

无线传感器网络中基于树的能量高效分布式 精确数据收集算法

陈 零,王建新,张士庚,奎晓燕

(中南大学信息科学与工程学院,湖南长沙 410083)

摘 要: 在大规模节点密集的多跳传感器网络中,精确数据收集存在着“热区”问题:越靠近 Sink 节点的传感器节点,其承担的数据转发量就越多,能量消耗也越快,从而成为瓶颈节点,缩短整个网络的生命周期.最大生命周期数据收集树的构建已被证明是 NP 完全问题.已有算法大多是集中式算法,不适用于大规模节点密集的传感器网络.本文提出一种分布式精确数据收集算法 EEDAT,在大规模节点密集的传感器网络中,不仅能够保证每个节点到 Sink 的路径是最短路径(最少跳数),而且能有效延长网络生命周期.EEDAT 分为两个基本步骤,首先随机生成一棵数据收集树,然后根据各个传感器节点的孩子数和剩余能量,对已生成的数据收集树进行调整,使得各个节点的负载尽量均衡,从而达到延长网络生命周期的目的.实验结果表明,与已有分布式算法 LMST 相比,EEDAT 所构造的数据收集树能延长网络生命周期平均 20%.

关键词: 无线传感器网络;数据收集;数据收集生成树

中图分类号: TP301

文献标识码: A

文章编号: 0372-2112 (2013) 09-1738-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2013.09.011

A Distributed Tree-Based Energy-Efficient Algorithm for Precise Data Gathering in Wireless Sensor Networks

CHEN Ling, WANG Jian-xin, ZHANG Shi-geng, KUI Xiao-yan

(School of Information Science and Engineering, Central South University, Changsha Hunan 410083, China)

Abstract: In this paper, we propose a distributed tree-based algorithm for precise data gathering in wireless sensor networks called EEDAT, which achieves longer network lifetime than existing distributed algorithms. The EEDAT algorithm has two steps. In the first step, a shortest path tree is constructed in a distributed manner. In the second step, EEDAT adjusts the load of nodes in the generated tree to balance energy consumption of different nodes, which effectively extends the lifetime of the network. In the adjustment, both the number of children of a node and its residual energy are considered. Simulation results show that EEDAT achieves longer lifetime than the LMST algorithm. On average, compared with LMST, EEDAT prolongs the lifetime by 20%.

Key words: wireless sensor networks; data gathering; spanning tree

1 引言

无线传感器网络(wireless sensor networks)是近年来的研究热点.其主要应用是对一定区域内的特定信息进行监测和采集.被监测的区域一般为人类难以进入和靠近的危险地带.传感器节点被部署后,其能量无法再度补充,当自身电池耗尽,也就意味着节点的死亡.因此,如何尽可能地减少各节点的能量消耗,延长网络的生命周期,成为传感器网络协议设计的重要目标.现有研究指出,通信开销是传感器节点能量消耗的主要形式,而

通信开销又主要产生于对感知数据的传输过程中.所以,设计能量高效的数据收集协议,对于延长无线传感器网的生存周期具有重要意义^[1].

数据收集(data gathering)是指各个传感器周期性地感知定量数据,然后通过多跳方式将其传送到 Sink 节点的过程.它分为两种模式^[1]:

(1)相关数据收集(correlated data gathering).在有些场景中,感知数据之间具有一定的相关性,节点可以将收到的数据和自己感知的数据进行聚合,再以固定长度的数据包发送出去.这种模式下,无论节点接收多少数

据包,它发送出的数据大小是相同的,即数据发送独立于数据接收。

(2)精确数据收集(precise data gathering).与上述相反,感知数据的相关性很弱,节点接收到的数据包长度不一,只能与自身感知数据进行低比例聚合或无法聚合.所以,每个节点发送数据包的长度,与接收数据以及自身感知数据呈线性关系.而且,在精确收集的过程中,越靠近 Sink 的中继节点,所承担的数据转发量就越大,能量消耗也就越快,易产生“热区”问题^[2],从而缩短网络整体生命周期。

