

时间感知整形器的时间性能研究

孟星, 胡怡红

(北京邮电大学信息与通信工程学院, 网络体系构建与融合北京市重点实验室, 北京, 100876)

摘要: 时间敏感网络 (TSN, Time Sensitive Network) 是支持工业互联网的以太网新标准, 通过引入时间感知整形器 (TAS, Time Aware Shaper) 和时间同步协议, 来精确控制不同优先级的数据帧转发, 使以太网能够支持低时延, 低抖动的确定性服务。在TAS的调度过程中, 时间同步和TAS时钟源的精度等时间性能, 会一定程度的影响最终的调度性能优劣。本文研究了时间性能对TAS调度结果的影响, 分析了时间误差的来源并建立了数学模型, 通过基于OMNeT++平台的仿真, 设计了多种不同的场景进行仿真测试和分析, 得出时延变化规律, 并提出了一个根据同步报文信息实时计算GCL的方法, 可以有效消除时延的增加。

关键词: 工业互联网; TSN; TAS; 时间同步; 仿真。

中图分类号: TN915.03

Research on Time Performance of Time Aware Shaper

MENG Xing, HU Yihong

(Beijing Key Laboratory of Network System Architecture and Convergence, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China)

Abstract: Time Sensitive Network (TSN) is a new Ethernet standard that supports the Industrial Internet. TSN uses Time Aware Shaper (TAS) and time synchronization protocols to precisely control the forwarding of data frames of different priorities, enabling Ethernet to support deterministic services with low latency and low jitter. In the TAS scheduling process, the accuracy of time synchronization and TAS clock source will affect the final scheduling performance. This paper studies the impact of time performance on TAS scheduling results, analyzes the sources of time errors, designs and tests a variety of different scenarios through simulation based on OMNeT ++. We also propose A method for calculating GCL in real time based on synchronization message information can effectively reduce the delay

Key words: IIoT; TSN ; TAS; time synchronization; simulation

0 引言

时间敏感网络^[1]是基于以太网, 提供确定性服务的一套标准, 是工业互联网的核心技术之一, 在 IT&OT 融合的未来网络中扮演着重要的角色。在传统的工业网络中, 物理过程由计算机等设备控制, 向控制器传输数据和执行命令, 由于许多过程由一组设备共同完成, 是高度时间敏感的, 并且设备分布在工厂各个地方, 为了保证时间准确性, 需要一个具有高确定性、低延迟网络来连接各个设备。目前的现场总线以及工业以太网等技术采用了不同的标准以满足这个要求。但这些网络技术由于底层实现不同, 彼此一般互不兼容, 并且很难与 IT 以太网进行互联通信, 并且传统的基于分组的网络只能将延迟降低到数十毫秒的数量级, 无法满足工业需求。在 2012 年成立的 IEEE 802.1 时间敏感网络工作组提出了一系列的

作者简介: 孟星 (1994—), 男, 汉族, 硕士研究生, 主要研究方向为时间敏感网络

通信联系人: 胡怡红 (1961—), 女, 汉族, 硕士生导师, 副教授, 主要研究方向为通信网技术. E-mail: yhhu@bupt.edu.cn

标准, 包括进行时间同步的 IEEE 802.1AS^[2], 进行流量调度的 IEEE 802.1Qbv^[3]、IEEE 802.1Qbu, 进行网络管理的 IEEE 802.1Qcc 等。通过这些标准提出的协议、功能, 可以利用以太网接口和交换设备, 达到工业网络的时延、抖动的需求, 实现 IT&OT 的融合。

在 TSN 众多标准中 IEEE 802.1AS、IEEE 802.1Qbv 对于流量的传输性能起到关键的作用。在网络环境中, 使用 IEEE 802.1AS 中的 gPTP 时间同步协议来进行时钟同步, 为网络上所有的用时节点提供统一的时间, 是保证网络实时传输的基准。交换设备使用 IEEE 802.1Qbv 中的时间感知整形器 (TAS, Time Aware Shaper) 进行流量调度, 保证关键流量的传输性能, 是低时延和低抖动的关键。然而, 在实际使用时, 时钟同步并不能保证时间的完全一致, 不同节点同步后的时间可能会存在一定的偏差, 并且各个节点的本地时钟也在时刻不停的各自偏移, 这样就可能会对最终的调度结果产生影响。本文就时间同步误差与本地时钟漂移对 TSN 交换机关键流量调度性能的影响展开了研究并进行了仿真, 展示了 TAS 调度性能在不同时间条件下的变化规律, 并针对较差的情况提出了修正方法。

