

学校编码: 10384

分类号\_\_\_\_\_密级\_\_\_\_\_

学号: X2011230397

UDC\_\_\_\_\_

厦 门 大 学

工 程 硕 士 学 位 论 文

小型服务器间时间同步软件的  
设计与实现

Design and Implementation of Time Synchronization Software  
Between Small Servers

徐丹丹

指导教师姓名: 曾文华 教授

专业名称: 软件工程

论文提交日期: 2013年4月

论文答辩日期: 2013年5月

学位授予日期: 2013年6月

指导教师: \_\_\_\_\_

答辩委员会主席: \_\_\_\_\_

2013年4月

## 厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下,独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果,均在文中以适当方式明确标明,并符合法律规范和《厦门大学研究生学术活动规范(试行)》。

另外,该学位论文为( )课题(组)的研究成果,获得( )课题(组)经费或实验室的资助,在( )实验室完成。(请在以上括号内填写课题或课题组负责人或实验室名称,未有此项声明内容的,可以不作特别声明。)

声明人(签名):

年 月 日

## 厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

1. 经厦门大学保密委员会审查核定的保密学位论文，  
于 年 月 日解密，解密后适用上述授权。

2. 不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年 月 日

## 摘 要

目前,大多数计算机之间的时间同步都是通过传统的网络通讯的方式进行的,但在一些物理隔离的计算机或网络之间以及无网络的少数几台近距离的计算机之间等情况,进行时间同步就有点困难。因此针对这一问题,可以通过利用串口通讯的方式来解决这一问题,并利用 CPU 时间戳与 Windows API 函数相结合的方法获取精度为  $1\mu\text{s}$  的系统时间及 CPU 的频率来提高时间同步的精度。利用回归自比较法进行了测试,测试结果表明同步精度较高,运行稳定。

本文在深入研究 NTP 协议的工作原理和工作模式的基础上,尝试着设计开发了一种简单的串口通信中基于 NTP 协议的服务器和客户端软件,实现了小型服务器间的时间同步。

本文研究内容主要包括:

- (1)研究了 NTP 协议的工作原理及工作方式。
- (2)在熟悉了 NTP 算法的基础上,了解如何进行本地时钟时间同步的硬件调整,以完善整个时间同步过程。研究了回归自比较对时方法。
- (3)充分物理隔离的计算机或网络之间以及无网络的少数几台近距离的计算机之间时间同步的问题,提出了一种简单的串口通信中的时间同步思路。
- (4)串口通信下的程序开发,研究在 VC++中的串口控制和相关协议的实现。并完成了整个系统的软件开发。

**关键词:** 时间同步; 串口通信; 时间戳; 物理隔离; 回归自比较法

## Abstract

At present, most of the time synchronization between computers are carried out through the network communication of traditional way, but between the physical isolation of the computer or network and the network without a few machines close to the computer, the time synchronization is a bit more difficult. Therefore, to solve this problem, can solve this problem by using serial communication mode, and by using CPU timestamp and Windows API functions combined with precision of  $1\ \mu\text{s}$  system time and the frequency of CPU to improve the accuracy of time synchronization. Using the regression from the comparison test, test results show that high synchronization accuracy, stable operation.

Based on the working principle and working mode of the NTP protocol deeply, we designed and developed a simple serial communication of NTP protocol based on the server and client software, to achieve a small server time synchronization.

This paper mainly includes:

- (1) The working principle and working mode of NTP protocol.
- (2) Based on the NTP algorithm, how to adjust local hardware clock, in order to improve the whole time synchronization process. Study on the comparative method of time synchronization in regression.
- (3) The time between the full physical isolation computer or network and the network without a handful of close computer synchronization problem, presents a simple serial communication in time synchronization method.
- (4) The development of serial communication program, implementation of serial control in VC and related protocols. And the completion of the entire system software development.

