

引用格式:曾宇涛,刘泽响,徐丹妮.一种时间敏感网络时钟同步的端口增强方法[J].微电子学与计算机,2021,38(7):85-90.
[ZENG Y T, LIU Z X, XU D N. A port enhanced method for time-sensitive network clock synchronization[J]. Microelectron-ics & Computer, 2021, 38(7): 85-90.]

一种时间敏感网络时钟同步的端口增强方法

曾宇涛,刘泽响,徐丹妮

(西安微电子技术研究所,陕西 西安 710054)

摘 要: 围绕时间敏感网络(time-sensitive network, TSN)时钟同步过程中存在的同步信息浪费与可靠性问题,开展提高网络时钟同步性能的研究.在原有时钟同步生成树基础上,提出了一种包含新型同步端口角色和时钟修正算法的端口增强方法.该方法利用原本被丢弃的同步信息,不仅为从时钟节点的同步提供参考校正,还为该从时钟节点的同步建立额外的冗余路径.在 OMNeT++ 上的仿真结果表明:该方法不仅提高了网络中同步信息使用率,而且在未增加网络负载的情况下,提高了节点时钟同步的准确性和健壮性.

关键词: 时间敏感网络;时钟同步;端口增强;时钟同步生成树

中图分类号: TP393.1

文献标识码: A

文章编号: 1000-7180(2021)07-0085-06

A port enhanced method for time-sensitive network clock synchronization

ZENG Yutao, LIU Zexiang, XU Danni

(Xi'an Microelectronics Technology Institute, Xi'an 710054, China)

Abstract: Aiming at the problem of synchronization information waste and reliability in time-sensitive network (TSN) clock synchronization process, the research on improving network clock synchronization performance was carried out. Based on the original clock synchronization spanning tree, a new port enhancement method is proposed, which includes a new synchronization port role and a clock correction algorithm. This method not only provides reference correction for synchronization of the slave clock node, but also establishes an additional redundant path for synchronization of the slave clock node by using the original discarded synchronization information. The simulation results based on OMNeT++ show that the method not only improves the utilization rate of synchronization information in the network, but also improves the accuracy and robustness of node clock synchronization without aggravating network traffic overhead.

Key words: time-sensitive network; clock synchronization; port enhancement; time-synchronization spanning tree

1 引言

随着工业互联网技术和自动驾驶系统的发展,传统的通讯总线 EtherCAT、SERCOS III、CAN 等已无法满足大量互联数据的共网传输需求.时间敏感网络作为实时通信和多网兼容通信的理想载体,相比传统通讯总线更加灵活、更加高效地实现网络

互联.设计具有实时性、灵活性、高速性的时间敏感网络已经成为目前工业网络通信领域的研究热点,具有重要的现实意义.时钟同步作为时间敏感网络的核心关键技术,得到广泛的关注. IEEE TSN 工作组在 2011 年发布了 IEEE 802.1AS 协议,建立起 TSN 时钟同步标准.该协议是从 IEEE 1588v2 发展而来的一种主从式时钟同步协议,通过在网络中选

取主时钟,构建以主时钟为根节点的时钟同步生成树,由主时钟定期广播同步报文实现整个网络的时钟同步^[1-2].

国内外的专家学者对 IEEE 802.1AS 时钟同步机制进行了大量的研究,文献[3]提出了一种重选主时钟的方法,能够在当前同步失效后,通过预设的主时钟竞争信息,快速的重新竞争选择出主时钟.文献[4]提出了一种在车载网络上,根据网络情况自动调整同步间隔的方法,能够有效降低失效风险和网络负载.但以上的研究仍存在面对故障时需要重新花费时间进行主时钟竞争,未能实现提升冗余可靠性的问题,以及单独调整同步间隔未能实现提升同步精度的问题等,本文围绕上述问题开展一种新的时钟同步方法研究.

2 IEEE 802.1AS 时钟同步技术

2.1 BMCA 与时钟同步生成树

BMCA(最佳主时钟竞争算法)是网络中时钟同步的第一步,网络中所有节点通过 BMCA 竞争出时钟域中带有最好时钟源的节点,设为主时钟,同时建立以主时钟为根节点的时钟同步生成树.在竞争阶段,网络中各相邻节点互相发送 Announce 报文交换最佳主时钟选择信息.每个 Announce 报文包含时钟同步生成树向量信息,该信息标记了网络中一个节点作为根节点,即主时钟.每个网络节点将接收到的 Announce 报文中的信息与自身节点的信息进行比较,计算选择主时钟信息更好的节点为根节点.

