

无线传感器网络的分布扩散时钟自同步协议研究

李敏¹, 郑国强¹, 李济顺²

(1. 河南科技大学 电子信息工程学院, 河南 洛阳 471023;

2. 河南科技大学 河南省机械设计及传动系统重点实验室, 河南 洛阳 471023)

摘要: 针对大规模无线传感器网络同步协议的精度较低、可扩展性差的问题, 提出一种分布式扩散时钟自同步协议 (DDCSS); DDCSS 是一个局部、并行执行的协议, 基于分布式扩散的思想, 以节点能量、分布和平均传输时延为依据, 每轮动态地选取执行局部扩散的一组主节点和扩散节点, 把主节点域内的节点平均时钟扩散有限的跳数, 周围节点以接收的所有主节点域平均时钟取平均更新本地时钟, 采用互扩散的方法使节点时钟近似同步到网络节点的平均时钟上, 从而实现全网的时间同步; 与 RBS、TPSN 协议相比较, 该协议收敛速度较快, 另外消除了误差累积, 同步误差较低, 扩展性较好。

关键词: 无线传感器网络; 分布式扩散; 时钟同步

Clock Self-Synchronization Protocol Based on Distributed Diffusion for Wireless Sensor Networks

Li Min¹, Zheng Guoqiang¹, Li Jishun²

(1. School of Electronic Information Engineering, Henan University of Science and Technology, Luoyang 471003, China; 2. Henan Key Laboratory for Machinery Design and Transmission System, Henan University of Science and Technology, Luoyang 471003, China)

Abstract: In view of the low precision and scalability of synchronization protocol for the large-scale wireless sensor networks (WSNs), clock self-synchronization protocol based on distributed diffusion (DDCSS) is proposed in this paper. The distributed diffusion-based protocol is full localized. Its implementations process is fully localized and parallel. According to the energy, distribution and the average transmission delay of nodes, DDCSS dynamic chooses a set of master node and diffusion nodes which are used to perform local diffusion in each round. And then the average clock in master node field is spread to surrounding nodes of hops. After that the surrounding nodes update the local clock by receiving the average clock. Eventually through several rounds of elections master nodes and diffusion nodes, networks implement mutual diffusion and the node clock is approximate synchronization the average clock of network nodes. By comparing DDCSS with RBS and TPSN, we show that, with the increasing of scale of sensor networks, DDCSS can synchronize the network quickly with good precision convergence speed and scalability, which appropriates for large-scale WSNs.

Keywords: wireless sensor networks; distributed diffusion; clock self-synchronization

0 引言

传感器网络的许多应用都要求整个网络节点维持一个相对一致的时间, 只有在邻近节点维持同步的情况下, 才能保证传感器节点之间相互稳定、协调的工作时间同步是传感器网络应用中的一项重要支撑技术^[1]。

在 WSNs 领域内, 存在一些典型的时间同步协议, 如 DMTS^[2]、RBS^[3]、TPSN^[4] 和 LTS^[5] 等。这些算法大多是采用提高单跳同步精度或选取最短路径同步来减少跳数, 以降低多跳误差累积, 却没有考虑到充分利用周围节点的时钟信息以降低误差随跳数累积的速度。目前采用成对同步机制的算法在实现中时间信息是由高层次节点向低层次节点索取的, 不能快速扩展至全网, 收敛速度较慢; 拓扑结构建立后不再改变, 全网同步效果极易受到节点失效等特殊状况的影响。

针对上述问题, 本文提出一种分布式扩散时钟自同步协议 (DDCSS), 它通过利用分布式扩散的思想, 以节点能量、分布和平均传输时延为依据, 动态选取主节点和扩散节点, 基于成对节点间的信息交换模式, 把主节点区域的平均时钟扩散有限的跳数, 采用互扩散的方法使整个网络节点时钟近似同步到网络节点平均时钟上, 从而使全网的同步误差更平滑, 算法更简单, 同步开销进一步减小, 同步收敛更快。

