

doi:10.11835/j.issn.1000-582X.2017.09.007

带非均匀节点分布无线传感器网络的高效节能数据存储

周双勇¹, 高家志¹, 邹书涵¹, 谢世恒¹, 郭松涛²

(1. 国网重庆市电力公司检修分公司, 重庆 400039; 2. 西南大学 电子信息工程学院, 重庆 400715)

摘要: 现有大部分无线传感器网络的分布式数据存储方法都依赖于传感器节点定位系统, 这会导致节点消耗大量的能量, 而且这些方法主要是针对均匀分布的无线传感器网络的, 并不适用于非均匀节点分布的传感网络。为了解决这个问题, 提出一种非均匀节点分布传感网络的大数据路由存储算法, 其目的在于减少传感器节点的实际分布和地址。为了进一步节省数据存储空间和能量消耗, 将布隆过滤器(Bloom filter)集成到节点上, 从而进一步减少数据丢失和网络能量消耗。文中算法提供了高效的搜索服务, 使数据在网络内的存储分布和路由能量消耗更加均匀, 进而提高了网络的生存时间。文中算法在容错情况下通过减少冗余数据来提供高效节能的存储, 并进一步减少数据的路由开销和存储空间的浪费。

关键词: 无线传感器网络; 数据存储; 负载均衡; 能量高效

中图分类号: TP393

文献标志码: A

文章编号: 1000-582X(2017)09-057-10

Energy-efficient data storage for wireless sensor networks with nonuniform distribution

ZHOU Shuangyong¹, GAO Jiazhi¹, ZOU Shuhan¹, XIE Shiheng¹, GUO Songtao²

(1. Maintenance branch, State Grid Chongqing electric power company, Chongqing 400039, P.R.China;

2. College of Electronic and Information Engineering, Southwest University, Chongqing 400715, P.R.China)

Abstract: Most of the existing distributed data storage methods in wireless sensor networks (WSNs) rely on locating system of sensor nodes, which will lead to much energy consumption, and in particular, these methods are for the WSNs with uniform sensor distribution, and are clearly not applicable for the WSNs with nonuniform sensor distribution. To address this problem, this paper proposes a data storing algorithm based on the non-uniform node distribution, which aims at estimating the real distribution and the addresses of sensor nodes. To further save storage space and energy consumption, we integrate the Bloom filter with sensor nodes. The proposed algorithm offers an efficient search service so as to make data storage distribution and routing energy consumption more even, which can prolong network lifetime. The proposed algorithm also reduces redundant data with toleration of a given error bound to provide energy-efficient storage and further decreases the overhead of data routing and the waste of storage space.

Keywords: wireless sensor networks; data storage; load balance; energy efficiency

收稿日期: 2017-03-22

基金项目: 国家自然科学基金(61373179)。

Supported by National Natural Science Foundation of China (61373179).

作者简介: 周双勇(1982—), 男, 主要从事无线传感器网络研究, (E-mail) 165523921@qq.com。

郭松涛(联系人), 男, 教授, 博士, 主要从事无线与移动网络、无线传感器网络研究, (E-mail) stguo@swu.edu.cn。

无线传感网络^[1] (WSNs, wireless sensor networks)是由大量廉价的传感器节点通过无线通信方式组成的多跳无线自组织网络。节点之间以协作方式感知、采集、处理和传输网络覆盖地理区域内被感知对象的信息,然后传输汇聚节点。目前,无线传感网络已经在监测输电线路检测^[2]、智能家居^[3]、城市交通、建筑物结构监控^[4]等方面得到了广泛应用。在这些应用中,由于传感器数量较多,会产生大量数据^[4],所以数据的存储与路由成为人们必须解决的问题。特别是人类的大量智能决策日益依赖于这些远程的感知数据,所以以较少的资源代价存储大量的多种类型的感知数据是很关键的。

为了使传感器网络能够提供高效便捷的数据查询服务,需要根据具体的应用场景设计不同的数据存储算法。无线传感器网络的数据存储主要解决感知数据在传感器网络中的存储问题,即确定感知数据在网络中的哪个传感器上存储。具体来说,当一个节点采集到数据后,存储算法需要为这些数据确定存储在哪个节点;当用户要查询这些数据时,就根据存储算法确定的存储位置以多跳路由方式检索满足查询要求的数据。所以,存储算法的设计对数据查询也起着重要作用。设计合适的存储算法不仅能降低算法所需要的系统资源消耗,而且能够减少数据存储和查询的路由开销。

目前,为了实现有效的数据存储,研究人员已经提出了一些存储模式。这些模式可以分为 3 类:外部存储、本地存储和数据中心存储。外部存储是感知数据被发送到感知现场外面的汇聚节点进行存储。本地存储是指传感器节点将感知数据存储在本地的,这种方式的优点是减少了从传感器节点到汇聚节点之间的路由代价。数据中心存储是指事件名称被 Hash 到一个地理位置。具有相同事件名称的所有数据将被存储在邻近该地理位置的节点上。然而,这些现有的存储算法存在以下不足^[14-16]:1)分布式存储算法中节点分布的不均匀会导致负载不平衡。这是由于网络中的每个区域存储的数据量与区域内的节点数据不匹配。2)这些现有的存储算法没有处理冗余数据,这会造成节点存储空间利用率降低,因为数据的冗余性会导致相同的数据被重复路由和存储。

