

引用格式: 修相宇, 王康, 武建锋, 等. 基于 MAC 的 PTP 同步优化设计与实现[J]. 时间频率学报, 2025, 48(2): 101-108.

基于 MAC 的 PTP 同步优化设计与实现

修相宇^{1,2}, 王康^{1,3}, 武建锋^{1,3}

1. 中国科学院 国家授时中心, 西安 710600;
2. 中国科学院大学 集成电路学院, 北京 101408;
3. 时间基准及应用重点实验室(中国科学院), 西安 710600

摘要: 针对 PTP (precise time protocol) 协议在应用层获取软件时间戳导致时钟同步精度下降的问题, 提出一种基于 MAC (media access control) 层获取硬件时间戳的 PTP 同步优化方案。设计了以 STM32F407 微处理器为核心的 PTP 时钟应用平台, 在 MAC 层实现了硬件时间戳获取, 避免了由于协议栈软件处理延时产生的不确定性; 针对 PTP 时钟晶振老化导致的时间同步偏差及网络延迟抖动问题, 采用迭代方法优化了本地时钟频率调节算法, 提高了频率校正精度。经实际测试, 主从时钟偏差的 RMS (root mean square) 优于 20 ns, 提升了时钟同步精度。

关键词: PTP 协议; MAC 层; 滤波; 时钟同步; 时间戳
DOI: 10.13875/j.issn.1674-0637.2025-02-0101-08

Design and implementation of PTP synchronization optimization based on MAC

XIU Xiang-yu^{1,2}, WANG Kang^{1,3}, WU Jian-feng^{1,3}

1. National Time Service Center, Chinese Academy of Sciences, Xi'an 710600, China;
2. School of Integrated Circuits, University of Chinese Academy of Sciences, Beijing 101408, China;
3. Key Laboratory of Time Reference and Applications, Chinese Academy of Sciences, Xi'an 710600, China

Abstract: To address the issue of reduced clock synchronization accuracy caused by obtaining software timestamps at the application layer in PTP (precise time protocol) protocol, this paper proposes a PTP synchronization optimization scheme based on hardware timestamp acquisition at the MAC (media access control) layer. A PTP clock application platform was designed with STM32F407 microprocessor as the core, implementing hardware timestamp acquisition at the MAC layer to eliminate uncertainties introduced by software protocol stack processing delays. For the problems of time synchronization deviation caused by PTP clock oscillator aging and network delay jitter, an iterative method was adopted to optimize the local clock frequency adjustment algorithm, improving frequency correction accuracy. Actual test results demonstrate that the root mean square (RMS) deviation of the master-slave clock achieves better than 20 nanoseconds,

significantly enhancing clock synchronization precision.

Key words: PTP (precise time protocol) protocol; MAC (media access control) layer; filtering; clock synchronization; time stamp

网络时间同步技术作为分布式测试系统的一项重要时间同步技术,广泛应用于导航与定位、测量与控制等各个领域。传统网络时间协议(network time protocol, NTP)的网络时间同步精度为毫秒级,满足不了特定网络的高精度时间同步需求,电气电子工程师学会(Institute of Electrical and Electronics Engineers, IEEE)制定了 IEEE1588 标准来优化提升网络同步精度。

IEEE1588 标准,全称为网络测量与控制系统的精密时间同步协议(Precision Clock Synchronization Protocol for Networked Measurement and Control Systems),简称为精密时间协议(precise time protocol, PTP)^[1]。该协议支持 3 种时间戳实现方式:基于支持 PTP 协议的物理层(physical layer, PHY)芯片在物理层获取硬件时间戳标记方式;基于 PTP 协议的网络层软件时间戳标记方式;基于支持 PTP 协议的微控制器在介质访问控制层(media access control layer, MAC)的介质独立接口(media independent interface, MII)处获取时间戳标记方式^[2]。文献[3]使用第一种时间戳标记方式,基于 DP83640 以太网芯片,能够达到亚微秒级时钟同步精度,但 DP83640 是一款在物理层获取时间戳的芯片,通用性不强,不便于进行物理层芯片硬件替换。文献[4]基于第二种实现方式获取软件时间戳,虽然实现简单,节约成本,但同步精度仅能达到 19 μs 。文献[5]基于第三种实现方式,在简化介质独立接口(reduced medium independent interface, RMII)处获取时间戳,同步精度可达 50 ns,但采用的算法较为复杂,会对有限的操作系统产生一定的延时。

