

Virtual Scoping: 轻量级可扩展的 TSN 时间同步精度测量方法

孙寅涵, 孙志刚, 付文文, 李文超, 黄容

(国防科技大学计算机学院, 湖南长沙 410073)

摘要: 针对现有基于时钟脉冲的时间同步精度测量方法存在测量系统构成复杂、测量规模受仪器采集接口数量限制、测量数据无法支持实时网络管理控制等问题, 基于时间敏感网络 (TSN) 集中式管控架构, 提出了一种轻量级可扩展的时间同步精度测量方法 Virtual Scoping (VS), 能够在网络控制平面虚拟化示波器的功能。该方法定义了可用于精确测量同步精度的参数 DriftTime, 该参数由同步报文发送时间戳、驻留时延、链路时延等协议状态数据计算。最后在网络控制平面实现 V-Scope 工具并部署在基于 OpenTSN 开源项目搭建的原型系统进行实验。实验结果表明, V-Scope 工具与专业测试仪的测量结果具有一致性。此外, 以精度优化为例对 VS 方法在支持网络管理控制功能的可行性和优越性进行探讨和论证, 设计实现了一种基于 V-Scope 的频差补偿算法, 最高可将原型系统同步精度提升 32.3%。

关键词: 时间敏感网络; 时间同步; 同步精度测量; 网络管理控制

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2024165

Virtual Scoping: lightweight and scalable TSN time synchronization accuracy measurement method

SUN Yinhan, SUN Zhigang, FU Wenwen, LI Wenchao, HUANG Rong

College of Computer, National University of Defense Technology, Changsha 410073, China

Abstract: Aiming at the problems of existing time synchronization accuracy measurement methods based on clock pulse, such as complex measurement system composition, limited measurement scale by the number of instrument acquisition interfaces, and inability of measurement data to support real-time network management and control, a lightweight and scalable time synchronization accuracy measurement method called Virtual Scoping (VS) was proposed based on the centralized management and control architecture of time-sensitive networking (TSN). Ability to virtualize oscilloscope functions in the network control plane. Firstly, the parameter DriftTime was defined by the method, which could be used to accurately estimate synchronization accuracy. This parameter was calculated from protocol status data such as synchronization message transmission timestamps, residence delay, and link delay. Finally, the V-Scope tool was implemented in the network control plane and deployed in a prototype system built on the OpenTSN open-source project for experimentation. The results demonstrate that V-Scope exhibits only nanosecond-level errors compared to measurements obtained from professional testing equipment. Additionally, the feasibility and advantages of the VS method in supporting network management functions are discussed. As an example of accuracy optimization, a frequency offset compensation algorithm based on V-Scope is designed and implemented, which improves the synchronization accuracy of the prototype system by 32.3%.

Keywords: time-sensitive networking, time synchronization, synchronization accuracy measurement, network management and control

收稿日期: 2024-05-10; 修回日期: 2024-08-17

通信作者: 付文文, fuwenwen@nudt.edu.cn

基金项目: 全国重点实验室基金资助项目 (No.2023-KJWPDL-14)

Foundation Item: The National Key Laboratory Foundation Project (No.2023-KJWPDL-14)

0 引言

随着智能化应用的发展,航电系统、车载控制系统等分布式实时系统对数据传输速率和带宽资源的需求急剧增大^[1-2]。在该背景下,传统以太网因其丰富的带宽资源和良好的可扩展性得到了业界的广泛关注,但是仅能提供尽力而为的通信服务,无法满足应用场景对通信服务实时性和确定性的需求。为此,IEEE 802.1 工作组开始推进时间敏感网络(TSN, time-sensitive networking)技术标准的研究与制定,通过在传统以太网的基础上新增高精度时间同步、时间感知的流量调度等机制,提供时延确定的通信服务^[3]。

高精度时间同步机制是 TSN 提供确定性通信服务的基础。TSN 时间同步技术由 802.1AS 标准^[4]定义,在 IEEE 1588 定义的精确时间协议(PTP, precise time protocol)^[5]的基础上进行修订。该协议使用主从式、逐级的同步模型,设置主时钟(GM, grandmaster)向相邻的从时钟设备发布同步基准时间,从时钟据此修正本地时间并向其相邻的下一级从时钟转发该基准时间,如此循环,直至最终实现全网时钟的时间同步。

TSN 通过预先规划关键流量在传输路径每个设备中发送和接收的时刻以提供确定性通信服务。时间同步精度越高,TSN 能够预先规划的流量规模越大,可承载的关键流量负载也越大^[6]。此外,时间同步精度是监测时钟同步状态的重要依据,网络异常往往会导致精度的骤增和抖动^[7]。因此,研究 TSN 时间同步精度测量方法具有重要的理论意义和现实价值^[8]。

时间同步精度最早由 Lamport 等^[9]进行定义,分布式实时系统的同步精度是系统内任意 2 个设备时钟在任意时刻下示数的差值的最大值。根据定义,精度测量需要同时观测系统内所有设备的时钟,而时钟在物理空间上的分散性使得同时观测十分困难。

目前,大部分精度测量解决方案要求被测设备周期性地输出时钟脉冲信号(如 1PPS 秒脉冲),然后通过示波器等专用仪器对信号进行采集和分析。此类方法主要存在以下 3 个问题。1) 测量系统构成复杂,既要求被测设备提供脉冲输出接口,也需要示波器或精度测试仪等设备通过独立的连线与被测设备接口连接,这在航空航天控制网络

等对设备尺寸、占用空间有严格要求的 TSN 应用场景中是无法容忍的。2) 测量规模受仪器采集接口数量限制,由于示波器采样时钟脉冲数据的接口数量和存储深度受限^[10-11],只能对少量设备进行采样。以实验室常用的示波器 RIGOL-DS1104Z 为例,采样通道数量为 4,脉冲信号波形数据的存储深度范围为 16 KB~32 MB,当采样频率设置为 50 MSaps 时,最多存储 0.64 s 的脉冲数据,只能记录 4 个时钟设备的 4~5 个同步周期(802.1AS 标准建议同步周期长度为 0.125 s)的数据。3) 测量数据无法支持实时网络管理控制,此种测量方法由于设备间的空间隔离性和人员手动操作,导致数据采集与分析的过程耗时长,而时钟属性的改变具有瞬时性,因此测量所得精度数据无法实时应用于性能优化、状态监测、故障诊断等网络管理控制功能。

知识定义网络(KDN, knowledge defined network)^[12]是一种高效的自动化网络管理范式,通过闭环控制架构从网络中收集全局状态数据,使用机器学习等方法将数据转化为知识,最终向网络提供推荐、优化、验证、估计等服务。802.1Qcc 标准^[13]定义了 TSN 集中式管理控制模型,在网络的控制平面设置了集中式用户控制器(CUC, centralized user controller)和集中式网络控制器(CNC, centralized network controller)。此外,802.1AS 标准中定义了丰富的管理对象属性,能够支持控制平面对数据平面设备的时间同步协议进行管理与控制。

为解决现有测量方法存在的问题,本文借鉴了 KDN 自动化闭环和 TSN 集中式管理控制的思想,提出了一种轻量级可扩展的 TSN 时间同步精度测量方法 Virtual Scoping(以下简称 VS),可在网络控制平面虚拟化示波器的功能,定义精确测量时间同步精度的参数 DriftTime。与现有测量方法相比,VS 方法既不依赖专用测量仪器,能够简化测量系统构成,还能够各种规模的 TSN 中进行扩展,进而应用于各种网络管理控制功能。本文主要贡献如下。

1) 提出了一种轻量级可扩展的 TSN 时间同步精度测量方法 Virtual Scoping,在该方法中定义了易于测量且能够精确测量同步精度的参数 DriftTime。该参数针对 TSN 同步主从逐级式的特点,

使用协议运行过程中的精确时间戳、驻留时延、链路时延等协议状态数据进行计算, 可用于估计时钟的实际运行频率、实际漂移率等物理属性。

2) 以本文所提方法为核心, 使用 C 语言代码实现 V-Scope 工具并部署在基于 OpenTSN 开源项目构建的原型系统上进行实验, 使用 V-Scope 工具和专业测试仪器分别测量系统各个时钟的同步精度, 结果表明二者仅存在纳秒级的误差, 从理论和现实两个层面证明了 VS 方法的优越性和可行性。