针对现有存储方法存在的不足,文中通过分簇来估计节点的分布,采用轮询方式更换簇头从而实现簇成员的能量平衡,进而提出节点非均匀分布的数据存储路由算法。该算法首先使用类似于投影 Hash 处理数据,然后根据分布估计将处理的数据映射到簇。这种处理是为了获取某一时刻哪些传感器曾经产生过相同数据,平衡传感器节点之间的存储负载。因此,文章提出了基于簇的路由算法,该算法可以有效地将数据路由到 Hash 位置。通过在路由算法中使用排斥法来避免重复的路由路径,从而平衡节点的路由能量消耗。实验表明,文中的存储路由可以有效平衡节点存储负载和降低网络能量消耗,从而提高网络服务质量和延长网络的生存时间。

1 算法设计

无线传感器网络中节点非均匀分布的大数据存储算法主要过程如下:

1)利用布隆过滤器(BF, Bloom filter)对数据进行预处理,在簇头上建立数据节点的索引表,从而减少不必要的数据存储量;

2)利用分簇算法估计真实的节点分布状况,进而基于所估计的节点分布对节点进行编址,并利用 Hash 函数确定数据的存储位置;

3)利用基于簇的路由算法实现数据存储。

需要注意的是,该算法使用投影 Hash 函数和路由算法来实现存储路由时的节点负载平衡。Hash 函数使存储在簇内的数据量与节点数量成正比,通过路由协议使转发簇内数据的能量消耗与簇内节点数目成反比。此外,算法还通过配置布隆过滤器来进一步提高能量的有效性。

1.1 布隆过滤器

布隆过滤器 BF^[6]是一种紧数据结构,在该结构中, n 个元素的集合 $S = \{s_1, s_2, \dots, s_n\}$ 可以通过 m 位的位数组来表示,且可以实现近似成员查询。首先,将 m 位的位数组初始化为 0。然后根据 K 个独立的 Hash 函数 h_1, h_2, \dots, h_k 向 BF 中添加元素。对于元素 s_i ,它的所有位 $h_j(s_i)$ 被设置为 1, $j = 1, 2, \dots, k$ 。当查询成

员 x 时,检查所有的比特位 $h_j(x), j=1,2,\dots,k$ 。如果这些比特位都为 1,则认为 x 以某个概率属于集合 S 。反之,则 x 不属于集合 S 。

通过最大化匹配概率,即减少错误识别的概率,可以优化布隆过滤器 BF。如果过滤器中相关联的比特位被集合中其他元素所占据而产生冲突,就会导致错误识别。假设 K 个 Hash 函数完全随机独立,并且令 BF 的错误识别率为 $p_{k,n,m}$,则识别率计算公式^[7]为

$$p^k = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k, \quad (1)$$

$$p^k < p_{k,n,m} \leq p^k \times \left(1 + O\left(\frac{k}{p} \sqrt{\frac{\ln m - k \ln p}{m}}\right)\right)。$$

除此之外,在给定位数组大小 m 和 S 中元素个数 n 的情况下,可以得到误识别率最小的 Hash 函数的个数 k ,即当 $k = \ln 2 \times \frac{m}{n}$ 时,误识别率 $p_{k,n,m}$ 最小。

1.2 带 BF 的传感器节点

无线传感器网络中节点分布不均匀的数据冗余可以分为时间数据冗余和空间数据冗余。时间数据冗余是指同一个节点所产生的数据随着时间的变化是相同的;空间数据冗余是指在空间上位置不同的节点在同一时间或不同时间所产生的数据是相同的。空间数据冗余要求不同节点感知的范围要几乎完全重叠。而时间数据冗余是对于同一个节点需要检测到周期的事件数据。空间数据冗余一般不易出现,而时间数据冗余经常发生。因此,文中主要考虑时间数据冗余。文中采用布隆过滤器 BF 来对实时性要求不高、能容忍低错误率的无线传感器网络的数据冗余进行判重处理。为了能有效地处理时间数据冗余,假设每个传感器节点都配备了一个布隆过滤器 BF,如图 1 所示。

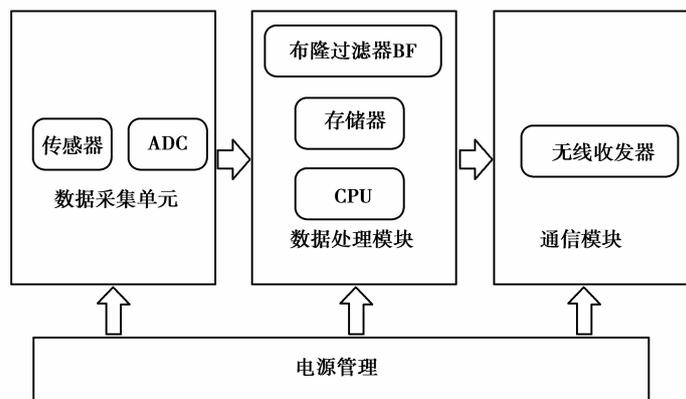


图 1 带布隆过滤器 BF 的传感器节点

Fig.1 Sensor node with Bloom filter

传感器节点主要由数据收集模块、数据处理模块、无线传输模块和电源管理模块组成。数据收集模块负责对监测目标的信息进行采集和转换;数据处理模块主要负责控制整个传感器节点的计算、存储和处理自身收集的数据以及其他传感器节点转发来的数据;无线传输模块主要负责传感器节点之间以及传感器节点与汇聚节点之间的通信,交互控制消息和收/发感知数据。

