

基于PTPd的精准时钟同步技术的研究

苏宇¹, 胡珩^{1,2}, 张涛¹, 王金波¹

(1. 中国科学院空间应用工程与技术中心, 北京 100094;
2. 中国科学院大学, 北京 100049)

摘要: IEEE1588 定义了一种精准时钟同步协议 (PTP), 因其具有高同步精度, 故逐渐取代传统的 NTP (Network Time Protocol) 被广泛应用于更高时钟同步精度的分布式测控系统中。PTP 协议可以通过纯硬件或纯软件方式实现, 纯硬件方式通过硬件编程实现, 同步精度可达纳秒量级, 但是开发难度大; 而纯软件实现的 PTP 利用协议实现完整的开源代码 PTPd 实现, 开发难度较低, 同步精度略低于纯硬件实现的 PTP 协议。文中在深入比较、分析了 IEEE1588 协议与 NTP 的同步机理后, 并对纯软件实现的 PTP (PTPd) 进行同步精度测试。结果表明, PTPd 的同步精度可达 $\pm 10 \mu\text{s}$, 远优于传统的 NTP 同步协议 $\pm 200 \text{ ms}$ 的同步精度, 而被成功应用于载人航天嵌入式软件第三方评测分布式测试系统中。

关键词: 时钟同步; 网络时间协议; IEEE1588; PTPd

中图分类号: TP202.2

文献标识码: A

文章编号: 1673-629X(2016)01-0175-06

doi: 10.3969/j.issn.1673-629X.2016.01.038

Research on Precision Clock Synchronization Technology Based on PTPd

SU Yu¹, HU Heng^{1,2}, ZHANG Tao¹, WANG Jin-bo¹

(1. Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences,
Beijing 100094, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: IEEE1588 defines a Precision Time clock synchronization Protocol (PTP) which is widely used in distributed measurement and control systems where higher synchronization precision are necessary, and PTP gradually takes the replace of the traditional NTP because of its high synchronization accuracy. The PTP can be achieved by pure hardware or pure software. The pure hardware method can achieve the synchronization precision up to nanoseconds through the hardware programming, but the development is difficult, while the pure software method can be achieved by the open source PTPd code, it is simple and able to obtain a lower synchronization accuracy than the pure hardware. In this paper, synchronization mechanism of NTP and IEEE1588 precision clock synchronization protocol was compared and analyzed, the PTP (PTPd) is implemented in pure software method were tested. Result of tests proves that the synchronization precision of PTPd is within $10 \mu\text{s}$, which is much better than the NTP's $\pm 200 \text{ ms}$, so PTPd has been successfully applied in the third-party embedded software testing of distributed testing system for manned spaceflight.

Key words: clock synchronization; NTP; IEEE1588; PTPd

0 引言

近年来,随着计算机技术和载人航天技术的飞速发展,航天嵌入式软件系统的复杂度越来越高,接口越来越丰富,信号种类越来越多,如 1553B 总线接口、RS422 接口、IEEE1394 总线接口等。在航天嵌入式软件测试时,会出现一个包含上述接口信号的航天嵌入

式被测组件同时与多个嵌入式仿真测试前端板(核心板+底板的结构,核心板采用 CPU+FPGA+存储芯片的通用架构,底板为相应对外接口)相连,即组成分布式测试系统。在分布式测试系统中,一部分测试用例的执行常常要各个嵌入式仿真前端协同动作——如某型号航天器嵌入式主控软件要求在 1553B 总线的 XXX 指

收稿日期: 2015-04-15

修回日期: 2015-07-17

网络出版时间: 2016-01-04

基金项目: 国家重大专项(Y2140102RN)

作者简介: 苏宇(1985-),男,硕士,助理工程师,研究方向为高可靠软件;张涛,研究员,博士生导师,研究方向为高可靠软件;王金波,副研究员,硕士生导师,研究方向为高可靠软件测试与验证。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160104.1510.046.html>

令注入后 $200\mu\text{s}\pm 25\text{ ns}$ 内注入 SPI 接口的 XXX 指令,这就要求两个测试前端能够严格按照此时序关系向被测对象注入测试激励数据。因此,需要一个精确的、统一的标准时间,从而同步分布式测试系统中的其他各个嵌入式仿真测试前端。

