

# 无线传感器网络中多跳时间同步算法的研究

周新莲, 黄力

ZHOU Xin-lian, HUANG Li

湖南科技大学 计算机学院, 湖南 湘潭 411201

School of Computer Science & Engineering, Hunan University of Science & Technology University, Xiangtan, Hunan 411201, China

E-mail: xlzhou2@hnust.cn

ZHOU Xin-lian, HUANG Li. Research on time synchronization algorithm for multi-hop in wireless sensor networks. *Computer Engineering and Applications*, 2009, 45(27): 102-104.

**Abstract:** This paper presents a Cluster-based Time Synchronization (CBTS) algorithm for multi-hop wireless sensor networks. It starts from the Leach algorithm which distributes the network into different clusters. Based on the partition, the algorithm works in two phases: Horizontal time synchronization phase and vertical time synchronization phase. In the horizontal synchronization phase, it adopts traditional pair-wise packet exchange mechanism to finish the time synchronization between the base station and cluster heads through establishing a hierarchical topology structure. In the vertical synchronization phase, it adopts pair-wise packet exchange and unidirectional reference broadcast mechanism to finish the time synchronization between cluster heads and cluster members. Moreover, it uses linear least square to estimate the clock offset of nodes which achieves a good precision. Finally, the local clocks of all nodes in the network can synchronize with each other. The results of which simulated on NS-2 show that CBTS has a good precision and low message cost.

**Key words:** wireless sensor network; time synchronization; multi-hop; cluster-based

**摘要:**提出了一种基于簇型的多跳时间同步算法——CBTS。利用 Leach 算法将网络划分成不同的簇,在此基础上把节点间的时间同步分为水平同步和垂直同步两个阶段来完成。在水平同步阶段,通过构建基站与簇头节点的层次拓扑结构,采用双向消息交换同步机制来完成簇头节点与基站的时间同步。在垂直同步阶段,采用双向消息交换和参考广播相结合的同步机制,来完成簇头节点和簇成员之间的时间同步,并利用最小方差线性拟合的方法估计了节点的时钟偏差,提高了时钟同步的精度,最终实现了整个网络节点的时间同步。经过仿真测试,证明该算法具有较低的消息交换开销和不错的同步精度。

**关键词:**无线传感器网络;时间同步;多跳;分簇

DOI: 10.3778/j.issn.1002-8331.2009.27.031 文章编号: 1002-8331(2009)27-0102-03 文献标识码: A 中图分类号: TP393

## 1 引言

无线传感器网络 (WSN) 是由分布在物理空间上大量传感器节点通过自组织的方式构成网络, 节点协作完成数据采集、融合处理、通信发送等任务。作为分布式系统, 无线传感器网络具有高灵活性、强鲁棒性等优点, 被广泛应用于国防、环保、交通、医疗以及制造业等多个领域<sup>[1-2]</sup>。时间同步技术是无线传感器网络的关键组成部分, 在无线传感器网络的许多应用和关键技术中都离不开时间同步。例如目标监测与跟踪、智能交通等许多应用要求网络节点预先指导自己的位置, 并在通信协作中利用位置信息完成应用要求。网络中的节点保持时间同步, 则声波在节点间的传输时间很容易被确定, 将其转换为距离也很方便。

## 2 相关研究工作

2002年8月, Jeremy Elson 和 Kay Romer 在 HotNets-I 国

际会议上率先提出并阐述了无线传感网中时钟同步机制的研究课题, 引起了无线传感器网络研究领域的广泛关注。经过近几年的不懈努力, 至今已经提出了多种时间同步算法<sup>[3-4]</sup>。

RBS 协议<sup>[5]</sup>遵从节点的接受者-接受者模式, 由一个节点广播发送时间的参考分组, 广播域内的节点分别采用本地时间记录参考分组到达的时间, 然后交换记录的时间来确定它们之间的时间偏移量, 实现它们之间的时间同步。RBS 算法的优点是利用了无线信道的特性, 信标节点相对于所有节点同时发送到物理信道上, 去除了发送时间和访问时间带来的时间同步误差。采用统计方法获得了接收节点间时间差异的平均值, 并用最小平方的线性回归法估计时钟频率和时钟偏差。缺点是具有很大的交换次数, 节点很多的时候, 计算开销也非常大。

TPSN 协议<sup>[6]</sup>采用发送者-接收者模式, 采用层次型网络结构, 首先将所有节点按照层次结构进行分级, 然后每个节点和上一级的一个节点进行同步, 最终所有节点与根节点同步。

