

doi:10.3969/j.issn.1001-2400.2014.04.029

无人值守传感器网络的高性能分布式存储算法

肖宜龙^{1,3}, 蒋海波^{2,3}

(1. 中煤平朔集团有限公司, 山西 朔州 036006; 2. 中国科学院 成都生物研究所, 四川 成都 610041;
3. 中国科学院 成都计算机应用研究所, 四川 成都 610041)

摘要: 为解决无人值守传感器网络的数据存储可靠性问题, 提出了一种具有低通信成本和低访问成本的分布式存储算法. 算法采用步数为 cn 的并行定向随机游走机制, 将网络中的 k 个源数据包按照一定的接收概率分散存储到网络中所有的 n 个节点, 在每个节点形成一个存储数据包. 理论分析和实验结果表明, 基于该算法的存储过程完成之后, 即使有部分传感器节点损坏, Sink 节点只要随机收集到 $k + \epsilon$, ϵ 大于等于 11 个存储数据包, 就能成功地计算出原来的 k 个源数据包. 与具有代表性的基于 LT 码的算法相比, 文中算法将存储每个源数据包的通信次数从约 $3n \ln n$ 降至约 $3n$; 将读取源数据包的节点访问次数从大于 $k + 100$ 降至约 $k + 11$.

关键词: 传感器网络; 数据存储; 分布式算法; 随机游走

中图分类号: TP211.9; TP393 **文献标识码:** A **文章编号:** 1001-2400(2014)04-0173-06

Distributed storage algorithm with high performance for the UWSN

XIAO Yilong^{1,3}, JIANG Haibo^{2,3}

(1. China Coal Pingshuo Group Ltd., Shuozhou 036006, China; 2. Chengdu Institute of Biology, Chinese Academy of Sciences, Chengdu 610041, China; 3. Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China)

Abstract: To solve the data storage reliability problem of the unattended wireless sensor network consisting of n sensor nodes, this paper proposes a new distributed storage algorithm based on parallel directional random walk with length cn . The proposed algorithm disseminates each of k source data packets in the network according to a given probability and forms a stored data packet in every node. Simulation results show that, after the storage process based on the proposed algorithm is completed, even if some stored data packets are missing, the data collector node can successfully recover the k source data packets from any survival $k + \epsilon$, $\epsilon \geq 11$ stored data packets. Compared with the LT codes based algorithm, the proposed algorithm reduces Sink node's query cost from $k + 100$ to $k + 11$ and reduces the network's communication cost from $3n \ln n$ to $3n$.

Key Words: sensor network; data storage; distributed algorithm; random walk

无人值守传感器网络^[1-2]通常部署在恶劣环境中, 数据收集节点 Sink 不固定在网络中, 而是每隔一段时间后在网络中出现, 收集各数据节点产生的数据包. 在 Sink 节点出现之前, 为避免数据包因节点损坏或是能量耗尽而丢失, 各数据包通常分布式地存储在整个网络中.

笔者针对无人值守传感器网络的分布式存储算法^[3-5]进行研究. 基本假设^[6]为网络由均匀分布的 n 个传感器节点构成, 其中 k 个为数据节点 ($n > k$), 每个数据节点用来感知物理量并产生一个大小固定的数据包, 称为源数据包, 其余的所有非数据节点只用来存储数据. 每个节点有相同的通信半径, 距离小于通信半径

收稿日期: 2013-04-29

网络出版时间: 2013-12-12

基金项目: 国家高技术研究发展计划资助项目(2008AA01Z402); 国家自然科学基金资助项目(61003034)

作者简介: 肖宜龙(1983-), 男, 博士, E-mail: yilongxiao@foxmail.com.

网络出版地址: <http://www.cnki.net/kcms/doi/10.3969/j.issn.1001-2400.2014.04.029.html>

的两个节点互称为邻居节点,可以直接通信;处于通信半径之外的节点通过多跳机制间接通信.网络是连通的,但网络中每一个节点都没有路由表记录如何与其通信半径之外的节点间接通信,因此,两个节点之间的间接通信是通过随机选择中间节点来完成的.

