

基于分簇的低功耗多跳无线传感器网络层次时间同步算法

王玉秀*, 黄 剑, 石 欣, 王小刚

(重庆大学 自动化学院, 重庆 400044)

(*通信作者电子邮箱 wangyuxiaozhao@163.com)

摘 要:针对典型时间同步算法在应用于多跳无线传感器网络(WSN)时主要集中于提高网络同步精度而忽略能耗、路径跳数、误差累积等问题,提出一种基于分簇的低功耗多跳 WSN 层次时间同步算法。该算法基于分簇的层次型网络结构,选取某一簇内节点与簇头进行双向同步,其邻居节点通过被动监听方式间接完成同步,减少了数据包传输数量,且数字签名方式保证了消息传输的安全性;引用了同步分组延迟的周期更新系数,以进一步减少报文开销。仿真结果表明,算法有效地降低了能量消耗,提高了网络寿命,具有一定的实用性。

关键词:无线传感器网络;时间同步;双向同步;低能耗;周期更新系数

中图分类号: TP926;TP393 **文献标志码:** A

Cluster-based and energy-efficient hierarchical time synchronization algorithm for multi-hop wireless sensor network

WANG Yuxiu*, HUANG Jian, SHI Xin, WANG Xiaogang

(School of Automation, Chongqing University, Chongqing 400044, China)

Abstract: Concerning the problem that typical time synchronous algorithms used in multi-hop Wireless Sensor Network (WSN) mainly focus on increasing the network synchronization precision, but ignore such issues as the network energy, path hop and error accumulation, a cluster-based and energy-efficient hierarchical time synchronization algorithm was proposed based on a cluster hierarchical network structure. In this algorithm, a cluster node was selected to take a two-way synchronization with the cluster head, and its neighbor nodes achieved bidirectional synchronization indirectly through the passive monitoring mode. In such a way, the number of the packet's transmission was reduced while the digital signature pattern ensured the safety of transmission. In addition, the cycle update coefficient of synchronous group delay was quoted to further reduce the consumption of synchronization packets. The simulation results show that the algorithm reduces the energy consumption and prolongs the network life time, therefore this algorithm has certain practicability.

Key words: Wireless Sensor Network (WSN); time synchronization; bidirectional synchronization; low energy consumption; cycle update coefficient

0 引言

无线传感器网络(Wireless Sensor Network, WSN)是由部署在检测区域内大量的微型传感器节点通过无线电通信形成的一个多跳的自组织网络系统,它能够实时监测、感知和采集各种环境或监测对象的信息并进行处理,可广泛应用于国防军事、工业控制、环境监测、交通管理等领域^[1]。由于传感器网络的数据采集、传输和处理均具有时序性,这通常要求网络中的节点拥有相同的时间基准,因此时间同步技术是传感器网络的重要支撑技术之一,也是传感器网络实现数据融合、时分多址(Time Division Multiple Access, TDMA)调度、协同睡眠等功能的基础。由于无线传感器网络具有规模大、动态性强、资源受限等特点,一些传统的时间同步协议已不再适用于无线传感器网络的环境,如以网络时间协议(Network Time Protocol, NTP)^[2]为代表的传统互联网的时间同步协议具有较高的计算开销,而以全球定位系统(Global Positioning System, GPS)^[3]为代表的硬件时间同步方案则具有较高的成

本和功耗。针对无线传感器网络自身特点,目前已经存在的经典的时间同步算法主要可以分为三种:一是基于发送者的同步机制,如延迟测量时间同步(Delay Measurement Time Synchronization, DMTS)^[4]、泛洪时间同步协议(Flooding Time Synchronization Protocol, FTSP)^[5];二是基于发送者—接收者交互的同步机制,如TPSN(Time Synchronization Protocol for Sensor Network)^[6];三是基于接收者—接收者的时间同步机制,如参考广播同步(Reference Broadcast Synchronization, RBS)^[7]。另外还提出了两种新的时间同步技术,即萤火虫同步技术^[8]和协作同步技术^[9],因这两种同步技术存在太多的假设,故只停留在理论探索和仿真阶段。

目前单跳同步研究已趋于成熟,以上经典的时间同步算法大部分是基于单跳同步机制的,但随着无线传感器网络的发展与普遍运用,传感器节点不断缩小,整体网络规模不断变大,同步误差累积现象必然越来越严重,具体表现在过大的同步开销和随跳数的增加而造成的累积误差,而目前多跳同步算法研究又相对薄弱。针对无线传感器网络在能耗、价格和复