同步的基准时间可以是上位机系统时间,或者是以其中某一个嵌入式仿真测试前端的时间作为标准时间(即主时钟),进而去同步其他前端的系统时间(从时钟),以达到测试的目的。

现阶段,被广泛使用的网络时间同步方法为网络时间协议(Network Time Protocol, NTP)。但是,此方法在距离不超过 2 000 公里万维网内 NTP 的同步精度为 10 ms,在局域网内的同步精度为 1 ms^[1],在同步精度要求较高的场合并不适用。而人们传统意识上的 GPS,虽然性能可观,但要求有同步卫星的覆盖,而且存在一定的政治安全问题,因此让人望而却步。在这种情况下,IEEE1588 协议应运而生,并迅速因其高精度与高可靠性受到人们的普遍欢迎。

文中的目的是分析 NTP、IEEE1588 协议的同步机理以及 PTPd(IEEE1588 的软件实现)的结构特点,并将 PTPd 应用于载人航天嵌入式软件第三方评测系统中。通过实际应用测试结果说明 PTPd 的高同步精度可以满足测试需要。

1 同步协议介绍

1.1 NTP 协议

1.1.1 NTP 协议概述

NTP 协议是由美国海军水面武器中心(NSWC)、美国国防部高级研究计划局(DARPA)以及美国国家自然科学基金(NSF)资助的网络时间同步项目,其第一个版本于 1988 年成功开发。截止到目前为止,NTP 协议已经发展到了第四个版本,应用到与时间同步的各个领域。为了能够达到令人满意的同步精度,NTP 采用了多服务器算法^[2],该算法十分复杂,这也在某种程度上限制了 NTP 协议的推广。但在大多数网络负载不大的情况下,NTP 协议能够达到毫秒级的同步精度,这一点使其成功应用在很多对同步精度要求不是很高的应用中;当 NTP 用于洲际间时间同步时,同步的精度只有秒级。

1.1.2 NTP 协议的同步机理

NTP 协议一般有对等工作模式、广播/多播工作模式以及客户端/服务器(C/S)模式。在对等工作模式下,客户端和服务端具有相同地位,任何一方都可以同步另一方,也可以被对方同步,一般来说,率先发出申请建立连接的一方工作在主动模式下,而另一方工作在被动模式下;广播/多播工作模式采用了一对多模

式,即作为服务器的一方主动发出携带时间戳的消息,作为客户端可以根据此消息中的时间戳信息来调整自己的时间;在 C/S 模式下,采用了一对一的连接方式,服务器能够同步客户端的时间信息,而客户端不能同步服务器的时间信息。不同的同步协议往往提供不同的时间同步策略,但都遵循两个通用的模型:(1)客户端发送请求到服务器,服务器用当前时间响应;(2)服务器发送时间信息给一组或所有用户^[3]。

一般来说,当今通常是采用 C/S 模式进行时间同步的^[4]。服务器端接收到客户端周期性发出的带有本地时间戳的 NTP 报文后,向客户端应答一个带有服务器时间戳的报文,根据以上报文携带的时间戳信息来计算客户端与服务器之间的时间偏移量(Offset)以及时间同步过程中的网络传输延时时间(Delay)这两个关键参数。同步原理如图 1 所示。

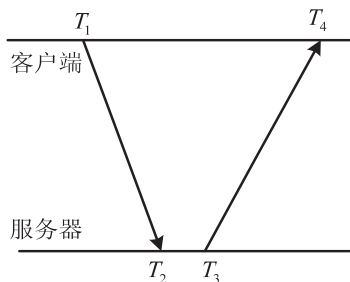


图 1 NTP 协议同步原理图

首先假定服务器的时间是精确的国际标准时间(UTC),而客户端与服务器之间的时间偏移量为 Offset,同步报文从客户端发送到服务器端经历网络路径延时时间为 Delay₁,而从服务器端发送应答报文到客户端的网络路径延时时间为 Delay₂,总的网络路径延时时间为 Delay = Delay₁ + Delay₂。

同步过程如下,首先客户端向服务器发送一个 NTP 同步请求报文,并加盖了其自身离开客户端估计时间 T₁;当服务器收到该报文时加盖了本地的接收时间戳 T₂,同时服务器向客户端回传一个携带本地发送时间戳 T₃ 的响应报文;客户端收到该报文后,加盖了客户端本地时间戳 T₄^[5]。

