

文章编号:1007-5321(2021)02-0020-06

DOI:10.13190/j.jbupt.2020-129

# TSN 网络中时钟同步可靠性提升方法

朱海龙<sup>1,2</sup>, 严园园<sup>2</sup>

(1. 北京邮电大学 网络与交换技术国家重点实验室, 北京 100876; 2. 网络通信与安全紫金山实验室, 南京 211100)

**摘要:** 为了提高时间敏感网络中时钟同步的可靠性, 在 IEEE 802.1AS 协议的基础上, 提出最近端口主时钟备份和主时钟相位偏移热备冗余的方法. 仿真实验结果表明, 最近端口主时钟备份方法比标准的主时钟重选方法的可靠性提高了 38%, 主时钟相位偏移热备冗余比标准的热备冗余方法在精度上可提升 48%.

**关键词:** 时钟同步; 冗余; 相位; 最近端; 重选

**中图分类号:** TP393.0

**文献标志码:** A

## Measures of Reliability Improvement of Clock Synchronization in Time Sensitive Networking

ZHU Hai-long<sup>1,2</sup>, YAN Yuan-yuan<sup>2</sup>

(1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. Purple Mountain Laboratories, Nanjing 211100, China)

**Abstract:** In order to improve the reliability of clock synchronization, based on the IEEE 802.1AS protocol, a method for the nearest port master clock backup and the master clock phase offset hot backup redundancy method is proposed. Simulation shows that the recent port master clock backup method is 38% more reliable than the standard master clock reselection method, and the master clock phase offset hot backup redundancy is 48% more accurate than the standard hot backup redundancy method.

**Key words:** clock synchronization; redundancy; phase; nearest port; reselection

随着“工业 4.0”概念的提出<sup>[1]</sup>, 以太网将作为下一代核心网络通信技术应用在工业网络和车载网络中, 但传统以太网“尽力而为”的数据转发机制不能满足工业互联网和无人驾驶的数据传输需求. 2012 年, 美国电气与电子工程师学会 (IEEE, institute of electrical and electronics engineering) 提出了时间敏感网络 (TSN, time sensitive networking) 标准协议簇, 能够为以太网提供确定性、低时延的数据转发机制. TSN 标准协议簇包含了时钟同步、流量调度、可靠性和系统配置等方面的一系列规范, 所有的协

议运行都必须建立在严格的时钟同步基础上, 否则将无法保障数据转发及传输的确定性和可靠性. 因此, 网络节点之间严格的时钟同步是保障 TSN 确定性和低时延的前提和基础.

TSN 使用 IEEE 802.1AS 协议实现严格的时钟同步<sup>[1]</sup>. 网络实际运行过程中, 作为时钟同步基准的主时钟有可能会发生晶振烧毁、断电等故障. 除了主时钟自身可能出现故障以外, 主时钟和各个从时钟之间的网络链路也可能出现故障. 在网络负荷较高时, 可能会出现网络拥塞的情况. 以上这些

收稿日期: 2020-08-20

基金项目: 国家重点研发计划项目 (2020YFB1805301)

作者简介: 朱海龙 (1987—), 男, 讲师.

通信作者: 严园园 (1989—), 男, 工程师, E-mail: yanyuanyuan@pmlabs.com.cn.

情况都会影响时钟同步的可靠性,导致 TSN 出现故障。Nikolaus 等<sup>[2]</sup>考虑了多种时钟同步异常的情况,提出冗余机制能够有效地保障时钟同步的可靠性,并分别为主时钟热备冗余和链路冗余提供了参考,但未进行深入地仿真研究<sup>[2-4]</sup>。Maryam 等<sup>[5]</sup>在 Opnet 平台上基于一种菊花链拓扑仿真了主时钟自身发生故障的情况,采用主时钟重选机制在一段时间之后重新选出了一个新的主时钟,以恢复时钟同步功能。