收稿日期:2012-08-15;修回日期:2012-10-01。

基金项目:国家科技支撑项目(2011BAJ03B13);国家科技重大专项(2009ZX07528-003-09)。

作者简介:王玉秀(1988-),女,山东菏泽人,硕士研究生,主要研究方向:无线传感器网络、智能感知;黄剑(1963-),男,重庆人,副教授,主要研究方向:智能信息处理、智能控制、智能感知;石欣(1978-),男,重庆人,讲师,主要研究方向:无线传感器网络、智能系统与控制;王小刚(1984-),男,陕西宝鸡人,博士研究生,主要研究方向:无线传感器网络、智能建筑。

杂度限制的条件下,提出一种扩大网络规模、提高无线传感器网络多跳的时间同步精度、减少网络能耗、降低网络复杂度的同步算法将具有重要的意义^[10-11]。本文在经典同步算法和同步机制的基础上,设计了一种基于分簇的低功耗多跳 WSN 层次时间同步 (Cluster-based and Energy-efficient Hierarchical Time Synchronization, CEHTS) 算法,在确保较高同步精度的前提下,改善传输机制,保证消息包在传输过程中的安全性,并有效地减少时间同步带来的能量消耗。

1 CEHTS 算法

为有效地解决能耗和精度在无线传感器网络时间同步算法中的均衡问题,CEHTS 算法在时间同步过程中,网络采用分簇的树形层次拓扑结构,簇内节点通过监听簇头节点的主动双向同步过程被动完成同步以实现全网时间同步,极大地降低了同步报文开销;同时增加了邻居列表信息选出网络内特定节点回复上层节点的时间同步信息,并充分利用网络无线信道的广播特性以减少同步分组的数量;还借鉴引用了文献[12]中提出的时间同步分组延迟的周期更新系数 (Cycle Update Coefficient, CUC),可根据传感器网络对实际同步精度的需求进行设置,不仅增加了算法的灵活性与可配置性,而且更进一步降低了节点能量消耗。

由于受到节点能量的限制,每个节点的广播范围均被严格限制在由传输功率决定的半径范围内,如图1所示,在节点 P 与 A 覆盖范围之内所有节点 (如节点 B) 均能够收到节点 P 与 A 的广播信息。假设节点 P 为根节点,节点 P 与节点 A 之间执行双向同步算法^[6],在节点 P 与 A 覆盖范围之内其他所有节点均能够被动监听一系列的同步消息,它们可以通过类似 RBS 算法的方式达到与根节点时间上的同步^[13]。

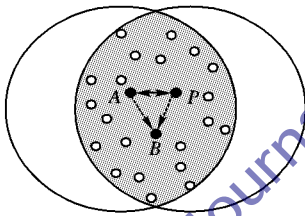


图1 CEHTS 算法单跳同步原理

为使时间同步算法更好地实现预期目标,要求网络能够实现如下前提条件:

- 1) 在物理层有完整的通信 (没有数据丢失或通信失败);
- 2) 基站或根节点必须有标准的时间作为参考,且预定为第0级;
- 3) 可以在 MAC 层打时间戳;
- 4) 网络中个别节点的失效不影响整个网络的连通性。

CEHTS 算法分为三个阶段:簇形成阶段、时间同步阶段和同步维持阶段。

1.1 簇形成阶段

如图2所示,该过程只在网络初始化时执行。参考文献[14-16]的分簇方式,首先选取根节点并将其定为第0级,由根节点广播 level_discovery 分级消息,分级消息中包含节点的 Level 值和 ID。所有接收到分级消息的节点把自己的级别定为该级别加1并把该节点 ID 和 Level 值加入邻居列表中;然后1级节点广播分级信息,接收到该分级信息的节点如果没有节点级别则把自己的级别定为1级节点广播的级别值加1,并把该节点号和级别加入邻居列表,已经有了节点级别的

节点则忽略此消息。以此类推,直到所有的节点都有自己的级别,并且每个节点均知道其所有邻居的 ID 和 Level 值。通过此过程可把所有节点分为三类:父节点、同级邻居节点及子节点。每个节点的子节点为一簇,选取同级邻居节点度数最大的节点作为该节点和所有其同级邻居节点簇头。最后,未收到 level_discovery 分级消息的节点以及新入网的节点向网络广播 level_request 分组,收到该分组的节点向其发送 level_response 分组 (包含着自己的 Level 值和 ID),接收节点比较收到的各 level_respons 分组中的 level 值后将 level 值最小的节点设为自己的父节点,并与父节点单独成簇 (如节点 R)。

