPSR 路由算法可能获得部分事件信息.本节提出了两跳邻居路由算法(two hops neighbors greedy routing protocol,简称 THN-GRP).

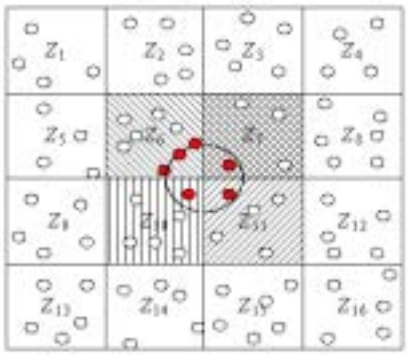

Fig.3 Event storage nodes with same event type are in Z_6

图 3 Z_6 中保存同一事件类型的事件存储节点

2.1 THN-GRP路由算法

具体算法如下:

1. 计算一跳(one hop)邻居集合.假设每个节点都知道自己的位置,初始时,每个节点都向自己的一跳邻居节点发送自己的位置信息,同时接收自己一跳邻居节点发来的位置信息.这样,每个节点都确定了一个关于自己的一跳邻居集合,节点 i 的一跳邻居集合记为 K_i.

2. 填充路由包.在如图 4 所示的路由包中,*dest* 域保存路由目标(*destx*,*desty*);*nearest-neighbors* 域保存与(*destx*,*desty*)最近的节点的坐标(可能有多个);*info* 域保存事件信息,查询或者查询结果.

Dest	Nearest-Neighbors	Info.

Fig.4 Structure of routing packet

图 4 路由包的基本结构

3. 选择下一跳节点.假设当前欲转发路由包的节点为 p,p 将根据其邻居位置信息决定将路由包转发给下一跳节点.设 p 的一跳邻居 *ID* 集合为 $K_p=\{q_1,q_2,\ldots,q_m\}$,$p$ 到(*destx*,*desty*)的距离为 D_p.p 计算出到(*destx*,*desty*)最近的一跳邻居 q_f,其到(*destx*,*desty*)的距离为 D_{q_f}.

下一跳节点的选择方法如下:

(a) 如果 $D_{q_f}<D_p$,则下一跳节点为 q_f,p 将信息包转发给 q_f;

(b) 如果 $D_{q_f}\geq D_p$,则节点 p 向它的一跳邻居 $q_1,q_2,\ldots,q_m$ 发出请求,要求 $q_1,q_2,\ldots,q_m$ 把自己的一跳邻居集合发送给节点 p.p 接收到 $\bigcup_{i=1}^{m}K_{q_i}$ 后,计算出 p 的两跳邻居集合为 $T_p=\bigcup_{i=1}^{m}(\overline{K}_p\cap K_{q_i})$.$p$ 从 T_p 中选择出到(*destx*,*desty*)最近的两跳邻居 g,其到路由目标的距离记为 D_g.

我们分别讨论 D_g 和 D_p 的关系:

(c) 如果 $D_g<D_p$,设 $g\in K_{q_j}$,则下一跳节点为 q_j,p 将信息包转发给 q_j;

(d) 如果 $D_g\geq D_p$,则分两种情况:(i) 如果 $D_g>D_{q_f}>D_p$,则说明在节点 p 和目标节点之间存在一个没有传感器节点的空白区域,此时采用右手规则选择下一个节点;(ii) 如果 $D_{q_f}>D_g>D_p$,设 $g\in K_{q_j}$,如果下一跳节点选为 q_j,则由于 $D_{q_j}\geq D_{q_f}>D_g>D_p$ 而没有受益,因此仍采用右手规则选择下一个节点.