布隆过滤器 BF 主要用于消除传感器自身产生的重复冗余数据。通过对传感器所产生的所有事件数据进行冗余处理,如果该数据不存在,则填充位向量,反之,就丢弃该事件数据。这里位向量即为在 1.1 节中所描述的表示 n 个元素集合的 m 位数组。为了使簇头节点在路由的时候能够知道该簇区域内是否有满足要求的数据,每个传感器节点需要周期性地将位向量传递给簇头节点进行位与操作。位与操作即为当查询成员 x 时,将所有的比特位 $h_j(x), j=1,2,\dots,k$ 进行与操作,如果这些比特位都为 1,那么位与操作的结果即为 1,则认为 x 以某个概率属于集合 S 。反之,则认为 x 不属于集合 S 。显然,由于这种操作是由单个节点独

立实现的,所以不会带来额外的信息传输开销。这样做的目的,就是减少数据传输开销,延长网络生存周期,尽可能地节约存储空间。

1.3 节点分布估计

算法利用布隆过滤器 BF 对传感器节点所产生的冗余数据进行滤除,此外,还需要估计出传感器网络中节点的分布情况。由于传感器网络中可能出现节点分布不均匀的问题,这会导致传感器节点中负载的不平衡。因此,文中采用分簇算法,例如 LEACH 算法,对网络内节点的分布情况进行估计。根据 LEACH 协议,可以生成一个具有 n 个簇的传感器网络。此外,对每个簇都将分配一个标号作为该簇的标识和地址,目的是为了便于在传感器网络中容易找到每个簇。一个簇内的传感器节点数目可以由簇头节点通过信标来得到,簇头节点计算该簇被选作存储其他簇数据的概率,最后以广播的方式使每个节点都得知此概率并得出分布概率向量 P ,使其作为 Hash 函数的输入。有关 Hash 函数的介绍将在下一小节进行描述。例如,对于簇 $i \in [1, n]$,假设该簇内有 m_i 个簇成员,则这个簇被选中的概率为 $p_i = m_i/N$,分布向量则为 $P = (p_1, p_2, \dots, p_n)$ 。因此,通过 Hash 函数感知的数据将被存储到每一个节点上,以使整个网络的负载达到平衡。

为了降低在实现负载均衡时所带来判决、分发、处理、传输等能耗和开销,首先利用布隆过滤器 BF 对数据进行预处理,从而在簇头上建立数据节点的索引表,减少不必要的数据存储量,然后对节点进行编址,利用 Hash 函数来确定数据的存储位置,最后利用基于簇的路由算法实现数据存储。

1.4 节点编址

在完成节点分布估计的基础上,每个簇头均可得知簇中成员节点的数量,并依据成员节点接收来自簇头的信号强度对簇中的成员节点连续地分配标号,目的是为了能够将数据均匀地保存在簇内节点上。节点编址的过程如图 2 所示。为了能更好地将负载均匀地分布在网络中,可以通过顺序循环簇内节点标号的方式来旋转簇头。例如,对于一个簇 i ,若它的成员节点数为 m_i ,假设其簇头节点的标识为 0,则簇中每一个成员节点均可以根据来自簇头信号强度得到一个标号 $s \in [1, m_i - 1]$ 。完成初始化之后,簇中节点的标号将不再做任何变化,而簇头节点则由簇内成员节点依次轮流担当。此外,簇头除了具有转发数据和平衡负载的功能之外,还具有数据存储的职责。因此,文中提出的节点编址方法不仅有助于数据路由而且能够减少能量开销。

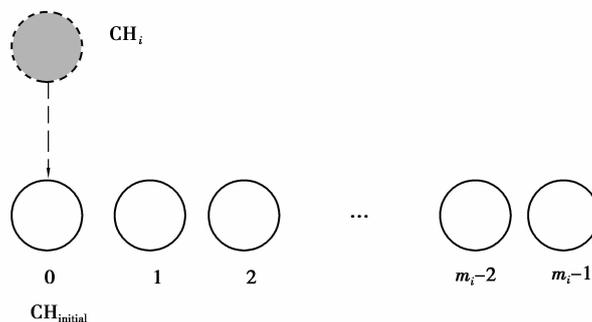


图 2 传感器节点编址示意图

Fig.2 Illustration of sensor node addressing

2 数据的存储过程

2.1 确定存储位置

对节点进行编址后,算法需要利用 Hash 函数来确定数据的存放位置。文献[8-9]得到确定数据的存储位置是通过标准的 Hash 函数来实现的。然而,无论该区域中传感器节点的数目有多少个,标准 Hash 函数都可以将数据均匀地映射到每个区域中,从而导致相同大小具有较多节点的区域被分配较少的数据。文中所提出的 Hash 函数具体描述如算法 1 所示。