3) 探讨和分析 V-Scope 应用于性能优化、状态监测、故障诊断等网络管理控制功能的可行性和优越性。以性能优化为例, 基于 V-Scope 设计实现了一种频差补偿算法, 采用频率补偿的方式有效提升了系统的同步性能, 并在原型系统上进行了实验验证。

1 相关工作

1.1 TSN 时间同步

1.1.1 基本原理

PTP 时间同步有 2 种同步模式: E2E (end-to-end) 模式和 P2P (peer-to-peer) 模式。在 E2E 模式下, 网络中的所有非主时钟通过与主时钟交互协议报文计算自身与主时钟设备之间的链路时延和时钟偏差。在 P2P 模式下, 非主时钟仅测量与相邻 (端口直连) 设备的单跳链路时延。相较于 E2E 模式, P2P 模式受链路时延非对称性的影响较小, 可以达到更优的精度。在通信路径更改时, P2P 模式下的从时钟不需要重新测量链路时延, 根据同步报文仍然可以继续同步^[14]。

出于对同步精度和高可用性的考虑^[4], TSN 时间同步协议标准采用的是 P2P 模式, 是主从式逐级的同步模型, 主要包含 2 个周期性的过程: 链路时延测量和同步校正。

首先对本文所使用的符号及含义进行介绍, 令 $t_{(p,rx)}^k$ 表示时间戳, 其中字母 t 的上标表示时间戳是在第 k 轮周期产生的, 下标是由 2 个元素构成的二元组 (p,rx) , 第一个元素表示时间戳是由时钟 p 记录的, 第二个元素表示时间戳的具体含义, 示例中 rx 表示接收时间戳, 其他更多含义如表 1 所示。

表 1 符号及含义

符号	含义
N	同步路径上经过的中转设备数量
R	同步路径上在中转设备上产生的所有驻留时延之和
L	同步路径上所有链路时延之和
T	同步的周期长度
$t_{(GM,b)}$	GM 同步过程发布的基准时间
$t_{(p,rx)}$	报文在节点 p 的接收时刻
$t_{(p,tx)}$	报文在节点 p 的发送时刻
$t_{(p,c)}$	节点 p 计算本地时钟修正时间
$t_{(p,u)}$	节点 p 校正修改本地时钟的时间
f_p	节点 p 本地时钟的频率
$l_{<p,q>}$	节点 p 和 q 的本地时钟之间的平均链路时延
$\lambda_{<p,q>}$	节点 p 和 q 的本地时钟之间的频率比
C_p	节点 p 本地时钟

链路时延测量是测量网络中相邻 2 个节点之间的链路传输报文的时延。测量的发起者 p 和响应者 q 经过请求报文和响应报文的交互后, 共记录 4 个时间戳: 请求报文发送时间戳 $t_{(p,tx)}$ 、请求报文接收时间戳 $t_{(q,rx)}$ 、响应报文发送时间戳 $t_{(q,tx)}$ 和响应报文接收时间戳 $t_{(p,rx)}$ 。基于上下行链路时延相等的假设, 使用往返时延 (RTT, round trip time) 的二分之一作为单向链路时延 $l_{<p,q>}$, 如式(1)所示。

$$l_{<p,q>} = \frac{(t_{(p,rx)} - t_{(p,tx)}) - (t_{(q,tx)} - t_{(q,rx)})}{2} \quad (1)$$

同步校正非主时钟设备根据主时钟 GM 同步过程发布的基准时间 $t_{(GM,b)}$ 修正本地时钟的过程。基准时间携带在同步报文中, 因此需要记录或计算报文在传输过程中产生的时延。对于 GM 至同步节点 p 之间共有 N 个中间设备的同步路径, 同步路径上在中转设备上产生的所有驻留时延之和 R 如式(2)所示, 所有链路时延之和 L 如式(3)所示。

$$R = \sum_{i=1}^N (t_{(i,tx)} - t_{(i,rx)}) \lambda_{<GM,i>} \quad (2)$$

$$L = \sum_{i=1}^N l_{<ii+1>} \lambda_{<GM,i>} \quad (3)$$

其中, $t_{(i,tx)}$ 和 $t_{(i,rx)}$ 分别表示中间设备 i 发送和接收同步报文的时间戳, $\lambda_{<GM,i>}$ 表示中间设备 i 和 GM 时钟的频率比, $l_{<ii+1>}$ 是中间设备 i 和 $i+1$ 之间的平均链路时延。

由于计算驻留时延使用的时间戳来自设备的本地时钟, 该时钟与 GM 存在频率偏差, 因此需要将本地计算得到的驻留时延乘以本地时钟与 GM 时钟的频率比 λ , 才能准确计算出 $t_{(GM,b)}$ 产生之后经过

的时间。同理，链路时延也应乘以 λ 。

GM 频率比 $\lambda_{<GM,p>}$ 的计算分为两步。首先利用第 k 轮和 $k+1$ 轮链路时延测量过程中的 $t_{(q,tx)}^k, t_{(p,rx)}^k$ 和 $t_{(q,tx)}^{k+1}, t_{(p,rx)}^{k+1}$ 时间戳计算相邻2个节点时钟的频率比 $\lambda_{<p,q>}$ ，如式(4)所示；然后从GM时钟累乘至同步节点 p ，如式(5)所示。

$$\lambda_{<p,q>}^{k+1} = \frac{t_{(p,rx)}^{k+1} - t_{(p,rx)}^k}{t_{(q,tx)}^{k+1} - t_{(q,tx)}^k} \quad (4)$$

$$\lambda_{<GM,p>} = \prod_{i \in P} \lambda_{<i,i+1>}^k \quad (5)$$

其中， P 表示GM到节点 p 之间转发同步报文路径上节点的集合。

最后同步节点根据式(6)计算得到修正时间 $t_{(p,c)}$ ，将本地时钟修改成该时间，完成与GM时钟的同步。

$$t_{(p,c)} = t_{(GM,b)} + R + L \quad (6)$$

接下来以图1所示系统为例，说明协议实际运行时实现802.1AS时间同步原理的具体流程，示例拓扑由 $N_0 \sim N_4$ 这5个节点组成，其中 N_0 为当前GM时钟。

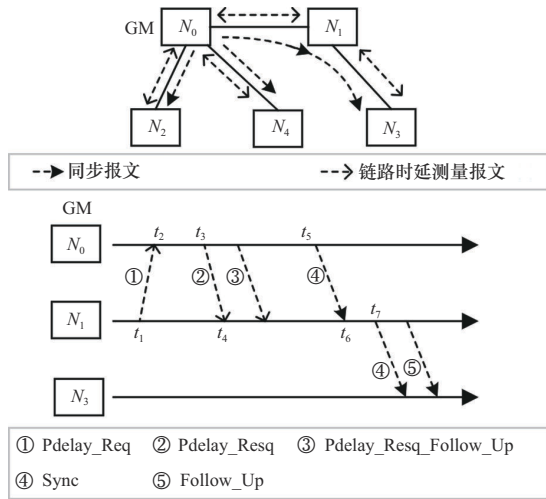


图1 802.1AS时间同步流程

链路时延测量包含3类报文：Pdelay_Req、Pdelay_Resp和Pdelay_Resp_Follow_Up，如图1中的①~③所示。测量发起者 N_1 首先发送报文①，本地记录发送时间戳 t_1 ，响应者 N_0 记录报文①的接收时间戳 t_2 后携带在报文②中发送至 N_1 并构造报文③携带报文②的发送时间戳 t_3 至 N_1 ， N_1 记录报文②的接收时间戳 t_4 。 N_1 在获取4个时间戳之后，按照式(1)计算 $\lambda_{<N_0,N_1>} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$ 。同时，

