

一种采用时隙对准方式的 TDMA 自组网同步协议

金 瑞 刘作学

(装备学院 北京 101400)

摘 要 通过对 TDMA 方式下的同步协议 STS 和 TISS 进行研究,提出一种基于时隙对准方式的 TDMA 自组网同步协议 MFSS。该协议以工作周期为自组网节点之间同步的标准,在节点初入网时采用双向交互和时隙对准方法,消除了传输时延误差和初始时间偏差,从而实现了快速初始同步;随后通过监测过程保证了节点之间产生的时钟漂移误差可自适应控制,同时减小了重新同步带来的开销。仿真结果证明,相比于 STS 协议和 TISS 协议,MFSS 协议在同步收敛速度、同步精度以及同步开销上都取得了更好的性能。

关键词 时间同步,工作周期,时钟漂移误差,监测过程

中图分类号 TN929 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.014

Synchronization Protocol of TDMA Ad hoc Network Based on Time Slot Alignment

JIN Rui LIU Zuo-xue

(Academy of Equipment, Beijing 101400, China)

Abstract Through researching the TDMA time synchronization protocols STS and TISS, this paper proposed a TDMA Ad hoc network synchronization protocol MFSS based on time slot alignment. The MFSS protocol uses the work cycle as the standard of synchronization among the nodes of Ad hoc network. When the node accesses the network firstly, the two-way interaction and time slot alignment are used to eliminate the transmission delay error and initial time deviation, so an initial synchronization can be completed quickly. Then, the clock drift error among nodes can be controlled by the monitoring process, and the overhead of resynchronization is also reduced. The simulation results show that compared with STS protocol and TISS protocol, the MFSS protocol achieves better performance in terms of synchronous convergence speed, synchronization accuracy and synchronization overhead.

Keywords Time synchronization, Work cycle, Clock drift error, Monitoring process

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,网络信道由多个节点共享,因此协调节点访问信道的介质访问控制机制(MAC)一直是 Ad hoc 网络的研究热点。其中,基于时分多址(Time Division Multiple Access, TDMA)的分配接入机制因具有高网络吞吐量、传输时延可控等优点,被广泛应用于自组网的 MAC 层^[1]。TDMA 方式下,自组网节点之间应保持统一的时间基准,以时隙为单位进行消息收发,因此必须保持严格的同步。目前,已有很多传统的同步协议被提出,其中 NTP(Network Time Protocol)^[2]通过双向交互机制基本消除了同步消息的传输时延;TPSN(Timing-sync Protocol for Sensor Networks)^[3]利用分层结构实现了多跳节点之间的同步;FTSP(Flooding Time Synchronization Protocol)^[4]采用线性拟合算法补偿了时钟漂移误差。这些协议在传递同步消息时都以时间戳记录具体的时间值,但时间戳易受异常数值影响,而且在同步精度要求较高时,时间值所占字节较多。在 TDMA 方式下,由于时帧结

构具有周期性和固定性,只通过时隙号就可以计算出具体的时间值,再利用时隙对准即可实现同步,节省了时间戳开销^[5],因此越来越多的 TDMA 同步协议^[6-8]将这种时隙对准方式与传统同步协议进行结合与改进。本文针对两种典型的 TDMA 同步协议存在的缺陷,设计了一种具有误差监测功能的同步协议 MFSS(Monitor and Fit-Skew Synchronization)。

本文第 2 节对两种 TDMA 同步协议进行介绍,通过分析其中的缺陷,提出 MFSS 协议的设计思路;第 3 节详细论述了 MFSS 协议的同步过程;第 4 节通过对比仿真分析了 MFSS 协议的同步性能;最后对全文进行总结。

2 相关工作

文献[7]在 FTSP 协议的基础上,利用时隙对准方式设计了一种适用于 TDMA 方式的同步协议 STS(Slot Time Synchronization)。该协议通过实验证明了网络节点的时钟漂移在短时间内基本不变,因此时钟模型可简化为式(1):

$$L(t) = (1 + \alpha)t + L(t_0) \quad (1)$$

其中, t 代表基准时间; α 代表时钟漂移,理想情况下时钟漂移

到稿日期:2017-05-01 返修日期:2017-09-01

金 瑞(1993—),男,硕士生,主要研究方向为无线自组网 MAC 层协议,E-mail:394744537@qq.com;刘作学(1962—),男,硕士,教授,主要研究方向为无线军事信息处理,E-mail:lzx626@sohu.com(通信作者)。

