

## 无线传感器网络中自适应数据存取\*

蔚赵春<sup>1,2</sup>, 周水庚<sup>1,2+</sup>, 肖斌<sup>3</sup>

<sup>1</sup>(复旦大学 计算机科学与工程系, 上海 200433)

<sup>2</sup>(上海市智能信息处理重点实验室, 上海 200433)

<sup>3</sup>(香港理工大学 电子计算学系, 香港)

### Adaptive Information Brokerage in Wireless Sensor Networks

YU Zhao-Chun<sup>1,2</sup>, ZHOU Shui-Geng<sup>1,2+</sup>, XIAO Bin<sup>3</sup>

<sup>1</sup>(Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China)

<sup>2</sup>(Shanghai Key Laboratory of Intelligent Information Processing, Shanghai 200433, China)

<sup>3</sup>(Department of Computing, Hong Kong Polytechnic University, Hong Kong, China)

+ Corresponding author: Phn: +86-21-55664298, Fax: +86-21-55664298, E-mail: sgzhou@fudan.edu.cn, <http://www.cs.fudan.edu.cn/~sgzhou/>

Yu ZC, Zhou SG, Xiao B. Adaptive information brokerage in wireless sensor networks. *Journal of Software*, 2008,19(1):103–115. <http://www.jos.org.cn/1000-9825/19/103.htm>

**Abstract:** Information brokerage in wireless sensor networks involves producers (such as sensor nodes) storing in storage positions a large amount of data that they have collected and consumers (e.g. base stations, users, and nodes) retrieving that information. In this paper, first, the data storage problem is formalized into a one-to-one (one producer and one consumer) model, a many-to-one ( $m$  producers and one consumer) model, and a many-to-many ( $m$  producers and  $n$  consumers) model with the goal of minimizing the total energy consumption. Second, based on the above models, two algorithms are proposed to determine the storage positions based on data rates of producers, query rates of consumers, and transmission scheme of information brokerage. The optimal data storage (ODS) scheme, a greedy algorithm, produces the global optimal data storage positions and the near-optimal data storage (NDS) scheme, an approximate algorithm, can greatly reduce the computational overhead while achieving local optimal positions. Both ODS and NDS are able to adjust the storage positions adaptively to minimize energy consumption that includes the costs of storing and querying the data. Simulation results show that NDS not only provides substantial cost benefits but also performs as effective and efficient as ODS in over 70% of the tested cases.

**Key words:** wireless sensor networks; information brokerage; data rate; geographical location

**摘要:** 数据存取,也称信息中介,是指生产者(传感器节点)将产生的感知数据按照某种策略存放在特定的位置上,而消费者(基站、用户、传感器节点)将查询请求按照对应策略路由到数据存放位置获得感兴趣的数据.利用数据

\* Supported by the National Natural Science Foundation of China under Grant No.90612007 (国家自然科学基金); the National Basic Research Program of China under Grant No.2007CB310806 (国家重点基础研究发展计划(973))

Received 2007-07-23; Accepted 2007-10-12

速率和地理位置信息来减少网状拓扑结构传感器网络中数据存取的成本。首先,依据生产者和消费者的关系建模为“一对一”(一个生产者,一个消费者)、“多对一”(多个生产者,一个消费者)、“多对多”(多个生产者,多个消费者)3种模型来对存取成本进行分析。其次,基于上述模型,提出利用数据速率和地理位置来确定数据存放位置的自适应全局最优贪婪算法 ODS(optimal data storage)和局部最优近似算法 NDS(near-optimal data storage)以及最优数据传输模式。最后,ODS 和 NDS 都依据数据速率、生产者和消费者地理位置、网络拓扑来决定存放位置,并且通过自适应地调整来减少数据存取能量消耗。实验结果表明:NDS 不仅能够节省能耗,而且在 70%的情况下达到与 ODS 相同的效果。

关键词: 无线传感器网络;数据存取;数据速率;地理位置

中图分类号: TP393 文献标识码: A

无线传感器网络中数据存取<sup>[1-4]</sup>主要研究感知数据的存储和访问机制,核心是确定节点产生的数据在网络中的存放位置,包括如何将数据存储在恰当的位置以及查询请求如何路由到存放位置并获取相关数据。实际上,这是一个信息中介(information brokerage)过程:生产者(传感器节点)将产生的感知数据按照某种策略存储在特定位置上,而消费者(基站、用户、传感器节点)将查询请求按照相应的策略路由到相关数据的存放位置,然后,满足查询条件的结果经过处理后反馈给消费者。由于传感器节点能量有限,若存取策略设计不恰当,不仅增加数据存取成本、消耗节点的宝贵能量,而且还会缩短整个网络生命周期。相反地,好的策略能够节省能耗,便于数据存储和访问,降低存取延时(delay)。因此,数据存取的基本原则是数据尽量存放在查询访问需要的地方。

影响数据存取成本的两个主要因素是存放节点(storage node)位置和数据速率。目前,大多数存取策略把整个传感器网络视为树型拓扑或网状拓扑来确定存放位置,目的是为了让消费者快捷、方便地获取数据。在树型拓扑结构<sup>[5-8]</sup>中,基站是树的根节点并且是唯一的消费者。此时,数据速率事先可确定或至少可预知,数据存放位置可以利用数学优化方法来求解。在网状(mesh)拓扑结构<sup>[9-12]</sup>中,一般存在多个生产者和消费者同时存取同一“事件”,在确定数据存放位置时,虽然考虑了生产者和消费者的地理位置,但是没有考虑数据速率。总之,现有的数据存取策略很少将数据速率和地理位置二者有机地结合起来,并且,数据存放位置没有自适应地动态变化以减少存取过程中的能量消耗。下面是一个典型的应用例子:为了监视动物园里动物的行踪,大量传感器被部署组成传感器网络,游客可以利用手持装置来查询他们感兴趣动物的事件。常规的方法是各个节点将数据发送到基站,而游客从基站查询信息。节省能耗的方法是,节点在本地处理感知数据并发送到邻近观光路径的存放节点,这样,游客经过时便可高效地获取到相关数据。

本文提出了一种自适应数据存取策略。该策略依据生产者和消费者数据速率、地理位置动态地调整数据存放位置来减少数据存取时的能量消耗。首先,本文将网络中数据存取依据生产者和消费者的关系建模为“一对一”、“多对一”、“多对多”模型。“一对一”模型依据数据产生速率和查询发生速率直接给出全局最优存放位置。“多对一”模型给出网格拓扑结构下最优数据存放位置。“多对多”模型给出近似最优存放位置。其次,基于上述模型,本文提出两种利用数据速率和地理位置来确定数据存放位置的算法:一种是求解全局最优数据存放位置的贪婪算法 ODS(optimal data storage);另一种是求解局部最优存放位置的近似算法 NDS(near-optimal data storage)。在确定了存放位置后,本文用整型规划给出数据传输模式。最后,ODS 和 NDS 都依据数据速率和地理位置来决定数据的存放位置,进而通过自适应地调整来减少数据存取的能耗。

本文第 1 节介绍相关工作。第 2 节给出自适应数据存取问题描述。第 3 节详细说明各种数据存取模型。第 4 节给出自适应数据存取算法和数据传输模式。第 5 节进行实验模拟和结果分析。最后是全文总结。

## 1 相关工作

树型结构中典型的存取策略是集中式存取<sup>[5]</sup>和本地存取<sup>[6]</sup>。集中式存取<sup>[5]</sup>是指每个节点将收集到的感知数据传输到基站存放,而查询直接从基站获取数据并进行网内查询处理。但这一策略并不是能量有效的,当网络规模很大且节点分布密集时,网络中大量的数据需要传输,靠近基站的节点会因转发数据而消耗能量太快,对网络构成瓶颈。本地存取<sup>[6]</sup>是指节点产生的感知数据存放在节点自身,而查询请求被“泛洪”到所有节点,收到请求的

节点依据查询条件反馈结果,并且数据传输途中各节点可对数据进行缓存与融合.其缺点是采用“泛洪”路由,查询请求盲目地在网络中传播,网络因为耗能太大而寿命缩短,故不适用于查询请求发生频繁的网络.