在数据收集过程中,为了最大化无线传感器网络的生命周期,需要构造生命周期最大的生成树.而已发表文献中,多为集中式生成算法,其在某种程度上已很接近理论最优上界.但集中式算法不适用于大规模节点密集的无线传感器网络,特别是在网络动态性较高的情况下,利用其实时重构路由树的开销(主要表现为通信开销)很大,从而影响算法的整体性能.而要避免集中式算法的缺点,比较好的思路就是分布式.本文提出一种适用于精确数据收集的能量高效分布式算法 EEDAT(Energy-Efficient precise data gathering Distributed Algorithm base on Tree).EEDAT 无需节点位置信息,只需每个节点获取初始树中自身的孩子数,再根据此信息对初始树进行调整优化.仿真实验表明,与比较典型的两种分布式算法相比,EEDAT 能构造生命周期更长的数据收集树.此外,该算法还能有效降低感知数据到达 Sink 节点的延时。

2 相关工作

目前,已有很多文章对数据收集算法进行了研究,主要有以下三类:

(1)基于簇的数据收集协议(cluster-based data gathering protocol).其基本思想是:将网络中的节点划分到不同的簇中,每个簇选出一个节点作为簇首.数据收集时,各节点将感知到的数据传输给各自的簇首,然后簇首再将数据直接或以多跳的方式转发给 Sink 节点.比较典型的簇收集协议有 LEACH^[3]等.簇协议易于实现和管理,但存在簇首分布不均、簇首负载过重、簇规模难以控制等缺点。

(2)基于链的数据收集协议(chain-based data gathering protocol).它将网络中的所有节点串在一条链上,然后再按照某种参量,从链中选取一个节点作为头节点,其它节点感知数据后,会将数据沿链传输给头节点,最后头节点再直接转发给 Sink 节点.典型的链协议有 PE-GASIS^[4]等.链协议可以避免簇协议中簇重构的算法开销,但其最明显的缺点是,唯一的头节点往往会成为最严重的瓶颈.而且其通信延时往往较大。

(3)基于树的数据收集协议(tree-based data gathering protocol).此协议是以 Sink 为根节点,在网络中生成一棵连通树,每个节点在感知数据后,将数据传输给各自的父节点.典型的树协议有 MNL^[5]、MILD^[6]、MAXLAT^[7]等.由图论可知,树是支持网络连通性的最小图结构^[8],因此,基于树的数据收集协议在保证网络连通性和可靠性方面具有天然的优势,而且还具有保证 QoS、便于实现高效的能量管理等优点,是目前研究的热点。

已有树收集协议大部分为集中式的.其中,MAXLAT 协议在精确收集情形下,算法性能已接近理论最优上界.其基本思想是:以各节点的子孙数作为负载参量,将节点分为瓶颈、次瓶颈和富裕三种类型.然后通过着色算法,将瓶颈节点的子孙转移到富裕节点的子树上去,最后会生成一棵近似最优的数据收集生成树。

但就如引言中所述,集中式算法协议通信开销过大,不适用于大规模密集型动态网络.分布式算法的特点是只需局部信息(通常为一跳信息)来进行路由计算,因此能很好地应用于大规模密集型动态网络.当然,分布式算法的难点也在于如何利用局部信息来构造有效路由,同时尽量减少各节点的通信开销.目前已经提出一些基于树的分布式数据收集算法,比较典型的是随机路由^[9]和近期提出的 LMST^[10]算法。

随机路由的基本思想是,由 Sink 节点广播建树消息,各节点在接收到建树消息后,随机建立数据收集生成树.只要网络是连通的,在建树消息传播一段时间后,一棵数据收集树便建立起来.随机路由中数据收集树可以逐层构建,能保证各节点到 Sink 的路径最短(即跳数最少)^[8].但是随机路由没有对建树过程进行控制,构建的数据收集树均衡性差,很容易产生“热区”问题,从而缩短网络的整体生命周期。