综上所述,可以得出以下方程:

$$T_2 - T_1 = \text{Offset} + \text{Delay}_1 \quad (1)$$

$$T_4 - T_3 = \text{Delay}_2 - \text{Offset} \quad (2)$$

$$\text{Delay} = \text{Delay}_1 + \text{Delay}_2 \quad (3)$$

为了简化问题,假设从客户端到服务器的网络传输延时等于从服务器到客户端的网络传输延时,即:

$$\text{Delay}_1 = \text{Delay}_2 = \text{Delay}/2 \quad (4)$$

那么以上三式可以简化为两式:

$$T_2 - T_1 = \text{Offset} + \text{Delay}/2 \quad (5)$$

$$T_4 - T_3 = \text{Delay}/2 - \text{Offset} \quad (6)$$

则由上述公式可推导出:

$$\text{Offset} = [(T_2 + T_3) - (T_1 + T_4)]/2 \quad (7)$$

$$\text{Delay} = (T_4 - T_1) - (T_3 - T_2) \quad (8)$$

NTP 协议包含一个 64 bit 的无符号定点数协调世界时时间戳。整数部分用高 32 bit 表示,对应于从基准时间(1900 年 1 月 1 日 00:00:00)直到当前的秒累计计数值;小数部分用低 32bit 表示,其时间最小分辨率是 200 ps。相对于时间基准,到 2036 年时,64bit 数将发生溢出,即每 136 年这 64 bit 的字段将归零,到时将废除此时间戳或重新定义新协议^[5]。

1.2 IEEE1588 协议

1.2.1 IEEE1588 协议概述

2002 年出现的 IEEE1588 (Precision Time Protocol, PTP)^[6] 协议,全称是“网络测量和控制系统精密时间同步协议标准”,简称 PTP (Precision Time Protocol) 协议。相对于 NTP 协议,PTP 为网络上亚微秒级的同步精度的设备提供一个标准方法,确保事件和时间戳在所有设备上使用相同的时间基准^[7]。

IEEE1588 精准时钟同步协议通常可以采用纯软件或纯硬件的方式实现。纯硬件的方式实现,采用硬件编程的方式,可以避免操作系统的调度和内存管理、UDP 和 IP 等协议的封装、网卡控制器里 FIFO(先进先出)队列的排队和总线仲裁等^[8]过程的影响,在网络的 MAC 层获得时间戳,能够达到纳秒级的同步精度^[9]。常见的纯硬件的实现又可分为通过 FPGA 实现和通过集成有 PTP 协议的网络收发芯片实现,同步精度高,但是需要有相应的硬件支持,成本高,且开发难度也较大;而采用纯软件方式实现的 IEEE1588 精准时钟同步协议,通用的方案是采用开源的 IEEE1588 协议实现代码(PTPd),开发难度大大降低;但因其是在协议上层获取时间戳信息,同步精度达不到纯硬件的实现方式的纳秒量级。

截止到目前,IEEE1588 协议已经发展到了第二版—IEEE1588V2。与 IEEE1588V1 相比,IEEE1588V2 引入了点延时机制,用来测量一对端口的平均路径延时时间;时间戳的表示由 64 bit 有符号数进行,时间分辨率可以达到 $1/2^{16}$ ns;引入了透明时钟模型以便将报文在透明时钟内的延时时间也能累加到时间修正域中,并采用了双向信道,以上措施的采用大大提高了同步精度。

文中在纯软件实现 IEEE1588 协议的基础上,将其与纯软件实现的 NTP 协议进行同步精度的对比,以体现纯软件实现的 IEEE1588 协议(PTPd)的优势。

1.2.2 IEEE1588 协议的同步机理

IEEE 1588 主要定义了四种多点传送的时钟报文类型:同步报文(Sync)、跟随报文(Follow_Up)、延迟请求报文(Delay_Req)和延迟请求响应报文(Delay_