1 相关研究

时间敏感网络是工业互联网的核心技术之一, 在未来的网络中扮演着重要的角色。工业 4.0 的核心是网络物理系统, 是高度时间敏感的, 使得 TSN 逐渐成为网络领域的研究热点。Nasrallah 等人提出了未来网络对超低延迟 (Ultra-Low Latency, ULL) 的需求, 并提供了有关 ULL 应用程序网络机制的标准和研究报告的最新综合调查, 涵盖了从 AVB 标准发展而来的 IEEE TSN 标准的详细介绍^[4]。他们还在延迟和丢包率的平均值和最大值方面对 TAS 进行了评估, 并与 ATS (异步整形) 做了对比, 还设计并评估了一种 TAS 自适应带宽共享 (ABS) 机制, 以提高链路利用率^[5]。L.Zhao 等人研究了 TSN 的时钟同步算法, 并讨论了对具有个别时序要求的流量类型的整形和调度策略, 然后展示了硬实时流量对软实时流量的影响, 并给出了最坏情况下的端到端延迟计算^[6]。Craciunas 等人讨论了影响 TSN 中时间敏感流量的确定性行为的问题, 并基于确定性行为功能参数的通用配置, 导出计算离线调度所需的约束, 以保证有界的抖动以及确定性端到端关键通信流的延迟^[7]。M. H. Farzaneh 等人提出了新的实验方法, 设立了多种试验场景, 包括不进行流量调度、没有进行时间同步的 TAS、进行了时间同步的 TAS 三种不同情况下的不同优先级流量以背景干扰流量传输性能, 对 TSN 时间触发的标准 IEEE 802.1Qbv 进行基准测试, 测试结果表明, TAS 机制能够防止背景流量负载和竞争性非计划帧的干扰, 可以保证高优先级流量非常低的延迟和抖动^[8]。文中还提到, 终端站的网络堆栈软件的确定性行为对于减少关键周期帧在最坏情况下的延迟和抖动方面的异常值数量起着重要作用。P. Heise 等人基于 INET 框架的基础上开发了时间敏感网络仿真框架 TSimNet^[9]。

2 时间感知整形器

在工业网络 and 传统以太网融合的工业互联网中, 包括普通以太网数据和时间敏感数据

等各种数据流量类型在同一个网络中进行传输，如何保证时间敏感数据流的传输性能成为了问题的难点。比如，工业控制流量要求在 Deadline 前完成传输，抖动尽可能的接近 0，而普通以太网设备远不能保证传输质量。为解决这个问题，TSN 工作组首先在 IEEE 802.1Q 中提出了流量优先级和缓存队列的机制，进入交换机的数据帧会通过端口上的优先级过滤器，根据报文中 VLAN 字段中的 Priority 值，将不同优先级的帧缓存到对应的队列中去。随后，TSN 工作组在 IEEE 802.1Qbv 标准化了 TAS（时间感知整形器）机制，如图 1 所示。