为 0,但实际上均存在时钟漂移,且不同节点的时钟漂移值也有微小偏差,由此引起的误差被称为漂移误差^[9]; $L(t_0)$ 代表节点的初始本地时间,由于自组网节点开机时间不一致,因此也会存在初始时间偏差。

STS 协议中,节点在发送同步消息时传递当前的时隙号 ASN(Absolute Slot Number),收到消息的节点立刻记录下本地时间 t_R ,并根据时隙号 ASN 计算发送节点的发送时间 t_S 。如果将接收节点的本地时间看作式(1)中的标准时间 t ,将发送节点的本地时间看作 $L(t)$,则将 t_S 和 t_R 相减,得到式(2):

$$t_S - t_R = \alpha t_R + L(t_S - t_R) \quad (2)$$

其中, α 和 $L(t_S - t_R)$ 分别表示发送节点相对于接收节点的时钟漂移偏差和初始时间偏差。接收节点在收集多组 (t_S, t_R) 后,可以通过一阶线性拟合算法计算出 α 和 $L(t_S - t_R)$,以补偿自己的时钟漂移,如此即可完成向发送节点的同步。

STS 协议中,节点对时钟漂移进行了补偿,在外界环境干扰不大的情况下,可以和同步节点保证较长时间的同步,延长了重新同步的周期,减小了同步开销。其中,重新同步周期值是根据 FTSP 协议中实验取得的最优值 30s 而设置的固定值,但在实际应用中节点之间的漂移误差会随着外界环境随机变化,不会严格符合线性拟合变化,因此仍有可能在重新同步前就产生较大的误差。另外,节点在同步时必须收集若干组 (t_S, t_R) 数据后才进行拟合计算,导致同步收敛速度较慢,且没有考虑传播时延。文献[8]提出了一种基于 NTP 双向交互机制的时隙同步协议 TISS(Two-way Interaction Slot Synchronization),同步过程如图 1 所示。

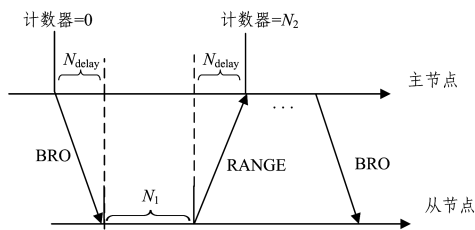


图 1 TISS 协议的同步过程

Fig. 1 Synchronization process of TISS protocol

主节点为同步源,在属于自己的时隙到来时发送记录了当前时隙号的同步消息 BRO,并启动一个从 0 开始的计时。从节点为被同步节点,在收到同步消息后将自己的时间调整到同步消息中时隙号对应的时间,但此时仍存在同步消息的传输时延误差 N_{delay} 。从节点随机等待一段时间 N_1 后发送一条回复消息 RANGE,该消息记录了等待时间 N_1 。主节点在计数到 N_2 时收到 RANGE 消息并提取 N_1 ,可认为短时间内 BRO 消息和 RANGE 消息的传输时延相同,因此主节点通过式(3)计算出传输时延。

$$N_{delay} = \frac{(N_2 - N_1)}{2} \quad (3)$$

主节点在下一个周期发送 BRO 消息时将计算的 N_{delay} 值返回给从节点,使其调整后同步。这种双向交互机制基本消除了传输时延误差,实现了节点的快速同步;但其缺陷在于没有对时钟漂移进行补偿,同时为了抑制初始同步后新的误差积累,节点之间只能在每个 TDMA 帧周期重新同步,这不仅导致同步开销过大,而且周期性修改本地时间也会影响到 TDMA 时序的连续性。

针对上述两种协议的不足,本文设计了 MFSS 协议。在初始同步时采用双向交互机制实现节点之间的快速同步,并在之后的每个 TDMA 周期通过监测过程计算节点之间的漂移误差,逼近最大允许误差时再重新同步,重新同步的时间将根据节点之间最新的漂移误差动态调整。为延长重新同步的时间,MFSS 协议还利用监测的漂移误差进行线性拟合,以补偿时钟漂移。相比于 STS 协议,MFSS 协议可以加快同步收敛的速度,消除传输时延误差,解决固定重新同步周期带来的误差隐患;相比于 TISS 协议,MFSS 协议可以减小同步开销。

