基于最大互邻集合的无线传感器网络单向时间同步*

黄 晓，罗树浩
（中山大学信息科学与技术学院，广东 广州 510006）

摘 要：时间同步是无线传感器网络的一项关键技术。针对目前时间同步算法能耗较大等问题，通过单向同步技术建立全网同步数学模型，提出一种基于最大互邻集合的同步算法。在层次发现阶段生成同层节点的最大互邻集合，利用有限的消息交互分布式地保留尽量少的广播节点，并加入子节点注册、低层节点监听和时序控制等策略提高算法效率。在 NS2 软件平台进行了仿真，并与相关文献算法对比，结果表明所提出的算法在达到相同同步精度前提下，可显著降低同步阶段的消息开销，提高成功同步节点比例。

关键词：无线传感器网络；时间同步；单向同步；最大互邻集合
中图分类号：TP393 **文献标志码**：A **文章编号**：0529－6579（2014）03－0001－07

Max Adjacent Set-based One-way Time Synchronization in Wireless Sensor Networks

HUANG Xiao, *LUO Shuhao*
(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510000, China)

Abstract: Time Synchronization is a critical issue in Wireless Sensor Networks. The energy consumption problem that exists in current synchronization algorithms is studied. A network-wide mathematical model is established with one-way synchronization technique, and a synchronization algorithm based on max adjacent set is proposed. During the level discovery phase, the max adjacent set of the same level nodes is generated. Nodes communicate with each other through few messages to reserve as less broadcast nodes as possible. Furthermore, some strategies such as child node registering, lower nodes overhearing and time sequence control are used to improve the efficiency of the algorithm. The simulation on NS2 platform and comparison to related work confirm that the algorithm can significantly reduce the message consumption during synchronization phase, and improve the ratio of nodes which achieve synchronization successfully.
Key words: wireless sensor network; time synchronization; one-way synchronization; max adjacent set

在无线传感器网络（Wireless Sensor Networks，WSNs）[1]中，节点搭载各种传感器，通过无线传输将监测的数据传输到汇聚点（sinks）。无线传感器网络的节点具有体积小、低能耗、低成本和自组织等特点，在环境监测、工业自动化以及人员监控管理等领域得到广泛应用[2]。

时间同步（Time Synchronization）在无线传感器网络的应用中有关键的作用[3]。当观测到事件发生时，多个节点需要将事件的时间以及空间信息进行数据融合（Data Fusioning），必须维护一个与全网一致的时钟。另一方面，精确的时间同步可降低网络的能耗，例如在周期性工作的网络中，所有节点在空闲周期同时关闭接收机进入休眠状态，等待工作周期到来后唤醒，进行数据的传输。此外，

* **收稿日期**：2013－09－01
基金项目：国家自然科学基金资助项目（61201087）；广东省科技计划资助项目（2012B010100022）；中山大学信息科技国家级实验教学示范中心实验教学研究资助项目
作者简介：黄晓（1968 年生），女；**研究方向**：ZigBee 无线传感器网络，物联网；E-mail：huangx@ mail. sysu. edu. cn

在应用信标（Beacon）机制的传感器网络中，以及采用TDMA（Time Division Multiple Access）等协议避免碰撞，都需要精确的时间同步。

无线传感器网络时间同步的首要目标是最小化同步误差。但是由于节点存在能量、运算以及通信能力方面的局限性，时间同步必须能量高效（Energy Efficient）。实际上，消息传输所需要的能量开销远大于运算所需开销，所以时间同步过程中应尽量减少同步消息的传递，同时可保证其他数据的有效传输。

1 研究现状

时间同步问题早年在分布式系统已有相关研究[4]。随着无线传感器网络的发展，针对其所进行的同步研究也有丰富的成果。根据同步过程的差异，将现有无线传感器网络时间同步算法划分为集中式和分布式两类。