上面这些策略没有考虑到网络中数据存取的动态性,一旦数据的存放位置确定就不再发生变化.SCOOP<sup>[7]</sup>综合考虑了数据产生速率、查询发生速率以及网络拓扑结构信息,按照“数据尽量存放在查询需要的地方”的原则,由基站定期地去收集相关信息,计算出数据存放位置表 SAT(storage assignment table),然后广播给各个节点,各个节点依据 SAT 存储数据.查询时也依据 SAT 到相关节点直接获得数据.文献[8]引入了存放节点和转发节点概念,利用数据速率、数据压缩比率在树型拓扑结构中计算存放节点最优位置,从而避免把所有数据传送到基站而带来的能量消耗.

在网状结构中,节点产生的感知数据不一定存储在本地,而是利用分布式技术存储在其他节点,并采用高效的信息中介机制来协调生产者和消费者之间的关系,使得查询请求能够被满足.这些信息中介机制包括哈希映射、构建索引、数据和查询请求按一定规则路由等.

哈希映射的典型例子是 GHT(geographic hash table)<sup>[9]</sup>.它采用基于属性的命名,将感知数据属性集命名为“事件”,借助于对等系统的分布式哈希表思想,通过位置哈希函数 GHT 将“事件”名称映射为网络中存放位置  $p$ ,然后采用基于位置的路由协议将数据存储到位置  $p$  上.查询请求通过对查询“事件”进行哈希映射直接路由到相关位置.具体过程如下:将“事件”作为键值  $K$ ,通过哈希函数  $h(K)$ 将键值映射为网络中位置  $p$ ,并且将数据对  $\langle K, Data \rangle$ 存储在邻近  $p$  的节点,同样,对包含数据  $K$  的查询请求也路由到邻近  $p$  的节点去获取  $\langle K, Data \rangle$ .为避免节点故障造成数据丢失,可采用结构化复制进行多点存储.其缺点是一种盲目的数据存取策略,如果数据产生频繁而查询请求稀少,则 GHT 的存储代价会很高.

构建索引是指在数据存储时建立索引,而查询访问可通过索引获得数据,典型代表是 DIM<sup>[10]</sup>,DIFS<sup>[11]</sup>和 DIMENSIONS<sup>[12]</sup>.DIM<sup>[10]</sup>不仅采用特殊性质的哈希函数来确定数据存放位置,而且引入了数据库中的索引结构  $k-d$  树来支持多维范围查询.DIFS<sup>[11]</sup>采用位置哈希和层次索引,通过在各节点中建立特殊的类似四分树的索引结构来支持高效范围查询.每个节点存储一定地理区域的感知数据并且数据值有一定的限制范围,高层节点存储的感知数据值域范围较小,但代表的地理区域较大.DIMENSIONS<sup>[12]</sup>利用数据压缩和索引技术,提供多分辨率的数据存储和访问.在时间维上,利用小波变换对数据进行压缩存储;在空间维上,采用分布式四分树将节点组织成“金字塔”型存储结构.查询请求首先从较高层节点获得粗粒度描述,然后采用“下钻(drill-down)”技术到底层节点获得细粒度详实数据.

数据和查询请求按一定规则路由是指数据不一定存储到某一些固定节点,而是可以按照一定路径来存储,而查询请求也按照一定路径传播,只要这两条路径相交,查询请求就能得到反馈.这种思想最早起源于 Rumor<sup>[13]</sup>路由协议,即生产者和消费者都在网络中随机游走,一旦二者路径发生相交,就可以对路径进行修正和改进.依据这种思想,一种最简单的策略是:生产者将数据沿着水平方向传播并且在经过的路径上保留副本,而消费者的查询请求沿着垂直方向传播,这样,在两条路径相交的节点上就能够获得相关数据.Double Ruling<sup>[14]</sup>利用平面上点和球面上点存在一一映射以及球面上任意两个圆心为球心的球面大圆总会相交的性质来完成数据存取,其中一个大圆代表生产者存储数据的轨迹,而另一个大圆表示消费者传播查询请求的轨迹.

## 2 问题描述

### 2.1 预备知识

下面首先给出数据存取过程以及相应代价:

- 1) 生产者将某类型“事件”数据依照预定策略存储到存放节点,其代价为存储代价,记为  $C_{store}$ .
- 2) 消费者传播查询请求到存放节点获取感兴趣的数据,其代价为查询代价,记为  $C_{query}$ .

现在面临的问题是如何确定数据存放节点,使得数据存储和查询访问的代价最小.这里,代价主要是指数据传输消耗的能量,因为传感器节点运算所消耗的能量相对于数据传输微不足道.数据传输代价不仅与数据量有关,而且还与数据存取策略有关.前者由数据速率决定,而后者由生产者位置、消费者位置以及存放节点位置决

定.为了形式化地描述问题,特给出如下假设:

- 基站具有强大的计算能力而且不受能量限制.它通过从网络中收集信息而获得数据速率、生产者和消费者位置,并且负责计算数据存放位置.其余节点具有相同特征,每个节点的无线通信半径为  $R$ .网络中共有  $N$  个节点,分别标记为  $1, 2, \dots, i, \dots, N$ .节点  $i$  的地理位置坐标是  $(x_i, y_i)$ .
- 若节点  $i$  和  $j$  在相互通信范围内,二者之间可以直接传输数据,传输  $s$  单位数据所消耗能量对发送者为  $\sigma_{tr} + \delta_{tr} \cdot s$ ,对接受者为  $\sigma_{re} + \delta_{re} \cdot s$ .其中  $\sigma_{tr}, \sigma_{re}$  分别是节点准备发送和接受数据的启动代价,与传输数据量无关,而  $\delta_{tr}, \delta_{re}$  分别是节点发送和接受单位数据(字节)所消耗的能量.若二者不能直接通信,它们可借助其他节点形成多跳路由来传输数据,其传输路径长度记为  $Dis(i, j)$ .注意,这里的长度是指传输时的路由跳数.
- 为了不失一般性,“事件”随机发生在网络中的任何位置,通常网络中有  $m$  个生产者和  $n$  个消费者在时间周期  $T$  内对同一个“事件”进行存取,而且在该周期内,生产者产生的数据对消费者是有效的.生产者  $i$  以速率  $r_d(i)$  产生感知数据,每个数据大小为  $s_d$ ,而消费者  $j$  以速率  $r_q(j)$  发送查询请求并获取满足条件的感知数据,查询请求大小为  $s_q$ ,反馈给消费者数据大小为  $\alpha \cdot s_d$ ,其中,  $\alpha (\alpha \in (0, 1))$  是数据压缩比例.