针对主时钟自身故障的情况,笔者提出了一种改进的提高时钟同步可靠性的方法,即最近端口主时钟备份算法。进一步地,在标准主时钟热备冗余的基础上,提出了主时钟相位偏移的优化算法,可以在保障高可靠性的前提下,进一步提升时钟同步的精度。最后,基于 OMNet++ 仿真平台,采用一种车载网络拓扑结构,对提出的提升时钟同步可靠性的几种算法分别进行了仿真验证。仿真结果表明,最近端口主时钟备份方法比标准主时钟重选方法的可靠性提高了 38%,主时钟相位偏移热备冗余方法比标准的热备冗余方法在精度上提升了约 48%。

1 IEEE 802.1AS 基本原理

IEEE 802.1AS 协议的基本流程为:同一个域内各时钟执行最佳主时钟算法 (BMCA, best master clock algorithm),根据各时钟的性能参数选出性能最好的时钟作为主时钟。BMCA 算法执行完毕,各时钟之间的连接关系和连接树即确定完成,且各时钟的角色也确定完成(主时钟或从时钟)。确定主时钟之后,主时钟会周期性地发送同步报文,从时钟接收同步报文后将本地时间修正为同步报文中的时间,从而达到主从时钟保持一致的目的。此外,主时钟还会周期性地发送 Announce 报文以表明自身的状态,从时钟根据 Announce 报文来判断主时钟的状态是否正常。主时钟发送时钟同步报文以后,该报文到达从时钟的时间存在一定的时延,需要测量该时延,以补偿同步的时间。从时钟根据时延测量过程中获取的  $t_1, t_2, t_3, t_4$  时间戳计算出主从时钟之间的时间偏移,有

$$T_o = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \tag{1}$$

计算出时间偏移值后,从时钟修正本地时钟,与主时钟一致,修正后的时钟值为

$$t_n = t_l + T_o. \tag{2}$$

其中  $t_l$  为从时钟的本地时钟值。

IEEE 802.1AS 协议中描述了重选机制,以解决时钟同步的可靠性。该机制可以在一段时间后重新选出一个新的主时钟,使时钟同步功能恢复正常,但仍会导致网络中有一段时间没有主时钟,且该时间有可能较长。假设在原主时钟异常、新的主时钟尚未选出期间,网络出现较多的时延敏感报文传输需求,可能会使报文无法及时有效传输,导致网络故障。因此,需要研究新的提升时钟同步可靠性的方案,以进一步提升时钟同步的可靠性。

2 主时钟重选机制及优化

2.1 主时钟重选机制

主时钟周期性地发送 Announce 报文,该报文用来标识主时钟状态。当主时钟发生故障后,从时钟将无法接收到 Announce 报文。当从时钟在一定周期内没有接收到主时钟发送的 Announce 报文时,则认为主时钟发生了故障,启动重选机制。从时钟将原主时钟的参数从本地剔除,重新发起 BMCA 流程,待 BMCA 流程结束后,即重新选出一个新的主时钟,该主时钟是时钟同步域中剔除原主时钟后性能参数最优的时钟。

2.2 主时钟重选机制优化

从时钟在一定周期内没有接收到主时钟发送的 Announce 消息,认为主时钟发生了故障,不重新发起 BMCA 流程,而是以离主时钟最近的从时钟节点作为时钟同步域新的主时钟,其他从时钟与该离原主时钟最近的节点进行时钟同步。确定主时钟最近端从时钟的方法为:优选到达主时钟节点跳数最小的从时钟,跳数相同时按照顺时针方向选取第 1 个从时钟。当网络拓扑搭建后,由于时钟参数最优的时钟节点是可知的,可直接配置最近端从时钟节点,该节点可直接使用相应配置。此优化方案可以节省重新执行 BMCA 算法的时间,从而提升时钟同步的可靠性。