集中式同步算法，是指在网络中存在一个根节点（root），向全网提供一个标准的时钟，并发起同步流程。RBS（Referenced Broadcast Synchronization）是经典的集中式同步算法[5]，通过同层节点交换时间戳的方法抵消发送时延，达到较高同步精度，但难以在多跳网络中应用。TPSN（Timing-sync Protocol for Sensor Networks）将同步分为两个阶段[6]，层次发现阶段每个节点获取层次属性，同步阶段每个节点都与父节点进行双向的消息交互从而同步，因此消息开销大。为解决开销问题，Kyoung-Lae等[7]提出了成对广播同步，一对节点双向交换消息同步，而其共同邻居能监听到这些消息并完成同步。King-Yip等[8]将成对广播同步应用到多跳网络中，采用分布式的方法并结合贪婪算法动态选择同步对，大大减少了同步过程中消息开销。为提高鲁棒性，RTSP（Recursive Time Synchronization Protocol）算法实现根节点的动态选举[9]，通过下层节点迭代式向上层请求同步，但延长了同步过程从而增加了误差。

相反地，分布式同步算法不存在根节点，同步在各个节点间异步进行，整个网络收敛于虚拟全局时钟。萤火虫同步算法结合了生物现象中的激发模型[10]，节点通过观测脉冲信号以及调整脉冲信号发射间隔，与周围节点保持同步。ATS（Average TimeSync）算法用随机矩阵方程的角度来解决同步问题[11]，但仅适用于网络规模较小的情况。一致时钟同步算法（Consensus Clock Synchronization）使用置信赋权平均方法对邻居时钟戳赋权重从而补偿本地时钟[12]，算法的收敛速度较快。RGCS（Robust Gradient Clock Synchronization）算法通过滑动平均赋权的方法[13]，更新本地的时钟，算法的鲁棒性较好。

分布式算法普遍存在着收敛时间长、适用网络规模受限等缺点，集中式算法虽然在应对网络动态变化（如根节点失效）方面稍显不足，但同步精度高、收敛快速。本文研究了集中式同步算法，在层次发现阶段，采用基于最大互邻集合的分布式策略获取精简的同步集，降低节点能耗和网络开销。在硬件平台和软件仿真平台进行的实验，证明了算法的有效性。

2 单向时间同步模型

2.1 单向时间同步和双向时间同步

节点同步时，由作为主节点的某个邻居提供所需的时间戳信息。单向同步是相对于双向同步而言的，节点不必进行双向的消息交互，只需接收主节点的时间戳即完成同步，消息单向传播。一般的同步算法（如TSPN）采用双向同步策略[6]，节点A发送同步消息S_1给B，而B回复S_2给A，同步消息中携带相关时间戳，过程如图1（a）所示。节点A获取四个时间戳，通过（1）式可补偿相位偏移Δ，利用线性回归等方法补偿速率漂移。

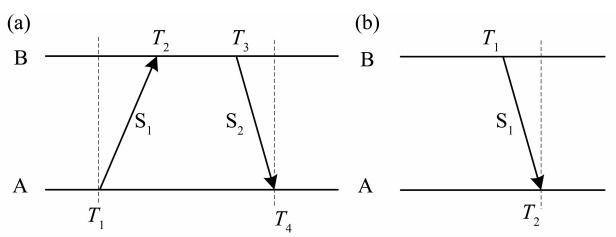

图1 双向同步（a）和单向同步（b）

Fig. 1 Two-way synchronization (a) and one-way synchronization (b)

$$\Delta = [(T_2 - T_1) - (T_4 - T_3)]/2 \quad (1)$$

文献［14］提出了一种新的时间戳标记方法，如图2所示。B生成同步消息并打开发射机，在发送帧首定界符SFD完毕的瞬间，硬件系统捕捉硬件时钟的值T_1并保存在寄存器中，此时消息其他字段正在传输。B立即将T_1嵌入消息末端，通过当前消息发送出去。A在收到消息的SFD时，也捕捉到本地时间戳T_2，并嵌入到消息的末端，交给上层处理。A由（2）式即可完成相位偏移补偿，过程见图1（b）。

