软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2011,22(Suppl.(1)):13−22

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

支持多维查询的数据存储策略的设计*

毛科技, 何文秀, 赵小敏, 夏 明, 欧艳强, 陈庆章+

(浙江工业大学 计算机科学与技术学院,浙江 杭州 310023)

Design of Data Storage Scheme Supporting for Multi-Dimensional Query

MAO Ke-Ji, HE Wen-Xiu, ZHAO Xiao-Min, XIA Ming, OU Yan-Qiang, CHEN Qing-Zhang+

(College of Computer, Zhejiang University of Technology, Hangzhou 310023, China)

+ Corresponding author: E-mail: qzchen@zjut.edu.cn

Mao KJ, He WX, Zhao XM, Xia M, Ou YQ, Chen QZ. Design of data storage scheme supporting for multi-dimensional query. *Journal of Software*, 2011,22(Suppl.(1)):13−22. http://www.jos.org.cn/1000-9825/11002.htm

Abstract: Wireless sensor network (WSN) can effectively perceive, collect, and process the data in the monitored area. It provides an approach for human to get information. In the field of WSN, the query and storage of data is very important. It mainly used to solve how to effectively manage the distributed data in the monitored area with extremely limited resources. Recent advances in technology have made the number of sensor of the sensing model developed from single sensor to multiple sensors. The existing storage scheme for one-dimensional is not suitable for the multi-dimensional data or when it used it costs too much energy. Meanwhile the scheme has a certain degree of robustness to packet loss and node failure. Finally, the study has created experiments on the platform of Matlab and the results have shown that it has some advantages compared with the existing methods.
Key words: wireless sensor network; multi-dimensional query; *K*-D tree; area division; data storage

摘 要: 数据的存储与查询是无线传感网络(wireless sensor network,简称 WSN)研究的重要内容之一,主要解决如何利用 WSN 极其有限的资源对监测区域内的分布式数据进行有效的管理.现在的传感器节点的感知模块已由单个传感器发展到多个传感器,已有一维数据的存储策略在面对多维数据时不可用或使用后网络能量消耗过大.在 *K*-D 树的启发下提出了一种支持多维查询的数据存储策略.它能够有效地将高维相似的数据存储到同一块二维区域中,在查询时通过对查询条件的解析能快速定位到事件的存储区域,然后将查询条件路由至该区域并取回查询结果,同时该数据存储策略对网络内分组丢失和节点失效的情况具备一定的鲁棒性.在 Matlab 平台上对该方法进了验证,实验结果表明,与已有方法相比,该方法具有一定的优越性.
关键词: 无线传感网络;多维查询;*K*-D 树;区域划分;数据存储

WSN(wireless sensor network)是指由布置在监测区域内的大量节点通过无线通信自组织方式形成的网络,其基本功能是信息的获取,用户可以通过布置在监测区域的节点获取监测对象的信息,甚至还可以对某些对象

* 基金项目: 浙江省自然科学基金(Y1110649); 浙江省教育厅科研项目(Y201120121); 浙江省公益性技术应用研究计划项目(2011C21014)

收稿时间: 2011-05-02; 定稿时间: 2011-07-29

进行控制[1].

传感器节点是 WSN 的基本组成部分,一个传感器节点通常包括感知模块、处理模块、通信模块以及电源模块[2].早期的传感器节点比较简单,感知模块上通常只安装一个传感器,节点在监测时只感知一个数据量,这些被监测的事件称为一维事件,对一维事件发出的查询称为一维查询.随着传感技术、微电子技术以及嵌入式等技术的发展,现在的一个传感模块上可以集成数个传感器,达到同时监测数个指标量的效果,例如温度、湿度、光照、气压等.当节点在采集数据的时候,会取得的一个类似〈温度,湿度,光照,气压,…〉的多维数据,这些被节点监测到的事件称为多维事件,对多维事件的网络发出的查询称为多维查询.根据查询条件中查询维度数量的不同和查询区间大小的不同可将多维查询分为多维精确查询、多维部分精确查询、多维完全区域查询以及多维部分区域查询[3].

现有的支持多维查询的数据存储方法主要有 DIM(distributed index for multi-dimensional data)算法[4]和 Pool 算法[5].DIM 算法是第 1 种可以用于解决多维事件的存储与查询的算法.通过对节点、区域和事件进行编码使三者之间产生对应关系,存储时依靠事件编码将事件信息存储到对应区域的节点上,在进行多维查询时,将多维查询分解成若干个子查询,然后分别编码,最后将子查询路由到各存储区域进行查询.Pool 算法是对整个网络区域进行网格划分,然后选取与事件维度的数量相同的存储池.在对数据进行存储时,依据多维数据中的最大和第二大的维度值决定其在存储池中的行与列,然后利用相关路由协议将数据存储至该单元.查找方法与存储方法类似,首先利用最大和第二大的维度值确定数据可能存在的存储空间,然后在这些存储空间内进行查找.这些支持多维查询的数据存储策略都从理论角度给出了解决方案,但仍然存在许多不足之处,主要表现在热点问题、规模扩展性差、平均查找代价过大、可操作性不足等问题[6].

