

## 基于移动参考节点的无线传感器时钟同步方法

张玉强<sup>1</sup>, 何泾沙<sup>2</sup>, 徐晶<sup>1</sup>, 赵斌<sup>2,3</sup>, 蔡方博<sup>2</sup>

(1. 北京工业大学 计算机学院, 北京 100124; 2. 北京工业大学 软件学院, 北京 100124; 3. 济宁学院 计算机科学系, 山东 曲阜 273155)

**摘要:** 现有对无线传感器网络节点中的时钟同步方法往往通过固定参考节点传递同步信息完成同步。对于固定参考节点通信范围外的节点, 需要同步信息在节点间的多跳传输, 这造成时间误差的累积以及节点多跳传输的能量损耗, 从而影响同步的精度和网络能耗。为解决此问题, 提出了基于移动参考节点和最优移动路径的时钟同步方法。首先, 采用正六边形的蜂窝覆盖技术选取最优的时钟同步点。其次, 结合最优覆盖和选址问题的数学模型, 设计最短移动距离的路径。最后, 提出了基于 MAC 层标记多个时间戳实现低能耗单跳同步的机制。与现有方法的实验对比表明, 该方法提高了同步时间精度, 并降低了通信能耗。

**关键词:** 时钟同步; 移动参考节点; 最优移动路径; 单跳时钟同步机制

**中图分类号:** TP393

**文献标识码:** A

## Time synchronization method for wireless sensor networks based on mobile reference nodes

ZHANG Yu-qiang<sup>1</sup>, HE Jing-sha<sup>2</sup>, XU Jing<sup>1</sup>, ZHAO Bin<sup>2,3</sup>, CAI Fang-bo<sup>2</sup>

(1. College of Computer Science, Beijing University of Technology, Beijing 100124, China; 2. School of Software Engineering, Beijing University of Technology, Beijing 100124, China; 3. Department of Computer Science, Jining University, Qufu 273155, China)

**Abstract:** Most existing time synchronization methods in wireless sensor networks depend on the existence of one or more fixed reference nodes and on the propagation of synchronization messages in the networks. Thus, they often suffer from the disadvantages of high energy consumption and large cumulative error due to multiple layers of communication. To address such issues, a new time synchronization method based on a mobile reference node and the optimal mobile path is proposed. Firstly, the cellular coverage technique is analyzed to identify the optimal synchronization points. Then, the optimal mobile path with minimal length is designed. Finally, an efficient single hop time synchronize mechanism based on multiple timestamps in the MAC layer is presented. Comparison with other methods shows that the proposed method can achieve both higher synchronization accuracy and lower energy consumption.

**Key words:** time synchronization; mobile reference node; optimal mobile path; single hop time synchronize mechanism

### 1 引言

近年来, 学术界和工业界对无线传感器网络产生了极大兴趣, 而作为无线传感器网络支撑技术之一的传感器时钟同步算法, 也同样成为研究热点。在无线传感器网络中, 时间信息对于节点间相互协

作, 如节点周期性睡眠-唤醒调度机制、协作传输机制以及安全协议以及数据的采集和有效整合都至关重要<sup>[1]</sup>, 同时, 时钟精度会对它们的性能产生关键性的影响。传感器需要通过自身的晶体振荡器(简称晶振)维持本地时钟。然而晶振在制造工艺上差别较大, 且易受到环境中电压、磁场、温度及晶体老

收稿日期: 2015-10-13

基金项目: 北京市自然科学基金资助项目(4142008); 国家高技术研究发展计划(“863”计划)基金资助项目(2015AA017204); 国家自然科学基金资助项目(61272500); 山东省自然科学基金资助项目(ZR2013FQ024)

**Foundation Items:** The Natural Science Foundation of Beijing(4142008); The National High Technology Research and Development Program of China(863Program)(2015AA017204); The National Natural Science Foundation of China(61272500); The Natural Science Foundation of Shandong Province(ZR2013FQ024)

化等多种因素的影响,从而在运行一段时间之后,传感器网络节点的时钟便无法保持同步。另一方面,由于无线传感器节点具有能量受限、体积和成本受限、通信距离受限等诸多限制,传统的时钟同步技术(如 NTP 等)不能直接应用。从而大量学者针对无线传感器网络的时钟同步问题进行了深入研究。

## 2 相关工作

2002 年, Jeremy Elson 等<sup>[2]</sup>首次研究了无线传感器时钟同步技术的问题。此后,各种同步算法被逐渐提出。通过对现有文献分析,本文将现有的同步方法分成 2 类,第一类方法基于固定参考节点,其选用时钟源的位置固定不变;第二类方法基于移动参考节点,其选用时钟源做周期性运动。第一类同步方法选用固定参考节点,通过同步信息的多跳传输实现整个网络的同步。如 RBS 算法<sup>[3]</sup>通过固定参考节点广播同步消息,节点通过比较同步信息的本地时间戳计算时间偏差,实现同步。但随着网络规模的增大,节点间的通信量以节点数量的平方增长。TPSN<sup>[4]</sup>算法采用双向成对同步机制。固定参考节点与单跳范围内的节点进行双向通信,节点通过计算传输时延和时钟偏移实现同步。全网的同步需求借助于同步信息的多跳传输不可避免地产生了误差的多跳累积,对时间精度影响较大;DMTS<sup>[5]</sup>算法利用单向同步机制,固定参考节点向节点单向通信,节点利用时间戳计算传输延迟实现同步,它降低了整个网络能量消耗,但牺牲了同步精度;TS/MS<sup>[6]</sup>算法采用双向成对同步机制,固定参考节点与单跳范围内的节点进行双向通信,节点通过估计时钟偏差实现同步,降低了通信开销,但要求节点时钟频偏和相偏保持不变,实际很难满足;FTSP<sup>[7]</sup>算法利用 MAC 层嵌入时间戳技术,消除了发送时间,访问时间,接收处理时间延迟。全网同步中,同步信息采用泛洪方式进行多跳传输,产生了时间偏差多跳积累,泛洪通信量较大。LTS 算法<sup>[8]</sup>、TSS 算法<sup>[9]</sup>、TDP 算法<sup>[10]</sup>等都采用固定参考节点,全网同步需要多跳传输。

这些算法的主要工作原理为:首先,固定参考节点同步通信半径内的邻居节点,继而,实现同步的邻居节点充当“参考节点”,同步邻居节点通信半径内参考节点通信半径外的节点。按照此规律实现全网同步,即同步信息经过多跳传输实现网络同步。这造成了同步误差的多跳累积和能量损耗的增