作为构建时钟同步生成树的一部分,每个节点的每个端口都通过与端口关联的状态机分配端口角色,共分为四种,如表 1 所示.

表 1 端口角色定义	
端口角色	描述
Master	节点上不是最接近根节点的,同时向外传递 Sync 和 Announce 报文的端口.
Slave	节点上最接近根节点的一个端口.每个节点只有一个 Slave,节点只使用从 Slave 端口接收到的同步信息.节点不从 Slave 端口传输 Sync 和 Announce 报文.
Passive	节点上端口角色不是 Master、Slave 或者 Disabled 的其他端口.在 Passive 端口接收到的 Sync 和 Follow_Up 报文都会被端口丢弃.
Disabled	节点上没有时钟同步功能的端口.

BMCA 在构建时钟同步生成树的过程中,需要配置的是各节点上各端口的端口角色.在选定主时钟节点后,该节点上的所有端口均设定为 Master 端

口,然后沿网络逐跳向外扩展时钟同步生成树.对于从时钟节点,首先需要选择设定一个唯一的接收同步信息的 Slave 端口.从时钟节点上 Slave 端口的选择依据 Announce 报文中{距主时钟的跳数;主时钟发送端口序列号;节点接收端口序列号}的优先级向量信息,从高位到低位逐次判断,越小越优.在选定节点上唯一的 Slave 端口后,将其他接收到不超过自身跳数的 Announce 报文的端口设定为 Passive.其它端口设定为 Master,不支持 IEEE 802.1AS 的端口设定为 Disabled.

在上述优先级向量{距主时钟的跳数;主时钟发送端口序列号;节点接收端口序列号}中,前一组件{距主时钟的跳数}是最主要的判断依据,而后两组件{主时钟发送端口序列号;节点接收端口序列号}只有在前面的向量组件都相等时,才作为判断依据加入决策.跳数能够成为影响网络中节点时钟同步精度的最主要因素,是因为每多增加一跳会带来累积的节点本地时钟漂移误差、端口抖动误差和链路延时误差.而端口序列号仅作为一种仲裁决策依据,用于选择出节点上唯一的 Slave 端口,对同步精度没有影响.

一种典型拓扑的时间敏感网络^[5]的主从层次结构如图 1 所示.主时钟节点的所有端口均为 Master 端口.其他从时钟节点有且只有一个 Slave 端口.每个从时钟节点通过 Slave 端口与上一级节点的一个 Master 端口相连接,这样逐级相连形成一条同步路径,多条由主时钟根节点开始的互不相交的同步路径组成了时钟同步生成树^[6].由图 1 构建的时钟同步生成树如图 2 所示.

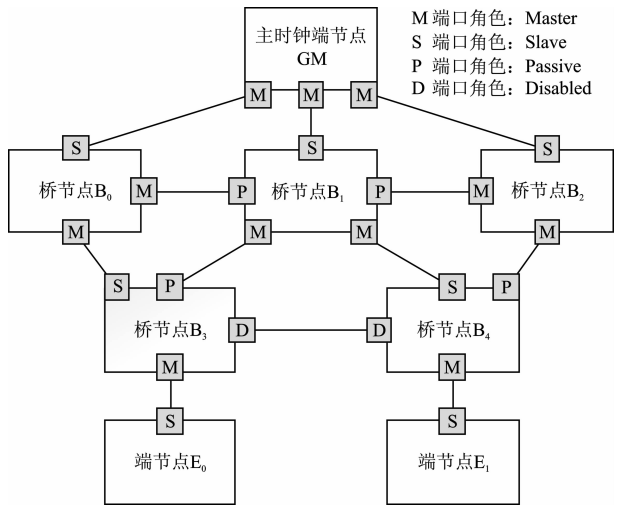


图 1 时间敏感网络的主从层次结构

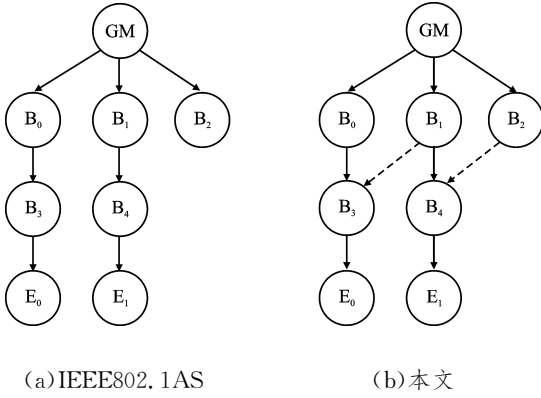


图 2 时钟同步生成树

2.2 时钟同步信息传递

在 TSN 网络中经过 BMCA 构建出时钟同步生成树之后,主时钟 GM 就会每隔一个时钟同步间隔,向网络中广播时钟同步信息.整个过程通过传递 Sync 和 Follow_Up 报文来实现,如图 3 所示.

