

移动低占空比无线传感器网络中低时延的数据持续性提高算法

蒋婵^{1,2}, 李陶深^{1,2}, 梁俊斌²

(1. 华南理工大学电子与信息学院, 广东 广州 510641;

2. 广西大学计算机与电子信息学院 广西多媒体通信与网络技术重点实验室, 广西 南宁 530004)

摘 要: 移动低占空比无线传感器网络是近年来出现的新型网络。在移动低占空比无线传感器网络中, 由于节点的存储空间有限, 并且节点的移动及睡眠会导致网络不连通、数据无法及时传输等问题, 使数据很难被快速分发并存储, 数据持续性较低。为此, 提出一种卢比变换码的分布式数据存储 (LT-MDS, Luby transform codes based mobile distributed storage) 算法, 该算法采用一种新的传染病式数据分发方法在节点不断移动的网络中分发数据, 使数据能以较低的时延被网络中绝大部分节点接收到, 提高了网络的可靠性; 节点在接收到数据的同时, 利用卢比变换码 (LTC, Luby transform code) 对数据进行编码存储, 使容量有限的节点可以保存更多的数据信息。理论分析和仿真实验表明, LT-MDS 算法能够以低时延完成数据分发和存储, 同时获得较高的数据持续性。

关键词: 移动低占空比无线传感器网络; 数据持续性; 低时延; 传染病式数据分发; 数据存储

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2018041

Low-latency algorithm for improving data persistence in mobile low-duty-cycle wireless sensor network

JIANG Chan^{1,2}, LI Taoshen^{1,2}, LIANG Junbin²

1. School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China

2. Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer and Electronics Information, Guangxi University, Nanning 530004, China

Abstract: Mobile low-duty-cycle wireless sensor network (MLDC-WSN) are a kind of new ad hoc networks that are appeared in recent years. In MLDC-WSN, the nodes only have limited storage spaces. Moreover, the nodes would move or sleep from time to time. Therefore, these networks have some problems such as connectivity is hard to be maintained and data are hard to be transmitted to their destinations for storage in time. As a result, data persistence (i.e., the probability that all data can be recovered after some nodes die in the networks) is low. A distributed algorithm named LT-MDS for improving data persistence in MLDC-WSN was proposed. The algorithm used a new infectious data dissemination method to transmit the data, which enabled the data to be received by almost all the mobile nodes in a network with low latency and improved the reliability of the network. When a node receives the data, it would use LT (Luby transform) codes to encode and save them. By this way, the nodes with limited storage spaces can save more data information. Theoretical analyses and simulations show that LT-MDS can complete the process of data dissemination and preservation with low latency, and it can achieve high data persistence.

Key words: mobile low-duty-cycle wireless sensor network, data persistence, low latency, infectious data dissemination, data preservation

收稿日期: 2017-09-15; 修回日期: 2018-01-21

基金项目: 国家自然科学基金资助项目 (No.61562005, No.61762010, No.61363067); 广西自然科学基金资助项目 (No.2015GXNSFAA139286); 广西高等学校千名中青年骨干教师培育计划资助项目 (桂教人(2017)No.49)

Foundation Items: The National Natural Science Foundation of China (No.61562005, No.61762010, No.61363067), The Natural Science Foundation of Guangxi Zhuang Autonomous Region (No.2015GXNSFAA139286), The Cultivation Plan For Thousands of Young and Middle-Aged Backbone Teachers in Guangxi Higher Education School (Guangxi Education People (2017) No. 49)

1 引言

无线传感器网络^[1] (WSN, wireless sensor network) 由大量部署在监测区域内的微型传感器节点构成, 是一种能够长期执行环境监测、事件跟踪等任务的无线多跳自组织网络, 在国防、军事、农业、工业等领域具有广泛的应用。在 WSN 中, 节点通常只有有限的通信、计算和存储能力, 它们的能量水平也较低。移动低占空比无线传感器网络^[2] (MLDC-WSN, mobile low-duty-cycle wireless sensor network) 是近年来出现的新型 WSN, 它可以把节点部署在移动的物体上, 利用物体的移动性来采集范围更广且动态的环境信息。例如, 在牧场环境监测应用中, 节点挂在每只羊的耳朵上, 整个羊群组成一个移动的 WSN。为了减少能量耗费, 它使节点长时间处于睡眠状态, 只在周期性唤醒时工作和通信, 可以极大地延长节点的工作寿命。这种工作模式就称为低占空比模式。

MLDC-WSN 一般部署在条件恶劣的环境, 节点容易遭到外力破坏而丢失数据。例如, 当羊群遇到雷暴雨时, 羊身上的节点容易损坏。因此, 在 MLDC-WSN 中, 网络一般没有固定的 Sink 来实时收集节点的数据。为了避免节点损坏而丢失数据, 一个可行的方法是采用冗余存储, 即一个节点在感知到数据后, 立即将该数据发送给网络中其他节点进行保存。这样即使这个节点损坏, 仍然可以从其他节点获取它的数据。

但是, 网络中往往有多个节点感知到数据 (即存在多个 (源) 数据), 而节点的存储容量有限, 因此, 很难存放多个数据。此外, 由于节点的移动性, 网络不一定能时刻保持连通, 数据分发过程可能被中断。因此, 如何有效地规划节点分发数据和存储数据的方式, 提高数据持续性, 是目前研究的一个挑战。其中, 数据持续性是指网络在节点随着工作时间增长而不断损坏的情况下, 全部源数据能够被恢复出来的能力。

本文提出一种面向 MLDC-WSN 的基于卢比变换码^[3] (LTC, Luby transform code) 的分布数据存储算法 (LT-MDS, Luby transform code based mobile distributed storage), 可以快速完成数据分发和存储, 并且增加节点有限存储空间里的数据量, 有效提高数据持续性。其中, LTC 是一种网络编码, 具有较低的编码和解码复杂性, 利于在计算能力有限的节