针对无人值守传感器网络设计的最新且最具代表性的分布式存储算法是文献[6]中提出的基于 LT 码的算法.存储过程中该算法采用简单随机游走规则将 k 个源数据包传递到网络中的 n 个节点;每个节点按照 LT 码的度分度函数接收一定数量的源数据包,并计算得到一个存储数据包存储起来. Sink 节点出现后,通过访问这些存储数据包恢复出原来的源数据包.该方法主要有两方面的缺点:一是简单随机游走的低效率以及算法本身的需求,导致每个源数据包为了遍历网络中所有的节点,在实际过程中至少需要传递 $3n \ln n$ 次.因此,这增加了网络的通信成本.二是 Sink 节点为了恢复出所有的 k 个源数据包,需要访问 $k + k\lambda$ 个存储数据包 ($\lambda > 0$,与 k 的取值有关).文献[6]的实验表明 k 的数量级大于等于 10^2 时, $k\lambda$ 的值大于 100.因此,这导致了 Sink 节点的访问成本较高.针对 LT 码算法的上述不足,尤其是考虑到如何降低网络的通信成本(有利于传感器网络节能),笔者提出了一种基于源数据包传递过程中并行定向随机游走规则和网络中每个节点对源数据包独立处理机制的分布式数据存储算法.

1 并行定向随机游走

为了用数值实验模拟和验证文中算法的有效性,将无人值守传感器网络抽象为一个连通的二维随机图^[7].各传感器节点用图中的各顶点表示.若两个传感器节点互为邻居节点,则用一条边连接图中对应的顶点;一个传感器节点的数据包传递到另一个节点的过程用随机几何图上一次随机游走表示.

二维随机图上的随机游走定义为按某一随机序列访问图中顶点的过程.一次随机游走开始于图中某一顶点,并且下一步要访问顶点是从当前访问顶点的邻居顶点中选取的;如果下一步要访问的顶点是从当前访问顶点的所有邻居顶点中等概率随机选取的,那么这样的随机游走称为简单随机游走.

与文献[6]中采用的简单随机游走规则不同,文中算法采用效率更高的并行定向随机游走规则^[8]在网络中传递每个源数据包.下面先介绍(串行的)定向随机游走规则,再介绍其并行化方式.

1.1 定向随机游走规则

按照定向随机游走规则,每当源数据包到达了当前访问的传感器节点 v 后,要从 v 的所有邻居节点中选出一个作为下一步要访问的节点 u .

如果用 $N(o)$ 表示任意一个节点 o 的邻居节点集合, $\delta(o)$ 表示节点 o 的邻居节点个数, $c(o)$ 表示定向随机游走目前已访问节点 o 的次数,那么,下一步将要访问的节点 u 的选取过程分为如下两步:

步骤 1 从当前访问节点 v 的邻居集合 $N(v)$ 中随机均匀地选择 2 个节点作为备选节点.

步骤 2 从 2 个备选节点(用集合 N' 表示)中选择满足下式的节点 u 作为下一步将要访问的节点.

$$c(u)/\delta(u) = \min_{u' \in N'} \{c(u')/\delta(u')\} .$$

1.2 并行定向随机游走规则

每个数据节点将其源数据包并行地向外出 m 个副本 ($0 < m < \ln n$),并分别按定向随机游走机制向前传递.当传递每个副本的定向随机游走的步数达到给定值后,该副本被丢弃.在传递过程中,同一源数据包的某一副本访问过某一节点 o 后,另一个副本再到达该节点时,该节点的访问次数 $c(o)$ 要增加 1.

对于一个连通的二维随机图,用 t 表示随机游走的步数,那么该随机游走对图的覆盖率(也就是访问到的不同顶点数占图中总的顶点数的比例),其理论值可以近似地表示^[9]为 $1 - \exp(-t/n)$.因此,对于无人值守传感器网络,当采用随机游走机制(串行或并行^[10])传递每个源数据包时,理论上传递次数达到 $O(n \ln n)$ 时,才能使得每个源数据包访问到网络中的每个节点.

2 高性能分布式数据存储算法

文献[6]提出的基于 LT 码的方法,为使每个源数据包遍历网络中所有 n 个节点,每个源数据包的传递

<http://www.xdxb.net>

次数(文中将其看做网络通信成本开销)必须达到 $O(n \ln n)$. 为降低这一开销,笔者提出了一种基于并行随机游走机制的高性能分布式存储算法,该算法只要求每个源数据包的传递次数达到 $O(n)$ 即可.