## 2.2 问题描述

基于上述预备知识,自适应数据存取核心是从能量消耗最小角度来确定数据存放位置,使得数据存储和查询访问代价最小.下面给出数学模型来量化这些代价:在时间  $T$  内,对存放节点  $k$  而言,数据存储代价是指所有生产者将产生的感知数据传输到节点  $k$  去存放,其代价为  $C_{store}(k)$ ;查询访问代价是指所有消费者发送查询请求到节点  $k$  并获得已经存储的数据,其代价为  $C_{query}(k)$ .故数据存取全部代价  $C_{total}(k)$  为二者之和.

$$C_{store}(k) = \sum_{i=1}^m r_d(i) \cdot s_d \cdot Dis(i, k) \quad (1)$$

$$C_{query}(k) = \sum_{j=1}^n r_q(j) \cdot (s_q + \alpha s_d) \cdot Dis(j, k) \quad (2)$$

$$C_{total}(k) = \sum_{i=1}^m r_d(i) \cdot s_d \cdot Dis(i, k) + \sum_{j=1}^n r_q(j) \cdot (s_q + \alpha s_d) \cdot Dis(j, k) \quad (3)$$

因此,存放节点  $k$  就是在所有节点  $t$  中选择代价最小的节点,即

$$\{k \mid C_{total}(k) = \min_{1 \leq t \leq N} \{C_{total}(t)\} \ \& \ k \in \{1, 2, \dots, N\}\} \quad (4)$$

确定存放节点具有很大的挑战性,因为数据产生速率、查询发生速率、网络中两个节点之间存在的多条传输路径、生产者位置、消费者位置以及网络拓扑结构等因素都会对其产生影响.下面依据生产者和消费者的关系,建模为“一对一”、“多对一”、“多对多”3 个模型来分析数据存放位置.

## 3 数据存取模型

### 3.1 “一对一”模型

首先在一种最简单情况下决定存放节点  $k$ .如图 1 所示,网络中存在一个生产者  $i$  和一个消费者  $j$  进行数据存取.由于网络是连通的,在  $i$  和  $j$  必定存在着一条数据传输路径  $P$  经过存放节点  $k$ .假设这条路径长度是  $L$  ( $L$  为整数),即  $Dis(i, j) = L$ ,而生产者  $i$  和节点  $k$  的距离为变量  $x$  ( $x$  为整数),那么消费者  $j$  和节点  $k$  的距离为  $L - x$ .根据公式(3),数据存取全部代价为

$$C_{total}(k) = r_d(i) \cdot s_d \cdot x + r_q(j) \cdot (s_q + \alpha s_d) \cdot (L - x) = \beta r_q(j) \cdot s_d \cdot L + (r_d(i) - \beta r_q(j)) \cdot s_d \cdot x \quad (5)$$

其中,  $\beta = (s_q + \alpha s_d) / s_d$ .

公式(5)的前半部分表示生产者  $i$  将产生的数据存储于节点  $k$  所花费的代价;而后半部分有两种代价:一种是消费者  $j$  发送查询请求到节点  $k$  的代价,另一种是节点  $k$  反馈查询结果给消费者的代价.

依据公式(4),存放节点就是传输路径上所有节点中代价最小的节点,即  $C_{total}(k) = \min_{t \in P} \{C_{total}(t)\}$ .因此,问题转

换为在给定数据产生速率  $r_d$ 、查询发生速率  $r_q$ 、数据包大小  $s_d$  和数据压缩比率  $\beta$  条件下求解公式(5)的极小值. 为了求解极小值,引入下面的函数并给出相应的定理:

$$f(x) = \beta r_q s_d L + (r_d - \beta r_q) s_d x \tag{6}$$

其中,  $x \in \{0, 1, \dots, L\}$ .

**定理 1.** 网络中给定一个生产者和一个消费者,代价最小的数据存取策略如下:如果  $r_d \leq \beta r_q$ ,则数据存放在消费者节点上;否则,数据存放在生产者节点上.

证明:函数(6)是一次线性函数  $f(x) = ax + b$ ,其中,  $x$  是正整数且  $a = (r_d - \beta r_q) s_d, b = \beta r_q s_d L$ .下面根据系数  $a$  来讨论函数极值:

- 1) 当  $r_d = \beta r_q$  时,  $a = 0, f(x) = r_d s_d L$  是常量,极小值与变量  $x$  的取值无关,可选择生产者或消费者节点存放数据,即  $k = i$  或  $k = j$ ;
- 2) 当  $r_d > \beta r_q$  时,  $a > 0$ ,函数  $f(x)$  单调递增,在  $x = 0$  时取得极小值,故选择生产者作为数据存放节点,即  $k = i$ ;
- 3) 当  $r_d < \beta r_q$  时,  $a < 0$ ,函数  $f(x)$  单调递减,在  $x = L$  时取得极小值,故选择消费者作为数据存放节点,即  $k = j$ .  $\square$

**定理 2.** 在“一对一”模型中,数据存取代价最小路径是生产者 and 消费者之间最短路径.

证明:继续分析函数(6)的极小值.在情况 1)和情况 3)中,函数极小值是  $f_{\min}(x) = r_d s_d L$ ;在情况 2)中,函数极小值是  $f_{\min}(x) = \beta r_q s_d L$ .总之,函数极小值都与路径长度  $L$  成正比, $L$  越小,数据存取代价也就越小.因此,代价最小路径是二者之间最短路径.  $\square$

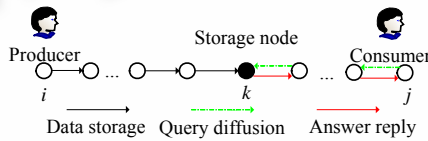


Fig.1 One-to-One model

图 1 “一对一”模型

### 3.2 “多对一”模型

下面考虑网络中存在多个生产者和一个消费者的情形,如图 2 所示.

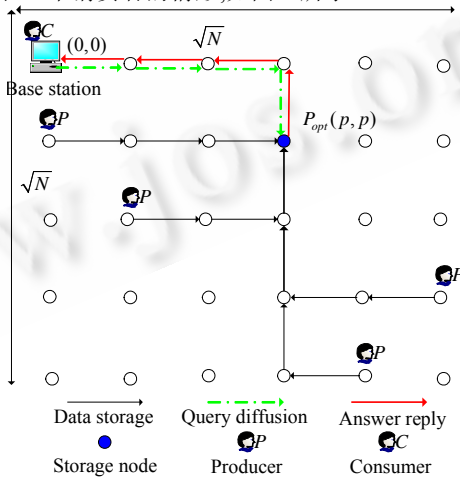


Fig.2 Many-to-One model

图 2 “多对一”模型

给定  $m$  个生产者和一个消费者,根据公式(4)确定存放节点  $k$ ,使得数据存取代价最小,即每个生产者到存放节点的加权最短路径之和加上消费者到存放节点的加权最短路径之和最小.故全部代价变为

$$C_{total}(k) = \sum_{i=1}^m r_d(i) \cdot s_d \cdot Dis(i,k) + r_q(j) \cdot (s_q + \alpha s_d) \cdot Dis(j,k) \tag{7}$$