4. 确定 *nearest-neighbors* 域.如果节点 p 采用右手规则选择了下一跳节点 q_j,则 p 是到目前为止距离(*destx*,*desty*)最近的节点.p 将路由包转发给 q_j 之前需要做如下工作:

(a) 如果 *nearest-neighbors* 域为空,则 p 将自己的坐标插入到 *nearest-neighbors* 中;

(b) 如果 *nearest-neighbors* 域非空且 D_p 小于 *nearest-neighbors* 中的坐标与(*destx*,*desty*)的距离,则 p 首先清空 *nearest-neighbors*,再将自己的坐标插入到 *nearest-neighbors* 中;

(c) 如果 *nearest-neighbors* 域非空且 D_p 等于 *nearest-neighbors* 中的坐标与(*destx*,*desty*)的距离,并且 *p* 的坐标与 *nearest-neighbors* 中的坐标不同,则 *p* 将自己的坐标插入到 *nearest-neighbors* 中;

(d) 如果 *nearest-neighbors* 域非空且 D_p 大于 *nearest-neighbors* 中的坐标与(*destx*,*desty*)的距离,则 *p* 不对 *nearest-neighbors* 做任何操作.

最后,*p* 将路由包转发给下一跳节点 q_j.

5. 判断是否到达目的地.如果路由包经过转发后又回到节点 *p*,则 *p* 检查自己的坐标是否在 *nearest-neighbors* 中.如果 *p* 在 *nearest-neighbors* 中,则 *nearest-neighbors* 中保存的坐标是距离(*destx*,*desty*)最近的节点坐标.如果路由包中保存的是事件信息,就将事件信息保存在节点 *p*;如果路由包中保存的是查询信息,则 *nearest-neighbors* 中保存的是同一事件类型的事件存储节点的所有坐标.

2.2 基于CM-DCS的时-空查询处理算法

本节讨论时-空查询 $Q\{[t_1,t_2],Rect\}$的处理算法,该算法大致分为 4 个阶段:

第 1 阶段:计算路由目标.计算出各个事件类型对应的观察区域 $Z_{i_1},Z_{i_2},\ldots,Z_{i_m}$ 以及这些观察区域距离网络中心最近的顶点坐标$\{(x_{i_1},y_{i_1}),(x_{i_2},y_{i_2}),\ldots,(x_{i_m},y_{i_m})\}$.从$\{(x_{i_1},y_{i_1}),(x_{i_2},y_{i_2}),\ldots,(x_{i_m},y_{i_m})\}$中选择出与 Sink 最近的坐标$(x_{i_j},y_{i_j})$作为路由目标,向网络中发送查询路由包.网络中的节点采用 THN-GRP 路由算法转发查询路由包.查询路由包结构如图 5 所示,其中 *dest* 和 *nearest-neighbors* 含义同图 4,*region-centers* 中保存$\{(x_{i_1},y_{i_1}),(x_{i_2},y_{i_2}),\ldots,(x_{i_m},y_{i_m})\}$;*Einfo* 中保存符合时-空查询条件的事件信息.*flag*=1 表示在此之前节点进行了事件信息的合并;*flag*=0 表示没有经过事件信息的合并.

Dest	Nearest-Nneighbors	Regions	Einfo.	Flag

Fig.5 Structure of spatio-temporal query routing packet

图 5 用于查询的路由包结构

第 2 阶段:事件信息的合并.假设节点 *p* 接收到查询路由包,*p* 就检查自己的坐标是否在 *nearest-neighbors* 中.如果 *p* 在 *nearest-neighbors* 中,*p* 就是事件存储节点,*nearest-neighbors* 中保存的是与 *p* 保存同一事件类型的其他事件存储节点的坐标.*p* 将做如下工作:

(a) 如果 *Einfo* 为空,*p* 将本地符合查询条件的事件信息保存到 *Einfo* 中;

(b) 如果 *Einfo* 非空,*p* 将本地符合查询条件的事件信息与 *Einfo* 中的信息合并,同时将 *flag* 置 1;

(c) 将 *p* 自己的坐标从 *nearest-neighbors* 中删除;

(d) 取 *nearest-neighbors* 中下一个坐标作为新的路由目标;

(e) 如果节点 *p* 发现 *nearest-neighbors* 为空且 *flag*=1,则从 *region-centers* 中取出下一个与路由目标最近的坐标作为新的路由目标.

第 3 阶段:返回合并后的查询结果.如果 *region-centers* 为空且 *flag*=1,则表明 $Z_{i_1},Z_{i_2},\ldots,Z_{i_m}$ 内的事件存储节点上的查询结果合并完毕,以 Sink 的坐标作为路由目标,将查询路由包向 Sink 转发.