大,同步过程如图 1 所示。

第二类同步方法选用移动参考节点,通过同步信息的单跳传输,实现网络同步。2004 年, Sichertiu 和 Ramadurai<sup>[11]</sup>首先利用移动参考节点进行了网络中节点定位的研究。Ssu 等<sup>[12]</sup>做了类似的研究,利用移动参考节点,提出了距离无关的测距定位方法。Pathirana 等<sup>[13]</sup>从进一步降低节点计算能耗的角度出发,提出了由移动参考节点直接计算节点的位置信息,节省了未知节点能耗。移动参考节点也被用于水下传感器网络的定位研究中<sup>[14]</sup>。

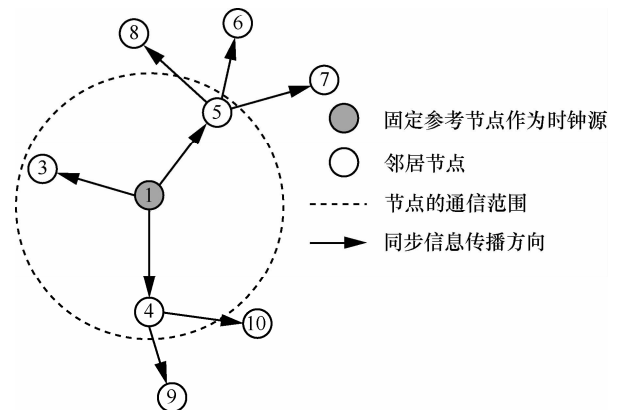


图 1 位置固定的参考节点作为时钟源进行同步的过程

2008 年, Boukerche 等<sup>[15]</sup>提出 LTS-MB 算法将时钟同步作为节点定位信息的另一个纬度,即 4 维度的定位问题。首次研究使用移动参考节点实现同步和定位。该方法中移动参考节点沿特定的曲线在网络中进行移动,对网络中的节点进行 3 次通信覆盖。参考节点移动到网络的特定位置时,与节点进行直接通信,采用数据分组延迟测量技术或数据分组的往返时间技术(RTT)实现节点同步。该工作与本文的主要不同在于:1)移动路径上,本文对参考节点的移动路径进行最优规划只需要与节点进行一次通信即可完成同步,相比 LTS-MB 算法缩短了参考节点的移动路径,在相同移动速度的条件下,缩短了移动周期,提高了 2 次同步周期时间段内的节点同步精度;2)通信方式上,本文同步信息传送采用时间戳技术,节点只需要接收数据,不需要发送数据,这种单向通信模式,相比 LTS-MB 算法采用的双向通信模式,大大减少了通信能耗。Shikoska 等<sup>[16]</sup>、Zhang 等<sup>[17]</sup>分别从网络拓扑和移动参考节点的误差角度继续了 Boukerche 等的研究,这与本文提出的研究方向不同。

这些算法的主要工作原理为:选用移动参考节

点,此节点通常装有 GPS 接收装置能够获取自身时间和位置信息,同时能够在整个网络中进行移动,当参考节点移动到特定位置时,向单跳通信范围内的邻居节点发送同步信息,实现节点同步,消除了因多跳传输产生的同步误差累积和能量损耗,如图 2 所示。

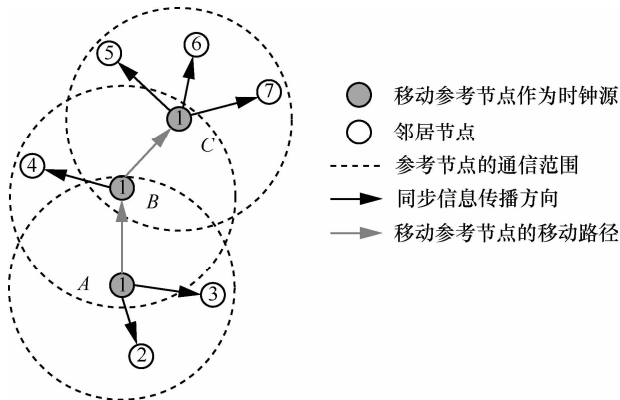


图 2 基于移动参考节点的无线传感器网络时钟同步模型

综上所述,基于固定参考节点的时钟同步方法,全网同步需要同步信息的多跳传输,产生了同步误差的累积和网络的能量损耗。基于移动参考节点的时钟同步方法,利用参考节点的移动、单跳通信实现全网同步。与前者相比,消除了同步信息的多跳传输,提高同步精度,降低能量损耗。但是到目前,还没有一种基于移动参考节点只针对同步问题进行研究的方法。

本文提出了一种基于移动参考节点的无线传感器时钟同步方法(MRN-CS, cycle time synchronous method based on mobile reference node)。参考节点在网络中沿规划路径周期性移动。参考节点移动到特定位置(同步点)时,与节点进行单跳通信,实现节点同步。

### 3 基于移动参考节点的时钟同步方法

在此首先定义同步区域和同步点。

同步区域为需要进行时钟同步的节点分布的网络区域。本文选取长为  $L$ 、宽为  $W$  的矩形同步区域为研究对象。每个同步区域利用一个移动参考节点进行时钟同步。对于大规模网络,可以划分为多个同步区域分别同步。

同步点为在同步区域内,移动参考节点直接与节点进行通信的坐标点。由于参考节点的通信覆盖范围为圆形,同步点选取的问题。其相应的数学模

型为使用圆形单次最优的覆盖矩形,圆心坐标为同步点的坐标值。

本文选用移动参考节点作为同步的时钟源,使参考节点在网络中按照预先规划的路径进行周期性的移动。参考节点移动到同步点时,与通信半径内的邻居节点进行一次单跳通信。参考节点发送同步信息,实现节点同步。参考节点移动一个周期后,实现网络中所有节点的同步。同步过程中,同步信息的传送采用单跳通信,消除了因多跳传输产生的误差累积。不需要节点充当“参考节点”,减小了能量损耗,如图 3 所示。