LMST(Local Minimum Spanning Tree)算法^[9]是一个基于随机路由的分布式改进算法协议,它利用节点间距离和通信能耗之间的关系,对数据收集树进行调整,以达到减少能量开销,进而延长网络生命周期的目的.此算法在建立局部最小生成树时,节点间至少需要两次的消息交换,其后还要进行多次控制交互,所以其通信开销比较大.再者,由于需要获取各节点间链路能耗,因此必须知道各节点的坐标信息,这必然也会增大通信开销。

本文提出的 EEDAT 算法以随机路由构建的数据收集树为基础,根据各个节点的负载进行调整,以此来改进数据收集树的均衡度,从而达到均衡各个节点的能量开销,延长网络生命周期的目的.而且,EEDAT 无需节点位置信息,通信开销远小于 LMST,适用于精确数据收集。

3 网络模型和问题描述

3.1 网络模型

在一个面积为 $M \times M$ 的正方形区域 A 内随机布撒 n 个传感器节点, 加上一个 Sink 节点, 这样形成的传感器网络可以用一个连通的无向图 $G(V, E)$ 来表示, 其中: V 是节点集合, $V = \{v_0, v_1, v_2, \dots, v_n\}$, $|V| = n + 1$, v_0 是 Sink 节点, v_1, v_2, \dots, v_n 是传感器节点; E 是 G 中边的集合. 假如节点 v_i 和 v_j 都在对方的通信半径之内, 则边 $(v_i, v_j) \in E$, $|E| = m$ 为边的数量. 我们假设网络具备如下性质:

(1) 整体网络是连通的, 节点在部署后不再移动.

(2) 假设中继节点不对数据进行聚合. 那么, 一棵收集树 T 上的节点, 会周期性地将其自身感知到的 l bit 数据连同孩子节点发过来的数据, 全部发送给父节点.

(3) 节点无法进行能源补充, 且初始能量可以互不相同.

(4) 节点之间利用已有的同步算法进行了粗粒度的同步. 目前, 很多同步算法已经能够提供毫秒级精度的同步^[11].

不失一般性, 假设节点在发射和接收数据时的功率固定, 用 E_t 表示发射 1bit 数据需要的能量, E_r 表示接收 1bit 数据需要的能量. 同时假定网络拥有较好的拥塞控制策略, 可避免数据在传输过程中发生拥塞和重传.

3.2 相关定义

参考已有工作^[7], 给出以下定义.

定义 1 轮: 是指所有传感器节点收集一次数据, 并将数据传送到 Sink 节点所要耗费的时间, 不管此时间要持续多久.

定义 2 节点生命周期: 是指节点 v_i 在一棵收集树中存活的轮数. 用式子 $E(v_i) > e(E(v_i))$ 指节点剩余能量, e 是一设定的阈值, 当节点能量小于 e 时, 将无法完成一轮的数据收集任务来判断节点是否存活. 则节点生命周期可用下式来计算:

$$L_{\text{node}}(T, v_i) = \left\lfloor \frac{E(v_i)}{lS(T, v_i)(E_r + E_t) + lE_t} \right\rfloor, i = 1, \dots, n \quad (1)$$

上式中 l 表示节点每轮感知数据的平均长度(单位为 bit), $S(T, v_i)$ 表示节点 v_i 在树 T 中, 子孙的数量.

定义 3 树的生命周期: 在数据收集过程中, 第一个死亡的节点所经历的轮数, 就是树 T 的生命周期. 可用下式表示:

$$L_{\text{tree}}(T) = \min_{i=1, \dots, n} \{L_{\text{node}}(T, v_i)\} \quad (2)$$

树的生命周期可等同为网络的生命周期.

定义 4 最优树: 就是所有数据收集生成树中, 生命周期最长的那棵树. 用下式表示为:

$$T_o = \{T \mid L_{\text{tree}}(T) = \max_{T^* \in TS(G)} L_{\text{tree}}(T^*)\} \quad (3)$$

上式中, T^* 是指网络 G 中任意一棵生成树, 而 $TS(G)$ 是 G 中生成树的集合.