算法 1 文中所提出的 Hash 算法
Algorithm 1 The proposed Hash algorithm

输入:数据 k , 分布向量 $P=(p_1, p_2, \dots, p_n)$
输出:簇的标识 $ID, ID \in [1, n]$

```

1:  $i = 0$ ;
2: While (true) do
3:    $(ID, z) = \text{Hash}(k + i)$ ;
4:    $i++$ ;
5:   if  $z < P_{ID}$  then
6:     return  $ID$ ;
7:   end if
8: end while
    
```

表面看来,该算法是一个伪随机数字产生器,它将节点分布向量 P 和数据 k 作为算法输入,并将簇的标号 ID 作为算法输出。实际上,它是采用一种与排斥法相似的一种策略,通过在有限区域中生成服从均匀概率分布的随机数。二元组 (ID, z) 被均匀随机产生,其中 $ID \in [1, n]$ 是簇的标号, $z \in (0, 1)$ 。如果 $z < P_{ID}$, 那么 ID 被返回,否则,二元组 (ID, z) 被重复产生直到 $z < P_{ID}$, 其中 P_{ID} 表示 ID 被返回的概率。

除此之外,该算法也可以对存储在同一个簇内的数据量进行分析。例如簇头 CH_i , 数据量为

$$n_i = \sum_{j=1}^n n_{i,j} = \sum_{j=1}^n m_j \cdot p_i = p_i \sum_{j=1}^n m_j = p_i \cdot N = m_i, \tag{2}$$

其中 $n_{i,j} = m_j \cdot p_i$, 表示由簇 j 所产生的数据被存储在簇 i 内的数据量。式(2)所表达的是簇 i 中的每个节点只负责对一个节点产生的数据进行存储,这种方式则保证了存储负载均衡。

2.2 建立数据索引表

数据存储的位置确定后,算法将在簇头上建立数据节点的索引表,从而降低在实现负载均衡时所带来的判决、分发、处理、传输等能耗和开销。传感器节点采集到数据后,首先使用布隆过滤器 BF 对采集到的数据进行过滤处理,如果布隆过滤器中有 K 个哈希值所对应的位都是 1, 则将该数据舍弃掉,并且只保存其对应的位数组,为了方便对以后的数据进行恢复;若布隆过滤器中 K 个哈希值对应的位均不都是 1, 那么将 BF 的位数组中对应位从 0 置为 1, 并存储该数据到本地。成员节点不仅将自己收集的数据进行过滤,而且会周期性地将其 BF 的位数组传输给簇头,簇头再将从成员节点获取的位数组与位数组表中对应节点的位数组进行或操作,可以通过该方式将数据节点的索引表建立在簇头节点上。此外,在位数组表中可以对每个簇头节点的位数组进行或操作,目的是便于在簇之间建立一个数据索引。例如,假设对于簇 i , 其有 l_i 个簇内成员, 则簇头创建一个 $l_i \times m$ 的位数组表, 用 BF_i 表示。如图 3 所示, 其中簇头的布隆过滤器位数组是用 0 来表示的, 它的值是由 BF 位数组的成员节点 $BF_{i,j}$ 通过与操作来获得。此外,簇头节点除了需要创建和维护索引表的职责之外,还需要承担数据转发或路由的任务。簇之间的位数组表可用图 4 表示。

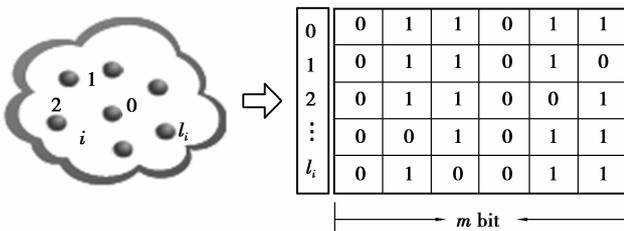


图 3 簇内 BF 的位数组表

Fig.3 Bit array in intra-cluster Bloom filter

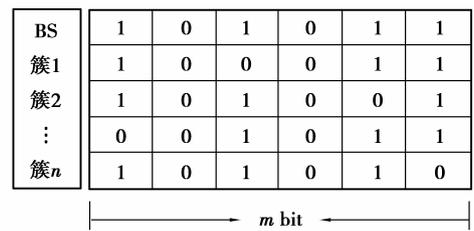


图 4 簇间 BF 的位数组表

Fig.4 Bit array in inter-cluster Bloom filter

2.3 路由实现存储数据

文中假设传感器系统中的节点不具备定位模块,而且并不知道每个传感器节点的地理坐标,设计了一种基于簇的路由算法。该算法类似于文献[10-11]中所提出的算法。因为簇在无线传感器网络中的分布具有不规则的特点,很难计算得到一个簇的具体大小。从 2.2 节得知,簇节点的位置由簇的标识和簇内成员节点的相对地址两部分组成,而且系统中的所有簇头都会维持一个邻居簇的路由表,如表 1 所示。其中,表 1 中的两列分别表示簇的邻居簇和邻居簇被选中作为继簇的概率,并且这个概率为