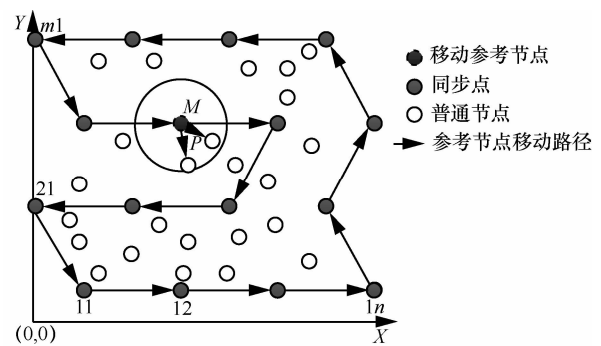


图 3 移动参考节点周期时钟同步移动路径

本文从移动参考节点在同步区域内的同步点的选取,移动路径规划,单跳通信 3 个方面论述基于移动参考节点的无线传感器时钟同步方法。

### 4 移动参考节点的同步点选取

参考节点在周期移动过程中,要求与同步区域内的所有节点进行一次单跳通信。为实现这一目标,参考节点在每个同步点形成的广播区域需要单次连续的覆盖整个同步区域,即参考节点的移动中的同步问题过程转化为同步区域的覆盖问题。

参考节点在同步点的广播域,是以同步点为圆心、以移动参考节点通信半径为半径的圆形区域。为实现单次、连续的覆盖,在数学模型上,借助正六边形蜂窝覆盖原理。将参考节点在同步点的广播圆等效成圆内接正六边形区域,对同步区域进行单次无缝覆盖。移动参考节点的同步点选取问题,等效为在同步区域中,需找半径为  $R$  的圆内六边形的圆心坐标问题,如图 4 所示。

本文选用长为  $L$ 、宽为  $W$  的矩形同步区域。建立二维坐标系,设区域的起点处恰在一个同步点上,坐标为  $(x_0, y_0)$ 。通过根据正六边形的位置关系,

得到有规律的圆心坐标计算公式，继而得到选取同步点坐标值。

由于正六边形对矩形区域进行覆盖时，区域边界处存在未被覆盖的部分。为了易于同步区域的扩展，在边界区域外选取额外同步点，保持圆心坐标公式规律不变。

### 4.1 同步区域内边界点的选取

设同步点坐标构成矩阵为  $m$  行  $n$  列，移动参考节点的广播圆半径为  $R$ ，按照正六边形蜂窝覆盖原理，任意相邻 3 个同步点构成边长为  $\sqrt{3}R$  的等边三角形原则，构成矩阵如下

$$A = \begin{bmatrix} \vdots & \vdots & \vdots & \ddots \\ 1 & 0 & 1 & \dots \\ 0 & 1 & 0 & \dots \\ 1 & 0 & 1 & \dots \end{bmatrix}, a_{ij} = \begin{cases} 0 & i, j \text{ 不同为奇数或偶数} \\ 1 & i, j \text{ 同时为奇数或偶数} \end{cases}$$

其中，“0”表示此位置没有同步点，“1”为此位置有同步点。 $i, j$  同时为奇数或偶数时，此位置选为同步点，否则，不选为同步点。利用同步点之间的位置关系，可知区域中的元素  $a_{ij}$  坐标为

$$(x_{ij}, y_{ij}) = \begin{cases} \left( \frac{\sqrt{3}R(j-1)}{2} + x_0, \right. & i, j \text{ 同为奇偶;} \\ \left. \frac{3(i-1)}{2}R + y_0 \right) & i = 1, \dots, m, j = 1, \dots, n \\ 0 & i, j \text{ 不同为奇偶;} \\ & i = 1, \dots, m, j = 1, \dots, n \end{cases} \quad (1)$$

由此确定元素矩阵中各元素坐标。

### 4.2 区域边界处的同步点选取

由于同步区域的起点与同步点重合，如图 4 所示，同步区域的右边界和上边界存在未覆盖区域。由式 (1) 可知，同步点矩阵相邻列间距为  $\frac{\sqrt{3}R}{2}$ ，相邻行间距为  $\frac{3}{2}R$ 。利用行间距和列间距来分割同步区域矩形的长  $L$ 、宽  $W$ ，得到同步点的行列个数。

$$\text{取整计算 } \lambda = \left\lceil \frac{L}{\left(\frac{\sqrt{3}R}{2}\right)} \right\rceil, \alpha = \left\lceil \frac{W}{\left(\frac{3}{2}R\right)} \right\rceil;$$

$$\text{取余计算 } \theta = \left\{ \frac{L}{\left(\frac{\sqrt{3}R}{2}\right)} \right\}, \beta = \left\{ \frac{W}{\left(\frac{3}{2}R\right)} \right\}。 \text{由此}$$

可计算得到同步点矩阵的行数和列数为

$$\text{行数: } m = \alpha + 1 = \left\lceil \frac{W}{\left(\frac{3}{2}R\right)} \right\rceil + 1$$

$$\text{列数: } n = \lambda + 1 = \left\lceil \frac{L}{\left(\frac{\sqrt{3}R}{2}\right)} \right\rceil + 1$$

#### 1) 对区域右边界分析

当  $\theta = 0$  时，区域右边界恰好位于最后一列，即  $\lambda$  列同步点上，右边界被完全覆盖，则不需要增加同步点。

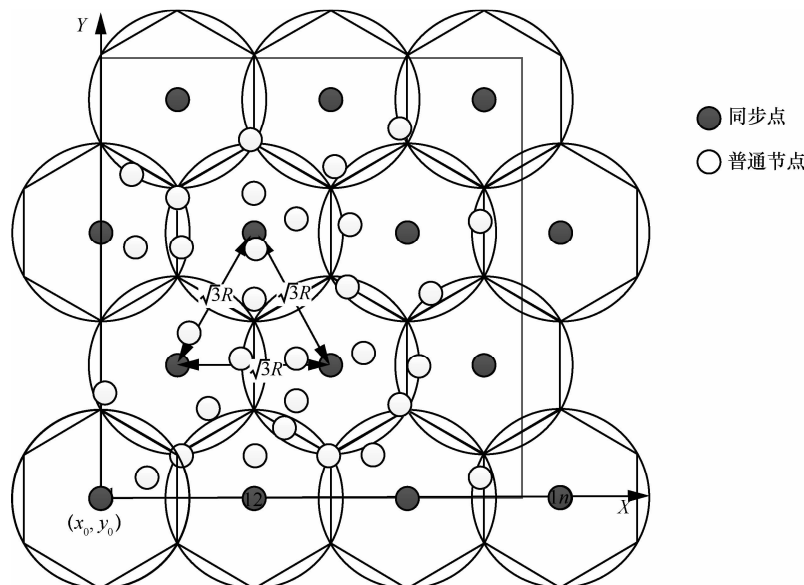


图 4 无线传感器网络同步区域的蜂窝覆盖及同步点

当  $\theta \neq 0$  时，区域右边界恰好位于最后一列之外，即  $\lambda$  列同步点外，右边界有未完全覆盖区域，需要额外增加 1 列同步点。