3 MFSS 协议

3.1 MFSS 协议的初始同步过程

MFSS 协议的 TDMA 帧结构如图 2 所示。一个同步时隙和后面若干个业务时隙组成一个子帧, n 个子帧构成的一个复帧作为一个 TDMA 帧周期,其中 n 为网络节点的数量。节点开机入网后固定占据一个同步时隙,作为发送自身同步消息的时隙。保护间隔作为节点之间同步后仍存在微小误差的缓冲时间。

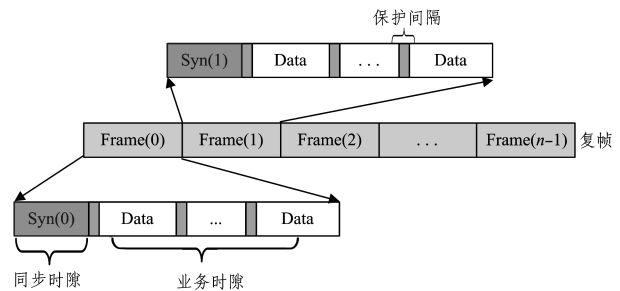


图 2 MFSS 协议的 TDMA 帧结构

Fig. 2 TDMA frame structure of MFSS protocol

协议采用入网后竞争成为主节点的方式,将节点开机后的工作周期数作为选取主节点和判断是否同步的标准。工作周期即图 2 中一个复帧的长度。节点开机后进行侦听,同时对工作周期进行计数,累计到某个值时如果仍没收到其他节点的同步消息,则默认自己为主节点并设置等级为 0,这样开机最早的节点会竞争成为主节点。主节点占据同步时隙 Syn(0)后发送同步消息,其中的主要内容如图 3 所示。

Mtype (消息类型)	ID (本节点 ID 号)	Slot (本节点的同步时隙)	Wcycle (本节点工作周期)	Grade (本节点等级)
Status (本节点状态)	S-ID (同步节点 ID 号)	CSD (粗同步时间差)	N-Slortable (邻节点时隙表)	

图 3 同步消息的主要内容

Fig. 3 Main content of synchronization messages

图 3 中, Wcycle 记录了本节点的工作周期,用于与其他节点进行比较。Slot 记录了本节点占用的同步时隙号。邻节点时隙表 N-Slortable 记录了所有同步时隙被节点占据的情况。同步消息可直接在 MAC 层产生或解析,并设置它在消息队列中发送的优先级最高,这样传输时延误差只包括 MAC 层生成消息到物理层发送消息的时延、消息在空中的传播时延以及被接收机完全接收并送到 MAC 层解析的时延。

在主节点选定后开机的节点成为从节点。从节点收到主节点的同步消息后解析 Wcycle,如果 Wcycle 大于自己的工

作周期,则选择向主节点同步,并设置等级为1。1级节点的工作周期调整到 W_{cycle} ,从而同步更远的节点(向1级同步的节点的等级为2,以此类推)。通过解析 N-Slortable,1级节点选择未占用的同步时隙,并更新时隙表。节点的同步过程如图4所示。假设 $t_0, t_1 \dots$ 表示同步时隙 $S_{yn}(0), S_{yn}(1) \dots$ 的起始时刻。主节点在 t_0 发送同步消息,1级节点收到后先将本地时间调整至 t_0 完成粗同步,但与主节点之间仍存在传输时延 Δd_0 。假设1级节点选择同步时隙 $S_{yn}(2)$,并在 t_2 时刻发送回复消息(内容和同步消息相同)。主节点记录下回复消息的到达时间 t_b ,解析 Slot 值后得知1级节点的发送时间是 t_2 ,因此传输时延误差的修正值 Δof_1 可以根据式(4)计算。主节点计算出 Δof_1 后,立刻填入反馈消息的 Timeoffset 字段并反馈给1级节点,反馈消息的主要内容如图5所示。1级节点在 t_c 时刻收到消息后调整本地时间,完成初始同步。

$$\Delta of_1 = \frac{t_b - t_2}{2} \quad (4)$$

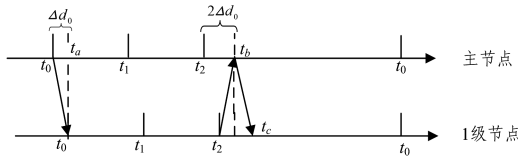


图4 同步过程图

Fig. 4 Time synchronization process

Mtype (帧类型)	ID (本节点ID号)	AckID (反馈节点ID号)	Timeoffset (修正时间)	RESYN (重新同步标示符)