$$\Delta = T_2 - T_1 \quad (2)$$

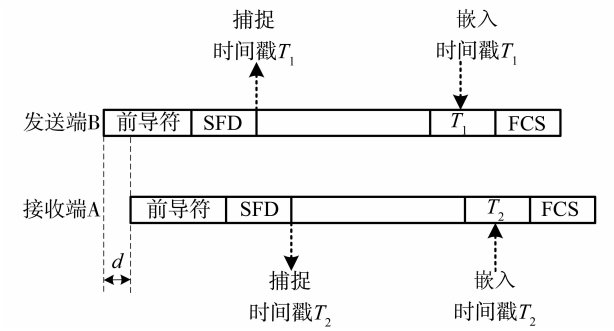

图 2　单向同步原理

Fig. 2　Principle of one-way synchronization

这种时间戳标记方法脱离了协议栈的层次结构，由硬件系统独立完成，因此比 MAC 层标记方法的精度更高。单向同步过程引入了消息在空气中的传播时延 d，由此带来的误差可忽略[6]，故认为可取得与双向同步相同的精度。另外，节点只需广播单个时间戳即可实现多个邻节点的同步，显著降低了同步通信开销。

2.2　单向时间同步数学模型

由于单向同步可达到双向同步的精度，而且具有降低消息开销、节省同步节点能耗的优势，故本文选择单向同步作为研究对象。同时由于无线传感器网络的能量资源少，运算能力弱，故本文算法的目标为降低能量开销，降低算法复杂度，提高同步成功率。

同步分成两个阶段。第一阶段为层次发现阶段，寻找包括根节点在内的所有广播节点组成的同步集，同时控制同步集节点数量以降低同步消息开销。第二阶段为同步阶段，同步消息自根节点起经同步集节点向外传播，其余节点监听，实现全网同步。本文使用单向同步技术实现全网同步，建立寻找最小同步集的数学模型。

设节点集合为 Ψ，其元素数量为 $\| \Psi \| = N$。对于任意两个节点 $\forall \psi_i, \psi_j \in \Psi$，定义其连通性 $c_{\psi_i\psi_j}$

$$c_{\psi_i\psi_j} = \begin{cases} 1, \psi_i, \psi_j \text{ 互为邻居} \\ 0, \text{其他} \end{cases} \tag{3}$$

设同步集为 $\Omega \subseteq \Psi$，根节点属于 Ω。Ω 中每个节点都有一个层次属性，表示与根节点的最短拓扑距离，根节点的层次规定为 0。对于任意两个节点 $\forall \varphi_p, \varphi_q \in \Omega$，定义变量 $\lambda_{\varphi_p\varphi_q}$ 表示两者的层次关系

$$\lambda_{\varphi_p\varphi_q} = \begin{cases} 1, \varphi_p \text{ 是 } \varphi_q \text{ 的上层节点} \\ 0, \text{其他} \end{cases} \tag{4}$$

对于非根节点的 $\forall \varphi_q \in \Omega$，$\Omega$ 至少存在一个节点与其互为邻居且是其上层节点，从而同步消息可由根节点开始沿广播节点向外传播

$$\sum_{p=1}^{\| \Omega \|} c_{\varphi_p\varphi_q} \cdot \lambda_{\varphi_p\varphi_q} \geqslant 1, \forall \varphi_q \in \Omega, \varphi_q \neq 0, p \neq q \tag{5}$$

为确保网络中所有节点可监听到同步消息，须保证每个节点至少与一个广播节点互为邻居，即

$$\sum_{p=1}^{\| \Omega \|} c_{\psi_i\varphi_p} \geqslant 1, \forall \psi_i \in \Psi, \psi_i \neq \varphi_p \tag{6}$$

目标是求取符合上述条件的具有最小元素数量的同步集 Ω

$$\text{argmin}_{\Omega} \| \Omega \| \tag{7}$$

将网络看作一个无向连通图，寻找最小同步集的问题等价于构造一个以根节点为树根的非叶子节点最少的生成树，即最多叶子生成树（MLST，Maximum Leaf Spanning Tree）[15]，是一个 NP-HARD 问题，可利用集中式优化算法求解。集中式算法通过节点把拓扑连接信息传输到汇聚节点，求取最优解，再将结果分发给每个节点。这种方法对网络链路造成较大负荷，当传输过程发生消息丢失时会造成结果错误。本文研究了更具实用意义的分布式算法，节点通过有限数量的消息交互，判断自身是否广播节点。