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60673061)。

作者简介: 周新莲, 女, 博士生, 研究方向为无线传感器网络。

收稿日期: 2008-05-23 修回日期: 2008-08-04

TPSN 的优点是在 MAC 层消息开始发送到无线信道的时刻才给同步消息加上时标, 消除了访问时间带来的时间同步误差。缺点是一旦根节点失效, 就要重新选择根节点, 增加了计算和能量开销。并且随着跳数的增加同步误差呈线性增长。

DMTS 协议<sup>[7]</sup>是基于发送者模型, 该机制中, 选择一个节点作为主节点广播同步时间。所有接收节点测量这个时间广播分组的延迟, 设置它的时间为接收到分组携带的时间加上这个广播分组的传输延迟, 这样所有接收到广播分组的节点都与主节点时间同步。DMTS 算法的优点是结合在链路层打时间戳的技术, 消除了发送时间和访问时间的影响, 算法简单, 通信开销小。缺点是同步精度相对较低。

可以看出目前大多的时间同步算法是基于单跳同步, 多跳同步的研究相对薄弱, 具体表现为过大的同步开销和随跳数增加而造成的累积误差。该文提出了一种基于簇型的多跳时间同步算法, 在能耗和误差中取得了很好的折中。

### 3 网络模型

假设  $n$  个传感器节点按均匀分布随机高密度部署在一个监测区域内, 具有如下性质:

(1)  $n$  个传感器节点被随机部署在监测区域, 基站不受能源限制, 其他传感器节点为电池驱动。

(2) 所有节点都为静止节点, 并且所有的位置信息可知。

(3) 每个节点采用全向天线, 节点之间为双向链路即  $A$  节点能和  $B$  节点通信,  $B$  节点也能和  $A$  节点通信。节点的通信范围为受限制, 其通信半径都为  $R$ 。

(4) 网络中的节点采用 Leach 算法被划分为不同的簇后, 每个簇由一个簇头节点和若干个簇成员节点组成。簇头节点管理簇内其他成员的时间, 位置等信息。

(5) 簇头节点的通信能力比普通成员节点强, 且簇中的所有簇成员都在彼此的通信范围内。

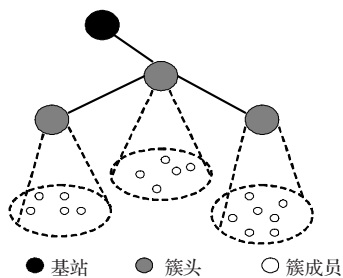


图1 CBTS 同步算法网络模型

从前述可知, 在无线传感器网络多跳时间同步中, 节点的同步误差随着其离时间基准点跳距的增加而增加。因此采用簇型网络的拓扑结构来减少同步路径上的平均跳数, 使得同步误差尽可能减小。

基站与簇头节点之间采用传统的双向成对同步机制。双向成对同步可以达到一定的同步精度, 但通信开销过大。假定同步广播域中有  $n$  个节点, 那么在一个同步周期内总共需要  $2n$  个分组交换, 其中信标 (beacon) 节点发送  $n$  个, 接收  $n$  个。为了提高精度, 增加同步频率会导致更多的能量损耗, 对于能耗敏感的无线传感器网络而言, 这样的开销有时是不可接受的。但是考虑到簇头节点的数目相对簇成员节点少很多, 不会造成过大的通信开销。

簇头节点与簇内成员节点之间采用双向同步与参考广播

相结合的时间同步机制。由于簇成员节点的数目众多, 采用该同步机制, 簇内的所有节点可以同时依据簇头节点发送的同步消息包分两次完成同步, 从而减少网络中消息包的碰撞。

### 4 基于簇型的多跳时间同步算法(CBTS)

CBTS 时间同步算法主要由三个关键算法组成: (1) 利用 Leach 分簇算法将网络划分成多个簇, 每个簇有且仅有一个簇头作为簇管理者。(2) 在水平同步阶段, 构建基站与簇头的层次结构, 完成簇头节点与基站的时间同步。(3) 在垂直同步阶段, 每个簇头节点完成与它的簇成员节点的时间同步。

#### 4.1 Leach 分簇算法

簇头节点相比簇内节点需要消耗大量的能量, 分布式轮询成簇算法有利于将高能耗成本均匀分摊到所有传感器节点。

在此阶段, 每个节点都要决定是否充当本轮的簇头节点。这个决定主要取决于整个网络中簇头节点占有节点数目的百分比 (由实际应用决定, 在初始化时设置), 并且取决于此节点已担当簇头节点的次数。具体的选择办法是: 一个传感器节点随机选择 0 和 1 之间一个值, 如果这个值小于阈值  $T(n)$ , 则此节点成为簇头节点。  $T(n)$  的定义如下:

$$T(n) = \begin{cases} \frac{p}{1 - p(r \bmod \frac{1}{p})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

