

# 基于生存周期的自组网同步研究

金瑞 刘作学

(装备学院信息装备系 北京 101416)

**摘要:** 主从式同步作为 TDMA 方式下自组网同步方式的一种,因其同步精度高、收敛速度快而被广泛研究。为了提高同步的收敛速度和同步精度,解决已有的主从式同步方法中存在的一些问题,文章利用网络节点的生存周期提出一种新的自组网同步方法。首先结合具体时帧设计和时隙接入详细论述了同步方法的实现过程,随后通过 OPNET 仿真软件对不同接入情况下所有网络节点的收敛时间和同步精度进行了测试。仿真结果表明即使当节点接入冲突数量达到最大时,在不同的入网状态下,所有节点中最长的同步收敛时间也不超过 1 s,并且源节点和距离 6 跳的节点之间的同步误差只有 0.37 ms,验证了同步方法的快速收敛性和精确性。

**关键词:** 自组织网络;时间同步;生存周期

**中图分类号:** TN929 **文献标识码:** A **国家标准学科分类代码:** 510.5015

## Research on time synchronization of Ad Hoc networks based on life-cycle

Jin Rui Liu Zuoxue

(Department of Information Equipment &amp; Equipment Academy, Beijing 101416, China)

**Abstract:** Master-slave synchronization has been widely studying because of its fast convergence speed and high synchronization precision as a synchronization methods of Ad Hoc networks based on TDMA. In order to improve the convergence speed and synchronization precision of master-slave synchronization, solve some problems in existing master-slave synchronization protocol, this paper proposes a new synchronization method for ad hoc networks using the life cycle of network nodes. This paper discuss the implementation of the synchronization method in detail with the time frame design and the time slot access, then the convergence time and synchronization accuracy of the network nodes in different access scenarios are tested by OPNET simulation software. The simulation results show that when the number of access conflicting nodes reaches a maximum, the longest synchronization time in different network state does not exceed 1s, and the synchronization error between the source node and the 6 jumps node is only 0.37 ms. The fast convergence and accuracy of the synchronization method are verified.

**Keywords:** Ad Hoc networks; time synchronization; life-cycle

## 1 引言

无线自组织网络(Ad Hoc)是一种不依赖基础设施而快速组建的临时性网络,网络中节点不仅具有移动终端所需的功能,还有报文转发能力<sup>[1]</sup>。自组网的媒体接入控制协议(MAC 协议)一直是目前研究的热点,主要分为竞争类和分配类<sup>[2]</sup>。其中基于 TDMA(时分多址)方式的分配类协议以其具有高网络吞吐量、传输时延可控等优点而备受青睐。时间同步是 TDMA 方式下 Ad Hoc 的技术支撑<sup>[3]</sup>。现有的同步方法主要包括互同步和主从式同步<sup>[4]</sup>,互同步方法中相邻节点之间同步,最终达到全网时间同步。但同步需要多次迭代,降低了同步的收敛速度<sup>[5]</sup>。主从式同步

通过一个时间主节点向其他节点发布基准时间来完成同步,对于距离主节点一跳以外的节点,TPSN 协议<sup>[6]</sup>提出了分层主从同步思想,网络先按照距离中心主节点的跳数建立等级层次,之后低等级层次节点同步高等级节点,最终完成向中心主节点的同步。但文章并没有考虑中心主节点失效后的问题。FTSP 协议<sup>[7]</sup>通过节点 ID 值选取中心节点,中心主节点失效后可以再通过剩下节点的 ID 值重新选取新的中心,但固定分配 ID 值失去了自组网的公平性。另外许多同步算法<sup>[8-9]</sup>设计的帧结构都采用固定的同步时隙,即节点和同步时隙已经固定对应,但这样对于不同节点的入网时间存在不公平性。针对以上问题,本文在设计动态同步时隙的 TDMA 帧结构基础上,提出一种基于生存周期的