定义 5 瓶颈节点: 是指收集树 T 中最早耗尽能量的节点, 其生命周期就相当于树的生命周期.

3.3 问题描述

无线传感器的实际应用中, 人们总是希望网络的工作时间越长越好. 因此, 可以把问题归结为在多个数据收集生成树中, 寻找生命周期最长的生成树. 可以表示为下式:

$$\max L_{\text{tree}}(T) \quad (4)$$

根据定义 2 和 3, 公式(4)可表示为:

$$\max L_{\text{tree}}(T) = \max \min_{i=1, \dots, n} \left\{ \left\lfloor \frac{E(v_i)}{lS(T, v_i)(E_r + E_t) + lE_t} \right\rfloor \right\} \quad (5)$$

上式中, E_r 和 E_t 为常量, 只有 $S(T, v_i)$ 为变量, 是主要的优化目标. 为了简化表达, 将 $l(E_r + E_t)$ 从分母中分离, 从而可将公式(5)简化为:

$$\max L_{\text{tree}}(T) \Leftrightarrow \max \min_{i=1, \dots, n} \frac{E(v_i)}{S(T, v_i) + c} \quad (6)$$

其中, $c = E_t / (E_r + E_t)$. 这很明显是一个负载均衡问题, 能量 $(E(v_i))$ 越大的节点, 应承担的子孙 $(S(T, v_i))$ 就越多; 反之则越少. 文献[5]已证明这个问题是 NP 完全的. 文献[7]提出一种 MAXLAT 算法, 能在有限时间内生成近似最优的收集树. 但它是一种集中式算法, 如前所述, 实际应用时通信开销过大. 这里以文献[7]的一些结论为基础, 提出一种通信开销很小的分布式算法 EEDAT.

4 EEDAT 的设计和分析

分布式算法利用局部信息延长网络生存周期的同时, 要尽量减少通信开销, 所以要使用合适的网络参数和控制方法来进行有效平衡.

4.1 EEDAT 的描述

经研究发现, 精确收集时, 如果各负载节点的子孙数越均衡, 则生成树的生命周期越长. 所以, 如何使得各负载节点的子孙数尽量均衡, 是延长网络生命周期的关键所在.

图 1 是一棵随机路由生成树, A 为 Sink 节点, B 为瓶颈(负载)节点, 因为其所拥有的子孙节点最多. 根据定义 2 和定义 3 可知, 图 2 生成树的生命周期两倍于图 1. 可以看出, 优化的关键是要将瓶颈节点的子孙节点,

合理调整到非瓶颈节点上去。

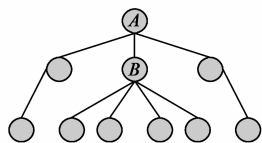


图1 随机路由生成树

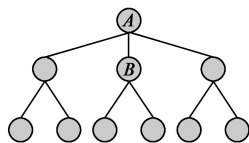


图2 最优生成树

由图论和实验得知,节点孩子数与其子孙数有直接关系,把它作为重要的路由控制参数,可以有效均衡数据收集树。有了合适的网络参数,就要考虑如何减少通信开销。经过研究,提出一种重新分配时间优先级的方法。其基本思想是:在随机路由后,以孩子数作为评判标准,为每个节点重新分配选取孩子的时间片,孩子数越少的节点,会获得较优先的时间片。而孩子数在随机路由的过程中就能获知,节点间无需为此参数进行额外的交互,这能有效的减少通信开销。

EEDAT算法的基本步骤如下:

(1)用“Hello Message”消息构建每个节点的邻居表。

(2)利用随机路由生成初始收集树,此过程中计算每个节点的孩子数并存储。

(3)用公式 $T_{\text{Reset}} = T_e + \frac{C_n}{E_n}t'$ (其中, T_e 为一时间常量, C_n 为节点孩子数, E_n 为节点剩余能量, t' 为一时间常量)计算每个节点的树调整包发送时间。