本文设计了一种基于 MAC 获取硬件时间戳的 PTP 时钟应用平台,使用 STM32F407 芯片作为主控芯片并获取硬件时间戳,搭配物理层芯片 LAN8720,实现 PTP 时钟时差校准和频率调节,并对 PTP 时钟应用平台频率补偿值进行有限脉冲响应(finite impulse response, FIR)滤波处理。

1 PTP 协议时钟同步机制

1.1 PTP 时间同步原理

PTP 协议的核心思想是采用主从时钟方式,对时间信息进行编码,利用网络的对称性和延时测量技术,实现主从时钟的频率、相位和绝对时间的同步^[6],本文选择单步方式来进行报文传递^[7],同步原理如图 1 所示。

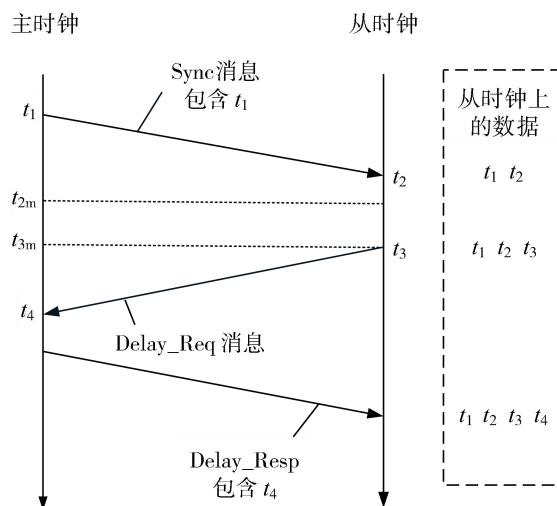


图 1 基于多播通信的 PTP 同步原理

PTP 主时钟周期性地向从时钟组播 Sync 报文, Sync 报文离开主时钟的系统时间戳为 t_1 ,并将 t_1 的值记录到报文当中;从时钟接收 Sync 同步报文同时捕获报文到达时间戳 t_2 ,并从报文中获得时间戳 t_1 ;随后从时钟会向主时钟发送 Delay_Req 报文,并记录发送报文时间戳 t_3 ;主时钟接收到 Delay_Req 报文,记录收到报文时间戳 t_4 ;最后,主时钟将时间信息及 t_4 封装至 Delay_Resp 报文中发送给从时钟,当从时钟接收 Delay_Resp 报文后,一个同步周期结束。从时钟使用 t_1 、 t_2 、 t_3 、 t_4 这 4 个时间戳值来计算时钟偏差与平均链路延迟^[8]。

定义主从时钟之间的时钟偏差为 F_i ,假设一个同步周期内主从时钟频率一致,主时钟到从时钟的链路延迟为 D_j ,从时钟到主时钟的链路延迟为

D_k , 则有:

$$t_2 - t_1 = D_j + F_i, \quad (1)$$

$$t_4 - t_3 = D_k - F_i. \quad (2)$$

假定主从时钟的往返链路延迟是近似相等的, 则平均链路延迟 D_i 为 D_j 和 D_k 二者和的均值, 由此根据式 (1) 和式 (2) 有:

$$F_i = (t_2 - t_1 + t_3 - t_4)/2, \quad (3)$$

$$D_i = (t_2 - t_1 + t_4 - t_3)/2. \quad (4)$$

最后从时钟根据时钟偏差 F_i 和平均链路延迟 D_i 来调整本地时钟并与主时钟保持时间同步。

1.2 时间戳硬件获取方式

时间戳的获取方式是网络设备能够在复杂的网络环境下实现高精度时间同步的关键部分。如图 2 所示, PTP 协议有 3 种时间戳获取方式, 3 种方式的区别在于获取时间戳的位置不同, 软件注入时

间戳易受任务切换、中断处理、网络协议栈等因素的干扰, 导致时间戳抖动较大, 同时在高负载的网络环境中, 软件处理能力可能成为瓶颈, 因此, 软件实现方式精度相对较差, 但实现方式相对简单^[9]; 物理层注入时间戳则由硬件直接触发, 避免了软件层所带来的随机抖动, 在此记录时间戳能够捕获最真实、最接近网络物理事件的时间, 精度最优, 可达纳秒级别, 但是需要特定的硬件支持, 像 DP83640 芯片, 内置 125 MHz 的时钟, 可作为本地时钟使用; 在 MAC 层注入时间戳介于二者之间, MAC 层硬件时间戳记录的時刻接近实际到达或发送物理时间, 可以通过滤波算法降低抖动, 进而使同步精度达到在物理层获取时间戳的精度, 保持精度在亚微秒级, 可通过直接内存访问 (direct memory access, DMA) 来传输时间戳, 不需要特殊的 PHY 芯片, 可替换性更好^[10]。因此本设计采用硬件方式在 MAC 层获取时间戳。