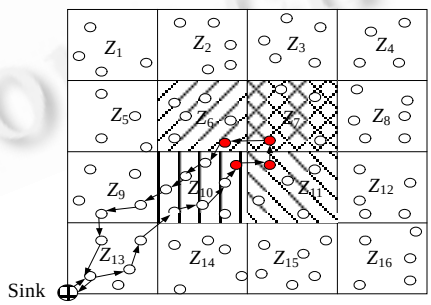

Fig.6 Routing query in sensor networks

图 6 查询在网络中的转发

图 6 显示了查询在网络中转发的过程.当查询路由包到达阴影区域内的事件存储节点时,就进行满足时-空查询条件的查询结果的合并.查询结果合并后,以 Sink 为路由目标将查询结果返回到 Sink 节点.

3 选择优化的存储策略

不同的存储策略会导致不同的能量消耗.能否针对事件生成以及查询应用的特点,给出最节省能量的存储策略?表 1 给出了能量分析中所用到的符号.

节点与邻居的平均距离估计为 $\frac{1}{\pi R^2}\iint_{0\le x,y\le R}\sqrt{x^2+y^2}\,\mathrm{d}x\mathrm{d}y=\frac{2R}{3}$,于是有 $N\approx\frac{X}{(2R/3)}\cdot\frac{Y}{(2R/3)}$.因为

$\frac{1}{(2R/3)}\sqrt{X^2+Y^2}\geq\frac{1}{(2R/3)}\sqrt{2XY}=\frac{1}{(2R/3)}\sqrt{2N.(2R/3)^2}=\sqrt{2N}$,所以相距最远的两个节点之间的跳数约为 $\sqrt{2N}$.

Table 1 Notations and their descriptions in energy analysis

表 1 能量分析中用到的符号及其含义

Notation	Description	Notation	Description	Notation	Description
N	Number of sensor nodes	W	Node density	$prob$	Event occurring probability
X	Size of field covered by sensors on X axis	N_q	Number of query in each round	R	Energy used to unicast a bit
Y	Size of field covered by sensors on Y axis	S_{result}	Size of the query answer	$\varnothing$	Size of the empty query answer
E_r	Energy used to receive a bit	E_u	Radio range	S_{query}	Size of query routing packet
Q_x	Size of query's window on X axis	E_b	Energy used to broadcast a bit	S_{event}	Size of event information
Q_y	Size of query's window on Y axis	ρ	Number of event types monitored by networks	V	Number of bit vector saved by node

3.1 外部存储方式下网络能量消耗的估计

在外部存储方式下,查询在 Sink 节点上执行,因此查询不消耗能量.假设 Sink 布置在网络的左下角坐标(0,0)处.各个节点到 Sink 节点的平均跳数为 $\frac{\sqrt{2N}}{2}$.每个工作周期中平均产生的事件数目为 $W.N.prob$.采用外部存储方式的能量消耗 $E_{external}=W.N.prob.(E_u+E_r)S_{event}\frac{\sqrt{2N}}{2}$.

3.2 CM-DCS存储方式下网络能量消耗的估计

CM-DCS 存储方式下,所有事件信息要汇集到中心区域的事件存储节点上,查询需要路由到事件存储节点上.获得查询结果后,将结果发到 Sink 上.网络中,其他节点到事件存储节点的平均跳数为 $\frac{\sqrt{2N}}{4}$,平均产生的事件数目为 $W.N.prob$.事件汇集到事件存储节点上的能量消耗为 $W.N.prob.(E_u+E_r)S_{event}\frac{\sqrt{2N}}{4}$;发送 N_q 个查询到事件存储节点消耗的能量为 $N_q(E_u+E_r)S_{query}\frac{\sqrt{2N}}{4}$;返回 N_q 个查询结果到 Sink 消耗的能量:$N_q(E_u+E_r)S_{result}\frac{\sqrt{2N}}{2}$.采用 CM-DCS 存储的能量消耗为

$$E_{cm-dcs}=W.N.prob.(E_u+E_r)S_{event}\frac{\sqrt{2N}}{4}+N_q(E_u+E_r)S_{query}\frac{\sqrt{2N}}{4}+N_q(E_u+E_r)S_{result}\frac{\sqrt{2N}}{2}.$$

3.3 本地存储方式下网络能量消耗的估计

在本地存储方式下,事件合成节点将事件保存到本地,所消耗的能量主要是完成查询所消耗的能量.

