

多时钟同步方案的比较与测试

战裕隆, 杨佳利, 朱纪洪

(清华大学 计算机科学与技术系, 北京 100084)

摘要: 时间同步是时间触发通信的关键技术, 但时间同步方案有多种, 难以选择; 文章提出了新的问题——寻找特定网络环境下最适用的同步方案, 论文首先回顾了时间同步技术发展的相关工作, 介绍了星型主从同步, 环型主从同步和 RBS 同步 3 种方案的原理与基于 STM32F407 芯片的系统基本原理和实现要点, 提出了修正时钟漂移速率的创新方法, 并对方案进行理论分析, 归纳各方案的优势、缺陷与适用场合; 最后在少节点的多时间同步系统中完成各方案测试, 证明 3 种方案同步精度都可以达到微秒级, 比较各同步系统的同步精度, 最终结果可以验证先前理论分析的准确性。

关键词: 时间同步; 星型主从同步; 环型主从同步; RBS 同步

Comparison and Testing of Multi-clock Synchronization Methods

Zhan Yulong, Yang Jiali, Zhu Jihong

(Tsinghua University, Beijing 100084, China)

Abstract: Time synchronization is the key technology of time-triggered communication, but there are many different kinds, which are difficult to choose from. The paper put forward a new question——finding the most suitable synchronization method in a specific network environment. It reviewed the work related to the development of clock synchronization; introduced principles and keys to practical system based on STM32F407 chip of the three methods, star master-slave synchronization, ring master-slave synchronization and RBS synchronization; put forward an innovative method for correcting clock drift speed. It also carried on theoretical analysis, summarized the advantages, shortcomings and applicability. Finally, the paper completed the test in multi clock synchronization with a small amount of nodes. The results show that the three methods can achieve microsecond synchronization accuracy. The results can verify the accuracy of the previous theoretical analysis after comparing the synchronization accuracy of the three synchronization systems.

Keywords: time synchronization; star master-slave synchronization; ring master-slave synchronization; RBS synchronization

0 引言

传统的网络通信, 网络通常采用“事件触发”机制。在事件触发网络中, 节点可能在任意时刻发送或接收数据, 一旦网络负载较大而发生冲突, 会产生丢包等问题, 效率低且可靠性差。另外, 多个节点在同一个时间段内进行数据的收发, 通信延迟高, 无法保证消息传递的确定性。

采用时间触发机制的网络通信是指在整个网络当中建立一个统一的时间基准, 节点根据预先设定的通信任务调度表, 按照时间先后次序进行数据收发。在时间触发机制网络中, 网络的利用率可以达到 80%~90%, 比事件触发网络的利用率高, 确定性好。网络通信按照调度表进行, 收发数据的时间可以预估, 而且建立全局系统状态有利于系统的维护操作。时间触发机制在网络发展的新形势下, 在航空航天等对于网络传输确定性有要求的领域中将会有更加广泛的应用。

时间触发网络通信的关键技术主要涉及全局的时间同步与调度表的设计, 其中全局的时间同步尤为关键, 不同的同步方法会有不同的耗时及通信开销和同步精度, 对时间触发网络的性能效果有很大的影响。目前存在多种时间同步方法, 能否适用于时间触发通信的实际系统尚待研究, 在多种方法中选择更

为合适的方案是时间触发网络研究的重要课题。

1 相关工作

近年来, 时间触发网络 TTP^[1] 与 TTE^[2] 凭借着良好的通信实时性与应用广泛的无线传感网络^[3] 一同成为了通信领域的热门方向。其中关键技术——时间同步也就成为了研究的热点。时间同步方案分类有多种方式^[4], 有内部同步与外部同步, 概率同步^[5] 与确定同步, 立即修正同步与维护时间同步, 而下文介绍的最常用的分类方法是主从同步与互同步, 不同方法有不同的应用背景。