3　基于最大互邻集合的单向同步

可相互传输消息的两个节点之间存在无线通信链路，而距离对链路通信质量存在影响。节点接收消息时，采样到对应无线链路的 LQI（Link Quality Indicator）参数，例如 ZigBee 采用接收信号强度指示值（Receive Signal Strength Indicator，RSSI）作为 LQI。在理想情况下，当两个节点之间没有障碍物时，LQI 值随距离的增大而减小；当障碍物存在时，以节点为中心向障碍物延伸的方向上，LQI 值虽呈规律性变化，但距离越远 LQI 值越小的趋势依然成立。因此，在实际环境中节点可通过消息的 LQI 值来获知邻居节的大概空间信息。

单向同步算法分为两个阶段，本文主要详述层次发现阶段，流程如图 3 所示。

Step 1　根节点启动同步流程，广播一跳层次发现帧 P_ DISCOVERY。在层次发现过程中的所有控制和数据帧都包括以下几个公共域：

MsgTpye	Seq	SourceID	SourceLevel

其中，MsgTpye 为帧类型；Seq 为层次发现帧序列号，根节点可增大 Seq 发起新一轮层次发现；

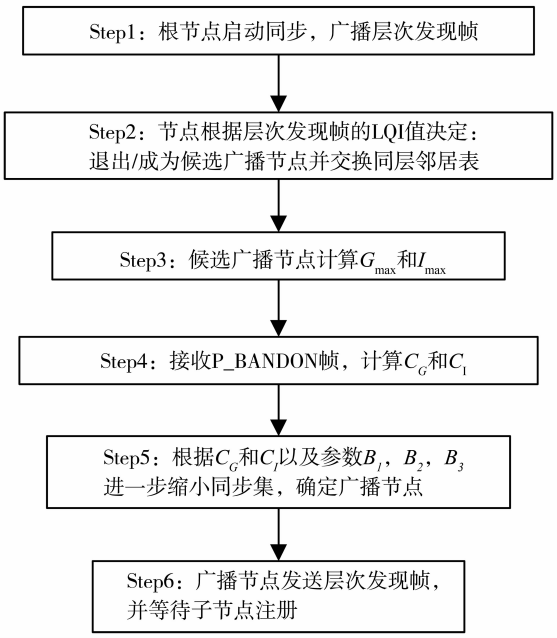

图 3 单向同步算法流程图

Fig. 3 Flow diagram of one-way synchronization algorithm

SourceID 为源节点的 ID 号，网络中每个节点都有唯一的 ID，根节点为 0；SourceLevel 是源节点的层次，根节点为 0，子节点的层次为父节点加 1。此外，节点保存一个状态参数来指示工作状态，初始时为 S_ INIT。

根节点置状态为 S_ FINISH，表示层次发现阶段结束。

Step 2 节点接收到 $i-1$ 层节点的 P_ DISCOVERY 帧，若状态为 S_ INIT 则接收该帧。节点收到多个帧时，选择最小的 ID 作为父节点并记录其 ID，并把自身的层次修改为 i。节点根据帧的 LQI 值判断：

1）LQI > 阈值 L_1，表明节点离其父节点较接近，则置状态为 S_ FINISH，完成层次发现并退出；

2）LQI < = L_1，表明节点处于父节点广播范围的边缘处，则置节点状态为 S_ EXCH_ NEIGHBOR，成为第 i 层候选广播节点，广播 P_ EXCH_ NEIGHBOR 帧，帧中携带一跳邻居信息。节点的邻居信息在网络层周期性的邻居扫描可获取，不需要额外的同步消息开销。

候选广播节点同时接收同层邻居节点的 P_ EXCH_ NEIGHBOR 帧，等待一段时间确保接收完毕。通过准确的时序控制，可确保同一层次的所有节点在同一时间段内进行同步。候选广播节点保存同层邻居集合 N_s 的邻居表及 LQI 值。

Step 3 第 i 层候选广播节点在同层邻居集合 N_s 中计算其最大互邻集合 G_{max} 和最近邻集合 I_{max}。

定义节点互邻集合 G，G 中任意节点 n_p 和 n_q 互为邻居；则节点的最大互邻集合 G_{max} 为其所在 K 个互邻集合中元素数量最多的一个，即

$$G_{max} = \mathrm{argmin}_{G_k} \| G_k \|, 1 \leqslant k \leqslant K \tag{8}$$

定义节点最近邻集合 I_{max}，表示同层邻居集合 N_s 中 LQI 值大于阈值 L_2 的所有节点的集合，即

$$I_{max} = \bigcup_{\forall n_p \in \mathbf{N}_s, LQI_{n_p} > L_2} (n_p) \tag{9}$$