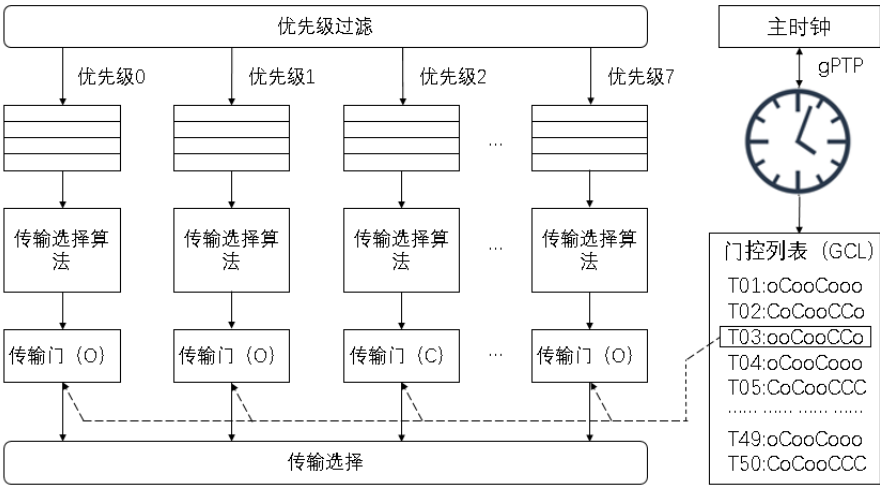


图 1 时间感知整形
Fig. 1 Time Aware Shaper

在 TAS 机制中，每个优先级缓存队列都有一个可开关传输门相关联；传输门的状态决定了缓存队列中的帧是否可以传输。通过控制门的开关，可以在以太网传输介质上划分出多段有限时间专用于各种不同优先级级别的数据帧。用户可通过手动计算或者算法计算出合适的 GCL（Gate Control List）来控制在不同的时间节点上各个门的开关动作。由于大部分时间敏感数据流（比如时间同步流量、工业控制流量等）具有严格的周期性，所以 GCL 是周期性的持续地循环执行，直到控制端下发停止指令，或者有新的 GCL 配置需要加载。每个 GCL 循环的开始时间与周期内门状态切换的时间由状态机控制，一般以一个 Tick 为最小时间单位，状态机的时间由设备时钟提供。所以，将交换机本地时钟与其他设备的时钟通过 gPTP 等时钟同步协议进行精确时钟同步，理论上就可以做到精确的调度时间片来分别发送时间敏感流量和其他流量，保证在时间敏感流量尽快的转发出去而不会被其他流量阻塞，从而保证传输质量。如图所示，在时间完全同步的理想情况下，忽略链路延迟，发送端发送的流量在到达交换机时，恰好对应的队列门打开，流量可以立刻转发到接收端，而不会发生阻塞,如图 2 所示。

然而，在实际的使用时，多个设备的时间不可能是完全一致的，而这就导致了设备时间误差较大的情况下，TAS 并不总是能够保证时间敏感数据流量的服务质量。通过实际测试和抓包分析，我们总结了两种代表性的会引起时延变化的典型情况。