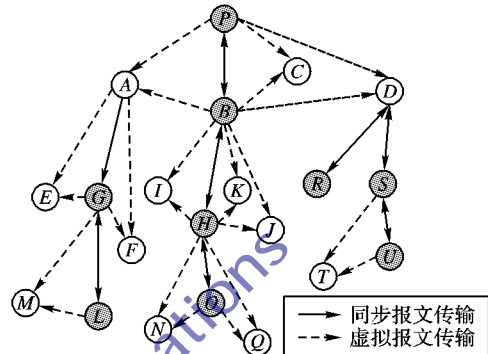


图2 基于分簇的层次网络拓扑结构

通过以上步骤生成一个基于分簇的层次型拓扑结构的网络,每个节点和它的子节点形成一个连通的单跳区域,这可将网络划分为很多单跳区域,单跳算法可以直接应用于其中。

1.2 同步阶段

在同步阶段,由于数据信息在网络通信时存在被否认、伪造、冒充、篡改的威胁而对网络的同步产生影响,本文对数据包采用基于公钥密码体制的数字签名方式以保证信息的完整性与真实性的传输。研究表明,在同样安全级别的密码体制中,椭圆曲线数字签名算法 (Elliptic Curve Digital Signature Algorithm, ECDSA) 具有密钥规模小、带宽和空间占用少、算法灵活、速度快等优点,比较适用于资源和存储空间受限的传感器网络^[17]。基于此,本文选择较成熟的 ECDSA 对消息包签名以防止外界对同步数据包的各种攻击。

层次分簇阶段完成后,根节点启动同步阶段。如图3所示,假设节点 P 为根节点或基站,节点 A 与 B 为节点 P 的子节点或下一级节点,且节点 A 为同级邻居节点度数最大的节点,为其邻居节点 B 提供同步信标,因此,节点 A 也可视为邻居节点 B 的上层节点。假设同步信息交换了 N 次,因此观察的数据包个数是 $N^{[6-7]}$ 。时间同步过程分三步进行:

1) 节点 A 在 $T_{1,i}^{(A)}$ 时刻发送利用 A 的密钥 $SK_{(A)}$ 签名的包含时间戳 $T_{1,i}^{(A)}$ 及 A 的 ID 在内的 sync_request 报文,根节点 P 在 $T_{2,i}^{(P)}$ 时刻收到该报文,同时节点 A 的邻居节点 B 在 $T_{2,i}^{(B)}$ 时刻也被动监听到此报文,根节点 P 及节点 B 利用 A 的公钥 $PK_{(A)}$ 解密 A 的消息,如解密失败,则丢掉数据包,否则根节点 P 在 $T_{3,i}^{(P)}$ 时刻向节点 A 回复利用 P 的密钥 $SK_{(P)}$ 签名的包含 $T_{1,i}^{(A)}$ 、 $T_{2,i}^{(P)}$ 、 $T_{3,i}^{(P)}$ 及根节点 P 的 ID 在内的 sync_response 报文,节点 A 在 $T_{4,i}^{(A)}$ 时刻收到该报文,同时节点 B 在 $T_{4,i}^{(B)}$ 也收到该报文信息,并且利用 P 的公钥 $PK_{(P)}$ 解密 P 的消息。

2) 如图3所示, A 与 P 之间进行了 N 次同步信息的交换,根据同步过程图, $T_{2,i}^{(P)}$ 、 $T_{4,i}^{(A)}$ 可以表达为:

$$T_{2,i}^{(P)} = T_{1,i}^{(A)} + \theta_{\text{offset}}^{(AP)} + d^{(AP)} + X_i^{(AP)} \quad (1)$$

$$T_{4,i}^{(A)} = T_{3,i}^{(P)} + \theta_{\text{offset}}^{(PA)} + d^{(PA)} + X_i^{(PA)} \quad (2)$$