在同步问题被提出时, 时钟模型一般被抽象为 $C(t) = ax + b$ (x 为单位时间, a 为时钟漂移速率, b 为时钟偏置), 时间同步就是通过校正使多个不同的时钟模型最终具有相同的时钟速率和偏置, 实现时间同步。1985 年 NTP^[6] 被提出, 是最早且当时应用较广的时间同步方法之一。随后, 时间同步一方面延续 NTP 的同步思路发展: 有 NTP 拓展的 TPSN^[7] 算法, 有现今应用较为广泛的 IEEE1588 协议, 以及 TPSN 的优化 LTS 低深度生成树算法^[8]。除了节点成对同步算法之外, 也有单向通信的同步算法, 相比于成对同步, 它牺牲了一部分时间精度, 但节约了能耗和开销, 同时也能避免通信交互带来的不确定性, 代表算法有 FTSP^[8] 与 DMTS^[9], 主要思想为发送方在消息发送时刻获取本地时间打上时间戳, 消息的接收方获得时间戳, 并记录接收消息的本地时间, 并以此为依据同步。主从同步是主节点发送带有自身信息的分组, 其他节点参照分组信息对自身时钟进行校正。因此, 由于对主控节点的选取和维护方法不同, 除了上述方法外, 也有如全节点同步的新

收稿日期: 2018-01-10; 修回日期: 2018-01-27。

基金项目: 国家自然科学基金(61701276); 国家自然科学基金(61603210); 国家自然科学基金(61673240)。

作者简介: 战裕隆(1994-), 男, 山东牟平人, 硕士研究生, 主要从事时间触发网络研究工作方向的研究。

主从同步方法被提出。

另一方面, 由于主从同步方法在大规模网络与动态拓扑网络中的表现不佳, 理论上更适用于大型网络的互同步方法进入了时间同步的研究范围, RBS^[10]是最早提出的应用最广的互同步方法之一, 而后也衍生出 PBS^[11]等同步方案。此外, 互同步方案不同于主从同步, 它的核心思想是每个节点都从多个其他节点收集时钟的参考值, 考虑节点距离与校正次数等不同的影响因素, 在一个时刻通过多个节点时间值的不同加权方法得到一个估算值。在互同步中, 每个节点重复这种获取估算值的操作, 让所有节点的时钟值不断收敛, 最终趋向于统一化。此类收敛函数式的互同步方案代表性的方法有 2006 年提出的同步扩散^[12]与非同步扩散方案^[14], 其中使用了节点交换信息的随机矩阵, 开拓了数学与时间同步的结合研究领域; 也有平均时间同步 ATS 算法^[15]和分布式时间同步 DTS 算法, 其核心思想是把同步分为斜率同步和偏差同步两个阶段^[14], 而 2012 年提出的 CCS 算法结合了两个同步的阶段, 并在估计加权上提出了置信参数的方法, 取得了比原有加权方法更好的效果。但以上收敛函数式互同步方案较为复杂, 仅在数学理论与仿真中检验了正确性, 在实际系统中的效果有待于进一步研究。

总体看来, 随着网络通信的发展, 时间触发网络得到了更广泛的应用, 作为时间触发网络的关键——时间同步技术产生了诸多方法上的创新, 每种方法都有各自特点。随着时间网络系统种类的多样化, 探究不同方案适合的网络情况必将是值得关注的研究领域。

2 方案介绍

结合上述的介绍, 主从同步方案中, IEEE1588 协议优化方法多, 同步精度高, 适用条件宽泛, 实际应用较多; 环型主从同步(全节点同步)通信开销较小, 将一段通信时间平均到每个相似节点的算法思路具有启发意义; 而互同步中 RBS 同步方案较为成熟, 具有代表性且在实际中检验了可行性与正确性。因此, 本文选择了星型主从同步(基于 IEEE1588 协议), 环型主从同步与 RBS 同步作为主从同步与互同步的代表方案进行理论分析并结合试验进行测试。