将最近端从时钟确定为主时钟后,新的主时钟可能有立即同步和先更新再同步这 2 种时钟同步方式。

2.2.1 即刻同步

确定最近端从时钟为主时钟后,其他从时钟从本地移除原主时钟信息。新的主时钟立即发起时钟同步,且发送 Announce 消息,时钟同步功能恢复。

### 2.2.2 更新后同步

确定最近端从时钟为主时钟后,其他从时钟从本地移除原主时钟信息.新确定的主时钟根据本地记录的上一次原主时钟同步的信息计算出一个基准时钟和基准时钟频率,将本地时钟和时钟频率更新为计算出的基准时钟和基准时钟频率.更新完成后,新的主时钟发起时钟同步,且发送 Announce 消息,时钟同步功能恢复,流程如图 1 所示.

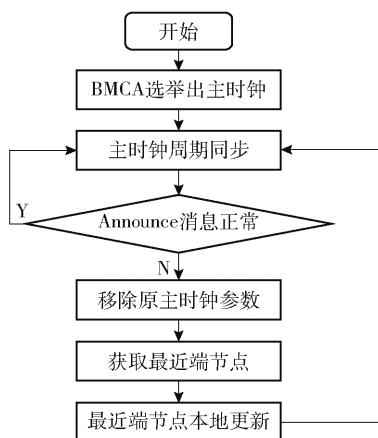


图1 带更新的最近端算法流程

时钟计数的频率是由晶振震荡周期与扩频器决定的,此处的时钟频率是指每一跳的时钟计数变化与相邻两跳之间时间间隔的对齐.更新时改变的是时钟计数变化值,计算方式为:时钟计数间隔设为  $T_0$ ,每一次计数的本地时间变化为  $J_0$ ,首先计算同步时间与同步前本地时间的差值  $T_d$ :

$$T_d = t_m - t_l \quad (3)$$

同时,从时钟将本地时间与主时钟同步,记为  $t_s$ .此时计算2次时钟同步的时间间隔  $T_i$

$$T_i = t_s - t_0 \quad (4)$$

其中,  $t_0$  为时钟上一次完成同步的时间,据此可以计算出从时钟与主时钟的频率差  $J_d$

$$J_d = T_0 (T_d / T_i) \quad (5)$$

由此调节时钟计数间隔  $J$  的值为

$$J = J_0 + J_d \quad (6)$$

至此,时钟的频率更新完成.原主时钟此时的时钟值  $t_u$  计算为

$$t_u = t_s + NJ \quad (7)$$

其中:  $N$  为从上一次同步成功至计算时为止所经过的 tick 数目,将新确定的主时钟的本地时钟值更新为  $t_u$ .

## 3 主时钟热备冗余机制及优化

IEEE 802.1AS 协议中的域由一个或多个时间感知设备和链路组成,这些设备和链路满足该标准的要求并且能按照该协议所定义的方式彼此通信.域定义了时钟同步消息通信、状态、操作、数据集和时间刻度等的作用范围,域编号相同的设备和链路位于同一个域,它们之间可以进行时钟同步.同一个时间感知设备可以同时处于几个不同的域中.

TSN 工作组为了解决时钟同步的可靠性问题,提出了主时钟热备冗余机制,且在 2020 年将其标准化.若采用主时钟热备冗余,即多个主时钟同时工作,当某个主时钟发生异常之后,仍有热备主时钟在正常工作,对时钟同步功能无影响,可有效解决主时钟发生故障的问题.若采用链路冗余,即时钟同步报文经过多条不同的链路进行传输,当某一条网络链路发生故障以后,冗余的链路仍然正常工作,报文可正常传输,有效解决了链路故障的问题.域冗余是将时钟节点同时设置在 2 个不同的域内,当某个域的主时钟发生异常后,仍可与另一个域的主时钟保持同步,同时实现主时钟热备冗余.通过采用冗余机制,能够实现时钟同步功能的无缝切换,保障时钟同步的可靠性<sup>[2]</sup>.