其中,  $p$  为网络中簇头节点所占总节点数目的百分比,  $r$  为当前的轮数,  $G$  是在前  $1/p$  轮中没有担当过簇头节点的节点集。通过这个阈值的设置, 网络中每个节点都会在前  $1/p$  轮内担当一次簇头节点, 平衡了节点的能量消耗。

确定簇头节点之后, 每个簇头节点都会向网络其他节点广播一个消息包。在广播过程中, 网络利用非持续的载波多路访问 (CSMA) MAC 协议来避免冲突, 所有簇头节点都以相同的发射功率发送消息包, 其他节点则一直处于侦听状态, 用于接收簇头节点的广播消息。每个节点都可能收到几个来自不同簇头节点的广播消息包, 节点就根据收到消息包的信号强度 (received signal strength), 选择信号最强的广播消息包的源节点作为自己的簇头节点, 加入那个分簇, 并向簇头节点报告自己加入的 JOM 消息。

当簇头节点收到来自其所有簇内节点的 JOM 消息后, 根据簇内节点的数目, 簇头节点会产生一个 TDMA 时隙表, 调度簇内节点发送数据时间。簇头节点将包含时隙表的 CBT 消息通过广播方式发送给簇内节点, 簇内节点只与自己簇头节点通信, 它根据时隙表确定的顺序将监测到的数据发送给簇头节点。

#### 4.2 水平同步

Leach 算法将网络划分成簇并且每个簇都有一个簇头后, 开始簇中节点时间同步的过程。同步过程分为两个阶段, 首先是水平同步阶段, 主要构建基站与簇头节点的层次拓扑结构,

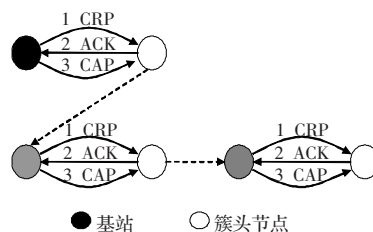


图2 水平同步阶段中的同步算法原理

完成簇头节点与基站的时间同步。其中只有簇头节点参与水平同步。

(1)网络以基站(第0层)为根节点沿着全网广播一个 ch-request 分组启动该阶段。

(2)根节点的邻居节点簇头  $i$  接收到 ch-request 这个广播分组后,就会发送 ACK 分组给基站,该分组包含发送响应分组的时间。

(3)根节点接收到响应分组后,发出一个包含自己本地时间戳的确认分组 ch-assign,任命节点  $i$  为 1 级簇头节点。

(4)节点  $i$  接收到 ch-assign 分组后,成为 1 级簇头节点。并利用修改后的 pair-wise 同步算法计算出与基站的时钟偏差,最后调整自己的时钟与基站同步。

(5)1 级簇头节点  $i$  继续沿着全网广播一个包含自己级别的 ch-request 包,重复上面基站的工作,确定新的级别的簇头节点,建立层次拓扑结构,最终网络中的每个簇头节点都同步到根节点。

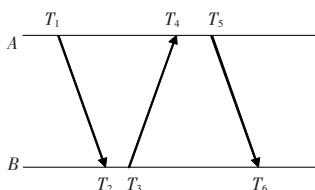


图3 相邻级别的节点时间同步原理

节点  $A$  属于第  $i$  级的簇头节点,节点  $B$  属于第  $(i+1)$  级簇头节点。 $T_1, T_4, T_5$  分别为节点  $A$  的本地时钟在不同时刻测量的时间, $T_2, T_3, T_6$  分别为节点  $B$  的本地时钟在不同时刻测量的时间。 $\Delta$  表示两个节点之间的时间偏差, $d$  表示分组的传播时延。节点  $A$  在  $T_1$  时间广播 ch-request 分组,节点  $B$  在  $T_2$  时间接收到这个分组时随机等待一段时间到  $T_3$ 。节点  $B$  在  $T_3$  时间发送携带它本地时钟的 ACK 分组,节点  $A$  在  $T_4$  时间收到这个响应分组,这里  $T_4=T_3+\Delta+d$ 。并在  $T_5$  时刻发送 ch-assign 分组,该分组包含时间信息  $T_3, T_4, T_5$ 。最后,节点  $A$  在  $T_6$  时刻收到分组,其中  $T_6=T_5-\Delta+d$ 。假设来回消息的延迟相同,可以得到:

$$\Delta = \frac{(T_4 - T_3) - (T_6 - T_5)}{2}$$

$$d = \frac{(T_4 - T_3) - (T_6 - T_5)}{2}$$

节点  $B$  可以调整自己的时钟与节点  $A$  同步。

### 4.3 垂直同步

在完成了基站和簇头同步后,开始了启动垂直同步阶段。在这个阶段主要完成簇头节点与簇内节点的同步。

(1)对于某个簇,簇头节点在时刻  $T_1$  广播同步信息 time\_sync,所有邻居节点用本地时间记录接收到这个数据包的时间,并随机指定其中一个节点  $j$  为响应节点。