1. 采用 WinFlood 向网络发送 $Q\{[t_1,t_2],Rect\}$所消耗的总能量[3]为

$$E_{local-storage}^{WinFlood}=\frac{\sqrt{2N}}{4}N_q(E_u+E_r)\left(2S_{query}+S_{result}\sqrt{\frac{Q_xQ_y}{XY}}+2S_{result}\right)+N_qS_{query}\left(E_b+E_r\frac{N}{XY}\pi R^2\right)\frac{N}{XY}Q_xQ_y.$$

2. 采用 FullFlood 向网络发送事件查询 $Q\{[t_1,t_2],Rect\}$所消耗的总能量[3]为

$$E_{local-storage}^{FullFlood}=N_q\left(E_b+E_r\frac{N}{XY}\pi R^2\right)N.S_{query}+N_q.\varnothing\left(E_u+E_r\frac{N}{XY}\pi R^2\right)\left(N-\frac{N}{XY}Q_xQ_y\right)+N_q(E_u+E_r)S_{result}\frac{\sqrt{2N}}{2}.$$

3.4 理论上的能量消耗分析比较

$E_b,E_r,E_u,R,\varnothing,S_{event},S_{query}$ 都是常数,影响时-空查询能量消耗的因素有:$\rho,N_q,N,prob,W,Q_x,Q_y$ 以及时间窗口的大小 t_2-t_1.如果在$[t_1,t_2]$内,在查询地理区域 $Rect$ 内有 k 个事件类型的事件发生,则查询结果可以表示为$(f_{i_1},f_{i_2},\ldots,f_{i_k})$,$f_{i_j}$ 表示事件类型为 ET_{i_j} 的事件在$[t_1,t_2]$内在网络覆盖的地理区域 $Rect$ 内发生了 f_{i_j} 次.查询结果的大小

$S_{result}=W\frac{Q_xQ_y}{XY}\frac{t_2-t_1}{V}S_{event}$.由于$\frac{Q_xQ_y}{XY}\frac{t_2-t_1}{V}\le 1$,因此查询结果可以近似表示为$S_{result}=W.S_{event}$.为方便分析,不妨设$E_b=E_u,E_r=1.5E_b,S_{event}=S_{query},\varnothing=0.1\times S_{event}$.

(1) CM-DCS 存储方式与外部存储方式之间的差异

$$E_{cm-dcs}-E_{external}=\frac{\sqrt{2N}}{4}(E_u+E_r)S_{query}((W+1)N_q-W.N.prob)\approx\frac{\sqrt{2N}}{4}W(E_u+E_r)S_{query}(N_q-N.prob).$$

如果$N_q>N.prob$,应采用外部存储方式;如果$N_q\le N.prob$,应采用 CM-DCS 的存储方式.

(2) CM-DCS 存储方式与采用 WinFlood 路由的本地存储方式之间的差异

$$E_{cm-dcs}-E_{local-storage}^{WinFlood}=\frac{5\sqrt{2N}}{8}E_uS_{event}\left[W.N.prob-N_q-N_q\sqrt{\frac{Q_xQ_y}{XY}}\left(\left(1+\frac{3N}{2XY}\pi R^2\right)\frac{8\sqrt{N.Q_xQ_y}}{5\sqrt{2XY}}+W\right)\right].$$

当节点密度$\rho=\frac{N}{XY}$与N_q增大时,应采用 CM-DCS 存储;当ρ为常数,N_q增大,查询的区域面积Q_xQ_y增大时,应采用 CM-DCS 存储;当ρ为常数且N_q较小时,应采用 WinFlood 路由的本地存储.

(3) CM-DCS 存储方式与采用 FullFlood 路由的本地存储方式之间的差异

$$E_{cm-dcs}-E_{local-storage}^{FullFlood}\approx\frac{5\sqrt{2N}}{8}E_uS_{event}\left[W.N.prob-\frac{8\sqrt{N}}{5\sqrt{2}}N_q\left(1+\frac{3N}{2XY}\pi R^2\right)\left(1+0.1\left(1-\frac{Q_xQ_y}{XY}\right)\right)\right].$$

当节点密度ρ与N_q增大时,应采用 CM-DCS 存储.