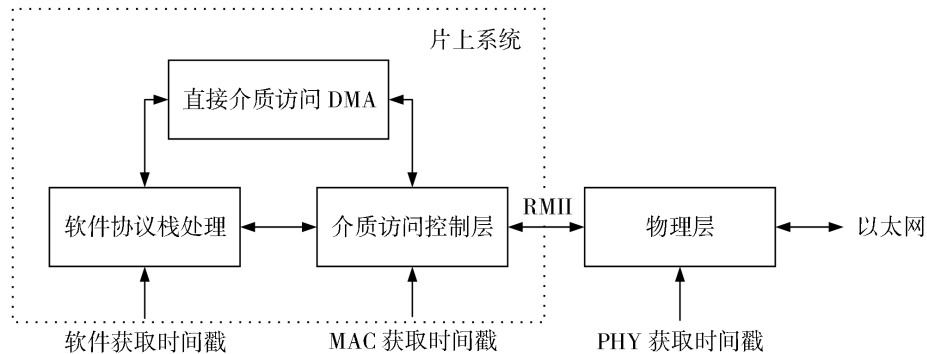


图 2 时间戳获取方式

2 PTP 时钟应用平台软硬件设计

2.1 PTP 时钟应用平台硬件结构设计

PTP 时钟应用平台选择由意法半导体公司研制的 STM32F407VGT6 微控制器, 如图 3 所示。

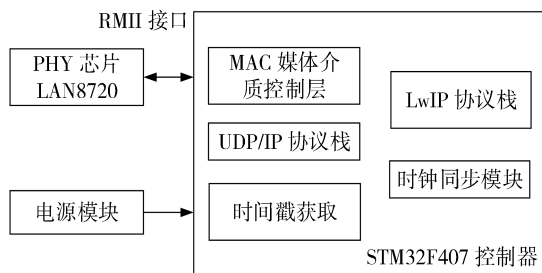


图 3 PTP 时钟应用平台硬件系统架构

该控制器支持 PTP 时间戳功能, 可输出秒脉冲, 支持在 MAC 层 RMII 接口处获得时间戳。PHY 芯片选用 LAN8720, 负责将同步报文信号转换为控制器可识别的物理信号, 通过 RJ45 端口进行网络报文交互。微控制器通过 RMII 接口与以太网 PHY 芯片连接, 外接 5 V 电源为整个平台供电。

2.2 MAC 媒体介质控制层

STM32F407 为时钟平台的主控芯片, 其内部集成了一个以太网外设, 无需外接 MAC 层控制芯片, 通过 DMA 控制器进行介质访问控制来实现 MAC 媒体介质控制层的任务, 支持工业标准接口

MII/RMII 用于与外部 PHY 芯片连接,其通过高级高速总线 (advanced high-performance bus, AHB) 主从接口与内核和寄存器相连, AHB 主接口用于控制数据传输,从接口用于访问 DMA 控制状态与工作模式寄存器空间。

在接收报文时,外部 PHY 芯片在收到报文信号后通过 RMII 传给 MAC 层,MAC 层在检测到起始帧定界符 (start frame delimiter, SFD) 时会记录时间戳,存储在相应的描述符中,随后将报文存储在一个 2 KB 大小的先进先出 (first in first out, FIFO)

缓存中,当 FIFO 中的报文大小超过预先设定的数值时,DMA 开始工作,将 FIFO 中的数据拷贝到预先设定的内存地址,相位描述符的拥有权标志位 (ownership flag, OWN) 置 1 表示拷贝完成,此时可用上层协议栈函数对内存中的报文进行处理。

在发送报文时,上层函数通过总线矩阵将报文存储在缓冲区中,随后 DMA 将报文传输到 FIFO 中,待发出后 MAC 将报文的时间戳存储在相应的描述符内存中,最终通过 PHY 芯片发送到以太网中,以太网外设功能框图如图 4 所示。