Resp)^[10]。

IEEE1588 协议是通过在主从时钟之间传输同步报文来实现同步,主要分为两个阶段:偏差测量阶段和延时测量阶段^[11]。

第一阶段:偏移测量阶段,用来修正主时钟和从时钟之间的时间偏差^[12]。同步过程如下,主时钟以多播的形式周期性向从时钟发送 Sync,主时钟紧接着又发送携带了 Sync 准确发送时间 T_{m1} 的 Follow_Up。从时钟收到 Sync 后记下 Sync 的准确到达时间 T_{s1} 。则由 T_{m1} 和 T_{s1} 可计算出主、从时钟之间的 Offset,这里由于同步报文的 Delay 未知,先假设 $\text{Delay} = 0$,这里同样假设网络是对称的,即从从时钟到主时钟的网络传输延时等于从主时钟到从时钟的网络传输延时。则主从时间偏差值的计算公式是:

$$\text{Offset} = T_{s1} - T_{m1} - \text{Delay} \quad (9)$$

第二阶段:延时测量阶段,用来测量主、从时钟间的 Delay。从时钟收到 Follow_Up 后,向主时钟发出一个 Delay_Req,同时记下 Delay_Req 精确发送时间 T_{s3} 。主时钟收到 Delay_Req 后记录下 Delay_Req 的精确到达时间 T_{m3} ,主时钟紧接着发送 Delay_Resp,把 T_{m3} 传递给从时钟。根据 T_{s3} 和 T_{m3} ,则可计算出从时钟到主时钟之间的网络传输延时时间。主时钟到从时钟的传输延时根据上一次同步报文的发送来计算,则:

$$\text{Delay} = [(T_{s2} - T_{m2}) + (T_{m3} - T_{s3})]/2 \quad (10)$$

经过 T_{m4} 到 T_{s4} 以及 T_{m5} 到 T_{s5} 的调整过程,最后主从时钟达到了同步。

时钟同步在实际通信活动中是周期性的行为。即便主、从时钟达到同步以后,从时钟依然要对主时钟发出的每一个时钟同步报文进行计算。如果计算出的偏移值不等于 0,说明两者之间的时钟又处于不同步状态,需要重新按照上面的步骤进行时钟同步^[13]。

2 PTPd 简介

自从 IEEE1588 诞生后,其纯硬件实现方式更多地受到学者和工程师的关注,纯软件的实现(PTPd)往往被人们忽视。文中使用纯软件实现的 IEEE1588 协议,即由著名的开源软件网站 SourceForge. Net 负责维护的开源的 IEEE1588 协议—PTPd。PTPd 包含了两个版本:Version₁用于支持 IEEE1588-2002,Version₂用于支持 IEEE1588-2008^[14]。文中采用 PTPd1.1.0 版代码,由于 PTPd 代码全部是由 C 语言开发,因此可以很方便地移植到 32 位或 64 位系统上,也很适合嵌入式系统。PTPd 的时钟伺服系统如图 2 所示。

由图 2 可知,PTPd 时钟同步测试程序是在用户空间加盖的发送时间戳,又由于嵌入式操作系统的存在,则同步的结果不可避免地会受到通信协议栈以及操作

系统任务调度延时的影响,因此,高精度的时钟同步可以分步实现,即在 CPU 这一层上面只实现同步精度达

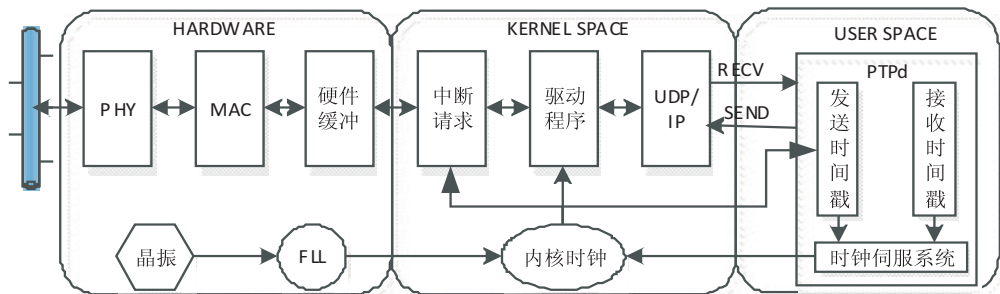


图 2 PTPd 的时钟伺服系统

到几十微秒级或微秒级的同步过程,更精确的纳秒级的同步可由 FPGA 实现,实现过程在这里不予详细说明。

PTPd 程序主要由 IEEE1588 协议引擎、最佳主时钟算法、时钟伺服系统、数据包的处理、网络服务、时间戳模块、启动的初始化、定时器以及其他模块组成。它们之间的关系如图 3 所示。

