

基于贪心思想的调度表优化算法设计

方 挺,王 勇,褚文奎,谭小虎

(空军工程大学 航空工程学院,西安 710038)

摘 要: 时间触发以太网使用时间触发速率单调调度(TT-RMS)算法生成通信调度表时,存在 RC 消息不能在最大时延内到达、TT 消息在错过时间窗后不能到达目的终端的问题。为此,设计一种基于贪心思想的调度表优化算法。通过规划新路径、分包与合包、更改 TT 消息发送时间 3 种方式对通信消息进行改进,并构造一种代价函数评价每次生成的调度表,多次迭代直至代价值稳定后得到最终结果。仿真结果表明,相比改进的 TT-RMS 算法,该算法能够提高 RC 消息实时性,降低 TT 消息平均时延百分比。

关键词: 时间触发以太网; 速率单调调度; 贪心思想; 平均时延; 调度表

中文引用格式: 方 挺,王 勇,褚文奎,等. 基于贪心思想的调度表优化算法设计[J]. 计算机工程,2018,44(9):280-285.

英文引用格式: FANG Ting, WANG Yong, CHU Wenkui, et al. Design of scheduling table optimization algorithm based on greedy thought[J]. Computer Engineering, 2018, 44(9):280-285.

Design of Scheduling Table Optimization Algorithm Based on Greedy Thought

FANG Ting, WANG Yong, CHU Wenkui, TAN Xiaohu

(College of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China)

【Abstract】 When Time-Triggered Ethernet(TTE) use Time-Triggered Rate Monotonic Scheduling(TT-RMS) algorithm to generate communication schedules, there is a problem that RC messages cannot arrive at the destination terminal within the maximum delay, and TT messages cannot reach the destination terminal after missing the time window. To solve this problem, a scheduling algorithm based on greedy thought is designed. By planning new paths, subpackages and packages, and changing TT message sending time, the communication messages are improved, and a cost function is constructed to evaluate the scheduling table that is generated each time, and the final result is obtained for multiple iterations until the value is stable. Simulation results show that, compared with the improved TT-RMS algorithm, the proposed algorithm can improve the real-time performance of RC messages and reduce the average delay percentage of TT messages.

【Key words】 Time-Triggered Ethernet(TTE); rate monotonic scheduling; greedy thought; average delay; scheduling table

DOI:10.19678/j.issn.1000-3428.0048623

0 概述

在交换式网络中,时间触发以太网(Time-Triggered Ethernet, TTE)可以实现亚微秒级的分布式时钟同步。TTE 网络具有强实时性这一特点为时间触发通信的调度创造了条件,使其能够满足工业应用的需求。TTE 是对 AFDX 网络的发展^[1],因此,其有机会成为下一代航电系统的通信基础。

为满足实时性和安全性的需求, TTE 网络将消息划分为 3 种类型^[2]: 时间触发(Time-Triggered, TT)消息, 速率受限(Rate-Constrained, RC)消息和尽力传(Best-Efforts, BE)消息。TT 消息用于对抖动、延迟等确定性要求较高的情况; RC 消息用于对确定

性要求较低的情况,其与 AFDX 网络中虚拟链路的传输方式兼容; BE 消息是传统的以太网通信消息,优先级较低。

目前,对 TTE 网络的研究焦点主要集中在网络配置的优化和 TT 调度表的离线设计。文献[3]根据一种分区的调度模式将基本周期分成专门发送 TT 消息和专门发送 RC 消息两段。文献[4]通过使用可满足性模理论(Satisfiability Model Theory, SMT)解析器,设计一种 TT 调度表生成方法,其使用设定的约束条件求得可行解。文献[5]基于禁忌搜索离线设计 TT 调度表,并对 RC 消息时序关系进行分析。文献[6]针对分区调度表设计一种基于时间窗检测的 TT 消息和 RC 消息转换策略,能够增强

基金项目: 航空科学基金(20165515001)。

作者简介: 方 挺(1993—),男,硕士研究生,主研方向为航电网络;王 勇,副教授;褚文奎,讲师;谭小虎,硕士研究生。

收稿日期: 2017-09-11 **修回日期:** 2017-11-21 **E-mail:** 17792028706@163.com

TT 消息传输的健壮性。此外,文献[7]提出一种基于速率单调调度机制的离线消息调度算法——时间触发速率单调调度(Time-Triggered Rate Monotonic Scheduling, TT-RMS)算法,对调度策略进行了优化。文献[8]对 TT-RMS 算法进行改进,设计一种支持非调和周期的通信消息。