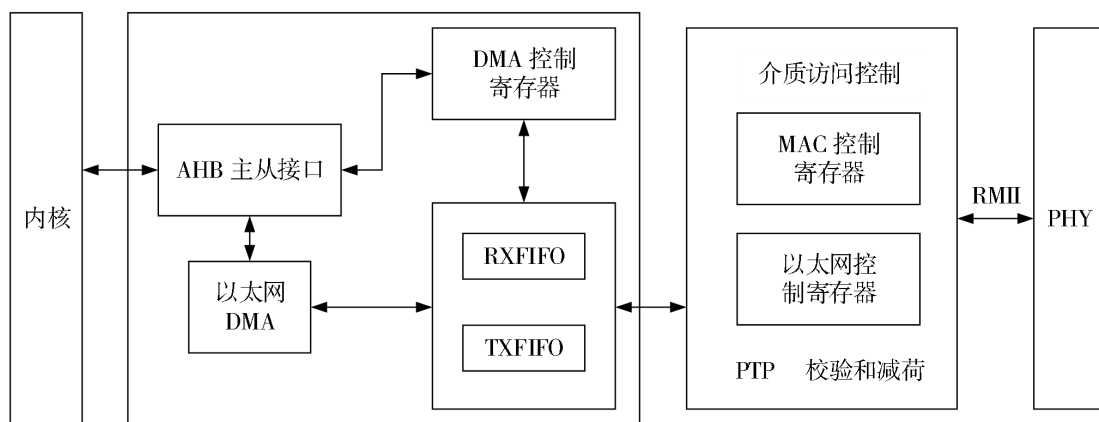


图 4 以太网外设功能框图

2.3 主程序流程图

采用 Keil uVision5 作为软件开发环境,将轻型协议栈 (lightweight IP, LwIP) 移植到无操作系统中来实现用户数据报协议 (user datagram protocol, UDP)。微控制器 STM32F407 运行的主程序流程图如图 5 所示,主要分为 4 个部分,分别是以太网相关功能初始化; PTP 协议相关功能初始化; 报文收发程序; 系统时钟调节。

对于以太网相关功能初始化,包括获取以太网接口初始化、LwIP 协议栈初始化。PTP 相关功能初始化包括时间戳寄存器初始化、PTP 协议栈初始化, PTP 协议栈初始化主要包括从时钟端口状态、链路延迟与时钟偏差初始化以及单步双步模式选择等。报文收发程序则包括加入多播组以及开启同步报文收发定时器。系统时钟调节主要包括本地时钟时差校准、频率校正和滑动平均滤波。

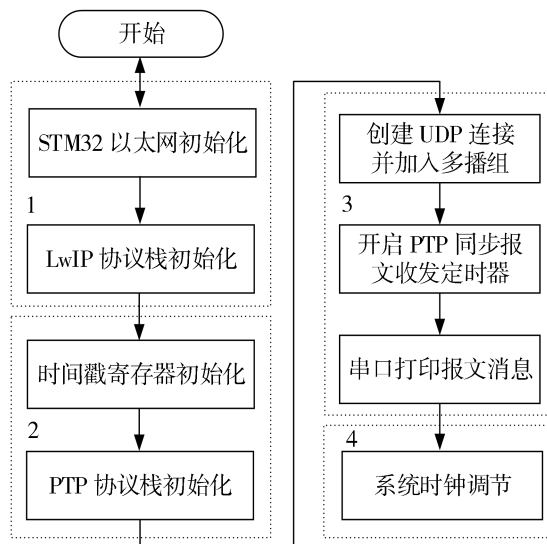


图 5 主程序流程图

3 PTP 时钟调节

3.1 PTP 时钟应用平台时差校准

PTP 时间存储的类型由两个无符号的整型变

量构成的结构体，分别用来表示时间戳的秒和纳秒，以便于在时差校准时分别写入时间戳高位更新寄存器和时间戳更新低位寄存器。时差校准就是计算出主从时钟之间的偏移量 F_i ，可由公式 (3) 得到，随后写入寄存器中，时差校准的周期可根据 Sync 报文的同步周期进行设置。本地时钟调节流程如图 6 所示，时钟时间戳计数器的秒初值和纳秒初值一般情况下均设置为 0，所以当时钟平台第一次接收到 Sync 报文中的时间戳 t_1 后，时钟平台调整值约等于 Sync 报文时间戳的值，在经过 8 次时差校准之后，时钟平台的偏移量与主时钟的偏移量大约在微秒级别，然后对时钟平台进行频率调节。