(4)每个节点根据计算好的 T_{Reset} 发送调整包,对初始树进行调整。此过程中,只有节点的孩子邻居节点会响应调整包,如此可以降低数据包传送延时。

(5)树调整后,经过 n 轮数据收集,重启整个算法过程。

此算法过程中,在随机路由后,每个节点只发送一个调整包,且包里只含基本标志信息,节点间无需做多次交互,因此,节点间通信开销小。对于时间常量 T_e 和 t' 要特别说明一下,前者要远远大于后者, T_e 是进行树调整的时间始点,单位为秒级。 t' 是每个节点的调整时间片单位,为毫秒级。

4.2 EEDAT 算法分析

对于 EEDAT 的时间开销,把它分为三个阶段进行分析。一是邻居信息收集阶段,设节点数为 n ,此阶段所花时间一般会预设为一常数 T_N ,大小与 n 成线性正比关系 $O(n)$,单位为秒级;二是随机路由阶段,所花时间 T_S 与 n 成线性正比关系 $O(n)$,单位也为秒级;三是树调整阶段,其预设时间 T_R 与 n 也成线性正比关系 $O(n)$,单位同样为秒级。所以,整个算法的时间开销为 $O(n)$ 秒。

从通信开销分析,EEDAT 在整个算法过程中,无须传送和接收局部网络拓扑信息,只需要基本的标志信

息即可,所以其通信上的耗费是相当低的。而文献[7]中的 MAXLAT 算法,须将整个网络的拓扑信息传送至 Sink 节点来满足算法的运行条件。假设 a_i 为节点的度数,节点地址占用 1bit,MAXLAT 最优树的长度则为 n bit,那么对于每个节点 i ,在一轮算法过程中,MAXLAT 比 EEDAT 多出的通信开销可表示为:

$$E_r(\sum_{j \in D_i} a_j + n) + E_l(a_i + \sum_{j \in D_i} a_j + c \times n) \quad (7)$$

上式中 c 为一常数,当节点为非叶节点时为 1,否则为 0。 D_i 是经由节点 i 转发消息的节点的集合。

对于文献[10]中提出的分布式收集算法 LMST,其需要在节点间传送和接收一跳内的网络拓扑信息。同样假设 a_i 为每个节点的度数,节点地址占用 1bit,在局部最优树中的每个节点度数为 b_i ,那么相对于 EEDAT,LMST 在一轮算法过程中,每个节点多出的通信开销为:

$$(a_i + b_i)E_l + (\sum_{i \in B_i} a_i + \sum_{i \in B_i} b_i)E_r \quad (8)$$

上式中 b_i 表示节点 i 的邻居节点的集合。

在一般的实际应用场景中,节点的平均度数为 6~15,所以在网络动态性较高的情况下,MAXLAT 和 LMST 的通信开销会远大于 EEDAT。

4.3 EEDAT 与 LMST 的对比分析

在精确收集情形下,EEDAT 的网络生存性能要优于 LMST 算法。这其中的原因,用两个图来进行对比说明。

图 3 和图 4 分别为 EEDAT 和 LMST 算法生成树,它们的瓶颈节点都是 B ,根据定义 2 与定义 3 可知,EEDAT 算法生成树的生命周期是 LMST 算法生成树的 1.7 倍。

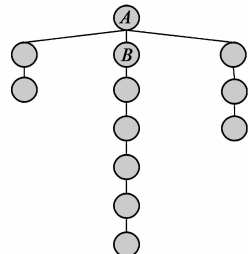
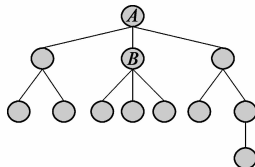