1 DDCSS 同步协议

同其它传统的大规模系统的时间同步一样, 传感器网络的时间同步也受到温度、频率噪声、相位噪声和链路不对称时延等的影响而产生一定的同步误差^[6-7]。除这些因素外, 传感器网络的同步还需要考虑传感器网络的自组织和资源有限等特点^[8-9], 所以提出的 DDCSS 协议是一个自组织的网络同步协议, 没有精确的时间服务器, 同步的目的就是尽可能使整个网络维持一个相对一致的时间, 而且邻居节点具有较小的同步误差。但由于节点时钟的漂移, 整个网络同步到一个相对一致的时间之后, 经过一定的时间, 不同节点的时钟之间就会产生较大的差别, 为此根据节点时钟的特性采用周期性同步; 另外考虑整个网络虽然能维持一个相对一致的时间, 但在用户使用时必须把网络时间转换为通用的标准时间, 所以提出图 1 系统结

收稿日期: 2014-01-16; 修回日期: 2014-03-16。

基金项目: 国家科技重大专项 (2012ZX04004011); 河南省基础与前沿技术研究计划项目 (122300413209); 河南省教育厅科学技术研究重点项目 (12B510010)。

作者简介: 李敏 (1984-), 女, 河南太康县人, 硕士研究生, 主要从事无线传感器网络时间同步方向的研究。

构。图 1 中，对感知区域的时间，Sink 节点利用时间转换算法，把其转换为通用的标准时间，而整个感知区域的节点，执行 DDCSS 维持一个相对一致的时间。

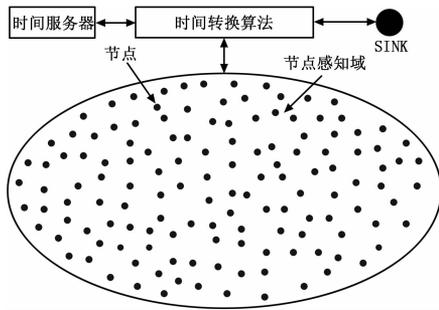


图 1 系统结构

DDCSS 的周期执行过程如图 2 所示。DDCSS 协议的每轮同步执行过程如下：①主节点选举：在同步时间 $m\tau$ 期间，每经过 τ 秒选取一次主节点，一次选取的主节点数量 S 要根据同步的精度和网络的节点分布等特征合理确定，同样 τ 和 m 的选取要根据网络的类型及应用要求和 S 的大小确定；②主节点域内平均时钟的估算：执行发送者—接收者的成对节点通信模式，获取主节点域内所有邻居节点的时钟，得到广播域内节点平均时钟；③扩散节点的动态选举和平均时钟的扩散：首先把主节点同步在广播域内所有节点的平均时钟上，然后以各主节点的时钟为参考时钟，从主节点开始以平均传输时延和节点能量为基准，选取扩散节点，将其扩散至距离主节点 ω 跳远的所有邻居节点；④节点的时钟更新：距离主节点 ω 跳内的节点在每轮同步时间 τ 内会分别接收到多个来自同一主节点和不同主节点的时钟同步信息，通过利用这些信息更新本地的时钟。按照这个过程执行 m 轮，就完成了了一次网络同步过程。

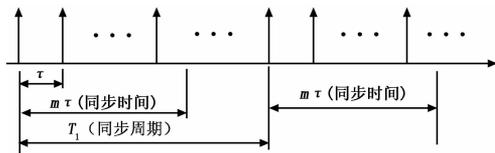


图 2 同步协议的周期执行过程

DDCSS 协议的基本假设是网络中的节点和其广播域内的邻居节点是强连接的，同步期间所有的节点都处于活动状态，网络中的所有节点时钟都工作于同样的振荡频率，时钟漂移率是有界的，在一轮同步期间节点时钟的累积漂移远小于给定的网络同步误差，最初所有节点的时钟时间随机均匀分布于有界的范围内，最初启动同步的初始化信息由 Sink 节点通过直接的广播或多跳的洪泛传送给网络中的所有节点。下面就 4 个阶段进行详细描述。

1.1 主节点选举算法

DDCSS 执行同步期间每轮选举的主节点必须考虑两方面的因素：1) 节点的能量；2) 节点的分布。为了使整个网络节点从能量上尽可能维持一个相对平衡的剩余能量，增加网络的寿命和局部的连通性，尽可能选举具有较多剩余能量的节点作为主节点，同时为了使主节点在执行时钟扩散过程中具有一定的独立覆盖范围，能够并行执行扩散，必须考虑主节点的分布。所以主节点的选举算法要满足下列两个规则。