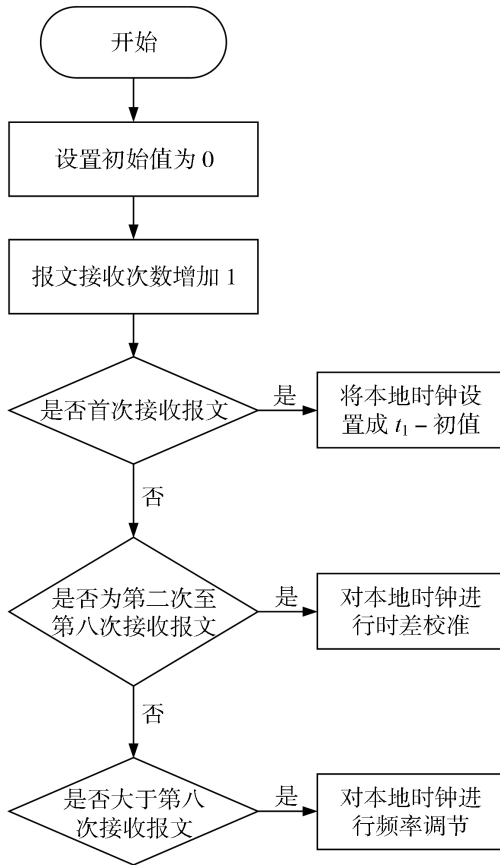


图 6 本地时钟调节流程图

频率调节能够确保从时钟与主时钟频率保持一致，弥补由于硬件时钟的自然漂移导致的误差，提高时钟同步精度，可通过算法计算出从时钟相对于主时钟的频率偏差，该算法如下所示。

在 M_n 时，主时钟向从时钟发送 Sync 报文，从时钟在 S_n 时收到消息，假设链路延迟 D_i 在同步周期 n 和同步周期 $n-1$ 是相同的，则有如下公式计算

主时钟时间 C_n ：

$$C_n = M_n + D_i \quad (5)$$

当前同步周期主时钟计数 I_n 的计算式为

$$I_n = C_n - C_{n-1} \quad (6)$$

当前同步周期从时钟计数 K_n 的计算公式为

$$K_n = S_n - S_{n-1} \quad (7)$$

从时钟的分频系数 Z_n 的计算公式为

$$Z_n = I_n / K_n \quad (8)$$

频率的补偿值 V_n 的计算公式为

$$V_n = Z_n \times V_{n-1} \quad (9)$$

式 (9) 中： V_n 的初始值和寄存器常数值 R 可根据所需要达到的时钟精度 P 和参考时钟的频率 L 计算得到，计算公式为：

$$V_n \times R = 2^{63} / L \quad (10)$$

$$P = R \times 10^9 / 2^{31} \quad (11)$$

由此计算出来的频率补偿值用于更新到时间戳加数寄存器中，但需要等待时间戳加数寄存器中的新值生效才算完成频率校正，因此可以通过在系统时间达到目标值后，通过触发中断来完成，目标值可以在目标时间戳高位寄存器和时间戳低位寄存器中设置时间值。

但在实际计算的过程中，若 I_n 与 K_n 之间的数值相差较小，考虑到嵌入式整数精度的限制，整数除法可能导致 I_n 与 K_n 的值为 1，这种情况下，频率补偿的计算不会发生变化，无法进行有效的补偿，尤其对于 STM32F407 这种嵌入式系统，由于整数运算舍入的影响，补偿精度可能进一步下降，为避免这一问题，本文对原有公式进行变形，采用递归的形式，计算如下。

$$V_n - V_{n-1} = V_{n-1} \times \frac{I_n - K_n}{K_n} \quad (12)$$

令

$$\Delta V = V_n - V_{n-1} \quad (13)$$

定义误差增量:

$$\Delta M = I_n - K_n. \quad (14)$$

则

$$\Delta V = V_{n-1} \times \frac{\Delta M}{K_n}. \quad (15)$$

最后, 频率补偿值的 V_n 值为

$$V_n = V_{n-1} + \Delta V. \quad (16)$$

3.2 频率补偿滤波器

由于协议栈所带来的不确定性, 直接根据公式在每个 Sync 报文同步周期后计算出频率补偿值更新到时间戳加数寄存器后, 会对时钟精度造成影响, 导致主从时钟偏差产生抖动, 因此需要对频率补偿值进行滤波处理。在 PTP 协议中, 选择不同类型的滤波器, 例如 FIR、IIR、卡尔曼滤波等。