2) 对区域上边界分析

同理，当  $\beta = 0$  时，区域上边界恰好位于最上一行，即  $\alpha$  行同步点上，上边界被完全覆盖，则不需要增加同步点。

当  $\beta \neq 0, \beta \leq \frac{R}{2}$  时，区域上边界位于最上一行同步点广播圆的覆盖内，上边界被完全覆盖，则不需要增加同步点。

当  $\beta \neq 0, \frac{3R}{2} > \beta > \frac{R}{2}$  时，区域上边界位于最上一行同步点广播圆覆盖外，上边界有未覆盖区域，需要额外增加 1 行同步点。

综上所述，当  $\theta = 0$  时，不需要增加同步点。当  $\theta \neq 0$  时，右边界增加 1 列同步点，同步点矩阵的列数  $n = \lambda + 2$ 。当  $\beta = 0$  时，不需要增加同步点。

当  $\beta \neq 0, \beta \leq \frac{R}{2}$  时，不需要增加同步点。当  $\beta \neq 0, \frac{3R}{2} > \beta > \frac{R}{2}$  时，上边界增加 1 行同步点，同步点矩阵的行数  $m = \alpha + 2$ 。

根据  $\theta, \beta$  的不同取值，形成了 6 种不同组合。以  $\theta \neq 0, \beta = 0$  为例，同步区域右边界需要增加 1 列同步点，根据同步点坐标式 (1)，对同步点行、列的奇偶性进行讨论，即需要对  $\alpha, \lambda$  的奇偶性进行讨论，形成了 4 种组合。

以  $\alpha$  为奇数， $\lambda$  为偶数例，区域右边界外增加 1 列，即同步区域中同步点矩阵新增加的  $\lambda + 1$  列，继而同步区域中所有同步点的坐标矩阵

$$\begin{bmatrix} (x_0, \frac{3}{2}\alpha R + y_0) & \vdots & \vdots & \vdots & (\frac{\sqrt{3}}{2}(\lambda + 1)R + x_0, \frac{3}{2}\alpha R + y_0) \\ \vdots & \vdots & \vdots & \vdots & \dots \\ \vdots & \vdots & \vdots & \dots & 0 \\ (x_0, 3R + y_0) & \vdots & \dots & \dots & (\frac{\sqrt{3}}{2}(\lambda + 1)R + x_0, 3R + y_0) \\ 0 & (\frac{\sqrt{3}R}{2} + x_0, \frac{3}{2}R + y_0) & \dots & \dots & 0 \\ (x_0, y_0) & 0 & (\sqrt{3}R + x_0, y_0) & \dots & (\frac{\sqrt{3}}{2}(\lambda + 1)R + x_0, y_0) \end{bmatrix} \quad (2)$$

区域中同步点的数量为矩阵 (2) 中元素的个数， $m = \alpha + 1, n = \lambda + 2$ ， $\alpha$  为奇数， $\lambda + 1$  为奇数。因此计算公式为

$$N = \frac{m}{2}n = \left(\frac{\alpha + 1}{2}\right)(\lambda + 2) \quad (3)$$

其中， $N$  表示为二维坐标系中移动参考节点所要经过的同步点数量。

5 移动路径规划

本文首先建立最优移动路径模型，继而分析，制定了移动路径规划原则。记同步点集合为  $I = \{1, 2, \dots, N\}$ ，并且同步点  $i$  和同步点  $j$  之间的距离为  $d_{i,j}$ 。为获得最优的参考节点移动路径，建立如下的最优旅行商模型

$$\begin{aligned} \min & \sum_{i \in I} \sum_{j \in I} d_{i,j} x_{i,j} \\ \text{s.t.} & \begin{cases} \sum_{j \in I, j \neq i} x_{i,j} = 1, & \forall i \in I \\ \sum_{i \in I, i \neq j} x_{i,j} = 1, & \forall j \in I \\ \sum_{i \in S, j \in S} (x_{i,j} + x_{j,i}) \geq 2, & \forall S \neq \emptyset, S \subseteq I, S \neq I \\ x_{i,j} \in \{0, 1\}, & \forall i \in I, j \in I \end{cases} \end{aligned}$$

其中，决策变量  $x_{i,j}$  为 0-1 变量，其取值为 1 表示参考节点将从同步点  $i$  移动到同步点  $j$ ；其值为 0 表示参考节点将不会从同步点  $i$  移动到同步点  $j$ 。该模型的目标在于最小化参考节点移动路径之和。第 1 个约束要求参考节点只能从任意同步点  $i$  移动到另一个新的同步点  $j$ ，相当于移动参考节点在同步点  $i$  的出口只能是同步点  $j$ 。第 2 个约束要求参考

节点到任意同步点  $i$  只能从另一个同步点  $j$  中移动过来, 相当于移动参考节点到同步点  $i$  的入口只能是同步点  $j$ 。第 3 个约束要求参考节点的路径只能构成一个整体环路, 不存在小的环路。

参考节点的最优移动路径规划模型包含  $N^2$  个决策变量,  $2^N + 2N - 2$  个约束。该模型是一个包含大量约束的 0-1 线性规划问题, 此类问题属于难于精确求解的问题, 不能直接用于无线传感器网络, 目前常见的算法包括基于线性松弛的求解算法、基于启发式策略的近似求解算法和基于智能算法的随机优化算法 (比如基于遗传算法、禁忌搜索等方法的算法)。

本文针对正六边形蜂窝覆盖, 根据同步点的空间位置关系设计最优的移动路径进行求解, 其中, 特殊条件是任意相邻的 2 个同步点的间距是固定值  $\sqrt{3}R$ , 相当于求解已知特殊条件下的旅行商问题。