图中 3 个临近的网络节点分别表示为 $i-1$, i 和

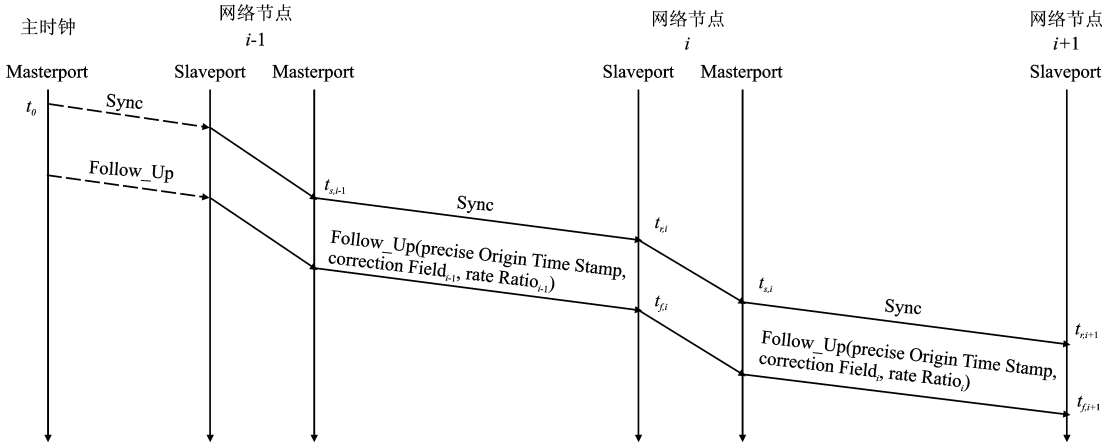


图 3 时钟同步信息传递

节点 i 接收到 Sync 报文,并记录本地时间戳 $t_{r,i}$,然后向节点 $i+1$ 发送一个新的 Sync 报文,并记录本地时间戳 $t_{s,i}$.在一段时间之后,从时钟节点 $i-1$ 接收到关联的 Follow_Up 报文,并记录本地时间戳 $t_{f,i}$.然后根据式(1),计算出同步时间,调整节点本地时钟与主时钟同步.

$$\text{timeSynced}_i = \text{preciseOriginTimestamp} + \text{Pdelay}_{i-1} + \text{correctionField}_{i-1} + \text{transmissionTime} + \text{waitTime}_i \quad (1)$$

式中, timeSynced_i 表示的是节点计算出的同步主时钟的时间; Pdelay_{i-1} 表示节点 $i-1$ 与 i 之间的链路延时; transmissionTime 表示节点端口发送 Sync 报文的时间,与报文长度和端口速率有关; waitTime_i

$i+1$.由主时钟发出的时钟同步信息从时钟节点 $i-1$ 传递到 i ,再传递到 $i+1$.节点 $i-1$ 先向节点 i 发送一个 Sync 报文,并记录本地时间戳 $t_{s,i-1}$.在一段时间之后,节点 $i-1$ 再向节点 i 发送一个关联的 Follow_Up 报文,其中包含了主时钟同步发起时刻 $\text{preciseOriginTimestamp}$,校正域 $\text{correctionField}_{i-1}$ 和时钟频率比 rateRatio_{i-1} 等元素.元素 $\text{preciseOriginTimestamp}$ 代表的是主时钟发起此次时钟同步,广播发出 Sync 报文的时刻点 t_0 .此元素没有下标标识,因为在遍历整个网络时,元素值不会发生改变.元素 $\text{correctionField}_{i-1}$ 表示的是当前节点 $i-1$ 上发送 Sync 报文的时刻点 $t_{s,i-1}$ 与 $\text{preciseOriginTimestamp}$ 之间的差值,即 $\text{preciseOriginTimestamp}$ 与 $\text{correctionField}_{i-1}$ 之和等于 $t_{s,i-1}$.元素 rateRatio_{i-1} 表示的是主时钟频率与当前节点 $i-1$ 本地时钟频率的比值.