无限脉冲响应滤波器存在反馈环路, 卡尔曼滤波器因包含动态更新过程和矩阵运算, 信号的响应时间变得复杂。FIR 滤波器也称为有限冲激响应滤波器, FIR 滤波器不需要复杂的实现步骤, 能够减少对操作系统带来的影响, 同时该滤波器只需要当前数据和历史数据, 不需要历史滤波处理数据, 这

能够在 Sync 报文发送周期相对较短时, 进行快速处理。此外 FIR 滤波器在系数满足一定条件下, 它的相频特性是线性的, 能够保留原始信号的相位信息, 因此本文采用 FIR 滤波器来处理频率补偿值。FIR 滤波器结构图如图 7 所示。

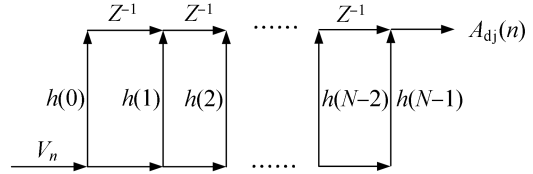
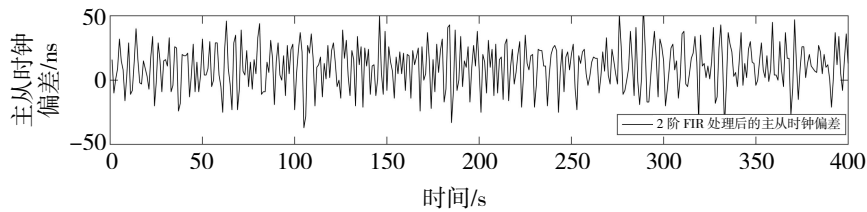


图 7 FIR 滤波器结构图

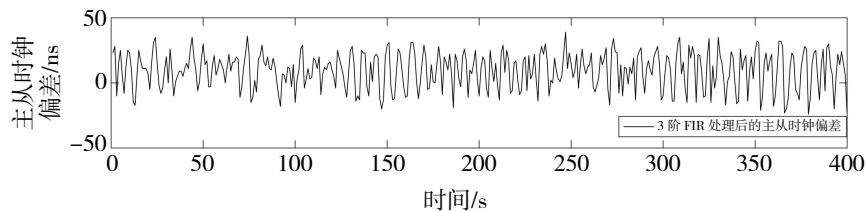
图 7 中, N 阶 FIR 滤波器的输出频率补偿值 A_{dj} 公式如下, 其中 h_n 为系数, N 为阶数, V_{n-k} 为输入数据。

$$A_{dj}(n) = \sum_{k=0}^{N-1} h_n V_{n-k}. \quad (17)$$

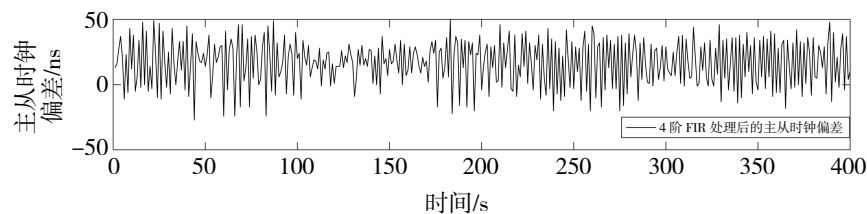
选用不同阶数的 FIR 滤波器来对频率补偿值进行滤波处理, 并比较其性能。如图 8 所示分别为 2、3、4、5 阶 FIR 滤波后的主从时钟偏差值运行曲线对比图, 2 阶 FIR 滤波系数为 1/2, 3 阶 FIR 滤波系数为 1/3, 4 阶 FIR 滤波系数为 1/4, 5 阶 FIR 滤波系数为 1/5。