链路时延测量过程中 N_1 也会使用相邻2轮测量的响应报文相关时间戳计算本地时钟与GM时钟的频率比 $\lambda_{<N_0,N_1>}$ 。

同步过程包含两类报文：Sync和Follow_Up，如图1中的④和⑤所示。同步过程可配置为一步或者两步模式：在一步模式下，Sync报文发送时间戳 t_5 会在报文从端口发出时将其记录在报文中，如图1中 N_0 和 N_1 之间；在两步模式下，时间戳 t_5 由Follow_Up报文携带至下游设备，如图1中 N_1 和 N_3 之间。

N_1 接收 N_0 发送的Sync报文后，需要完成校正与转发操作。校正操作是根据报文④中携带的精确时间戳 t_5 计算修正时间 $t_{(N_1,c)} = t_5 + l_{<N_0,N_1>} \lambda_{<N_0,N_1>}$ ，这里由于 N_1 与GM相邻，所以设有驻留时延。转发操作是将报文④转发至下游设备 N_3 ，需要对报文做两处修改。一处是驻留时延字段累加报文在 N_1 产生的时延 $r_{N_1} = (t_7 - t_6) \lambda_{<N_0,N_1>}$ ；另一处是将GM频率比字段修改为 $\lambda_{<N_0,N_1>}$ 。

N_3 接收 N_1 修改并转发的报文④和⑤后，因为不存在下游设备，所以不需要进行转发操作，仅需要代入式(6)计算本地时钟的修正时间，如式(7)所示。

$$t_{(N_3,c)} = t_5 + r_{N_1} + l_{<N_0,N_1>} \lambda_{<N_0,N_1>} + l_{<N_1,N_3>} \lambda_{<N_0,N_1>} \lambda_{<N_1,N_3>} \quad (7)$$

1.1.2 精度定义

根据802.1AS标准和时间同步相关研究的描述，本文对时钟间的偏差和同步精度作如下定义。

定义1 2个时钟的偏差是同一时刻在2个时钟示数的差值。

$$\delta_{<p,q>} = |C_p(t_0) - C_q(t_0)| \quad (8)$$

定义2 同步精度是在观测时间段 T 内系统 S 中任意2个时钟的偏差的最大值，如式(9)所示。

$$\Delta = \text{MAX}(\delta_{<p,q>}), \forall p, q \in S, \forall t_0 \in T \quad (9)$$

同步精度是一个系统参数，需要定义目标系统的组成和观测时间范围后才能进行测量。需要注意的是，在描述单个时钟同步性能的情况下，考虑到TSN时间同步是主从式的，可以将GM主时钟和单个从时钟视为由2个时钟共同组成的同步子系统。为避免产生歧义，本文在该情况下使用“某时钟的同步精度”，与“系统同步精度”加以区分。在第

4节的实验验证中,对时钟偏差、单个时钟同步精度和系统同步精度均有测量。

1.2 相关工作

时间同步精度是衡量同步协议性能的核心指标,所以准确测量同步精度是实现同步性能优化、状态监测、故障诊断等网络管理控制功能的基础。目前相关工作中对同步精度的测量方法可以分为两类,基于专用测试仪器和基于同步测量协议。

1) 基于专用测试仪器的测量方法要求被测设备能够周期性地输出统一规格的时钟脉冲,如秒脉冲或毫秒脉冲,然后通过示波器、逻辑测试仪或时间同步精度专用测试仪等设备分析脉冲之间的偏差作为精度计算的依据。Najafi等^[15]使用示波器采集秒脉冲测量时钟同步精度并估算时钟频率偏差,建立时钟实际频率与周围环境温度的模型,软件根据频差周期性进行补偿,提升时钟运行的稳定性。文献^[16]使用软硬件协同的方式实现PTP时间同步系统,在实验环节使用专用测试仪器对实现系统的同步精度进行测量。文献^[17]在802.1AS时间同步网络中实现最早截止时间优先调度算法,在评估同步精度对调度算法性能的实验中使用示波器测量系统同步精度。

在引言部分已简要分析此类方法存在测量系统构成复杂、测量规模受仪器采集接口数量限制、测量数据无法支持实时网络管理控制的问题,在此结合图2所示的TSN工控应用场景对该问题的现象和原因做进一步分析。图中的线性拓扑是工控网络的典型拓扑,可编程逻辑控制器(PLC, programmable logic controller)作为端设备与交换机连接,交换机之间互相连接组成主干网络,所有设备之间进行时间同步。基于专用测试仪器的测量方法存在以下问题。① 如果采用具有4个采集通道的示波器,那么只能测量局部的同步精度,并连接在不同示波器的设备之间,如交换机2和交换机3是无法通过示波器测量的。② 定制同步精度专用测试设备或者使用逻辑分析仪器可以解决商用示波器脉冲采集通道数量较少和存储深度小的问题,但是会大幅增加构建测试系统的时间成本和经济成本。例如, Tektronix TLA6400系列(如TLA6401)逻辑分析仪包含34个采样通道,售价在5000美元起; Spirent TestCenter测试仪支持GPS外部时钟源精度测量,所以可以多台测试仪级联测试,单台售价在2万美元左右。③ 专用测试仪器测量的测量数据

难以直接支持实时网络管理控制功能的实现。测量数据会存储在仪器内部,而管理控制功能通常在网络控制器上实现,且物理上的隔离性为数据的有效利用带来困难。

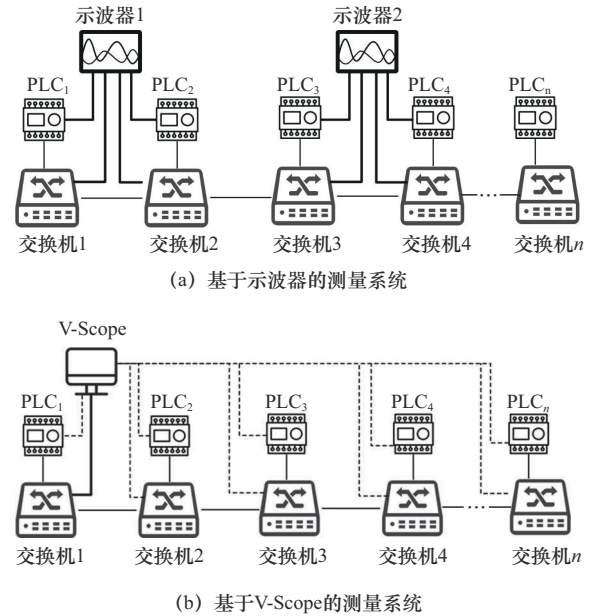


图2 面向工控应用场景的TSN同步精度测量方法对比

2) 基于测量协议的测量方法是通过在网络中传输专用的探针报文收集被测设备的时间信息后,计算设备间同步精度。文献^[18]提出了纳秒级数据中心时间协议(DTP, datacenter time protocol)修改了标准物理层的硬件设计,在该层新增了用于时间同步和精度测量的逻辑子层DTP-enabled-PHY。文献^[19]提出了一种基于IEEE 1588的时间同步精度测量协议ReverseSync,该协议利用PTP多域的特性,在网络中配置专用的同步精度测量域,使从时钟主动向主时钟发送同步报文,反向测量网络中从时钟与主时钟的偏差,进而计算全网同步精度。Geng等^[20]在数据中心网络中使用自定义的探针报文收集所有设备时钟的时间信息作为样本数据,然后基于支持向量机对样本数据进行筛选,精确估计各个时钟之间的偏差,最后根据有效数据周期性地对各个时钟进行校正。

此类方法要求被测设备具备识别测量协议报文、使用本地时钟向报文注入时间戳信息等定制化功能,甚至需要专用硬件模块的支持,适用于由单一厂商定制设备组成的网络,难以应用于由通用的、符合802.1AS标准的TSN设备组成的同步网络。

时间同步精度测量方法对比如表2所示,本文

表2 时间同步精度测量方法对比

测量方法	测量数据通道限制	对被测设备要求	测量数据传输接口/介质	设备成本
示波器	4通道	时间脉冲信号输出	RF 射频同轴连接器/射频线	低
逻辑分析仪	8~136通道	时间脉冲信号输出	探针引脚/探针连接线、飞线等	高
同步精度专用测试仪	支持级联可扩展	时间脉冲信号输出	RF 射频同轴连接器/射频线	高
DTP-enabled-PHY	—	DTP使能的PHY层	网络接口/网线	低
ReverseSync	—	支持PTP多同步域	网络接口/网线	低
Huygens	—	为特定流量注入硬件时间戳	网络接口/网线	低
VS方法	—	标准TSN同步设备	网络接口/网线	低