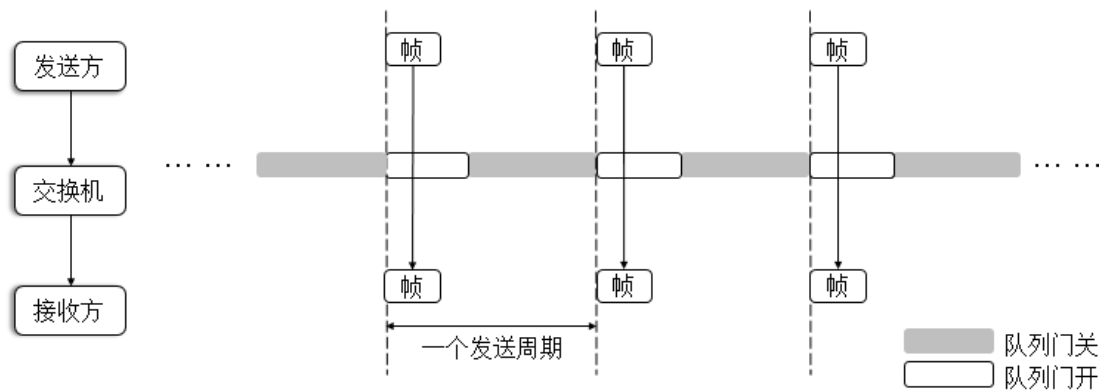


图 2 理想调度结果

Fig. 1 Ideal scheduling results

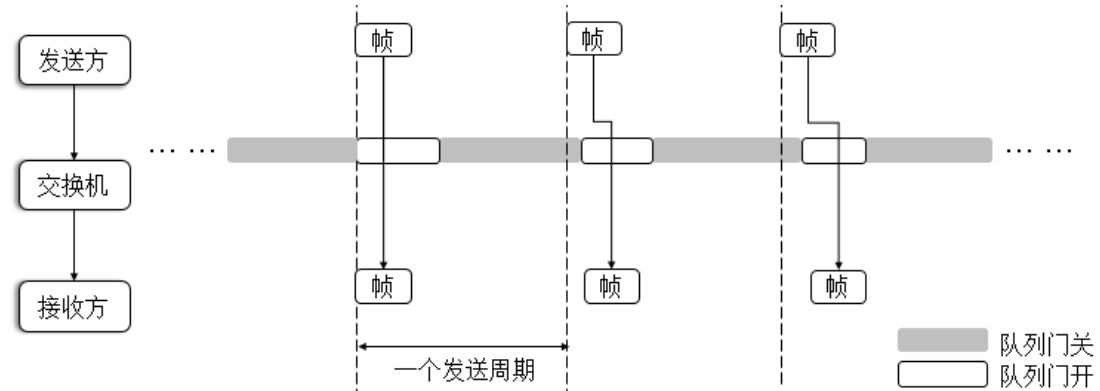


图 3 实际可能调度结果 1

Fig. 3 Possible scheduling results 1

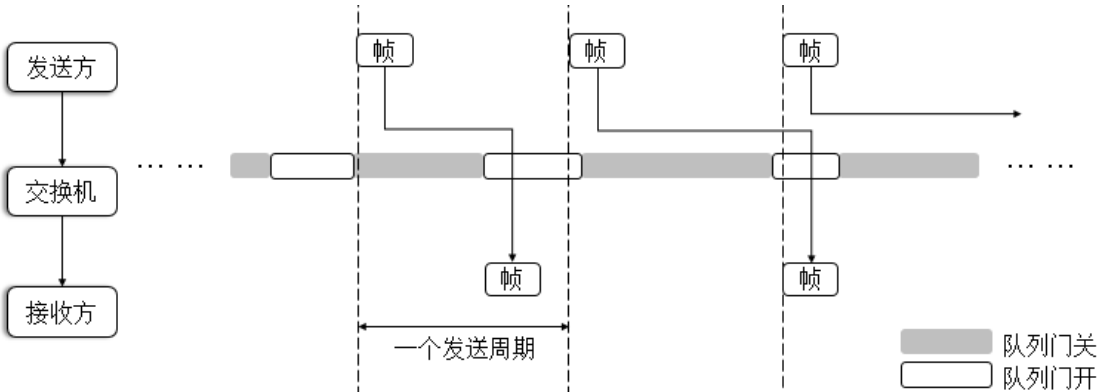


图 4 实际可能调度结果 2

Fig. 4 Possible scheduling results 2

情况 1: 时间的不同导致了传输窗口的轻微错位, 数据帧到达交换机后不能立刻传输, 被短暂阻塞, 从而导致时延和抖动轻微增加, 如图 3 所示。

情况 2: 时间的不同导致了传输窗口的严重错位, 使得数据帧到达交换机后错过了对应的传输窗口, 从而只能在缓存队列中等待, 直到下一个周期传输窗口进行传输, 从而占用了下一个帧的传输窗口, 引发连锁反应, 使得后续所有的数据帧都会延后传输, 导致时延和抖动较大幅度的增加, 如图 4 所示。

3 时间误差分析

3.1 时钟运行偏差

网络设备的时间，一般都是来自于自身的时钟。时钟周期性的输出一个额定频率的脉冲信号，而设备通过计数器计算脉冲振荡次数，得到我们平时所说的时间值。

在一个网络中，每一个设备的时钟初始化都具有随机性，如果没有专门校准，各个节点的初相位之间会存在一个差值，称为时钟偏移或者相位偏移（offset）；另外，由于环境、温度等因素的影响，时钟实际的震荡频率与额定频率之间也会存在这一定范围的偏差，称为频率偏移（skew），单位为 ppm（part per million），变化范围在 1~100ppm 之间。即使网络中的所有设备在开始时 offset 为 0，而 skew 不为 0 且各不相同，经过一段时间后，各个设备间还是会出现一个时钟偏移。

3.2 时钟同步误差

时间同步目的就是为了校正时钟运行偏差。TAS 一般结合 IEEE 802.1AS 中规定的 gPTP 时间同步协议进行使用。gPTP 协议默认为每 1 秒钟进行一次时间同步，也就是每过 1 秒钟会对时钟偏移进行一次修正。然而，时间同步在进行修正的同时，又会引入新的干扰因素：同步误差。基于数据包交换的时间同步协议要求时间戳的时间尽可能的精确，但是在获取时间戳时，通常会由于 CPU 的处理时延、传输时延、中断等各种因素的影响，导致不能获取到完全精确的时间戳，导致误差的产生。gPTP 在计算时延对设备时间进行补偿时，默认从发起者到接收者的时延和从接收者到发起者的时延是相等的，然而由于网络波动、传输路径的不确定性等因素的影响，两者并不会完全一致，导致进行延时补偿时也会引入误差。