其中: $\theta_{\text{offset}}^{(AP)}$ 表示节点 A 和根节点 P 之间的相偏, 且 $\theta_{\text{offset}}^{(AP)} = -\theta_{\text{offset}}^{(PA)}$; $d^{(AP)}$ 与 $X_i^{(AP)}$ 分别代表节点 A 到节点 P 的确定性传输延迟和随机性传输延迟。参考文献[18], 利用最大似然估计估计可得到节点 A 和节点 P 相偏 $\theta_{\text{offset}}^{(AP)}$ 的估计值 $\hat{\theta}_{\text{offset}}^{(AP)}$, 如式(3)所示:

$$\hat{\theta}_{\text{offset}}^{(AP)} = (\bar{U} - \bar{V})/2 \quad (3)$$

其中: \bar{U} 代表上行的测量延迟平均值, 且 $\bar{U} = \frac{1}{N} \sum_{i=1}^N [T_{2,i}^{(P)} - T_{1,i}^{(A)}]/N$; \bar{V} 代表下行的测量延迟平均值, 且 $\bar{V} = \frac{1}{N} \sum_{i=1}^N [T_{4,i}^{(A)} - T_{3,i}^{(P)}]/N$ 。在此假设节点的频偏在一定的时间内是保持不变的。

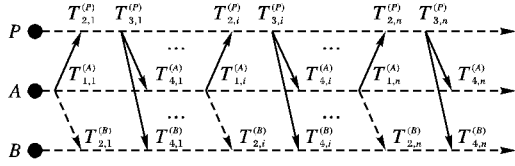


图3 节点间的时间同步过程

3) 节点 B 通过监听节点 A 的同步过程间接获得同步所需同步信息, 尽管节点 B 没有直接向基准节点 P 发送同步请求报文, 但通过监听节点 A 的同步交互, 可以模拟节点 B 向基准节点 P 发送同步请求的过程以完成同步, 且其同步过程类似于 RBS 算法的同步过程, 参考图 3 中的虚线部分, 可得到式(4)与式(5):

$$T_{2,i}^{(B)} = T_{1,i}^{(A)} + \theta_{\text{offset}}^{(AB)} + \theta_{\text{skew}}^{(AB)} (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AB)} + X_i^{(AB)} \quad (4)$$

$$T_{2,i}^{(P)} = T_{1,i}^{(A)} + \theta_{\text{offset}}^{(AP)} + \theta_{\text{skew}}^{(AP)} (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AP)} + X_i^{(AP)} \quad (5)$$

其中: $\theta_{\text{offset}}^{(AB)}$ 代表节点 A 与 B 之间的相偏, $\theta_{\text{skew}}^{(AB)}$ 与 $\theta_{\text{skew}}^{(AP)}$ 分别表示节点 A 与 B 、 A 与 P 之间的频偏, $d^{(AB)}$ 与 $X_i^{(AB)}$ 分别代表节点 A 到节点 B 的确定性传输延迟和随机性传输延迟。利用线性回归技术可以同步节点 B , 并且能够得到节点 B 与节点 P 之间的相偏 $\theta_{\text{offset}}^{(BP)}$ 与频偏 $\theta_{\text{skew}}^{(BP)}$ 的估计值, 由式(4)与(5)可得式(6):

$$T_{2,i}^{(P)} - T_{2,i}^{(B)} = \theta_{\text{offset}}^{(BP)} + \theta_{\text{skew}}^{(BP)} (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(BP)} + X_i^{(BP)} \quad (6)$$

参考文献[13,18], 通过利用最小二乘估计并经过一些简单的数学处理可得到矩阵(7):

$$\begin{bmatrix} \hat{\theta}_{\text{offset}}^{(BP)} \\ \hat{\theta}_{\text{skew}}^{(BP)} \end{bmatrix} = \frac{1}{N \sum_{i=1}^N D_i^2 - \left[\sum_{i=1}^N D_i \right]^2} \times \begin{bmatrix} \sum_{i=1}^N D_i^2 \sum_{i=1}^N x[i] - \sum_{i=1}^N D_i \sum_{i=1}^N [D_i \cdot x[i]] \\ N \sum_{i=1}^N [D_i \cdot x[i]] - \sum_{i=1}^N D_i \sum_{i=1}^N x[i] \end{bmatrix} \quad (7)$$

其中 $D_i = T_{1,i}^{(A)} - T_{1,1}^{(A)}$, 并且

$$x[i] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & T_{1,2}^{(A)} - T_{1,1}^{(A)} & \cdots & T_{1,N}^{(A)} - T_{1,1}^{(A)} \end{bmatrix}^T \begin{bmatrix} \theta_{\text{offset}}^{(BP)} \\ \theta_{\text{skew}}^{(BP)} \end{bmatrix}$$

4) 节点 A 与节点 B 分别通过式(3)与(7)完成时间同步校正。因整个网络每个节点和它的子节点可形成一个连通的单跳区域, 此单跳同步算法可以直接应用到其中, 下层节点重