根据移动参考节点周期同步要求, 移动路径的规划设计原则为如下。

1) 移动参考节点周期移动, 起点和终点为同一点。

2) 移动参考节点在 2 个同步点移动的距离均为  $\sqrt{3}R$  (任意 2 个相邻同步点的间距为  $\sqrt{3}R$ )。

3) 移动参考节点在移动过程中经过所有的同步点, 同时每个同步点有且经过一次。

移动参考节点在满足时钟同步原则的前提下, 其移动路径可以根据同步精度、定位效率、能量损耗等参数, 进行自主规划, 比较灵活, 移动路径不唯一。

根据移动路径的规划设计原则, 可知移动参考节点的移动路径长度为  $N\sqrt{3}R$ 。

结合同步点数量计算式 (4), 移动参考节点同步周期  $T$  为

$$T = N \frac{\sqrt{3}R}{V} + Nt = N \left( \frac{\sqrt{3}R}{V} + t \right) = \left( \frac{\alpha+1}{2} \right) (\lambda+2) \left( \frac{\sqrt{3}R}{V} + t \right) \quad (5)$$

其中,  $N$  为二维坐标系中移动参考节点所要经过的同步点的数量,  $V$  表示移动参考节点的移动速度,  $t$  表示移动参考节点在同步点的广播时间。

## 6 单跳通信

为了进一步降低节点能耗, 采用基于 sender-

receiver 的单向同步算法, 节点不发送只接收同步信息即可完成同步。移动参考节点利用 MAC 层嵌入时间戳的技术<sup>[7]</sup>, 在发送的同步信息中嵌入多个发送时刻。节点在 MAC 层接收时, 记录对应的多个接收时刻。即移动参考节点在单个广播消息分组的多个位置记录下各自的发送时间戳, 并嵌入在广播消息分组中一起发送给节点。节点接收数据分组时, 记录每个发送时间戳对应的接收时间戳。节点根据多个时间戳估算出中断等待时间以及静态设定的编码时间, 继而采用线性回归时间进行补偿, 得到精确的同步时刻。

节点首先建立一张线性回归表用于存储接收时间戳和发送时间戳。由于节点的计算和存储能力受限, 线性回归表只存储最新的  $n$  个时间戳。节点利用线性回归方法, 得到时钟漂移率  $skew$

$$skew = \frac{\sum_{i=1}^n (T_{offset_i} - \overline{T_{offset}})(T_{arrive_i} - \overline{T_{arrive}})}{\sum_{i=1}^n (T_{arrive_i} - \overline{T_{arrive}})^2} \quad (6)$$

$$T_{offset_i} = T_{send_i} - T_{arrive_i} \quad (7)$$

其中,  $n$  为线性回归表中时间戳的个数,  $skew$  表示通过线性回归得到的时钟漂移率。  $T_{send_i}$  表示移动参考节点标记的第  $i$  个发送时间戳。  $T_{arrive_i}$  表示第  $i$  个发送时间戳对应的接收时间戳。  $T_{offset_i}$  为相对于第  $i$  个发送时间戳的时钟偏差。由式 (6) 计算得到,  $\overline{T_{offset}}$  和  $\overline{T_{arrive}}$  分别表示相对于参考节点的时钟偏差和接收时间戳的平均值。

由式 (6) 得到的时钟漂移率  $skew$ , 来调整节点的时钟

$$T_{global} = T_{local} + \overline{T_{offset}} + skew(T_{local} - \overline{T_{arrive}}) \quad (8)$$

其中,  $T_{global}$  为节点调整后与参考节点一致的全局时间,  $T_{local}$  为节点的本地时间。

## 7 仿真实验

在仿真实验中, 采用 SmeshLink 公司研发的无线网络开发套件和 Contiki Studio 操作系统。分别测试 MRN-CS 算法、LTS-MB 算法、TPSN 算法和 FTSP 算法的同步误差、同步误差的概率累积分布、同步周期内时钟偏移变化和能量损耗实验, 分析 MRN-CS 算法同步性能优点及不足。实验条件同 LTS-MB 算法<sup>[15]</sup>所示。

在 92 m×92 m 的同步区域中根据 MRN-CS 算法，选取的同步点如图 5 所示。

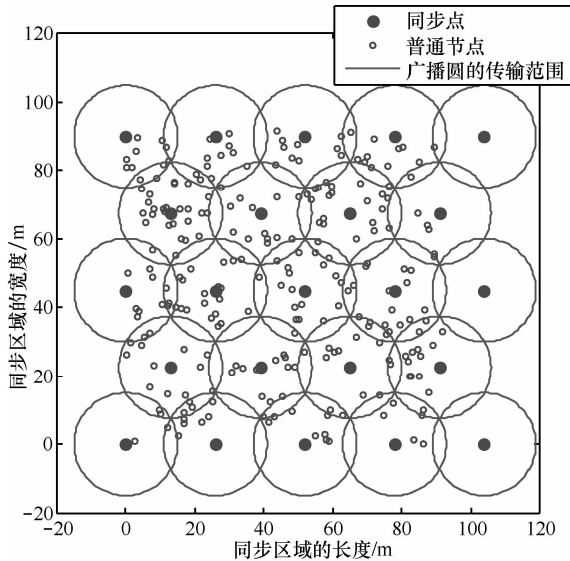


图 5 同步区域中移动参考节点的同步点位置及广播圆

由图 5 可知，相邻两同步点间距为  $\sqrt{3}R$ 。根据同步区域外的同步点选取法则，需要在同步区域的右边界外，增加了新的一列同步点。在每个同步点处形成的广播圆完成了对同步区域的覆盖。

根据提出最优路径规划原则，MRN-CS 算法中参考节点的移动路径如图 6 所示。

从图 6 中可见，移动参考节点的路径形成了一个闭合曲线。移动参考节点在两同步点间移动的距离为  $\sqrt{3}R$ 。移动过程中经过所有的同步点，同时有且经过一次。