### 3.1 标准主时钟热备冗余

冗余是解决主时钟发生故障的有效手段.一个时钟同步域内,同时存在多个时钟性能参数最优且相同的时钟节点,执行 BMCA 算法以后,时钟同步域内产生多个主时钟节点,这些主时钟节点同时工作.当某个主时钟节点发生故障后,其他的主时钟节点仍然正常工作,从而网络的时钟同步功能一直处于正常状态.采用冗余机制,在主时钟发生故障的情况下,能够实现无缝切换,保障时钟同步的可靠性.一种可能的主时钟热备冗余示意图如图 2 所示.

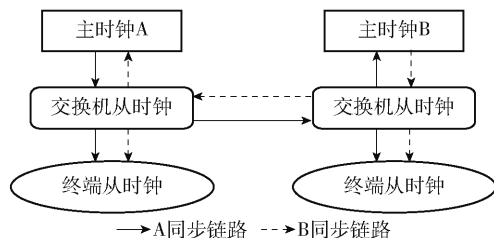


图2 主时钟热备冗余

3.2 主时钟热备冗余优化

在采用标准主时钟热备冗余的网络中,多个主时钟同时工作.若多个主时钟在同一时刻发送时钟同步报文,从时钟会同时收到多个同步报文,然后根据收到的报文更新本地时钟.考虑一种优化算法:在多个主时钟中,有一个主时钟被选出以后就开始发送初始时钟同步报文,然后以固定周期发送;其他的主时钟分别延迟一段时间再发送初始时钟同步报文,然后以相同的固定周期发送,如图 3 所示.

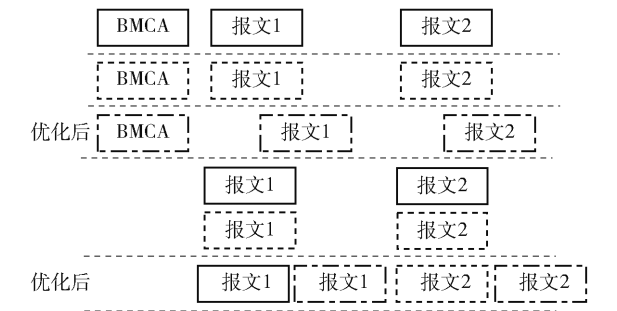


图 3 主时钟热备冗余优化算法

从图 3 可以看到,采用不同的主时钟分别延迟一段时间再发送初始时钟同步报文的算法时,从时钟在不同的时刻收到不同主时钟的时钟同步报文,每收到一次时钟同步报文即更新本地时钟,与主时钟保持一致.此算法的优点在于,在多个主时钟保障时钟同步可靠性的同时,实现了增加时钟同步频率的功能.而有研究表明,主从时钟之间进行时钟同步的频率越高,时钟同步的精度也会越高<sup>[5]</sup>.

4 仿真设计和结果分析

基于 OMNet++ 平台对所提出的几种提高时钟同步可靠性的算法进行仿真验证,并对比分析仿真结果.笔者在 OMNet++ 平台中使用 C++ 语言开发出了 IEEE 802.1AS 协议功能,并构建出了支持 IEEE 802.1AS 协议的终端、交换机、多媒体等节点<sup>[6-9]</sup>.

4.1 仿真拓扑和参数

TSN 网络在车载以太网中的应用越来越多<sup>[5,10]</sup>.仿真设计了一种车载网络拓扑结构,同时包含星形结构和环形结构,以验证本研究所提出的算法,如图 4 所示.