图5 反馈消息的主要内容

Fig. 5 Main content of feedback messages

3.2 MFSS 协议的监测过程

3.1节的初始同步过程未考虑主节点和1级节点之间的漂移误差。下面通过对初始同步过程进行误差分析,介绍MFSS协议的监测过程。仍以3.1节的假设为例,如图6所示,主节点在 t_0 时刻发送的同步消息被1级节点收到并粗同步,此时主节点和1级节点的时间差除了传输时延 Δd_0 外,还有在这段时间内主节点相对于1级节点的漂移误差 $\Delta S(\Delta d_0)$ 。当1级节点在时间 t_2 发送回复消息时,主节点和1级节点的时间差增加了从 t_0 到 t_2 这段时间内的漂移误差 $\Delta S(t_0, t_2)$ 。回复消息被主节点收到时,又经历了1级节点到主节点的传输时延 $\Delta d_0'$ 以及这段时延内主节点相对于1级节点的漂移误差 $\Delta S(\Delta d_0')$,这样式(4)计算出的修正值 Δof_1 变为:

$$\begin{aligned} \Delta of_1 &= \frac{\Delta d_0 + \Delta d_0' + \Delta S(\Delta d_0') + \Delta S(\Delta d_0) + \Delta S(t_0, t_2)}{2} \\ &= \Delta d_0 + \Delta S(\Delta d_0) + \frac{\Delta S(t_0, t_2)}{2} \end{aligned} \quad (5)$$

其中, Δd_0 和 $\Delta d_0'$ 在短时间内基本不变。主节点立刻反馈修正值,1级节点若在 t_2' 时刻收到该值,则与主节点的时间差又增加了 $\Delta t(t_2, t_2') = 2[\Delta S(\Delta d_0')]$ 。这样,从粗同步时刻 t_0 到当前时刻 t_2' ,1级节点和主节点的累积时间差 ΔV_1 为:

$$\begin{aligned} \Delta V_1 &= \Delta d_0 + \Delta S(\Delta d_0) + \Delta S(t_0, t_2) + 2\Delta S(\Delta d_0') \\ &= \Delta d_0 + 3\Delta S(\Delta d_0) + \Delta S(t_0, t_2) \end{aligned} \quad (6)$$

1级节点补偿修正值 Δof_1 后仍与主节点存在时间差 ΔR_1 ,如式(7)所示。因此,1级节点完成初始同步后虽然基本消除了传输时延,但仍与主节点之间存在非常小的漂移误差。

$$\begin{aligned} \Delta R_1 &= \Delta d_0 + 3 \times \Delta S(\Delta d_0) + \Delta S(t_0, t_2) - \Delta of_1 \\ &= 2 \times \Delta S(\Delta d_0) + \frac{\Delta S(t_0, t_2)}{2} \end{aligned} \quad (7)$$

下一个工作周期,主节点仍在同步时隙 $S_{yn}(0)$ 的起始时刻(如图6中的 t_3)发送同步消息,假设此时1级节点的本地时间为 t_3' ,则它从上个周期完成初始同步的 t_2' 时刻到当前 t_3' 时刻,又与主节点产生了漂移误差 $\Delta S(t_2', t_3')$ 。1级节点在 t_4 时刻接收到同步消息时又经历了传输时延 Δd_1 ,此时1级节点不再重新向主节点粗同步。1级节点将同步消息中时隙对应的的时间值 t_3 加上初始同步过程中的修正值 Δof_1 之后再减去本地时间 t_4 ,计算得到的差值 Δw_2 如式(8)所示。其中, ΔR_1 是上个周期初始同步后仍存在的误差,将短时间内的 Δd_0 和 Δd_1 视为相同。

$$\begin{aligned} \Delta w_2 &= \Delta R_1 + \Delta S(t_2', t_3') - \Delta d_1 + \Delta of_1 \\ &= 2 \times \Delta S(\Delta d_0) + \frac{\Delta S(t_0, t_2)}{2} + \Delta S(t_2', t_3') - \Delta d_1 + \\ &\quad \Delta d_0 + \Delta S(\Delta d_0) + \frac{\Delta S(t_0, t_2)}{2} \\ &= \Delta S(t_2, t_3') + \Delta S(t_0, t_2) + \Delta S(\Delta d_0) - \Delta d_1 + \Delta d_0 \\ &\approx \Delta S(t_0, t_4) \end{aligned} \quad (8)$$