同步方法,并对源节点组网同步,节电入网同步和源节点失效后的重新同步进行了描述,最后通过仿真测试了该同步方法的性能。

## 2 基本概念

### 2.1 生存周期

生存周期是自组网节点开机工作后的 TDMA 帧周期数量,最先开机的节点生存周期最大,这类节点称为同步源节点。生存周期的单位属于时间段,不会出现因时钟抖动而造成的误差<sup>[10]</sup>,所以作为同步源节点的选取依据和入网判别标准具有很强的稳定性和准确性。同时生存周期的提出可以使本地时钟只计时在一个复帧的范围内,因此能够提高计时精度,减小时钟传递的开销。

### 2.2 生存周期寄存器

生存周期寄存器由本节点生存周期寄存器(life-cycle register)和源节点生存周期寄存器(source life-cycle register)组成,以下简称本寄存器和源寄存器。本寄存器存储自身的生存周期,随着生存周期计数值增加而更新。源寄存器存储同步源的生存周期,通过记录源节点同步报文中 S-Lcycle 值而更新。

## 3 TDMA 帧结构设计

不同节点的时钟晶振频率都有差别,频率偏差长期积累后产生时钟偏差<sup>[11]</sup>,所以需要主节点周期性同步来控制时钟误差。又因为采取分层式的主从同步,所以每个节点都有可能成为某一层的主节点,因此本文将一帧内用于同步和接入的同步接入时隙 SJ(synchronous join slot)数目和最大允许节点数  $n$  相同,并将  $SJ(0) \sim SJ(n-1)$  这  $n$  个同步接入时隙拆分到  $n$  个子帧中保证节点可以快速接入。 $n$  个 SJ 时隙并不是固定分配给所有节点,节点入网时需通过侦听邻节点的时隙占用表来寻找未被占用的 SJ 时隙进行接入,接入成功后便可以进行业务时隙的预约。整个 TDMA 帧结构如图 1, $n$  个子帧 F 组成一个复帧 MF,一个复帧代表一个 TDMA 帧周期。

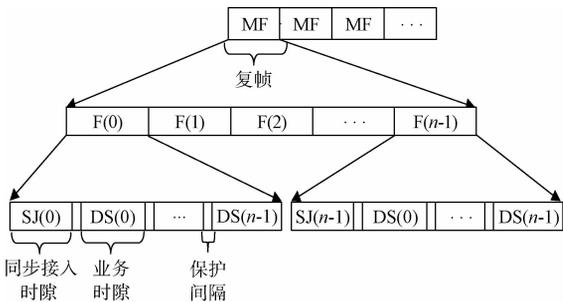


图 1 TDMA 帧结构

节点在 SJ 时隙发送同步报文,同步报文的主要信息字段如图 2。其中源节点代表中心主节点,等级为 0,其他节

点的等级根据距离源节点的跳数来增加。本节点和源节点的时隙号分别是自己占据 SJ(x)的时隙号  $x$  和同步源节点占据 SJ(y)的时隙号  $y$ 。邻节点时隙表记录自己 and 一跳邻节点的 SJ 时隙占用情况,这样节点在接入前侦听到两跳范围内节点的 SJ 占据情况就可以选择未被占用的时隙中时隙号最小的 SJ,从而避免冲突。但如果多个节点开机相差时间很短则会同时选择未被占用的时隙中时隙号最小的 SJ,尤其是在源节点开机之后的侦听期间,如果有一个以上的节点开机,则一定都会选择占据 SJ(1)时隙而导致冲突。针对冲突,本文采取退避算法:节点在一段时间内若收不到同步确认报文,就随机退避若干个子帧的时间重新发送同步报文,直到成功入网。

Ptype (报文类型)	ID (本节点ID号)	Slot (本节点时隙号)	Lcycle (本节点生存周期)	Grade (本节点等级)
S-ID (源节点ID号)	S-Slot (源节点时隙号)	S-Lcycle (源节点生存周期)	S-Grade (源节点等级号)	N-Slortable (邻节点时隙表)