面对多维事件出现以及现有的支持多维查询的数据存储策略中存储的不足,本文提出了一种基于 *K*-D 树思想的存储方法.它能够在一定程度上解决以上问题,有效地节省节点能量,延长网络寿命.

1 支持多维查询的数据存储策略的设计

1.1 *K*-D树的介绍

K-D 树(*k*-dimensional tree)[7]是一种有效存储多维数据的树型结构,*K*-D 树是由二叉树发展而来的,它与二叉树的不同点在于它的每层都代表一个维度,可以用该维度值作为一个分裂器对相应数据做出分枝决策,顶层节点由 *K* 维中的任意一个维度决定划分,第 2 层由另一个维决定划分,以此类推,当树的深度大于数据的维度时,可以将维度重复判定.

K-D 树的好处在于将数据进行了归类,按照各维度的大小进行了划分,各维数据值越接近的数据越可能被划归到一起,这一思想正好适用于 WSN 中多维查询的问题.在设计支持多维查询的数据存储策略中,最重要的一点就是如何解决将高维数据保持局域性的映射至二维空间[8];其次,它可以进行维度的轮循,每当所有维度一轮分裂完以后,可以重复分裂,从而将各维度数据分割的更小,查找时可以更精确地定位;最后,它可以按照分裂值对二维平面进行区域划分,这一思想同样可以应用于对无线传感网络监测区域的网格划分.

本文解决数据存储问题的基本思路如下:假设被监测事件的维度为 k,先构造一棵深度为 d 的二叉树,其中 $d \geqslant k$.树的各层与数据的各维相对应,当 $d \leqslant k$ 时,在树的各层节点上,按照对应的维度以 0.5 为分割阈值进行分割;当 $k \leqslant d \leqslant 2k$ 时,各维按照 0.25 和 0.75 进行左右子树的划分,依此类推,后一轮的分裂值是前一轮分裂后的值域的中间值,最终数据均存于叶子节点上.

1.2 数据存储机制

1.2.1 事件映射至存储区

假设在一片传感网络的监测区域内,传感器节点均匀分布,节点间能通过信息的交互进行沟通,并且所有节点均知道自己的地址位置坐标.当节点监测到事件时,会产生关于事件的数据元组,用 E 表示监测到的 K 维事件:$E=\langle V_1,V_2,...,V_k\rangle$, 其中每个属性值均是规一化以后的值.对于用户发出的查询条件,用 Q 表示, $Q=\langle[L_1,U_1],$

$[L_2,U_2],...,[L_h,U_h]\rangle, h \leqslant k$; 其中的 L_i 和 U_i 分别表示查询的第 i 个属性的最小值和最大值.

在将事件映射至存储区的过程中首先要建立一棵虚拟二叉树,利用这棵树完成对存储区域的划分以及事件信息的插入.在构造这棵二叉树的过程中最为关键的是树的深度的确定,树的深度将决定整个网络内的存储片区.假设树的深度为 d,事件维度为 k,为了对事件的每一维进行一次比较划分,深度需要满足条件: $d \geqslant k$. 关于 d 的最大值,我们假设在网络分割后的每个网格中,节点会收到各个方向的数据存储和查询信息且概率相同,为了以最均衡的负载方式存储来各方向的数据,我们为每个方向上设置一个存储节点,则网格内至少应该有 8 个节点.假设整个网络内的节点数为 N,则网格的数量最多为 $N/8$,树的最大深度为 $d \leqslant \sqrt{N/8}$, 所以整个树的深度应该满足以下条件: $k \leqslant d \leqslant \sqrt{N/8}$. 在算法的实际运行过程中,$d$ 的值可根据事件维度的多少以及网络监测区域的大小灵活选取.

当插入树的深度 d 确定后,整个网络监测区域的被划分成 2^d 个存储区,当节点有感知数据产生时,需要将该数据存储到相应的存储片区中.在将事件映射到存储区的过程中,首先要对监测到的事件进行编码,对于事件 $E=\langle V_1,V_2,...,V_k\rangle$, 为其分配 d 位的二进制串,对于每一位 i,当 i 从第 1 位至第 k 位时,如果 $0 \leqslant V_i < 0.5$, 则该位编码为 0,如果 $0.5 \leqslant V_i \leqslant 1$, 则该位编码为 1;当 i 从第 k+1 位至第 $2k$ 位时,如果 $(0 \leqslant V_{i-k} < 0.25) \cup (0.5 \leqslant V_{i-k} < 0.75)$, 则该位编码为 0,如果 $(0.25 \leqslant V_{i-k} < 0.5) \cup (0.75 \leqslant V_{i-k} \leqslant 1)$, 则编码为 1,依此类推,直至 d 位二进制数分配完成,最终将事件 E 编码成 E_{code}.