点上实现。最后, 通过理论分析和仿真实验证明了 LT-MDS 的有效性。

2 相关工作

目前, 已经有一些工作研究如何利用网络编码存储数据以提高数据持续性。文献[4]分析了网络编码技术的有效性, 通过对比传统的编码方式 (如 RSC (RSC, reed-solomon code)、随机线性码 (RLC, random linear code) 等) 得到以下结论。各种网络编码均能实现数据的持续性, 但是不同编码方式要求的存储空间不同, 例如, RSC 需要的存储空间要比 RLC 的高。增长码^[5] (GC, growth code) 是一种具有较低存储空间要求的网络编码, 它通过节点与邻居之间的信息交互, 不断增加存储的编码数据的信息量。但是, 随着时间的增长, 节点中存储的编码数据信息量越来越大, 解码也越来越困难, 不利于数据持续性的提高。文献[6]提出了优先随机线性码 (PRLC, priority random linear code), 通过具有优先级的随机线性码来提高重要数据的持续性。它对重要性越高的数据采用越低的编码率, 即减少多个数据合并成一个编码数据的概率, 使重要的数据可以更容易地被恢复出来。文献[7]提出了地理随机线性码 (GRLC, geometric random linear code), 利用节点的地理信息, 使距离事件发生地点越近的区域, 节点存储的信息越多, 这样便于数据采集者在事件区域附近进行数据恢复和收集。文献[8]提出了结构化随机线性码 (SRLC, structured random linear code), 利用编码数据中信息分布的稀疏性, 重新设计码字的结构, 可以有效降低解码复杂性。文献[9]提出了一种基于 RLC 且能提供一定容错率的分布式存储方案。文献[10]设计了一种基于压缩感知 (CS, compressive sensing) 的编码方法, 进一步减少编码数据的容量, 以此增加网络中保存的信息量。

以上编码方式的优点是便于分布式实现, 节点只需要和邻居节点交换信息即可完成数据的编码存储。但是, GC、RSC 和各种 RLC 的解码复杂度仍然较高, 达到 $O(k^3)$, 影响了数据持续性的提高, 其中, k 是源数据的数量。为了进一步提高数据持续性, 需要考虑解码复杂度更低的编码方式。喷泉码 (FC, fountain code) 是一种具有较低的编码和解码复杂度 $O(k \ln k)$ 的编码方式, 目前受到越来越多的关注。LTC 是喷泉码中最典型的实现方式, 节点可

以根据特定的概率分布选择需要编码存储的数据的数量, 然后根据从源数据随机挑选出相应数量的数据进行异或 (XOR) 操作即可。

EDFC^[11]是第一种基于 LTC 的数据持续性提高算法, 它使源节点通过随机行走 (RW, random walk) 向网络中以多条路径分发数据并进行存储。但是, 由于随机行走覆盖整个网络需要相当长的时间和大量的数据通信, 完成数据分发的时延很大, 而且节点的能耗相当大。因此, EDFC 并不利于快速实现数据持续性, 且能量有效性较低。RCDS^[12]针对 EDFC 进行改进, 通过限定随机行走的长度, 减少数据分发时延和节点通信的能耗。此外, 在数据分发前, 利用低密度奇偶校验码^[13] (LDPC, low-density parity check code) 对数据进行预编码, 降低数据恢复的解码复杂度。文献[14]提出了一种多数据副本的分发方案, 使源节点可以将多个数据副本随机分发到附近的邻居, 减少了编码过程中网络节点的能量消耗。文献[15]设计了一种基于速龙码 (RC, raptor code) 的数据保存方案。RC 也是一种喷泉码, 具有比 LTC 更低的解码复杂度, 但是它要求对数据进行预处理, 需要节点间进行更多的数据交互。

为了进一步降低数据分发时延和节点的能耗, 一些研究人员采用广播 (broadcast) 来分发数据。文献[16]提出了一种分布存储算法 (DSA, distributed storage algorithm)。假设网络中有 2 种节点: 数据 (源) 节点和存储节点。源节点将通过广播, 发送自己的数据给通信范围内的存储节点进行保存。DSA 要求存储节点具有较大的存储空间, 因此, 不利于在存储容量有限的节点上实现。文献[17]提出了一种带随机能量控制的纠删码 (ECPC, erasure coding with randomized power control), 考虑了网络断裂的情况, 节点进行数据中继时, 可以预测网络拓扑的连通性, 并根据预测的结果调整广播的传输距离, 使数据最终能以高概率被所有节点接收到并进行编码存储。文献[18]提出了一种自适应的基于概率广播的协议 (APBDP, adaptive probabilistic broadcast based protocol), 节点可以通过统计邻居的数量, 以概率决定是否转发数据, 不需要网络中的全局信息, 实现完全分布式的数据分发。由于网络中仅有一小部分节点转发数据, 不仅降低了数据分发时延, 还减少了网络的总能耗。文献[19]提出了根据节点能量水平来决定数据转发概率的概率广播方案, 并以此为基础设计了基于 LTC 的数据保

存机制。

以上的工作均是在传统 WSN 上进行研究的, 没有考虑节点的移动性和低占空比特性, 很难在 MLDC-WSN 中应用。因此, 需要考虑新的数据分发方式, 并结合网络编码的特点, 设计新的数据持续性提高算法。

另一方面, 针对移动分布式网络中的数据分发, 目前已有大量的工作, 如各类广播、随机行走、地理路由等, 具体可参考文献[20]。但是, 目前的数据分发方法主要针对传统 WSN, 很难应用于 MLDC-WSN。例如, 对于广播操作, 传统 WSN 中一个节点收到数据后立即转发, 它的邻居均可接收到数据; 而在 MLDC-WSN, 如果节点接收到一个数据并立即转发它, 可能邻居节点均在休眠状态而无法接收到数据。