4 基于 OMNeT++的仿真

我们通过使用^[9]中 OMNeT++平台下基于 INET 框架的基础上开发的时间敏感网络仿真框架 TSimNet，来进行仿真。对于主时钟的节点，我们仍然使用 OMNeT++中的 API 函数 simTime()获得的仿真时间作为精确的时间来源，但从时钟的节点使用单独的一个 Clock 模块来模拟不准确的从时钟。

在文献^[10]中，作者提出了一种自回归模型，时钟偏移率被建模成一个自回归方式中具有微小波动的时变过程。在此基础上，我们提出了一种简单的离散的时钟偏移数学模型：

$$\theta(k+1) = \theta(k) + \gamma(k) \cdot \Delta t \quad (1)$$

$$r(k+1) = \gamma(k) + \omega_s(k) \quad (2)$$

其中, k 代表用时设备从时钟取时的次数, $\theta(k)$ 对应第 k 次时间的 offset, Δt 为取时的时间间隔, $\gamma(k)$ 对应第 k 次获取时间时, 时钟的漂移率 skew, $\omega_s(k)$ 代表均值为 0 的噪声,

服从高斯分布，方差为 σ_s^2 。

对于时间同步过程中链路传输时延的补偿误差的数学模型，我们对不同情况组网下的 pps 相位偏移进行了测试，得出了多组测试结果，通过分析结果我们发现，一般相位偏移量在 $\pm T$ 内震荡，取值范围在 100ns~500ns 之间，因此我们将时间同步过程中传输时延补偿误差数学模型建立为一个有着上下限的随机过程：

$$t_r(i) = t_c(i) + \omega_r(i) \quad (3)$$

其中， $t_c(i)$ 表示主时钟第 i 次进行 gPTP 同步时的准确的时间值， $t_r(i)$ 表示从时钟第 i 次进行 gPTP 同步后获取到的实际时间戳的值，而 ω_r 代表均值为 0，方差为 σ_r^2 的高斯分布。

从时钟相对于主时钟的最终偏差为两者的积累，到第 i 次进行时间同步时，将本地时钟起始时间 $t(0)$ 赋值为 $t_r(i)$ ，这样第 i 次时间同步到第 $i+1$ 次时间同步之间的第 k 次获取到的实际时间为：

$$t(k) = t_r(i) + k \cdot \Delta t + \theta(k) \quad (4)$$

随后将这个值赋给从时钟的 Clock 模块，并使仿真交换机中的 GCL 执行模块按照 Clock 模块的时间执行，通过改变时钟偏移以及时间补偿误差的大小，来得出不同时间性能下 TAS 调度性能的变化。

我们设置一个没有任何误差的“理想情况”作为对比项，来对比不同参数设置情况下时延的恶化程度。所有仿真情况参数设置见表 1。

表 1 理想情况参数设置
Tab. 1 Ideal situation parameter setting

	σ_s	$\gamma(0) / ppm$	$\Delta t / us$	σ_r
理想情况	0	0	1	0
1.1	0	0	1	10^{-7}
1.2	0	0	1	4×10^{-7}
2.1	10^{-9}	1	1	0
2.2	10^{-8}	10	1	0

仿真使用的拓扑图如图 5 所示。Host1 的流量发生器的设置为以 1000us 为一个周期向 host2 发送流量，每个设置测试两种情况，一个周期发送测试数据帧个数分别为 1 个/10 个。交换机的 TAS 同样以 1000us 为一个调度周期，在每个周期开始时打开测试数据帧优先级的队列门，门开时间以数据帧在千兆端口传输时长，周期的其余时间所有传输门关闭。结果见图 6~图 13。