在 E_{code} 中,我们规定奇数位将网络区域进行左右的对称分割,左边为 0,右边为 1;偶数位对网络区域进行上下对称分割,上边为 0,下边为 1,所以每个二进制串最终确定一个存储空间,这个存储空间 R 是由左下角 (X_{low}, Y_{low}) 和右上角 (X_{up},Y_{up}) 两组范围值定义,我们以二进制编码个数为偶数时描述计算方法,当为奇数时类似.

假设 $E_{code}=[x_1,y_1,x_2,y_2,...,x_{d/2},y_{d/2}]$, 首先决定 x 轴方向上的下界 X_{low}, 例如,如果 x_1=1,则其值将落在 0.5~1 之间;如果 x_2=0,则其值将落在 $0.5 \sim 0.5+(0.5)^2$ 之间,依此类推,可以得到: $X_{low}=x_1(0.5)+x_2(0.5)^2+...+x_{d/2}(0.5)^{d/2}$.

同理可得 y 轴方向的下界: $Y_{low}=y_1(0.5)+y_2(0.5)^2+\cdots+y_{d/2}(0.5)^{d/2}$.

因此,定义一个矩阵 M:

$$M=\begin{bmatrix} \left(\frac{1}{2}\right) & 0 \\ 0 & \left(\frac{1}{2}\right) \\ \left(\frac{1}{2}\right)^2 & 0 \\ 0 & \left(\frac{1}{2}\right)^2 \\ \vdots & \vdots \\ \left(\frac{1}{2}\right)^{d/2} & 0 \\ 0 & \left(\frac{1}{2}\right)^{d/2} \end{bmatrix},$$

可以求出存储空间 R 在 x,y 轴的下界 $R_{low}: R_{low}=[X_{low},Y_{low}]=E_{code}\cdot M$.

在确定了下界后,需要求出 R 的上界 X_{up} 和 Y_{up}, 先举例说明如何求解:如果 $x_1=1$, 则将区域竖直划分后将改变 X_{low}, 但不会被改变 X_{up}. 相反地,如果 $x_2=0$, 则划分后 X_{up} 将改变, X_{low} 不变.因此,可先对 x_1 和 x_2 进行异或运算 $x_1=x_1 \oplus 1=0$, $x_2=x_2 \oplus 1=1$, 然后可以求得 $X_{up}=1-(x_1(0.5)+x_2(0.5)^2+...+x_{d/2}(0.5)^{d/2})$, 具体算法如下:首先对 E_{code} 进行异或运算: $\overline{E_{code}}=E_{code} \oplus J_d$, 其中, $J_{d/2}=[1,1,\ldots,1]_{1\times d}$, 然后计算 $\overline{R_{up}}: \overline{R_{up}}=[\overline{X_{up}},\overline{Y_{up}}]=\overline{E_{code}}\cdot M$. 最后计算真正的上界 $R_{up}: R_{up}=[X_{up},Y_{up}]=J_2-\overline{R_{up}}$.

最后,存储空间的范围可以由 R_{low} 和 R_{up} 确定: $R=[X_{low}\sim X_{up},Y_{low}\sim Y_{up}]$.

1.2.2 事件至存储片区的路由

当计算出一个事件的存储空间范围后,需要将该事件路由至该范围内的节点上进行存储,其路由过程如下:事件的产生节点首先将自己的地理坐标和事件存储空间范围坐标比较,如果该节点恰好在该存储范围内,则将事件存储在该节点上.否则,该节点使用 GPSR 路由协议将事件传送至这一存储空间.

GPSR 协议是一种地理路由协议,使用该协议的好处是只要告知源节点和目的节点的地理位置,该协议就可以以贪婪模式将数据分组传送到目的节点[9].该协议的数据分组中除了自身携带的信息外,还应当包括存储空间的左下角和右上角坐标以及空间的中心地理位置,其数据分组格式如下:

ID	…	X_{low}	Y_{low}	X_{up}	Y_{up}	X_{mid}	Y_{mid}

其中, X_{mid},Y_{mid} 是事件对应的存储片区 R 的中心位置坐标,它们是 GPSR 协议的目标地址,在利用贪婪模式到达目标节点的过程当中经过的每个节点 $S(x,y)$ 都会用自己的地址坐标和存储空间 R 的坐标比较,当有节点满足 $(X_{low}\leqslant x\leqslant X_{up})\cap(Y_{low}\leqslant y\leqslant Y_{up})$ 的条件时,表明事件已经路由到了存储片区,接下去是要在存储片区内选择一个节点存储该事件,为了保证存储片区内的负载均衡,存储节点需要满足一定的条件.