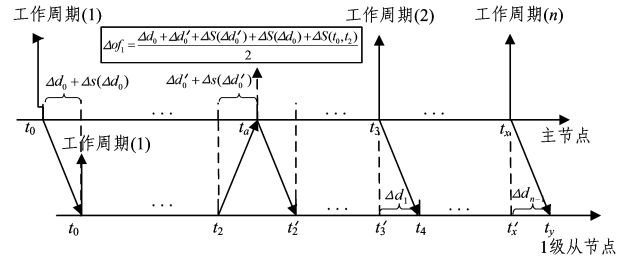


图6 监测过程

Fig. 6 Monitoring process

依次类推,假设第 n 个工作周期内主节点在它的同步时隙起始时刻 t_x 发送同步消息,此时1级节点对应的本地时间为 t_x' 。经过传输时延 Δd_{n-1} ,1级节点在本地时间 t_y 收到同步消息,然后将其中时隙对应的的时间值 t_x 加上初始同步过程中的修正值 Δof_1 ,再减去本地时间 t_y ,得到的 Δw_n 值如式(9)所示。其中,传输时延值 Δd_{n-1} 相对于 Δd_0 的变化,主要由节点位置变化引起的空中传播时延决定,当节点之间静止或运动速度不是很快时,变化量十分微小,可以视为0。

$$\begin{aligned} \Delta w_n &= \Delta R_1 + \Delta S(t_2', t_x') - \Delta d_{n-1} + \Delta of_1 \\ &= 2 \times \Delta S(\Delta d_0) + \frac{\Delta S(t_0, t_2)}{2} + \Delta S(t_2', t_x') - \\ &\quad \Delta d_{n-1} + \Delta d_0 + \Delta S(\Delta d_0) + \frac{\Delta S(t_0, t_2)}{2} \\ &= \Delta S(t_2, t_x') + \Delta S(t_0, t_2) + \Delta S(\Delta d_0) - \Delta d_{n-1} + \Delta d_0 \\ &\approx \Delta S(t_0, t_y) \end{aligned} \quad (9)$$

可以看出,1级节点在每个周期收到主节点同步消息后求出的差值 Δw ,近似于从完成初始同步到目前主节点相对于1级节点的漂移误差。因此,监测过程如下:从节点向主节点初始同步后,每个工作周期收到主节点新的同步消息时,利用同步消息中时隙对应的的时间、初始同步中的修正值 Δof 和

收到同步消息时的本地时间计算出漂移误差 Δw 。从节点可以根据需求设置允许漂移误差的最大阈值 θ ,当 Δw 值超过阈值 θ 时,从节点在下一个周期重新向主节点同步。为了延长重新同步的时间,从节点还可以利用每个工作周期计算的 Δw 值拟合漂移误差曲线,以补偿自身的时钟漂移。本文采用最小二乘法拟合漂移误差曲线。从节点将漂移误差值 Δw_i 和本地时间 t_i 值作为一组离散数据 $(t_i, \Delta w_i) (i=1, 2, \dots, n, n$ 表示工作周期数)并保存,收集 m 组后进行一阶线性拟合,即 $\Delta w(t) = k_0 + k_1 t$ 。其中, k_1 和 k_0 分别代表主节点和 1 级节点的时钟漂移偏差和时间差(相当于式(2)中的 α 和 $L(t_s - t_R)$),其计算公式如式(10)^[10]所示。本文设定 $m=6$,1 级节点计算出 k_1 值和 k_0 值后,对自身时钟的时钟漂移和本地时间进行补偿。通过补偿,1 级节点和主节点的时钟漂移近似相同,降低了两个节点之间漂移误差的增长速度。

$$k_1 = \frac{\sum_{i=1}^m t_i \times \Delta w_i - m \times \bar{t}_i \times \overline{\Delta w_i}}{\sum_{i=1}^m t_i^2 - m \times \bar{t}_i^2} \quad (10)$$