图 2 同步报文主要结构

## 4 基于生存周期的同步过程

本文取网络内最大允许节点数  $n$  为 16,针对此条件下的帧结构,在不同网络状态下对同步过程进行描述。任何节点开机后先进入侦听状态,生存周期从 0 开始累加计数,每经历一个复帧的时间计数器加 1。节点如果一直没收到任何其他节点的同步报文,则计数到  $m$  时成为源节点进行组网过程,否则进行节点入网过程。 $m$  作为侦听时间的取值不能太大,否则会增加节点接入冲突的概率。

### 4.1 源节点组网

节点在开机的  $m$  个生存周期时间内未收到同步报文则默认自己为源节点,选择同步接入时隙 SJ(0),更新邻节点占用时隙表(在时隙表 SJ(0)处标明自己的 ID 表示已被占用),设置自身等级为 0 然后发送同步报文。另外节点内部有一个以一个复帧长度为周期的循环计时器,当计数到 SJ(0)时隙的起始时刻发送同步报文。发送同步报文后,源节点在 SJ(0)以外的同步接入时隙都只进行侦听,如果仍未收到其他节点的同步报文,则此时源节点的生存周期不能再无意义的继续计数,要保持计数值  $m$  直到收到同步报文时才继续增加。

### 4.2 节点入网

节点开机后如果在侦听期间收到同步报文则进入入网过程,开始同步判决。生存周期作为判决的优先标准,决定了节点的同步方向。如果节点的生存周期小于发送同步报文节点的生存周期则要向其同步,反之则放弃同步。当生存周期都相同时则继续选择网络数量进行判别,具体判决流程如图 3 所示。节点在判决时应先取源寄存器的值进行判决,如果为空才取本寄存器的值继续判决,因为如果节点

已经有同步源,利用源节点的生存周期作为判决标准会比自己的生存周期更加稳定。另外许多同步协议在传递同步信息是将自己的本地时间值传送,而为了保证同步精确,时间值往往需要具体到微秒级。其实时隙、子帧、复帧的长度都已经确定,所以源节点只传递同步接入时隙号,接受节点就可以确定源节点发送时的本地时间,然后调整自己的时隙号进行同步,这样就舍去了时间值的传递,减小了同步时的信息开销。

下面具体介绍节点收到同步报文后的同步过程:当同步报文的 S-Lcycle 值大于本寄存器的值(因为刚开机所以源寄存器一定为空),节点便向发送同步报文的节点粗同步。节点先通过同步报文获取源节点的本地时隙号 Slot,并把自己的本地时钟调整到 Slot 的起始时刻。如图 4,其中  $t_0 \sim t_{15}$  代表同步接入时隙 SJ(0)~SJ(15) 的起始时刻。假设当前本节点的本地时间是  $t_6$ ,收到源节点同步报文的时隙号是 1,所以粗同步就是将本地时钟由  $t_6$  调整到 SJ(1) 的起始时刻  $t_1$ 。但由于节点内部的传输时延及空中传播时延等因素<sup>[5]</sup>,节点调整到  $t_1$  时源节点实际上已经越过  $t_1$  到达  $t(x)$  时刻,这样造成的时间差为  $\Delta t = t(x) - t_1$ 。