表示节点在接收到 Sync 报文和关联 Follow_Up 报文之间的间隔时间.

当节点 i 完成时钟同步之后,计算新的 correctionField_i ,即当前节点 i 上发送 Sync 报文的时刻点 $t_{s,i}$ 与 $\text{preciseOriginTimestamp}$ 的差值.因此,需要计算出 $t_{s,i-1}$ 与 $t_{s,i}$ 之间的时间间隔.这个时间间隔主要由以下部分组成:

- (1) 节点 $i-1$ 与 i 之间的链路延时 Pdelay_{i-1} ;
- (2) Sync 报文在节点 i 上的驻留时间 residenceTime_i ,即 $t_{s,i}$ 与 $t_{r,i}$ 之间的差值;
- (3) Sync 报文在端口的传输时间 transmissionTime .

计算如式(2)所示:

$$\text{correctionField}_i = \text{correctionField}_{i-1} +$$

$$\text{Pdelay}_{i-1} + \text{residenceTime}_i + \text{transmissionTime} \quad (2)$$

节点 i 在完成校正域 correctionField_i 更新后, 就会向节点 $i+1$ 发送包含新信息的 Follow_Up 报文. 如此沿着时钟同步生成树逐级完成整个网络的时钟同步.

3 新型端口增强方法

从第 2 节中可以看到, 整个 TSN 网络都是通过时钟同步生成树, 通过主时钟逐级向下传递 Sync 和 Follow_Up 报文来实现网络时钟同步. 同步报文从上一级节点的 Master 端口发出, 在下一级节点的 Slave 端口接收, 而在 Passive 端口接收到的同步报文都会被直接丢弃, 其存在两个问题:

- (1) Passive 端口丢弃报文造成同步信息浪费;
- (2) 同步路径单一, 可靠性差.

针对于以上问题, 本文提出了一种时钟同步端口增强方法, 即在构建时钟同步生成树的过程中, 为一些特定节点上的端口新增一个端口角色 Reference, 替代原 Passive 端口. Reference 定义为当节点上有 Slave 存在时, 其它距主时钟根节点的跳数和 Slave 相同的端口. 新增端口角色 Reference 的依据在于, 根据 BMCA 中选定 Slave 的判定方法, 是按照 {距主时钟的跳数; 主时钟发送端口序列号; 节点接收端口序列号} 的优先级顺序依次判定, 而在同一节点上 Reference 和 Slave 距主时钟的跳数相同, 区别只在于端口序列号. 因此, 根据之前对于同步优先级向量的描述, 可以认为同跳数的 Reference 和 Slave 同步精度相当. 如图 1 所示, 节点 B_3 、 B_4 上的 Passive 和 Slave 距主时钟根节点的跳数都是 2 跳, 因此, Passive 可以改为 Reference 使用. 而节点 B_1 上的 Passive 距离根节点 2 跳, Slave 距离根节点 1 跳, 因此, Passive 不可以改为 Reference 使用. 最后构建出来的时钟同步生成树如图 2(b) 所示.

新增 Reference 的主要功能有:

- (1) 与 Slave 一样接收从上级节点 Master 传递的 Sync 和 Follow_Up 报文, 避免信息、带宽浪费;
- (2) 在接收处理 Sync 和 Follow_Up 后, 得到的主时钟同步信息作为 Slave 的参考信息, 为节点提供更准确的时钟同步;
- (3) 节点 Slave 失效或者与其相连的链路失效、报文传输超时、丢包时, Reference 可以作为节点备份的同

步端口实现时钟同步, 通过链路冗余提高可靠性.

为了判断两条同步路径可能存在的接收超时, 在节点上加入了一个双同步超时时器进行仲裁. 在节点上同时存在 Slave 和 Reference 时, 当其中一个端口完成接收同步信息 Sync 和 Follow_Up, 如果另一端口还未完成接收此次同步信息, 就会启动此计时器. 计时范围内, 另一端口完成此次同步信息接收, 就使用两个端口接收到的信息, 通过式 (3) 分别计算两端口的同步时间再取平均值来调整节点本地时钟; 如果计时溢出, 另一端口未完成此次同步信息接收, 就判断为另一端口同步超时, 仅使用当前端口的同步信息调整节点本地时钟.

$$\begin{aligned} \text{slaveSynced}_i = & \text{preciseOriginTimestamp} + \\ & \text{Pdelay}_{i-1} + \text{correctionField}_{i-1} + \\ & \text{transmissionTime} + \text{waitProcessTime}_i \end{aligned} \quad (3)$$

式中, slaveSynced_i 表示 Slave 端口计算的同步时间, waitProcessTime_i 表示节点在当前端口接收到 Sync 报文到节点判定同步信息接收完成开始计算同步时间之间的间隔时间. Reference 端口同步时间 referenceSynced_i 计算与式 (3) 相同.