本文首先分析由改进 TT-RMS 算法生成的调度表,阐述其存在的问题,并对可优化的方法分别进行描述,总结这些方法的优点与缺点;然后针对所提问题建立代价函数并设计优化算法——基于贪心思想的调度表优化算法;最后进行仿真实验,通过与改进 TT-RMS 算法生成的调度表的对比,验证本文算法的可行性和效率。

1 改进 TT-RMS 算法分析

改进的 TT-RMS 算法被证明能够在复杂的拓扑环境下基于不同长度和周期的消息,生成 TT 消息调度表。与原 TT-RMS 算法相比,改进 TT-RMS 算法继承了前者计算效率高、负载大的优点,但同时也出现了 RC 消息不一定能在最大时延内到达、TT 消息在错过时间窗后不能到达目的终端的问题。

1.1 算法介绍

改进 TT-RMS 调度表生成算法基于如下假设:

- 1) 网络中要调度的 TT 消息全部为周期性消息,且 TT 消息集及消息属性已知;
- 2) 网络中所有通信链路均可以使用,且所有链路的属性都相同;
- 3) 数据帧可以在交换机上停留一段时间,等交换机空闲时再发送;
- 4) 网络中每次发送的 TT 消息都能封装在一个帧里,且在一个时间槽内传输完毕。

在算法生成调度表的过程中,首先计算消息周期的最小公倍数,以此确定集群周期的长度;然后利用速率单调原则对所有 TT 消息进行优先级排序,确定消息调度的顺序;最后根据网络资源的状态调度消息。文献[8]对该算法进行的改进主要体现在:

- 1) 确定了调度表的集群周期长度,支持非调和周期消息的调度;
- 2) 不同长度的消息可以分配不同长度的时间槽;
- 3) 使算法能够应用于多跳网络;
- 4) 能够判断生成的调度表中所有消息是否能够在最大时延内到达目的节点。

1.2 RC 消息传输问题

TTE 网络离线生成静态调度表调度 TT 消息,因此,在调度表生成后就确定了 TT 消息的时延大小。当 TT 消息的时延小于最大时延时,调度表能够满足实时性要求。但是,RC 消息采用动态调度的方式,无论是先来先服务策略还是静态优先级调度策略,都只能确定消息的时延上界,而无法精确确定时延大小。

图 1 所示为影响 RC 消息传输时延的 4 个主要因素^[9],分别为:

1) 用 Q^{TT} 表示的 TT 消息调度延迟。在图 1 中, TT_1 、 TT_2 和 TT_3 表示 TT 消息占用的时间槽。因为 TT 消息的优先级比 RC 消息高,所以在 TT 消息传输时,RC 消息被延迟发送。

2) 用 Q^{RC} 表示的其余 RC 消息调度延迟。由于终端发送 RC 消息时间未确定,交换机里将会产生一个 RC 消息队列。 RC_x 消息只有在 RC_1 消息和 RC_2 消息都发送完毕后才开始传输。

3) 用 Q^{TB} 表示的 TT 消息和 RC 消息调度整合所产生的延迟。因为这些时间段不足以传输 RC 消息队列的第一个消息,所以只能作为空闲的时间槽。

4) 用 Q^{TL} 表示的工程技术上的其他延迟,比如交换机因为需要清理通道而产生的延迟。带宽为 100 Mb/s 的 TTE 网络交换机,延迟通常至少有 4 μ s。

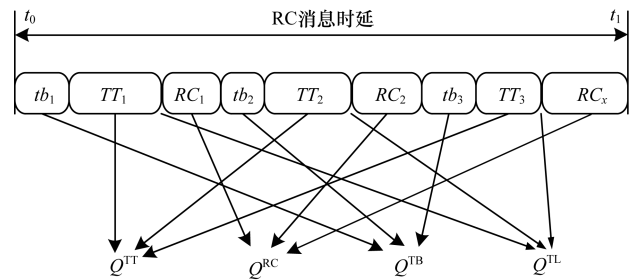


图 1 RC 消息传输时延影响因素

因为以上 4 个因素的影响,在 t_0 时刻产生的 RC_x 消息,到 t_1 时刻才传输完毕。

改进 TT-RMS 调度表生成算法只考虑 TT 消息及其时延需求,而没有考虑 RC 消息。这导致其生成调度表时,部分 RC 消息的时延可能会超出最大时延,从而无法进行调度。本文根据文献[8]中设置的网络拓扑结构实现该改进的 TT-RMS 算法并进行分析。网络拓扑结构如图 2 所示,其中,ES 表示终端,NS 表示交换机,dp 表示链路。