G_{max} 和 I_{max} 由候选广播节点遍历搜索 N_s 的邻居表获得。为降低搜索过程的运算量，可将 N_s 按其 ID 号的大小排序后存储。对排序后的 N_s 搜索 G_{max}，运算复杂度为 O（$n\log n$），其中 n 为 N_s 的元素数量且值较小，故运算可在资源受限的传感器节点上进行。获取 I_{max} 只需搜索 N_s 的 LQI 值即可，运算复杂度为 O（n）。另外初始化计数值 $C_G=0$ 和 $C_I=0$。

Step 4 候选广播节点退避等待随机时间 t_{rand}，节点状态修改为 S_ BACKOFF，在此期间接收 N_s 中节点发送的 P_ ABANDON 帧。这些邻居节点称作放弃节点，说明该节点放弃成为广播节点，在同步阶段只作监听，不发送同步消息。候选广播节点将放弃节点 ID 写入集合 $S_{abandon}$，并判断：

1）若放弃节点 $n_A \in G_{max}$，则使计数值 $C_G = C_G + 1$，即更新记录最大互邻集合中放弃节点的数量；

2）若放弃节点 $n_A \in I_{max}$，则使计数值 $C_I = C_I + 1$，更新记录最近邻集合中放弃节点的数量。

Step 5 候选广播节点随机等待时间 t_{rand} 到达，分析是否成为广播节点。

在最大互邻集合 G_{max} 中，任意节点广播消息集合内其余节点都能监听到，因此 G_{max} 中的所有节点处于一个相对较小的空间范围内。若 G_{max} 中均为广播节点，则同步时广播的同步消息覆盖范围有较大面积的重叠，造成不必要的能量和网络开销。实际上，只需在 G_{max} 中选取部分广播节点，并使其分布合理，那么同步覆盖范围约等于 G_{max} 中所有节点的同步覆盖范围。

候选广播节点作以下判断：

1）在 t_{rand} 内收到 G_{max} 中所计数的放弃节点数量为 C_G，若 $C_G < \| G_{max} \| - B_1$，在 G_{max} 中搜索，若存在未放弃的节点，且该节点为最近邻集合 I_{max} 中节点，即：

$$(G_{max} - S_{abandon}) \cap I_{max} \neq \phi \tag{10}$$

则放弃成为广播节点，广播 P_ ABANDON 帧通知邻居节点；若上述条件不满足，则进入 2）。

2）在 t_{rand} 内收到 $I_{\max}$ 中放弃节点数量为 C_I，若 $C_I < \| I_{\max} \| - B_2$，即 $I_{\max}$ 中存在过多候选节点，则放弃成为广播节点，同样广播 P_ ABANDON 帧；否则，进入 3）。这个条件保证了 $I_{\max}$ 中保留不多于 B_2 个同步节点，因为集合内的节点进行同步广播所覆盖的有效面积会有大部分重叠，这是不必要的。

3）若 $C_G < \| G_{\max} \| - B_3$，说明 $G_{\max}$ 中潜在的广播节点数量超过最大值 B_3，强制放弃成为广播节点，广播 P_ ABANDON 帧。否则，成为同步集中的广播节点。为了控制同层节点在同一时间段内进行层次发现，保证算法流程准确，加入一个时序控制策略。广播节点在此前退避了随机的周期 t_{rand} 监听同层邻居节点的 P_ ABANDON 帧，变量 t_{rand} 由其本地保存；同时节点希望与同层的广播节点在尽量接近的时刻广播 P_ DISCOVERY 帧告知下层节点。因此，设定每层的层次发现过程的执行周期为 T，则广播节点在分析完毕后应该等待时间（$T - t_{\text{rand}}$）后，再广播 P_ DISCOVERY 帧，进入 Step 6。

同层候选广播节点邻居收到 P_ ABANDON 帧，按 Step 4 处理。上述的 B_1，B_2，B_3 为可变参数，根据实际网络情况调整其值大小使同步达到最佳状况。

