

一种基于超帧机制的 Zigbee 时间同步算法研究是实现

高广恩, 刘全利*, 贾灵利, 王伟

(大连理工大学 控制科学与工程学院, 辽宁 大连 116024)

摘要: 对于几乎所有无线传感器网络应用来说, 时间同步是非常重要的. 由于无线传感器网络的多节点、低功耗等特点, 一种合适的时间同步算法的实现尤为关键. 以自主研发的硬件平台为基础, 设计并实现了一种适用于 Zigbee 无线通信协议的单跳和多跳网络的时间同步算法, 并给出了详细的算法分析及实现流程. 实验结果表明所提出的时间同步算法在 Zigbee 单跳及多跳信标网络中均有良好的性能表现, 满足了 Zigbee 协议低功耗、高同步精度的要求, 在实现 Zigbee 休眠节能机制的同时, 算法达到了较高的时间同步精度.

关键词: 时间戳; 时间同步算法; Zigbee 网络

中图分类号: TN925 **文献标志码:** A

0 引言

无线传感器网络 (wireless sensor network, WSN) 是近几年一个新兴的研究热点, 可广泛应用于军事、环境监测、工业控制等领域. 在无线传感器网络中, 无论数据融合、节点定位还是传感节点的协同都要求网络同步. 因此, 时间同步技术已成为无线传感器网络的关键技术之一^[1]. 无线传感器网络有着自组织、多跳、资源受限、低功耗、低带宽的特点, 这使其对时间同步算法提出特殊的要求. 在某些特定网络环境或应用场合, 现有的时间同步算法有着较好的同步精度和较短的收敛时间. 但针对 Zigbee 协议的时间同步算法较少, 不能和协议很好兼容. 针对无线个域网 Zigbee 联盟提出了 Zigbee^[2] 标准, 它为用户提供了一种成本低、功耗小、速率低、可靠性高、灵活性好的解决方案. Zigbee 技术被广泛应用于智能家居、监控监测、智能交通、过程控制等领域, 完成数据采集和处理, 并对采集对象进行控制等功能. 已有的 Zigbee 网络时间同步算法, 考虑的多为星形网络中的同步问题, 而对多跳网络研究较少. 基于此, 借鉴已有的时间同步算法思想并结合 Zigbee 网络的特点, 本文提出一种适用于 Zigbee 网络的时间同步算法 Zigbee-Sync. Zigbee-Sync 充分利用超帧机制, 具有低开销、高

精度、高可靠性的特点. 最终通过搭建测试平台验证算法的性能.

1 无线传感器网络时间同步

目前国内外已有多篇文献对无线传感器网络的时间同步问题进行了研究, 提出了不同的时间同步算法. 下面将几种具有代表性的时间同步算法进行介绍和分析.

RBS 算法^[3] 利用无线信道的广播特性, 节点发送参考信标到邻居节点, 邻居节点交换它们收到参考信标的时间信息, 根据此信息接收节点调节本地时间, 达到和其他邻居节点同步的目的. RBS 算法消除了发送时间和访问时间的影响, 提高了邻居节点间的同步精度. TPSN 算法^[4] 是一种基于层次结构的时间同步算法, 整个同步过程分为两个阶段: 一是层次发现阶段, 二是时间同步阶段. TPSN 算法可实现全网同步, 其同步收敛时间的长短和网络规模有关, 若需要长时间保持同步, 则需要周期性地执行同步过程. LTS 算法^[5] 是一种轻量级的同步算法. 当节点 i 需要同步时, 就发送同步请求到其邻居节点. 若邻居节点此时处于同步状态, 则发送同步时间信息到节点 i , 完成节点 i 的时间同步; 否则邻居节点继续沿着朝

向时钟源节点的方向发送同步请求,直到完成同步. LTS 算法降低了同步所需的信息开销,其同步收敛时间和节点所处的深度有关系.

每一种同步算法都有其优势和不足,所以针对不同的应用网络,要充分考虑其网络特点来设计合适的算法, Zigbee-Sync 便是针对 Zigbee 网络设计的同步算法.

2 Zigbee-Sync 算法

Zigbee 是 Zigbee 联盟在 IEEE 802. 15. 4^[6] 定义的物理层和媒体访问控制层的基础之上,又定义了网络层和应用层,因此它包含 IEEE 802.