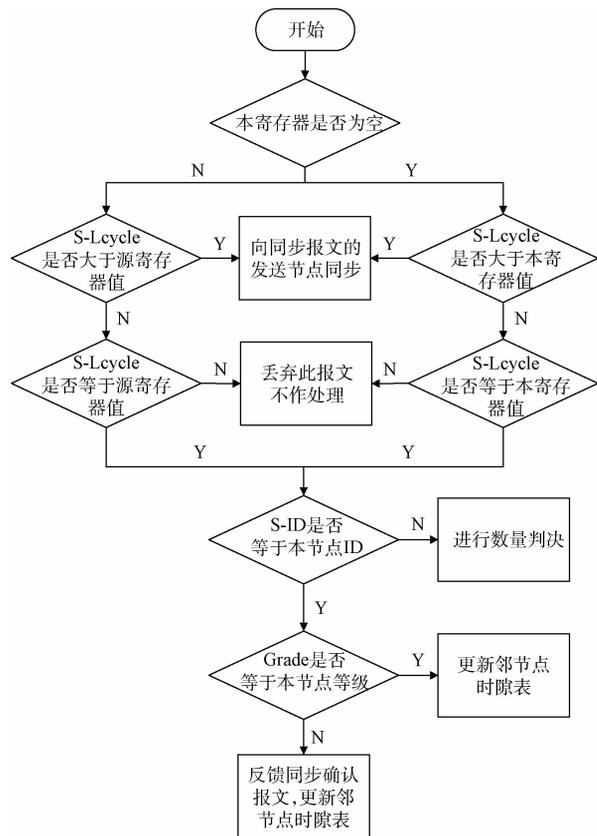


图3 同步判决流程

粗同步后,节点将同步报文的 S-Lcycle 值更新到自己的源寄存器中,并通过邻节点占用时隙表选择同步接入时隙,假设选择 SJ(3)。时间到达 SJ(3) 的起始时刻  $t_3$  时,节点发送

自己的同步报文,源节点在  $t(y)$  时刻接受到。考虑到节点间的时延在短时间内基本相同,因此源节点通过同步接入时隙 SJ(3) 的起始时刻  $t_3$  和  $t(y)$  可以估算出两个节点间的时间差  $\Delta t = [(t_4 \Delta t_5) / 2]$ ,将此值作为修正时间填入同步确认报文后立刻反馈。同步确认报文结构如图 5,它记录了报文类型、源节点、目的节点的地址以及修正时间。节点收到同步确认报文将自己的本地时钟增加  $\Delta t$ ,完成时间细同步。

通过以上同步方法,即使距离源节点一条距离以上的节点收不到它的同步报文,也可以将已经和源节点同步的节点作为自己的主节点并向其同步,最终实现多跳节点收敛到源节点的时间。

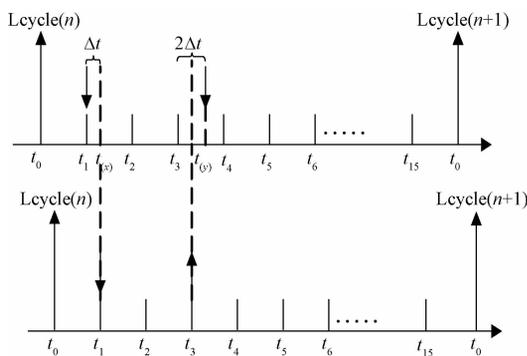


图4 时间同步示意图

Ptype (报文类型)	ID (本节点ID号)	AckID (回复节点号)	Timeoffset (修正时间)
-----------------	----------------	------------------	----------------------

图5 同步确认报文结构

### 4.3 源节点失效后的重组网

源节点一旦失效,时间同步协议需要能够快速恢复并不影响其余节点正常工作。本文利用生存周期重新竞争的原则,在原来的时隙分配基础上自动进行新源节点的选取。当源节点退网后,所有子节点都不能收到源节点的同步报文,因此它们的源寄存器的值也不会增加。如果源寄存器的值持续 3 个复帧一直不增加,则子节点将源寄存器的值清空,利用本寄存器的值重新竞选同步源节点。

图 6 是一个已经同步的网络,节点 A 为源节点,它的子节点为 B,C,D,其中 B 和 C 距离两跳不能通信。假设节点 A,B,C,D 的同步接入时隙分别是 SJ(0)~SJ(3)。当节点 A 生存周期计数到 100 时从拓扑中消失,在第 101 个生存周期的 SJ(0) 时隙内,节点 B,C,D 都没有收到同步报文,因此源寄存器的值仍为 100。节点 B,C,D 将从第 101 个生存周期的 SJ(0) 起始时刻开始一个 3 个复帧的计时,当第 104 个生存周期的 SJ(0) 起始时刻到来时计满 3 个周期。此时节点 B,C,D 的源寄存器全部清空。等到 SJ(1) 起始时刻到来时 B 将自己等级提高到 0 作为源节点发送同步报文,S-Lcycle 值填入本寄存器的值。节点 C 不能与节点 B