Step 6　第 i 层候选广播节点广播 P_ DISCOVERY 帧，置状态为 S_ WAIT_ REGISTER，此时并未最终成为广播节点。节点开启定时器，在设定时间内等待子节点注册。状态若为 S_ INIT 的邻居节点可能收到多个 P_ DISCOVERY 帧，选择 ID 最小的父节点，进入层次发现流程 Step 2，并立即单播一个注册帧 P_ REGISTER 给父节点。当父节点收到了该帧，正式成为广播节点。这个策略防止了子节点有多个潜在的父节点时，造成某些广播节点“悬空”，即实际上没有子节点的情况。

另外，第 i 层放弃节点 A 广播 P_ ABANDON 帧，状态为 S_ INIT 的邻居 B 收到后，将 A 的 ID 存储起来，并开启定时器等待 t_{abandon} 时间。若在此期间，B 没有收到任何的 P_ DISCOVERY 帧，说明了自身在层次发现阶段被遗漏了。此时 B 发送 P_ INFORM 帧告知该放弃节点 A，A 重新成为广播节点，发送 P_ DISCOVERY 帧。

在实际场景中，网络边缘可能存在少量的节点，其仅有的邻居在 Step 2 的 LQI 阈值判断时放弃成为广播节点，导致这些节点被遗漏从而同步失败。这种极端的情况，可应用简单的策略解决。具体在同步阶段，被遗漏节点经过长时间没有监听到同步消息，确认同步失败，向邻居节点提出同步请求，等待其回复同步消息即可。

4　实验与仿真

4.1　单双向同步对比

为验证单向同步策略的同步精度，本文进行了相关的实验，并与双向同步技术对比。实验在 CC2530 为核心的 ZigBee 平台上进行，协议栈为 TI 的 Z-Stack2.5，利用两个节点进行单跳的时间同步。单双向同步各进行 8 次实验，并使用双踪示波器测量同步误差，结果见表 1。硬件实验数据表明，单向同步的平均同步误差约为 12.5 μs，双向同步约为 11 μs。由此可知利用时间戳捕捉技术进行单向同步，可达到与双向同步一致的精度，而结果的差异来源于同步时引入的系统随机误差，并非同步方法所导致。

表 1　单向与双向同步误差对比

Table 1　Errors of one-way and two-way synchronization

μs

序号	1	2	3	4	5	6	7	8	平均
单向	17	14	6	16	15	11	12	9	12.5
双向	19	4	16	14	18	21	8	10	11

4.2　大规模网络实验

在单向同步的基础上，本文提出了基于最大互邻集合的同步算法。为验证算法的有效性，在 NS2（Network Simulator 2）软件平台上进行大规模网络下的仿真，同时使用文献［8］中的算法进行对比。文献［8］提出了基于双向同步的成对广播时间同步算法，控制同步消息开销，与本文算法目标一致，新颖且具权威性。本文实验中，参数取值分别为 $B_1=1$，$B_2=2$，$B_3=3$，实验结果见图 4 至图 7。

图 4 是网络节点数从 100 至 400 范围变化的实验结果，平均节点邻居数大约保持在 10 个，即节点密度不变。在不同网络规模下，本文的算法得到的广播节点的数量均比文献中算法结果少。在同步阶段，本文算法的广播节点只需发送一个同步消息，同步消息数量与广播节点数量一致；而成对同步需要消息交换，因此消息开销是同步集节点数量的两倍。本文算法在同步阶段平均消息开销比单向同步算法低 61.43%，在降低能耗方面更有效。

图 5 中是网络节点数固定为 250 个、节点密度变化时的实验结果。本文算法的广播节点数比文献中方法同步集节点数量平均少 20.74%，消息开销

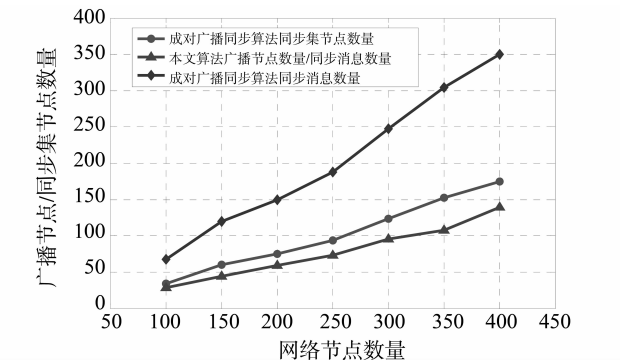

图 4 网络规模 100～400，同步集节点数量和同步消息开销

Fig. 4 Network size from 100 to 400, synchronization set size and synchronization message consumption