图3 EEDAT生成的数据收集树 图4 LMST生成的数据收集树

造成这种结果的原因是,LMST 算法是专为相关数据收集而设计的,它的侧重点是尽量减少每个节点的度数(如图所示,图 4 生成树点的最高度数为 1,而图 3 中为 3。),它没有考虑各负载节点间的子树平衡问题。而 EEDAT 算法,利用孩子数这一参量,对负载节点的子树进行有效的均衡控制,使得每个负载节点尽量公平的承担传输任务。

5 模拟实验

模拟实验是用 NS2 来实现的,其版本为 2.33,实验场景的基本设置和可变参数如下所示:

场景面积: $100\text{m} \times 100\text{m}$; 节点数量: 61, 81, 101, 161, 181, 201 (包括 Sink 节点在内); 节点最大通信距离: 20m.

节点的数据产生率为 128 bits/round. 为了使所有算法能进行公平的对比, 节点均采用固定发射功率, 发送能耗大约是接收能耗的 2 倍, 即 $E_t = 2E_r$, $E_r = 50\text{nJ/bit}$. 因为路由构造和网络调整时的控制通信开销可以独立分析, 所以模拟实验时, 只关注数据收集时的通信能耗和树的生命周期. 此外, 节点不对感知数据进行汇聚, 是精确收集的情形. 实验结果是算法执行 30 次后的平均值.

5.1 各算法在 Sink 节点位置变化时的对比

为了考察算法的有效性, 将 EEDAT 与单纯的随机路由以及 LMST 算法进行对比, 并以 Sink 节点位置和节点数作为变量, 设置了四个实验场景:

(1) Sink 节点在区域中心, 坐标为 (50, 50), 除 Sink 节点外有 100 个节点随机分布在区域内. (图 5(a))

(2) Sink 节点在区域边角, 坐标为 (1, 1), 除 Sink 节点外有 100 个节点随机分布在区域内. (图 5(b))

(3) Sink 节点在区域中心, 坐标为 (50, 50), 除 Sink 节点外有 200 个节点随机分布在区域内. (图 5(c))

(4) Sink 节点在区域边角, 坐标为 (1, 1), 除 Sink 节点外有 200 个节点随机分布在区域内. (图 5(d))

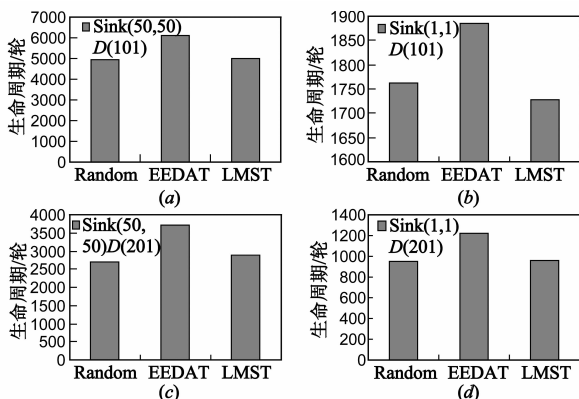


图5 各场景下各算法生命周期柱状对比

图 5 中是不同算法间, 树生命周期的对比. 纵坐标表示树的生命周期, 单位为轮数. 如图 5(a) 所示, 场景 (1) 中 EEDAT 的树生命周期为 6100 轮, 随机路由为 4930 轮, LMST 为 4997 轮. 随机路由没有任何网络参数的约束, 所以网络生存周期最差. 而 LMST 重点关注于节点平均度数, 不能有效均衡瓶颈节点的子孙数, 所以其生存轮数比 EEDAT 要差. 总体而言, EEDAT 比其它两种算法, 在网络生命周期上要提高 20% 以上.

如图 (图 5(d)) 所示, 在场景 (4) 中, 所有算法的生命周期树都有显著下降, 这是因为 Sink 节点处于区域边缘, 导致能直接跟 Sink 通信的节点减少, 从而使各算法生成树上的子树的数量减少, 子树上的节点数量增多, 使得瓶颈节点的负载进一步加重, 最终造成网络生

命周期的下降. 在此种情况下, EEDAT 仍然比其它算法的生命周期性能要高 20% 以上.