**Key words:** Time Synchronization; Serial Communication; Timestamp; Physical Isolation; Regression From Comparative Law

# 目 录

<b>第一章 绪论</b> .....	<b>1</b>
1.1 研究背景和意义 .....	1
1.2 国内外研究现状 .....	1
1.3 论文的主要研究内容 .....	3
1.4 论文结构安排 .....	3
<b>第二章 相关技术分析</b> .....	<b>5</b>
2.1 时间同步协议概述 .....	5
2.1.1 NTP 协议 .....	5
2.1.2 SNTP 协议 .....	5
2.1.3 PTP 协议 .....	6
2.2 NTP 的工作原理和工作模式 .....	8
2.2.1 网络时间服务的层状结构 .....	8
2.2.2 NTP 协议的对时方法 .....	9
2.2.3 NTP 的工作模式 .....	10
2.2.4 NTP 时间戳格式 .....	11
2.2.5 NTP 报文格式 .....	11
2.3 串口通信 .....	13
2.3.1 串口通信概述 .....	13
2.3.2 串口通信在 VC++ 中的控制技术 .....	16
2.4 本章小结 .....	19
<b>第三章 系统需求分析</b> .....	<b>21</b>
3.1 系统业务需求分析 .....	21
3.2 系统功能性需求分析 .....	22
3.3 系统非功能需求分析 .....	23
3.4 本章小结 .....	24
<b>第四章 系统总体设计</b> .....	<b>25</b>
4.1 系统架构设计 .....	25
4.1.1 服务器端工作流程 .....	25
4.1.2 客户端工作流程 .....	26
4.2 系统功能模块设计 .....	28
4.3 系统数据库设计 .....	29
4.4 本章小结 .....	30
<b>第五章 系统详细设计与实现</b> .....	<b>31</b>
5.1 系统开发、运行环境及软件 .....	31
5.1.1 系统开发环境 .....	31
5.1.2 系统运行环境 .....	31
5.2 系统详细设计 .....	32

5.2.1 串口通信.....	32
5.2.2 高精度时间戳提取.....	34
5.2.3 同步实现.....	38
<b>5.3 服务器端软件使用 .....</b>	<b>40</b>
5.3.1 服务器端系统的硬件连接.....	40
5.3.2 软件安装.....	42
5.3.3 软件启动.....	46
5.3.4 运行操作.....	47
<b>5.4 客户端软件设计 .....</b>	<b>63</b>
5.4.1 硬件连接.....	63
5.4.2 软件安装.....	64
5.4.3 软件启动.....	68
5.4.4 操作说明.....	69
<b>5.5 本章小结 .....</b>	<b>81</b>
<b>第六章 系统测试 .....</b>	<b>82</b>
6.1 测试方法 .....	82
6.2 测试环境 .....	82
6.3 测试结果 .....	82
6.3.1 测试结果及分析.....	82
6.3.2 串口错误.....	84
6.3.3 配置错误.....	86
6.3.4 系统冲突.....	86
6.3.5 实验分析总结.....	86
6.4 本章小结 .....	87
<b>第七章 总结与展望 .....</b>	<b>88</b>
7.1 总结 .....	88
7.2 展望 .....	89
<b>参考文献 .....</b>	<b>90</b>
<b>附 录.....</b>	<b>93</b>
<b>致 谢.....</b>	<b>95</b>

# Contents

<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Background and Significance .....	1
1.2 Overview of Domestic and Foreign .....	1
1.3 Main Content.....	3
1.4 Organizational Structure .....	3
<b>Chapter 2 Related Technology Introduction.....</b>	<b>5</b>
2.1 Time Synchronization Protocol Overview .....	5
2.1.1 NTP protocol .....	5
2.1.2 SNTP protocol.....	5
2.1.3 PTP protocol.....	6
2.2 NTP Working Principle And Working Mode.....	8
2.2.1 The layered structure of network time service .....	9
2.2.2 Method of time synchronization in NTP protocol.....	9
2.2.3 NTP model .....	10
2.2.4 NTP timestamp format.....	11
2.2.5 NTP message format.....	11
2.3 serial communication.....	13
2.3.1 serial communication overview .....	13
2.3.2 serial communication control technology in VC++.....	16
2.4 Summary.....	19
<b>Chapter 3 System Requirements Analysis.....</b>	<b>21</b>
3.1 system demand analysis .....	21
3.2The functional requirements of the system analysis.....	22
3.3 the system non-functional requirements analysis.....	23
3.4 Summary .....	24
<b>Chapter 4 System Design .....</b>	<b>25</b>
4.1 system architecture design .....	25
4.1.1The server process .....	25
4.1.2 client work process .....	26
4.2The function modules of the system design.....	28
4.3 the system database design .....	29
4.4 Summary .....	30
<b>Chapter 5 Detailed System Design and Implementation .....</b>	<b>31</b>
5.1 system development environment .....	31
5.1.1 system development environment .....	31
5.1.2 The operating system environment .....	31
5.2 Detailed system design.....	32