首先在规则的网格状结构中分析数据存取代价.假设  $N$  个节点均匀分布在  $\sqrt{N} \times \sqrt{N}$  的网格中,除了周围四边节点外,其余节点均有上下左右 4 个邻居.消费者(基站)位于网格最左上角的原点位置,而生产者均匀地分散在网格中,它们产生的数据通过最短路径传输到数据存放位置  $P_{opt}(x,y)$ .由于网格结构的对称性和生产者均匀分布可知, $P_{opt}(x,y)$ 一定位于对角线上,否则从平均角度而言,对角线一侧的节点到  $P_{opt}(x,y)$ 的距离要比另一侧大.因此,存放节点坐标为  $(p,p)$ .由于消费者位置固定并且事先已知,用  $l_c(p)$ 表示从消费者到  $P_{opt}(p,p)$ 的最短路径长度,故  $l_c(p)=x+y=2p \cdot \bar{l}_p(p)$ 表示从生产者到存放节点  $P_{opt}(p,p)$ 的平均最短路径长度,这里直接给出结论<sup>[15]</sup>:

$$\bar{l}_p(p) = \frac{1}{N-1} [N \cdot \sqrt{N} - N - 2N \cdot p + 2\sqrt{N} \cdot p^2 + 2\sqrt{N} \cdot p] \tag{8}$$

将  $Dis(i,k) = \bar{l}_p(p)$ ,  $Dis(j,k) = l_c(p)$ 引入公式(7)可得:

$$C_{total}(p) = \frac{s_d}{N-1} \sum_{i=1}^m r_d(i) [N \cdot \sqrt{N} - N - 2N \cdot p + 2\sqrt{N} \cdot p^2 + 2\sqrt{N} \cdot p] + 2r_q(j) \cdot (s_q + \alpha s_d) \cdot p \tag{9}$$

利用  $\sqrt{N} + 1 \approx \sqrt{N} - 1 \approx \sqrt{N}$ ,  $N-1 \approx N$  进一步简化:

$$C_{total}(p) = s_d \cdot \sum_{i=1}^m r_d(i) \left[ \sqrt{N} - 2p + \frac{2(p^2 + p)}{\sqrt{N}} \right] + 2(s_q + \alpha s_d) \cdot r_q(j) \cdot p \tag{10}$$

依据公式(4),数据存放位置  $P_{opt}(p,p)$ 满足  $\min_p \{C_{total}(p)\}$ ,故问题转换为求解使  $C_{total}(p)$ 取得极小值的  $p$ ,因为公式(9)是关于变量  $p$  的 2 次函数,令其一阶导数为 0 可求解出  $p$ .

$$p = \sqrt{N} \cdot \left[ s_d \cdot \sum_{i=1}^m r_d(i) - (s_q + \alpha s_d) \cdot r_q(j) \right] / \left[ 2s_d \cdot \sum_{i=1}^m r_d(i) \right] \tag{11}$$

如果  $p < 0$ ,则意味着数据存放位置  $x$  和  $y$  坐标都小于 0,最优数据存放位置是基站,所有生产者把产生的数据直接送到基站存储.此时,消费者的查询代价为 0,而生产者的存储代价为  $\sqrt{N} \cdot s_d \cdot \sum_{i=1}^m r_d(i)$  ( $\sqrt{N}$  是生产者到基站的平均传输路径长度).如果  $p > 0$ ,依据公式(11)求解出最优存放位置  $(p,p)$ .

### 3.3 “多对多”模型

现在考虑最一般的情况,如图 3 所示,网络中有  $m$  个生产者和  $n$  个消费者.为了求解数据存放节点  $k$ ,引入  $l_p(i)$ ,  $l_c(j)$ 分别表示从生产者  $i$  和消费者  $j$  到节点  $k$  的最短路径长度,因此,数据存取全部代价为

$$C_{total}(k) = \sum_{i=1}^m r_d(i) \cdot s_d \cdot l_p(i) + \sum_{j=1}^n r_q(j) \cdot (s_q + \alpha s_d) \cdot l_c(j) \tag{12}$$

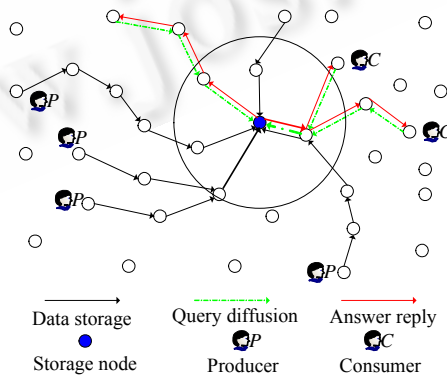


Fig.3 Many-to-Many model

图 3 “多对多”模型

将传感器网络抽象成图  $G=(V,E)$ ,其中,  $V=s_1,s_2,\dots,s_N$  表示节点集合,节点  $s_i$  的位置坐标为  $(x_i,y_i)$ ;  $E$  是由节点之间通信连接构成的边集合.若两个节点  $s_i,s_j$  在彼此通信范围之内,则它们之间就建立了一条边,并对应单位数据传输代价  $c(s_i,s_j)$ .为了求解最优数据存放节点  $k$ ,进一步把数据速率和数据包大小折算成代价,即对生产者  $i$ ,它到节点  $k$  的代价  $c_{pro}(i,k)=r_d(i)\cdot s_d\cdot l_p(i)$ ;对于消费者  $j$ ,它到节点  $k$  的代价  $c_{con}(j,k)=r_q(j)\cdot (s_q+\alpha s_d)\cdot l_c(j)$ .如果网络中节点分布比较密集而且均匀,则节点之间的最短路径长度和欧氏距离近似成正比;对于节点分布不均匀的网络,文献 [16]给出了节点间最短路径长度和欧氏距离之间的关系.因此,节点之间的最短路径长度可用欧氏距离来近似地代替,那么这个求解最优存放位置的问题就转换为求  $m+n$  个点的 1-median<sup>[17]</sup>,即

$$\min_{k \in \mathbb{R}^2} \left\{ \sum_{i=1}^m c_{pro}(i,k) + \sum_{j=1}^n c_{con}(j,k) \right\}.$$

1-median 是经典的 NP-Complete 问题,研究者提出了许多近似算法来减少运算的复杂性<sup>[17]</sup>,这里给出一种简化的近似算法.依据公式(4),数据存放节点就是使公式(12)取得极小值的节点.为了求解近似极小值,引入如下函数:

$$f(x,y) = s_d \cdot \sum_{i=1}^m r_d(i) \cdot [(x-x_i)^2 + (y-y_i)^2] + (s_q + \alpha s_d) \cdot \sum_{j=1}^n r_q(j) \cdot [(x-x_j)^2 + (y-y_j)^2] \quad (13)$$

现在,求解近似最优存放位置转换为求解点  $(x_0,y_0)$  使得函数(13)在  $(x_0,y_0)$  处取得极小值.因为函数(13)是定义在  $xy$  平面上封闭空间内的二维连续函数,对其微分并令其对  $x$  和  $y$  的偏导数为 0,可求出极小值点  $(x_0,y_0)$ ,并给出定理说明函数(13)在位置  $(x_0,y_0)$  取得极小值.

$$x_0 = \frac{\sum_{i=1}^m r_d(i) \cdot x_i + \beta \sum_{j=1}^n r_q(j) \cdot x_j}{\sum_{i=1}^m r_d(i) + \beta \sum_{j=1}^n r_q(j)}, y_0 = \frac{\sum_{i=1}^m r_d(i) \cdot y_i + \beta \sum_{j=1}^n r_q(j) \cdot y_j}{\sum_{i=1}^m r_d(i) + \beta \sum_{j=1}^n r_q(j)} \quad (14)$$

其中,  $\beta=(s+\alpha s)/s$ .