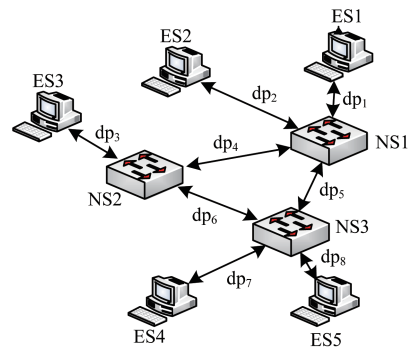


图 2 网络拓扑结构

图 3 所示为一个由改进 TT-RMS 调度表生成算法生成的局部调度表。其中,消息 1~消息 5 是 TT 消息,消息 6、消息 7 是 RC 消息。图 3 中阴影部分是空闲的时间槽,表示 RC 消息的等待时间。

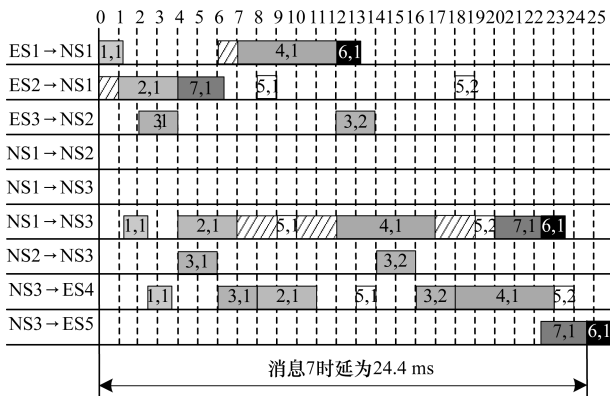


图3 局部消息调度表1

假定消息7的时限为16 ms,下面对图3中消息7的时延情况进行分析。假设ES2在0时刻产生一个RC消息,即消息7。因为消息2是TT消息,已经确定发送时刻,所以消息7想在传输消息2之前发送,只能利用1 ms的空闲时间,但是传输消息7需要2.2 ms,因此,只能在4 ms时刻发送,并在6.2 ms时刻使其到达交换机NS1。在从交换机NS1传到交换机NS3的过程中,消息7在20 ms时刻发送,并在22.2 ms时刻存储在交换机NS1里。而接下来的3个空闲时间槽均无法传输消息7,因此,只能从20 ms时刻传输。消息7在22.2 ms时刻完全到达交换机NS3,并立即进行下一链路的传输。最终,消息7在24.4 ms时刻完成整个传输过程,延迟为24.4 ms,大于最大延迟16 ms,不能满足实时性的要求。因此,消息7将被抛弃。

1.3 TT消息传输问题

根据SAE AS6802标准,TTE网络离线生成TT消息调度表,能够保证TT消息传输时的确定性和实时性。但在工程实践中,受到晶振不稳定或其他因素的影响,不能一直保持理想的同步状态,可能会发生消息冲突的情况。在不采用消息转换策略的网络中,交换机会在TT消息错过时间窗时将其丢弃。如果此时消息是关键性的任务流量,就会严重影响航电网络的实时性和确定性。在采用消息转换策略的网络中,交换机在TT消息错过时间窗时会将其转换成RC消息而继续传输。因此,其网络健壮性可以大幅提高。

使用消息转换策略的前提是预留足够的时间,但是改进TT-RMS调度表生成算法在生成TT消息调度表时,没有预留时间,以至于TT消息的平均传输时延较大,甚至部分消息会达到最大时延。这样不利于将TT消息转换成RC消息,即无法满足网络的健壮性要求。