车载网络仿真拓扑包含 10 个车载传感器节点、7 个交换机节点以及 1 个人机接口 (HMI, human

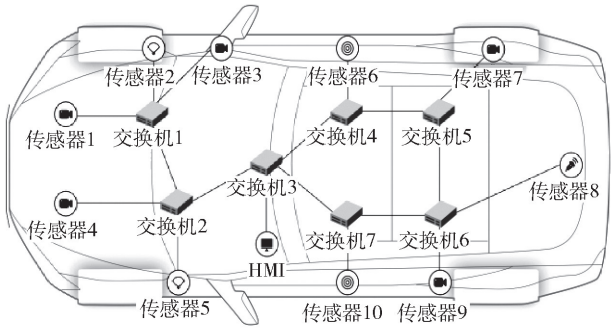


图 4 一种车载网络仿真拓扑

machine interface), 这些节点都支持 IEEE 802.1AS 协议. 仿真所采用的主要参数如表 1 所示.

表 1 仿真参数

参数名称	参数值
交换机之间链路带宽/(Gbit·s <sup>-1</sup> )	10
传感器 9 链路带宽/(Mbit·s <sup>-1</sup> )	100
传感器 10 链路带宽/(Mbit·s <sup>-1</sup> )	100
传感器 1~8 链路带宽/(Gbit·s <sup>-1</sup> )	1
时钟同步周期/ms	1

4.2 仿真设计

- 结合仿真拓扑图,提出以下仿真设计:
- 1) 传感器 1 是时钟性能参数最优的时钟节点,作为主时钟,其他传感器是从时钟;
  - 2) 交换机 1 是传感器 1 的最近端时钟节点,用于验证最近端备份算法;
  - 3) 各个时钟的时钟频率晶振偏移既有正向的,又有负向的;
  - 4) 冗余时,传感器 1 和传感器 4 的时钟性能参数相同,且都是最优的,即有 2 个主时钟;
  - 5) 不同主时钟经过延时之后再发送初始时钟同步报文时,传感器 4 的延时为 0.5 ms;
  - 6) 通过代码让主时钟产生故障;
  - 7) 在仿真运行到 1.3 ms 处主时钟发生故障,传感器 1 发生故障;
  - 8) 仿真运行时间 10 ms,即仿真验证 10 个仿真周期的情况.

以上设计均在仿真配置文件中实现,仿真运行过程如图 5 所示.

4.3 结果分析

仿真结果采用主从时钟之间的时钟同步精度  $T_c$  和时钟同步功能中断时长  $T_a$  两个指标来度量,有



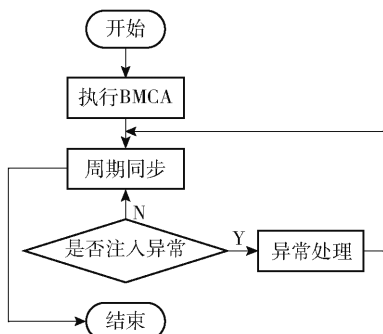


图5 仿真运行过程

$$T_c = t_m - t_l \quad (8)$$

其中  $t_m$  为主时钟的时钟值.

$$T_a = t_r - t_a \quad (9)$$

其中:  $t_r$  为时钟同步功能恢复的时钟值,  $t_a$  为主时钟发生故障的时钟值.

选取传感器9的一个从时钟进行分析,其他从时钟结果类似.

主时钟发生故障后,若不采取措施,则网络失去时钟同步功能.若采用重选机制,从时钟检测到主时钟异常后,重新选出一个主时钟,一段时间以后,时钟同步功能恢复.仿真结果如图6所示.

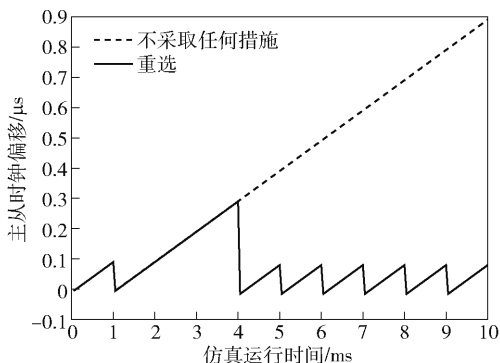


图6 不采取措施和主时钟重选

从图6可以看到,主时钟发生故障后,若无恢复措施,网络则会失去时钟同步功能,导致网络各个从时钟的时钟偏移值不断增大,满足不了时间敏感报文的传输需求,可能会使网络功能异常.当采用主时钟重选机制以后,从时钟可以在经过  $T_a$  时间以后恢复时钟的同步功能,而使  $T_c$  稳定在一个较为理想的范围内.