$$k_0 = \overline{\Delta w_i} - k_1 \times \bar{t}_i$$

3.3 MFSS 协议的重新同步过程

从节点重新向主节点同步时,如果再从粗同步开始,则又会产生传输时延误差。因此,从节点不向主节点粗同步,而是计算同步消息中 Slot 对应的的时间值和本地时间的差值 $\Delta(t_{local} - t_{slot})$ 。从节点在自己的同步时隙到来时,将差值 $\Delta(t_{local} - t_{slot})$ 放入回复消息的粗同步时间差 CSD 字段中。主节点收到回复消息后,用 Slot 对应的的时间值减去 CSD 值(模拟从节点在重新同步时向主节点粗同步的情况下,从节点在发送回复消息时的时间),再利用式(4)计算出修正值 Δof ,并反馈给从节点。这样,即使不进行粗同步调整,从节点也可以重新向主节点同步。

3.4 针对多跳同步的改进

1 级节点向主节点初始同步后,从下个工作周期开始在自己的同步时隙发送同步消息,从而同步距离主节点更远的节点。但在多跳同步下,这种监测过程仍存在如下问题:如图 7 所示,节点 A 为主节点,节点 B 向节点 A 同步,节点 C 向节点 B 同步,节点 A 和节点 C 相距两跳范围,无法直接通信,节点 A、节点 B 和节点 C 的等级分别是 0,1,2。若节点 B 在某个工作周期监测到与节点 A 的漂移误差为 $\Delta w_1 < \theta$,同一个周期假设节点 C 监测到与节点 B 的漂移误差 $\Delta w_2 < \theta$,但节点 C 相对于节点 A 的漂移误差等于 $\Delta w_1 + \Delta w_2 > \theta$,则节点 A 和节点 C 之间的时钟误差有可能会大于保护间隔时间。在 TDMA 中,每个节点应至少保持和两跳范围内的邻节点时间同步,才能避免数据发送冲突,因此两跳范围节点之间也要保持漂移误差不能超过阈值 θ 。



图 7 多跳网络拓扑

Fig. 7 Multi-hop network topology

图 7 中,1 级节点 B 可以从主节点 A 收到自身的修正值 Δof_B ,又因为 2 级节点 C 向它同步,所以节点 B 同时也可以

计算出节点 C 的修正值 Δof_C 。在监测过程中,节点 C 是通过修正值 Δof_C 、节点 B 同步信息中时隙号对应的的时间以及本地时间来计算出与节点 B 的漂移误差 Δw_{BC} ,但节点 B 同样也可以通过修正值 Δof_C 、节点 C 同步信息中时隙号对应的的时间以及本地时间计算出与节点 C 的时钟漂移误差 Δw_{CB} ,且 $\Delta w_{BC} = \Delta w_{CB}$ 。这样,1 级节点 B 即可以计算出主节点 A 相对于自身的漂移误差 Δw_{AB} ,也可以计算出 2 级节点 C 相对于自身的漂移误差 Δw_{BC} ,进而节点 B 将 Δw_{AB} 和 Δw_{BC} 相加,以判断其是否超过 θ 。若超过,则节点 B 会给自己等级高的节点 C 发送反馈消息,其中在重新同步标识符 RESYN 字段用 1 表示需要重新同步。节点 C 收到反馈消息后解析 RESYN,如果为 1 就在下个工作周期重新向节点 B 同步。因此,通过中间节点的双向监测,可以保证两跳节点之间的时钟漂移误差不会超过阈值 θ 。