4 实 验

我们在 ns-2 环境下用 C++和 TCL 实现了本文提出的路由算法 THN-GRP 及基于 CM-DCS 存储方式的时、空查询处理算法.实验设备是一台运行 WinXP 及 cygwin1.59,具有 P4 处理器、512M 内存的 PC.实验指定的 MAC 层为 802.15.4[14].表 2 列出了实验模拟时用到的实际参数.

Table 2 Simulation parameters used in ns-2

表 2 ns-2 环境下的实验参数

Parameters	Values	Parameters	Values	Parameters	Values
Radio range (m)	10	Transmitting power in watts (W)	0.66	Receiving power in watts (W)	0.4
Sleeping power in watts (W)	0.035	Initial energy per node in joules (J)	10 000	Number of bit vector saved by node	200

根据第 3 节的讨论,$\rho,N,W,N_q,prob,Q_x,Q_y$以及时间窗口的大小$t_2-t_1$是影响节点能量消耗的几个重要参数.因此,本文将在不同的存储方式下做以下几组实验:(1) 考察ρ对节点能量消耗的影响;(2) 考察N对节点能量消耗的影响;(3) 考察W对节点能量消耗的影响;(4) 考察N_q对节点能量消耗的影响;(5) 考察$prob$对节点能量消耗的影响;(6) 考察时-空查询空间区域Q_xQ_y大小对节点能量消耗的影响;(7) 考察时-空查询时间窗口大小t_2-t_1对节点能量消耗的影响.

(1) 在不同的存储方式下,考察ρ对节点能量消耗的影响.在实验中,我们将网络拓扑区域设置为正方形($X=Y$),节点密度是变化的.给出 5 组随机的网络拓扑,ρ增大的同时,拓扑区域边界X和Y以及节点数目N也增大.表 3 列出了实验中用到的 5 组网络拓扑区域边界长度、节点数目及节点密度的对应关系.传感器节点均匀随机地布置在$X\times X$正方形区域内.本文利用 ns-2 下的拓扑模拟生成工具 setdest 随机生成了 5 组满足表 3 的网络拓扑.距离(0,0)最近的节点定义为 Sink 节点.

Table 3 Five sensor networks topologies when node density varies

表 3 ρ变化时的 5 组网络拓扑

Field size ($X=Y$)	198	280	323	351	370
N	1 000	2 500	4 000	5 500	7 000
ρ (nodes/m^2)	8/314	10/314	12/314	14/314	16/314

事件发生的概率 $prob$=0.5,事件类型数目 W=4,每个工作周期发送查询数目 N_q=33.查询的空间区域占整个网络区域的百分比服从(0,1)上的均匀分布,查询的时间窗口长度为 10 个工作周期.网络工作 200 个工作周期,前 100 个工作周期产生数据;后 100 个工作周期既产生数据又发送查询.图 7 显示了随着网络节点密度的增大,平均每个节点在 1 个工作周期内的能量消耗情况.从图 7 可以看出,当网络节点密度增大时,CM-DCS 存储回答时-空查询最节省能量.

(2) 在不同的存储方式下,考察 N 对节点能量消耗的影响.我们给出如表 4 所示的 5 组随机的网络拓扑.随着拓扑区域边界(X=Y)的增大,N 有比例地增大,ρ=8nodes/314m^2.$prob$=0.5,W=4,N_q=45. 图 8 显示了 ρ=8nodes/314m^2 的情况下,随着网络节点数目的增大,平均每个节点在 1 个工作周期内的能量消耗情况.从图 8 可以看出:当网络中节点数目增大时,采用 CM-DCS 存储最节省能量.