2.1 星型主从同步方案

2.1.1 同步原理

星型主从同步的主要结合了 IEEE1588 协议进行工作。其同步原理为成对同步, 对于一对节点, 确定一个为主时钟, 另一个为从时钟。两个 CPU 通过两次互相收发带有时间戳的报文计算两个 CPU 之间的通信延迟和时钟偏差来完成时间同步, 具体执行过程如图 1 所示。

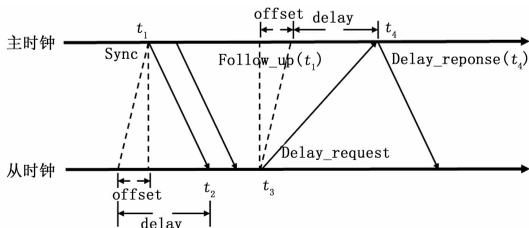


图 1 IEEE1588 协议流程

1) 主时钟发送 Sync 报文, 同时记录下发送报文的本地时间 t_1 ;

- 2) 主时钟再发送 Follow_up 报文, 带有时间戳 t_1 ;
- 3) 从时钟接收报文, 收到 Sync 报文时记录本地时间 t_2 , 从 Follow_up 报文中解析时间 t_1 ;
- 4) 从时钟发送 Delay_request 报文, 同时记录发送报文的本地时间 t_3 ;
- 5) 主时钟接收到 Delay_request 报文, 记录本地时间 t_4 , 发送带有 t_4 时间戳的 Delay_response 报文;
- 6) 从时钟解析 t_4 后通过 t_1, t_2, t_3, t_4 四个时间进行计算与时钟校正^[17], 由图 1 可知:

$$t_2 - t_1 = t_{\text{delay}} - t_{\text{offset}}$$

$$t_4 - t_3 = t_{\text{offset}} + t_{\text{delay}}$$

$$t_{\text{master}} = t_{\text{slave}} + t_{\text{offset}}$$

由此, 可以计算出时间偏置。

2.1.2 系统实现

星型主从同步系统框架如图 2 所示。

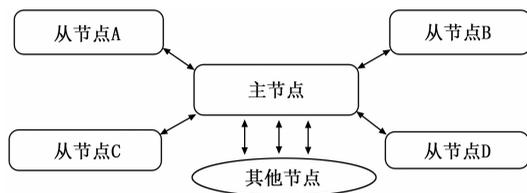


图 2 星型同步系统框架

本文的时间同步的通信中的每个报文长度都是 16 字节, 具体划分如图 3 所示。



图 3 报文结构

2.1.3 时钟漂移速率校正

星型系统中, 在完成系统一个周期的整体同步之后, 可以认为每一个节点都与主节点时间一致。而传统的 IEEE1588 协议只能保证在同步的一个时刻, 两个时钟的值相等, 然而时钟的漂移速率仍有差异, 同步间隔中, 时钟之间的偏差会持续增大, 无法达到时间触发系统的要求。在已有的研究中, 有论文提出了寄存器存储频率比系数辅助速率校正^[18], 然而经过实践检验, 这种方法在 STM32F407 芯片上效果不佳, 究其原因有二: 首先, STM32F407 芯片不支持 double 类型, 频率比系数必须要很高的精度才能取得校正小误差的效果, 因此从系统实现角度来看, 无法使用高精度小数修正时钟速率。其次, 在本实验系统中使用 float 类型进行校正, 然而 float 类型有效位数少, 精度太低, 每次浮点数向整数转换也存在舍去的精度, 修正可在一定程度上提高同步精度, 但是最终效果无法令人满意。

为了解决 STM32F407 中的时钟漂移速率修正的问题, 经过思考与实践, 本文最终在系统中采用了一种人工计算和设置结合而不使用小数的修正从时钟速率的方法如图 4 所示。

- 1) 确定一个时钟周期, 在每个周期同一位置做标记;
- 2) 相邻两个周期的时钟 counter 作差, 分别记录不同 CPU 在同一时间段内的计数;
- 3) 计算主从时钟计数的偏差值, 用主时钟计数值/偏差值作为变量 a (偏差相对时钟计数较小, 则 a 会取成比较大的整