规则 1.1: 假设每个节点维持一个阈值 φ ，主节点选举期

间，每个节点产生随机数 λ ，且 λ, δ 表示节点当前的剩余能量与节点最初能量的比值，按照下式计算得到 ζ ：

$$\zeta = \lambda - (1 - \delta) \tag{1}$$

如果 $\zeta > \varphi$ ，那么该节点就声明为一个主节点；否则就不能声明为主节点。这里主节点占网络节点的比例为 $\gamma = 1 - \varphi$ 。

规则 1.2: 满足规则 1.1 的所有节点，随机等待一定时间在其广播域内发出一个声明为主节点的报文，若其广播范围内存在其它的节点满足规则 1.1，则这些节点就退出作为主节点的竞争；若其广播范围内的邻居节点接收到不同的声明报文或发生碰撞的声明报文，则这些节点立即发送报文冲突的响应信息，发出声明报文的节点一旦接收到该响应信息，就按照 1/2 概率决定是否继续发送作为主节点的声明，直至发送声明报文的节点广播域内不存在接收到来自不同节点声明报文的邻居节点为止；若其广播范围内的邻居节点仅收到该声明报文，那么发送声明报文的节点在等待一定的时间之后就开始执行本轮的同步过程。主节点选举是多轮的，在同步时间 $m\tau$ 内，每经过 τ 都会重新进行选举。

1.2 主节点域内节点的平均时钟

假设一轮主节点选举过程中产生 S 主节点，每个主节点广播域内的节点数为 $n_j (j = 1, 2, \dots, s)$ ，表示在时刻 l 所选举的主节点 j 广播域内的 n_j 个邻居节点的时钟值， $c_k^l (k = 1, 2, \dots, n_j)$ 是节点在 l 时的时钟值，其中 c_1^l 表示主节点时钟。主节点广播域内平均时钟的获取过程如下：

1) 主节点首先广播一个 ch-quest 分组（包括启动同步标志 sync-start、主节点 ID、本地时钟值等）启动新的同步；

2) 邻居节点根据接收到 ch-quest，随机等待一定的时间后，执行基于发送者—接收者的时钟信息交换模式发送包含时间戳的 ACK 应答分组（包括接收到 ch-quest 时的本地时钟间值、发送 ACK 时本地时钟值和该节点的编号）；

3) 主节点接收到应答分组，估算节点间的传播时延，随后发出一个 sync-continue 确认分组给邻居节点（sync-continue 中包含标志 sync-flag、主节点同做出应答的邻居节点 k 间的时延 d_k 以及邻居节点 k 的编号和当前广播信息的发送时间），邻居节点 k 接收 sync-continue 后就可以获取 d_k ，其它的邻居节点通过接收 sync-continue 分组重复上面的信息交换，直至主节点再发出 sync-continue 信息后等待一个最大的延时时间 D_{max} 而未能接收到来自邻居节点的时间戳信息，说明主节点获取了所有邻居节点的时钟信息。图 3 说明了主节点 A 和邻居节点 B 在 l 时刻按上述步骤执行的信息交换。

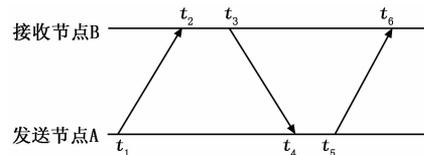


图 3 基于发送节点 A—接收节点 B 的信息交换

若 Δ_k 表示第 k 个邻居节点同主节点间的时钟漂移， d_k 表示第 k 个邻居节点同主节点间的传播时延，则：

$$t_2 = t_1 + \Delta_k + d_k \tag{2}$$

$$\begin{cases} \Delta_k = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \\ d_k = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \end{cases} \tag{3}$$

在 l 时刻以主节点时钟 $c_1^l = t_1$ 为基准的邻居节点的时钟时间为:

$$c_k^l = c_1^l + \Delta_k \quad (4)$$

4) 利用式 (3) 和 (4) 得到主节点广播域内所有节点在时刻 l 的平均时间 $\overline{c_j^l}$ 和平均传播时延 $\overline{d_j^l}$:

$$\begin{cases} \overline{c_j^l} = \sum_{k=1}^{n_j} c_k^l / n_j = c_1^l + \sum_{k=1}^{n_j} \frac{\Delta_k}{n_j} \\ \overline{d_j^l} = \sum_{k=1}^{n_j} d_k / n_j \end{cases} \quad (5)$$