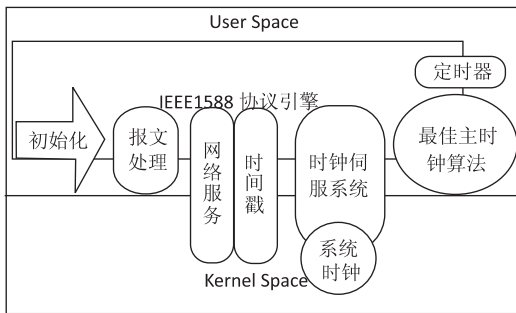


图 3 PTPd 的整体结构

PTPd 各个模块的功能如下:

(1) IEEE1588 协议引擎状态机。

主要由 protocol. c 实现。根据系统当前运行的状态,保证状态机在正常状态下处于各个状态之间的循环切换状态,并且能在发生错误的情况下退出循环。

(2) 初始化。

由 dep/startup. c 实现,主要功能是对 PTPd 程序执行前对运行环境等进行初始化操作。

(3) 报文处理。

由 dep/msg. c 函数实现,它的主要功能是根据协议的规范将欲发送的报文打包,同时将接收到的报文进行解包处理。

(4) 网络服务。

由 dep/net. c 实现,初始化并建立 socket 套接字,并建立 UDP 网络连接;发送、接收报文,并获取时间戳信息。

(5) 时钟伺服系统。

由 dep/servo. c 函数实现,它的主要功能是根据服务器至客户端时钟之间的网络传输延时和客户端至服务器时钟之间的网络传输延时,计算出服务器与客户

端之间的时钟偏差,并对本地时钟进行修正。

(6) 最佳主时钟算法。

由 bmc. c 函数实现 IEEE1588 协议定义的最佳主时钟算法,该函数返回经过自身仲裁的服务器、客户端时钟各自的状态。

(7) 定时器。

由 dep/timer. c 函数实现,用于控制客户端时钟发送 Delay_Req 报文和服务器时钟发送 Sync 报文的周期,以及判断 Sync 报文接收是否发生超时等。

综上所述,不难看出,NTP 协议与 IEEE1588 协议的同步机理还是有所区别的,除了 IEEE1588 协议的软件 PTPd 会根据最佳主时钟算法选取一个相对较稳定的主时钟外,在 IEEE1588 协议中,主时钟发送 Sync 后,还紧接着发送了 Follow_Up,在 Follow_Up 中,携带了 Sync 精确的发送时间,而 NTP 协议则用 Sync 自身携带的本报文的预计发送时间,因此,NTP 协议的同步精度要比 IEEE1588 协议(包括 PTPd)同步精度低。

3 同步精度测试

为了更好地说明 PTPd 的同步精度,在开始测试 PTPd 同步精度之前,首先测试一下 NTP 的同步精度。

3.1 NTP 同步精度测试

测试之前,先来了解一些关于时间的术语:

GMT 时间:格林威治时间,国际标准时间,中国位于东八区,时间为 GMT+8,即北京时间;

UTC 时间:世界协调时间,可认为跟 GMT 是相同的含义;

时间戳:NTP 服务器上的时间戳为从 1900 年 1 月 1 日 0 时 0 分 0 秒开始到至今的秒数(UTC 时间);由于 UNIX 操作系统是 1970 年正式发布的,因而 PC 机上的时间戳为 1970 年 1 月 1 日 0 时 0 分 0 秒到至今的秒数(因此也叫 UNIX 时间戳),需在 PC 机的时间戳秒数的基础上加上从 1900 年 1 月 1 日 0 时 0 分 0 秒到 1970 年 1 月 1 日 0 时 0 分 0 秒的秒数:即 $3\ 600\ s \times 24\ h \times (365\ days \times 70\ years + 17\ days)$ 。