双同步超时时器的主要作用在于降低可能存在的某一链路拥塞、失效或者丢包对于此次时钟同步精度的影响. 因为当此节点有一端口已经完成接收同步报文, 在等待另一端口接收报文的同时, 本地时钟由于与主时钟的晶振频率不同会不断的累积偏移误差, 等待时间越长, 误差越大. 因此需要选择一个合适的值作为超时条件, 避免由于过长的等待时间影响本次同步精度.

4 实现与验证

4.1 实验设计

为了评估本文端口增强方法对时钟同步性能的影响作用, 在 OMNeT++ 5.2 软件^[7]上使用 INET 框架搭建了基于图 1 所示架构的网络模型, 如图 4 所示.

为了分别验证 Reference 的参考校正功能和同步冗余功能, 预设了两个网络实验场景, 仿真时间设为 40 s, 主时钟的同步间隔设为 0.125 s, 节点上的双同步超时时器设为 5 ms.

实验一, 测试 Reference 的参考校正能力, 进行对比实验. 设定为所有的链路均正常. 第一次测试

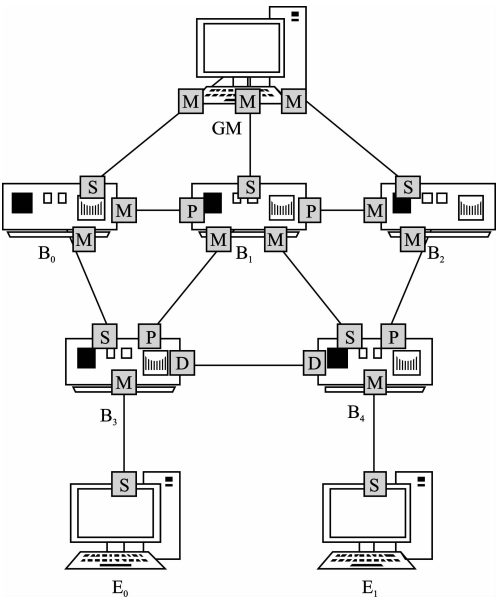


图 4 仿真网络模型

IEEE 802.1AS 协议的同步性能,采用图 1 所示端口角色,未加入 Reference 端口角色.第二次测试本文所提方法的同步性能,加入 Reference 端口角色,将 B_3 与 B_0 的连接端口设定为 Slave,与 B_1 的连接端口设定为 Reference; B_4 与 B_1 的连接端口设定为 Slave,与 B_2 的连接端口设定为 Reference.

实验二,测试双同步路径的冗余可靠性,通过设定链路失效,来模拟同步丢包、超时以及链路失效等在单一链路上同步无法完成的情况.场景在实验一的基础上,设定 B_4 与 B_1 的连接链路在仿真时间 28 s 时失效,即 B_4 上 Slave 接收超时,使用单 Reference 进行同步.

4.2 验证分析

实验一中, B_3 、 B_4 上分别用 802.1AS 单端口方法和本文双端口方法为节点进行时钟同步调整.图 5 是截取的 B_4 本地时钟同步误差,可以看出,本文方法相较于原 802.1AS 协议方法能有更小的同步抖动,更加稳定.经过数据计算,在整个仿真过程中, B_4 上原协议方法同步误差的平均值为 25.855 2 ns,方差为 19.012 4;而本文新方法同步误差的平均值为 24.647 8 ns,方差为 13.914 5.由以上可得,本文方法,既充分利用了原 Passive 端口丢弃的信息,又通过取 Slave 和 Reference 两者的平均值,降低 Slave 端口获取的时钟同步信息的抖动对时钟同步的影响.