15. 4 的所有功能和特点.

2.1 软时钟构造

在实际的无线节点上,处理器所使用的时钟周期和无线网络时间同步所需要的同步参数有着较大的差别,因此需要在程序设计中构建时间同步所需的软时钟. 此软时钟不仅要精度高、可靠性高,更要符合网络需求和便于实现. 在 Zigbee 协议中,超帧是网络设备访问信道的基本周期,协调器维护超帧,为网络内设备分配访问时隙,图 1 为 Zigbee 协议的超帧结构,其中 BO (beacon order)、SO(superframe order)分别为信标阶数和超帧阶数,SD_b 表示基本超帧周期.

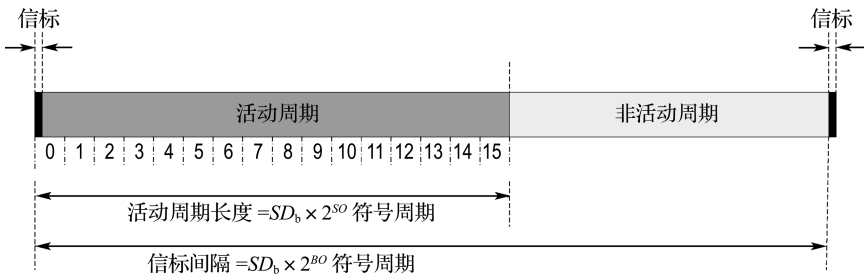


图 1 超帧结构
Fig. 1 Structure of superframe

在 Zigbee 网络中,时间同步指的是两个设备间的相对同步,而非和国际标准时间的绝对同步. Zigbee 标准中,维护超帧所需的最小时间单位为符号(symbol)周期,其实际时间长度为 16 μ s,因此选取符号周期为基本的时间单位. 符号周期为软时钟的基本时钟节拍(tick),超帧周期用节拍的数目表示. 这利于超帧维护,同时便于退避周期、时隙等时间参数的计算. DLUTWIA-PA 平台上 MCU 的主时钟频率为 11. 059 2 MHz,本文通过定时器分频得到时间长度为 16. 004 7 μ s 的基本时钟节拍,其与 16 μ s 的误差很小,可以满足网络要求. 在信标开始时对节拍从零进行计数,在本次超帧结束,下一信标帧到来前记录此计数值,表示超帧周期的大小,通过设定此计数值的大小可达到维护超帧的目的. 软时钟的同步精度为一个节拍,为 16 μ s. 网络内的两个设备可调节本地的计数值达到同步的目的,其最大时间偏差应为两个节拍,即 32 μ s,可达到较高的同步精度.

2.2 时间戳标记

时间戳为发送设备在发送数据帧时所获取的本地节拍计数值,其准确性对网络同步精度有很大影响. 因此在中断中获取此计数值,尽量减少程

序处理带来的误差. 在 Zigbee-Sync 算法中,发送设备将时间戳信息添加在信标帧负载中,以广播的方式告知网络内其他节点. 从节点捕获信标帧并解析负载中的时间戳信息,根据此信息调节本地时钟,以达到和发送设备同步的目的.

Zigbee 协议规定在发送数据帧时首先传输同步头和定界符(SFD). 射频芯片可在发送数据帧时产生 SFD 中断,发送设备在 SFD 中断中获取本地时间信息,并将其作为时间戳放在信标帧负载中. 接收设备在数据接收完成中断中可解析信标帧负载中的时间戳信息,并根据信标帧的长度估算出数据接收时间. 发送设备和接收设备处理时间戳的时刻如图 2 所示.

Mica2 等一些平台可以通过处理器直接控制发送数据流,从而可以很容易地标记和读取时间戳^[7]. 而在 DLUTWIA-PA 平台上处理器通过 SPI 接口和射频芯片相连,数据的发送与接收都需通过 SPI 接口进行. 所以在时间同步算法中要考虑数据在 SPI 接口上传输所需要的时间.