复以上同步过程, 并通过相应公式完成各簇内节点的时间同步校正, 从而进一步完成整个网络的时间同步。

1.3 同步维持阶段

参考文献[12]给出的周期更新系数 (Cycle Update Coefficient, CUC), 其定义如下: 该值为不小于1的正整数, 如设该值等于 n , 则表示网络分组延迟时间 d 经过 n 个周期进行一次更新, 该值的大小会影响网络的同步精度。其中分组延迟时间 d 包括上述传输过程中的确定性延迟和随机性延迟。网络的时间同步过程完成后, 根据网络对时间同步精度的实际需求, 本算法在每个节点内设置了时间同步分组延迟的周期更新系数 CUC , 可通过设置 CUC 的大小来调节网络对同步要求的精度高低。如图4所示, 如同步周期是 T , 则分组延迟时间 d 的更新周期为 $T * CUC$, 其他子节点也据此来更新其分组延迟时间。当 $CUC > 1$ 且不需要更新分组延迟时间时, 每到一个同步周期开始时, 根节点只需单向广播上次同步时收到的消息包及本轮广播时打的时间戳即可, 根节点与不同的下层从节点只需根据式(3)与式(7)即可完成同步过程, 此时式(3)中的 \bar{U} 和 \bar{V} 中的 $T_{3,i}^{(P)}$ 与 $T_{4,i}^{(A)}$ 用本轮同步过程的上层节点的发送时间和下层节点接收时间代替。网络对同步精度要求越低, CUC 值会越大, 当 $CUC = 1$ 时网络的同步精度最高, 也是本算法所能提供的最高的同步精度。 CUC 的设置增加了算法的可配置性与灵活性, 网络可根据实际对精度要求进行调整。 CUC 在基于上述簇内节点被动监听方式减少同步分组数量的基础上进一步减少同步报文开销, 大大降低了时间同步过程的能耗。公钥密码体制数字签名方式的应用保证了消息包在传输过程中的安全性。

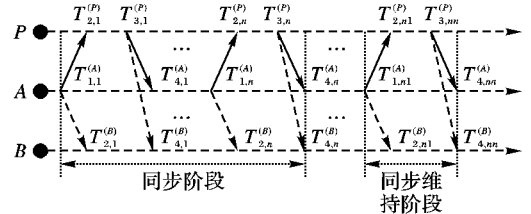


图4 同步维持阶段过程

2 CEHTS 算法性能分析

2.1 误差分析

本算法采用的基于簇的双向同步与只接收同步相结合的同步方式, 相对于传统的同步算法更具有优越性。由文献[6]可知, 报文在传输过程中产生的延迟主要有发送延迟、访问延迟、传输延迟、传播延迟、接收延迟与接收处理延迟等, 而这些延迟中有的延迟是确定的, 如传输延迟、传播延迟与接收延迟等; 有的延迟是不确定的, 如发送延迟、访问延迟与接收处理延迟等。由于节点本身存在时钟漂移而引起的时间偏差与节点在传输消息过程中产生的不确定延迟, 可导致算法在同步过程中产生一定误差。如图3所示, 节点 A 与节点 P 的同步方式为类似于 TPSN 的双向同步, 节点 B 与节点 P 的同步方式为只接收的同步方式, 节点 A 根据 CEHTS 算法在同步阶段时得到的信息, 在整个同步过程中, 节点 A 的同步误差表达为

$$Error = \hat{\theta}_{\text{offset}}^{(AP)} - D_{t4}^{A \rightarrow P} = \frac{S_{AP}}{2} + \frac{P_{AP}}{2} + \frac{R_{AP}}{2} + \frac{RD_{t41 \rightarrow t44}^{A \rightarrow P}}{2} \quad (8)$$

其中: $D_{t4}^{A \rightarrow P}$ 表示在 $t4$ 时刻 A 与 P 之间产生的同步时间偏差; S_{AP} 、 P_{AP} 、 R_{AP} 分别表示在发送端、接收端、传输过程中的不确

定性; $RD_{tA1 \rightarrow tA4}^{A \rightarrow P}$ 表示节点 A 与节点 P 在一轮同步过程中产生的时钟漂移。

由以上分析可知,节点 B 采用的同步算法类似于经典算法 RBS 算法,且本算法与 RBS 算法相比,对时钟偏差的估计也不会对同步精度产生任何损失,节点 B 的同步误差表达为

$$Error = \hat{\theta}_{offset}^{(BP)} - D_{tA}^{B \rightarrow P} = S_{BP} + P_{BP} + R_{BP} + RD_{tA1 \rightarrow tA4}^{B \rightarrow P} \quad (9)$$