5.2 节点密度变化时的算法对比

为了考察算法的扩展性, 以节点密度和 Sink 节点位置作为变量, 设置不同场景进行模拟实验. 密度变化时, 保持区域面积 $100\text{m} \times 100\text{m}$ 不变, 传感器节点数从 60 递增至 200 为止. Sink 节点位置的变化与上述相同, 即中间和边缘.

图 6 是 Sink 节点在区域中央时的密度变化生命周期对比曲线图, 三种算法的网络生命周期, 随着节点密度的增加而逐步下降, 这是因为密度的增加会使感知数据随之上升, 各节点的通信能耗也会随之上升, 从而导致生命周期的缩短. 但不管在何种密度下, EEDAT 算法的生命周期性能都要优于其它两种, 且随着节点密度的增加, 其相对的性能提升会稳定在 25% 左右. 在低密度时, 跟 Sink 直接相邻的节点较少, 使得 EEDAT 的均衡作用难以充分体现, 所以三种算法的性能差异相对不明显. 而随着密度的增加, Sink 的邻居节点数也逐步增多, EEDAT 相较于其它两种算法, 能更好的均衡负载节点的子孙数, 从而使得网络的生命周期更长.

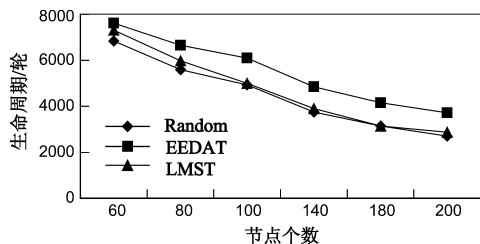


图6 Sink节点处于中间

图 7 是 Sink 节点在区域边缘时的密度变化生命周期对比曲线图, 与图 6 相比, 各算法的网络生命周期有显著下降, 原因在 5.1 节中已经说明. 此外, 与图 6 相比, 图 7 的曲线较为不规律, 这是因为 Sink 节点处于边缘位置, 用来均衡负载的邻居节点比较少, 从而影响了算法的最终效果. 但就算在此极端情况下, EEDAT 相对于其它两种算法, 平均性能提升也有 12%.

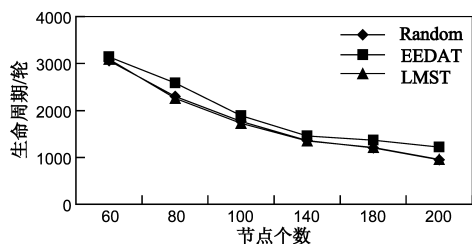


图7 Sink节点处于边缘位置

5.3 各算法网络时延对比

在基于树的数据收集协议中, 算法的网络传输延

迟可以通过生成树的平均高度来表征.图 8 是三种算法的生成树平均高度对比柱状图,纵坐标为树的平均高度,如图所示,随机路由和 EEDAT 的树平均高度为 5.4,而 LMST 为 13.5,是前两者的 2.5 倍.因此,算法 EEDAT 的延迟要远小于 LMST.

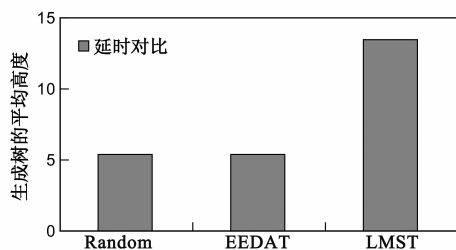


图8 延时对比柱状图

上述实验结果表明,EEDAT在各种情况下,其网络生存性能都优于其它两种算法,改进百分比略高于20%,且网络时延较低.此外,EEDAT特别适用于 Sink 节点设置在区域中心,并密集部署大量传感器节点的应用场景.