SPI 接口数据传输速率为 5 Mbps,射频收发器的数据传输速率为 250 kbps. 并且射频收发器支持先执行“发送”命令,然后再写入数据到射频

收发器内部的缓存,只要保证数据发送过程中缓存中有数据可用.在 IEEE 802.15.4 物理层标准中,定义了 5 个字节的同步头,发送完毕需要 160 μs ,且射频收发器从空闲状态切换到发送状态需要 192 μs ,在这 352 μs 时间内处理器可将数据写

入射频收发器的内部缓存.并且射频收发器的数据传输速率小于 SPI 接口的速率,不会产生内部缓存被读空的情况.时间戳信息共 32 位,通过 SPI 接口其传输需要 6.4 μs ,所需时间很短.图 3 为携带时间戳负载的信标帧.



图 2 时间戳标记
Fig. 2 Mark of timestamp

Time (us)	Length	Frame control field					Sequence number	Source PAN	Source Address	Superframe specification					Beacon payload	
+1266750 =1266750	18	Type	Sec	Pnd	Ack.req	Intra.PAN				BO	SO	F.CAP	BLE	Coord	Assoc	
		BCN	0	0	0	0	0xF6	0x1234	0x0000	06	02	15	0	1	1	12 00 00 00 04

图 3 信标帧
Fig. 3 Beacon frame

如图 3 所示,信标帧中 Beacon payload 部分为时间戳信息,表示发送方信标帧发送时刻的本地时间.由于每次发送信标帧之前都要对本地计数器清零,在超帧周期一定的情况下,其时间戳大小不会有太大变化.在时间戳信息中,前 4 个字节 0x00000012 为 SFD 触发时的本地时间,即第 18 个符号周期.最后一个字节 0x04 为本地修正时间,表示读取本地时钟以及时间戳处理的时间,主要通过理论计算的方式,根据具体的软硬件环境进行设定.

2.3 算法描述

在单跳的星形网络中,协调器周期性广播信标帧,网络内其他节点只需跟踪捕获信标帧,根据时间戳信息调节本地时钟便可达到和协调器同步.为了进一步降低网络能耗,网络内设备并不转发信标帧,这样节点可在无数据传输时处于睡眠状态,延长网络生存时间.树形网络是一种多跳拓扑的网络.假如两个相邻设备同时发送信标帧,则会产生信标帧冲突,造成子设备无法正常接收来自父设备的同步信息,进而影响网络同步.如何避免信标帧冲突便成为一个重要问题^[8].

可以通过分时复用的方式将各设备发送信标帧的时刻错开以避免冲突.在设备入网时根据设备在拓扑中的位置为其分配网络内短地址.根据此地址信息计算出父子设备间发送信标帧的时间

偏移量(Δt),通过在父设备发送信标帧的起始时刻添加偏移量,子设备发送信标帧的时刻便与父设备错开.各子设备间的偏移量有所不同,所以多个子设备发送信标帧的时刻也不会相同.在 Zigbee 网络中,一个发送信标帧设备的典型信标时序如图 4 所示.

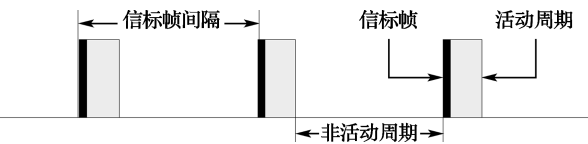


图 4 信标设备典型信标时序
Fig. 4 Classic beacon timing of beacon device

在 Zigbee 中,信标间隔 $BI=SD_b \times 2^{BO}$,活动周期 $CAP=SD_b \times 2^{SO}$,其中时间长度为 960 个符号.在活动周期,网络内设备进行数据传输,而在非活动周期则处于空闲,子设备可利用非活动周期发送信标帧.一个超帧周期可包含 $BI/CAP = 2^{(BO-SO)}$ 个 CAP. 非活动周期内还可以容纳 $2^{(BO-SO)} - 1$ 个 CAP,这些周期均可合理调度用来发送信标帧.以一个 $BO = 6$, $SO = 2$ 的网络为例,选取父子设备的时间偏移 $\Delta t = 2 * CAP$,经过信标调度后的协调器和两个子设备间的信标发送时序如图 5 所示.

经过这样的信标调度之后,不会产生邻居节

点间信标冲突的情况,每个节点只需侦听其父设备的信标帧,根据信标帧负载中的时间戳信息完成与其父设备的同步,达到间接和协调器同步的效果.这种机制简化了整个网络的同步机制,减少了节点间的信息交换,同时在网络规模不是很大的情况下可保证一定的时间同步精度.