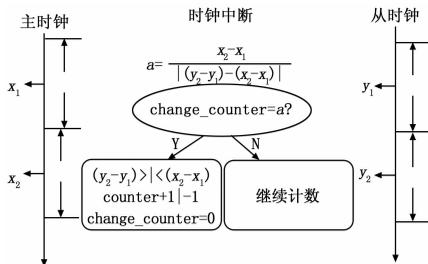


图 4 时钟速率修正算法示意图

数)；

4) 在时钟中断处再设立一个计数器 `change_counter`，它在和 `counter` 执行同样操作基础上，每当 `change_counter` 计数达到 a 时，清零 `change_counter`，如果从时钟相对于主时钟慢，给从时钟 `counter` 加 1；相对主时钟快，则给从时钟 `counter` 减 1；

图 5 为本文时钟修正速率方法的示意图，把偏差分散均匀校正。

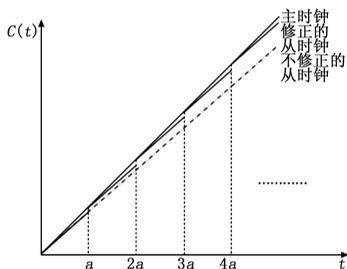


图 5 时钟速率修正效果图

通过加入上述的时钟漂移速率校正方法，星型主从同步可以取得更高的精度，实验对比结果详见后文。本文后续介绍的方案都需要通信节点参数相同，因而都需要使用上述的校正时钟速率的方法。

2.2 环型主从同步方案

2.2.1 同步原理

环型主从同步也被称为基于全节点的同步，需要同步的节点构成环型拓扑，在一轮同步过程中，主节点发送的报文环绕节点一周，每个从节点记录接收报文的本地时间及在环型中的顺序，同时记录主节点发送报文的起始和结束时间，随后将这一时间段平均到每一个从节点上。在环型当中的节点处理数据的能力与速度基本相同，且每一条通信线路上报文传播时间基本相同的情况下，可以通过从节点先前的本地时间和在环型中的位置估算主时钟和从时钟之间的偏差，从而进行校正。

2.2.2 系统实现

环型主从同步系统实现是建立在一个将所有节点排成环型的拓扑上(保证数据包至少经过每个节点一次)，选定一个节点作为主节点，按照如图 6 所示的算法进行：

- 1) 通过本文上述的同步时钟速率方案统一所有通信节点的时钟速率；
- 2) 主节点发送一个报文，同时记录发送报文时刻的本地时间 t_{start} ，报文当中包含报文在接收之前经过了多少个节点(假设环型上共有 n 个节点)；
- 3) 每个从节点在接收到报文时，记录自己在环型当中的相对顺序 k ，以及接收报文时刻的本地时间 t_k ，对报文进行处

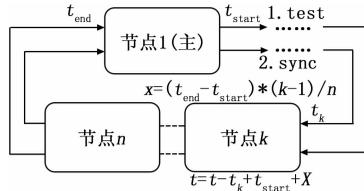


图 6 环型主从同步算法示意图

理后转发；

4) 当报文传输过每一个节点至少一次之后，记录报文返回到主节点的本地时间 t_{end} ；

5) 主节点再次发送报文，报文当中包含 t_{start} 与 t_{end} 的时间戳信息，其中时间戳要打在数据帧的相同位置，保证各节点的通信延迟基本相等；

6) 对于每一个从节点 k ，从主节点发送的报文当中提取起始时间和结束时间的信息，计算 $(t_{end} - t_{start}) * (k - 1) / n$ ，取与其数值上更为接近的整数，设为 x ；

7) 对于从节点 k ，当前其本地时间为 t ，则将其时钟调整为 $(t - t_k + t_{start} + x)$ 。

上述的式子可以拆开分析，在报文到达 k 节点时， k 节点时间为 t_k ，而 $(t_{start} + x)$ 可以认为是起始时间加报文到 k 节点的传输时间，即主节点此时的对应时间，相减得到从节点与主节点的时钟偏差，最后通过偏差值对从时钟进行校正。