1.2.3 存储片区内的负载均衡

当存储的事件路由到它的存储片区后,需要选择一个节点对它进行存储,选择的方法有很多,在本方案中我们这样规定:选择存储节点时选择沿着数据来源方向上离存储片区中心点 (X_{mid},Y_{mid}) 最接近的节点进行存储.如图 1 所示,假设存储片区 R 的左下角坐标为 $R_{low}(X_{low},Y_{low})$,右上角坐标为 $R_{up}(X_{up},Y_{up})$,中心点坐标为 $R_{mid}(X_{mid},Y_{mid})$,则要找的待存储节点 $S(x,y)$ 需满足目标函数: $\mathrm{Min}\sqrt{(x-X_{mid})^2+(y-Y_{mid})^2}$,同时, $(X_{low}\leqslant x\leqslant X_{up})\cap(Y_{low}\leqslant y\leqslant Y_{up})$.在整个网络内每个节点的存储空间都是有限的,我们为节点的存储负载设定一个阈值,当存储片区内某个节点的存储负载达到最大时,该节点不再接收事件,在选择存储节点时将该节点跳过,按这种方法可使片区内的存储节点从内到外依次使用,最终每个节点都会拥有存储事件的机会.

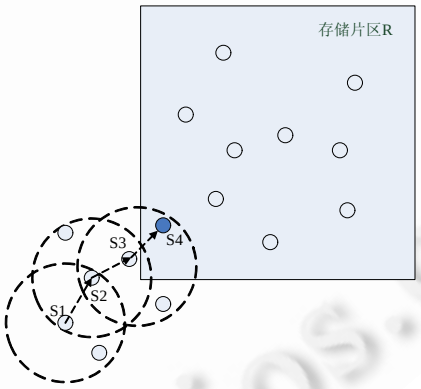

图 1 存储片区内负载均衡

1.3 多维数据的查询方法

数据的查询包括查询条件的解析和路由两部分,解析的目的是为了确定网络内具体的查找范围,路由的目的是为了向确定的查找范围发送查询条件并取回相应的数据.

1.3.1 查询条件的解析

多维查询包括精确查询和区域查询.精确查询比较简单,与事件的插入过程类似,对待查找的事件编码后会得到一个存储空间地址,将查询信息路由到该区域进行查询即可.多维区域查询包括点多维完全区域查询和多维部分区域查询,区域查询涉及的存储片区较多,所以需要先对查询条件进行解析.下面分别说明其解决办法.

1. 多维完全区域查询

所谓多维完全区域查询,是指对于网络内的监测事件 $E=\langle V_1,V_2,...,V_k\rangle$, 当用户发出的查询条件 $Q=\langle[L_1,U_1],[L_2,U_2],...,[L_h,U_h]\rangle$ 满足 $(h=k)\cap(L_i\leqslant U_i)(1\leqslant i\leqslant k)$ 时的查询.对于这类查询,我们的解析办法如下:如果插入树的深度与维度相同,则查看各维度的查询范围是否包括 0.5,如果包括,则将该维度分裂成两个子查询;如果插入树深度大于维度,则将各维度的查询范围在更小范围内进行划分.例如,假设有一个查询条件 $Q=\langle[0.3,0.6],[0.7,0.8],[0.3,0.4]\rangle$, 当其插入树的深度为 4 时,经过解析以后,查询条件被分解为以下 4 个子查询:

$$Q_1=\langle[0.3,0.5),[0.7,0.8],[0.3,0.4],[0.3,0.5]\rangle,\quad Q_2=\langle[0.3,0.5),[0.7,0.8],[0.3,0.4],[0.5,0.6]\rangle,$$

$$Q_3=\langle[0.5,0.6],[0.7,0.8],[0.3,0.4],[0.3,0.5]\rangle,\quad Q_4=\langle[0.5,0.6],[0.7,0.8],[0.3,0.4],[0.5,0.6]\rangle.$$

这 4 个子查询条件编码后的二进制串分别为 0101,0100,1101,1100,对应网络空间内的 4 块存储区域.对于编码后的结果,0101 和 0100 以及 1101 和 1100 仅最后一位不同,可以证明它们在网络空间内对应的存储区域是两块相邻的存储区域.因此,对解析后的各子查询条件编码后,如果一对子查询仅最后一位不同,则这对子查询可合并,去掉最后一位,合并后的子查询编码比原来少一位,计算存储区域时,用合并后的编码进行计算.如该例中二进制串合并后的结果为 010 和 110.这种解析合并方式的好处在于,原本需要对 4 个存储区域建立路由路径,现在只需 2 条,这样就减少了查询时的能量开销,进一步为整个 WSN 节省了能量.

2. 多维部分区域查询

对于多维部分区域查询,例如对一个监测三维事件的网络发出以下查询条件: $Q=\langle[L_1,U_1],[L_2,U_2]\rangle=\langle[0.3,0.6],[0.7,0.8]\rangle$, 对于这个查询,第 3 维上的值可为任意值,这意味着存在更多的满足条件的结果,查询的范围也会扩大.对这类查询,我们的处理办法是将其看成完全区域查询,即将查询条件 Q 转变成 $Q=\langle[0.3,0.6],[0.7,0.8],[0,1]\rangle$, 然后利用完全区域查询方法对其解析合并.