实验二中,在 28 s 时 $B_1 \sim B_4$ 链路失效的情况

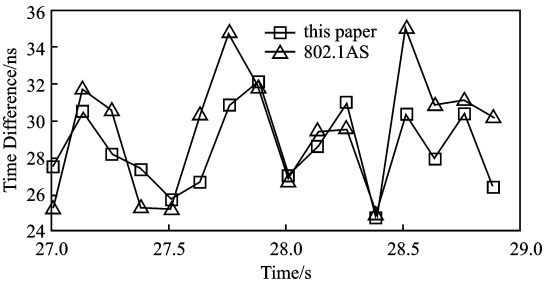


图 5 B_4 同步误差对比

下,节点 B_4 只能单独使用 Reference,通过 $B_2 \sim B_4$ 链路进行同步.图 6 包含节点 B_4 上 Slave、Reference 端口以及节点的时钟同步误差.可以看出,在 28 s 之前,节点使用 Slave 和 Reference 双端口同步信息完成同步,而在 28 s 之后,Slave 端口连接链路失效,节点使用 Reference 单端口完成同步.由以上可得,面对一条同步路径失效的情况,通过 Reference 端口建立的同步冗余路径,同样可以确保网络节点与主时钟实现同步,并且同步精度不受影响.

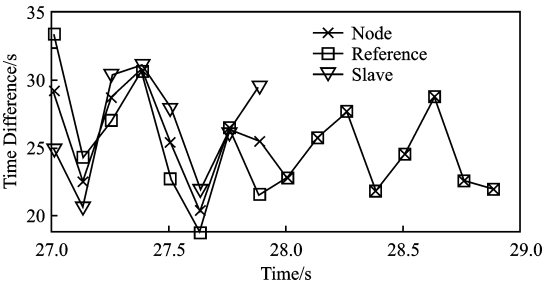


图 6 B_4 同步可靠性测试

5 结束语

时间敏感网络中时钟同步的性能直接影响实时通讯和时间敏感应用,本文提出的时钟同步端口增强方法,通过在现有 BMCA 中新增端口角色 Reference 改进时钟同步机制,仿真结果表明在使用原有同步报文,未增加网络负载的情况下,节点时钟同步的准确性和健壮性得到了提高.

参考文献:

[1] IEEE. IEEE Std 802.1AS-2011 IEEE standard for local and metropolitan area networks-timing and synchronization for time-sensitive applications in bridged local area networks[S]. IEEE, 2011. DOI: 10.1109/IEEESTD.2011.5741898.

[2] IEEE. IEEE Std 1588-2008 IEEE standard for a precision clock synchronization protocol for networked

measurement and control systems[S]. IEEE, 2008. DOI: 10.1109 / IEEESTD. 2008. 4579760.

[3] JEON Y, LEE J, PARK S. An efficient method of re-selecting grand master in IEEE 802. 1AS[C]//The 20th Asia-Pacific Conference on Communication (APCC2014). Pattaya: IEEE, 2014. DOI: 10.1109/APCC. 2014. 7091652.

[4] KIM Y J, CHEON B M, KIM J H, et al. Time synchronization method of IEEE 802. 1AS through automatic optimal sync message period adjustment for in-car network[C]//2015 IEEE International Conference on Information and Automation. Lijiang: IEEE, 2015. DOI: 10.1109/ICInfA. 2015. 7279520.

[5] 丛培壮, 田野, 龚向阳, 等. 时间敏感网络的关键协议及应用场景综述[J]. 电信科学, 2019, 35(10): 31-42. DOI: 10.11959/j. issn. 1000-0801. 2019227.

CONG P Z, TIAN Y, GONG X Y, et al. A survey of key protocol and application scenario of time-sensitive network[J]. Telecommunications Science, 2019, 35(10): 31-42. DOI: 10.11959/j. issn. 1000-0801. 2019227.

[6] GARNER G M, GELTER A, TEENER M J. New simulation and test results for IEEE 802. 1AS timing performance[C]//2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication. Brescia: IEEE, 2009. DOI: 10.1109/ISPCS. 2009. 5340214.

[7] WALLNER W. Simulation of the IEEE 1588 precision time protocol in OMNeT++[C]//Proceedings of the 3rd OMNeT++ Community Summit, Brno University of Technology. Czech Republic, 2016.

作者简介:

曾宇涛 男,(1995-),硕士研究生. 研究方向为嵌入式计算机系统结构. E-mail: zengyutao1020@126. con.

刘泽响 男,(1978-),研究员. 研究方向为嵌入式计算机系统结构.

徐丹妮 女,(1989-),高级工程师. 研究方向为嵌入式计算机系统结构.