文中算法的基本原理为:①存储过程开始之前:每个节点(数据节点和非数据节点)都存储有一个初始值为 0 的存储数据包,用 Y_i 表示, $i = 1, 2, \dots, n$. 此外,每个数据节点都有一个源数据包,用 X_j 表示, $j = 1, 2, \dots, k$. 源数据包与存储数据包大小相同. ②存储过程开始后:从每个数据节点开始 m 个步数设定为 cn/m 的定向随机游走向前传递该数据节点的源数据包的 m 个副本;每当一个副本到达一个新的节点(即同一源数据包的所有副本都未访问过的节点)时,新节点按概率 $a \ln k/k$ 接收源数据包,并将接收的源数据包异或到自己的存储数据包中;不论是否接收,新节点接着按照定向随机游走规则将源数据包的副本继续向前传递. 当副本传递次数达到定向随机游走的给定步数 cn/m 后,该副本被丢弃,不再向前传递. 当所有的 k 个源数据包的 km 个副本都被丢弃后,整个存储过程完成.

2.1 算法流程

文中算法的具体实现流程用代伪码表示如下(命名为算法 1):

算法 1 无人值守传感器网络的高性能分布式存储算法

输入: k 个源数据包 $X_v, v = 1, \dots, k$.

输出: n 个存储数据包 $Y_u, u = 1, \dots, n$.

/* 初始化阶段 */

1: For 每个数据节点 $v = 1$ to k

2: 为源数据包 X_v 添加头信息: ID_v 号及定向随机游走步数计数器 $N = 0$;

3: For 每个节点 $u = 1$ to n

4: 初始化每个存储数据包的值: $Y_u = 0$;

5: 初始化每个源数据包 X_v 已访问节点 u 的当前次数: $c_v(u) = 1, v = 1, \dots, k$; /* 分布式存储阶段 */

6: For 每个数据节点 $v = 1$ to k

7: 按照概率 $a \ln k/k$ 接收 X_v , 并且更新自己的存储数据包: $Y_v = Y_v \oplus X_v$;

8: 按照定向随机游走规则将源数据包 X_v 的 m 个副本分别传递出去;

9: $c_v(v) = c_v(v) + 1$;

10: For 每个节点 $u = 1$ to n

11: For 每个到达节点 u 的源数据包 X_j

12: If X_j 是第一次访问节点 u

13: 节点 u 按照概率 $a \ln k/k$ 接收 X_j , 并且更新自己的存储数据包: $Y_u = Y_u \oplus X_j$;

14: $c_j(u) = c_j(u) + 1$;

15: 源数据包 X_j 更新头信息: $N = N + 1$;

16: If $N < cn/m$

17: 节点 u 按照定向随机游走规则将源数据包 X_j 传递到它的一个邻居节点;

18: Else

19: 节点 u 丢弃 X_j ;

算法结束.

算法 1 完成之后,每个节点存储了一个存储数据包,源数据包不再存储. 算法有两个重要参数:一个是定向随机游走步数,即每个源数据包副本的传递次数,设置为 cn/m ,用变量 N 来计数;副本每传递一次 N 的值加 1,当 $N > cn/m$ 时,副本被丢弃,不再传递. 另一个参数是每个节点接收一个新到达的源数据包副本的概率,算法中设置为 $a \ln k/k$. 算法性能主要由这两个参数决定. 下面做具体分析.

2.2 算法性能分析

当算法 1 执行完成之后, k 个源数据包被分布式地存储到网络的 n 个节点中,每个节点存储的存储数据包都是若干个源数据包的异或和(每个源数据包和存储数据包都看做是二元域上的比特向量). 下文将分析 Sink 节点收集到多少个存储数据包之后可以恢复原来的 k 个源数据包.