5.2.1 Serial communication.....	32
5.2.2 High-precision time interception.....	34
5.2.3 Achieve synchronization .....	38
<b>5.3 Server software design.....</b>	<b>40</b>
5.3.1 The server system hardware connection.....	40
5.3.2 Software Installation.....	42
5.3.3 Software startup .....	46
5.3.4 Running operation.....	47
<b>5.4 Client software design .....</b>	<b>63</b>
5.4.1 Hardware connection .....	63
5.4.2 Software Installation.....	64
5.4.3 Software startup.....	68
5.4.4 Operating Instructions.....	69
<b>5.5 Summary .....</b>	<b>81</b>
<b>Chapter 6 System Test.....</b>	<b>82</b>
6.1 Test method .....	82
6.2 Test environment.....	82
6.3 Test results of .....	82
6.3.1 Test results and analysis.....	82
6.3.2 Serial ports error.....	84
6.3.3 Configuration error.....	86
6.3.4 System conflict.....	86
6.3.5 Summary of the experimental analysis.....	86
6.4 Summary .....	87
<b>Chapter 7 Conclusions.....</b>	<b>88</b>
7.1 Summary.....	88
7.2 Outlook .....	89
<b>References.....</b>	<b>90</b>
<b>Appendix .....</b>	<b>93</b>
<b>Acknowledgements.....</b>	<b>95</b>

## 第一章 绪论

### 1.1 研究背景和意义

对于一个由多计算机组成的系统中,其中各个节点计算机都有自己的系统时间。计算机网络控制系统中各子系统时钟的一致性,是网络控制系统中的核心问题之一,它的准确性对网络控制系统的实时性和准确性有重大的影响。但是由于系统在运行中受到设计缺陷、环境温度变化、电磁干扰、负载等多种因素的影响,计算机的时钟大多数是不精确的,随着时间的推移,这种误差逐渐累积,时间的偏差将会越来越大。在一些比较重要的应用中(如实时数据采集、监控、工业控制网络以及电信等)<sup>[1][2][3]</sup>,时间精度要求控制在毫秒级、亚毫秒级甚至更高的精度,因此时间同步变得尤为必要。于是,实现一个精确的系统时间变得非常重要。时钟同步实现方法,常用的有硬件和软件方法:硬件同步是通过各仿真节点都连入专用的时钟信号线来进行的,它精度非常高,但成本也高,操作复杂;软件同步方法,是完全利用软件来完成分布式系统中各时钟的同步。对于软件同步方法,目前常用的时间协议有NTP协议<sup>[4]</sup>、SNTP协议以及PTP协议<sup>[6]</sup>等。对于这些时间同步协议的一般实现方法是通过计算机网络等方式通讯实现的,但是由于计算机网络的一些潜在的威胁,如偶然或恶意的原因遭到破坏、更改、泄露,使网络系统不能连续可靠性地正常运行,从而导致时钟同步系统不能正确运转。因此本文提出的是利用串口的通讯方式来实现NTP协议中确定延时与偏移的过程,从而实现计算机间相对时间同步。使用串口可以用在某些计算机网络控制系统中,尤其是一些需要网络物理隔离的子系统间各模块的同步,比如一些监控系统中的备份服务器等;另外,利用串口通讯,实现起来简单、系统更稳定以及通讯延时更确定,可以达到比较好的精度和稳定性。

### 1.2 国内外研究现状

NTP的发展在国际上可以分为三个时期:

#### 1. NTP v1 之前的工作

最早的时间协议的实现是记载在 Internet Engineering Note [IEN-173] 之中，具有白毫秒级的精确度。在此之后出现了第一个被命名为 DCNET 互联网时间服务的时间协议的规范： [RFC-778]，它是借助于 Internet Control Message Protocol (ICMP)，即互联网控制消息协议中的时间戳和应答消息提供服务的。NTP 这个名称的第一次出现是在 [RFC-958]之中，也被称为 NTP v0。其目的是为 ARPA 提供时间同步服务。它对于本地时钟的误差和精密度等运算、网络上的分组数据包及其消息格式等进行了描述。但是不对误差进行补偿，也没有规定同步的算法。