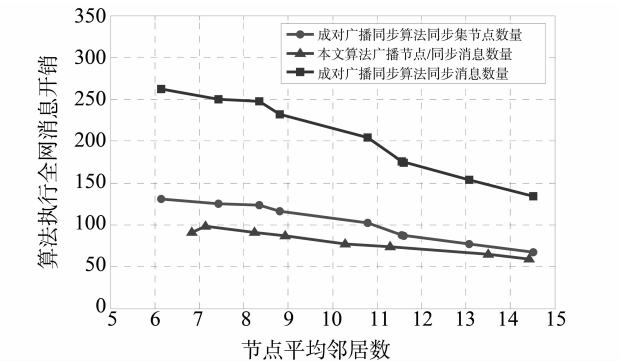

图 5 网络规模 250，同步集节点数量和同步消息开销

Fig. 5 Network size equals 250, synchronization set size and synchronization message consumption

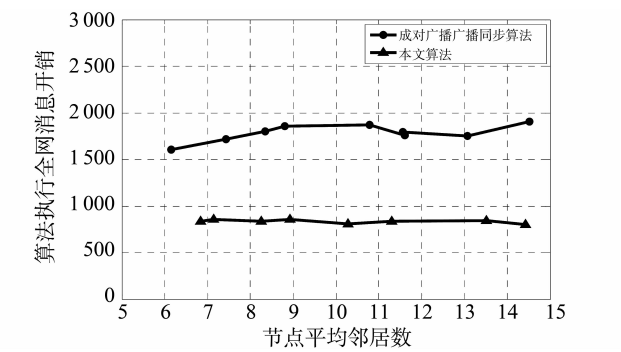

图 6 网络规模 250，执行算法的消息开销

Fig. 6 Network size equals 250, message consumption of algorithm execution

节约 60.37%。与网络规模变化时相似，在不同网络密度下本文的方法在节省网络开销和节点能耗方面更优秀。

图 6 是节点密度变化时，在层次发现阶段算法执行所需要传递的消息数量对比情况。表明本文方法所产生的平均消息开销为文献中算法的 46.8%，远小于后者。文献中算法在层次发现流程中，节点选择下层同步子节点需要多次的消息交互，而本文算法中节点只需固定数量的消息开销。

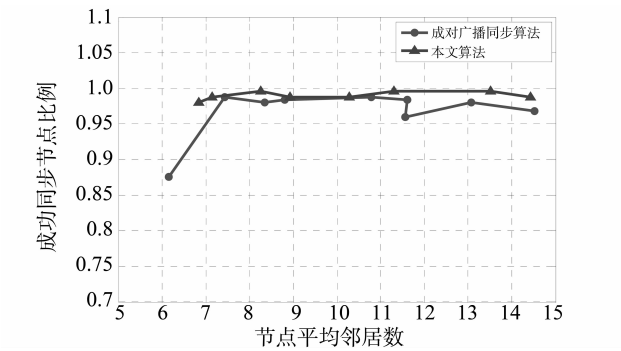

图 7 网络规模 250，成功同步节点比例

Fig. 7 Network size equals 250, node ratio of successful synchronization

图 7 是节点密度变化时成功同步节点数量占网络节点数量的比例。由于消息碰撞等原因，消息传递时可能会丢失少量帧，节点获取邻居信息不准确，从而造成少量节点被遗漏的情况。由于在本文中加入了遗漏节点处理、时序控制、放弃节点重新同步等策略，故同步算法更有效，成功同步节点的比例为 99.07%，额文献中方法缺乏相关处理，比例为 96.76%。

在运算时间复杂度方面，本文算法在搜索最大互邻集合时的运算复杂度为 O（$n\log n$），同层邻居获取最近邻集合为 O（n），n 为节点的平均同层邻居数，而其余的步骤均为 O（1）。在文献中的算法中，获取并排序所有低层邻节点的邻居表过程的复杂度为 O（$m\log m$），其后搜索最大共同邻居集合和 $sync\_num$ 则为 O（n^2），筛选最大的 $sync\_num$ 为 O（m），且需多次迭代，最后根据 not_MAX 消息作出的处理的复杂度为 O（$n*m$）或 O（1），其中 m 和 n 分别为同层和上层邻居数。对比可知，本文算法的运算复杂度更低，流程更简单。