其中: $D_{tA}^{B \rightarrow P}$ 表示在 tA 时刻 B 与 P 之间产生的同步时间偏差; S_{BP} 、 P_{BP} 、 R_{BP} 分别表示发送端、接收端、传输过程中的不确定性; $RD_{tA1 \rightarrow tA4}^{B \rightarrow P}$ 表示节点 B 与节点 P 在一轮同步过程中产生的时钟漂移。

2.2 同步开销分析

由以上对本算法的介绍可知,网络完成簇形成阶段之后,当 $CUC = 1$ 时,CEHTS 算法在每个同步周期只需要 $2N$ 个同步报文;当 $CUC > 1$ 时,在一定的同步周期内,各个节点的分组延迟时间是不需要进行更新的,其所需要的同步报文数为 $2N$ 或 N 。CEHTS 算法与其他经典的同步算法如 TPSN、RBS、FTSP 等在每个同步周期所需要的同步报文数如表 1 所示,其中 M 为除去参考节点外所需要同步的节点个数。

表 1 不同算法同步报文开销比较

算法	发送的报文数
RBS	$N + M(M - 1)/2$
TPSN	$2N(M - 1)$
FTSP	NL
CEHTS ($CUC = 1$)	$2N$
CEHTS ($CUC > 1$)	$2N$ 或 N

由表 1 可知,以上所有算法在每个同步周期需要发送的同步报文个数均与网络中同步节点的个数有关,而 CEHTS 算法同步报文并不依赖网络中节点的个数。随着网络中节点个数的增加,其他算法的报文开销也随之增加,而 CEHTS 算法的报文开销保持不变,相对于 RBS 算法避免了大量报文的交互和高复杂度的运算,降低了功耗;相对于 TPSN 算法本文算法不但保持了双向时间同步特征,减少了传送时延、传播时延和接收时延的影响,而且簇内节点采用被动监听的同步方法,极大地降低了同步报文开销,在保证同样同步精度的情况下更好地保证了节点能量的有效性;相对于 FTSP 算法本文算法不但避免了报文的洪泛现象,也降低了消息包传输数量;且 CEHTS 算法引进的周期更新系数在增加了算法灵活性与可配置性的同时,进一步减少了同步过程中的报文开销,降低了能量消耗。

3 仿真结果分析

由于开源的轻量级嵌入式操作系统 TinyOS 具有体积小、结构高度模块化、基于组件的架构方式、良好的跨平台性和高效性、低功耗等特点,使得它能够突破传感器节点各种苛刻的限制,可快速实现各种应用,非常适用于资源相对有限的 WSN 环境,因此本文选择基于 TinyOS 操作系统的仿真平台。

3.1 误差比较

为验证 CEHTS 算法的性能,仿真基于本文提出的分簇的层次拓扑结构,在实验平台上实现了本文算法与 RBS、TPSN、FTSP 算法,并对这四种算法获得的测试数据经过统计学方法处理后进行分析比较。在 $100\text{ m} \times 100\text{ m}$ 的方形区域内布置了 50 个节点,节点通信距离设置为 15 m,根节点布置在测试

方形区域正中央,其余节点随机分布。为使获得的四种算法的测试数据完全具有可比性,所有节点的测试数据都是在相同平台相同环境下获得的。

图 5 为编写程序运行后得到四种算法随着时间的增加误差变化量的比较,同时表 2 给出了不同算法的同步精度参数作为参考,表 2 中的数据是不同算法在本仿真环境下通过多次仿真实验获得并经过统计学方法处理后计算出来的平均值。由图 5 和表 2 可知,当 $CUC = 1$ 时,CEHTS 算法的同步精度与 TPSN 算法的同步精度相比相差不大,但 CEHTS 算法的同步误差与 RBS、FTSP 算法的同步误差相比,其变化范围较小,平均同步精度较高,这证实了本文提出的同步算法有效缩小了时间偏差的误差范围,提高了同步精度,得到了比较精确的时间偏差值,能够满足对时间精度要求较高的场合。