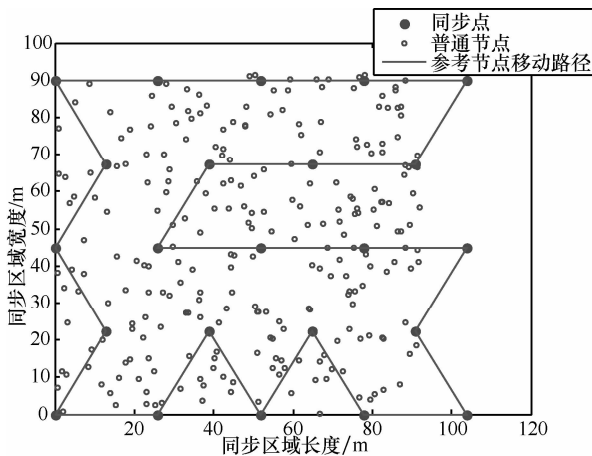


图 6 移动参考节点的移动路径

首次提出利用移动参考节点进行时钟同步的 LTS-MB 算法，其移动路径如图 7 所示。

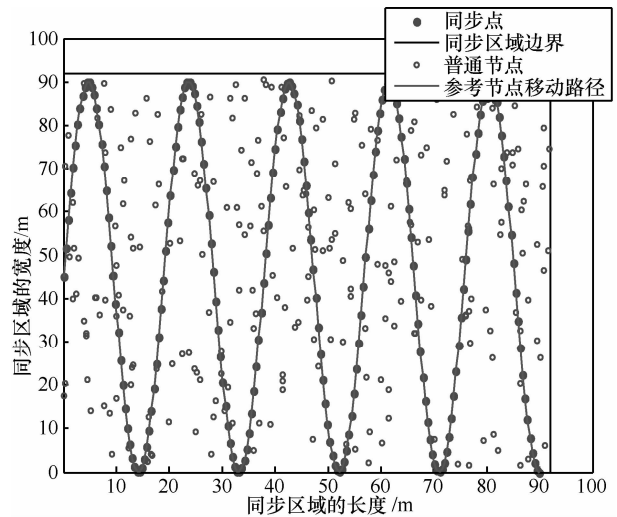


图 7 LTS-MB 算法参考节点移动轨迹

由图 7 可知，参考节点的移动路径为正弦曲线，LTS-MB 算法中没有提出移动参考节点怎么进行周期移动，在此假设移动参考节点以坐标 (0, 45) 为起点，沿正弦曲线进行移动。

### 7.1 同步误差比较

设 MRN-CS 算法、TPSN 算法、FTSP 算法的参考节点统一设置在同一位置，即同步点坐标 (52, 45) 处。LTS-MB 算法参考节点的设置在 (0, 45) 处。TPSN 算法、FTSP 算法需要 5 跳才能覆盖同步区域中的所有节点。

实验过程中，分别采集从 1 跳至 5 跳的节点与参考节点的同步误差。为进一步降低误差影响，要求采集节点刚刚完成同步时的误差。即每个一个同步周期结束后立刻采集 1 次，每种算法各采集 50 个同步周期，然后取采集数据的均值。在相同条件下对 4 种算法进行测试，最终得到 MRN-CS 算法、TPSN 算法、FTSP 算法、LTS-MB 算法的平均同步误差随同步消息传递跳数的变化关系，实验结果如图 8 所示。

从图 8 中同步误差的变化趋势可知，随着跳数的增加，FTSP 算法和 TPSN 算法的同步误差逐渐增大。MRN-CS 算法和 LTS-MB 算法的同步误差随同步信息传递跳数而变化，MRN-CS 算法优于 LTS-MB 算法。随着跳数的增加，MRN-CS 算法和 LTS-MB 算法的同步误差远小于其他 2 种算法，同时 FTSP 算法优于 TPSN 算法。

从图中同步误差的数量关系可知，MRN-CS 算法的同步误差稳定在一跳平均同步误差为 2 μs 左右。LTS-MB 算法的同步误差范围稳定在一跳平均同步误差 6 μs 左右。FTSP 算法第 1 跳时的同步误

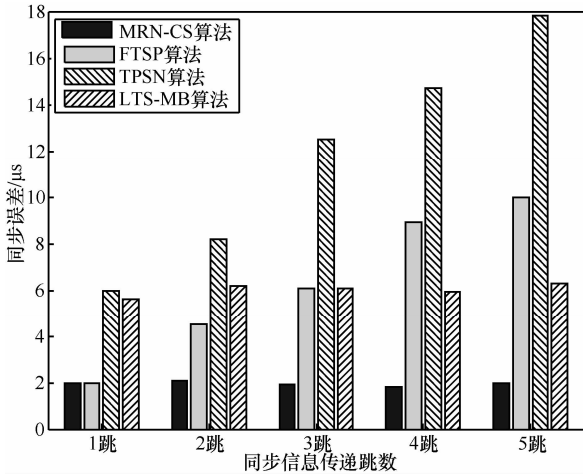


图8 同步误差的比较

差与MRN-CS算法相同为2 μs,但在第5跳时的平均同步误差为10 μs,从第1跳到第5跳,平均每跳增加了2 μs。TPSN算法第1跳的同步误差为6 μs,在第5跳的平均同步误差为18 μs,从第1跳到第5跳,平均每跳增加了3 μs。综上MRN-CS算法和LTS-MB算法的同步误差不受跳数变化的影响,MRN-CS算法优于LTS-MB算法,且优于另外2种算法。

### 7.2 同步误差的概率累积比较

根据节点具有的同步误差进行分类。对同步区域中的256个节点进行概率统计,4种同步算法的概率累积分布函数曲线如图9所示。

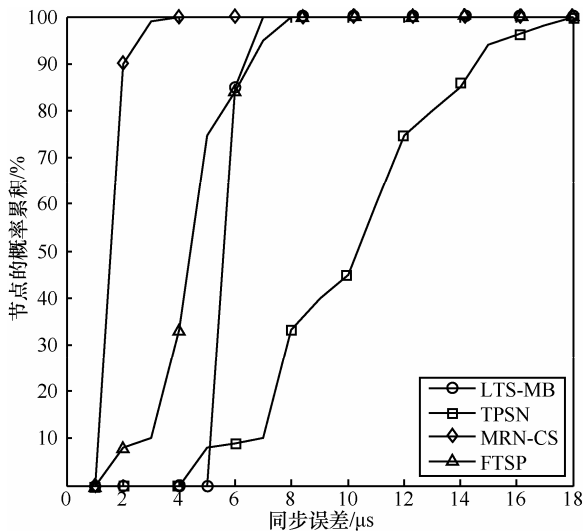


图9 同步误差的概率累积分布函数曲线

曲线的陡峭程度,反映了相同同步误差下节点数量,越陡峭说明越多的节点拥有越小的同步误差。从图10可以看出,MRN-CS算法和LTS-MB

算法具有陡峭的上升沿,而TPSN算法和FTSP算法上升沿坡度较缓。MRN-CS算法中90%的节点误差小于2 μs,几乎所有节点的误差都在3 μs之内,因此,MRN-CS算法优于LTS-MB算法,且优于TPSN和FTSP算法。