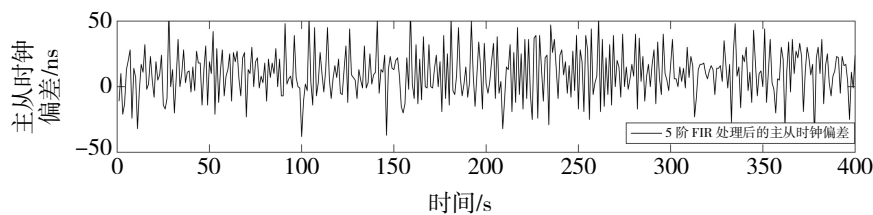
(a) 2 阶 FIR 处理后



(b) 3 阶 FIR 处理后



(c) 4 阶 FIR 处理后



(d) 5 阶 FIR 处理后

图 8 经过不同阶数 FIR 滤波器处理后的主从时钟偏差

由图 8 可以观察到,不同阶数 FIR 滤波器对主从时钟偏差的影响存在差异,经过 2 阶和 3 阶 FIR 滤波处理后,在一段时间内主从时钟偏差抖动幅度变小,4 阶与 5 阶滤波器虽然理论上滤波能力更强,但在嵌入式系统运行过程中,易受中断、定时抖动等微小扰动以及晶振老化抖动的影响,导致滤波器对这些扰动产生过度反应,使得输出偏差曲线更加稠密、波动频繁,增加了信号处理的延时,影响实时性能,反而降低了系统的时间同步稳定性。相比之下,3 阶 FIR 滤波器在抑制抖动的同时,具有良好的鲁棒性与动态响应能力,在面对晶振及网络环境等不可避免的轻微抖动时,能保持输出偏差相对平稳、波动适中。

如表 1 所示为使用不同阶数的 FIR 滤波器处理后的主从时钟偏差,对比发现经过 2 阶和 3 阶 FIR 滤波器处理后使主从时钟偏差的均方根值 (root mean square, RMS) 减小,经过 4 阶 FIR 滤波器处理后相比于前者, RMS 增加,滤波效果有所下降,经过 5 阶 FIR 滤波后的 RMS 值相比于 4 阶变化不大。因此本文选用 3 阶 FIR 滤波器进行滤波处理,均值在 10 ns 以内,标准差值在 15 ns 以内。

表 1 不同阶数的 FIR 滤波器处理后的主从时钟偏差
单位: ns

阶数	均值	标准差	RMS
1 阶	11.46	18.40	21.70
2 阶	10.35	17.71	20.50
3 阶	9.78	14.17	17.20
4 阶	11.62	15.84	19.21
5 阶	10.13	16.03	20.06

为了更直观地比较不同阶数 FIR 滤波器对时钟同步性能的影响,图 9 给出了主从时钟偏差在各阶滤波处理后的均值、标准差及 RMS。

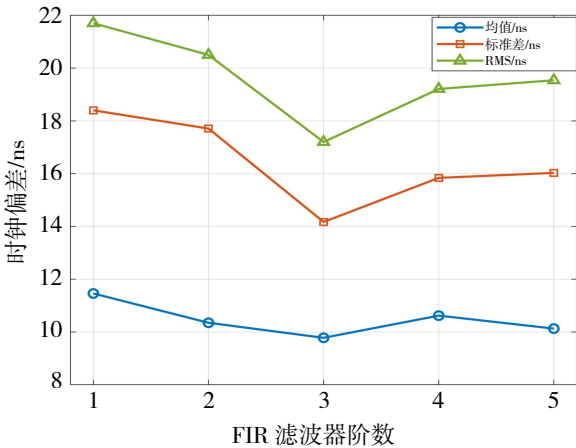


图 9 主从时钟偏差在各阶滤波处理后的均值、标准差及 RMS

4 测试与验证

搭建了一个主时钟两个从时钟的测试环境,用于验证多播报文的完整性,本文设计的 PTP 时钟平台作为从时钟,由中国科学院国家授时中心研制的 N5100 网络精密时间服务器作为主时钟,该服务器支持多播模式和 PTP 协议,能够配置秒脉冲差值、同步报文间隔,在一个组播组中可作为主机时间源。将两个时钟平台通过交换机与主时钟相连接来搭建测试环境,通过 CH340 串口转通用串行总线 (universal serial bus, USB) 模块与电脑端连接,查看报文信息,如图 10 所示。