**定理 3.** 函数(13)在  $(x_0,y_0)$  处取得极小值.

证明:为确定函数  $f(x,y)$  在  $(x_0,y_0)$  处的极值类型,计算函数的二阶偏导数,并依据  $d = f''_{xx} \cdot f''_{yy} - f''_{xy}{}^2$  在  $(x_0,y_0)$  处的取值来确定极值类型.经计算可知:

$$f''_{xx}(x,y) = 0, f''_{xx}(x,y) = f''_{yy}(x,y) = 2 \left[ \sum_{i=1}^m r_d(i) + \sum_{j=1}^n r_q(j) \right] > 0, d(x_0,y_0) > 0, f''_{xx}(x_0,y_0) > 0,$$

所以函数在  $(x_0,y_0)$  处取得极小值. □

## 4 算 法

### 4.1 最优数据存放(ODS)

ODS 是一种贪婪式算法.它从所有节点中选择最优数据存放节点,选择标准是依据公式(3)所计算的总体代价.它采用试探并验证策略,把每个节点作为存放节点,计算其数据存储和查询访问代价总和,最后从所有节点中选择代价最小的.值得注意的是,本算法能够利用数据速率、数据大小、生产者和消费者地理位置等参数在“一对一”情况下直接给出最优存放位置.

在算法 1 的伪码中,  $PS$  表示生产者集合,  $PS_p$  表示集合中第  $p$  个生产者;而  $CS$  表示消费者集合,  $CS_c$  表示集合中第  $c$  个消费者.本算法首先判断网络是否满足“一对一”模型,若是,则直接给出最优存放位置.然后判断是否满足“多对一”模型,若满足,就给出初始参考存放位置.在“多对多”模型中,测试所有节点并计算其总体代价,选择代价最小节点作为最优数据存放节点(见代码 4~11 行).数据存取代价由两部分构成:一部分是生产者将数据存储到存放节点所花费的存储代价,可依据公式(1)计算(见代码第 6、第 7 行);另一部分是消费者传播查询请求到存放节点并获得相关数据所花费的查询代价,可依据公式(2)计算(见代码第 8、第 9 行).

该算法复杂性为  $O(N \cdot (m+n) \cdot L)$ ,其中,  $N$  为网络中节点个数,  $m$  是生产者个数,  $n$  是消费者个数,而  $L$  是网络中任意两节点路由计算复杂度(与具体路由协议相关,这里不予讨论).显然,算法效率低下,因为没有利用地理位置

信息以及数据速率来减少搜索空间,只是盲目地去计算每个节点的代价,这样不仅费时,而且在大规模网络中是不可行的.

#### 算法 1. 最优数据存放(ODS).

输入:节点集合  $NS$ ,生产者集合  $PS$ ,消费者集合  $CS$ ,节点个数  $N$ ,数据大小  $s_d$ ,查询大小  $s_q$ ,数据压缩比例  $\alpha$ ;

输出:最优的数据存放节点  $ods$ .

- (1) 若满足“一对一”模型,则直接给出最优存放位置  $ods$ ;
- (2) 若满足“多对一”模型,给出初始参考存放位置,并在其邻居节点中选择代价最小的节点作为存放位置.
- (3) 初始化: $C_{\min}=+\infty, ods=null$
- (4) For  $k=1$  to  $N$
- (5)  $C_{store}(k)=0, C_{query}(k)=0, C_{total}(k)=0$
- (6) For  $p=1$  to  $|PS|$
- (7)  $C_{store}(k)+=r_d(p) \cdot s_d \cdot Dis(PS_p, k)$
- (8) For  $c=1$  to  $|CS|$
- (9)  $C_{query}(k)+=r_q(c) \cdot (s_q + \alpha s_d) \cdot Dis(CS_c, k)$
- (10)  $C_{total}(k)+=C_{store}(k)+C_{query}(k)$
- (11) If  $C_{total}(k) < C_{\min}$  then  $C_{\min}=C_{total}(k), ods=k$
- (12) 选择  $ods$  作为存放节点

## 4.2 近似最优数据存放(NDS)

近似最优数据存储 NDS(见算法 2)没有采用 ODS 的盲目搜索,而是采用启发式搜索来避免测试所有节点,通过试探部分节点来获得局部最优.其核心思想是:首先利用 3 种模型计算出一个初始参考位置,然后在该位置周围的邻居节点中选取代价最小的节点作为近似最优数据存放位置.

该算法首先对生产者和消费者的关系进行判断:若满足“一对一”模型(见代码第 2、第 3 行),直接给出最优数据存放位置;若满足“多对一”模型,则依据公式(11)计算出数据存放的参考位置;若满足“多对多”模型,则依据公式(15)计算出数据存放的参考位置.然后,寻找距离该位置最近的节点作为初始数据存放节点(见代码第 6 行).最后在初始节点的邻居中寻找存取代价最小的节点作为存放节点(见代码第 9~17 行).注意,初始化时选择离参考位置最近的节点作为数据初始存放节点是为了避免其邻居节点代价都比初始节点高(见代码第 7 行).

NDS 充分利用基站的强大计算能力,首先计算出参考存放位置,然后在其邻居节点范围内搜索.相对于盲目搜索,搜索范围从整个网络减小到 1 跳范围之内.与 ODS 相比,虽然额外增加计算参考位置,却极大地减少了搜索空间.下面分析 NDS 复杂性和近似比:假设网络覆盖区域面积是  $S$ ,节点无线通信范围是  $R$ ,那么节点平均密度为  $N/S$ ,因此,在初始节点 1 跳范围内的邻居节点的平均个数是  $N\pi R^2/S$ ,平均而言,NDS 复杂性是 ODS 的  $\pi R^2/S$ .在“一对一”模型中,NDS 和 ODS 近似比为 1:1,在“多对一”和“多对多”模型中,近似比为  $S/(\pi R^2)$ .

#### 算法 2. 近似最优数据存放(NDS).

输入:节点集合  $NS$ ,生产者  $PS$ ,消费者  $CS$ ,数据大小  $s_d$ ,查询大小  $s_q$ ,数据压缩比例  $\alpha$ ;

输出:近似最优的数据存放节点  $nds$ .

- (1)  $\beta=(s_q+\alpha s_d)/s_d$
- (2) 判断是否满足“一对一”模型,即 $|PS|=1$  并且 $|CS|=1$
- (3) 若满足,如果  $r_d \geq \beta r_q$  则输出  $PS_0$ ;否则,输出  $CS_0$
- (4) 判断是否满足“多对一”模型,即 $|PS|>1$  并且 $|CS|=1$
- (5) 若满足,则依据公式(11)计算初始参考位置 $(x_0, y_0)$ ,否则,依据公式(15)计算初始参考位置 $(x_0, y_0)$
- (6) 确定离 $(x_0, y_0)$ 最近的节点  $N_{init}$
- (7) 初始化: $nds=N_{init}, C_{\min}=C_{total}(N_{init})$
- (8) 确定  $N_{init}$  的邻居节点个数  $NUM_{neighbor}$



- (9) For  $k=1$  to  $NUM_{neighbor}$
- (10)  $C_{store}(k)=0, C_{query}(k)=0, C_{total}(k)=0$
- (11) For  $p=1$  to  $|PS|$
- (12)  $C_{store}(k)+=r_d(p) \cdot s_d \cdot Dis(PS_p, k)$
- (13) For  $c=1$  to  $|CS|$
- (14)  $C_{query}(k)+=r_q(c) \cdot (s_q + \alpha s_d) \cdot Dis(CS_c, k)$
- (15)  $C_{total}(k)+=C_{store}(k)+C_{query}(k)$
- (16) If  $C_{total}(k) < C_{min}$  then  $C_{min}=C_{total}(k), nds=k$
- (17) 选择  $nds$  作为存放节点

### 4.3 最优数据传输模式

第 4.2 节给出了最优或近似最优的存放位置,但在实际网络中,对应位置并不一定有传感器节点,本文采用基于位置的路由协议 GPSR(greedy perimeter stateless routing)<sup>[18]</sup>来寻找距离存放位置最近的节点作为数据存放节点.当存放节点负载较大时,生产者数据传输经过的路径上备份,而且存放节点收到数据后广播一次,邻居节点收到广播后在本地备份数据.这相当于在以存放节点为圆心、半径为节点通信范围的圆形区域内存储数据.这样不仅均衡了负载,增强了网络健壮性,而且也利于消费者获得数据.