在 MLDC-WSN 中, 目前常用的数据分发机制为基于分组的传送 (GT, group-based transmission)^[21,22] 和概率传送 (PT, probabilistic transmission)^[23,24]。其中, GT 是使网络中的节点根据地理距离形成若干分组, 然后在每个分组中选择一个组长, 负责将数据逐一分发给其他时刻唤醒的组内节点, 而数据将通过组内的边缘节点扩散到其他分组。但是, 这种方法在移动的网络中维持分组信息并不容易。PT 是由数据发送节点根据一定的概率选择若干个邻居进行传送, 这些邻居在收到数据后, 将根据自己邻居的唤醒时刻、是否接收过数据等情况, 再以概率选择若干个邻居进行转发, 直至在规定时刻内没有找到合适的接收者为止。概率传送容易分布式实现, 但是它的数据接收率不高, 即数据分发结束后, 仍然有很多节点未收到数据。

3 系统模型和问题描述

3.1 网络模型

假设网络中存在 n 个节点, 它们随机分布在一个大小为 $M \times M$ 的正方形区域。节点的通信半径为 r 。如果 2 个节点之间的地理距离不大于 r , 则它们可以相互通信。但是, 网络不一定保持随时连通, 即 2 个节点间不一定总是存在一条通信路径。

节点保持随机的运动模式。目前有多种随机运动模型, 如随机位点模型 (RWM, random waypoint model)、随机行走模型 (RW, random walk)、随机方向模型 (RDM, random direction model) 等。由于 RDM 与现实世界当中的动物种群运动具有较高相

似度^[25], 适合本文研究的背景, 因此本文主要采用该模型。在随机方向模型中, 节点会周期性地在其为中心的点上, 从 $(0, 2\pi)$ 范围内随机选择一个方向, 然后再以固定的速度 v 行走指定的时间长度。值得注意的是, 无论采用哪种模型, 均不会影响本文算法的实现和性能。本文算法可以在任何随机运动模型上实现。

与文献[26]和文献[27]一样, 网络中的数据收集按轮 (round) 运行。每轮有固定的时间长度, 并且被分为以下 3 个阶段。

1) 数据获取阶段。在该阶段, 节点处于唤醒状态, 不断感知外部环境。如果监测到目标事件, 则将监测的数据保存在自己的存储空间。然后, 广播一个短消息通知网络中其他节点。该阶段结束时, 网络中存在 k 个感知到数据的节点 (即源节点), 而网络中每个节点均知道 k 的大小。

2) 数据分发及存储阶段。节点进入低占空比状态, 尽可能保存自己的能量。每个源节点需要将数据以能量有效且低时延的方式分发给网络中的其他节点, 而其他节点在接收到数据时将其进行编码存储。该阶段是本文重点关注的阶段。

3) 数据收集阶段。节点进入睡眠状态, 只有当移动数据收集器经过它附近并唤醒它时, 它才重新启动并发送自己的数据给数据收集器。数据发送完毕, 节点又重新睡眠。该阶段的时间长度远远大于第一和第二阶段, 具体由应用场景决定。在本阶段, 部分节点会被损坏而丢失数据。而移动数据收集器可以在任意时刻, 从任意地点进入网络并访问仍然完好的节点。

节点进入低占空比状态时, 将自己的时间划分为多个固定长度的周期 T , 每个周期 T 中包含 m 个固定长度的时间单元 (time unit)。节点随机选择一个时间单元唤醒, 而在其他时间单元保持睡眠。由于不同的节点进入低占空比状态的时间不一样, 它们之间的通信是异步的。由于网络中的节点是可移动的, 因此, 网络拓扑不断在变化。此外, 由于在任意时刻网络中均有大量节点处于睡眠状态, 因此节点很难通过信息交互而获知邻居的情况 (如数量、唤醒时间等)。

每个节点在每轮感知到的数据大小为 a bit, 而节点的存储空间只能保存一个数据。不同节点感知到的数据之间没有相关性, 不同的数据不能进行汇聚和融合操作。此外, 节点也没有足够的计算能

力将数据进行压缩。如果需要保存多个数据, 节点可以通过异或 (XOR) 操作将它们合并为一个编码数据。

网络中存在可靠的 MAC 协议如 E^2 -MAC^[28]能有效处理多个节点同时给同一个目标节点发送数据时的通信冲突。

3.2 问题描述

本文研究的主要问题是给定一个具有 n 个节点的 MLDC-WSN, 假设在数据获取阶段有 k 个节点感知到了数据, 则在数据分发及存储阶段, 这 k 个源节点如何以能量有效且低时延的方式, 分发数据给网络中的其他节点进行编码存储。本文目标是在数据收集阶段, 移动数据收集器只要访问尽量少的节点, 就能恢复出全部的源数据。

4 算法设计

4.1 算法的基本思想

提高数据持续性主要包括 2 个主要操作: 数据分发和编码存储。因此, 本文提出的分布数据存储算法 LT-MDS 的基本思想如下。

1) 针对 MLDC-WSN, 由于节点不断移动可能会导致网络断裂或部分节点无法接收到数据, 提出一种新的传染病式数据分发 (IDD, infectious data dissemination) 方法。IDD 的基本思想是采用被动式广播操作, 即每个节点唤醒时通知其附近的转发节点 (有数据需要发送的节点), 转发节点进而判断该节点是否接收过它的数据, 如果没有则将数据发送给该节点。被动式广播不需要转发节点记录邻居节点的情况 (如唤醒的时刻), 也不需要通信的同步, 适合在异步的移动网络中分发数据。

2) 在数据分发过程中, 本文将有数据需要发送的节点看作感染对象 (IO, infectious object), 而没有数据需要发送的节点看作易受感染的对象 (SO, susceptible object)。一个 SO 根据自己的占空比被唤醒时, 将发送一个消息通知自己的邻居。如果邻居中存在一个 IO, 则该 IO 判断是否以前将数据发送给过这个 SO, 如果没有则进行数据发送, 否则, 不做任何操作。SO 接收到数据后, 则被传染, 它转变为 IO 并保持唤醒状态一段时间 t 。 t 时间过后, IO 恢复为 SO, 称为康复 (recover)。SO 可重新进入占空比状态。当网络中没有 IO 时, 则数据分发过程结束。