6 总结

本文针对无线传感器网的精确收集问题,研究如何分布式的构建能量高效的数据收集树.已有的基于树的集中式数据收集算法(如 MAXLAT),虽然可以获得近似最优的均衡树,但都普遍存在难以实现和通信代价过高的问题.我们利用节点孩子数和其剩余能量作为衡量标准,提出一种分布式的,根据局部信息计算优先级,重新分配调整时间片的算法(EEDAT)来构造能量高效的数据收集树.模拟实验表明,EEDAT 算法能有效提高无线传感器网络的生存周期,而且特别适用于网络节点密度较高, Sink 节点部署在区域中央的应用场景.

然而,EEDAT 算法在某些特定网络场景下,性能提升相对不明显,这其中的深层机理还有待进一步研究,以求对算法做出进一步的改进.

参考文献

- [1] Cristescu R, Beferull-Lozano B, Vetterli M. On network correlated data gathering[A]. Proc of the 23rd IEEE Conf on Computer Communications[C]. Piscataway: IEEE Press, 2004. 2571 - 2582.
- [2] Liu JN, Adler M, Towsley D. On optimal communication cost for gathering correlated data through wireless sensor networks [A]. Proc of the 12th Annual Int'l Conf on Mobile Computing and Networking[C]. New York: ACM Press, 2006. 310 - 321.
- [3] Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless micro-sensor networks [A]. Proc of the 33rd Hawaii Int'l Conf on System Sciences [C]. Washington: IEEE Computer Society, 2000. 3005 - 3014.
- [4] Lindsey S, Raghavendra CS. PEGASIS: Power efficient gathering in sensor information systems [A]. Proc of the IEEE Aerospace Conf[C]. San Francisco: IEEE Computer Society, 2002. 1 - 6.
- [5] Liang WF, Liu YZ. Online data gathering for maximizing network lifetime in sensor networks [J]. IEEE Trans on Mobile Computing, 2007, 6(1): 2 - 11.
- [6] 梁俊斌, 王建新, 陈建二. 在传感器网络中构造延迟限定的最大化生命周期树[J]. 电子学报, 2010, 38(2): 345 - 351.
Liang Jun-bin, Wang Jian-xin, Chen Jian-er. On the construction of a delay-constrained maximum lifetime tree in wireless sensor networks [J]. Acta Electronic Sinica, 2010, 38(2): 345 - 351. (in Chinese)
- [7] 梁俊斌, 王建新, 李陶深, 陈建二. 传感器网络中基于树的最大生命精确数据收集 [J]. 软件学报, 2010, 21(9): 2289 - 2303.
- [8] Khan M, Pandurangan G, Vullikanti A. Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks [J]. IEEE Trans on Parallel and Distributed Systems, 2009, 20(1): 124 - 139.
- [9] Kook-Hee Han, Young-Bae Ko, Jai-Hoon Kim. A novel gradient approach for efficient data dissemination in wireless sensor networks [A]. VTC2004-Fall, 2004 IEEE 60th [C]. Los Angeles: IEEE Transportation, 2005. 2979 - 2983.
- [10] Tan, H O, Korpeoglu I, Stojmenovic I. Computing localized power-efficient data aggregation trees for sensor networks [J]. IEEE Trans on Parallel and Distributed Systems, 2011, 22(3): 489 - 500.
- [11] Li Qun, Daniela R. Global clock synchronization in sensor networks [J]. IEEE Trans on Computers, 2006, 55(2): 214 - 226.

作者简介



陈 零 男, 1981 年 10 月出生于湖南省湘潭市. 博士研究生. 主要研究方向为无线传感器网络.

E-mail: chen_snake@qq.com

王建新 男, 1969 年 12 月出生于湖南省邵阳市. 博士, 教授, 博士生导师. 主要研究方向为无线传感器网络, 网络优化理论.

E-mail: jxwang@mail.csu.edu.cn

张士庚(通信作者) 男, 1981 年 12 月出生于山东. 博士, 讲师. 主要研究方向为无线传感器网络, 室内定位系统, RFID 系统.

E-mail: zhangshigeng@gmail.com