1.3.2 查询条件的路由

查询条件的路由与事件到存储区的路由相似,在经过对查询条件的分解处理后,会得到若干子查询,分别对它们编码处理,然后对编码进行合并,计算出相应的存储空间,然后利用 GPSR 协议将查询分组路由到存储区域.对应存储空间内的节点一旦接收到子查询,就将此子查询在存储空间内洪泛,所有的节点回复符合查询条件的结果至发出洪泛查询的节点,然后经由相同的路径回传给查询者.

2 仿真实验与结果分析

2.1 仿真实验参数设计及性能评价指标

在 Matlab 平台上对算法进行仿真,选取 DIM 和 Pool 为比较对象,考虑到多维查询中的多维精确点查询和多维部分精确点查询比较简单,无法明显地体现出各算法的差异性,这里主要对多维完全区域查询和多维部分区域查询中的算法处理代价进行比较.实验参数与 Pool 和 DIM 算法中实验参数一致,其具体参数如下:

1. 传感器节点设置

在实验中,传感器节点在网络内均匀放置,其数量为 300~800 个,假设每个节点的射频半径为 40m,并且在其射频范围内平均有 20 个节点,每个节点的事件存储量为 20 个.

2. 事件的产生

在本实验中,每个传感器节点平均产生 3 个事件,每个事件都是四维的,也就是说,当整个网络内有 N 个节点时,整个网络内的事件量为 $3N$.

3. 完全多维区域查询

在 Pool 算法中和 DIM 算法中,针对区域查询中区域的宽度有 4 种分类,为使比较具有合理性,本文也分别在这 4 种情况下进行比较.这 4 种分类分别称为统一宽度、边界宽度、代数宽度、指数宽度.分类的依据是 *mp* 和 *rs* 两个参数,其中 *mp* 是 *mid-point* 的缩写,假设某一维度的查询范围为⟨*low*,*up*⟩,则 *mp*=(*low*+*up*)/2,表示的是这个维度中查询范围的中间值,*rs* 是 *range size* 的缩写,指的是区域的宽度.一般而言,当给定了 *mp* 和 *rs* 的值后,这

个查询的区域可以表示成〈MAX($mp-rs/2,0$),MIN($mp+rs/2,1$)〉. 在这 4 类区域查询中,mp 的范围均在[0,1]之中,不同区域查询的 rs 值不一样.

在统一区域宽度查询中,每维的 rs 值的范围在[0,1]中选择,相当于查询条件的各维可以自由选取,没有任何约束条件;在边界区域宽度查询中,首先要确定一个边界参数,与 Pool 算法和 DIM 算法一样,在本实验中选择 0.5,则 rs 的选择范围落在[0,0.5]之间;在代数区域宽度查询中,rs 的选择没有绝对的规定,但在所有的查询条件中绝大多数的 rs 要在[0,0.25]选择;在指数区域宽度查询中,要求所有查询条件的各维的 rs 必须在[0,0.25]中选择.

4. 部分多维区域查询

在查询时,如果对某一维度没有指定任何查找条件,则这种查询被称为部分多维区域查询.针对这种情况,处理办法是将未指定的维度以[0,1]替代.在部分多维查询中,因为查询的维度数目不确定,所以可能出现多种情况,我们规定 m 维部分查询指的是有 m 个维度已给出查询条件,则在四维事件的部分多维查询中 m 的值可为 1,2,3.当 m=3 时,表示有一个维度的查询条件未给出,对于四维部分查询来说,这一维度可能出现在 4 个维度中的任何一维中,我们规定 $m@n$ 表示在 m 维的部分查询中,未指定的维度出现在第 n 维上,例如在四维事件的部分查询中,3@1 表示第 1 维查询条件未指定,后面三维查询条件指定的查询.之所以将部分查询进行这么细致的划分,是因为数据查询时可能会因为未指定维度位置的不同而出现查询代价不同的现象.

本文选用的算法性能标准与已有算法相同,分别采用以下的指标:

平均存储代价:将事件信息存储至存储节点的过程,数据包要经过多跳传递出去,平均存储代价指的是在将事件信息从监测节点发至存储节点过程中发送的平均消息数量.

平均查询代价:指的是将一个查询信息分解成多个子查询后,将这些子查询路由到存储节点并返回待查询数据过程中发送的平均消息数量.

存储热点数量:在本文中为了实现负载均衡的效果,为每个节点都设置了存储阈值,当一个节点的存储量达到该阈值后即认为该节点为存储热点.

2.2 仿真结果比较与分析

2.2.1 平均存储代价比较

按照上述实验的参数配置,分别在不同网络规模大小的情况下对 3 种算法的平均存储代价进行比较,仿真结果如图 2 所示.从图 2 中我们可以看出,3 种算法在将事件信息存入存储节点的过程中,传递的消息数量差别并不大,这是因为消息存入的过程仅仅是将事件信息从源节点发送到目的节点,而且这 3 种算法的路由基础均为 GPSR 协议.但从图 2 中我们可以看出,Pool 算法和本文算法要比 DIM 算法略好,这是因为 DIM 算法在存储的时候并没有明确的目标节点位置,而是在各节点中随机的传送,通过事件编码的比较一步一步的逼近存储位置,这在一定程度上要比其余两种方法能够传递更多的消息.