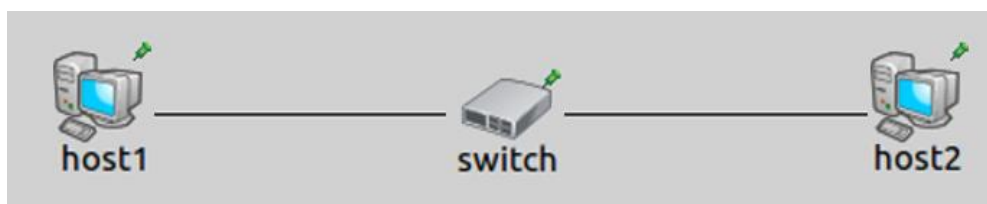


图 5 仿真拓扑
Fig.5 Simulation topology

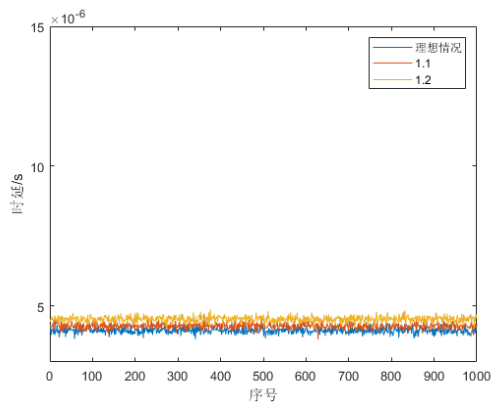


图 6 每周期 1 个帧结果(1)
Fig.6 One frame per cycle result (1)

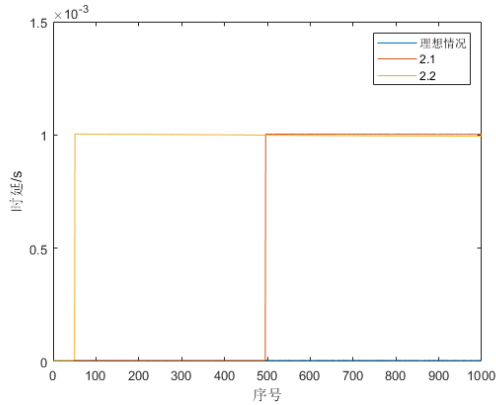


图 7 每周期 1 个帧结果(2)
Fig.7 One frame per cycle result (2)

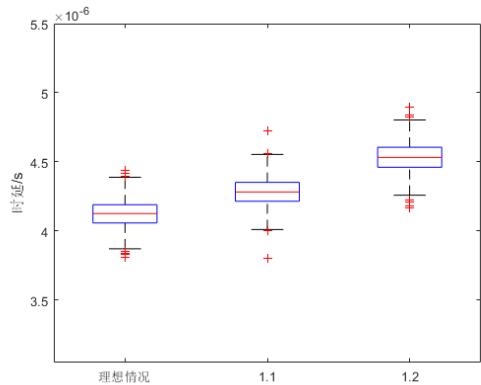


图 8 每周期 1 个帧结果图(3)
Fig.8 One frame per cycle result (3)

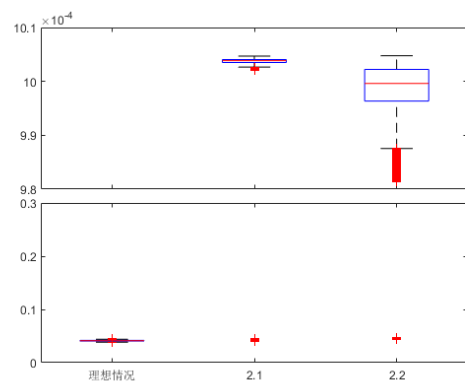


图 9 每周期 1 个帧结果图(4)
Fig.9 One frame per cycle result (4)

根据仿真结果显示，当发送周期的测试帧数量固定时，时钟运行精度和时间同步精度越差，测试帧经过 TAS 调度后的传输性能越差；当时钟运行精度和时间同步精度固定时，一个周期较多数据数量经过 TAS 调度比较少的数据帧数量经过 TAS 调度，对时钟运行精度和时间同步精度的变化更加不敏感。时钟运行精度的变化会更加直观的影响 TAS 的调度结果，是导致上文提到的“错位”现象的主要原因。我们推测这是由于是在一个时间同步的间隔内，时间同步的误差影响是固定的，而时钟运行时产生的误差会导致时延不断的变化，另外，PTP 以及 gPTP 时间同步协议只会对时钟便宜 offset 进行补偿，并不会对时钟偏移率 skew 进行补偿，这就导致了随着时间的增加，每个同步间隔内的时间性能越来越差，加剧了这一现象的产生。