3) 节点在第一次接收到一个数据时, 将根据

LTC 进行存储。但是, LTC 要求编码数据由 k 个源数据产生, 而节点的存储容量有限, 不可能在接收到全部源数据后再编码存储。因此, 本文根据概率选择需要保存的源数据。该概率由 LTC 及源数据的数量 k 共同决定。

下面, 将详细介绍算法的设计与实现, 并分析它的性能。

4.2 LTC 简介

本文算法主要根据 LTC 存储数据, 为此在介绍算法的设计之前, 首先简单介绍 LTC 的原理。LTC^[3] 属于喷泉码的一种, 它是一种无率的纠删码 (rateless erasure code)。给定 k 个源数据 $\{x_1, x_2, \dots, x_k\}$, LTC 可以组合它们产生无限个编码数据 $\{y_1, y_2, \dots\}$ 。组合的过程是利用一个概率分布 $\Omega(j)$ 来为每个编码数据 y_m 产生一个编码度 d_m , $1 \leq j \leq k$, $1 \leq d_m \leq k$, 而每个编码数据 y_m 由 d_m 个随机选择的源数据通过异或操作合并产生。每个编码数据都是独立和有意义的, 解码时只需要获得 $k + \varepsilon$ 个编码数据, 就能够以 $1 - \delta$ 的概率恢复出全部的源数据, 不用关心编码数据获取的次序, 其中, $\varepsilon = O\left(\sqrt{k} \ln^2\left(\frac{k}{\delta}\right)\right)$, $0 < \delta < 1$ 。

在 LTC 中, $\Omega(j)$ 主要采用顽健孤子分布 (RSD, robust soliton distribution), 它的形式化定义如下。首先, 定义一个理想孤子分布 (ISD, ideal soliton distribution) 为

$$\Omega_{\text{is}}(j) = \begin{cases} \frac{1}{k}, & j = 1 \\ \frac{1}{j(j-1)}, & j = 2, 3, \dots, k \end{cases} \quad (1)$$

然后, 再定义一个变量 $R = c \ln\left(\frac{k}{\delta}\right) \sqrt{k}$, $c > 0$ 是一个常量, 有

$$\tau(j) = \begin{cases} \frac{R}{jk}, & j = 1, \dots, \frac{k}{R} - 1 \\ R \frac{\ln\left(\frac{R}{\delta}\right)}{k}, & j = \frac{k}{R} \\ 0, & j = \frac{k}{R} + 1, \dots, k \end{cases} \quad (2)$$

RSD 表示为

$$\Omega(j) = \frac{\tau(j) + \Omega_{\text{is}}(j)}{\beta}, \quad j = 1, 2, \dots, k \quad (3)$$

其中, $\beta = \sum_{j=1}^k (\Omega_{\text{is}}(j) + \tau(j))$ 。

4.3 LT-MDS 算法的详细描述

LT-MDS 是一种完全分布式的算法, 在每个节点 v_i 上独立运行。网络中的节点将通过消息通信来完成相互之间的协作。算法的具体实现如算法 1 所示。

算法 1 LT-MDS 算法

当数据分发及存储阶段开始时:

- 1) 设置 v_i 的状态为 “Susceptible”;
- 2) 初始化 v_i 的编码数据 $y_i = \text{“”}$;
- 3) 根据 RSD 计算 d_i 的值;
- 4) if v_i 在数据获取阶段感知到了一个源数据 x_i
- 5) $y_i = x_i$;
- 6) $d_i = d_i - 1$;
- 7) 设置 v_i 的状态为 “Infected”;
- 8) v_i 保持唤醒;
- 9) 设置计时器 $timer = t$;
- 10) end if

当计时器 $timer$ 倒数完成时:

- 1) 设置 v_i 的状态为 “Susceptible”;
- 2) v_i 进入占空比工作状态。

当 v_i 在它的工作周期中唤醒时:

发送一个短消息 $wakeup(v_i)$ 给自己的邻居。

当接收到一个邻居 v_j 发送来的短消息 $wakeup(v_j)$:

1) if v_i 的状态为 “Infected” 并且 v_i 没有发送接收到的数据 x_k 给 v_j ;

2) 发送一个消息 $Sensed_data(x_k)$ 给 v_j ;

3) 记录 x_k 已经被发送给 v_j ;

4) end if.

当接收到一个消息 $Sensed_data(x_k)$ 时:

1) if (x_k 是第一次被接收到)

2) $temp = rand(1)$;

3) if ($temp < \frac{d_w}{k}$) // 根据 RSD 将 x_k 以概率保存

到编码数据 y_i

4) $y_i = y_i \text{ XOR } x_k$;

5) end if

6) 设置 v_i 的状态为 “Infected”;

7) v_i 保持唤醒;

8) 设置计时器 $timer = t$;

9) end if

在算法 1 中, 当节点 v_i 进入数据分发及存储阶

段时，它的状态设为“Susceptible”，即易被感染状态。此状态说明节点可以接收其他节点发送来的数据。同时，节点根据 RSD 计算自己的代码度 d_i 。如果 v_i 是源节点，则当前的编码数据 y_i 就只有源数据 x_i ，同时将代码度减 1，表示已经存储一个数据。由于源节点持有数据， v_i 的状态变为“Infected”并保持唤醒状态 tS ，等待邻居节点醒来接收数据。当 v_i 保持“Infected”的状态达到 tS ，可以康复并转变为“Susceptible”，同时进入低占空比状态以保存能量。当 v_i 从低占空比状态中唤醒，它将广播一个携带其 ID 的短消息给自己的邻居，通知它可以接收数据。如果 v_i 接收到从另一个邻居 v_j 发送来的短消息，它检查自己的状态和数据发送记录。如果发现自己的状态为“Infected”并且自己需要转发的数据 x_k 没有发送给 v_j ，则将 x_k 发送给 v_j ，并记录这次发送。当 v_i 接收到一个数据 x_k ，它将判断 x_k 是否第一次接收到。如果是，则以概率 $\frac{d_w}{k}$ 存储该数据。 $\frac{d_w}{k}$ 表明 v_i 需要从 k 个源数据中随机挑选 d_w 个数据进行存储，目的是使自己的代码度符合 LTC 所遵循的 RSD 的要求。无论 v_i 是否对 x_k 进行存储，它都需要继续转发 x_k 。因此， v_i 的状态转变为“Infected”并保持唤醒状态 tS 。 tS 过后， v_i 康复并转变为“Susceptible”，进入低占空比状态以保存能量。当网络中没有状态为“Infected”的节点，算法结束。