<http://www.xdxb.net>

假定 Sink 节点出现后,随机地从 $k + \epsilon$ 个未损坏节点中收集了 $k + \epsilon$ 个存储数据包. 为便于描述,这 $k + \epsilon$ 个存储数据包表示为 $Y_i, i = 1, 2, \dots, k + \epsilon$. 每个存储数据包都是源数据包的线性组合,因此,任意一个 $Y_i, i = 1, 2, \dots, k + \epsilon$, 都可以表示为下式(式中的乘法为二元域上的元素与二元域上的向量之间的数乘):

$$Y_i = g_i \cdot [X_1, X_2, \dots, X_k]^T, \quad (1)$$

其中, X_1, \dots, X_k 表示 k 个未知的源数据包, g_i 是一个独立的且取值在二元域上的随机行向量,称为存储数据包 Y_i 的生成行向量. 当 g_i 的某个元素 g_{ij} 取值为 1 时表示对应的源数据包 X_j 被接收. 假定步数为 $t = cn$ 的并行定向随机游走覆盖的各节点均匀地分布在网络中,那么,由算法 1 可知: g_i 中每个元素 $g_{ij}, j = 1, \dots, k$, 都是独立取值的,且都服从式(2)定义分布.

$$P_r(g_{ij} = r) = \begin{cases} (1 - \exp(-c))a(\ln k/k) & , r = 1 \\ 1 - (1 - \exp(-c))a(\ln k/k) & , r = 0 \end{cases} \quad (2)$$

这 $k + \epsilon$ 个存储数据包的生成行向量构成了一个 $(k + \epsilon) \times k$ 阶矩阵 $G_{(k+\epsilon) \times k}$, 称为这 $k + \epsilon$ 个存储数据包的生成矩阵,即 $G_{(k+\epsilon) \times k} = [g_1, g_2, \dots, g_{k+\epsilon}]^T$. 借助生成矩阵,这 $k + \epsilon$ 个存储数据包可以表示为

$$[Y_1, Y_2, \dots, Y_{k+\epsilon}]^T = G_{(k+\epsilon) \times k} \cdot [X_1, \dots, X_k]^T. \quad (3)$$

若生成矩阵 $G_{(k+\epsilon) \times k}$ 存在一个可逆 $k \times k$ 阶子矩阵,也就是 G 列满秩,则式(3)定义的方程组存在惟一解,解方程组后可求得 k 个未知的源数据包 X_1, \dots, X_k . 因此, Sink 节点可由任意 $k + \epsilon$ 个存储数据包计算得到 k 个未知源数据包的概率等于这 $k + \epsilon$ 个存储数据包的生成矩阵 $G_{(k+\epsilon) \times k}$ 列满秩的概率. 如果用 P^{failure} 表示二元域上的生成矩阵 $G_{(k+\epsilon) \times k}$ 列不满秩的概率,当式(2)成立时,得到如下结论:

引理 1 当 $G_{(k+\epsilon) \times k}$ 的每个元素的取值服从式(2)定义分布时, P^{failure} 的上界为

$$P^{\text{failure}} \leq \sum_{w=1}^k \binom{k}{w} \cdot \left\{ \sum_{s=0,2,\dots,2\lfloor w/2 \rfloor} \binom{w}{s} \cdot \left((1 - \exp(-c))a \frac{\ln k}{k} \right)^s \cdot \left(1 - (1 - \exp(-c))a \frac{\ln k}{k} \right)^{w-s} \right\}^{k+\epsilon}. \quad (4)$$

证明 若 $G_{(k+\epsilon) \times k}$ 的各列线性相关,则 $G_{(k+\epsilon) \times k}$ 列不满秩,因此

$$P^{\text{failure}} = P_r \{ \exists x \in F_2^k, x \neq \mathbf{0}: G_{(k+\epsilon) \times k} x^T = \mathbf{0} \} \leq \sum_{x \in F_2^k, x \neq \mathbf{0}} P_r \{ G_{(k+\epsilon) \times k} x^T = \mathbf{0} \}, \quad (5)$$

令 R 表示 $G_{(k+\epsilon) \times k}$ 的任意一个行向量,用 w 表示向量 x 中 1 的个数,因为行向量 R 的每一个元素取值独立,且取值为 1 的概率由式(2)定义,因此

$$P_r \{ R x^T = \mathbf{0} \} = \sum_{s=0,2,\dots,2\lfloor w/2 \rfloor} \binom{w}{s} \left((1 - \exp(-c))a \frac{\ln k}{k} \right)^s \left(1 - (1 - \exp(-c))a \frac{\ln k}{k} \right)^{w-s}. \quad (6)$$