对于给定的参数,获得数据存放节点后,需要进一步去求解数据传输模式  $T$ ,使得在该模式下数据存取能量消耗最小.所谓传输模式,是指生产者和消费者如何在其他节点协助下完成数据存取,即生产者如何将数据传输到存放节点和消费者如何把查询请求传输到存放节点并获得感兴趣的数据.给定传感器节点集合  $V$  和存放位置  $P_{opt}$ ,下面用整型规划方法给出最优数据传输模式  $T(P_{opt})$ .

考虑在  $V$  上定义的完全有向图  $G'$ ,对于  $G'$  中每条边  $u'v'$  都对应一个权重,若它在生产者  $i$  存储数据路径上,则权重为  $r_d(i) \cdot s_d$ ;若它在消费者  $j$  查询访问路径上,则权重为  $r_q(j) \cdot (s_q + \alpha s_d)$ .为了描述数据传输模式,引入如下变量:  $x_{u'v'}$  表示单位时间内边  $u'v'$  传输的数据量,而  $X_{u'v'}^{(k)}$  表示由节点  $s_k(k=1,2,\dots,N)$  产生而经由边  $u'v'$  传输并存储到  $P_{opt}$  的数据量.每个节点可用能量为  $e(u')$ ,而且它必须发送自己产生的  $F_k(k=1,2,\dots,N)$  单位的数据到存放位置  $P_{opt}$ .下面,通过整型规划求解出一个最优数据传输模式:

- (1) 最小化  $\sum_{u' \in V'} \sum_{v' \in V'} x_{u'v'} \cdot c_{u'v'}$  满足如下条件:
- (2)  $F_k + \sum_{u' \in V'} X_{u'v'}^{(k)} = \sum_{v' \in V'} X_{kv'}^{(k)}$
- (3)  $\sum_{v' \in V'} x_{u'v'} \cdot c_{u'v'} \leq e(u')$
- (4)  $x_{u'v'} > 0, c_{u'v'} > 0, F_k > 0$

其中:(1)指明了优化目标,使得数据存储和查询访问消耗能量总和最小;条件(2)说明了每个节点数据量守恒原则,即流入每个节点的数据量再加上节点本身产生的数据量等于流出该节点的数据量;条件(3)限制了每个节点能量消耗必须在节点可用能量之内;条件(4)针对传感器网络中的数据存取,给出了各个变量的限制.

## 5 实验和结果分析

本节对自适应存取策略与其他策略进行比较,并且测试了节点个数  $N$ 、生产者个数  $m$ 、消费者个数  $n$ 、数据速率、数据压缩比例等不同参数对性能的影响.除了 ODS 和 NDS 以外,本文还比较了其他 3 个相关的数据存取策略 LDS(local data storage),CDS(centralized data storage)和 GHT(geographic hash table).局部存取策略(LDS)中生产者将产生的数据存储在本节点,而消费者将查询请求“泛洪”到网络中而获得相关的数据.由于“泛洪”代价巨大,在实验分析中省略了该策略.集中式存取策略(CDS)中,生产者将数据存储到基站而消费者从基站获得数据,本文选择网络区域左上角的节点作为基站.位置哈希存取策略(GHT)<sup>[9]</sup>中生产者应用位置哈希函数将数据映射到网络某个位置,而消费者采用同样的函数来获取数据.GHT 采用密码学中 MD5 哈希函数作为位置哈希函数,以生产者和消费者的地理位置以及数据速率为函数的输入,将产生的 128 位报文转换为长整型,并对

网络长度和宽度分别取模获得存放位置.因为所有策略都依赖地理位置信息,故采用 GPSR<sup>[18]</sup>作为底层路由协议.

本文模拟环境以 Crossbow 公司生产的 MICA2 节点为标准.节点被随机地部署到 400m×400m 的网络区域中,基站位于区域最左上角.下面是实验参数的具体取值<sup>[19]</sup>: $R=50m, s_d=s_q=40Byte, \sigma_{tr}=0.645mJ, \sigma_{re}=0.387mJ, \delta_{tr}=0.0144mJ/Byte, \delta_{re}=0.00864mJ/Byte, \alpha=0.5$ .实验模拟过程分为两个阶段:第 1 阶段中,基站收集网络参数,包括生产者和消费者的位置、数据速率,然后用这些参数计算出数据存放位置和数据传输模式并广播到各个节点.第 2 阶段中,生产者和消费者通过数据存放节点按照最优数据传输模式进行数据存取.为获得自适应数据存取策略在连续查询中的长期收益,第 1 阶段所花费的代价可以在后续阶段加以弥补,所以实验结果中忽略了第 1 阶段的代价.

下面通过模拟实验在网络中存在  $m$  个生产者和  $n$  个消费者的情形下比较 ODS 和 NDS 的性能.为了提高实验结果的准确性,本文取 100 次实验平均结果.每次实验在给定  $N$  个网络节点中随机地选择  $m$  个生产者和  $n$  个消费者,并随机设置数据速率为 $[1,20]$ .引入  $f_c$  表示 NDS 中初始节点的邻居节点个数占网络节点总数的百分比,而  $AVG(f_c)$  表示 100 次实验的平均值.本实验中记录了  $f_c$  的最大值并发现其在 3%~6%之间,故只给出 100 次平均结果.  $P_{equal}$  表示在 100 次实验中 ODS 和 NDS 效果相同的次数. $E_{AVE}$  表示 100 次实验的平均能量消耗, $Error=[E_{AVE}(NDS)-E_{AVE}(ODS)]/E_{AVE}(ODS)$  表示二者之间的误差.

图 4 显示,在  $N=400, m=150$  的条件下,网络中平均能量消耗和消费者数目  $n$  的变化关系.随着消费者个数从 0 增加到 50,各种策略所消耗能量都在增加,但 ODS 和 NDS 增加较为缓慢.NDS 平均效果几乎与 ODS 相同,可以从图中两条曲线重合看出.这是因为当网络节点分布比较均匀时,大多数情况下,NDS 初始节点的邻居节点包含最优数据存放节点.表 1 给出了 ODS 和 NDS 的细微差别.

Table 1 Energy consumption ( $N=400, m=150$ )

表 1 能量消耗表( $N=400, m=150$ )

$n$	$AVG(f_c)$	$P_{equal}$	Error (%)	ODS	NDS
1	3.79	85	0.089 8	5 497.953 6	5 502.890 4
10	3.72	99	0.003 7	5 873.962 6	5 874.182 3
20	3.32	75	0.205 5	6 951.819 5	6 966.108 7
30	3.88	89	0.021 2	7 104.627 1	7 106.136 4
40	3.84	87	0.015 3	7 754.697 4	7 755.885 3
50	5.62	86	0.002 2	8 088.512 8	8 088.691 7

图 5 显示,在  $N=400, n=10$  的条件下,网络中平均能量消耗随生产者数目的变化趋势.随着生产者个数从 1 增加到 200,各种策略所消耗能量逐渐增加,而 NDS 平均效果几乎与 ODS 相同(图中两曲线重合)且优于 CDS 和 GHT,因为 NDS 在 1 跳范围内能找到最优存放节点.限于篇幅,这里不再给出 ODS 和 NDS 的细微差别.图 4 和图 5 中,GHT 能量消耗比 CDS 要小,这是因为,CDS 中存放节点是固定的,而 GHT 会动态变化.