如图4所示,消息3在调度表中的延迟已经等于最大延迟。如果其错过时间窗,将无法获得足够的时间来转换为RC消息进行传输,而只能被丢弃。

针对该情况,只要在设计调度表时降低消息3的时延,就可以完成消息类型的转换,继而完成传输。因此,在设计调度表时,预留出足够的消息类型转换时间很有必要。

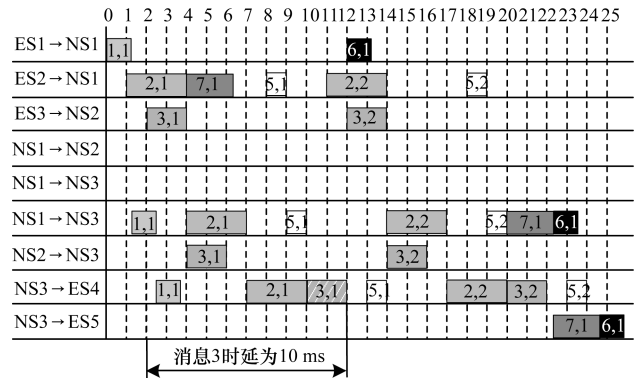


图4 局部消息调度表2

2 基于贪心思想的调度表优化算法

2.1 调度表优化策略

针对改进TT-RMS算法存在的问题,本文采用3种优化策略对调度表进行优化:规划新的路径^[10-11],分包与合包^[12-13],更改TT消息在调度表中的调度时刻^[14]。

2.1.1 新传输路径规划

改进TT-RMS调度表生成算法为消息选择的路径都是最短路径。这种选择方式会导致网络中的部分路径变成关键路径,从而降低调度表生成的成功率。此外,由于这些路径负载率过高,消息等待的时间可能会延长。如果更改一些传输路径,即减少关键路径,可以减少等待时间,降低消息时延。

在图2的网络拓扑结构中,因为终端ES1、ES2、ES4、ES5发送和接收的消息较多,选择最短路径作为传输路径时,会使链路dp₅成为关键路径。在这条路径上经过的消息较多,会提高调度表的生成难度。如果把部分消息路径改成链路dp₄和链路dp₆,会减少消息时延同时降低调度表的生成难度。

此外,对于RC消息,重新规划路径这个措施也同样有效,其可以避免关键路径的生成,同时提高消息的实时性,减少网络阻塞。本文消息的路径规划使用负载均衡算法^[15],对于含有N条物理链路的网络,设每条链路的负载率为U_i,则平均负载率为:

$$\bar{U} = (\sum_{i=1}^N U_i) / N \tag{1}$$

负载率均方差为:

$$\delta_U = \sqrt{(\sum_{i=1}^N (U_i - \bar{U})^2) / N} \tag{2}$$

负载均衡算法能够使δ_U达到最小。由文献[15]可知,负载均衡算法的时间复杂度与最短路径算法相同,均为O(n³)。

2.1.2 分包与合包

分包与合包方法包括2种:1)将一个较长的消息封装在几个较短的数据帧里;2)将几个长度较短、路径和周期相同的消息封装在一个数据帧里。

对于TT和RC消息而言,把一个较长消息封装在几个较短数据帧里的方法均可行。如果将TT消息封装在较短的数据帧里,可以显著减少链路被占用的情况,从而生成密集网孔状的调度表。RC消息能利用这些时间空隙传输,而不会出现TT消息持续占用链路导致RC消息传输超过最大时延的情况。如果将RC消息封装在较短的数据帧里,就可以使用那些较短的时间空隙传输,这样同样提高了RC消息的实时性。

需要注意的是,分包会增加数据帧的数量,造成帧头这类无效信息的重复传输,从而导致网络负载的增加,因此,在明确可以达到优化调度表的目的时才使用分包;合包的消息路径和周期应当相同,否则无法实现。仿真时可以设置2个阈值 a 和 b ,若消息长度小于 a ,则将该消息放入合包队列,大于 b 则将该消息放入分包队列。为简化计算,仿真时分包数和合包数均不超过2个。

2.1.3 TT消息发送时间更改

更改TT消息发送时间这种优化方法的操作较简单,但是其会直接影响TT消息的时延大小,且会对整个调度表的结构进行调整。该方法的优点是增加负载,但是更改后的调度表不一定有利于所有消息的传输。因此,其优化效果不够理想,本文在进行仿真时不采用该策略。

2.2 代价函数建立

本文优化算法的目标是使TT消息和RC消息的时延上界都达到最小,为此构造如式(3)所示的代价函数:

$$cost = w_{TT} \times \delta_{TT} + w_{RC} \times \delta_{RC} \quad (3)$$

其中, $cost$ 是总代价值, δ_{RC} 和 δ_{TT} 分别为RC消息和TT消息的时延评估值, w_{RC} 与 w_{TT} 分别为RC消息和TT消息的权重值。如果有TT消息或RC消息不可调度,则给这个消息设置一个极大值,例如1000。如果所有的消息均能调度,且都可以在截止期内到达目的终端,就对 w_{TT} 和 w_{RC} 设置较小的权值。在本文算法进行仿真时,将 w_{TT} 设置为1, w_{RC} 设置为2,该设置是为突出对RC消息的优化效果,因为TT消息侧重于时间确定性,而RC消息更注重消息实时性。 δ_{RC} 和 δ_{TT} 的计算公式如式(4)和式(5)所示。