交互同步报文,所以在 SJ(2)起始时刻也作为源节点发送自己的同步报文。节点 D 因为在 SJ(3)起始时刻才发送同步报文,因此在侦听期间收到了 B 和 C 的同步报文,通过选择最大生存周期的节点作为真正的同步源。重组网期间节点的同步接入时隙占用情况保持不变,避免了时隙的接入冲突,实现了重组网的快速灵活性。

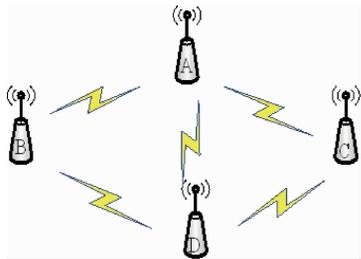


图 6 重组网示意图

### 5 仿真分析

仿真利用 OPNET Modeler 14.5 在无噪声条件下进行,网络拓扑在仿真期间处于静态<sup>[12]</sup>。仿真设置节点之间的距离固定为 500 m,同步接入时隙和业务时隙长度(加保护间隔)分别为 0.15 ms 和 0.5 ms,收发信机速率为 10 Mbps,节点开机时的侦听复帧数  $m$  为 5。

同步协议的优劣性主要体现在同步收敛时间和同步精度。本协议的同步收敛时间就是节点从接入网络到完成细同步的时间,而节点如果接入时遇到冲突就会产生退避,收敛时间也因此延长。另外都会有不同的同步收敛时间。所以仿真将分别在源节点开机侦听的组网阶段和源节点已经组网的迟入网阶段设置不同的接入冲突情况,进行同步收敛时间和同步精度的测试。

#### 5.1 同步收敛时间

首先在迟入网阶段,通过改变节点的开机时间,分析节点之间发生接入冲突时对同步收敛时间的影响。网络层拓扑如图 7,节点 1~14 距离节点 0 一跳距离,设置节点 0 开机时间为 0.1 s(源节点),其他节点的时间均设置比 0.1 s 大。



图 7 网络层拓扑

仿真绘制统计图如图 8 所示,其中五角星标注折线是节点之间没有接入冲突的情况,所有节点的同步收敛时间

基本保持在 0.1 s 左右。由三角标注的折线可以看出节点 1~4 有接入冲突,它们的同步收敛时间都有所增加。冲突节点的数量增加后,大部分节点的同步收敛时间明显增大,说明了虽然节点退避是随机的,但冲突数量的上升仍会使整个网路的平均同步收敛时间延长。当冲突数量达到最大值 14 时,此时最大的同步收敛时间为 0.26 s 左右。

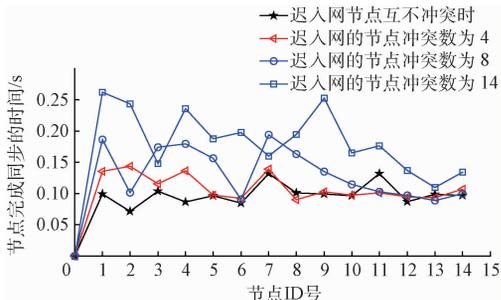


图 8 迟入网状态下的同步时间

保持拓扑不变,将节点的开机时间设置在源节点的开机侦听期间,此时节点处于源节点开机侦听的组网阶段。分别取 1、8、14 个节点进行仿真,从统计图 9 可以看到当组网节点数量为 1 时,同步收敛时间在 0.55 s 左右,此时虽然没有接入冲突,但远大于迟入网阶段下的同步收敛时间,因为在组网阶段源节点的开机侦听延长了节点的接入时间。当达到最大组网数量 14 时,所有节点中最长的同步收敛时间在 0.8 s 左右。因此节点即使在接入冲突概率很大的组网状态下,同步收敛时间也很小。