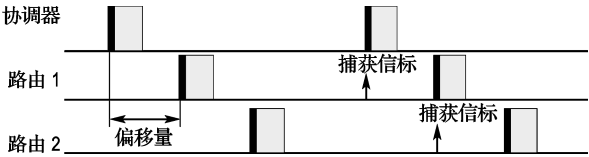


图 5 父子设备信标时序关系

Fig. 5 Beacon timing relationship between child and father device

3 算法实验验证

本时间同步算法在 TinyOS^[9] 软件平台中实现,硬件平台为自主研发的 DLUTWIA-PA 平台. DLUTWIA-PA 平台选用 LPC2136 作为微处理器,提供了丰富的外围接口,射频芯片为 TI 公司的 CC2520,该芯片兼容 Zigbee 协议.

3.1 实验方案

在实验室环境下,将同步算法在 DLUTWIA-PA 上加以实现,并对算法的性能进行了实验验证.网络分为星形网络和树形网络 2 种.在星形网络中包括 Zigbee 网络协调器,协调器上电后完成本地参数初始化并周期性广播信标帧. 5 个 Zigbee 终端设备扫描网络,并尝试加入网络.终端设备入网成功后执行本地的时间同步算法,解析时间戳信息,完成与协调器的时间同步.在超帧的起始时刻处理器触发一个 IO 的翻转操作,可通过示波器捕获波形,来观察设备间的同步误差.树形网络包括 1 个 Zigbee 协调器、3 个路由设备和 5 个终端设备.协调器和路由设备均发送信标帧,终端设备捕获信标帧,最终达到全网同步.为了更真实地模拟现场工作环境,整个网络在发送信标帧的同时,还在 CAP 时段和 GTS 时段传输数据.在实验过程中,网络连续运行 12 h,不断统计此期间各设备间的时间偏差.具体实现流程如图 6 所示.

3.2 算法验证

在星形网络中,每隔 5 min 采样一次主从节点和从从节点间的同步偏差,每次采样 20 个超帧周期.进行多组实验,发现各组间时间偏差并不大,同步算法有着较好的稳定性.在此给出一组时

间同步精度的统计结果,如图 7 所示, n 表示第 n 个超帧周期, Δt 表示时间偏差.

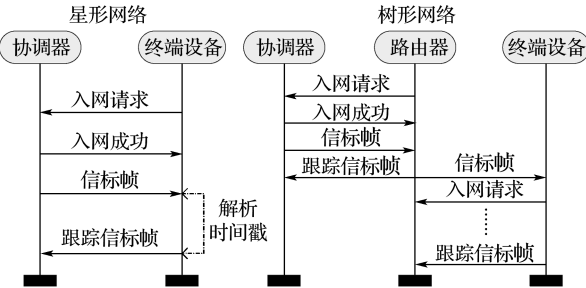


图 6 算法实现流程

Fig. 6 Process of algorithm realization

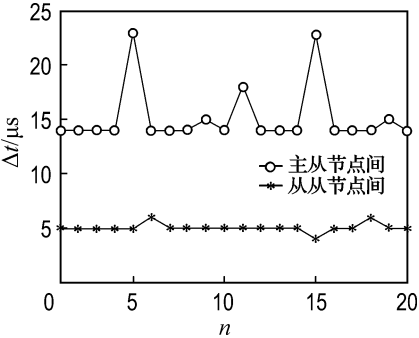


图 7 星形拓扑时间同步精度

Fig. 7 Time synchronization precision for star topology

由图 7 可见,主从节点间的同步精度随时间变化波动较大,这主要是受信标帧长度变化的影响.在网络工作过程中,信标帧除了需传递时间戳信息,还需传递 GTS 分配等信息.而不同的 GTS 分配情况会带来信标帧负载长度的变化.接收方只能在接收完整个信标帧后才能对时间偏差进行处理和调整,而信标帧负载长度的变化会对此处理时间产生影响,进而影响到同步精度.用 T_a 表示从节点所需调整的本地时间偏差,那么 $T_a = T_s + t$,其中 T_s 表示时间戳信息, t 为根据信标帧长度估算出的一个补偿变量.从从节点间的时间偏差波动较小,因为各从节点的时间戳信息均来自同一信标帧,由于超帧长度变化对各从节点造成的影响相同,所计算出的时间偏差相差不大.