NTP 是基于 UDP 协议进行传输的,网络端口号为

123,所涉及的时间同步报文均封装在UDP协议报文中进行传输。

编程实现NTP的本地客户端代码,根据式(1)~(8)可以计算出客户端与NTP服务器之间的Offset以及单程Delay/2。

在NTP同步精度验证过程中,选用远程NTP同步服务器,来同步本地PC机上的时钟。本地PC机的操作系统为Fedora20,首先调整本地时间超出标准时间10 min左右,然后Fedora进入root模式,执行NTP的客户端同步程序,同步周期为3 s,同步结果如图4所示。

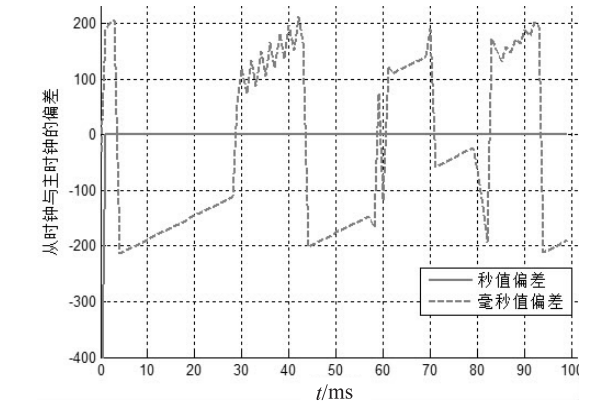


图4 NTP协议同步结果

由图4可以观测到,同步过程开始后,秒级偏差在同步算法的调整下迅速归零,而毫秒级的偏差则以 $\pm 200\text{ ms}$ 为区间在零值附近震荡,本地时间被调整为标准时间。从图4不难看出,NTP协议的同步精度为几百毫秒级,因此NTP协议可以应用于对同步精度要求不太高的领域中,如网络授时系统。

3.2 PTPd同步精度测试

在这里,利用某型号航天器嵌入式主控软件的1553B仿真测试前端和SPI仿真测试前端作为PTPd同步精度测试的主、从时钟模块,用以太网将它们连接起来。

将Linux操作系统移植到ARM11(S3C6410)处理器上,即:移植Bootloader(uboot)、裁剪标准Linux内核生成内核映像文件以及制作支持读写操作的文件系统(YAFFS2),具体移植过程在此不予详述。

将编译好的Bootloader、内核映像文件以及文件系统烧写到ARM11(S3C6410)的存储芯片中。

交叉编译PTPd1.1.0代码,并下载到嵌入式测试前端板中,PC机系统时间为北京时间(GMT+8),嵌入式仿真测试前端系统时间为嵌入式Linux系统的开机时间,即2000年1月1日00:00:00。

先启动作为主时钟(同步服务器)的程序,再启动作为从时钟(同步客户端)的程序,具体操作如下:

主时钟端:#sudo ./ptpd -a 3,12 -y 0 -p -c

从时钟端:#sudo ./ptpd -a 3,12 -y 0 -g -c
IEEE1588精准时钟同步协议的同步结果如图5和图6所示。图5为PTPd开始同步直到主从时钟趋于稳定的过程,而图6为主从时钟同步趋于稳定后的放大的结果。