Table 4 Five sensor networks topologies when ρ=8nodes/314m^2

表 **4** ρ=8nodes/314m^2 时的 5 组网络拓扑

Field size (X=Y)	198	280	323	351	370
N	1 000	2 000	2 660	3 140	3 490

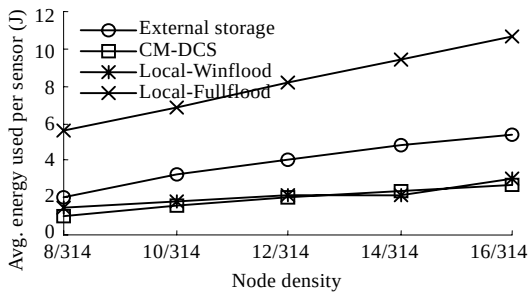

Fig.7 Effect of node density on energy

图 7 节点密度对能量消耗的影响

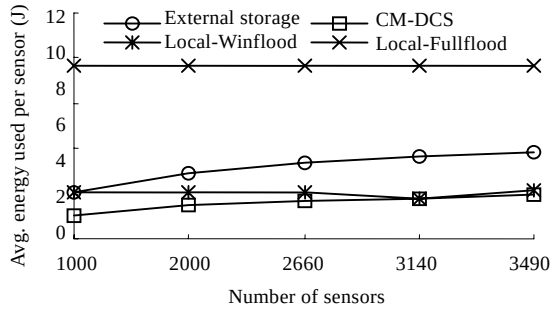

Fig.8 Effect of number of sensors on energy

图 8 节点数目对能量消耗的影响

(3) 在不同的存储方式下,考察 W 对节点能量消耗的影响.网络拓扑区域选取 370m×370m 的正方形区域,N=7000.节点随机地布置在该区域内.$prob$=0.1,W 分别取 4,8,16,32,64;N_q=40.图 9 显示了随着事件类型数目的增大,平均每个节点在 1 个工作周期内的能量消耗情况.从图 9 可以看出:W<32 时,CM-DCS 存储策略要优于本地存储策略;当 W>32 时,采用基于本地存储的 WinFlood 最节省能量.事件类型数目增大,使得网内事件信息量急剧增大,导致事件传输消耗了大量的能量.通过分析事件类型间的相关性降低监测的事件类型数目,再采用数据为中心的存储策略.分析事件类型之间的相关性将是本文未来的研究工作.

(4) 在不同的存储方式下,考察 N_q 对节点能量消耗的影响.网络拓扑区域选取 370m×370m 的正方形区域,N=7000.节点随机地布置在该区域内.$prob$=0.1,W=4.N_q=10,20,30,40,50;图 10 显示了随着 N_q 的增大,平均每个节点在 1 个工作周期内的能量消耗情况.从图 10 可以看出:当每个工作周期内查询数目增大时,CM-DCS 存储策略以及外部存储策略最节省能量.

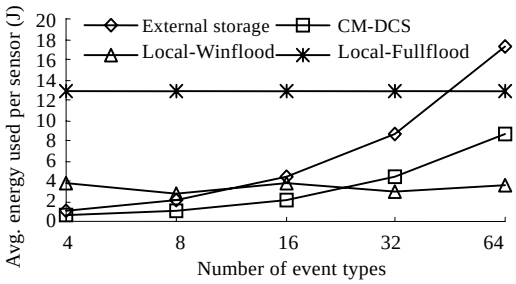

Fig.9 Effect of number of event types on energy

图 9 事件类型的数目对能量消耗的影响

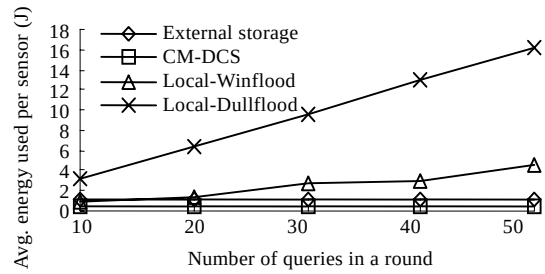

Fig.10 Effect of number of queries in a round on energy

图 10 一个工作周期内查询数目对能量消耗的影响

(5) 在不同的存储方式下,考察 *prob* 对节点能量消耗的影响.网络拓扑区域选取 370m×370m 的正方形区域,N=7000.节点随机地布置在该区域内.*prob* 分别取 0.1,0.3,0.4,0.5,0.7,W=4,N_q=30.图 11 显示随着事件发生概率的增大,平均每个节点在 1 个工作周期内的能量消耗情况:当 *prob*<0.5 时,CM-DCS 存储策略最节省能量;当 *prob*>0.5 时,基于本地存储的 WinFlood 最节省能量.