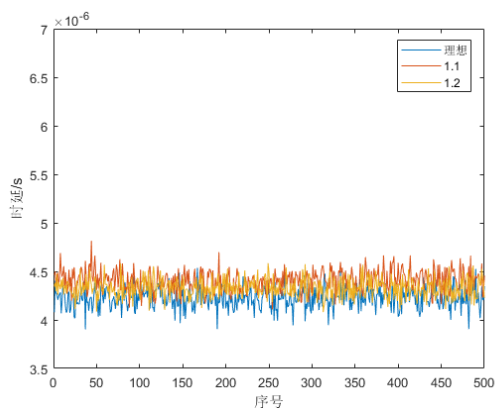


图 10 每周期 10 个帧结果图(1)

Fig.10 One frame per cycle result (3)

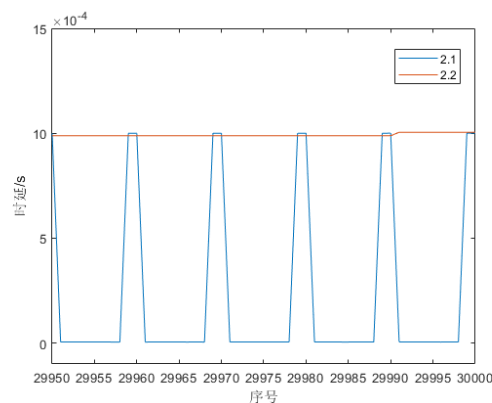


图 11 每周期 10 个帧结果图(2)

Fig.11 One frame per cycle result (4)

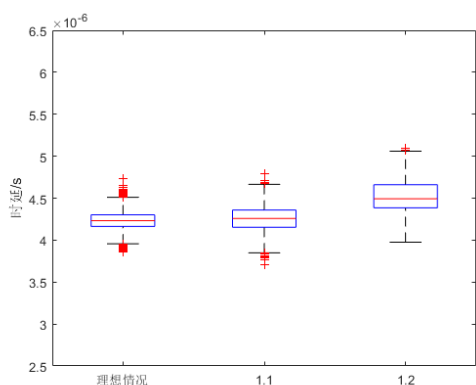


图 12 每周期 10 个帧结果图(3)

Fig.12 One frame per cycle result (3)

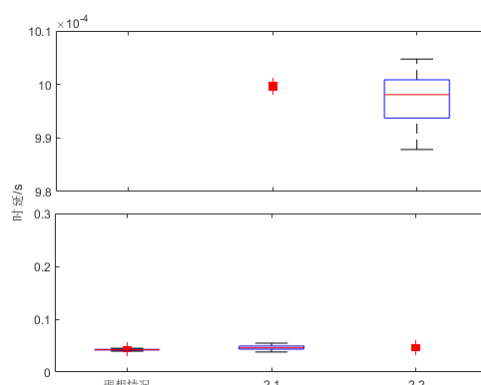


图 13 每周期 10 个帧结果图(4)

Fig.13 One frame per cycle result (4)

5 对于时延增加的修正

通过仿真得知，错位现象的主要原因是 TAS 的本地时钟偏移导致的传输窗口与流量发送周期的错位，为了弥补这种错位，可以采用以下三种方法：1 提高时间同步的频率。PTP 支持 1 秒同步 1~128 次，提高频率可以减小时钟偏移率的影响，优点是仅需修改 PTP 的配置文件，操作简单，但是会占用更多的带宽，并可能会与其他帧产生额外的竞争。2.时间同步时也对时钟偏移率进行修正。目前已经有不少研究成果，比如 ITU 的 G.8625，可以通过 PTP 的同步报文估计时钟的时钟偏移率，并对其进行修正，极大改善本地时钟的性能,但需要对系统时钟的同步补偿功能进行修改，操作复杂；3.增加 GCL 传输窗口的长度，以保证即使错位数据帧也有足够的时间完成传输，但如果增加的长度太短，则效果不明显，太长在网络环境复杂、需调度流量较多的情况下，会有与增加时间同步频率同样的问题。因此本文提出了一种实时计算 GCL 的方法，通过第 T 次同步时交换机的本地时间 t_s 和同步报文中携带的时间 t_m ，以及同步间隔 ΔT ，来得出时钟偏移率 $\gamma(T)$

$$\gamma(T) = \frac{t_m - t_s}{\Delta T} \quad (5)$$

并以此来计算下一个同步周期内的传输窗口长度。我们假设 TAS 的 Tick 长度为 T_{Tick} ，帧的理论传输窗口区间为 (t_b, t_e) ，则需要增加 T_{Tick} 的个数 n

$$n = \left\lceil \frac{(t_e - t_b) \cdot \gamma(T)}{T_{Tick}} \right\rceil \quad (6)$$