5 结 语

本文对无线传感器网络的时间同步问题进行了研究。利用单向同步的时间戳标记技术，将时间同步转化为构造最小同步集问题，并建立数学模型，提出分布式的算法求解。算法通过在层次发现阶段节点生成最大互邻集合和最近邻集合，减少同步集节点数量。加入子节点向父节点注册、层节点监听放弃帧以及时序控制等策略提高了算法性能。ZigBee 硬件平台的实验证明了单向同步可达到较高精度；NS2 平台的仿真表明本文方法在不同网络规

模、不同网络密度下可降低同步消息开销，提高成功同步节点比例，能更有效地解决时间同步问题。

参考文献：

[1] AKYILDIZ I F, SU W, SANKARASUBRAMANIAM Y, et al. Wireless sensor networks: a survey[J]. Computer Networks, 2002, 38(4): 393 – 422.

[2] ARAMPATZIS T H, LYGEROS J. A survey of applications of wireless sensors and wireless sensor networks [C] // Proceedings of the 13th Mediterranean Conference on Control and Automation, Limassol, 2005: 719 – 724

[3] SOMMER P, WATTENHOFER R. Gradient clock synchronization in wireless sensor networks[C] // Information Processing in Sensor Networks, San Francisco, 2009: 37 – 48.

[4] LISKOV B. Practical uses of synchronized clocks in distributed systems[J]. Distributed Computing, 1993, 6(4): 1 – 9.

[5] ELSON J, GIROD L, ESTRIN D. Fine-grained network time synchronization using reference broadcasts[C]. Proceedings of the 5th Symposium on Operating Systems Design and Implementation, New York, 2002, 36: 147 – 163.

[6] GANERIWAL S, KUMAR R, SRIVASTAVA M B. Timing-sync protocol for sensor networks[C] // 1st International Conference on Embedded Networked Sensor Systems, New York, 2003: 138 – 149.

[7] KYOUNG-LAE N, ERCHIN S, KHALID Q. A new approach for time synchronization in wireless sensor networks: pairwise nroadcast synchronization [J]. IEEE Transactions on Wireless Communications, 2008, 7(9): 3318 – 3322.

[8] KING-YIP C, KING-SHAN L, YIK-CHUNG W, et al. A distributed multihop time synchronization protocol for wireless sensor networks using pairwise broadcast synchronization[J]. IEEE Transactions on Wireless Communications, 2009,8(4): 1764 – 1772.

[9] MUHAMMAD A, TAREK R S. RTSP: an accurate and energy-efficient protocol for clock synchronization in WSNs[J]. IEEE Transactions on Instrumentation and Measurement, 2013, 62(3): 578 – 589.

[10] GEOFFREY Werner-Allen, GEETIKA Tewari, ANKIT Patel, et al. Firefly-inspired sensor network synchronicity with realistic radio effects[C] // Proceedings of the 3rd International Conference on Embedded Networked Sensor Ssystems, New York, 2005: 142 – 153.

[11] LUCA Schenato, GIOVANNI Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network[C] // Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, 2007: 2289 – 2294.

[12] MAGGS M K, O'KEEFE S G , THIEL D V. Consensus clock synchronization for wireless sensor networks[J]. IEEE Sensors Journal, 2012, 12(6): 2269 – 2277.

[13] PINHO A C, FIGUEIREDO D R , FRANCA F M G. A robust gradient clock synchronization algorithm for wireless sensor networks[C] // Communication Systems and Networks (COMSNETS), 4th International Conference on, Bangalore, 2012: 1 – 10.

[14] COX D, JOVANOV E, MILENKOVIC A. Time synchronization for ZigBee networks[C] // Proceedings of the 37th Annual Southeastern Symposium on System Theory (SSST '05), 2005:135 – 138.

[15] BONSMA S PAUL , TOBIAS B, GERHARD J. A faster FPT algorithm for finding spanning trees with many leaves[C] // Mathematical Foundations of Computer Science, 28th International Symposium, Bratislava, 2003: 259 – 268.