方法不依赖专用的设备和特殊的信号接口，也不需要被测设备支持额外的精度测量协议，以轻量级、易部署的软件形态虚拟化示波器的功能。本文方法运行在 TSN 的控制平面，通过 TSN 支持的网管协议获取同步状态数据即可完成精度的测量。

2 VS方法设计

2.1 总体设计

基于 VS 方法构建的测量系统整体架构包含 4 个部分：控制器、V-Scope、同步管理对象和 TSN 时间同步设备，如图 3 所示。

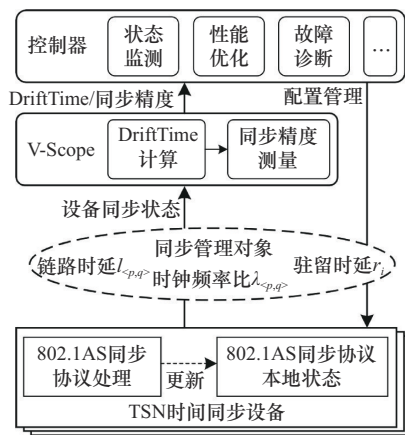


图3 基于VS方法的测量系统整体架构

控制器是符合 802.1Qcc 标准的逻辑集中式网络控制器，对 TSN 执行配置管理功能。控制器可从 V-Scope 获取时间同步精度和 DriftTime 参数实现具有状态监测、性能优化、故障诊断等网络管理控制功能，在第 2.3 节会对应用方法进行详细讨论。

V-Scope 是 VS 方法虚拟化示波器功能的核心组件，实现访问同步设备的本地同步状态 DriftTime 计算和同步精度测量的功能。DriftTime 参数定义、计算方法及性质详见第 2.2 节。

同步管理对象是描述数据平面设备中与时间同

步协议相关的本地状态数据，在 802.1AS 标准第 11 章中进行详细定义（包含 17 类对象和 162 个状态属性），以支持控制器能够通过 SNMP（simple network management protocol）和 NETCONF（network configuration）等南向管理协议配置设备同步参数、获取同步设备协议状态。VS 方法需要获取的状态属性包括链路时延、时钟频率比以及每个设备的驻留时延。

数据平面设备是由符合 802.1AS 标准的 TSN 时间同步设备组成，具备同步和链路时延测量等协议处理功能。此外，该设备需要能够处理控制平面对 802.1AS 同步协议本地状态数据的读请求。本文方法有以下优势。

1) 可扩展性。基于 VS 方法的测量系统不需要接入示波器等专用设备，也不需要数据平面设备设置额外的脉冲输出接口。同步精度测量是根据 802.1AS 标准定义的协议状态数据计算 DriftTime 参数，而协议状态访问则是通过控制器已有的南向管理协议，因此被测网络规模不会因为专业测量仪器在采集通道数量、存储深度、连线距离等限制。

2) 轻量级。V-Scope 逻辑位于网络的控制平面，可以复用 TSN 集中控制器的南向管理协议。根据 802.1AS 标准定义，测量使用的状态数据类型分别是：链路时延和驻留时延（4 B 整型）、时钟频率比（8 B 浮点型）以及精确时间戳（10 B 整型）。如果测量周期设置为 125 ms，则每个设备带宽的占用约为 1.4 kbit/s，这对动辄千兆万兆带宽的以太网的资源占用是十分微小的。

3) 实时性。基于 VS 方法的测量系统能够形成同步状态数据采集、同步精度测量到网络管理控制的自动化控制闭环，消除管理员手动操作环节，控制器、V-Scope 以及数据平面设备之间均通过网络进行通信，既降低了数据交互的复杂度，也提升了网络管理效率。

2.2 DriftTime 参数

2.2.1 参数定义

本文定义易于计算和获取的 DriftTime 参数用于测量同步精度。

定义3 DriftTime (在公式中缩写为 DT) 参数是非主时钟节点 p 在每轮同步过程中计算得到的修正时间 $t_{(p,c)}$ 与实际修改本地时钟的更新时间 $t_{(p,u)}$ 的差值, 如式(10)所示。

$$DT = t_{(p,u)} - t_{(p,c)} \quad (10)$$

DriftTime 语义示意如图4所示, 节点 p 在收到同步报文并按照式(6)计算得到 $t_{(p,c)}$ 后, 需要将本地时钟修改为该时刻。在实际修改时, 将 p 的本地时钟 C_p 示数记作更新时间 $t_{(p,u)}$ 。

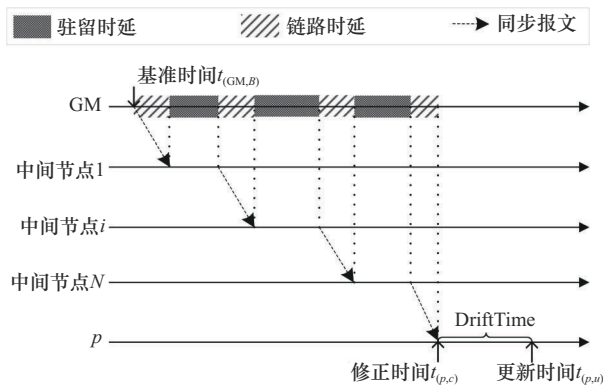


图4 DriftTime 语义示意

2.2.2 参数性质

DriftTime 参数具有2个关键的属性: 时钟频率相关性和 GM 时钟偏差有界性。时钟频率相关性使其能够真实反映时钟的物理属性 (如频率、漂移率等), GM 时钟偏差有界性则确保其能够用于精确测量同步精度。

定理1 时钟频率相关性。网络中任意节点 p 在一轮同步中计算得到的 DriftTime 参数与本地时钟 C_p 和主时钟 C_{GM} 的频率比线性相关, 如式(11)所示。

$$DT = C_{GM}(T) \left(1 - \frac{f_{GM}}{f_p} \right) + \varepsilon \quad (11)$$

其中, T 为周期性同步的间隔时长, ε 为在获取计算所依赖的时间数据时产生的误差 (如链路时延非对称性、时间戳读取不准确等)。

证明 详见附录1。

TSN 时间同步中的所有非主设备依据 GM 的时间校正本地时钟, 因此在 TSN 中使用的所有时刻

和时段应参考 C_{GM} 时钟刻度。在其他时钟上读取的时刻或时段应换算成 C_{GM} 读数。每轮同步产生的偏差都是本地时钟的实际运行频率与 GM 不一致导致的。两次同步间隔得越久, 时钟之间的 DriftTime 参数越大, 这与时钟偏差与同步间隔的关系具有一致性。

定理2 GM 时钟偏差有界性。网络中任意节点 p 在第 k 轮同步结束时, 计算得到的 DriftTime 参数与 $\delta_{<GM,p>}^k$ 的差值是常量, 如式(12)所示。

$$DT_p^k - \delta_{<GM,p>}^k = \varepsilon_p^k \quad (12)$$

其中, $\delta_{<GM,p>}^k$ 表示此时刻该节点本地时钟和 GM 时钟的偏差,

证明 详见附录1。

DriftTime 参数是能够在协议运行过程中实时计算并反映时钟之间实际偏差的数据。GM 时钟偏差有界性意味着 DriftTime 参数可以在一定误差范围内视作每个节点时钟与 GM 时钟的偏差值, 进而能够计算得到节点和系统的同步精度。

2.3 VS 方法应用及举例

VS 方法使控制器能够直接测量网络同步精度, 从而支持控制器实现性能优化、状态监测、故障诊断等网络管理控制功能。本节以同步性能优化为例, 分析 VS 方法在网络控制平面应用中的可行性和有效性。

在系统运行过程中, 不同设备的本地时钟产生偏差的本质原因是驱动时钟的晶振器件的实际运行频率不一致。晶振器件的振荡频率极易受温度、电压、材料老化等因素的影响, 使其无法完全按照标称频率运行^[21-22]。据此, 本文提出了一种采用频差补偿 (FOC, frequency offset compensation) 的时间同步精度优化算法。该算法使用 DriftTime 参数估算设备本地时钟与主时钟之间的频率偏差。由于物理器件设计的封闭性, 无法直接修改物理器件的运行频率, 所以该算法通过修改本地时钟拍长的方式间接地修改时钟运行频率。