4.4 性能分析

定义 1 传染度 F 指一个状态为“Infected”的节点 v_i 在 t 时间内传染的节点的数量，即 t 时间内接到 v_i 发送的数据 $B = \pi r^2 + 2tvr$ 的节点数量。

在随机方向模型中，一个状态为“Infected”的节点在 t 时间内通信的区域如图 1 所示，其中， r 为节点的通信半径， v 为节点的速度。

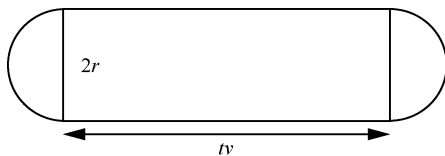


图 1 “Infected”节点在 t 时间的通信范围

从图 1 可以推算，一个“Infected”节点在 t 时间的通信面积为 $B = \pi r^2 + 2tvr$ ，则期望的传染度 $E(F)$ 可近似为

$$E(F) \approx B\lambda\eta = (\pi r^2 + 2tvr) \frac{\pi r^2}{TM^2} \quad (4)$$

其中， $\lambda = \frac{\pi r^2}{M^2}$ 为网络的平均密度， $\eta = \frac{1}{T}$ 为网络中节点的占空比， M 为正方形网络区域的边长。

定理 1 在一个具有 n 个节点的 MLDC-WSN 中，对于任意一个源节点发出的数据，算法 LT-MDS 结束时，接收到该数据的节点占网络中总节点的比率 $C \leq 1 - \frac{1}{E(F)}$ 。

证明 首先，推算当 IDD 分发数据完成时，网络中没有接收到数据的节点的比率为

$$q = \sum_{i=0}^n q^i f_i(i) \quad (5)$$

其中， $f_i(i)$ 为节点在网络中的分布。在随机方向模型中，节点的分布通常服从泊松点过程^[29]。因此，式(5)可以表示为

$$\begin{aligned} q &= \sum_{i=0}^n q^i \frac{(E(F))^i e^{-E(F)}}{i!} \\ &= e^{-E(F)} \sum_{i=0}^n \frac{(qE(F))^i}{i!} \\ &\leq e^{-E(F)} e^{qE(F)} \\ &\Rightarrow \ln q \leq (q-1)E(F) \\ &\Rightarrow q \geq \frac{1}{E(F)} \end{aligned} \quad (6)$$

因此，算法结束时，有 $C = 1 - q \leq 1 - \frac{1}{E(F)}$ 。

得证。

定理 2 在一个具有 n 个节点的 MLDC-WSN 中，对于任意一个源节点发出的数据，算法 LT-MDS 结束时，全部节点接收到这个数据的最小时延 $delay$

为 $\left[1 + \frac{\ln n}{\ln \left(1 + \frac{E(F)t}{T} \right)} \right]$ 。

证明 假设 N_k 为到第 k 个周期时，受到感染(接收到数据)的节点的总数； F_T 为一个周期 T 中，一个受到感染的节点可以继续感染新的节点的期望个数，则有 $N_k \leq N_{k-1} + N_{k-1}F_T = N_{k-1}(1 + F_T)$ 。由于 $N_0=1$ ，则 $N_k \leq (1 + F_T)^k$ 。因此，有

$$\begin{aligned} delay &\geq \arg \max_k (N_k \leq n) \\ &\geq \arg \max_k ((1 + F_T)^k \leq n) = \left\lceil 1 + \frac{\ln n}{\ln(1 + F_T)} \right\rceil \end{aligned}$$

由于 $F_T = E(F) \frac{t}{T}$ ，因此有

$$delay \geq \left[1 + \frac{\ln n}{\ln \left(1 + \frac{E(F)t}{T} \right)} \right] \quad (7)$$

得证。

定理 3 在一个具有 n 个节点的 MLDC-WSN 中,利用算法 LT-MDS 对 k 个数据进行分发及存储。如果节点的感染时间 t 充分大,则在数据收集阶段,数据采集者可以通过获得 $k+\varepsilon$ 个编码数据,以 $1-\delta$ 的概率恢复出全部的源数据。

证明 由算法 LT-MDS 可以看到,节点是根据 $\frac{d_w}{k}$ 的概率保存数据。算法结束时,节点保存的编码数据的实际代码度 d'_w 服从以下分布

$$\begin{aligned} \Pr(d'_w = i) &= \sum_{d_w=1}^k \binom{k}{i} \left(\frac{d_w}{k} \right)^i \left(1 - \frac{d_w}{k} \right)^{k-i} \Omega(d_w) \\ &= \sum_{d_w=1}^k \Pr(d'_w = i | d_w) \Omega(d_w) \end{aligned} \quad (8)$$

从定理 1 可以看到,当 t 充分大时,网络中接收到数据的节点的比率近似于 1,即几乎全部节点接收过源数据。因此, $d'_w \approx k \frac{d_w}{k} = d_w$, 由此可得

$$\Pr(d'_w = i) \approx \Omega(d_w) \quad (9)$$

即算法结束时,节点保存的编码数据的实际代码度非常接近于 RSD。根据 4.2 节 LTC 的性质,解码时只需要获得 $k+\varepsilon$ 个编码数据,就能够以 $1-\delta$ 的概率恢复出全部的源数据,可以很容易得到结论。得证。