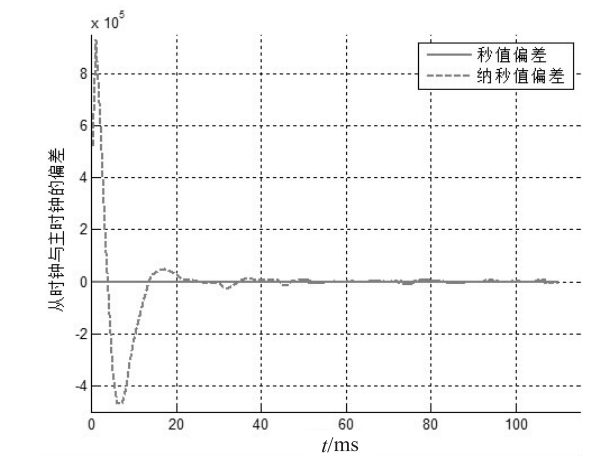


图5 PTPd同步结果图

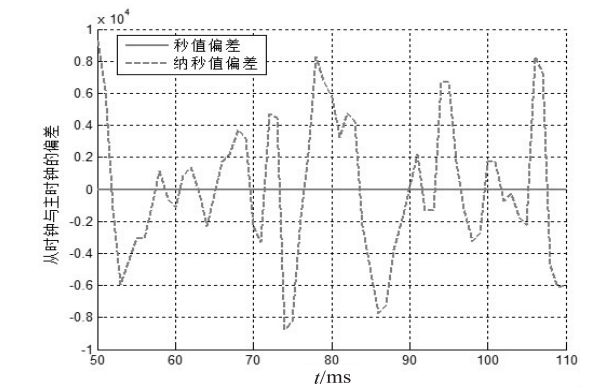


图6 PTPd同步趋于稳定后局部放大图

当同步过程开始后,PTPd的时钟伺服系统不断计算主从时钟的偏差与网络路径传输延时,并不断调整本地时钟。从图5可以观测到,主从时钟的秒级时间差在时刻0时为非零值,在大于0时刻,秒级时间差在同步算法的调整下迅速归零;纳秒级时间差在经过一段时间的正负调整后,逐渐逼近零点。从图6可以观测到,在主从时钟调整趋于稳定后,秒级时间差为0,纳秒级时间差在正负10微秒区间来回震荡,即能够达到微秒级的同步精度。

这里由于嵌入式Linux并不是硬实时的操作系统,再加上PTPd在应用层获取时间戳,因此中断服务的延时和网络协议栈的抖动都会对同步精度造成影响。因此在CPU负载过大时,会导致系统中断延时时间不能够得到确切的保证,从而影响时钟的滴答和消息的接收,进而影响PTPd同步的精度;再次,由于在实际网络传输时其他计算机数据流的存在,也会导致网络传输延时变大甚至暂时堵塞,这样网络传输的延时是非对称的,也会影响同步精度。但是对于一些对

同步精度要求不是很高的同步系统来说,如 CPU 级的时钟同步,这个精度已经够用了。

4 结束语

综上所述,PTPd 实现的 IEEE1588 精准时钟同步协议的微秒级的同步精度虽然没有达到 IEEE1588 协议规定的纳秒级同步精度,但是它要远远优于传统 NTP 协议的几百毫秒级的同步精度,因此,将 PTPd 实现的 IEEE1588 精准时钟同步协议应用于载人航天某航天器嵌入式软件第三方评测的分布式仿真测试系统前端中,同时配合仿真测试前端的 FPGA,进行分级时钟同步,取得了令人满意的同步精度,顺利完成了对某航天器嵌入式主控软件的第三方测试任务。同时也为 IEEE1588 精准时钟同步协议的推广使用拓宽了道路。

参考文献:

- [1] Cisco Systems, Inc. Network time protocol; best practices white paper[EB/OL]. 2008. <http://www.cisco.com/c/en/us/support/docs/availability/high-availability/19643-ntpm.html>.
- [2] 李晓珍. 基于 IEEE1588 的网络时间同步系统研究[D]. 北京:中国科学院,2011.
- [3] Li Dejun, Wang Gang, Yang Canjun, et al. IEEE 1588 based time synchronization system for a seafloor observatory network[J]. Journal of Zhejiang University Sci C, 2013, 14(10): 766-776.
- [4] Zhang Xiangli, Tang Xiaoqi, Chen Jihong. Time synchroniza-

(上接第 174 页)

- [2] 王钦辉, 叶保留, 田 宇, 等. 认知无线网络中频谱分配算法[J]. 电子学报, 2012, 40(1): 147-154. [3] 刘婷婷. 认知无线电中关键技术的研究[D]. 南京: 南京理工大学, 2011.
- [4] 王 超, 陈少刚, 武明川. 认知无线电下的频谱管理建议[J]. 中国科技信息, 2014(22): 202-203.
- [5] 张奎鹏. 认知无线电中的协作频谱感知关键技术研究[D]. 成都: 电子科技大学, 2012.
- [6] 张 培, 孟庆民, 朱卫平. 一种可以减少时间的频谱感知方法[C]//2010 年通信理论与信号处理学术年会. 北京: 电子工业出版社, 2010: 238-242.
- [7] 夏东瀛. 认知无线电系统中的频谱分配算法的研究[D]. 上海: 东华大学, 2013.
- [8] Amir G, Elvino S S. Opportunistic spectrum access in fading channels through collaborative sensing[J]. Journal of Communications, 2007, 2(2): 71-82.
- [9] Liang Y C, Zeng Z, Pech E C Y, et al. Sensing-throughput tradeoff for cognitive radio networks[J]. IEEE Transactions on

tion of hierarchical real-time networked CNC system based on ethernet/internet[J]. International Journal of Advanced Manufacturing Technology, 2008, 36(11-12): 1145-1156.