算法1 频差补偿算法 FOC

输入 M 个节点组成的 TSN 时间同步网络内在第 k 轮同步的 DriftTime 参数集合 $\{DT_0^k, DT_1^k, \dots, DT_M^k\}$

输出 每个设备本地时钟的拍长更新值集合 $\{\omega'_0, \omega'_1, \dots, \omega'_M\}$

1) 定义并初始化用于计算频差比的参数集合 $\{DT_0, DT_1, \dots, DT_M\}$, 每个元素为0

- 2) 循环
- 3) for $i=1:1:k$
- 4) for $m=1:1:M$
- 5) $DT_m = DT_m + DT_m^i$
- 6) end for
- 7) end for
- 8) 循环
- 9) for $m=1:1:M$
- 10) 根据式(13)和式(14)计算校正拍长 ω_m'
- 11) end for

控制器通过 V-Scope 获取每个节点 p 在每轮同步过程中的 DriftTime 参数。根据 DriftTime 参数的时钟频率相关性, 可按式(13)估算设备本地时钟与当前主时钟的频率比。

$$\frac{f_{GM}}{f_p} \approx 1 - \frac{DT}{T} \quad (13)$$

$$\omega' = \left(\frac{f_{GM}}{f_p} \right) \omega \quad (14)$$

FOC 算法采用修改时钟拍长的方式间接地调整时钟运行频率。假设此时节点 p 本地时钟的拍长为 ω , 即本地时钟在晶振每振荡产生一次脉冲时就自增 ω , 根据频率差计算出新的拍长 ω' 。最后, 网络中的所有时钟运行频率会向主时钟运行频率收敛, 从而提高系统的同步精度。

3 基于 OpenTSN 的原型系统实现

为验证 VS 方法在真实 TSN 系统测量的有效性和可行性, 本文基于 OpenTSN 开源项目^[23-24]实现原型系统。OpenTSN 是包含 TSN 交换机、接口控制器、相关配套工具及管理软件的开源设计项目, 支持用户根据使用需求快速定制搭建 TSN 原型验证系统, 实现对时间同步、流量调度等关键技术的验证。

3.1 同步状态数据获取

由于 OpenTSN 设备具备状态上报的功能, 因此实现时不需要在其数据平面的硬件逻辑新增开发, 只需要使用控制器配置硬件打开状态上报的功能和上报状态的内容即可, 状态数据上报周期可设置为与同步周期相同, 以便于及时获取状态更新数据。

在状态上报的功能被打开后, 控制器便能够从各个设备的 802.1AS 处理逻辑中收集同步报文携带的时间戳、链路时延信息、端口的链路时延以及修改本地时钟的校正时刻, 作为 V-Scope 中 DriftTime

参数计算逻辑的输入。控制器按照式(10)计算 DriftTime 参数并按照设备标识, 以队列的组织形式进行存储。控制器应用可根据设备标识获取需要设备的 DriftTime 参数和系统的同步精度。

3.2 处理流程

原型系统采用软硬件协同的方式实现, 硬件部分复用 OpenTSN 数据平面设备的报文转发、时间同步等逻辑, 软件部分使用 C 语言实现 V-Scope 程序, 程序调用控制器提供的用于实现报文接收、发送等基础的功能编程接口。

软硬件整体处理流程如图 5 所示。首先, 配置数据平面硬件设备寄存器使能同步状态上报, 设备在本地定时器超时后根据同步协议的本地状态构造报文, 并通过设备的管理网口发送至软件。报文格式遵循由 OpenTSN 定义的时间敏感管理协议 (TSMP, time sensitive management protocol)。由 V-Scope 软件接收并解析报文, 获取计算 DriftTime 的原始数据, 完成计算后更新对应设备存储 DriftTime 队列, 并计算同步精度。控制器应用 FOC 优化器获取 V-Scope 的最新数据, 估计设备与主时钟设备的频差估计值, 进而计算该设备拍长的校正值, 构造相应的配置更新报文下发至该设备。配置报文同样采用 TSMP 报文, 修改拍长寄存器的值, 完成对各个设备本地时钟的频率补偿。

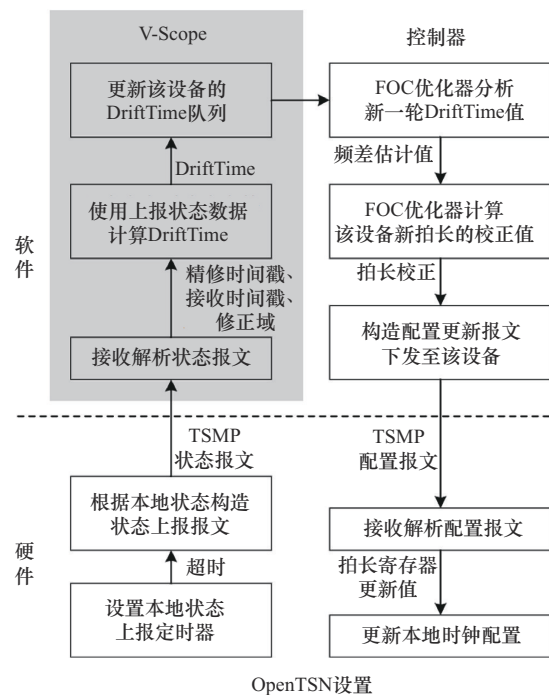


图 5 软硬件整体处理流程

4 实验验证

本文首先设置实验对比分析 VS 方法和基于专用测量仪器的方法所测得的时间同步精度, 结果表明, 二者仅存在纳秒级的误差 (4.2 节)。另外, 为证明 VS 方法在支持网络管理控制功能的优越性, 本文对第 2.3 节应用举例中提出的 FOC 优化算法进行测试, 对比使用 FOC 优化算法前后的系统同步性能 (第 4.3 节)。实验中所使用和分析的数据均是在系统长时间运行过程中采集。

4.1 实验系统

实验环境共由 25 个参与同步节点 $N_0 \sim N_{24}$ 组成, 实验系统拓扑示意和实物如图 6 所示, 其中 $N_0 \sim N_7$ 是 TSN 交换机, $N_8 \sim N_{24}$ 是 TSN 端系统。节点 N_6 配置为 GM 时钟, 同步节点中与其相隔的最大跳数为 4。V-Scope 运行在 OpenTSN 控制器所在的 Linux 端系统。硬件设备的详细配置如表 3 所示, 包含 FPGA 板卡型号, PHY 芯片型号以及板载晶振的标称漂移率。标称漂移率是指在正常运行过程中, 晶振实际频率与标称频率 125 MHz 相差的最大百分比。

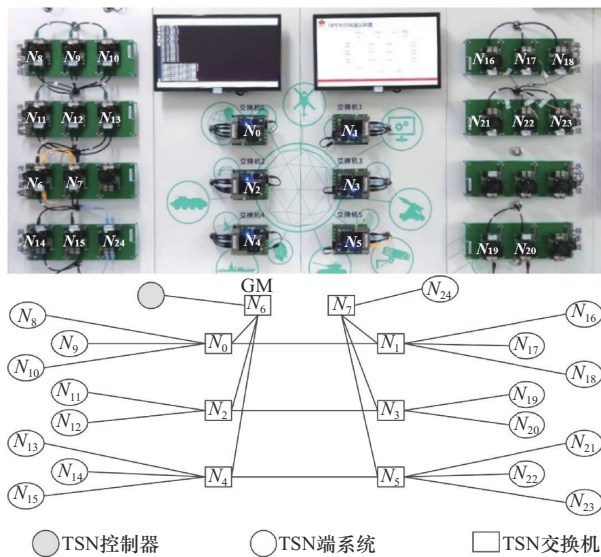


图 6 实验系统拓扑示意和实物

表 3 硬件设备信息

节点标识	FPGA 板卡型号	PHY 芯片型号	板载晶振的标称漂移率/ppm
$N_0 \sim N_7$	Xilinx-XC7K325T	BCM5464S	50
$N_8 \sim N_{24}$	Arria10-10AX048H	VSC8504	50