定理 4 算法 LT-MDS 的时间复杂度为 $O(1)$, 消息复杂度为 $O(kn)$ 。

证明 由算法 1 的过程可以看到,在 LT-MDS 中没有出现循环语句,每一步骤的执行均为算术运算。因此,算法可以在 $O(1)$ 时间内完成。

在算法执行过程中,一个源数据的分发需要 $O(n)$ 个节点进行转发。每次转发时,接收节点需要发送一个短消息,而发送节点需要发送数据,一共需要 2 次广播。因此,一个源数据分发到全网,需要 $O(n)$ 次广播。由于网络中有 k 个源数据,因此,一共需要 $O(kn)$ 次广播。得证。

5 模拟实验

本文利用 Matlab 7 开发模拟实验平台。假设网络为一个正方形区域,大小为 $100 \text{ m} \times 100 \text{ m}$, 网络中

有 n 个随机分布的节点。节点的工作周期 $T=100 \text{ s}$, T 被划分为 100 个时间单元,每个时间单元为 1 s。节点只在其中一个随机选择的时间单元内醒来,而在其他时间单元则保持睡眠状态。节点唤醒时,采用随机方向模型进行运动,每次均从 $(0, 2\pi)$ 范围内随机选择一个方向,然后再以固定的速度 $v=1 \text{ m/s}$ 行走一个时间单元。节点的通信半径 $r=25 \text{ m}$ 。节点间的通信存在 MAC 层机制以保证可靠性,即如果 2 个节点间的距离小于 r ,则它们可以相互收到对方发送的数据分组。假设网络中有 $k=0.1n$ 个节点感知到了数据,即网络中有 k 个源节点。数据分组的大小为 1 000 bit,节点发送 1 bit 数据的单位能耗为 $E_t=100 \text{ nJ/bit}$,而接收 1 bit 数据的单位能耗为 $E_r=50 \text{ nJ/bit}$ 。选择目前最有代表性的分布式数据持续性方案 EDFC 和 APBDP 来和本文算法进行对比。

5.1 参数 t 的选择

根据第 4 节的分析,参数 t 的大小会影响数据分发过程中接收到某个数据分组的节点的数量。因此,本文分别在 $n=100$ 和 $n=500$ 的网络中,测试数据分发结束时,不同的 t 值对接收到数据的节点数量的影响,实验结果如图 2 所示。

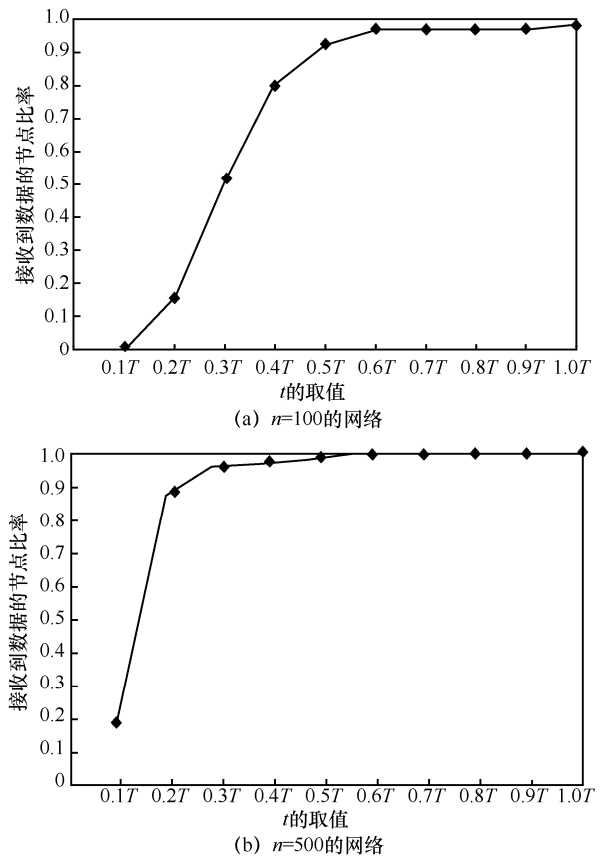
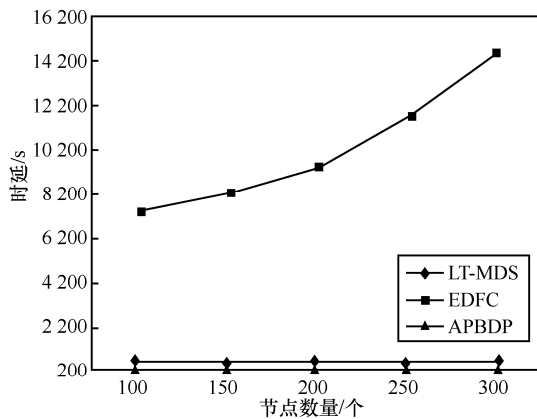


图 2 t 的取值对接收到数据分组的节点数量的影响

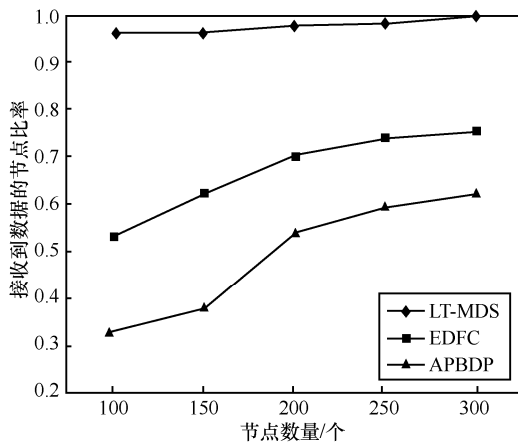
从图 2 可以看到,在 $n=100$ 的网络,需要 $t \geq 0.6T$ 才能使超过 90% 的节点均能接收到数据分组。而在 $n=500$ 的网络,只需要 $t \geq 0.3T$ 就能使超过 90% 的节点均能接收到数据分组。这是因为在节点密度小的网络中,一个受到感染的节点需要保持唤醒较长时间,才能遇到另一个未受到感染且唤醒的节点。而在节点密度大的网络,受感染节点与未受感染节点相遇的概率较大,更容易在短时间内传染其他节点。此外,虽然在部分网络中, $t \geq 0.6T$ 时仍然有少部分节点无法接收到数据,但是 5.3 节将通过实验证明 LT-MDS 仍然能取得近似最优的性能。