$$p_{i,j} = 1 - \frac{p_j}{\sum_{j \in V_{i,N}} p_j} \quad (3)$$

表 1 簇 i 的邻居簇路由表

Table 1 Routing table of neighbor clusters of cluster i

邻居簇	概率 $P_{i,j}$
簇 $i - 1$	0.56
簇 $i + 1$	0.32
簇 $i + 2$	0.28
簇 $i + 3$	0.36

所以,如果有一个节点需要传输数据时,簇节点首先将数据发送到簇头,簇头进而在簇中进行匹配操作。如果匹配成功,则簇头会把数据存储到簇内;否则,簇头会发送目的标号和它的邻居簇进行匹配操作。如果匹配成功,则该簇头会直接发送数据到匹配的邻居簇中进行存储;否则,该簇头会从其邻居簇中选择一个簇作为下一跳的中继,而该中继的选择是依据路由概率。该过程会一直运行直到找到目的标号。路由算法具体的描述如算法 2 所示。

算法 2 路由算法

Algorithm 2. Routing algorithm

```

1:  $V_{i,N}$ :簇  $i$  的邻居簇集合,  $CH_j \in V_{i,N}$ ;
2:  $P_{i,N}$ :簇  $V_{i,N}$  的概率集合  $P_j \in P$ ;
3: 通过公式(3)计算  $P_{i,j}$ ,  $p_{i,j} \in P_{i,N}$ ;
4: if  $ID_{CH} = ID_{hash}$  then
5:     在本簇内存储数据包;
6: end if
7: if  $ID_{hash} = CH_j, (CH_j \in V_{j,N})$  then
8:     簇  $i$  向簇  $j$  发送数据包去存储;
9:     返回;
10: else
11:     while  $ID_{hash} \neq CH_j, (CH_j \in V_{j,N})$  do
12:         while (true) do
13:             产生随机数  $z, j \leftarrow 1$ ;
14:             if  $z < P_{i,j}$  then
15:                 簇  $i$  向簇  $j$  发送数据包;
16:                 簇  $j$  作为下一跳并转发簇  $i$  的数据包;
17:                 计算  $V_{i,N}$ , 更新  $P_{i,N}$ ;
18:                 退出;
19:             end if
20:              $j \leftarrow \text{mod}(j, \text{Size of}(V_{i,N})) + 1$ 
21:         end while
22:     end while
23:     簇  $i$  向簇  $j$  发送数据包;
24: end if

```

文中所提出的方法分别可以通过数据存储、簇内路由和簇间路由来平衡能量消耗。其中,在数据存储方面,采用分布概率把数据分散到各个节点中,是为了让每一个节点具有相同的存储负载。此外,在簇间路由方面,为了平衡簇间的能量消耗,选择簇成员少的簇作为下一跳的目的节点。最后,在簇内路由方面,为了避免因为簇内节点过载而引起的数据丢失问题,采用轮换簇头的方法来平衡簇头和成员节点的能量消耗。

3 仿真分析

3.1 仿真设置

在该小节中,将评估非均匀节点分布存储和搜索算法。

假设 400 个节点按照均值为 0 和标准差为 20 的高斯随机分布在 $100\text{ m} \times 100\text{ m}$ 的矩形区域内,如图 5 (a)所示。由于在高斯随机分布中采用了较大的标准差,所以这种节点分布基本上可以刻画节点分布的非均匀性。节点被分成 100 个簇,每个簇内平均有 4 个传感器节点。设置每个节点的通信范围是 16 m,节点的初始能量为 10 J。仿真时间为 1 000 s。节点转发每单位数据所消耗的能量为 10^{-6} J 。其中,每个数据项中有 20 个字节并且包括簇和节点的标号信息。首先,以 LEACH 协议为标准对网络进行分簇,然后分析使用布隆过滤器 BL 前后的存储空间的占用率,这指的是采用布隆过滤器之后的存储数据量与未使用布隆过滤器的存储数据量之间的比值。对使用布隆过滤器前后哈希函数如定向扩散算法 DD^[12]、地理哈希表 GHT^[8] 和文中所提出的算法在数据映射误差、网络寿命方面进行比较。