4.2 同步精度测量对比

4.2.1 实验设置

本实验使用专业的时间同步精度测试仪 XZ-T3000 采集网络设备输出的时钟脉冲, 测量各时钟与主时钟之间的同步精度, 并与 VS 方法的测量结果进行对比。

测试仪使用 SMA 接口采集每个设备的时钟脉冲输出, 通过分析每个为脉冲信号记录的测试时间戳, 计算各时钟与主时钟之间的频率偏差和同步精度。测试仪共有 10 个 SMA 接口, 其中, 1 个接口与主时钟 GM 设备连接, 其余 9 个接口可以与任意设备连接。考虑到同步报文传输经过的跳数会影响误差 ϵ , 因此分别在 1、2、3、4 跳数的设备中选取部分节点接入测试仪进行测量。

测量结果的分析对比是基于样本数据的整体分布, 而不是按照时间序列对比单个样本数据。首先, 测试仪采集的时钟脉冲输出是标准的 1PPS 信号, 即每秒产生一个信号输出, 而 V-Scope 的精度测量周期配置为 128 ms。其次, 测试仪需要手动开关, 无法精确与 V-Scope 对准开始测量时刻。开始时刻不对齐和测量周期不一致使得 2 种方法测量结果的数据不能够一一对应到同一时刻的系统状态, 因此本文采用从数据整体分布判断相似性的方式分析测量结果。

JS 散度 (Jensen-Shannon divergence) 是衡量两组数据概率分布相似性的度量指标^[25], 取值区间为 $[0, 1]$, 0 表示两组数据分布完全相同, 1 表示两组数据分布完全不同。在本实验中, 本文计算同一观测时间段内 2 种测量方法测得时钟 GM 偏差数据分布之间的 JS 散度, 以此判断二者是否具有相似性。

4.2.2 结果分析

图 7(a) 是 2 种测量方法对 9 个设备测量得到的 GM 时钟偏差的平均值, 二者存在误差, 主要原因如下。

1) 时间戳记录的不准确性。V-Scope 的测量本质上是基于时间戳的, 当硬件设备为某事件执行时间戳获取逻辑时, 所记录时刻会与事件真实发生时刻存在 1~2 拍 (实验系统的设备时钟每拍长度为 8 ns) 的误差。这部分误差会体现在报文每跳驻留时延和链路时延的计算中, 最终影响

测量结果。测试仪测量是基于脉冲信号的，被测设备通过专用通道与测试仪连接并传输信号数据。

2) 链路时延测量的非对称性。采用 802.1AS 标准中定义的测量方法是基于上行和下行链路时延相等的假设，但在实际系统中，由于接收和发送报文的处理流程不同、PHY 芯片处理时延抖动等原因，上下行链路时延是不相等的，所以测得的链路时延与实际链路时延存在误差。

3) 驻留时延计算的不完整性。实验系统的设备是在硬件 MAC 层记录报文的接收时间戳，但是在时间戳记录完成后到报文解析、计算校正时间的处理逻辑前仍存在一段时延，这部分时延未被记录，所以 V-Scope 测量过程使用的驻留时延比实际驻留时延小。

从总体结果来看，随着节点距离 GM 时钟的跳数不断增加，2 种测量方法所测结果的差值也有增大的趋势，由图 7(a)可以看出，二者测量结果相差最大的是跳数为 4 的 N_{24} ，差值为 4.5 ns。这是由于跳数增多使得测量过程中时间戳记录、链路时延测量、驻留时延计算引入的误差不断累积，与式(11)描述的性质一致，符合预期。

GM 时钟偏差的标准差如图 7(b)所示。标准差都在 10.1 ns 以内，说明数据分布离散程度低，未出现较大的离群异常值，表明时钟在运行过程中表现稳定，所以使用样本数据的平均值作为描述测量结果特征的数据是可信的。因此，图 7(a)中平均值对比所得结论可以作为证明 VS 方法与测试仪测量结果相似性的依据。

图 7(c)是各个时钟的 GM 偏差的最大值对比，即每个时钟在观测时间段内的同步精度。2 种方法测得的精度的差值在 5~14 ns。在 TSN 流量规划调度中对设备间的时间同步精度要求是微秒级或亚微秒级^[6]，所以这个数量级的测量误差在关键流量调度的应用中是可以容忍的。

为了量化测量结果的相似性，本文计算了 9 个非 GM 节点在 2 种测量方法下 GM 偏差分布的 JS 散度，如表 4 所示。JS 散度最小值是 N_2 ，为 0.052 8；最大值是 N_{24} ，为 0.068 3。JS 散度值接近于 0，表明 2 个样本分布的位置和形状是相似的，即 VS 方法测量的同步精度与测试仪所测结果具有一致性。

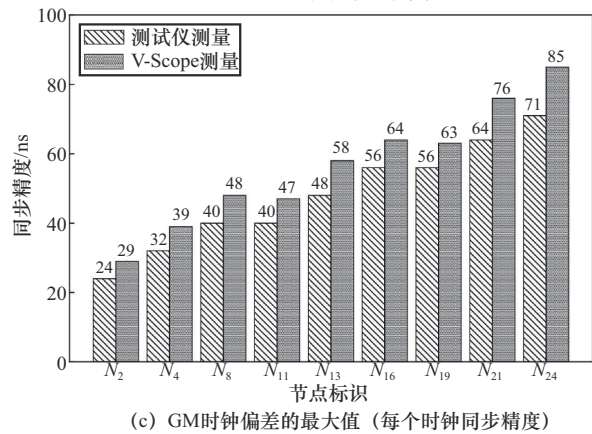
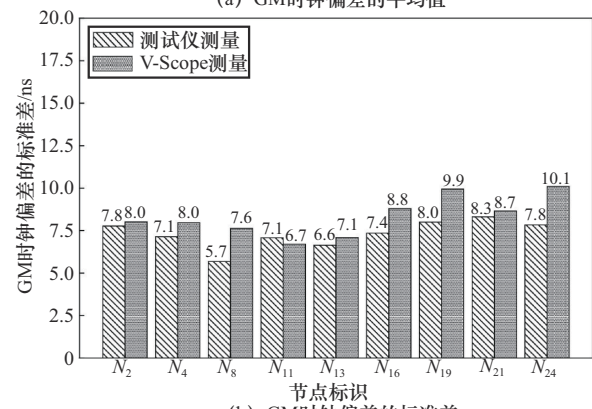
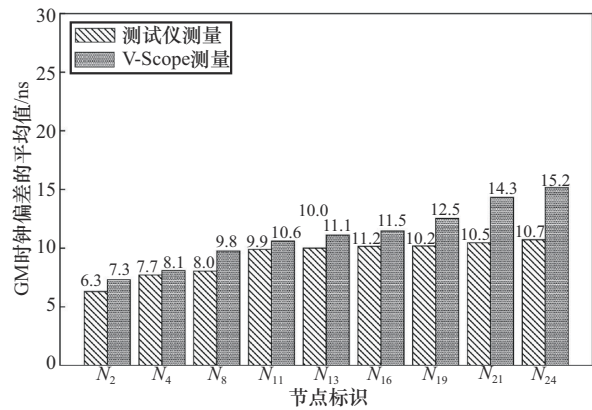


图 7 测试仪和 V-Scope 的测量结果统计值对比

表 4 V-Scope 与测试仪测量结果的 JS 散度

节点标识	JS 散度
N_2	0.052 798 67
N_4	0.058 952 57
N_8	0.061 513 38
N_{11}	0.063 710 23
N_{13}	0.064 451 98
N_{16}	0.064 604 01
N_{19}	0.065 731 81
N_{21}	0.067 811 63
N_{24}	0.068 305 36