3.5 MFSS 协议的流程图

MFSS 协议的完整流程如图 8 所示。

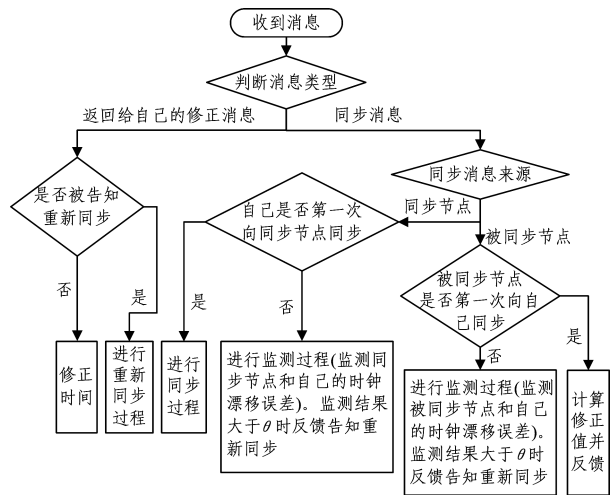


图 8 MFSS 协议的整体流程图

Fig. 8 Overall flowchart of the MFSS protocol

其中,节点在完成初始同步过程后就对自己的同步节点进行监测。当某个周期监测到时间误差超过阈值时,则在下一个周期进行重新同步。另一方面,它也会同步其他节点并监测这些被同步节点,当某个周期监测到被同步节点和同步节点的时间误差和超过阈值时,它将通知被同步节点重新同步。

4 仿真验证及分析

对 MFSS 协议、STS 协议以及 TISS 协议进行仿真,仿真场景(见图 9)及参数设置(见表 1)均相同。节点 0、节点 1 和节点 2 呈链状拓扑,设置节点 0 和节点 2 不能直接通信。时钟模型采用式(2),阈值只在 MFSS 协议中配置。

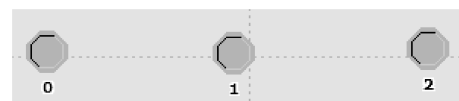


图 9 仿真场景

Fig. 9 Simulation scenario

首先,对比 3 种协议的重新同步次数:STS 协议中,节点 1 和节点 2(节点 0 开机最早,是主节点,不向其他节点同步)的重新同步次数最少,都是 20 次,这是因为 STS 协议的重新

同步周期固定设置为 30 s,但这种周期下的同步精度不一定最优,之后会在同步精度的仿真结果中进一步分析;MFSS 协议中节点 1 和节点 2 的重新同步次数也比较少,分别为 85 次和 127 次;而在 TISS 协议中,节点 1 和节点 2 的重新同步次数都多达 4000 次,这是因为节点在初始同步后还会在每个 TDMA 周期重新同步,当周期很短时,大量的重新同步将会带来不必要的开销。MFSS 协议由于是通过监测过程,发现漂移误差超过阈值时才重新同步,因此相比于 TISS 协议,其明显减少了重新同步的次数,降低了同步开销。

表 1 仿真参数配置表

Table 1 Simulation parameter configuration table

	节点 0	节点 1	节点 2
开机时间/s	0.1	2	3
仿真时间/min	10	10	10
时钟变化率初值	5E-006	2E-006	0
时钟漂移	5E-009	2E-009	0
数据传输速率/bps	10E006	10E006	10E006
阈值 θ /us	1	1	1
TDMA 帧周期/s	0.1304	0.1304	0.1304

图 10 展示了 3 种协议中节点 1 和节点 2 在初始同步时与它们同步节点的时间差。其中,节点 1 的同步节点是节点 0,节点 2 的同步节点是节点 1。MFSS 协议和 TISS 协议中的节点 1 在仿真时间为 2s 左右时与同步节点 0 的时间差基本为 0,说明此时节点 1 已经完成初始同步。而 STS 协议中,节点 1 向节点 0 初始同步的时间是在仿真 3s 左右。同样,节点 2 在 STS 协议中向节点 1 初始同步的时间也要比 MFSS 协议和 TISS 协议晚,这是因为 MFSS 协议和 TISS 协议中新开机的节点可以通过双向交互机制快速完成初始同步,而 STS 协议需要收集若干组数据后才通过线性拟合完成同步,因此 MFSS 协议在同步收敛速度上比 STS 协议更快。

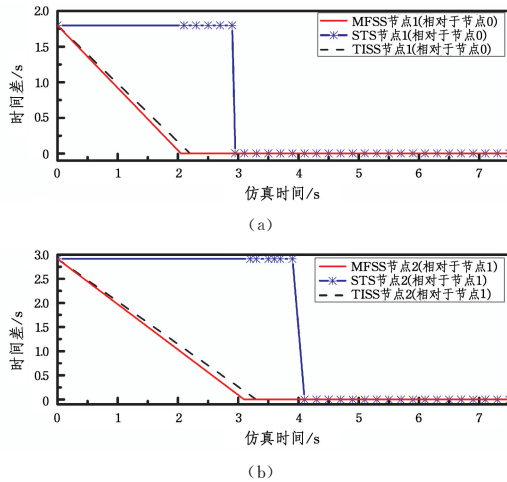


图 10 3 种协议中节点 1 和节点 2 的初始同步
Fig. 10 Initial synchronization of node 1 and node 2 among three protocols