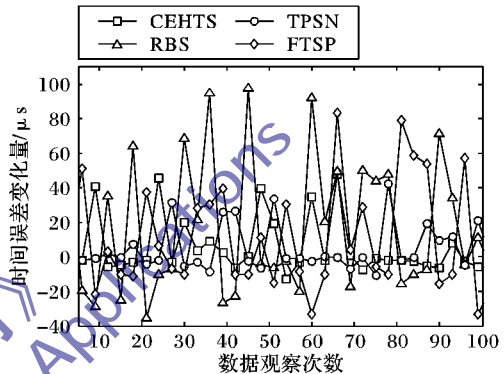


图 5 不同算法同步误差比较

表 2 不同算法同步精度参数比较 μs

同步算法	同步误差均值	同步误差标准差	最大同步误差	最小同步误差
RBS	38.15	22.85	98.83	0
TPSN	17.82	16.29	42.94	0
FTSP	27.54	21.47	83.86	0
CEHTS	18.47	16.86	47.25	0

接下来验证参数 CUC 在本文设定的仿真环境下对 CEHTS 算法精度的影响。如图 6, 当 $CUC = 1$ 时的同步精度最高,当 $CUC = 5$ 时 CEHTS 的平均同步精度有所下降,这证明了报文在传输过程中产生的延迟随时间的推移有所变化,并且传输延迟受时钟漂移和外界环境的影响,随着 CUC 值的增大,CEHTS 算法的同步精度会有所降低,此结论与文献 [12] 结论相一致,并且在不同的网络环境下, CUC 的值可不同程度地影响到网络的同步精度,网络可根据实际场合对同步精度的要求进行调整。

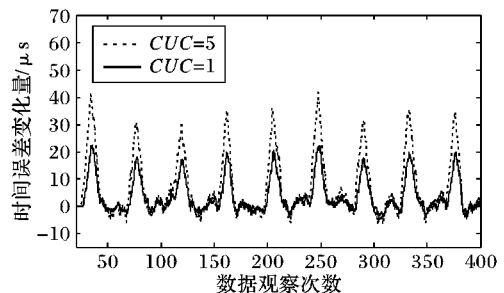


图 6 CUC 不同时的 CEHTS 算法与 RBS 算法的误差比较

3.2 同步开销比较

为验证本文算法在网络中的报文开销,图 7 给出了 RBS、TPSN、FTSP、CEHTS 算法在单跳情况下发送的同步数据包个

数与节点数 M 、广播数据包包次数 N 之间的关系的三维仿真图。假设所有增加的节点都在父节点和邻居节点的广播范围之内,节点数 M 的取值范围是 $0 \sim 50$,广播数据包包次数 N 的取值范围是 $0 \sim 20$ 。由图可知,随着节点的增加,RBS、TPSN、FTSP 算法的报文开销分别有不同程度的增加,而 CEHTS 算法并没有明显的变化,这说明了 CEHTS 算法的同步数据包个数与网络中的节点的个数并没有明显的直接关系,而 CEHTS 算法的报文开销会随着广播数据包包次数的增加呈线性增加。

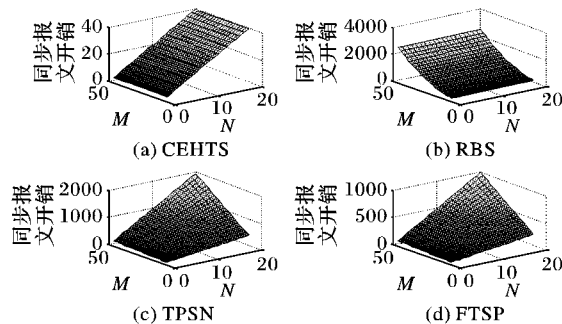


图7 不同算法的同步报文开销与 N 、 M 的关系

图8给出了不同算法在多跳网络中的平均同步报文开销随着节点数增加的变化趋势,假设广播消息包的次数固定为 $N = 20$ ^[6],节点数 M 的取值范围是 $0 \sim 50$ 。从图可以看出,随着节点数的增多, $CUC = 1$ 时,CEHTS 的报文开销明显小于 RBS、TPSN 与 FTSP 的报文开销,节点越多时这种优势越明显,进一步证实了本文算法的低能耗性,且适于用在多跳的网络环境中。