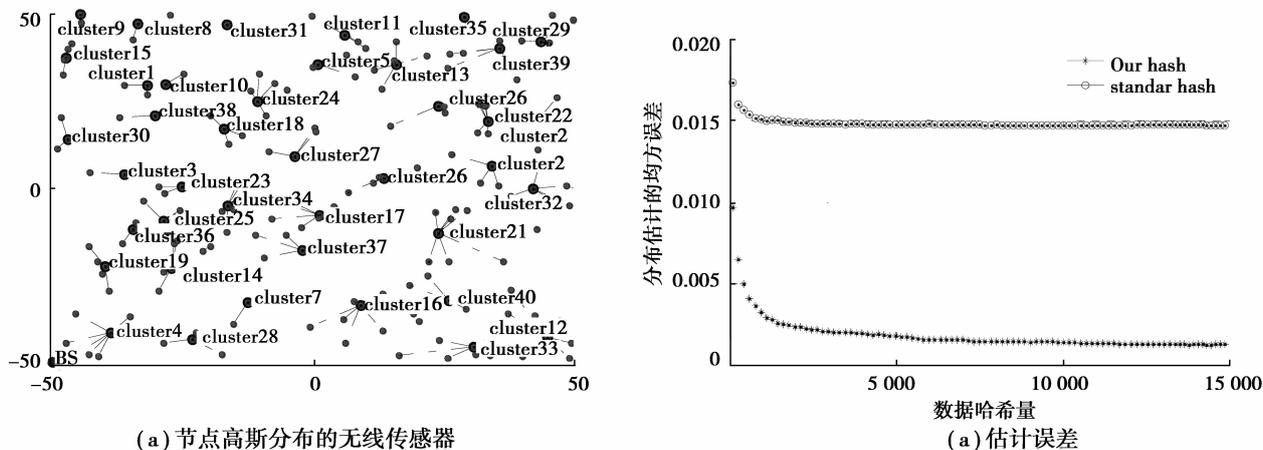


图 5 高斯分布估计误差

Fig.5 Estimation error of Gauss distribution

最后,为了评价文中提出的存储方法,以存储空间利用率的性能为指标进行衡量。在实验中,能量消耗的计算采用张法等^[17]提出的网络系统能量消耗模型,该模型从路由和调度的角度研究了能量消耗优化方法,能很好地适用文中提出的算法。参数列表如表 2 所示。

表 2 参数列表

Table 2 List of parameters

符号	定义	值	符号	定义	值
N	节点个数	400	Q	查询个数	100
n	簇的个数	100	S	节点内存 /M	20
k	维数	3	c	转发单位数据的花费/J	10^{-6}

3.2 数据映射到簇的误差

图 5(b)描述了节点服从高斯分布的传感器网络中数据映射误差。均方误差(MSE, mean squared error)是用来量化标准哈希函数和提出的哈希算法的方法。在图 5(b)中,虽然误差值随着哈希数据量的增长而表现出逐渐减少,但是文中所提出的哈希函数的误差却总位于标准哈希函数之下。出现这种情况是因为标准哈希函数会将数据均匀地映射到每个簇中,然而这可能会丢失数据或造成额外的开销;但是在文中提出的哈希函数中,它会依据一个确定的概率使数据映射到每一个簇中。当哈希数据数量不断增长,误差会逐渐减少,直到最后会逼近于一个极小的值。因此,无论有多少的哈希数据量,文中方法总能够很好地将数据进行分散。

3.3 应用 BF 之前网络寿命

到目前为止,对于网络寿命这个概念尚且没有一个精确的定义。然而,根据文献[13]所给的定义,可以通过 3 种方式来衡量网络的寿命:1)以网络部署为起点直到网络中第一个节点死亡,该过程中所花费的时间;2)当网络中死亡的节点数目达到一定比例时所花费的时间;3)网络中的所有节点都死亡时所花费的时间。文中采用两种方式来衡量网络寿命,一个是通过网络能量的方式,另一个方式是采用网络内剩余节点的数目来反映。图 6(a)表示了定向扩散算法 DD^[12]、地理哈希表 GHT^[8]和文中提出的算法之间的对比,可以得出 3 种算法中网络总剩余能量几乎都随时间作线性变化,这是由于转发数据包的跳数和与同一时间内的路由能量消耗是近似相等的。进而,可以得到这样的结论:1)DD 由于广播所以消耗能量更快;2)剩余能量方面,文中提出的算法比 GHT 节省更多能量。这是由于在查询范围内,GHT 需要发送更多的查询请求,因此会带来更多的能量消耗。

图 6(b)表明了随时间的变化,采用 3 种算法所得到的网络存活节点数目。可以得出,网络中存活的节点数最少的是 DD 方法。那是因为对于一些特殊的节点,广播会快速地消耗节点能量;然而与 GHT 方法相比,文中方法有更多的存活节点,这是因为 GHT 算法的路由数据包在具有相同的目的节点的情况下,都选择同样的最短路径,这对某些特殊节点来讲,会造成更多的能量开销并且较早地从网络中死亡。然而,文中方法是尽可能从多条路径中来选择其中一条,目的是为了分散节点中的能量消耗,进而有效地延长网络寿命。