按照相同的实验方法,统计了多跳树形网络中父子路由设备之间以及各终端节点间的时间同步偏差情况,统计结果如图 8 所示.

将实验中获取的时间偏差统计结果和加州大学伯克利分校在 MICA 平台上实现的 RBS 和 TPSN 算法^[4] 进行比较,结果如表 1 所示.本文算

法取得的平均时间偏差要小于其他两种算法。

图 9 给出了不同大小的时间偏差在所有偏差中所占百分比 p 。

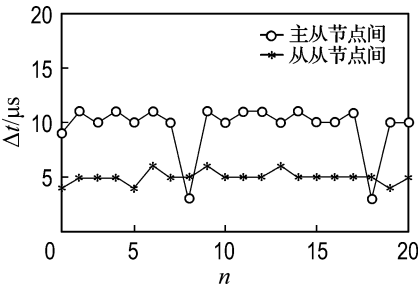


图 8 树形网络时间同步精度

Fig. 8 Time synchronization precision for tree network

表 1 时间同步偏差

Tab. 1 Error of time synchronization

算法	平均时间 偏差/μs	最大时间 偏差/μs	最小时间 偏差/μs	低于平均 时间偏差 百分比/%
RBS	29.13	44	0	64
TPSN	16.90	93	0	53
Zigbee-Sync	14.70	28	3	54

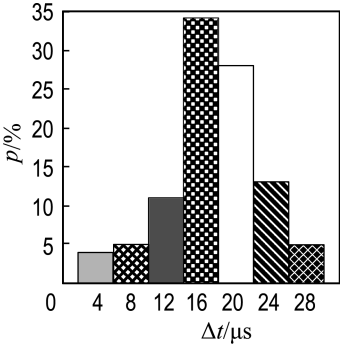


图 9 节点间时间偏差

Fig. 9 Time deviation between nodes

功耗对 Zigbee 网络设备来说是一个非常重要的指标,根据实际的实验环境通过仿真软件 NS2 仿真得出网络节点的能耗曲线图,如图 10 所示.具体的仿真参数设置见表 2.通过仿真结果可以看出,同异步网络相比,引入同步算法的 Zigbee 网络,其网络设备能耗大大降低,延长了网络生存时间。

Zigbee-Sync 算法借鉴了 RBS 算法的思想,借助第三方广播传递时间戳信息,并达到了较好的同步精度.针对 RBS 算法不适于多跳网络的缺陷,本文通过引入信标帧调度机制,成功将 Zigbee-Sync 算法在多跳网络中实现. TPSN 算法

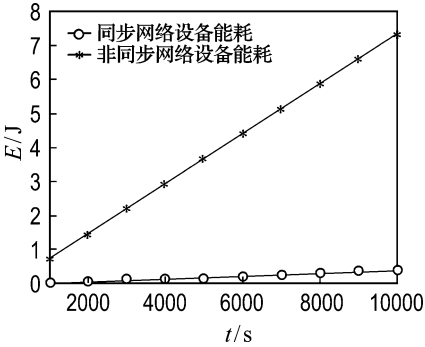


图 10 节点能耗曲线图

Fig. 10 Energy-consuming graph for nodes

表 2 仿真参数

Tab. 2 Parameters of simulation

参数	取值
仿真时间/s	10 000
节点个数	协调器(1),终端节点(6)
拓扑结构	星形
数据流/(packet · s ⁻¹)	0.1
数据包大小/bit	32
BO	6
SO	2
初始能量/J	100
接收功率/mW	35
发送功率/mW	37
空闲功率/μW	712
睡眠功率/nW	1 114

将时间戳信息标记在 MAC 层,可达到 16.9 μs 的同步精度^[4], Zigbee-Sync 算法将时间戳放在 MAC 层的信标帧负载中,其同步精度和 TPSN 算法不相上下,但却有效降低了同步过程中信息交换的次数,使节点由于传输同步信息所消耗的能量大大降低.网络内设备还可以进行按需同步,在无任务处理之时,转入休眠状态;当有中断唤醒新任务时,可按照已获取的网络信息重新捕获信标帧,这可进一步降低能耗.对于轻量级的 LTS 算法而言,其所取得的同步精度要低于 Zigbee-Sync 算法。