(6) 在不同的存储方式下,考察查询中的空间区域大小对能量消耗的影响.网络拓扑区域选取 370m×370m 的正方形区域,N=7000.节点随机地布置在该区域内.*prob*=0.5,W=4,N_q=35,查询的空间区域占整个网络区域的百分比分别取 6%,15%,30%,45%,65%.图 12 显示了随着查询空间区域的增大,平均每个节点在 1 个工作周期内的能量消耗情况:查询的空间区域占整个网络区域的百分比小于 20%,采用基于本地存储的 WinFlood 进行时-空查询消耗能量最少;当百分比大于 20%时,CM-DCS 存储最节省能量.

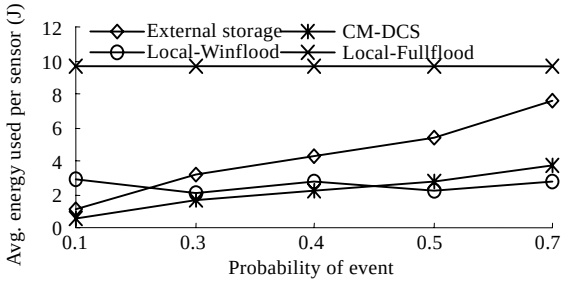

Fig.11 Effect of probability of event occuring on energy

图 11 事件发生的概率对能量消耗的影响

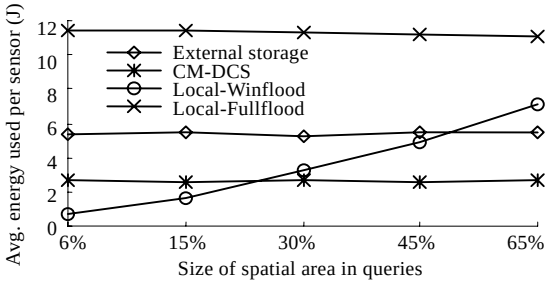

Fig.12 Effect of size of spatial area in query on energy

图 12 查询中的空间区域对能量消耗的影响

(7) 在不同的存储方式下,考察查询中的时间窗口大小对能量消耗的影响.本实验考虑的时间窗口为滑动窗口,即时间窗口的形式为$[t_{now}-h,t_{now}]$,t_{now} 是当前工作周期的编号,h 是时间窗口的宽度,即时间窗口包括 h 个工作周期.网络拓扑区域选取 370m×370m 的正方形区域,N=7000.节点随机地布置在该区域内.*prob*=0.5,W=4,N_q=35.查询的时间窗口长度分别为 10,20,30,40,50 个工作周期.图 13 显示了随着时间窗口的增大,平均每个节点在 1 个工作周期内的能量消耗情况:当时间窗口增大时,CM-DCS 最节省能量.

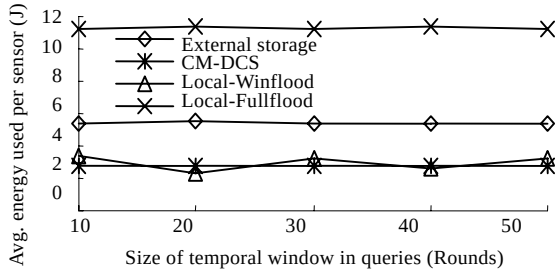

Fig.13 Effect of size of temporal window size in query on energy consumption

图 13 查询中的时间窗口大小对能量消耗的影响

5 结 论

在传感器网络环境下进行时-空查询处理,影响节点能量消耗的最主要的方面是存储的选择.决定采用哪种存储策略的因素有ρ,N,W,N_q,*prob*,Q_x,Q_{yt} 以及时间窗口的大小 t_2-t_1.本文对这些因素进行了理论与实验上的分析,结果表明:本地存储策略与外部存储策略也有其适用范围;当事件发生的概率较低时,查询比较频繁,采用 CM-DCS 存储;当事件发生的概率较高,对网络进行不频繁的时-空查询且查询的地理区域较小时,则采用本地存储.在多数情况下,采用本文提出的 CM-DCS 存储策略回答时-空查询最节省能量.