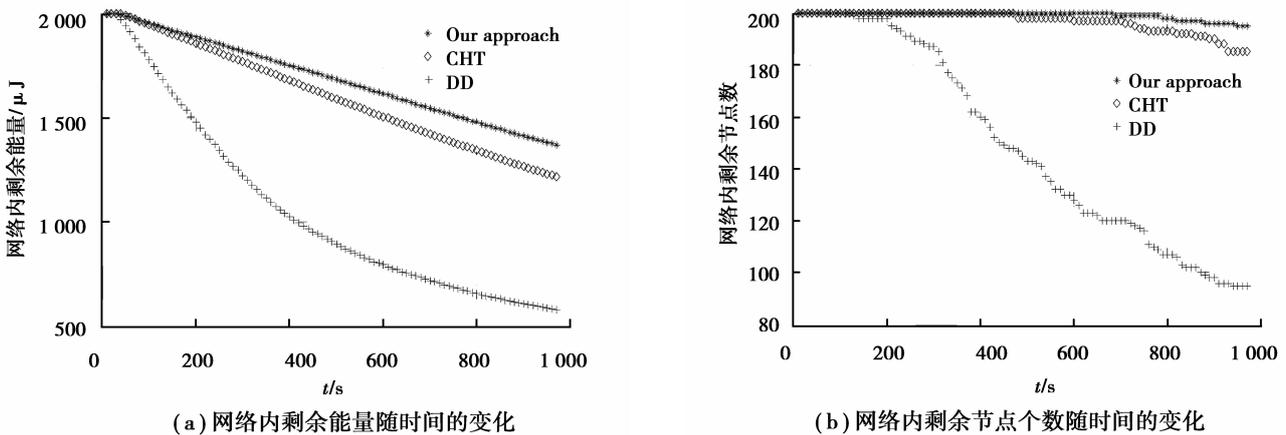


图 6 网络能量效率的比较

Fig.6 Comparison of network energy efficiency

3.4 应用 BF 之后的网络寿命比较

图 7(a)描述了在随着时间的变化,文中所提算法、DD 算法与 GHT 算法在采用布隆过滤器之后网络的剩余能量情况。可以看出,在使用布隆过滤器之后,DD 和 GHT 这 2 种算法在能量消耗方面都有所改善,并且其改善效果随着网络的运行时间的增加会逐渐变得明显。这个是由于在网络初始阶段,基本没有产生数据或者产生的冗余数据十分少,所以能量消耗也并没有明显地减少。然而,从图中可以得出,在能量节约方面,文中所提的算法仍然比使用布隆过滤器之后的 DD 和 GHT 算法更优。

图 7(b)表达了 3 种算法使用布隆过滤器前后网络剩余节点数目随着时间的变化关系,其中采用的 3 种算法分别是 DD、GHT 和文中所提出的算法。因为布隆过滤器具有消除重复冗余数据的功能,所以减少了能量消耗,其中包括网络路由和存储所消耗的能量。因此可以得出,采用布隆过滤器后的 DD 算法和 GHT 算法得到的网络剩余节点数目总是位于原 DD 和 GHT 之上。此外,在 DD、GHT 与文中所提算法应用布隆过滤器之后,可以看出文中算法仍然优于 DD 和 GHT 这 2 种算法。

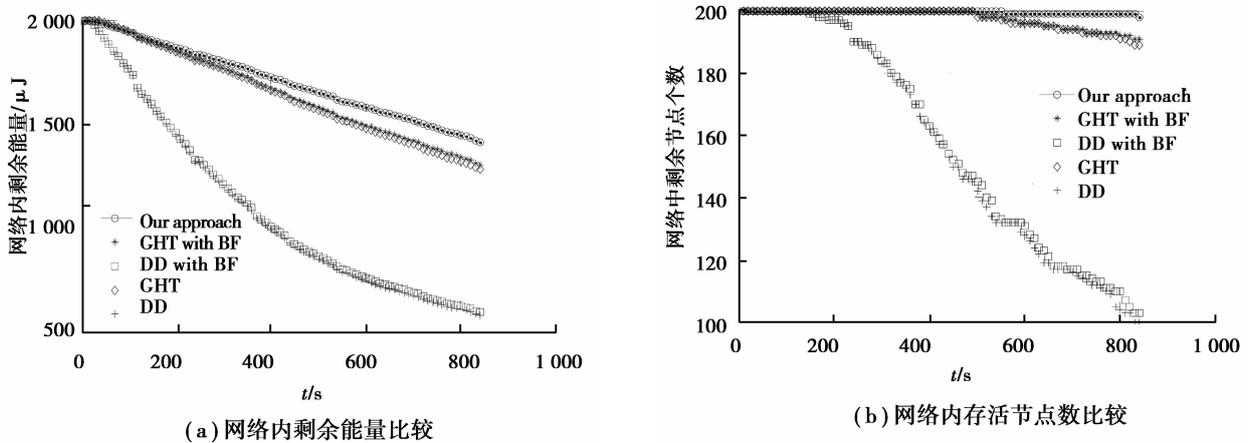


图 7 GHT、DD 和我们所提算法在使用布隆过滤器后在网络寿命方面的比较
 Fig.7 Comparison of network lifetime of GHT、DD and our algorithm with Bloom filter

3.5 存储空间利用率

为了更直接地表示布隆过滤器处理数据冗余的能力,采用存储空间利用率作为衡量布隆过滤器处理数据冗余能力的指标。可以看出,如果存储空间利用率越小,则布隆过滤器处理的冗余数据的能力越强,那么处理冗余数据也就越多。然而在一些无线传感器网络中,如果存在较大的数据冗余,可以使用布隆过滤器来消除冗余的数据,从而进一步增加网络存储数据的能力。如图 8 中所描述的,随着网络中数据量的不断增加,网络空间利用率是逐渐变小的。在无线传感器网络运行的初始阶段,因为节点还没有开始产生数据或者产生的数据比较少,所以也没有产生重复的数据,因此此时的空间利用率为 1。然而,随着网络中数据量的不断增长,重复的冗余数据量也随着逐渐变大,因而空间利用率也会逐渐变小,节省了网络中的存储空间。所以,在能够容忍一定错误的前提下,该方法能够很好地处理传感器网络中的数据冗余问题,并且进一步提高了网络的存储能力。