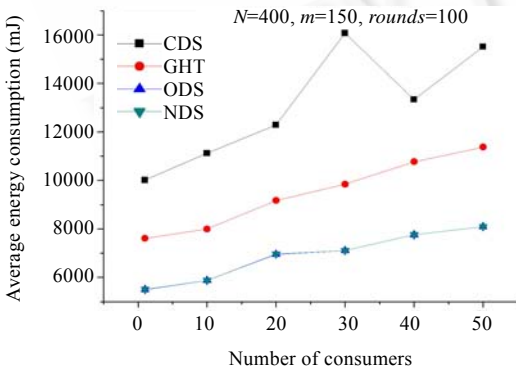


Fig.4 Number of consumers vs. energy consumption

图 4 不同数目消费者的能量消耗

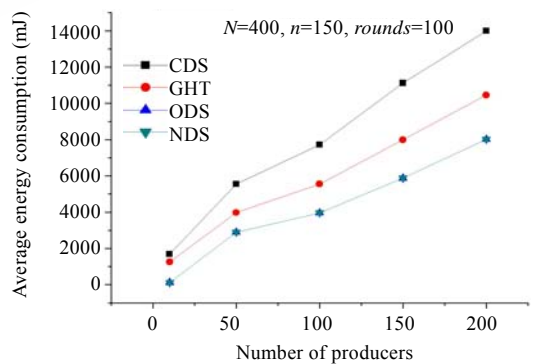


Fig.5 Number of producers vs. energy consumption

图 5 不同数目生产者的能量消耗

图 6 是在  $m=50, n=1$  的条件下,网络中平均能量消耗和节点数目  $N$  的变化直方图.从中可以看出,能量消耗并不一定随着节点数目的增加而增多,这是因为随机设置数据速率的缘故.从直方图中可以看出,ODS 和 NDS 的能量消耗几乎相同,而且优于其他策略.为了表现细微差别,图中标注出  $AVG(f_c)$ 和  $P_{equal}$ .

从上面分析可以看出,ODS 和 NDS 的能量消耗差别微乎其微,即使最大误差也不超过 5%.下面来比较二者在相同运行环境下运行时间的差别.CDS,GHT,ODS,NDS 运行在一台配置为 1.7GHz CPU,256M 内存的 PC 上.图 7 显示了在  $m=200, n=50$  的条件下,运行时间随着节点数目  $N$  的变化关系.随着  $N$  的增加,ODS 运行时间急剧增加,NDS 缓慢增加,而 CDS 和 GHT 几乎不发生变化而且趋势一致(两条曲线重合).这是因为,随着节点数目的增加,节点密度增加, ODS 要计算每一个节点,而 NDS 初始节点的邻居节点数目也在增加,所以 NDS 缓慢增加,而其他算法与节点数目关系不大.

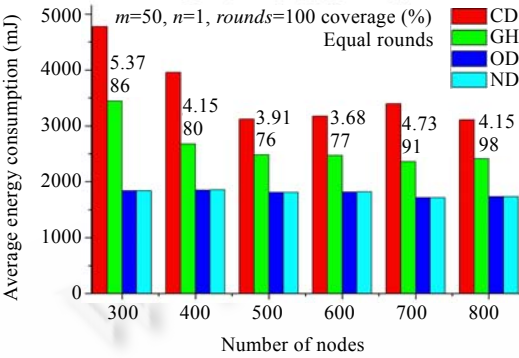


Fig.6 Number of nodes vs. energy consumption

图 6 不同数目节点的能量消耗

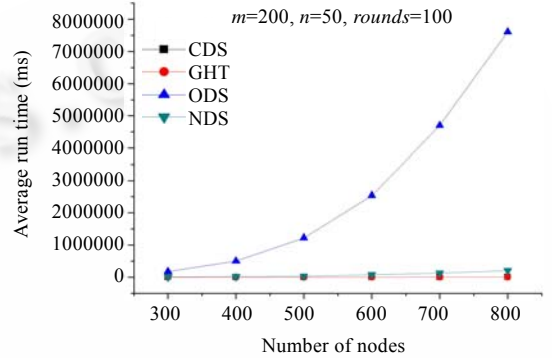


Fig.7 Number of nodes vs. run time

图 7 不同数目节点的运行时间

除了能量消耗和计算复杂性以外,生产者 and 消费者之间的延时在传感器网络数据存取中也扮演着重要角色.本文中,延时定义为消费者与生产者到存放节点的跳数之和的最大值.图 8 显示了在  $m=200, n=50$  的条件下,100 次实验中延时平均值和节点数目  $N$  的关系.图中 NDS 和 ODS 两条曲线重合说明具有相同延时,平均延时约为 4.4 跳( $1100/(200+50)\approx 4.4$ ),小于 CDS 和 GHT.

图 9 显示了在  $N=400, m=200, n=50$  的条件下不同数据压缩比例对能量消耗的影响.随着数据压缩比例从 0.1 增加到 1,网络中能量消耗逐渐增加.这是因为,查询代价与数据压缩比例成正比,而且查询访问是数据存取能量消耗重要的组成部分.此外,CDS 和 GHT 对数据压缩比例比 ODS 和 NDS 要敏感(ODS 和 NDS 两条曲线重合),说明数据融合在数据存取中扮演着非常重要的角色.

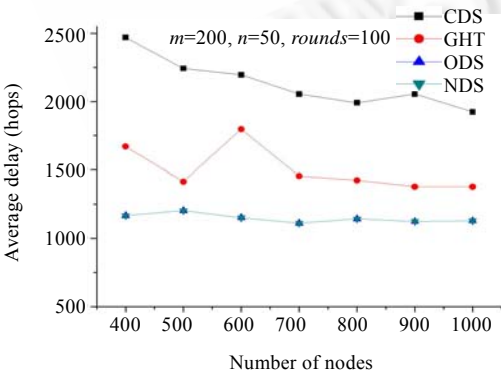


Fig.8 Number of nodes vs. delay

图 8 不同数目节点的延时

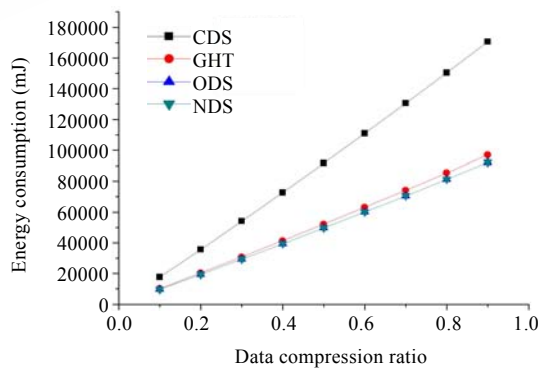


Fig.9 Data compression ratio vs. energy consumption

图 9 不同数据压缩比例的能量消耗

## 6 结束语

本文在网状拓扑结构网络中将数据速率和地理位置结合起来考虑了生产者和消费者进行数据存取问题。依据生产者和消费者的关系,建模为“一对一”、“多对一”、“多对多”3种模型。其次,基于上述模型,本文提出利用数据速率和地理位置来确定数据存放位置的自适应算法 ODS、NDS 和最优数据传输模式。相对于 ODS 和 NDS,计算复杂性大为减少。最后,ODS 和 NDS 通过自适应地调整数据存放位置来减少数据存取的能量消耗。实验结果显示,NDS 只需要 ODS 5%的计算代价,但是在 70%的情况下达到相同的效果,而且访问延时几乎相同。