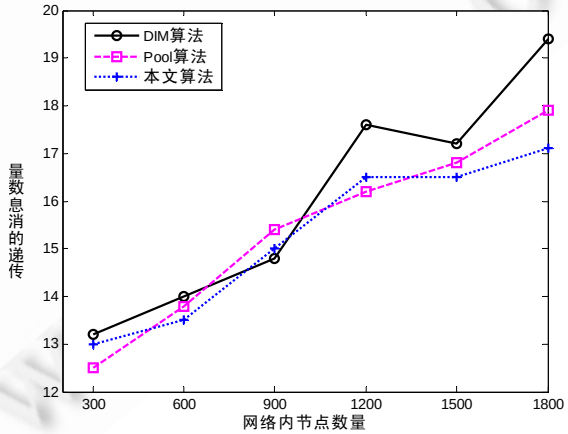

图 2 数据存储代价比较结果

2.2.2 平均查询代价比较

1. 完全多维区域查询代价的比较

实验分别在统一宽度、边界宽度、代数宽度、指数宽度这 4 种查询宽度条件的情况下对查询代价进行了比较,比较结果如图 3~图 6 所示.

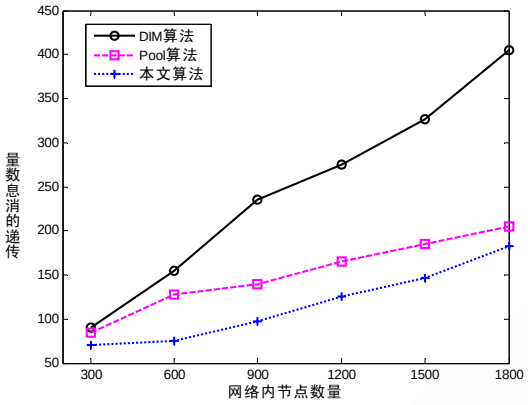

图 3 统一查询宽度的数据查询代价比较结果

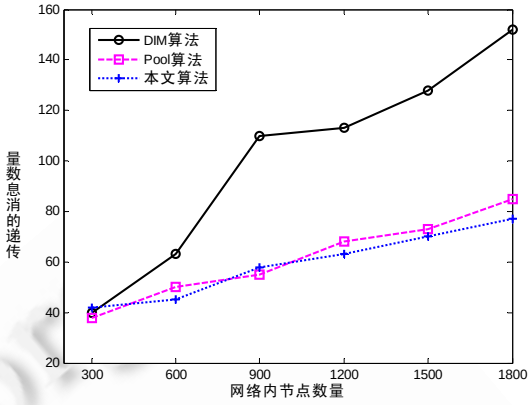

图 4 边界查询宽度的数据查询代价比较结果

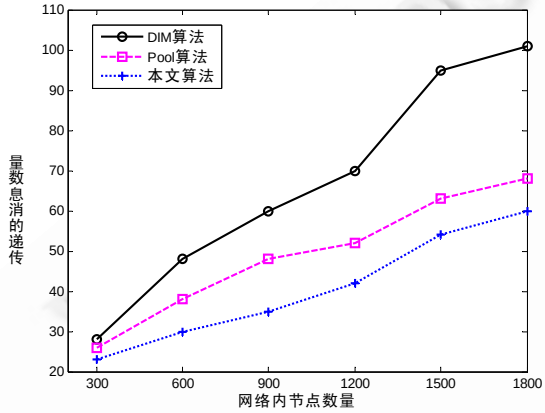

图 5 代数查询宽度的数据查询代价比较结果

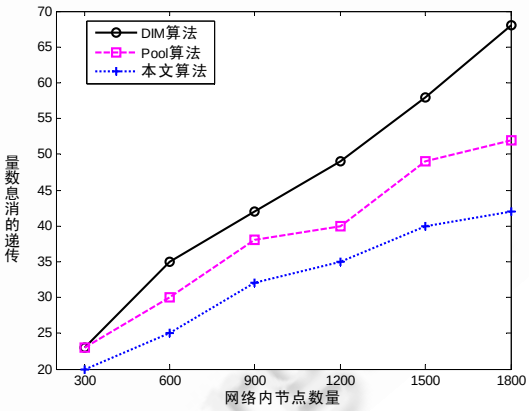

图 6 指数查询宽度的数据查询代价比较结果

从这 4 幅图的比较结果可以总结出以下两点:

(1) 对于 3 种算法而言,查询条件中各维度的查询宽度会极大地影响它们的性能.例如,图 6 中信息传递数量明显少于图 3 中的信息传递数量,这是因为数据是依据 0.5,0.25 这些分裂值进行存储的,查询宽度越小,查询解析以后就越有可能刚好落入某一查询片区内而不用进行条件的分解.相反,当查询条件是在统一宽度下时,查询需要访问更多的存储空间与索引节点才能取回数据.