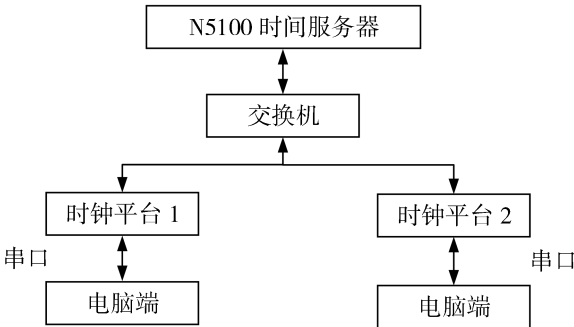


图 10 测试平台搭建

按照表 2 的测试环境和条件进行测试。

表 2 测试环境及条件

测试环境	中间设备	时钟频率源	网络速度 /Mbit/s
25℃室温	交换机	25 MHz 恒温晶振	100

时间服务器同步周期为 1 s，时钟平台每 1 s 会对本地时钟进行一次校准，采用 3 阶 FIR 滤波器对频率补偿值进行滤波处理，本文验证时钟平台 1 与时间服务器的时钟偏差，如图 11 所示为时钟平台 1 在时差校准和频率校正后的运行曲线，主从时钟偏差平均值为 9.81 ns，标准差为 14.20 ns，RMS 为 17.28 ns。

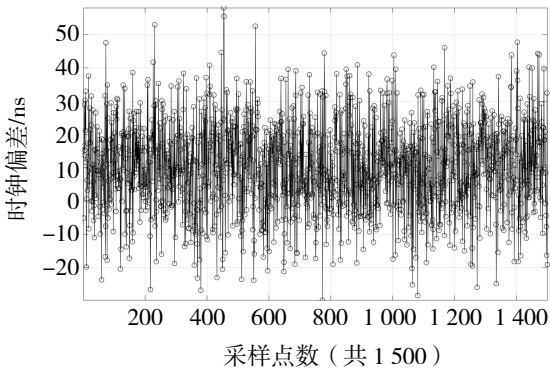


图 11 PTP 时钟平台频率校正后时钟偏差值运行曲线

直方图如图 12 所示。

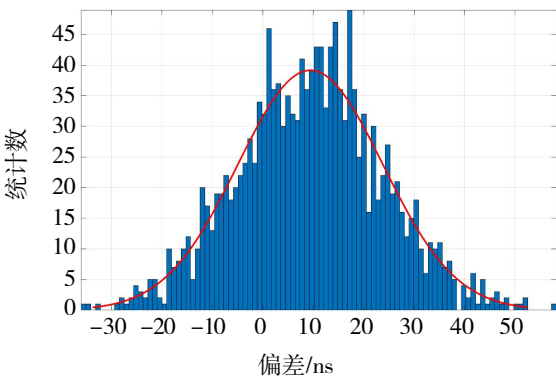


图 12 PTP 时钟平台频率校正后时钟偏差值直方图

5 结论

本文针对在应用层获取软件时间戳导致精度不足的问题，选择在 MAC 层获取硬件时间戳来实现 PTP 从时钟功能，不需要专用物理层芯片，提升了硬件可移植性，同时使用 FIR 滤波处理，提升同步精度。测试结果表明，本文所实现的 PTP 精密时钟同步协议，主从时钟偏差的 RMS 可以控制在 20 ns 之内。

参考文献：

[1] 高兆强. 基于 IEEE1588V2 的无线网络时间同步技术[J]. 通信技术, 2021, 54(1): 147-151.

[2] 徐卓汀, 商艳娟, 王成群. 基于 FPGA 的 IEEE 1588 时钟同步系统的设计与实现[J]. 智能计算机与应用, 2023, 13(5): 64-69.

[3] 孙建鹏, 刘欣, 洪应平, 等. 基于 DP83640 的以太网时钟同步方法设计[J]. 电子学报, 2021, 49(5): 1033-1040.

[4] 陶稳静, 陆阳, 卫星, 等. 基于 PTPd2 的精密时钟同步软件实现方法[J]. 计算机工程, 2019, 45(3): 47-53+59.

[5] 陶征亮. 基于 IEEE1588 协议的高精度时钟同步技术研究[D]. 赣州: 江西理工大学, 2022.

[6] 王晋祺. 基于 IEEE 1588v2 协议的分布式测试系统时统系统研究[D]. 太原: 中北大学, 2019.

[7] 孔祥瑞. 基于 IEEE1588 协议的铁路时间同步技术研究[D]. 秦皇岛: 燕山大学, 2020.

[8] 曾庆琦. 高精度网络时间同步的研究与实现[D]. 武汉: 中国科学院大学(中国科学院武汉物理与数学研究所), 2020.

[9] 黄正宇. 基于 IEEE1588 的无线以太网时间同步实现[D]. 哈尔滨: 哈尔滨工业大学, 2022.

[10] 宋仕坤. 基于以太网 MAC IP 核的 IEEE1588 协议的设计与实现[D]. 西安: 西安电子科技大学, 2018.