- [5] 陈 敏. 基于 NTP 协议的网络时间同步系统的研究与实现[D]. 武汉: 华中科技大学, 2005.
- [6] IEEE Std 1588-2008. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems[S]. New York: IEEE, 2008.
- [7] Zhong Chongquan, Jin Haibo, Han Yechen, et al. Research and implement of distributed network motion control system[J]. Informatics in Control, Automation and Robotics, 2011, 2: 383-391.
- [8] 张洪源. 基于 IEEE1588 的数字化变电站时钟同步技术的应用研究[D]. 成都: 西南交通大学, 2010.
- [9] 魏 丰, 孙文杰. IEEE-1588 协议时钟同步报文的精确时间标记方法研究[J]. 仪器仪表学报, 2009, 30(1): 162-169.
- [10] 陈永标. IEEE 1588 时间精确同步协议(PTP)在电力系统应用的可行性研究[D]. 上海: 上海交通大学, 2009.
- [11] 常夏勤. IEEE1588 时间同步协议分析与实现[D]. 南京: 南京大学, 2011.
- [12] 苏 宇. 嵌入式实时测试前端的设计与实现[D]. 北京: 中国科学院大学, 2013.
- [13] 张亚超. 基于 IEEE1588 协议的嵌入式网络节点研究与实现[D]. 广州: 华南理工大学, 2010.
- [14] 宋升金, 许永辉, 姜守达. 基于 PTPd 改进的高精密时钟同步实现[J]. 计算机工程与应用, 2013, 49(2): 72-75.


Wireless Communications, 2008, 7(4): 1326-1337.

- [10] Li Meiling, Yuan Chaowei, Li Lin, et al. Performance analysis and optimization of cooperative spectrum sensing for maximizing secondary throughput[J]. Journal on Communications, 2011, 32(2): 53-60.
- [11] Letaief K B, Zhang W. Cooperative communication for cognitive radio networks[J]. Proceedings of the IEEE, 2009, 97(5): 878-893.
- [12] 刘 鑫, 谭学治, 马 琳. 认知无线电多簇联合频谱感知算法[J]. 哈尔滨工业大学学报, 2013, 45(1): 50-54.
- [13] 岳文静, 郑宝玉, 孟庆民, 等. 衰落信道下提高协作频谱感知能力的方法[J]. 中国科学: 信息科学, 2011, 41(2): 207-218.
- [14] Kozal A S B, Merabti M, Bouhafs F. Spectrum sensing-energy tradeoff in multi-hop cluster based cooperative cognitive radio networks[C]//Proc of 2014 IEEE INFOCOM Workshops on green cognitive communications and computing. [s. l.]: [s. n.], 2014: 765-770.

基于 PTPd 的精准时钟同步技术的研究

作者：[苏宇](#), [胡珩](#), [张涛](#), [王金波](#), [SU Yu](#), [HU Heng](#), [ZHANG Tao](#), [WANG Jin-bo](#)

作者单位：[苏宇, 张涛, 王金波, SU Yu, ZHANG Tao, WANG Jin-bo \(中国科学院 空间应用工程与技术中心, 北京, 100094\)](#), [胡珩, HU Heng \(中国科学院 空间应用工程与技术中心, 北京 100094; 中国科学院大学, 北京 100049\)](#)

刊名：[计算机技术与发展](#)

英文刊名：

年, 卷(期): 2016 (1)

引用本文格式: [苏宇](#). [胡珩](#). [张涛](#). [王金波](#). [SU Yu](#). [HU Heng](#). [ZHANG Tao](#). [WANG Jin-bo](#) [基于 PTPd 的精准时钟同步技术的研究](#)[期刊论文]-[计算机技术与发展](#) 2016 (1)