(2) 从 4 幅图中可以发现对于网络规模扩展能力的适应性,本文算法最强,Pool 次之,DIM 最差.这是因为在 DIM 算法中,所有的节点均作为存储节点且均是网络构造树的叶子节点,当整个网络内的节点数量上千之后,需要在网络内首先构造出一棵庞大的树型结构,依据这棵树进行查找势必会造成消息的泛滥;在 Pool 中,数据池的数量是依据维度个数而定的,当网络规模扩大后,池的数量不会发生改变,同一消息传递到目的节点的过程中就需要路由经过更多的节点从而发生更多的消息交流;在本算法中,构造的插入树的大小是可调的,最小等于维度的个数,当网络规模扩大时,可以通过加深插入树的深度将网络化分成更小的网格,查询时依然是在小范围内进行洪泛,信息传递数量不会有太大变化.

2. 部分多维区域查询代价的比较

在进行部分多维区域查询代价比较时,为了方便比较,本实验是在网络节点为 900 个情况下进行的,分别对 3 种算法在一维、二维、三维部分区域查询进行了比较,实验结果如图 7 所示.

从图 7 可以看出,对于 3 种方法,不确定的维度数量越多,查询过程中需要传递的消息数量就越多.因为当不确定的维度越多时,满足条件的结果就越多,为了取回这些结果必然需要访问更多的存储节点,传递更多的消息.另外,在同一种维度下的部分区域查询中,DIM 算法需要查询更多节点,因为它在查询时是将查询条件随机发送的,这样很可能路由到许多无关节点上,而另外两种方法则是在源节点和目的节点地址明确的情况下利用 GPSR 协议传递的,寻找路由路径是以贪婪方式进行的,所以传递的消息数相对较少.在一维部分查询时,本文的方法与 Pool 方法接近,甚至传递消息数比它还多,这是因为当不确定维度较多时,本文方法需要访问的存储片区会增多,相邻的存储片区会合并成更大的片区,查询时在这些片区内洪泛就会带来更多的信息传递.但当不确定维度逐渐减少时,本文方法要比 Pool 好,这是因为 Pool 一直都在固定存储区域内查询,而本文方法随着确定维度的增多,查找片区会大大缩小.

在部分多维区域查询中,当不确定维度数量固定时,不确定维度的位置可以变化.图 8 是当有一个维度不确定,且其位置分别在第一、二、三、四维时,各方法的查询情况.

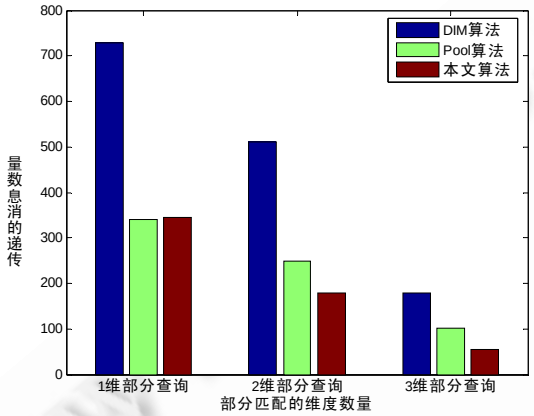

图 7 不同维度的部分多维区域查询代价比较

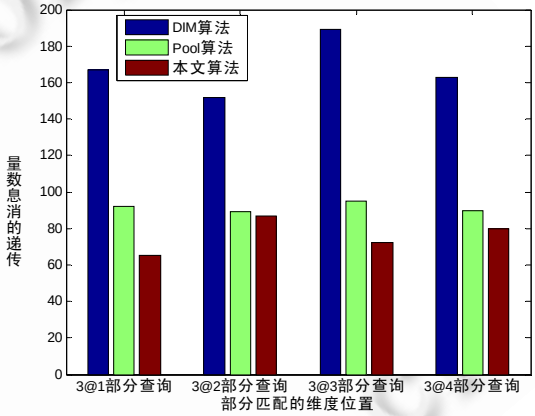

图 8 不同维度位置的部分多维区域查询

从图 8 中可以看出,当不确定维度位置变换时,对 3 种方法影响不大,其中 Pool 算法受影响最小,这是因为 Pool 算法本身就是按维度数量选择存储区域的,并且存储区域内的网格划分完全一致.当不确定维度位置变换时,各存储区域被选择的概率完全相等,唯一会发生变化的是从 sink 节点到存储区的路径长度.在 DIM 算法和本文算法中,维度位置的变换会改变存储区的个数和位置,所以从图 8 中可以发现,4 种情况下,其传递的消息数量是一种上下波动状态.