(2)响应节点记录下 time\_sync 消息到达的时刻  $T_2$ ,并在  $T_3$  时刻返回回答消息。簇头节点在  $T_4$  时刻收到应答消息后,利用传统的双向消息成对方法计算出响应节点与簇头节点的时间偏差  $\delta$ 。

(3)簇头再广播一个同步消息 time\_bag,记录了响应节点接收消息时间  $T_2$  和与簇头的时间偏差  $\delta$ 。簇内其他节点接收到该消息后,比较自己接收 time\_sync 消息的时间  $T_{2k}$  和响应节点接收的时间  $T_2$ ,得到与簇头的时间偏差  $\delta' = T_{2k} - T_2 + \delta$ 。

(4)于是响应节点  $j$  对应的时间对为:  $(T_1 + \delta, T_2)$

非响应节点  $k$  对应的时间对为:  $(T_1 + \delta', T_{2k})$

假设簇头节点的本地时间  $T_{Head}$ ,簇内节点的本地时间为  $T_{CM}$ ,则  $(T_{Head}, T_{CM})$  有如下关系:

$$T_{CM} = aT_{Head} + b \tag{1}$$

其中  $a$  表示时间漂移率, $b$  表示时间偏移。可以看出  $T_{Head}$  和  $T_{CM}$  之间是确定的线性关系。这样可以采用最小二乘法线性回归估计式(1)中的参数  $a$  和  $b$ ,从而对节点的时间漂移和时间偏移进行补偿,来提高同步精度。

重复步骤(1)~(4),这样簇内每个节点得到 4 组时间对,通过线性拟合的方法,估计出参数  $a$  和  $b$  的值,调整其时钟与簇头节点同步。

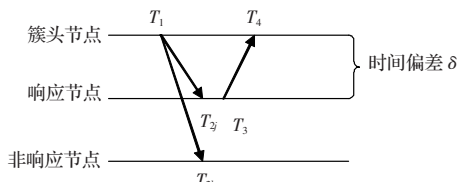


图4 垂直阶段的同步算法原理

## 5 仿真结果

CBTS 算法中结合了 TPSN 协议和参考广播同步算法思想,并且与 TPSN 协议一样都是基于层次结构。因此为了验证 CBTS 算法的优劣性,仿真时在 NS2 模拟器上将 CBTS 与 TPSN 协议的性能进行详细对比。

仿真场景布置如下:在  $(100 \times 100) \text{ m}^2$  的区域范围内,随机分布 100 个静态节点,汇聚节点远离监测区域,坐标为  $(50, 175)$ 。节点使用双向天线,天线的传输范围是 20 m,通讯带宽是 250 kb/s,无线传输为双向无差错通信。

无线传感器网络中,节点的能量资源、计算能力和带宽等资源都非常有限,衡量时间同步算法的性能主要从两个方面进行。一是同步误差,是指传感器节点相对于参考标准时间的差量,同步误差越小,同步精度越大。二是能量消耗,其中节点的通信开销是能量消耗的主要部分。在相同的条件下,消息包的传输量越多,同步开销越大。

图 5 是在节点个数为 100,节点的无线通信距离是 20 m 的情况下,CBTS 和 TPSN 两种同步算法的同步精度比较图。从图中可以看出,在相同的条件下,CBTS 算法的同步精度要比 TPSN 高一些。

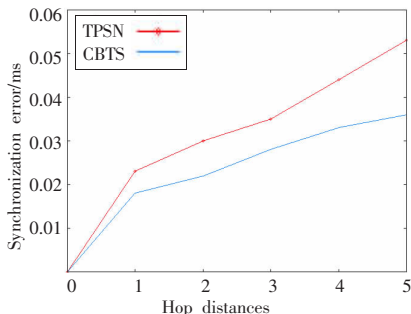


图5 CBTS 算法和 TPSN 的平均同步精度

图 6 是节点个数分别为 100、150、200、250、300 情况下,对 CBTS 算法和 TPSN 进行仿真,得到的消息传输量比较图。从图中可以看出,随着节点数目的增加,CBTS 算法的消息交换明显