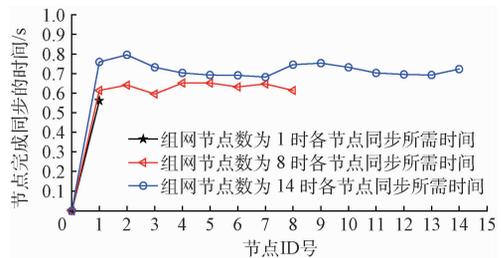


图 9 组网状态下的同步时间

#### 5.2 同步误差分析

将网络拓扑内的节点数目设置为 7,并依次在距离源节点最大跳数为 1~6 的 6 种网络拓扑下进行仿真如图 10 所示。仿真后将同步误差统计成图 11。从距源节点最大跳数为 1 的曲线可以看出当节点距离源节点 1 跳范围内时,同步误差基本为 0。随着距离源节点跳数的增加,同步误差增大,且曲线斜率逐渐升高表明误差增大的幅度也是随着跳数增加而上升。当节点 6 距离源节点 6 跳时,同步误差大概为 0.37 ms。

### 6 结 论

仿真证明基于生存周期的同步方法虽然在接入冲突情



图10 6种网络拓扑

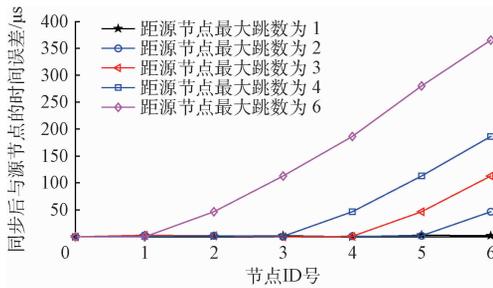


图11 距源节点不同跳数的同步误差

况下同步时间有所延长,但仍然可以实现快速同步。同时本方法作为一种主从式同步,能够在主节点缺失后快速完成自动重组,具有较强的稳定性和灵活性。对于同步精度,在距离源节点跳数较少的范围内误差能够控制在微秒级别,但随着跳数的增加同步误差有所上升,而这也是主从式同步法存在的通病。所以对主从式多跳误差积累进行消除将极大改善主从式同步的性能,这也是下一步将重点研究的内容。

## 参考文献

- [1] 王波,叶晓慧,赵玉亭,等. 自组织网络时钟同步研究综述[J]. 计算机科学, 2010, 37(5): 30-33.
- [2] 钱静丰. 无线自组织网协同同步方法与信道接入技术研究[D]. 厦门: 厦门大学, 2014.
- [3] 陈珍萍,李德权,黄友锐,等. 无线传感器网络混合触发一致性时间同步[J]. 仪器仪表学报, 2015, 36(10): 2193-2199.

- [4] 杨勇. 一种等级化分布式时间同步算法[J]. 现代电子技术, 2013, 36(17): 37-40.
- [5] 漆璐,陈旻. Ad Hoc 网络时钟同步研究[J]. 计算机应用研究, 2011, 28(12): 4702-4704.
- [6] KHEDIRI S E, NASRI N, HEDFI H, et al. An improved TPSN algorithm for time synchronization in wireless sensors networks [C]. International Conference on Distributed Computing Systems Workshops, IEEE, 2013: 279-284.
- [7] XU N, ZHANG X, WANG Q, et al. An improved flooding time synchronization protocol for industrial wireless networks [C]. International Conference on Embedded Software and System, IEEE, 2009: 524-529.
- [8] 杨琦,石江宏,陈辉煌. 无线自组织网络分布式时隙同步算法[J]. 系统工程与电子技术, 2011, 33(4): 888-893.
- [9] 徐明霞,赵民建,董芳,等. 基于参数解析的 Ad hoc 网络时隙同步方法[J]. 浙江大学学报, 2005, 39(6): 901-905.
- [10] 冯文江. 实现自组网互同步的一种单源同步法[J]. 电路与系统学报, 2009, 14(3): 23-27.
- [11] 庄晓燕,王厚军. 基于卡尔曼滤波器的 IEEE1588 时钟同步算法[J]. 电子测量与仪器学报, 2012, 26(9): 747-751.
- [12] 蓝李丹,杨木清,陈军东,等. OPNET 软件在协议性能分析中的应用[J]. 国外电子测量技术, 2007, 26(5): 66-68, 71.

## 作者简介

金瑞,在读研究生,学员,主要研究方向为无线自组网通信。

E-mail:394744537@qq.com