4.3 FOC 同步优化

4.3.1 实验设置

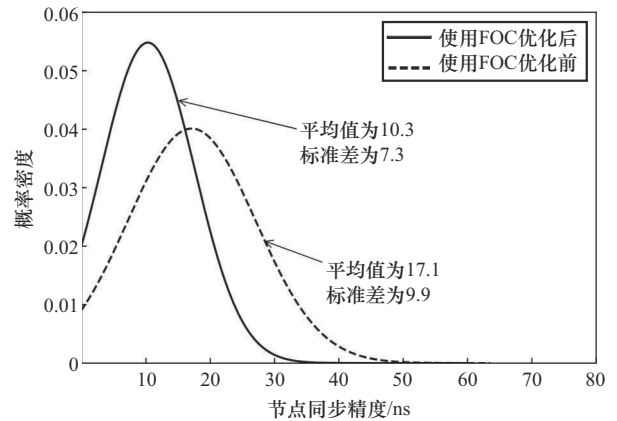
第2.3节以同步精度优化为例论证了VS方法在支持网络管理控制功能的可行性,提出频率补偿的优化算法,本实验对FOC优化算法的优化效果进行测试。实验同样是在图6所示的原型系统中进行。影响同步精度的主要因素有同步周期长度、过程误差和时钟频率比。时钟频率比会被算法补偿修改,过程误差的影响可以通过不同节点与GM之间的跳数观察。因此,为全面测试FOC优化算法的性能,实验设置不同的同步周期长度(32 ms、64 ms、128 ms、256 ms、512 ms),使用VS方法测量各个设备的GM偏差和同步精度,对比使用FOC优化算法前后的变化。

4.3.2 结果分析

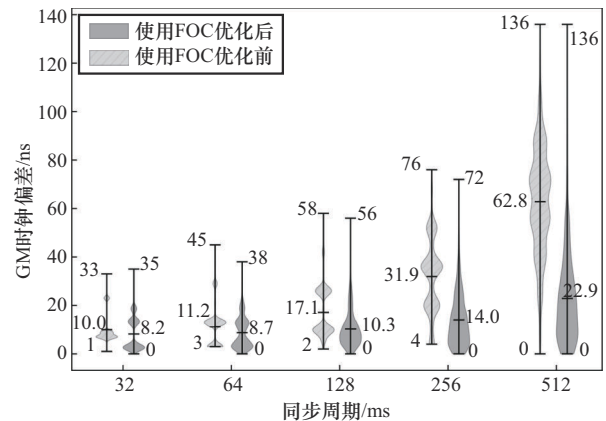
首先,以 N_{19} 为例分析FOC优化算法对单个节点的影响。图8(a)呈现的是节点 N_{19} 在同步周期长度设置为128 ms时,同步精度在使用FOC优化算法前后概率密度分布的变化,优化后的同步精度标准差减小,精度更集中分布在30 ns以内;平均值从17.1 ns降低至10.3 ns,降低了39.8%。图8(b)对比了节点 N_{19} 不同同步周期长度设置下使用FOC优化前后的GM时钟偏差对比情况,小提琴图的上中下标注的横线分别为最大值(同步精度)、平均值和最小值。可以看出,使用FOC优化后的GM偏差的平均值下降,分布更集中,意味着FOC优化算法能使时钟运行得更加稳定。

然后,分析FOC优化算法对系统的影响。图9是不同同步周期长度设置下系统同步精度优化前后对比。系统同步精度是取所有节点在运行过程中与主时钟同步精度的最大值。从图9中可以观察到,当同步周期小于128 ms时,对系统整体优化效果较小,这是由于同步周期长度较短时,频差导致的时钟偏差的基础值偏小,因此限制了频率补偿方式对精度优化的影响。当同步周期长度设置为256 ms和512 ms时,优化效果显著,系统同步精度分别减少了18.2%和32.3%。

以上结果说明,VS方法能够在控制平面上有效支持性能优化类网络管理控制功能的实现。



(a) 同步周期长度为128 ms



(b) 不同同步周期长度设置

图8 节点 N_{19} 与GM时钟偏差优化前后对比

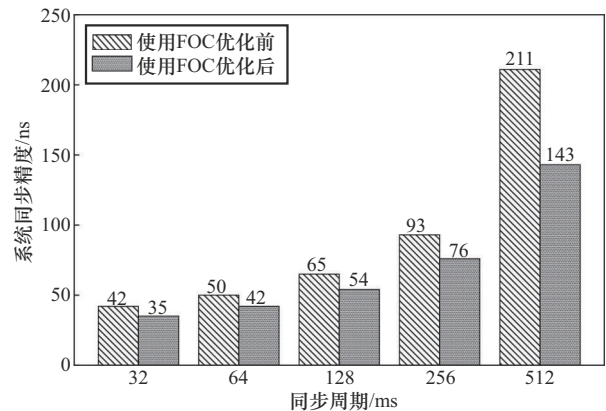


图9 系统同步精度优化前后对比

5 结束语

本文针对现有基于时钟脉冲的同步精度测量方法存在测量系统构成复杂、测量规模受仪器采集接口数量限制、测量数据无法支持实时网络管理控制等问题,提出了一种轻量级可扩展的TSN时间同步精度测量方法Virtual Scoping,实现了虚拟化示波器的功能。本文方法基于TSN集中式管控架构

定义了 DriftTime 参数。该参数具有时钟频率相关性和 GM 时钟偏差有界性，可精确测量同步精度且不依赖专用测量设备。此外，VS 方法能够支持在网络的控制平面实现性能优化、状态监测、故障诊断等网络管理控制功能，本文以性能优化为例，设计实现一种采用频率补偿的同步精度优化算法 FOC。

本文基于开源项目 OpenTSN 实现原型系统并搭建验证环境。在开发过程中，没有在其数据平面的端系统和交换设备上增加额外的处理逻辑。V-Scope 程序调用控制器提供的南向接口集中收集网络的同步状态数据，实现 DriftTime 参数计算和同步精度测量。V-Scope 工具已向 OpenTSN 开源社区提交开源。在真实的系统中对比 VS 方法和专业测试仪的测量结果，验证二者的一致性。在控制器上实现了 FOC 优化算法，显著地提高了系统的同步精度，验证了 VS 方法支持性能优化类管理控制功能的可行性。

下一步工作将继续研究基于 VS 方法实现同步状态实时监测、故障诊断回溯等 TSN 管理控制功能。

附录 1 DriftTime 性质证明

1) 时钟频率相关性

由于不同时钟的实际运行频率存在差异，对于给定的时间段 T ，参考不同时钟的读数也不同，与实际频率成正比，对于任意时钟 p, q ，有式(15)成立。

$$\frac{C_p(T)}{C_q(T)} = \frac{f_p}{f_q} \quad (15)$$

对于网络中的任意时钟 p ，在接收到 GM 发出的同步报文后，会准确记录此刻时间戳 $C_p(t_{(p,c)})$ ，并且使用式(6)原理推断此刻主时钟时间 $C_{GM}(t_{(p,c)})$ 。因此，参与 DriftTime 计算的 $t_{(p,c)}$ 和 $t_{(p,u)}$ 分别存在如式(16)和式(17)所示的等式关系，图 4 能更为直观地展示该关系。

$$t_{(p,c)} = C_{GM}(t_{(GM,b)}) + C_{GM}(t_{(p,c)} - t_{(GM,b)}) + \varepsilon \quad (16)$$

$$t_{(p,u)} = C_p(t_{(GM,b)}) + C_p(t_{(p,c)} - t_{(GM,b)}) \quad (17)$$

考虑相邻两轮校正过程，第 k 轮开始时刻的差值是上一轮同步后积累的误差，所以有式(18)成立。

$$\begin{aligned} C_{GM}(t_{(GM,b)}^k) - C_p(t_{(GM,b)}^k) = \\ C_{GM}(t_{(GM,b)}^k - t_{(p,c)}^{k-1}) - C_p(t_{(GM,b)}^k - t_{(p,c)}^{k-1}) \end{aligned} \quad (18)$$

将式(16)~式(18)代入 DriftTime 参数的定义式(10)可得

$$DT = C_{GM}(t_{(p,c)}^k - t_{(p,c)}^{k-1}) - C_p(t_{(p,c)}^k - t_{(p,c)}^{k-1}) + \varepsilon \quad (19)$$