仿真结果表明,采用重选机制可以在一段时间后让时钟同步功能恢复正常,但是存在一定的中断时间.

重选机制的中断时间主要由从时钟检测到主时

钟异常所需要的时间和重新选出主时钟所需要的时间2部分组成,因此需要减少这2部分时间.笔者设计的最近端备份的算法可以将主时钟重选的时间降低为0,从而减少时钟同步功能的中断时长,进一步提高时钟同步的可靠性.重选机制、最近端备份算法的仿真性能对比如图7所示.

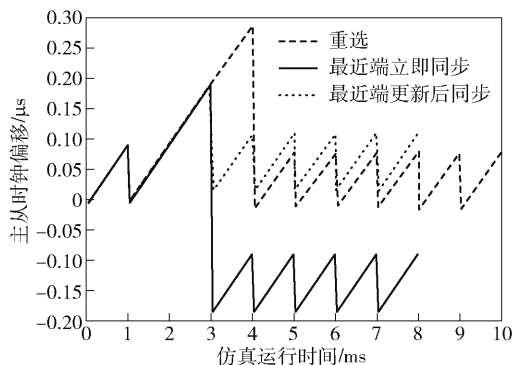


图7 最近端节点作为主时钟

从图7可以看到,当主时钟发生故障后,各个从时钟的偏移值会越来越大.以最近端节点作为主时钟的算法,可以较快地恢复时钟同步功能,其中断时长小于重选机制.当最近端节点不更新自身参数,直接作为主时钟进行时钟同步时,由于最近端节点本身存在时钟偏移和时钟频率偏移,所以从时钟在最近端节点的偏移基础上进行时钟同步,从而引入了一个固定偏移值.当最近端节点先根据上一次同步成功的信息更新本地信息,再发起时钟同步时,  $T_c$  值稳定在一个较为理想的范围内.

仿真结果表明,以最近端节点作为主时钟,先调整自身的本地时钟和时钟频率,再发送同步报文之后,可以在一段时间后恢复时钟同步功能,且减少了中断时间.

冗余机制可以进一步减少时钟同步功能的中断时长,它可以将从时钟检测到主时钟异常所需要的时间和重新选出主时钟所需要的时间都降低为0,即  $T_a = 0$ ,实现主时钟的无缝切换,从而提供高可靠性的时钟同步功能.在冗余机制的基础上,所提出的不同主时钟按照不同的初始相位发起时钟同步功能的算法,能够进一步优化冗余机制.冗余机制在无主时钟发生故障情况下的仿真结果如图8所示.

从图8可以看到,采用主时钟冗余机制,在主时钟无异常时,若2个主时钟(传感器1和传感器4)的初始相位相同,时钟同步周期也相同,那么从时钟

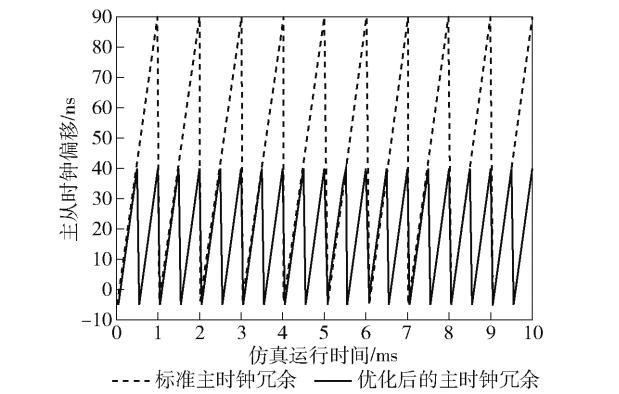


图 8 无故障时的冗余机制

会同时接收到 2 条同步消息,同步精度与单个主时钟一样. 当把某个主时钟(传感器 4)的初始相位延迟半个时钟同步周期(0.5 ms)后,从时钟每半个时钟同步周期即可与主时钟进行一次同步,相当于时钟同步频率提高了一倍,相应地  $T_c$  也能提高一倍.