2.2.3 存储热点数量比较

好的算法应当具备良好的负载均衡能力.在本文讨论的 3 种算法中均有相应的机制.在本实验中,我们对其进行了比较.实验是在 600 个节点的情况下进行的,每个节点发出 3 个事件信息,每个节点的存储阈值为 20 个事件,图 9 是 3 种算法的比较结果.从图 9 中观察发现,DIM 热点增加最快,Pool 次之,本文算法最佳.DIM 算法的负载均衡机制是在为每个节点选择一个后备节点,当源节点存储满以后,起用后备节点进行存储.在事件的存储过程中,一个多维事件对应唯一的一个域空间,一个域空间对应唯一的一个节点,所以是由事件直接确定存储节点.这种情况下,节点的存储很快就会达到饱和.在 Pool 中,存储片区的选择依赖于最大的维度值,存储片区内存储点的选择则依赖于事件中第二大的维度值,这从一定程序上将高维相似的事件再次进行了划分,保证了单个节点的存储量不会过快增长.在本文方法中,存储片区的选择依赖于事件编码,存储节点则是选择事件来源方向上离存储片区中心位置最近的节点,当事件从不同方向发向存储区时,即使是高维相同的事件也有可能存储在

不同节点上,这更加增加了节点负载均衡能力,从图 9 中也确实可以看出本文方法中热点产生的数量增长最慢.

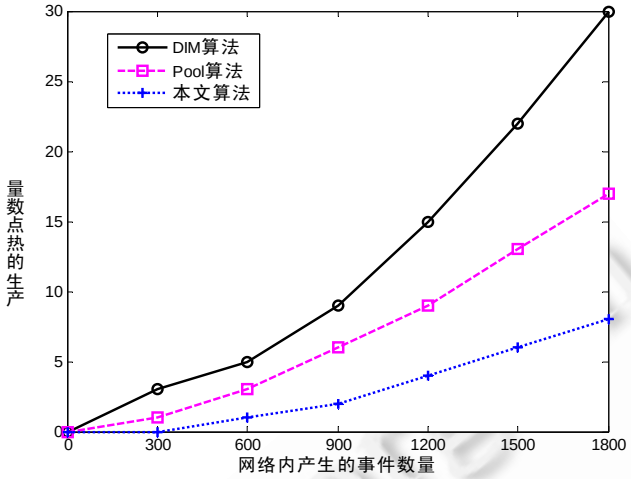

图 9 网络内热点数量的比较

3 总 结

数据的存储与查询是无线传感网络数据管理的一个重要方面,好的数据存储策略能够为查询带来便利并为网络节省大量能量,极大地延长网络寿命.针对无线传感网络中多维事件的出现,已有学者提出相关方法以保证网络能量的有效性.本文基于 *K*-D 树的思想提出了一种新的数据存储策略,通过对事件进行编码,计算出事件的存储区域,然后利用路由协议将事件信息存储到相应的存储片区.存储片区内的存储节点与事件来源方向有关,可存储在不同的节点上,这在一定程度上保证了网络内的覆载均衡效果.在查询的时候,通过对查询条件解析、编码、合并以实现对多维查询的支持,另外本文还针对分组信息在传递过程中的丢失,网络内节点失效的情况制定了相应的应对机制,以保证算法的完整性.最后,本文在 Matlab 仿真环境中对提出的方法进行了仿真验证,从实验结果看出,本文算法与 DIM 算法相比有较大优势,比 Pool 略好,但本文算法的算法复杂度更小,且在鲁棒性方面更强.

References:

[1] 孙利民,李建中,陈渝.无线传感器网络.北京:清华大学出版社,2005.

[2] 李晓维,徐勇军,任丰原.无线传感器网络技术.北京:北京理工大学出版社,2007.

[3] 于宏毅,李鸥,张效义.无线传感器网络理论、技术与实现.北京:国防工业出版社,2008.

[4] Li X, Kim YJ, Govindan R, Hong W. Multi-Dimensional range queries in sensor networks. In: Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems. Los Angels, 2003. 63–75.

[5] Chung YC, Su IF, Lee C. Supporting multi-dimensional range query for sensor networks. In: Proc. of the Int'l Conf. on Distributed Computing Systems (ICDCS). 2007. 35–44.

[6] 陶孜谨,龚正虎,卢汉新.两种新的 push-pull 平衡的大数据量无线传感网络数据分发算法.计算机研究与发展,2008,45(7):1115–1125.

[7] Bentley JL. Multi-Dimensional binary search trees used for associative searching. Communications of the ACM, 1975,18(9): 475–484.

[8] Xie L, Chen LJ, Chen DX, Xie L. A decentralized storage scheme for multi-dimensional range queries over sensor networks. In: Proc. of the 15th Int'l Conf. on Parallel and Distributed Systems (ICPADS 2009). 2009.

[9] Kung HT, Karp B. GPSR: Greedy perimeter stateless routing for wireless networks. In: Proc. of the ACM Int'l Conf. on Mobile Computing and Networking. 2000.

毛科技(1979－),男,浙江诸暨人,博士生,助理研究员,主要研究领域为无线传感网络.

何文秀(1979－),女,博士生,讲师,主要研究领域为无线传感网络.

赵小敏(1976－),男,博士生,副教授,主要研究领域为无线传感网络.

夏明(1980－),男,博士,讲师,主要研究领域为无线传感网络.

欧艳强(1985－),男,硕士,主要研究领域为无线传感网络.

陈庆章(1955－),男,博士,教授,主要研究领域为无线传感网络,CSCW.