2.3 RBS 同步方案

RBS 同步是一种常见的互同步方案，其同步原理是通过参考节点发送广播报文实现接收节点之间的同步如图 7^[19]。

发送的报本身不需要带有时间戳，具体发送时间没有硬性要求，只需要保证连接正常，信道畅通即可。RBS 同步通过一次报文的广播使得多个接收数据节点进行时间同步，在足够多次的广播与同步后，节点的时钟达到一致。

基于 STM32F407 芯片实现的 RBS 方案具体实施如下：

- 1) 同步所有通信节点的时钟速率；
- 2) 选定发送节点，同时向 a, b 两个节点(也可多个)广播不带有时间戳的报文；
- 3) 两个接收节点记录接收报文时刻本地时间，通过报文返回发送节点 (a, b 之间有通信连接也可以直接通信完成校正)；
- 4) 发送节点处理 a, b 节点反馈的报文，计算时钟偏差值；
- 5) 接收节点根据时钟偏差信息完成校正。

3 理论分析

以下本文会结合上述的原理介绍对 3 种方法进行理论分析，主要对比 3 个方案优势，缺陷以及其适用场景。

3.1 优势与缺陷

星型主从同步方案和环型主从同步方案过程简单，从节点校正目标明确，同步效果可控，收敛速度快，由于算法是确定性的，在编写难度上较小。其中星型主从同步方案基于 IEEE1588 协议实现，应用广泛，适用于分布式系统，同步精度高^[17]。而环型主从同步，在每轮同步过程当中，每个节点仅需要收发报文两次，相比于其他算法，通信开销低，节约能耗。

两种方案有主从同步固有的优势，也有主从同步无法回避的缺陷：两个方案都极度依赖于主节点，当主节点发生变化或离开网络时，时间同步会受到很大的影响。此外，两个方案在

大型网络以及拓扑结构变化较大的网络上不适用, 星型主从同步的主节点本身通信负担较重, 容易产生拥塞, 一旦节点过多则会对系统性能产生很大影响; 而环型主从同步对连接的形式要求较高, 在多节点网络和动态网络较难实现环型的通信传输, 同步周期较长, 此外, 一旦拓扑上的一个节点出现故障, 可能会引起整个系统无法正常运行。

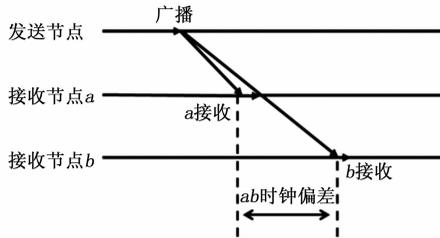


图 7 RBS 同步原理

除了机制上的缺陷, 星型主从同步基于 IEEE1588 协议, 要求节点成对通信时往返传输延迟是相同的错误! 未找到引用源。然而实际的网络未必能完全满足这个假设前提, 往往会产生一定的偏差, 对同步的效果造成影响。同理, 环型主从同步建立在“节点处理数据能力相同, 所有线路传输延迟相同”的假设条件下, 这个假设条件在实际系统中很难成立, 因此实际精度受到影响。

RBS 同步通过广播消除了发送方延迟对同步精度的影响, 算法的运行不依赖于某一个节点, 满足连接要求的节点都可以实现时间同步。其同步算法可以分别在某一个局部运行, 通信开销较为均匀, 不易产生因为同步通信导致的拥塞问题, 因此, RBS 同步可用于大型网络以及动态拓扑结构网络。

缺陷方面, RBS 同步相比于上述的方案编写算法更加复杂, 系统中的节点达到时间同步需要一定的收敛时间, 无法像主从同步一样让通信的几个节点立刻同步。在节点数较少的情况下, 同步精度与收敛速度会比适用的主从同步方案差。