1.3 扩散节点的选举和平均时间扩散

首先以主节点的平均时间为参考,对扩散节点进行如下规则的选举。

规则 1.3: (能量要求) 若节点在接收到上级主节点或扩散节点 ch-quest 分组后,节点按照主节点选举方法,产生随机数 η ,且 $\eta \in (0,1)$,并根据节点能量计算: $\zeta = \eta - (1 - \delta)$,若 $\zeta \geq \varphi_1$ 则该节点可以成为扩散节点,否则该节点不能成为扩散节点。门限值 φ_1 决定了选举的扩散节点的数目,它与网络的节点密度、节点的通信半径等有关。由于 δ 是随能量变化的,所以每轮同步之后必须对门限 φ_1 进行调整即 $\varphi_1 = \varphi_1 - \nu$, ν 可根据应用设置为任意小的正数。

规则 1.4: (时延要求) 节点在接收到上一级的主节点或扩散节点的 ch-quest 分组后,若 $d_k > \overline{d_j^l} \cdot \overline{d_j^l}$ 表示平均单跳时延, ω_k 表示传播的跳数, d_k 表示节点同主节点或上一级扩散节点的信息时延,则该节点可以成为下一级的扩散节点,否则该节点不能成为扩散节点。时钟扩散从主节点开始,过程如下。

0 级扩散: 主节点发送包含主节点 ID, 平均时间 $\overline{c_j^l}$, 传播跳数 ω (每扩散一次该值减 1) 和平均时延 $\overline{d_j^l}$ 的时钟扩散同步分组信息。

1 级扩散: 主节点执行 0 级扩散之后,其邻居节点按照扩散节点的选举规则,确定能否作为 1 级扩散节点,符合规则的邻居节点通过对信道的侦听,作为 1 级扩散节点执行时钟的扩散。同样 1 级扩散节点首先和其广播域内的未获取相同主节点的平均时钟信息的邻居节点,使每个邻居节点获取其同该扩散节点之间的信息单跳时延,并计算同所有邻居节点的平均单跳时延 $\overline{d_j^l}$,然后 1 级扩散节点把接收的主节点的平均时钟更新为: $\overline{c_j^l} = \overline{c_j^l} + d_{0,k}$, 这里 $d_{0,k}$ 表示当前的扩散节点同其主节点间的信息传播时延即公式 (3) 中的 d_k ; 将接收到的扩散分组中的 ω 减 1; 把主节点同邻居节点间的平均时延 $\overline{d_j^l}$ 用当前的 1 级扩散节点同其邻居节点间的平均时延 $\overline{d_{1,k}^l}$ 代替。把更新后的扩散分组广播给其下一跳的邻居节点。

f 级扩散: 同 1 级扩散过程类似。整个扩散过程一直持续到距离主节点 ω 跳的邻居节点为止。

这里 ω 取决于同步的精度和速度,必须能确保相邻主节点之间的节点至少获得两主节点区域的平均时间信息并作为参考时间进行同步更新,所以 ω 满足:

$$\begin{cases} \frac{\sqrt{2}W}{R} \geq \frac{2W}{(R\sqrt{\pi\gamma N})} \geq 2 \\ SS^m \omega \geq \sqrt{2} \frac{W}{R} \end{cases} \quad (6)$$

设置主节点 ID, 以避免重复接收来自相同主节点域信息。另

外扩散节点执行扩散过程中,邻居节点根据记录的主节点编号,确认已经接收来自同一主节点的平均时钟扩散信息,就不再参与信息交换;若扩散节点的所有邻居节点全部接收来自同一主节点的扩散信息,则该节点在等待一定时间后就终止其扩散过程。