图 10 只能大致看出 3 种协议初始同步后,同步误差都降到 0s 附近,因为节点从开机到初始同步的时间差变化幅度远比之后的时间差变化幅度大。为了进一步分析协议的同步精度,将纵坐标的时间差取值设定在 0.01 ms 级别,以观察 3 种协议中节点 1 和节点 2 的同步精度,如图 11 和图 12 所示。可以看到,在 STS 协议中,节点 1 和节点 2 与它们的同步节点的时间差都保持在 11 us 左右。在 MFSS 协议中,节点 1 相

对于节点 0 的时间差虽然在初始同步后又继续增加,但到达 1 us 左右时重新减小,并且之后的时间差也都是达到 1 us 附近后重新减小,这是因为 MFSS 协议的监测过程保证了节点之间的时间差超过阈值后可以重新进行同步;同样地,节点 2 相对于节点 1 的时间差也是超过 1 us 后又重新减小。因此, MFSS 协议的同步精度显然比 STS 协议更高。STS 协议的同步精度不高,一方面是因为其没有消除传播时延误差,另一方面是因为其重新同步周期过长。而 MFSS 协议则可以根据最大允许的时间误差自适应地调节重新同步周期。在 TISS 协议中,节点 1 和节点 2 相对于它们的同步节点的时间差在大部分时间内都很小,但可以看出每过一个 TDMA 周期,其时间差又会突然增大到 10 us 左右,这是因为其重新同步时又从粗同步重新开始;而 MFSS 协议在重新同步时,并没有因为粗同步而影响同步精度,保证了时序的稳定性。

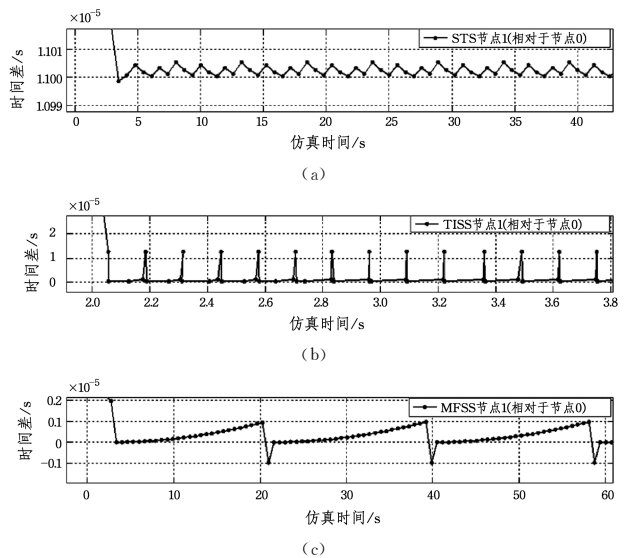


图 11 3 种协议中节点 1 相对于节点 0 的时间差
Fig. 11 Time difference of node 1 and node 0 among the three protocols

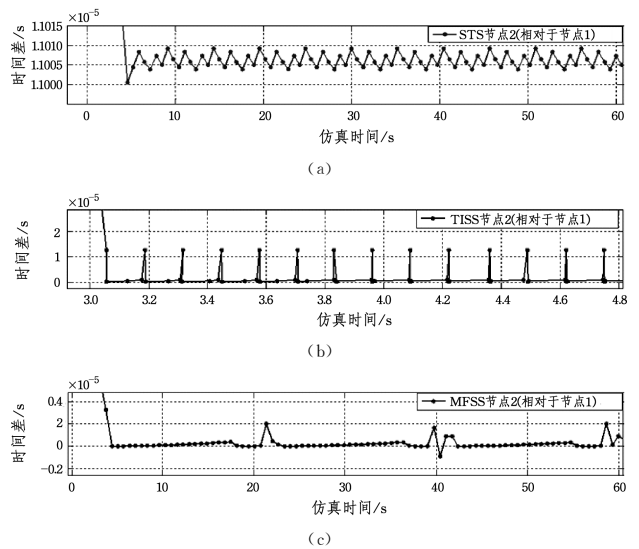


图 12 3 种协议中节点 2 相对于节点 1 的时间差
Fig. 12 Time difference of node 2 and node 0 among the three protocols