3.2 适用场景

结合上述的优势与缺陷的分析, 可以总结各方案特点, 确定方案的适用场景如表 1 所示。

表 1 同步方案特点总结

	优势	缺陷	适用场景
星型主从同步	收敛速度快同步精度高	对主节点依赖高通信开销大, 要求通信往返延迟相同	节点数目少, 同步精度要求高
环型主从同步	收敛速度快通信开销均匀能耗小	对拓扑有固定要求需要节点和线路基本一致	节点数目适中, 网络拓扑相似度高
RBS 同步	不依赖于某一个节点可局部运行	收敛速度慢发送节点未被同步	大型网络和动态拓扑网络

4 时间同步测试

在进行了上述的理论分析后, 使用 3 个 STM32F407 芯片搭建 CPU, 使用不同的拓扑形式, 实现上述的 3 种时间同步方式的性能测试与对比。

4.1 时间同步测试方案

每一个方案的测试过程首先是分别给 CPU 进行上电, 可

以保证每个 CPU 在初始运行时刻时钟各不相同。在 CPU 通信时, 保证通信方式与参数均相同。

检验时间同步的效果主要使用示波器进行正确性检测在时钟中断处根据时钟值触发 GPIO 引脚输出方波, 使用示波器的多个通道对比方波的跳变边沿, 差值在微秒级。

4.2 测试结果与分析

使用相同的 STM32F407 芯片, 分别运行不同的时间同步方法, 时钟稳定运行后, 每次同步间隔 5 s, 记录 100 次以上的同步结果, 求取平均值作为衡量同步方案的指标, 时钟速率校正方法测试结果如表 2 所示。

表 2 速率校正测试结果

	不校正	原校正方法	本文创新方法
平均时钟偏差/ μs	253.4 919	161.3 388	5.6 204

3 种使用本文提出的校正时钟速率方法后的方案测试结果如表 3 所示。

表 3 多方案同步测试结果

	星型主从同步	环型主从同步	RBS 同步
平均时钟偏差/ μs	5.6204	9.1298	9.3776

从上述的实验结果当中可以看出, 3 种时间同步方案的精度都可以达到微秒级, 可以满足一般的时间触发总线的通信要求。比较分析, 在需要参与时间同步节点数目较少的情况下, 基于 IEEE1588 协议的星型主从同步方法, 通信负担较重, 但在同步精度上明显好于其他两种方法。环型主从同步与 RBS 同步在同步精度和网络负担上较为相近, 在节点较少的情况下不适用, 环型主从同步更适用于网络情况节点数较少, 连接方式能形成环型, 各节点与线路在处理数据和通信传输能力相似, 在系统采用的节点和线路有所差别时, 会产生一定的偏差。而 RBS 同步更适合动态拓扑网络, 对于网络连接方式要求不高, 但在所有节点的同步速度会较慢, 在节点数目较少的情况下, 精度会比适用的主从同步差。测试结果检验了上文理论分析的正确性。

5 结语

本文首先对时间同步的相关工作进行了介绍, 从中选取了 3 种典型的方案——星型主从同步, 环型主从同步与 RBS 同步进行了算法介绍和系统实现。随后, 本文从 3 个方案的特点进行了理论分析, 明确了 3 种方案的优势, 缺陷以及适用的场景。最后, 本文结合基于 STM32F407 芯片搭建的时间同步系统, 通过示波器检验验证了系统的正确性; 试验结果表明, 3 种同步方案精度可以达到微秒级别; 比较了在需要同步节点较少情况下, 3 种方案取得同步效果的优劣, 进一步验证了之前所做分析——“星型主从同步更适用于节点数目少的时间触发网络”。

参考文献:

[1] Kopetz H, Grunsteidl G. TTP — A time-triggered protocol for fault-tolerant real-time systems [J]. 1993: 524-533.
 [2] Kopetz, Hermann, Ademaj, et al. The Time-Triggered Ethernet (TTE) Design [J]. 2005: 22-33.