1.4 节点的时钟更新

节点根据接收到的同步扩散分组,按照下列规则更新节点时钟。

规则 1.5: 每轮同步周期内,节点若是第一次接收到来自某个主节点的平均时钟同步扩散分组,按 $T_{new} = \overline{c_M(l)} + d_k$ 估算新的时间。如果 $|T_{new} - T_{local}| > \delta$ (δ 为根据节点一轮同步周期的最大漂移和同步要求给定的一个大于 0 的小数),那么该节点的时钟就更新为 T_{new} , 否则维持本地时钟 T_{local} 不变。

规则 1.6: 若节点接收了 Z 个主节点的同步扩散分组后,又接收到不同的主节点的同步扩散分组,则根据接收该消息时的本地时钟 T_{local} 和消息中包含的同步时钟信息 $\overline{c_M(l)}$ 计算新的时间 T_{new} :

$$T_{new} = \frac{\overline{c_M(l)} + d_k + Z \cdot T_{local}}{(Z+1)} \quad (7)$$

若 $|T_{new} - T_{local}| > \delta$, 则用 T_{new} 更新 T_{local} , 否则维持当前的本地时间不变。

2 实验结果及分析

为了验证 DDCSS 的同步效果,基于 Mica Z 平台进行了网络仿真实验,相关参数设置如下:监测区域面积为 $500 \times 500 \text{ m}^2$, 随机布置 1 000 个节点,节点通信半径 $R=10 \text{ m}$,耦合强度 $\epsilon=0.02$,周期 $T=10 \text{ s}$,仿真时间为 50 min,每一种情况都是对 100 次仿真取平均值。

假设在时刻 l ,所有节点的时间同标准时间的偏差均匀分布于区间 $[L_l, H_l]$ 内,那么 DDCSS 满足如下收敛定理。

定理 2.1: 对大规模传感器网络,DDCSS 分布式平均时钟扩散的自同步算法渐进收敛于 C (C 等于网络中所有节点的平均时钟)。

证明: 假设每个同步周期选举的主节点数都为 S , H_l^j, L_l^j 分别表示第 j 轮同步周期后 S 个主节点区域的平均时钟 $\overline{c_M(l)}$ 与标准时钟的偏差最大值和最小值,用 $c_n^j(l), c_{sd}^j(l)$ 分别表示第 j 轮同步周期后任意节点的时间和标准时间,则 m 轮同步周期后,网络中任意节点的时间 $c_n^m(l)$ 满足:

$$\begin{aligned} L_l < L_l^1 < \dots < L_l^{m-1} < L_l^m \leq c_n^m(l) - c_{sd}^m(l) \leq \\ H_l^m < H_l^{m-1} < \dots < H_l^1 < H_l \end{aligned} \quad (8)$$

显然 $H_l^j \geq C$, 设定 $\lim_{l \rightarrow \infty} H_l^j = M \geq C$, 假设 $M \neq C$, 则设定

变量 $\psi(\alpha) = \sum_{i=1}^{\alpha} n^i \psi$, 其中 ψ 满足 $M - \frac{n^{m+1} - 1}{n - 1} \psi = M - \psi(n) = C$, n 是节点数。对于任意的 α ($\alpha = n, n-1, \dots, 1$), 时钟值大于 $M - \psi(n)$ 的节点归入 Γ_n^{α} , 剩余节点归入 Γ_n^{α} 。对于 ψ , 存在时刻 t 满足 $H_l^t < M + \psi$; 同时必然存在部分节点的时间值小于 $C = M - \psi(n)$ 。首先获取平均时间在 Γ_n^{α} 和 Γ_n^{α} 中, 由于存在部分节点时间值小于 $C = M - \psi(n)$, 则有 $|\Gamma_n^{\alpha}| \geq 2$ 。然后在 Γ_{n-1}^{α} 和 Γ_{n-1}^{α} 中获取平均时间得到 $|\Gamma_{n-2}^{\alpha}| \geq 3$ 。最终 $\Gamma_1^{\alpha} = n^5$, 这与 $\lim_{l \rightarrow \infty} H_l^j = M$ 相矛盾, 所以 $\lim_{l \rightarrow \infty} H_l^j = C$ 。同理, 可证 $\lim_{l \rightarrow \infty} L_l^j = C$, 所以 m 轮同步周期后任意节点的时间收敛于 C 。

为了证明 DDCSS 的优越性,从收敛时间、同步误差和信息开销 3 个方面,将 DDCSS 协议与 RBS 和 TPSN 进行比较。