## 2. NTP v1 到 NTP v3

1988 年 6 月出现 NTP v1 的首次完整的描述了 NTP 的规范和相关算法。这个版本开始采用了 Client/Server 模式和对称操作。但它的缺陷是不支持鉴权和 NTP 控制消息。1989 年 9 月推出了 NTP v2。

在这时，DEC 公司也推出了一个数字时间同步服务的时间同步协议，Digital Time Synchronization Service (DTSS)。

1992 年 3 月，NTP v3 诞生，这个版本综合了 NTP 先前版本和 DTSS 的优点，引入了校正原则，并改进了时钟滤波及时钟选择的算法，并使用了广播模式发送时间消息。

## 3. NTP v3 后的进展

NTP v3 发布后的改进版本标注为 xntp3.y(x 表示试验,y 表示第几次修改)。NTP 实现了对计算机操作系统的时钟调整功能。在 NTP v3 发展的同时，对于操作系统核心中的时间保持功能的改进研究有了很大进步。1994 年出现名为精密时间保持的核心模式 (A Kernel Model for Precision Timekeeping)，该模式下可以把操作系统的时钟保持在微秒数量级精确度上。于此同时，NTP v4 的改进建议也被提出来了。对通信模式、适配规则、本地时钟调整算法、新的时钟驱动器等方面的改进方向作了具体的描述。

SNTP (Simple Network Time Protocol) 简单网络时间协议是 NTP 发展的另一分支。SNTP 适用于时间精确度要求低于 NTP 的客户机，仅限于使用在时间同步网的终端位置。1992 年诞生的 [RFC-1361] 的 SNTP，它的精确度为秒级。1995 年 3 月，提出的 [RFC-1769] 时间精确度为数百毫秒级。SNTP 的最新规范是 1996 年的

[RFC-2030]，被称为简单网络时间协议 SNTPv4。SNTP 的实现和使用比较简单，尤其对于客户端的实现。许多操作系统直接支持 客户端的 SNTP 协议。

在国内方面，研究和使用的 NTP 的高校和科研院所越来越多。高校的校园网内已经出现了一个网络中心标准时间服务组织（英文简称 CERNET），它的一级和二级时间服务器已经由清华大学、东南大学、北京大学、北京邮电大学共同建立起来。此组织已经在为全国各地区和省市提供便捷准确的时间服务。

### 1.3 论文的主要研究内容

本文研究了时间同步协议 NTP 的相关内容，通过对各种协议的分析确定本课题研究所遵循的协议标准，对影响性能的 NTP 协议的工作原理和工作模式进行了详尽的分析，在对其进行充分了解的基础上结合所掌握的其串口通信技术，最终对系统进行了一个详细的实现过程。在研究的过程中，笔者提出了利用串口通信的方式来实现 NTP 协议中确定延时与偏移的过程，这也是系统开发过程中至关重要的一个决定，通过对同步协议和串口通信的研究，最终实现了计算机间相对时间同步，这也达到了系统最初的设计初衷，满足了用户需求规格说明书的相关要求，这是软件设计的灵魂所在。利用这一思想开发实现了小型服务器间时间同步的软件，并对相关的软件细节进行了说明。在系统的开发完成之后，对系统的功能正确性进行了大量的测试工作，测试工作是为了验证软件功能的正确性，但测试的过程是一个找错的过程，目的旨在发现更多的软件漏洞，找出程序的有错性，而不是证明程序的无错性，这是软件测试工作的基本要求，最后对测试的结果进行了分析，对实验分析的结果进行了总结。给出了详尽的用户使用说明，增强系统的友好性和人机交互性。

### 1.4 论文结构安排

本文内容安排如下：

第一章：分析本课题的研究背景和选题意义，确立本课题研究的必要性，之后在对国内外相关技术的研究及发展现状的分析之下提出了本文的主要研究内容，并对本文的结构安排进行了说明。

第二章：主要介绍了影响本课题研究成果的一些重要的相关技术，主要包括 NTP 协议、SNTP 协议、PTP 协议，同时着重介绍了 NTP 协议的工作原理及工作模式及串口通信技术。