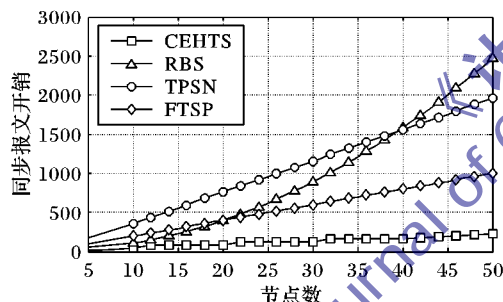


图8 不同算法节点数与报文开销关系曲线

4 结语

本文针对现有的典型同步机制存在的不足,充分利用无线传感器网络中节点信息传输的特点,把直接双向同步机制与间接双向同步机制相融合,结合分簇的层次型网络拓扑结构,引入同步分组延迟的周期更新系数,提出了一种新的开销的无线传感器网络时间同步算法。该算法只需要对传感器节点以双向同步方式进行同步,其他节点利用被动监听的方式以达到一个更广范围节点同步的目的,基于公钥密码体制数字签名方式的应用保证了数据包的安全性,周期更新系数的设置在增加网络实用性和可调性的同时进一步降低了网络能量的消耗。对 CEHTS 算法进行的性能分析和实验验证表明该算法具有较高的同步精度和低同步分组开销的特点,很适用于资源受限的多跳无线传感器网络中。而对于这种传输机制及主要理念可否全部或部分地应用到目前存在的其他经典的协议或新设计的协议中,并且随着环境的变化,此算法是否受到很大影响等问题,还需进一步研究。

参考文献:

[1] 王殊,阎毓杰,胡富平,等. 无线传感器网络的理论及应用[M].

北京:北京航空航天大学出版社,2007.

- [2] MILLS D L. Network time protocol (Version3) specification, implementation and analysis [S]. Newark: University of Delaware, DARPA Network Working Group, 1992.
- [3] ELSON J, ESTRIN D. Time synchronization for wireless sensor networks [C]// The 2001 International Parallel and Distributed Processing Symposium, Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing. San Francisco: [s. n.], 2001: 1965 - 1970.
- [4] PING S. Delay measurement time synchronization for wireless sensor networks, IRB-TR-03-013 [R]. Intel Research Centre, 2003.
- [5] MARÓTIM M, KUSY B, SIMON G, *et al.* The flooding time synchronization protocol [C]// SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems. New York: ACM, 2004: 39 - 49.
- [6] GANERIVAL S, KUMAR R, SRIVASTAVA M B. Timing-synch protocol for sensor networks [C]// SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems. New York: ACM, 2003: 138 - 149.
- [7] SICHITIU M L, VEERARATTIPHAN C. Simple, accurate time synchronization for wireless sensor networks [C]// WCNC 2003: IEEE Wireless Communications and Networking Conference. Piscataway: IEEE, 2003: 1266 - 1273.
- [8] HONG Y-W, SCAGLIONE A. A scalable synchronization protocol for large scale sensor networks and its applications [J]. IEEE Journal on Selected Areas in Communications, 2005, 23(5): 1085 - 1099.
- [9] KROHN A, BEIGL M, DECKER C, *et al.* Syncob: collaborative time synchronization in wireless sensor networks [C]// INSS '07: Fourth International Conference on Networked Sensing Systems. Piscataway: IEEE, 2007: 283 - 290.
- [10] 阳韬. 无线传感器网络时间同步算法的研究[J]. 计算机工程, 2011, 37(23): 116 - 118.
- [11] 湛华金,何佳洲. 一种基于数据链路层的局域网时间同步新机制[J]. 计算机应用, 2010, 30(5): 1183 - 1187.
- [12] 吴成伟,黄文君. 无线传感器网络比对广播时间同步算法[J]. 传感技术学报, 2009, 22(12): 1789 - 1794.
- [13] NOH K-L, SERPEDIN E, QARAQE K. A new approach for time synchronization in wireless sensor networks: pairwise broadcast synchronization [J]. IEEE Transactions on Wireless Communications, 2008, 7(9): 3318 - 3322.
- [14] 汪付强,曾鹏,于海斌. 一种低开销的双向时间同步算法[J]. 仪器仪表学报, 2011, 32(6): 1357 - 1363.
- [15] GAUTAM G C, SHARMA T P, KATYAR V, *et al.* Time synchronization protocol for wireless sensor networks using clustering [C]// ICRTIT 2011: IEEE International conference on Recent Trends in Information Technology. Piscataway: IEEE, 2011: 417 - 422.
- [16] JIANG C J, XIANG M, SHI W R. Overview of cluster-based routing protocols in wireless sensor networks [C]// ICEICE 2011: International Conference on Electric Information and Control Engineering. Piscataway: IEEE, 2011: 3414 - 3417.
- [17] JOHANSON D, MENEZES A, VANSTONE S. The Elliptic Curve Digital Signature Algorithm (ECDSA) [J]. International Journal on Information Security, 2001, 1(1): 36 - 63.
- [18] 苏淳. 概率论[M]. 北京: 科学出版社, 2010.