本算法基于节点较少的无线网络对同步算法的性能进行了验证.要想更好地在多节点、大规模网络中实现此同步算法,则需要考虑引入信标帧调度机制,保证路由由节点间的信标传输不会发生冲突,以可靠地传输同步信息.一个中间路由设备在接收到其父设备的信标帧后,根据自身与父设备之间的时间偏移量以及信标帧中的时间戳信息来校正本地时钟,从而确定出自身的信标时序.因为路由设备和父设备间的时间偏移量是一个预先

分配的固定值,此值对本算法的同步精度不会造成负面影响,多跳网络中的路由设备及其父设备之间依然能够保证较好的同步精度.可见在引入信标帧调度机制的情况下,本同步算法在多节点、大规模的无线网络中依然是可行的、有效的.

4 结 语

本文提出了一种基于超帧机制的适合于 Zigbee 网络的时间同步算法,该算法有效降低了同步过程中的信息交换次数,降低了网络内节点能耗,并且具有良好的稳定性.将 Zigbee-Sync 算法在实际平台 DLUTWIA-PA 上进行了实验验证,通过实验结果可知其可达到较高的同步精度,且运行稳定.本文设计的通过软件来标记时间戳的方式,可广泛用于其他无法通过处理器直接发送数据的硬件平台.

参考文献:

- [1] 康冠林,王福韵,段渭军.无线传感器网络时间同步综述[J].计算机测量与控制,2005,13(10):1021-1030. KANG Guan-lin, WANG Fu-yun, DUAN Wei-jun. Survey on time synchronization for wireless sensor networks [J]. **Computer Measurement & Control**, 2005, 13(10):1021-1030. (in Chinese)
- [2] Zigbee Alliance. Zigbee Standards. [2011-04-02]. <http://www.Zigbee.org>.
- [3] Jeremy E, Lew G, Deborah E. Fine-grained network time synchronization using reference broadcasts

- [C] // **Proceedings of the 5th Symposium on Operating Systems Design and Implementation**. New York:ACM Press, 2002:147-163.
- [4] Saurabh G, Ram K, Mani B S. Timing-sync protocol for sensor networks [C] // **Proceedings of the First International Conference on Embedded Networked Sensor Systems**. New York:ACM Press, 2003:138-149.
- [5] Jana V G, Jan R. Lightweight time synchronization for sensor networks [C] // **Proceedings of the Second ACM International Workshop on Wireless Sensor Networks and Applications**. New York:ACM Press, 2003:11-19.
- [6] ZHOU M, ZHANG L N. Analysis and design of Zigbee MAC layers protocol [C] // **2010 International Conference on Future Information Technology and Management Engineering**. Piscataway:IEEE Press, 2010:211-215.
- [7] Cox D, Jovanov E, Milenkovic A. Time synchronization for Zigbee networks [C] // **Proceedings of the Thirty-Seventh Southeastern Symposium**. Piscataway:IEEE Press, 2005:135-138.
- [8] Ahn S, Cho J, An S. Slotted beacon scheduling using Zigbee Cskip mechanism [C] // **The Second International Conference on Sensor Technologies and Applications**. Piscataway: IEEE Press, 2008: 103-108.
- [9] Levis P A. TinyOS: An open operating system for wireless sensor networks [C] // **Proceedings of the 7th International Conference on Mobile Data Management**. Piscataway:IEEE Press, 2006:1-11.

Research and implementation for Zigbee synchronization algorithm based on superframe mechanism

GAO Guang-en, LIU Quan-li*, JIA Ling-li, WANG Wei

(School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: Time synchronization is essential for almost all the wireless sensor network application. Because of the multi-nodes and low energy consumption feature of wireless sensor network, the realization of a proper time synchronization algorithm is especially important. Based on a self-developed hardware platform, an appropriate single-hop and multi-hop time synchronization algorithm of Zigbee protocol is presented and a detailed algorithm analysis and implementation flow are given. The experimental results show that the time synchronization algorithm has a favorable performance in the single-hop and multi-hop beacon networks, which ensures a high synchronization accuracy and the low power consumption. At the same time of implementing Zigbee energy-saving mechanism, the algorithm provides high synchronization.

Key words: timestamp; time synchronization algorithm; Zigbee networks