### 7.3 同步周期内时钟偏移的比较

节点的时钟偏移,即累积误差随着同步周期的大小而变化。由于节点晶振频率不稳等原因,需要参考节点进行周期性的时钟同步。然而基于移动参考节点的时钟同步方法中,只有当参考节点移动到同步点时,节点才能通过接收同步信息实现同步。即在一个同步周期内,节点将会产生误差累积,大小取决于同步周期的长短。

实验条件设置如下:MRN-CS算法和LTS-MB算法,选取测试节点为周期同步起点处的距离参考节点一跳范围内的节点,且2种算法的同步周期只考虑参考节点的周期移动时间。TPSN算法由于是位置固定的参考节点,其同步周期时间设定为5 s。选取的测试节点为距离参考节点一跳范围内节点,每隔1 s采集一次数据,同时将每次采集的数据求其平均值,实验时间为60 s。实验结果如图10所示。

从图10可知,节点的累积误差随测试时间逐渐增大总体呈线性趋势。LTS-MB算法中参考节点的同步周期时间较长接近44 s,节点的累积误差1830 μs,是3种算法中累积误差最大的。TPSN算法采用位置固定的参考节点进行时钟同步,同步周期只有5 s,此时节点的累积同步误差只有200 μs。而本文提出的MRN-CS算法中参考节点的同步周期为30 s,此时节点的累积误差1220 μs,因此,同步周期内时钟偏移角度,MRN-CS算法优于LTS-MB算法,但逊于TPSN算法。

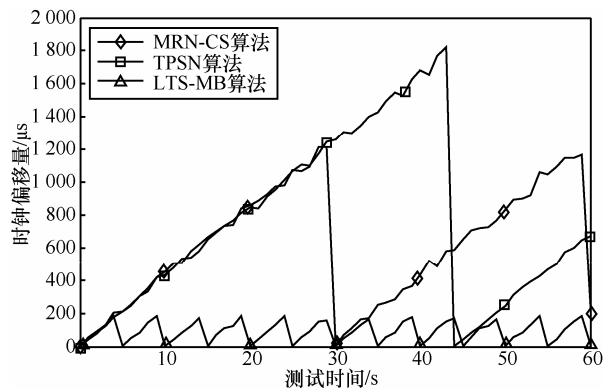


图10 同步周期内时钟偏移比较

## 7.4 能量损耗比较

从网络能耗的角度,对 MRN-CS 算法、LTS-MB 算法和 TPSN 算法进行比较。网络中节点的能量主要由电池供给,当节点的电池能量耗尽时,节点会停止工作而失效。因此,必须要考虑时钟同步过程中的各步骤的能量消耗。

对于位置固定的参考节点同步算法,以 TPSN 算法为例。在节点的单跳同步过程中,节点需发送 1 次,接收 1 次同步信息。多跳同步时,节点充当“参考节点”为下一跳节点服务,因此每增加一跳,需要增加 1 次发送和 1 次接收同步信息。本实验过程中不考虑层次发现阶段产生的网络能耗,只关心同步阶段的网络能耗。

对于基于移动参考节点同步算法,以 LTS-MB 算法和 MRN-CS 算法为例,它们消除了同步信息在网络中的多跳传输,但在能耗方面差距较大。LTS-MB 算法,将同步和定位作为一个问题进行研究。在参考节点移动过程中,与每个节点通信 3 次,分别进行 3 次同步,每次同步采用 RTT 机制,需要节点发送 1 次,接收 2 次同步数据。然而 MRN-CS 算法,在同步全过程中,节点无需发送数据,只需要接收参考节点 1 次同步数据。

实验中其他参数设置如下:  $E_0 = 50 \text{ nJ/bit}$ ,  $\varepsilon = 0.1 \text{ nJ}/(\text{bit} \cdot \text{m}^2)$ ,  $k = 2$ ,  $\eta = 0.5$ , 同步信息长度  $n = 25 \text{ bit}$ , 选取在通信半径处的节点  $d = R = 15 \text{ m}$ 。随着网络规模的增大,测试 3 种时钟同步算法的网络能耗,如图 11 所示。

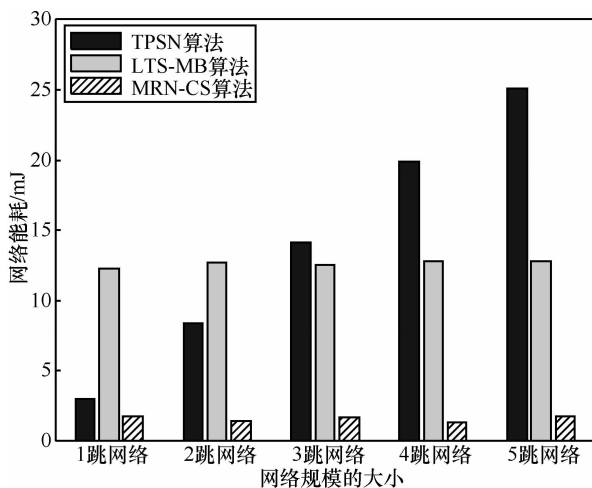


图 11 不同网络规模下能量消耗比较

从图 11 可以看出,移动参考节点时钟同步算法(LTS-MB 算法和 MRN-CS 算法)造成的网络能耗,

整体上小于位置固定的参考节点时钟同步算法(如 TPSN 算法)造成的网络能耗。随着网络规模的逐渐增大,TPSN 算法产生的网络能耗逐渐增大,网络规模每增大一跳,网络能耗的增长量是单跳网络能耗的 2 倍。

对于 LTS-MB 算法和 MRN-CS 算法随网络规模的增大,造成的网络能耗基本保持不变,同时 MRN-CS 算法要远优于 LTS-MB 算法。MRN-CS 算法造成的网络能耗比 TPSN 算法要小得多,随着网络规模以多跳方式的增长,MRN-CS 算法的网络能耗一直能够保持很低的水平,这一优势尤其在大规模网络应用中显得尤为重要。MRN-CS 的优势是不仅可以为每一个节点节省能量,而且可以使网络中的节点的能量平均地消耗,从而防止一些节点由于过早地耗尽能量而失效,延长了整个网络的生命周期。并且使网络中的节点无线通信量相同,这些因素对于有效减少节点整体能耗有着十分重要的作用。