因为矩阵 $G_{(k+\epsilon) \times k}$ 的每一个行向量独立,因此, $G_{(k+\epsilon) \times k} x^T = \mathbf{0}$ 的概率为

$$P_r \{ G_{(k+\epsilon) \times k} x^T = \mathbf{0} \} = \left(\sum_{s=0,2,\dots,2\lfloor w/2 \rfloor} \binom{w}{s} \left((1 - \exp(-c))a \frac{\ln k}{k} \right)^s \left(1 - (1 - \exp(-c))a \frac{\ln k}{k} \right)^{w-s} \right)^{k+\epsilon}. \quad (7)$$

当向量 x 取所有不同的值时,根据式(5)可得到引理 1 中的结论.

综合上述分析,可用如下定理 1 描述算法 1 的性能.

定理 1 采用算法 1, Sink 节点可由随机选取的 $k + \epsilon$ 个存储数据包计算得到原来的 k 个源数据包的概率 P^{success} 满足:

$$P^{\text{success}} \geq 1 - \sum_{w=1}^k \binom{k}{w} \left\{ \sum_{s=0,2,\dots,2\lfloor w/2 \rfloor} \binom{w}{s} \left((1 - \exp(-c))a \frac{\ln k}{k} \right)^s \left(1 - (1 - \exp(-c))a \frac{\ln k}{k} \right)^{w-s} \right\}^{k+\epsilon}. \quad (8)$$

另外,通过数值实验发现(鉴于篇幅关系,这里不再列出相关实验结果),当式(8)中的参数 $(1 - \exp(-c))a \geq 2$ 时, P^{success} 的下界主要由 ϵ 的值决定,可以用简化表达式 $1 - (1/2)^\epsilon$ 近似表示 P^{success} 的下界.

3 数值实验

算法的数值实验在 MATLAB 环境下进行. 用随机生成的二维随机图模拟网络, 在 $L \times L = 100 \times 100$ 的正方形区域随机均匀地分布 n 个点模拟传感器网络各个节点, 其中数据节点的比例设置为 40%, 即 $k = 0.4n$. 每个节点的通信半径 R 满足 $R^2 = (2 \ln n / (\pi n)) L^2$ (按照随机几何图的理论, 这一条件保证了网络极大的连通概率). 实验分为两组, 第 1 组测试网络的通信成本, 即存储过程中每个源数据包在达到一定的网络覆盖率前提下需要在网络中的传递次数. 通信成本越低, 越有利于节约网络中各节点的通信能耗. 第 2 组测试读取源数据包时的访问成本, 即存储完成之后, Sink 节点需要访问多少个存储数据包才能恢复出原来的 k 个源数据包. 访问成本越低, 允许损坏的传感器节点就越多, 网络的可靠性就越高.

3.1 存储源数据包的通信成本实验

每个源数据包都按照并行随机游走机制在网络中传递, 每个源数据包并行传递的副本数设为 $m = 2$, 当 m 个副本总的传递次数, 即随机游走的总步数为 t 时, 网络覆盖率的理论值^[9] 近似为 $1 - \exp(-t/n)$. 实验中将并行定向随机游走的总步数设为 $t = cn$ 时, 单个副本的传递次数设置为 cn/m . 实验测试了 k 个源数据包对网络各节点的平均覆盖率与系数 c 之间的关系. 作为对比, 也测试了采用简单随机游走机制时的相关性. 图 1、图 2 所示为 n, k, c 取不同值时的实验结果.

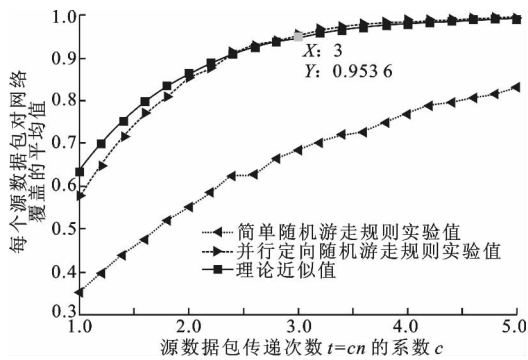


图 1 源数据包通信成本测试 1

(随机游走步数 $t = cn, n = 500, k = 200$)