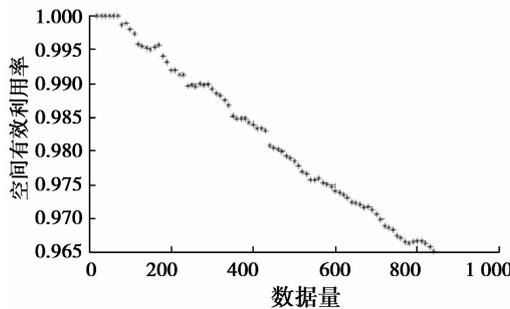


图 8 数据量与空间利用率的关系
 Fig.8 Relationship between the amount of data and space utility

4 总 结

提出了一个基于非均匀节点分布无线传感器网络的存储方案。通过传感器节点的分布估计来得到实际网络中的分布情况,便于得出数据被均匀地存储在哪些节点中。为了减少数据传输所消耗节点的能量,文中

提出的一个新的存储方法,该方法既保留了本地存储的优点又结合了数据中心存储的优点。同时,通过布隆过滤器的方法能够有效地去除重复数据,从而很大程度上节省了存储空间与路由所带来的能量消耗。最后通过仿真结果可以得到,文中所提算法在增强负载平衡、延长网络寿命、提高存储空间利用率和能量有效性等方面均优于现有的算法。

参考文献:

- [1] Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey[J]. *Computer networks*, 2008, 52(12): 2292-2330.
- [2] 吕治安. ZigBee 网络原理与应用开发[M]. 北京:北京航空航天大学出版社, 2008.
LYV Zhian. The network theory of ZigBee and it's application development [M]. Beijing: Beihang University press, 2008. (in Chinese)
- [3] Helal S, Mann W, El-Zabadani H, et al. The gator tech smart house: A programmable pervasive space[J]. *Computer*, 2005, 38(3): 50-60.
- [4] Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey[J]. *Computer networks*, 2008, 52(12): 2292-2330.
- [5] Yan Z T, Li Z L. Analysis of the transmission tower-line system in heavy ice regions subjected to cable rupture[J]. *Proceedings of the CSEE*, 2010, 30(7): 92-97. (in Chinese).
- [6] Bloom B H. Space/time trade-offs in hash coding with allowable errors[J]. *Communications of the ACM*, 1970, 13(7): 422-426.
- [7] Bose P, Guo H, Kranakis E, et al. On the false-positive rate of bloom filters[J]. *Information Processing Letters*, 2008, 108(4): 210-213.
- [8] Ratnasamy S, Karp B, Shenker S, et al. Data-centric storage in sensornets with GHT, a geographic hash table[J]. *Mobile Networks and Applications*, 2003, 8(4): 427-442.
- [9] Shen H, Zhao L, Li Z. A distributed spatial-temporal similarity data storage scheme in wireless sensor networks[J]. *IEEE Transactions on Mobile Computing*, 2011, 10(7): 982-996.
- [10] Yu J, Qi Y, Wang G, et al. A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution[J]. *AEU-International Journal of Electronics and Communications*, 2012, 66(1): 54-61.
- [11] Wu X, Chen G, Das S K. Avoiding energy holes in wireless sensor networks with nonuniform node distribution[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2008, 19(5): 710-720.
- [12] Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks[C]// *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. [S.l.]: ACM, 2000: 56-67.
- [13] Gamwarige S, Kulasekera E. An algorithm for energy driven cluster head rotation in a distributed wireless sensor network [C/OL]// *Proceedings of the International Conference on Information and Automation*, December 15-18, 2005, Colombo, Sri Lanka. 2005[2016-10-10]. <https://www.researchgate.net/publication/228974663>.
- [14] Maia G, Guidoni D L, Viana A C, et al. A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks[J]. *Ad Hoc Networks*, 2013, 11(5): 1588-1602.
- [15] Shen H, Zhao L, Li Z. A distributed spatial-temporal similarity data storage scheme in wireless sensor networks[J]. *IEEE Transactions on Mobile Computing*, 2011, 10(7): 982-996.
- [16] Chen T, Guo D, He Y, et al. A bloom filters based dissemination protocol in wireless sensor networks[J]. *Ad Hoc Networks*, 2013, 11(4): 1359-1371.
- [17] 张法, Anta A F, 王林, 等. 网络能耗系统模型及能效算法[J]. *计算机学报*, 2012, 35(3): 603-615.
ZHANG Fa, Anta A F, WANG Lin, et al. Network energy consumption models and energy efficient algorithms[J]. *Chinese Journal of Computers*, 2012, 35(3): 603-615. (in Chinese)