仿真结果表明,通过采用初始相位移位算法对标准主时钟热备冗余机制进行优化,可以将时钟同步的精度提升近一倍.

采用冗余机制,当某个主时钟(传感器 1)发生故障以后,由于另一个主时钟仍在正常工作,故从时钟仍然可以与主时钟进行校时,时钟同步功能不受影响,能够实现主时钟的无缝切换. 冗余机制在主时钟发生故障情况下的仿真结果如图 9 所示.

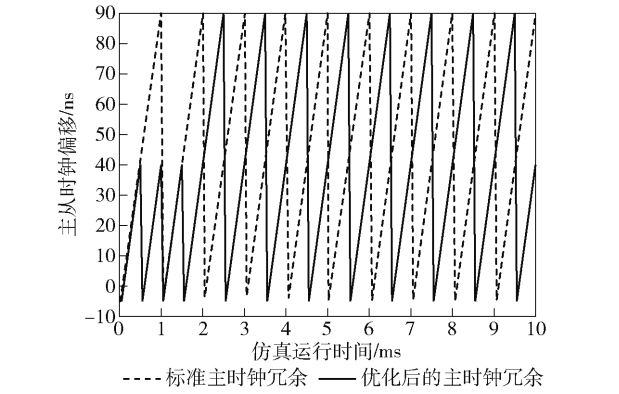


图 9 有故障时的冗余机制

从图 9 可以看到,在主时钟发生故障之前,采用不同初始相位的冗余机制,其同步频率和精度  $T_c$  都较高. 当有某个主时钟发生故障以后,时钟同步功能仍正常,在每个同步周期到达时,没有任何中断,从时钟能够无缝的与主时钟进行同步,提供高可靠的时钟同步功能.

综合分析以上仿真结果,时钟同步可靠性的几

种算法所能达到的主要性能指标对比情况如表 2 所示.

表 2 不同可靠性算法的性能对比

场景	$T_c$ 最大值/ns	$T_c$ 平均值/ns	$T_a$ /ms
未采取措施	不断增长	不断增长	不断增长
重选	305	94.45	2.65
最近端立即同步	285	179.94	1.65
最近端更新同步	205	57.20	1.65
冗余相位偏移	55	32.50	0
标准热备冗余	105	52.50	0

从表 2 可以看到,采用重选机制保障可靠性时,存在时钟同步功能中断时间、可靠性低于最近端备份的方式. 采用最近端节点备份主时钟的算法,仍然存在时钟同步功能中断时间,但是该值小于重选机制,其可靠性比标准重选机制提升了 38%. 冗余机制可以实现主时钟的无缝切换,  $T_a$  为 0, 为网络提供高可靠的时钟同步功能. 从表 2 可见,与标准热备冗余相比,采用初始相位偏移的热备冗余优化算法,可使时钟同步精度的最大值从 105 ns 提升到 55 ns,精度的平均值从 52.5 ns 提升到 32.5 ns,精度的最大值提升约 48%,精度平均值提升约 38%.

5 结束语

针对主时钟重选机制存在时钟同步中断时间的问题,提出了最近端备份的算法,以降低该中断时间,提高时钟同步的可靠性. TSN 工作组提出用冗余机制来保障时钟同步功能的可靠性. 对冗余机制进行优化,提出了初始相位移位的算法. 仿真结果表明,冗余机制可以提供高可靠的时钟同步功能,通过初始相位移位优化算法,可以在保障时钟同步可靠性的基础上提升精度. 下一步将对冗余信息的处理和链路冗余方案作深入研究,为 TSN 在工业互联网和车载以太网中的应用提供参考.