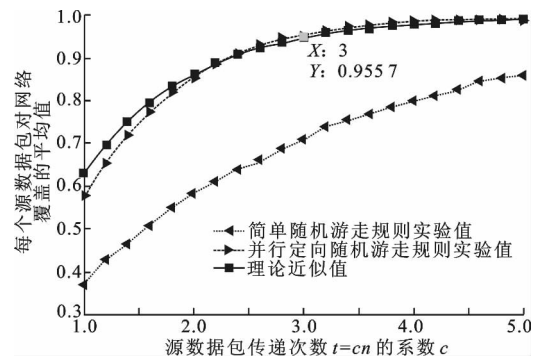


图 2 源数据包通信成本测试 2

(随机游走步数 $t = cn, n = 1000, k = 400$)

从图 1 和图 2 中可以看出, 采用并行定向随机游走机制, k 个源数据包对网络覆盖率平均值的实验值与理论值 $1 - \exp(-t/n)$ 是近似相等的. 当 $t = n$ 时, 实验值接近 60%, 当 $t = 3n$ 时, 基本上每个源数据包都能覆盖网络中超过 95% 的节点. 而采用简单随机游走规则, 实验值明显小于理论值, 比并行随机游走机制的网络覆盖率平均低约 20%. 因此, 采用定向随机游走机制能有效降低网络的通信成本. 同时, 采用并行定向随机游走机制时, 网络中每一时刻平均有 mk 个源数据包被同时处理, 提高了网络的吞吐率, 节省了整个存储过程的耗时; 而采用简单随机游走机制时, 网络中每一时刻只有 k 个源数据包处理.

需要说明的是, 当每个源数据包并行传递的副本数 m 小于 $\ln n$ 值时, 实验结果与图 1 和图 2 所示类似; 而当 m 大于 $\ln n$ 值时, 网络覆盖率随 m 的增大而减小.

3.2 读取源数据包的访问成本实验

测试存储过程完成之后, Sink 节点为得到 k 个源数据包而访问的存储数据包的个数. 实验结果用如下两个变量之间的关系描述: 源数据包恢复开销 η , 即 Sink 节点访问到的存储数据包个数 h 与源数据包个数 k 之差; 源数据包恢复成功率 S , 即 Sink 节点能从 h 个存储数据包计算出 k 个源数据包的概率.

实验中主要调节参数为每个源数据包的并行定向随机游走的总步数 cn 的系数 c 以及每个传感器节点接收一个新到达的源数据包的概率 $a \ln k / k$ 的系数 a . 图 3 至图 6 所示为不同网络规模下, c 和 a 的取值不同时, S 与 η 之间的关系(图中 S 的每个值都是 1000 次实验的平均值).

从图 3 至图 6 可以看出, 当参数 $a \geq 2$ 时, 只要 c 的取值增大, 就能以较少的源数据包恢复开销成功计

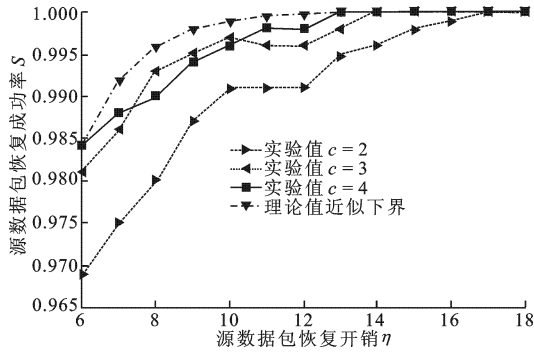


图 3 源数据包访问成本测试 1

($n = 500, k = 200$, 参数 $a = 2$)

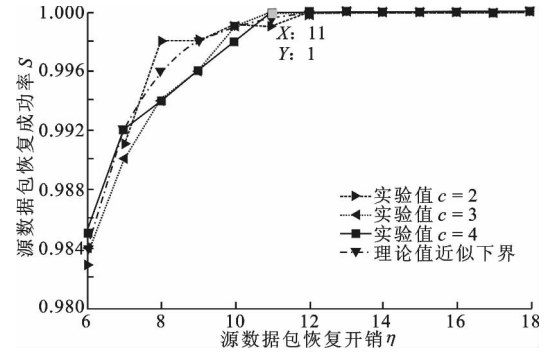


图 4 源数据包访问成本测试 2

($n = 500, k = 200$, 参数 $a = 3$)