第三章：对本课题的研究进行了需求分析，分别从系统的业务需求、功能需求和非功能需求等方面进行了介绍，严格按照软件工程的观念对系统进行了需求分析。

第四章：在第三章需求分析的基础上实现了对系统的整体设计，分别对系统的功能架构、功能模块和数据库等都进行了说明，这也是软件开发过程中一个非常重要的过程。

第五章：介绍了小型服务器时间同步技术软件的具体实现方法，对本系统的相关使用进行了说明，为用户可以方便的对系统进行相应的操作奠定了良好的基础。

第六章：在软件开发完成的基础上，对系统的各项功能进行了相关的测试，同时与需求分析结果相对比，看系统的功能是否真正达到了用户的需求，以验证系统功能的正确性。

第七章：对全文的工作进行了全面的总结，并对本课题的后续工作进行了下一步的展望

## 第二章 相关技术分析

### 2.1 时间同步协议概述

所谓时间同步,简单的说就是让在系统(或者网络)中每台设备的系统时间同标准时间的误差保持在一定范围内<sup>[7]</sup>。时间标准主要有原子时、世界时以及世界协调时三种<sup>[8]</sup>。时间同步协议有 Time Protocol (RFC 868)、Daytime Protocol (RFC 867)、网络时间协议 NTP (Network Time Protocol)、简单网络时间协议 SNTP (Simple Network Time Protocol) 等。当前的时钟同步系统中,主要使用线面所列三种协议:

#### 2.1.1 NTP 协议

美国 Delaware 大学的 Mills 教授最早提出了 NTP 协议,该协议既能估算消息包在网络中的传输延迟时间,也可估算各个终端之间时钟偏差;NTP 可以为复杂的网络提供高精度的时间服务,把终端机的时间同步到某些时间标准<sup>[9]</sup>。NTP 以服务器 (Server) / 客户机 (Client) 模式进行通信:客户机发出一个请求数据包,服务器收到后回送一个应答数据包。两个数据包都携带有发送和接收的时间戳,客户机依据这四个时间戳来确定其与服务器之间的时间偏差和延时,来修正本地时间<sup>[10]</sup>。

NTP 是一个非常复杂繁琐的协议,可适用于较复杂的网络环境。从 1982 年 NTP 协议被提出,到今天已经有了三十年的发展历史,NTPv4 精确度已经达到了 200ms。NTP 协议与其他时间服务器或终端连接采用三种方式:广播/多播模式 (broadcast/multicast)、客户端/服务器模式 (client/server) 和 symmetric 模式<sup>[11]</sup>。

#### 2.1.2 SNTP 协议

SNTP 协议主要通过交换时间客户端和服务器的时间戳,算出客户端相对服务器的偏差和延时,从而实现时间同步<sup>[12]</sup>。SNTP 协议是简化 NTP 协议,该协议不但能使得对时间同步的开发和应用变得简单,同时也保证了与 NTP 协议相同

的时间精确度。NTP 协议主要在网络中，SNTP 用在不需要实现 RFC1305 所描述的 NTP 的全功能的情况下，SNTP 协议和 NTP 协议版本虽然没有本质不同，但是操作更简单，它采用的是一种无状态远程过程调用。SNTP 是简化了的 NTP 客户端和 NTP 服务器的存取策略，NTP 和 SNTP 的客户端和服务端可以混合通用。两者的客户端不同的是：NTP 客户端能在多服务器模式下操作，而 SNTP 客户端在单服务器下进行。与 NTP 协议相似，SNTP 协议同步精度也是毫秒级，也支持三种工作方式<sup>[13]</sup>。

### 2.1.3 PTP 协议

前面所讲的两协议仅仅是针对毫秒级的时间同步需求系统设计的，不能满足实时控制系统与网络化测试中对时间同步更高精度的要求。于是便产生了更精密时间同步协议 PTP，该协议有针对性的解决了测试与控制系统中节点在空间分布上相对集中、微秒级至亚微秒级的同步精度、自主管理、高端设备和廉价低端设备都能够访问等需求。