References:

[1] Li JZ, Li JB, Shi SF. Concepts, issues and advance of sensor networks and data management of sensor networks. Journal of Software, 2003,14(10):1717−1727 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1717.htm

[2] Alexandru C, Mario AN, Jörg S. A framework for spatio-temporal query processing over wireless sensor networks. In: Alexandros L, Samuel M, eds. Proc. of the 1st Int'l Workshop on Data Management for Sensor Networks in Conjunction with VLDB 2004. New York: ACM Press, 2004. 104−110.

[3] Alexandru C, Jörg S, Mario AN. An analysis of spatio-temporal query processing in sensor networks. In: Ramesh G, Cyrus S, eds. Proc. of the 1st IEEE Int'l Workshop on Networking Meets Databases in Cooperation with 21st IEEE Conf. on Data Engineering (ICDE 2005). Washington: IEEE Computer Society, 2005. 120−125.

[4] Sylvia R, Brad K, Scott S, Deborah E, Ramesh G, Li Y, Fang Y. Data-Centric storage in sensornets with GHT, a geographic hash table. Mobile Networks and Applications, 2003,8(4):427−442.

[5] Scott S, Sylvia R, Brad K, Ramesh G, Deborah E. Data-Centric storage in sensornets. ACM SIGCOMM Computer Communication Review, 2003,33(1):137−142.

[6] Wensheng Z, Guohong C, Tom LP. Data dissemination with ring-based index for wireless sensor networks. In: Kevin A, Ken C, eds. IEEE Int'l Conf. on Network Protocols (ICNP 2003). Washington: IEEE Computer Society, 2003. 305−314.

[7] Abhishek G, Jens G, John C. Resilient data-centric storage in wireless ad-hoc sensor networks. In: Arkady Z, ed. Proc. of the 4th Int'l Conf. on Mobile Data Management (MDM 2003). London: Springer-Verlag, 2003. 45−62.

[8] Deepak G, Deborah E, John H. Dimensions: Why do we need a new data handling architecture for sensor networks. ACM SIGCOMM Computer Communication Review, 2003,33(1):143−148.

[9] Deepak G, Ben G, Denis P, Deborah E, John H. An evaluation of multi-resolution storage for sensor networks. In: Ian A, Deborah E, eds. Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems. New York: ACM Press, 2003. 89−102.

[10] Benjamin G, Deborah E, Ramesh G, Sylvia R, Scott S. DIFS: A distributed index for features in sensor networks. In: Erdal C, Taieb Z, Eylem E, eds. Proc. of the 1st IEEE Int'l Workshop on Sensor Network Protocols and Applications Anchorage. Washington: IEEE Computer Society, 2003. 163−173.

[11] Xin L, Young JK, Ramesh G, Wei H. Multi-Dimensional range queries in sensor networks. In: Ian A, Deborah E, eds. Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems. New York: ACM Press, 2003. 509−517.

[12] Wendi RH, Anantha C, Hari B. Energy-Efficient communication protocol for wireless microsensor networks. In: Jay N, Ralph S, eds. Proc. of the 33rd Hawaii Int'l Conf. on System Sciences. Washington: IEEE Computer Society, 2000. 8020−8029.

[13] Chalermek I, Ramesh G, Deborah E. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: Raymond P, Sajal KD, eds. Proc. of the 6th Annual Int'l Conf. on Mobile Computing and Networks (MobiCOM 2000). New York: ACM Press, 2000. 56−67.

[14] http://ees2cy.engr.ccny.cuny.edu/zheng/pub/. 2004.

附中文参考文献:

[1] 李建中,李金宝,石胜飞.传感器网络与感知数据管理的概念、问题与研究进展.软件学报,2003,14(10):1717−1727. http://www.jos.org.cn/1000-9825/14/1717.htm

郭龙江(1973－),男,河北丰宁人,博士生,讲师,主要研究领域为数据库,数据流,传感器网络.

李建中(1950－),男,教授,博士生导师,CCF高级会员,主要研究领域为数据库,并行计算技术,传感器网络.

李贵林(1979－),男,博士生,主要研究领域为传感器网络路由协议技术.