本文的后续工作包括将集中式数据存取优化转化为分布式优化,即不依赖于基站协助就可以完成优化工作以及采用其他度量标准进行优化。更为重要的是,将本文的数据存取策略付诸实践,应用到实际无线传感器网络项目中。

## 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] Li GL, Gao H. A load balance data storage method based on ring for sensor networks. *Journal of Software*, 2007,18(5):1173-1185 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/1173.htm>
- [3] Shenker S, Ratnasamy S, Karp B, Govindan R, Estrin D. Data-Centric storage in sensor networks. *ACM SIGCOMM Computer Communications Review*, 2003,33(1):137-142.
- [4] Ganesan D, Greenstein B, Estrin D, Heidemann J, Govindan R. Multi-Resolution storage and search in sensor networks. *ACM Trans. on Storage*, 2005,1(3):277-315.
- [5] Madden S, Franklin M, Hellerstein J, Hong W. Tinydb: An acquisitional query processing system for sensor networks. *ACM Trans. on Database Systems*, 2005,30(1):122-173.
- [6] Intanagonwiwat C, Govindan R, Estrin D, Heidemann JS, Silva F. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. on Networking*, 2003,11(1):2-16.
- [7] Gil TM, Madden S. Scoop: An adaptive indexing scheme for stored data in sensor networks. In: Dogac A, Ozsu T, Sellis T, eds. *Proc. of the 23rd Int'l Conf. on Data Engineering*. Istanbul: IEEE Computer Society, 2007. 89-102.
- [8] Sheng B, Li Q, Mao W. Data storage placement in sensor networks. In: Conti M, Sivakumar R, eds. *Proc. of the 7th ACM Int'l Symp. on Mobile Ad Hoc Networking and Computing*. Los Angeles: ACM Press, 2006. 344-355.
- [9] Ratnasamy S, Karp B, Yin L, Yu F, Estrin D, Govindan R, Shenker S. Ght: A geographic hash table for data-centric storage. In: Raghavendrv CS, ed. *Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications*. New York: ACM Press, 2002. 78-87.
- [10] Li X, Kim Y, Govindan R, Hong W. Multi-Dimensional range queries in sensor networks. In: Ian A, Deborah E, eds. *Proc. of the 1st ACM Conf. on Embedded Networked Sensor Systems*. New York: ACM Press, 2003. 63-75.
- [11] Greenstein B, Estrin D, Govindan R, Ratnasamy S, Shenker 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*. Washington: IEEE Computer Society, 2003. 163-173.
- [12] Ganesan D, Greenstein B, Perelyubskiy D, Estrin D, Heidemann J. An evaluation of multi-resolution storage for sensor networks. In: Ian A, Deborah E, eds. *Proc. of the 1st ACM Conf. on Embedded Networked Sensor Systems*. New York: ACM Press, 2003. 89-102.
- [13] Braginsky D, Estrin D. Rumor routing algorithm for sensor networks. In: Raghavendra CS, ed. *Proc. of the 1st Workshop on Sensor Networks and Applications*. Atlanta: ACM Press, 2002. 22-31.
- [14] Sarkar R, Zhu X, Gao J. Double rulings for information brokerage in sensor networks. In: Petrioli C, Ramjee R, eds. *Proc. of the 12th Int'l Annual Conf. on Mobile Computing and Networking*. Los Angeles: ACM Press, 2006. 286-297.
- [15] Kapadia S, Krishnamachari B. Comparative analysis of push pull query strategies for wireless sensor networks. In: Gibbons P, ed. *Proc. of the Int'l Conf. Distributed Computing in Sensor Systems*. Berlin: Springer-Verlag, 2006. 185-201.
- [16] Wang C, Xiao L. Locating sensors in concave areas. In: Azcorra A, Touch J, Zhang ZL, eds. *Proc. of the IEEE INFOCOM*. New York: IEEE Communications Society, 2006. 1-12.
- [17] Har-Peled S, Mazumdar S. Coresets for  $k$ -means and  $k$ -median clustering and their applications. In: Babai L, ed. *Proc. of the 36th Annual ACM Symp. on Theory of Computing*. New York: ACM Press, 2004. 291-300.

[18] Karp B, Kung H. Greedy perimeter stateless routing for wireless networks. In: Pickholtz R, ed. Proc. of the 6th Annual ACM/IEEE Int'l Conf. on Mobile Computing and Networking. Boston: ACM Press, 2000. 243-254.

[19] Silberstein A, Braynard R, Yang J. Constraint-Chaining: On energy-efficient continuous monitoring in sensor networks. In: Chaudhuri S, ed. Proc. of the 25th ACM Int'l Conf. on Management of Data. New York: ACM Press, 2006. 157-168.

附中文参考文献:

[1] 李建中,李金宝,石鹏飞.传感器网络及其数据管理的概念、问题与进展.软件学报,2003,14(10):1717-1727. <http://www.jos.org.cn/1000-9825/14/1717.htm>

[2] 李贵林,高宏.传感器网络中基于环的负载平衡数据存储方法.软件学报,2007,18(5):1173-1185. <http://www.jos.org.cn/1000-9825/18/1173.htm>



蔚赵春(1981-),男,内蒙古四子王旗人,博士生,主要研究领域为高级数据管理,无线传感器网络.



肖斌(1974-),男,博士,助理教授,主要研究领域为网络安全,P2P 计算,网格和普适计算,传感器网络.



周水庚(1966-),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,数据挖掘,信息检索,生物信息学.

第 5 届智能 CAD 与数字娱乐学术会议

征文通知

由中国图象图形学会计算机动画与数字娱乐专业委员会、中国人工智能学会智能 CAD 与数字艺术专业委员会、以及中国工程图学学会国际联络工作委员会联合主办,大连大学承办的第 5 届智能 CAD 与数字娱乐学术会议,将于 2008 年 7 月 22 日在美丽的海滨城市大连举行。

一、征文内容(主要包括,但不限于)

- 智能 CAD
- 数字艺术
- 计算机动画
- 虚拟现实
- 网络游戏
- 可视化技术
- 模式识别
- 人机交互
- 计算机图形学
- 图像处理
- 信息融合
- 多媒体技术
- 计算机视觉
- 人工智能
- 数字内容管理
- 交互式玩具
- E-Home
- 运动捕获动画
- 数字博物馆
- 人脸表情跟踪与识别

二、论文格式及注意事项

论文相关要求请登陆会议网站: <http://202.199.159.247/cide2008/>

电子投稿,请将 WORD 格式的文件发到: [xpwei@dlu.edu.cn](mailto:xpwei@dlu.edu.cn),投稿时务必在电子邮件正文中留下通讯作者的详细通讯地址、邮政编码、电话,以便联系。

三、重要日期

截稿日期: 2008 年 3 月 31 日                      录用日期: 2008 年 4 月 30 日

修改稿接收及注册截止日期: 2008 年 5 月 31 日

四、联系方式

联系人及联系电话: 张强(0411-87403733)

电子信箱: [zhangq30@gmail.com](mailto:zhangq30@gmail.com)