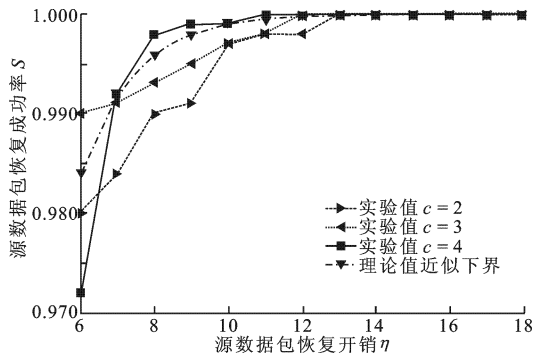


图 5 源数据包访问成本测试 3

($n = 1000, k = 400$, 参数 $a = 2$)

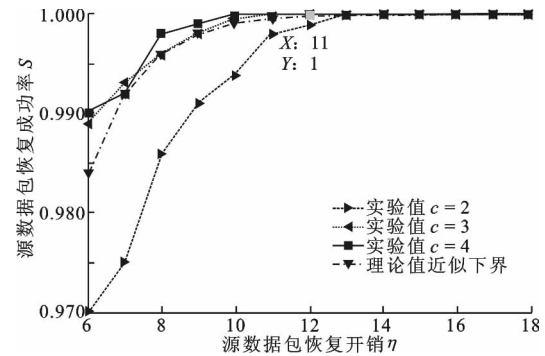


图 6 源数据包访问成本测试 4

($n = 1000, k = 400$, 参数 $a = 3$)

算出原来的源数据包. 进一步地, 从图 4、图 6 可以看出, 当参数 $a = 3$ 时, 只要 $c \geq 3$, Sink 节点就能从任意收集到的 $k + 11$ 个以上的存储数据包以 100% 的概率恢复出 k 个源数据包. 此时, 式 (8) 中的 $(1 - \exp(-c))^a \geq (1 - \exp(-3))^3 \approx 2.85$. 按定理 1 的结论, 当 $\eta \geq 11$ 时, 理论上 Sink 节点可由任意 $k + \eta$ 个存储数据包计算得到原来的 k 个源数据包的概率 S 近似大于 $1 - 2^{-11} \approx 99.95\%$, 实验值基本上验证了理论值.

鉴于篇幅有限, 参数 a 和 c 取其他值时的实验结果不在文中列出. 不过笔者发现, 当 $(1 - \exp(-c))^a$ 小于 2 时, 实验结果不稳定; 当 $(1 - \exp(-c))^a$ 取值不是很大时, 实验结果与 $(1 - \exp(-c))^3$ 时的类似; 而当 $(1 - \exp(-c))^a$ 取值较大时, 反而会使得成功恢复源数据包的概率降低. 因此, 建议在实际应用中, 取 $a = 3, c = 3$ 即可, 这样既能获得好的稳定性, 也能使网络各节点的计算成本较低 (因为 a 的值越大, 每个节点接收源数据包的概率就越大, 因而计算成本就会提高).

3.3 性能比较

文献 [6] 提出的基于 LT 码的算法中, 单个源数据包在网络中总的传递次数为 $O(n \ln n)$, 并指出这一值达到 $3n \ln n$ 时实验效果较好. 而文中方法所需传递次数为 $O(n)$, 实验中将这一值设置为 $3n$ 时就能得到很好的源数据包恢复性能. 另外, 采用基于 LT 码的算法, Sink 节点为了恢复出 k 个源数据包而需要访问的存储数据包的个数与 LT 码的译码性能有关. 文献 [6] 的实验表明, 当 k 大于 100 时, 只有当源数据包恢复开销 η 的值较大 (大于 100) 时, Sink 节点才能成功恢复出源数据包. 而采用文中方法, 不论 k 的取值如何, 成功恢复 k 个源数据包, Sink 节点只需访问 $k + 11$ 个存储数包. 对于某些实验参数, 文中算法 (源数据包传递次数设置为 $3n$, 参数 a 的值设置为 3) 与基于 LT 码算法 (源数据包传递次数设置为 $3n \ln n$, 译码时采用极大似然译码算法) 的具体对比结果列于表 1 中.