参考文献:

[1] IEEE. IEEE 802.1AS-timing and synchronization-2020 [S]. [S.1.]: IEEE, 2020: 1-78.

[2] Nikolaus K, Andreas P, Thomas K, et al. Performance and reliability aspects of clock synchronization techniques for industrial automation [J]. Proceedings of the IEEE, 2019, 107(6): 1011-1026.

- Networks DRCN 2020. Milan: IEEE, 2020: 1-7.
- [4] Alshorman R. Toward proving the correctness of TCP protocol using CTL[J]. International Arab Journal of Information Technology, 2018, 16(3): 407-414.
- [5] Chen Dawei, Dong Rongsheng, Guo Yunchuan, et al. SPIN model detection of IKEV2 protocol[J]. Computer Engineering, 2006, 5(57): 164-166, 246.
- [6] Yi Huifan, Wan Liang, Huang Nana, et al. Research on attacker modeling method based on SPIN security protocol [J]. Information Network Security, 2018, 2(9): 61-70.
- [7] 张卫杰. 基于反例的安全协议漏洞挖掘[D]. 成都: 电子科技大学, 2019: 26-35.
- [8] Holzmann G J, Lieberman W S. Design and validation of computer protocols [M]. Englewood Cliffs: Prentice Hall, 1991: 135-145.
- [9] Zhong Xiaomei, Xiao Meihua, Li Wei, et al. Formal analysis and improvement of RFID ultra-lightweight authentication protocol RCIA [J]. Computer Engineering and Science, 2018, 40(12): 2183-2192.
- [10] Wu Chang, Xiao Meihua, Luo Min, et al. Efficient and automatic generation of security protocol validation model [J]. Computer Engineering and Applications, 2010 (2): 83-86.

(上接第25页)

- [3] Cena G, Scanzio S, Valenzano A. Experimental evaluation of seamless redundancy applied to industrial Wi-Fi networks[J]. IEEE Trans Ind Informat, 2017, 13(2): 856-865.
- [4] Puhm A, Mahmood A, Bigler T, et al. Synchronizing an IEEE 1588 slave clock over both paths of a redundant Ethernet system[C]//IEEE Int Symp Precis Clock Synchronization Meas, Control, Commun (ISPCS). Sweden: IEEE, 2016: 1-6.
- [5] Maryam P, Balakrishna B, Roman O, et al. Simulation framework for clock synchronization in time sensitive networking [C]//2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC). [S.l.]: IEEE, 2019: 213-220.
- [6] Jiang Junhui, Li Yuting, Hong Seung Ho, et al. A simulation model for time-sensitive networking(TSN) with experimental validation [J]. IEEE Communications Letters, 2019, 18(2): 153-160.
- [7] Henning P, Peter D, Enkhtuvshin J, et al. A simulation model of IEEE 802.1AS gPTP for clock synchronization in OMNeT++ [J]. EPiC Series in Computing, 2018, 56(1): 63-72.
- [8] Falk J, Hellmanns D, Carabelli B, et al. NeSTiNg: simulating IEEE time-sensitive networking (TSN) in OMNeT++ [C]//2019 International Conference on Networked Systems (NetSys). [S.l.]: IEEE, 2019: 1-8.
- [9] Irfan A, Rabbia I, Bilal K, et al. Performance evaluation of IEEE 1588 protocol with modified LibPTP in OMNeT++ [C]//2018 Annual IEEE International Systems Conference (SysCon). [S.l.]: IEEE, 2018: 1-5.
- [10] Lim H-T, Herrscher D, Völker L, et al. IEEE 802.1 as time synchronization in a switched Ethernet based in-car network [C]// Vehicular Networking Conference (VNC). [S.l.]: IEEE, 2011: 147-154.