则我们设置实际的传输窗口区间为 $(t_b - n \cdot T_{Tick}, t_e + n \cdot T_{Tick})$ 。我们使用时延最差的每周
期 1 个帧的 2.2 参数设置重新进行仿真，得到结果如图 14~图 15。结果表明，经过补偿后，
TAS 的调度结果得到了明显的改善。

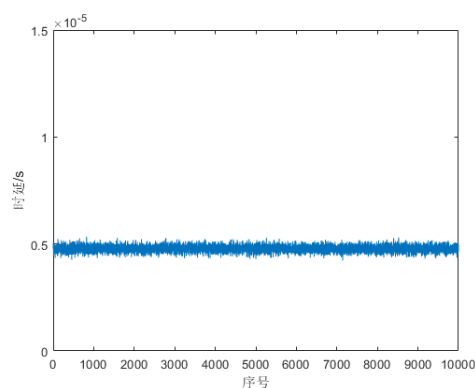


图 14 修正后结果图(1)

Fig.14 Compensation results (1)

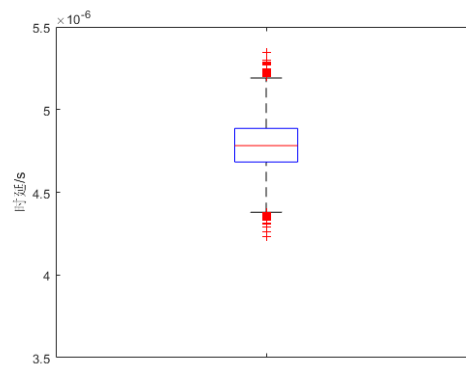


图 15 修正后结果图(2)

Fig.15 One frame per cycle result (2)

结论

本文在 OMNET++ 平台上对时间误差进行了模拟，来研究 TAS 调度性能的变化规律。结果表明，由于时钟偏差的增加，导致传输窗口的错位，时延会一定程度增加。一个周期较多数据数量经过 TAS 调度比较少的数据帧数量经过 TAS 调度，时钟运行精度和时间同步精度的变化更加不容易引起时延的增加。时钟运行精度的变化会更加显著的影响 TAS 的调度结果。针对这个问题，本文提出了对 GCL 的补偿方法，经过进一步仿真验证，证明该方法可以较大程度改善 TAS 在较差时间性能下的调度结果。

[参考文献] (References)

- [1] Time Sensitive Networking Task Group. IEEE Standards 802.1, Institute of Electrical and Electronics Engineers[EB/OL],2016-07-06[2020-01-17]. <http://www.ieee802.org/1/pages/tsn.html>
- [2] IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic: IEEE Std 802.1Qbv-2015[S]. Time Sensitive Networking (TSN) Task Group. 2016.
- [3] IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks: IEEE Std 802.1AS-2011[S].
- [4] Nasrallah et al., Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research[J], in IEEE Communications Surveys & Tutorials, 2019,21(1):88-145.
- [5] Nasrallah et al.Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS)[J].IEEE Access,2019,7:44165-44181.
- [6]L. Zhao, P. Pop and S. S. Craciunas. Worst-Case Latency Analysis for IEEE 802.1Qbv Time Sensitive

Networks Using Network Calculus[J],IEEE Access,2018,6: 41803-41815.

[7]Craciunas S S , Oliver R S , Martin Chmelfk, et al. Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks[C]. 24th International Conference on Real-Time Networks and Systems. ACM, 2016.

240 [8]M. H. Farzaneh and A. Knoll. Time-sensitive networking (TSN): An experimental setup[C]. 2017 IEEE Vehicular Networking Conference (VNC), Torino, 2017, 23-26.

[9]P. Heise, F. Geyer and R. Obermaisser. TSimNet: An Industrial Time Sensitive Networking Simulation Framework Based on OMNeT++[C]. 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Larnaca, 2016, 1-5.

245 [10]Benjamin R. Hamilton, Xiaoli Ma, Qi Zhao. ACES: Adaptive clock estimation and synchronization using Kalman filtering[C]. Proceedings of the 14th Annual International Conference on Mobile Computing and Networking, MOBICOM 2008, San Francisco, California, USA,2008, September 14-19.