5.2 时延性能

本节将测试算法分发数据所需的时延,即从源节点开始分发数据,到网络中没有任何节点在广播数据时的时间间隔。对于 LT-MDS,选择 $t=0.6T$,这样既能获得较高的接收节点率(即分发过程结束时,网络中接收到数据的节点比率),又不用节点长时间保持唤醒而耗费大量能量。分别在 $n=100、150、200、250、300$ 的网络中测试算法的性能。实验结果如图 3 所示。



(a) 时延性能对比



(b) 接收节点率对比

图 3 LT-MDS 的时延性能

从图 3(a)可以看到, LT-MDS 的时延略高于 APBDP 的时延,而远远低于 EDFC 的时延。这是因为 LT-MDS 中节点有可能被多次感染而延长数据分发的广播过程。APBDP 采用概率广播来分发数据,只有少部分节点需要转发数据,所以数据分发会较早结束。EDFC 采用多个数据副本以随机行走的方式进行分发,需要经过较大的额定步长才能完成全部过程,所以时延最大。另一方面,从图 3(b)可以看到,虽然 APBDP 的时延最小,但是接收到数据的节点比率是最低的,而 LT-MDS 拥有最高的接收节点率。因此,综合而言,LT-MDS 分发数据时具有较低的时延和较高的接收节点率。

5.3 数据持续性

为了测试算法的数据持续性(即算法运行完后,如果网络中部分节点死亡,则移动 Sink 访问部分仍然存活的节点后,全部源数据仍然能被恢复出来的概率),定义以下指标。

定义 2 访问率是指被移动 Sink 访问并采集编码数据的节点与源节点的数量比值。

定义 3 成功解码率是指 k 个源数据能够全部被恢复出来的概率。在一定的访问率下,如果执行算法 N_a 次,统计出 k 个源数据被全部恢复的次数为 N_s ,则成功解码率为 $\frac{N_s}{N_a}$ 。

本文实验的结果是执行 20 次以后的平均值,即 $N_a=20$ 。对比的结果如图 4 所示。

从图 4 可以看到,LT-MDS 获得的数据持续性要高于 EDFC 和 APBDP。这是因为 LT-MDS 有效地完成了数据分发和编码存储,使最后移动 Sink 可以访问数量较少的节点就能恢复出全部的源数据。而 EDFC 和 APBDP 的节点接收率较低,使节点中保存的编码数据存在信息缺失,因此,成功解码率较低。

6 结束语

本文对 MLDC-WSN 中如何有效提高数据持续性的问题进行研究。MLDC-WSN 具有节点占空比低、网络无法保持持续连通等特点,传统的数据分发及存储算法很难应用。为此,提出了一种分布式算法 LT-MDS 来解决以上问题。首先,综合考虑 MLDC-WSN 网络的特点,采用一种新的传染病式数据分发方法来分发数据。该方法不需要任何全局信息的支持,节点只需根据接收到的邻居信息来决

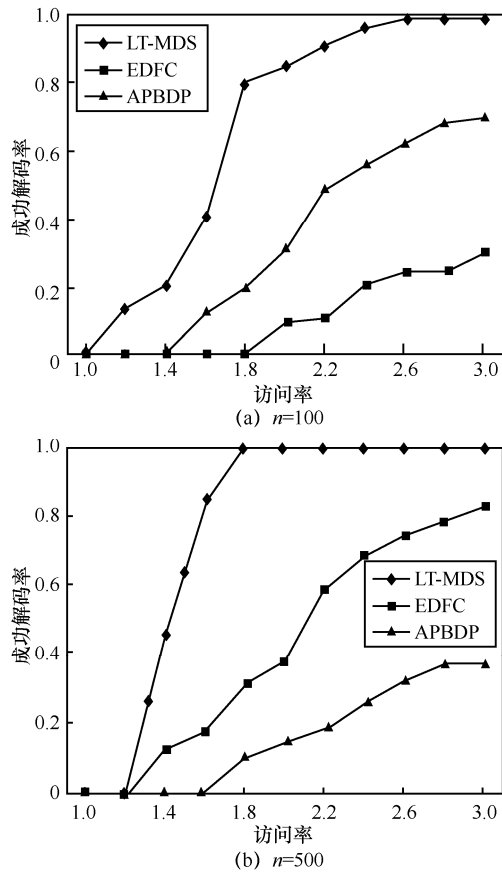


图4 数据持续性

定是否转发数据,适用于节点不断移动的动态网络。此外,该方法可以使数据以较低的时延被网络中绝大部分节点接收到,具有较高的可靠性。然后,基于该数据分发方法,使节点在接收到数据的同时,利用LTC对数据进行编码存储。理论分析和仿真实验均表明,LT-MDS能够以低时延完成数据分发和存储,同时具有较高的数据持续性。

下一步将结合更实际的应用场景,考虑动态性更强的网络条件,例如,链路存在不可靠性和严重冲突、网络的节点受到外力可能突然大面积死亡等情况,研究具有更高的可靠性、更强的网络抗毁能力并且同时具有更低时延和能耗的分布式数据分发方法。此外,还将研究满足不同应用需求且具有低编/解码复杂度的数据编码保存方法,可以根据不同的优先级或时—空关系对数据进行分类存储,使重要的数据能够获得更高的数据持续性。

参考文献:

- [1] 刘伟, 刘军. 时延敏感传感器网络中分布式动态资源管理研究[J]. 通信学报, 2017, 38(7): 70-77.
LIU W, LIU J. Study on distributed and dynamic resource management for delay-sensitive sensor network[J]. Journal on Communications, 2017, 38(7):70-77.
- [2] CHEN L Y, GU Y, GUO S, et al. Group-based discovery in low-duty-cycle mobile sensor networks[C]//2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON). 2012: 542-550.
- [3] LUBY M. LT codes[C]//The 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002). 2002: 271-280.
- [4] ACEDANSKI S, DEB S, MEDARD M, et al. How good is random linear coding based distributed networked storage[C]//First Workshop on Network Coding, Theory, and Applications (NetCod 2005). 2005: 1-6.
- [5] KAMRA A, MISRA V, FELDMAN J, et al. Growth codes: maximizing sensor network data persistence[C]//The 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2006). 2006: 1-12.
- [6] LIN Y F, LI B C, LIANG B. Differentiated data persistence with priority random linear codes[C]//27th International Conference on Distributed Computing Systems (ICDCS 2007). 2007: 47-54.
- [7] LIN Y F, LIANG B, LI B C. Geometric random linear codes in sensor networks[C]//2008 IEEE International Conference on Communications (ICC 2008). 2008: 2298-2303.
- [8] MATSUZONO K, ROCA V, ASAEDA H. Structured random linear codes (SRLC): bridging the gap between block and convolutional codes[C]//2014 IEEE Global Communications Conference (Globecom 2014). 2014: 1211-1217.
- [9] AL-AWAMI L, HASSANEIN H. Distributed data storage systems for data survivability in wireless sensor networks using decentralized erasure codes[J]. Computer Networks, 2016, 97(3): 113-127.
- [10] TALARI A, RAHNAVARD N. CStorage: decentralized compressive data storage in wireless sensor networks[J]. Ad Hoc Networks, 2016, 37(2): 475-485.
- [11] LIN Y F, LIANG B, LI B. Data persistence in large-scale sensor networks with decentralized fountain codes[C]//26th IEEE International Conference on Computer Communications (INFOCOM 2007). 2007: 1658-1666.
- [12] KONG Z, ALY S, 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.
- [13] LIU K K, EL-KHAMY M, LEE J. Finite-length algebraic spatially-coupled quasi-cyclic LDPC codes[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(2): 329-344.
- [14] AL-AWAMI L, HASSANEIN H. Robust decentralized data storage and retrieval for wireless networks[J]. Computer Networks, 2017, 128(9): 41-50.
- [15] KONB B, ZHANG G, ZHANG W, et al. Data persistence in planetary surface network using raptor codes and probabilistic broadcasting[J]. International Journal of Distributed Sensor Networks, 2016, 12(10): 1-13.
- [16] AZIMI N H, GUPTA H, HOU X X, et al. Data preservation under spatial failures in sensor networks[C]//The Eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2010). 2010: 171-180.
- [17] XU M, SONG W Z, HEO D, et al. ECPC: preserve downtime data persistence in disruptive sensor networks[C]//2013 IEEE 10th International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2013). 2013: 281-289.

- [18] 梁俊斌, 李陶深. 无线传感网中基于自适应概率广播的数据保存[J]. 计算机研究与发展, 2012, 49(10): 2229-2240.
LIANG J B, LI T S. An adaptive probability broadcast-based data preservation in wireless sensor networks[J]. Journal of Computer Research and Development, 2012, 49(10): 2229-2240.
- [19] KONB B, ZHANG G X, ZHANG W, et al. Efficient distributed storage for space information network based on fountain codes and probabilistic broadcasting[J]. KSII Transactions on Internet and Information Systems, 2016, 10(6): 2606-2626.
- [20] RUIZ P, BOUVRY P. Survey on broadcast algorithms for mobile ad hoc networks[J]. ACM Computing Surveys, 2015, 48(1): 1-35.
- [21] CHEN L Y, SHU Y C, GU Y, et al. Group-based neighbor discovery in low-duty-cycle mobile sensor networks[J]. IEEE Transactions on Mobile Computing, 2016, 15(8): 1996-2009.
- [22] CHEN P P, CHEN Y, GAO S, et al. Efficient group-based discovery for wireless sensor networks[J]. International Journal of Distributed Sensor Networks, 2017, 13(7): 1-12.
- [23] CHAKCHOUK N. A survey on opportunistic routing in wireless communication networks[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2214-2241.
- [24] SO J M, BYUN H J. Load-balanced opportunistic routing for duty-cycled wireless sensor networks[J]. IEEE Transactions on Mobile Computing, 2017, 16(7): 1940-1955.
- [25] CAMP T, BOLENG J, DAVIES V. A survey of mobility models for ad hoc network research[J]. Wireless Communications and Mobile Computing, 2002, 2(5): 483-502.
- [26] WU Y, MAO Z J, FAHMY S, et al. Constructing maximum-lifetime data-gathering forests in sensor networks[J]. IEEE/ACM Transactions on Networking, 2010, 18(5): 1571-1584.
- [27] KUI X Y, ZHANG S G, WANG J X, et al. An energy-balanced clustering protocol based on dominating set for data gathering in wireless sensor networks[C]//2012 IEEE International Conference on Communications (ICC).2012: 193-197.
- [28] MIAO G W, AZARI A, HWANG T W. E2-MAC: energy efficient medium access for massive M2M communications[J]. IEEE Transactions on Communications, 2016, 64(11): 4720-4735.
- [29] GROENEVELT R, NAIN P, KOOLE G. The message delay in mobile ad hoc networks[J]. Performance Evaluation, 2005, 62(1-4): 210-228.

[作者简介]



蒋婵 (1980-), 女, 广西合浦人, 华南理工大学博士生, 主要研究方向为无线传感器网络。



李陶深 (1957-), 男, 广西南宁人, 博士, 广西大学教授, 主要研究方向为分布式系统、无线网络。



梁俊斌 (1979-), 男, 广西南宁人, 博士, 广西大学教授, 主要研究方向为无线传感器网络。