线路延时对时间同步精度造成的影响是各种同步算法中必须要考虑到的问题。线路延时主要有两部分组成：物理网络上的传输延时和通信栈延时。传输延时是由于分布式测控系统中的测控设备在网络中所处位置、布线方式、布线长度以及目前网络技术中的固有问题造成的<sup>[14~15]</sup>，网络交换设备如集线器等在网络中的使用也会产生抖动和延时。根据 IEEE 802.3 协议，同步报文的前面都必须有长信息包，集线器所采用的数据包处理方式，严重依赖于网络的负荷情况；理论上的最坏情况数据在传输过程中将会产生大约 122  $\mu$ s 的延迟抖动<sup>[16-17]</sup>。通信栈延时是由于通信栈软件执行过程中的不确定性太多，导致通信栈的延时抖动相对很大。在一般的时间同步系统中，例如 SNTP 协议和 NTP 协议，其总的同步延时由物理线路延时和通信栈延时组成，因此 SNTP 协议和 NTP 协议的同步精度只能达到毫秒级，如图 1-1 所示。

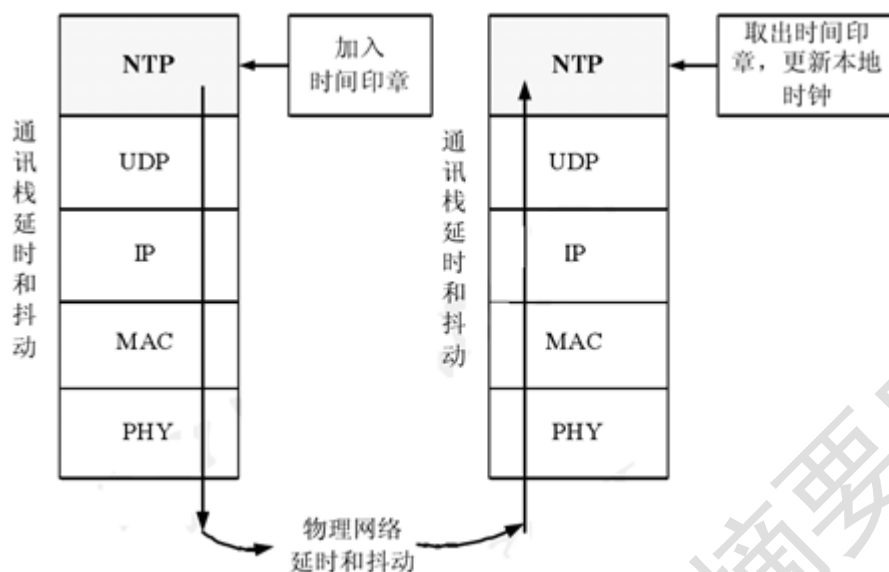


图 2-1 NTP 的同步方式

PTP 协议针对通信栈延时和传输延时的问题，采取了如下的两种方法<sup>[18]</sup>，有效地降低了这两种延时，很大程度的提高了同步精度。第一，PTP 协议采取了点对点的通信方式，通过使用具有边界时钟功能的网络交换设备，使网络交换设备成为网络中的一个节点，这样网络中就全部都是点对点的通信连接，主时钟和从时钟之间也仅有物理网络上的延时抖动，这种点对点的操作能够提供更高的精确度。如图 1-2 所示。

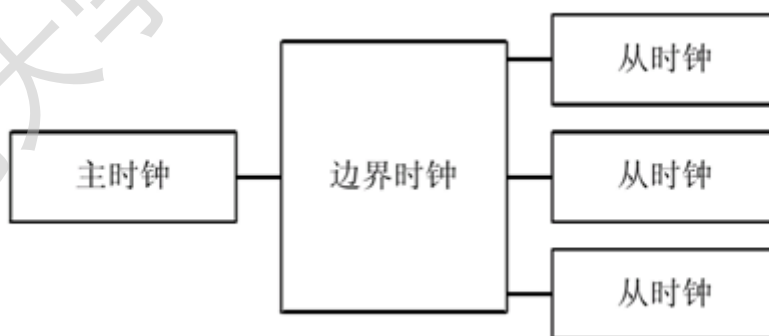


图 2-2 边界时钟的使用

第二，针对同步方法中存在的通信栈延时问题，PTP 同步协议最先使用了一种新的同步解决方案，其主要思想就是将时间戳的加入时间向网络的底层下移，移到 MAC 层以下，甚至 PHY 层以上，如图 1-3 所示。这样 PTP 同步过程中的通信栈延时几乎可以消除，总延时主要为物理层延时，由于物理层延时的抖动比



Degree papers are in the "[Xiamen University Electronic Theses and Dissertations Database](#)". Full texts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to [etd@xmu.edu.cn](mailto:etd@xmu.edu.cn) for delivery details.

厦门大学博硕士论文摘要库