综上所述得出结论: 采用文中算法可以有效节约存储过程中的通信成本和 Sink 节点的访问成本.

(下转第 185 页)

IEEE, 2011; 1-4.

- [11] 裘君, 赵光宙, 齐东莲. PMSM 的哈密顿反馈耗散增益调度控制[J]. 控制工程, 2010, 17(2): 149-153.
Qiu Jun, Zhao Guangzhou, Qi Donglian. PMSM Gain Scheduling Control Based on Hamiltonian Feedback Dissipation [J]. Control Engineering of China, 2010, 17(2): 149-153.

(编辑: 齐淑娟)

(上接第 178 页)

表 1 性能比较

	通信成本		访问成本				
	理论值	实验值	理论值	实验值			
				$n = 500$ $k = 200$	$n = 700$ $k = 280$	$n = 1000$ $k = 400$	$n = 1200$ $k = 480$
文中算法	$O(n)$	约 $3n$	$k + \epsilon$	约 $k + 11$	约 $k + 11$	约 $k + 11$	约 $k + 11$
基于 LT 码算法	$O(n \ln n)$	约 $3n \ln n$	$k + k\lambda$	约 $k + 100$	约 $k + 130$	约 $k + 130$	约 $k + 160$

4 结束语

笔者研究了无人值守传感器网络的高可靠性数据存储问题, 依靠源数据包传递过程中的并行定向随机游走机制和网络中每个节点对源数据包的独立处理机制, 提出了一种执行简单, 性能高效的分布式数据存储算法. 与同类方法相比, 文中方法能有效降低存储过程中各节点的通信成本和 Sink 节点的访问成本.

参考文献:

- [1] Reddy S K V L, Ruj S, Nayak A. Distributed Data Survivability Schemes in Mobile Unattended Wireless Sensor Networks [C]//Proceedings of IEEE GLOBECOM. Piscataway: IEEE, 2012; 979-984.
- [2] 郭江鸿, 马建峰, 张留美, 等. 高效的无线传感器网络加密数据汇聚方案 [J]. 西安电子科技大学学报, 2013, 40(3): 95-101.
Guo Jianghong, Ma Jianfeng, Zhang Liumei, et al. Efficient Encrypted Data Aggregation Scheme for Wireless Sensor Networks [J]. Journal of Xidian University, 2013, 40(3): 95-101.
- [3] Mitra S, De Sarkar A, Ray S. A Review of Fault Management System in Wireless Sensor Network [C]//Proceedings of the CUBE International Conference on Information Technology. New York: ACM, 2012; 144-148.
- [4] Vitali D, Spognardi A, Mancini L V. Replication Schemes in Unattended Wireless Sensor Networks [C]//Proceedings of 4th IFIP International Conference on New Technologies, Mobility and Security. Piscataway: IEEE, 2011; 1-5.
- [5] 任伟, 任毅, 张慧, 等. 无人值守无线传感器网络中一种安全高效的数据存活策略 [J]. 计算机研究与发展, 2009, 46(12): 2093-2100.
Ren Wei, Ren Yi, Zhang Hui, et al. A Secure and Efficient Data Survival Strategy in Unattended Wireless Sensor Network [J]. Journal of Computer Research and Development, 2009, 46(12): 2093-2100.
- [6] Kong Z, Aly S A, Soljanin E. Decentralized Coding Algorithms for Distributed Storage in Wireless Sensor Networks [J]. IEEE Journal on Selected Areas in Communications, 2010, 28(2): 261-267.
- [7] Cooper C, Frieze A. The Cover Time of Random Geometric Graphs [J]. Random Structures and Algorithms, 2011, 38(3): 324-349.
- [8] Avin C, Krishnamachari B. The Power of Choice in Random Walks: An Empirical Study [J]. Computer Networks, 2008, 52(1): 44-60.
- [9] Tzevelekas L, Oikonomou K, Stavrakakis I. Random Walk with Jumps in Large-scale Random Geometric Graphs [J]. Computer Communications, 2010, 33(13): 1505-1514.
- [10] Cooper C, Frieze A, Radzik T. Multiple Random Walks in Random Regular Graphs [J]. SIAM Journal on Discrete Mathematics, 2009, 23(4): 1738-1761.

(编辑: 李恩科)

<http://www.xdxn.net>