其中， $t_{(p,c)}^k - t_{(p,c)}^{k-1}$ 表示相邻两轮同步的时间间隔，即同步周期 T ，又有式(15)的相同时间段的频率比例关系，最终可得

$$DT = C_{GM}(T) \left(1 - \frac{f_{GM}}{f_p}\right) + \varepsilon \quad (20)$$

2) GM 时钟偏差有界性

在理想状态下，第 i 轮同步后 p, q 的时钟偏差值为 0，那么在第 $i+1$ 轮同步过程中产生的偏差值 $\delta_{<p,q>}^i$ 可由式(21)计算。时钟频率在 2~3 s 的变化是可以忽略的，又因为同步周期标准建议值是 125 ms，所以本文认为在一个同步周期内时钟频率 f_p, f_q 是常量。

$$\delta_{<p,q>}^i = \int_{t_0^i}^{t_0^i + T} \left(1 - \frac{f_p}{f_q}\right) dt \quad (21)$$

在式(21)中考虑节点 q 为 GM 的情况，与式(20)作差，以 GM 时钟作为全局时间段的参考时钟，则有

$$\begin{aligned} DT_p^i - \delta_{<GM,p>}^i &= C_{GM}(T) \left(1 - \frac{f_{GM}}{f_p}\right) + \varepsilon - \\ &\int_{t_0^i}^{t_0^i + T} \left(1 - \frac{f_p}{f_q}\right) dt = C_{GM}(T) \left(1 - \frac{f_{GM}}{f_p}\right) + \\ &\varepsilon - \left(1 - \frac{f_{GM}}{f_p}\right) (t_0^i + T - t_0^i) = \varepsilon \end{aligned} \quad (22)$$

证毕。

参考文献:

- [1] THI M T, SAID S B H, BOC M. SDN-based management solution for time synchronization in TSN networks[C]//Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Piscataway: IEEE Press, 2020: 361-368.
- [2] YAN J L, QUAN W, JIANG X Y, et al. Injection time planning: making CQF practical in time-sensitive networking[C]//Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 616-625.
- [3] NASRALLAH A, THYAGATURU A S, ALHARBI Z, et al. Ultra-low latency (ULL) networks: the IEEE TSN and IETF DetNet standards and related 5G ULL research[J]. IEEE Communications Surveys & Tutorials, 2019, 21(1): 88-145.
- [4] IEEE. IEEE standard for local and metropolitan area networks-timing and synchronization for time-sensitive applications: IEEE Std 802.1AS-2020[S]. New York: IEEE, 2020.
- [5] IEEE. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems IEEE Std 1588[S]. New York: IEEE, 2019.
- [6] OLIVER R S, CRACIUNAS S S, STEINER W. IEEE 802.1Qbv gate control list synthesis using array theory encoding[C]//Proceedings of the 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). Piscataway: IEEE Press, 2018: 13-24.
- [7] NAGRA A S, ALLAHI I, PASHA M A, et al. Design and FPGA based implementation of IEEE 1588 precision time protocol for synchronisation in distributed IoT applications[J]. Australian Journal of Electrical and Electronics Engineering, 2022, 19(1): 31-39.
- [8] 李晓辉, 王先文, 樊韬, 等. 5G-TSN 系统下的高精度时间同步[J]. 系统工程与电子技术, 2023, 45(2): 559-565.
LI X H, WANG X W, FAN T, et al. High-precision time synchronization of 5G-TSN system[J]. Systems Engineering and Electronics, 2023, 45(2): 559-565.

- [9] LAMPORT L, MELLIAR-SMITH P M. Byzantine clock synchronization[C]//Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing-PODC' 84. New York: ACM Press, 1984: 68-74.
- [10] 国家质量监督检验检疫总局, 中国国家标准化管理委员会. 数字存储示波器通用规范: GB/T 15289—2013[S]. 北京: 中国标准出版社, 2014. General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China, Standardization Administration of the People's Republic of China. General specification for digital storage oscilloscope: GB/T 15289—2013[S]. Beijing: Standards Press of China, 2014.
- [11] 梁志国, 何昭, 冯秀娟, 等. IEEE 1057 数字波形记录仪标准的发展述评[J]. 计量学报, 2023, 44(7): 1113-1120. LIANG Z G, HE Z, FENG X J, et al. The IEEE 1057 digital waveform recorder standard development review[J]. Acta Metrologica Sinica, 2023, 44(7): 1113-1120.
- [12] MESTRES A, RODRIGUZE-NATALA, CARNER J, et al. Knowledge-defined networking[J]. ACM SIGCOMM Computer Communication Review, 2017, 47(3): 2-10.
- [13] IEEE. IEEE standard for local and metropolitan area networks--bridges and bridged networks--amendment 31: stream reservation protocol (SRP) enhancements and performance improvements[S]. New York: IEEE, 2018.
- [14] KIM H J, LEE U, KIM M, et al. Time-synchronization method for CAN-Ethernet networks with gateways[J]. Applied Sciences, 2020, 10(24): 8873.
- [15] NAJAFI A, WEI M. Graham: synchronizing clocks by leveraging local clock properties[C]//19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). Berkeley: USENIX Association, 2022: 453-466.
- [16] DONG M, QIU Z L, PAN W T, et al. The design and implementation of IEEE 1588v2 clock synchronization system by generating hardware timestamps in MAC layer[C]//Proceedings of the 2018 International Conference on Computer, Information and Telecommunication Systems (CITS). Piscataway: IEEE Press, 2018: 1-5.
- [17] PÉREZ H, GUTIÉRREZ J J. EDF scheduling for distributed systems built upon the IEEE 802.1AS clock-a theoretical-practical comparison[J]. Journal of Systems Architecture, 2022, 132: 102742.
- [18] LEE K S, WANG H, SHRIVASTAV V, et al. Globally synchronized time via datacenter networks[C]//Proceedings of the 2016 ACM SIGCOMM Conference. New York: ACM Press, 2016: 454-467.
- [19] AVNU Alliance. 802.1AS recovered clock quality testing (Revision 1.0)[S]. 2016.
- [20] GENG Y L, LIU S Y, YIN Z, et al. Exploiting a natural network effect for scalable, fine-grained clock synchronization[C]//16th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2018: 81-94.
- [21] YUAN H Y, ZHANG K, ZHENG T, et al. Crystal oscillator frequency compensation technology of high precision clock synchronization for time-triggered Ethernet[C]//Proceedings of the 2020 China Semiconductor Technology International Conference (CSTIC). Piscataway: IEEE Press, 2020: 1-3.
- [22] GAMPER H. Clock drift estimation and compensation for asynchronous impulse response measurements[C]//Proceedings of the 2017 Hands-free Speech Communications and Microphone Arrays (HSCMA). Piscataway: IEEE Press, 2017: 186-190.
- [23] FU W W, YAN J L, QUAN W, et al. Fenglin-I: an open-source tsn chip for scenario-driven customization[J]. IEEE Transactions on Computers, 2023, 72(1): 140-153.
- [24] QUAN W, FU W W, YAN J L, et al. OpenTSN: an open-source project for time-sensitive networking system development[J]. CCF Transactions on Networking, 2020, 3(1): 51-65.
- [25] MATEOS D M, RIVEAUD L E, LAMBERTI P W. Detecting dynamical changes in time series by using the Jensen Shannon divergence[J]. Chaos: An Interdisciplinary Journal of Nonlinear Science, 2017, 27(8): 083118.

[作者简介]



孙寅涵 (1997-), 男, 江苏连云港人, 国防科技大学博士生, 主要研究方向为确定性以太网、时间敏感网络、高精度时间同步等。



孙志刚 (1973-), 男, 江苏连云港人, 博士, 国防科技大学研究员、博士生导师, 主要研究方向为软件定义网络、时间敏感网络、网络体系结构、FPGA 设计、网络安全等。



付文文 (1994-), 男, 江西南昌人, 博士, 国防科技大学助理研究员, 主要研究方向为可编程网络、时间敏感网络等。



李文超 (2000-), 男, 河南洛阳人, 国防科技大学博士生, 主要研究方向为时间敏感网络、时间同步等。



黄蓉 (1989-), 女, 湖南衡阳人, 国防科技大学硕士生, 主要研究方向为时间敏感网络等。