综上,从同步误差实验和概率累积比较实验结果比较可知,MRN-CS 算法优于 LTS-MB 算法,同时明显优于 TPSN 算法和 FTSP 算法。基于移动参考节点的时钟同步算法(如 MRN-CS 算法和 LTS-MB 算法)在一跳范围内完成时钟同步,消除了多跳传输对同步误差的累积,因此刚刚完成时钟同步时,节点的同步误差要远小于位置固定的参考节点时钟同步算法(TPSN 算法和 FTSP 算法)。

从同步周期内时钟偏移实验可知,MRN-CS 算法优于 LTS-MB 算法,但是明显差于 TPSN 算法。这是由于移动参考节点的时钟同步算法同步周期较长,在一个同步周期内,节点会产生时钟偏移的累积,为了减少误差的累积,在参考节点移动速度恒定的条件下,需要研究参考节点的最短移动路径。

从能量损耗实验可知,MRN-CS 算法优于 LTS-MB 算法,同时明显优于 TPSN 算法和 FTSP 算法。MRN-CS 算法只需要节点接收同步信息,不需要发送,因此能量损耗最低。

## 8 结束语

本文提出了基于移动参考节点的时钟同步算法 MRN-CS 算法,从同步区域中采集的选取、移动路径的规划、单跳通信等 3 个方面阐述了本算法。本算法不需要同步信息在网络中的多跳传送,因此消除了同步误差的多跳积累。仿真实验结果显示,相比于 LTS-MB 算法、TPSN 算法和 FTSP 算法,

MRN-CS 算法有效降低了同步误差, 大幅提高同步精度, 降低了网络能耗。然而本算也存在一些缺点, 例如节点在同步周期内会产生时钟偏移的累积, 可以通过研究最短移动路径进行解决。

### 参考文献:

- [1] ZHENG G Q, LI J D, ZHOU Z L. Energy-efficient data gathering protocol for multihop wireless sensor networks[J]. Journal of Software, 2010, 21(9):2320-2337.
- [2] XU C N, XU Y J, LI X W. New time synchronization techniques for wireless sensor networks[J]. Journal of Computer Research and Development, 2008, 45(1):138-145.
- [3] ELSON J, GIROD L, ESTRIN D. Fine-grained network time synchronization using reference broadcasts[A]. Proc 5th Symposium on Operating Systems Design and Implementation[C]. Boston, MA, USA, 2002. 1-17.
- [4] GANERIWAL S, KUMAR R, SRIVASTAVA M B. Timing-sync protocol for sensor networks[A]. Proc 1st International Conference on Embedded Networked Sensor Systems[C]. Los Angeles, CA, USA, 2003. 138-149.
- [5] PING S. Delay measurement time synchronization for wireless sensor networks[R]. Research Report IRB-TR-03-013, Intel Research Berkeley Lab, 2003.
- [6] SICHITIU M L, VEERARITIPHAN C. Simple, accurate time synchronization for wireless sensor networks[A]. Proc IEEE Wireless Communications and Networking Conference[C]. New Orleans, LA, USA, 2003. 1266-1273.
- [7] MAROTI M, KUSY B, SIMON G, *et al.* The flooding time synchronization protocol[A]. Proc 2nd International Conference on Embedded Networked Sensor Systems[C]. Baltimore, MD, USA, 2004. 39-49.
- [8] GREUNEN J V, RABAEY J. Lightweight time synchronization for sensor networks[A]. Proc 2nd ACM International Conference on Wireless Sensor Networks and Applications[C]. San Diego, CA, USA, 2003. 11-19.
- [9] ROMER K. Time synchronization in ad hoc networks[A]. Proc 2nd ACM Symposium on Mobile Ad-Hoc Networking and Computing[C]. Long Beach, CA, USA, 2001. 173-182.
- [10] SU W, AKYILDIZ I F. Time-diffusion synchronization protocol for sensor networks[J]. IEEE/ACM Trans Networking, 2005, 13(2): 384-397.
- [11] SICHITIU M L, RAMADURAI V. Localization of wireless sensor networks with a mobile beacon[A]. In MASS'04[C]. Florida, USA, 2004.
- [12] SSU K F, OU C H, JIAU H. Localization with mobile anchor points in wireless sensor networks[J]. Vehicular Technology, IEEE Transactions, 2005, 54(3):1187-1197.
- [13] PATHIRANA P, BULUSU N, JHA S, *et al.* Node localization using mobile robots in delay-tolerant sensor networks[J]. IEEE Trans on Mobile Comput, 2005, 4(4):83-103.
- [14] CHANDRASEKHAR V, SEAH W K, CHOO Y S, *et al.* Localization in underwater sensor networks: survey and challenges[A]. In WU-WNet'06[C]. New York, NY, USA, 2006. 33-40.
- [15] BOUKERCHE, AZZEDINE, *et al.* Localization in time and space for wireless sensor networks: a mobile beacon approach[A]. World of Wireless, Mobile and Multimedia Networks[C]. 2008. 1-8.
- [16] SHIKOSKA U R, DAVCEV D. An Energy-Efficient Approach for Time-Space Localization in Wireless Sensor Networks[M]. Springer Berlin Heidelberg, 2012.
- [17] ZHANG, VICTORIA Y, ALBERT K W. Closed-form solution for joint localization and synchronization in wireless sensor networks with and without beacon uncertainties[J]. International Journal of Wireless Information Networks, 2013, 20(2):120-139.
- [18] ZHANG V Y, WONG A K S. Simultaneous localization and time synchronization in wireless sensor networks via semidefinite programming[A]. Wireless Communications and Networking Conference (WCNC) [C]. 2010. 1-6.

### 作者简介:



张玉强 (1982-), 男, 山东临沂人, 北京工业大学博士生, 主要研究方向为信息安全、无线传感器网络。

何泾沙 (1961-), 男, 美籍华人, 北京工业大学教授、博士生导师, 主要研究方向为网络安全、测试与分析 and 云计算。

徐晶 (1983-), 女, 山东临沂人, 北京工业大学博士生, 主要研究方向为信息安全、无线传感器网络。

赵斌 (1979-), 男, 山东滕州人, 北京工业大学博士生, 济宁学院副教授, 主要研究方向为网络安全、信息取证、云计算。

蔡方博 (1990-), 女, 辽宁营口人, 北京工业大学硕士生, 主要研究方向为信息安全、无线传感器网络。