$$\delta_{RC} = \begin{cases} c_1 = \sum_j \max\left(0, \frac{R_j - RC_j^{deadline}}{RC_j^{deadline}}\right), c_1 > 0 \\ c_2 = \sum_j \left(\frac{R_j - RC_j^{deadline}}{RC_j^{deadline}}\right), c_1 = 0 \end{cases} \quad (4)$$

$$\delta_{TT} = \begin{cases} c_1 = \sum_i \max\left(0, \frac{T_i - TT_i^{deadline}}{TT_i^{deadline}}\right), c_1 > 0 \\ c_2 = \sum_i \left(\frac{T_i - TT_i^{deadline}}{TT_i^{deadline}}\right), c_1 = 0 \end{cases} \quad (5)$$

其中, R_j 表示RC消息的时延上界, $RC_j^{deadline}$ 表示RC消息的截止期,截止期即最大允许时延, T_i 表示TT消息的时延上界, $TT_i^{deadline}$ 表示TT消息的截止期。

如果TT或RC消息不可调度,就说明在截止期内消息无法到达目的终端,即 $T_i - TT_i^{deadline} > 0$ 或 $R_j - RC_j^{deadline} > 0$,同时 $c_1 > 0$ 。当两类消息都可以调度时,则 $T_i - TT_i^{deadline}$ 和 $R_j - RC_j^{deadline}$ 都小于0,同时 $c_1 = 0$,此时将 c_2 作为消息时延的评价值。使用该方法可以有效地将不可调度消息与可调度消息区分开。

2.3 算法步骤

本文优化算法的关键步骤是基于贪心算法的设计思想求解出局部最优值,并不断迭代,最终完成调度表的设计。算法具体步骤如下:

步骤1 执行初始化操作,输入网络拓扑结构、消息集及其属性等信息。

步骤2 使用改进TT-RMS调度表生成算法得到一个初始解。在生成的调度表中,不是所有TT消息都可调度,对于不可调度的TT消息,设置一个极大的权重值。计算调度表中TT消息和RC消息的时延上界。RC消息的时延上界计算方式采用文献[16]提出的端到端最大时延分析方法。

步骤3 基于本文建立的代价函数计算初始解的代价值。

步骤4 根据消息的时延与截止期的百分比(以下简称时延百分比)进行排序,按照从大到小的顺序生成一个消息队列,即时延队列。为不可调度消息的时延百分比设置一个极大值,并将该消息置于时延队列顶端。

步骤5 消息优先进行分包与合包操作,不满足条件的对其重新规划路径。

步骤6 计算使用不同优化方法优化后的解的代价值。若优化后的代价值没有减小,则将该消息放入一个新的已优化的队列,并在原时延队列中删除该消息。否则,重新计算该消息的时延百分比,并再次插入到时延队列中。

步骤7 在循环次数未达到设定值或消息有改变时,若时延队列没有消息,就把已优化队列中的消息复制进时延队列中,返回步骤4;当时延队列中的消息数量为0时,将已优化消息队列中的消息复制到时延队列中,并继续执行步骤4。

步骤8 在循环次数达到设定值或消息优化后没有改变时,停止循环,给出优化后的调度表。

算法伪代码表示如下:

```

输入初始解;
do
{
for(遍历消息集)
{计算消息时延;}
排序消息集并生成时延队列;
计算代价值;
for(遍历消息集)
{
if(消息长度 > b)
消息分包;
else if(消息长度 < a)
消息合包;
else 重新规划路径;
if(代价值减小)
该消息放入时延队列;
else
该消息放入已优化队列;
}
}
while(循环次数未达到设定值或解没有变化)

```

3 实验与结果分析

本文实验所用计算机配置为 Intel(R) Core(TM) i5 处理器,处理器频率为 3.0 GHz,内存大小为 4 GB。使用仿真软件 OPNET Modeler 14.5 建立交换式网络模型,网络中各链路的带宽设置为 100 Mb/s。

3.1 消息设置

由于本文算法未考虑 BE 消息,所以生成的消息集中不存在 BE 消息。TT 消息的配置采用文献[7]的配置方式,并增加部分 RC 消息。因为静态优先级调度算法能够保证消息的实时性,所以本文中 RC 消息采用静态优先级调度策略进行调度。RC 消息配置如表 1 所示。其中,BAG 表示宽带分配间隔。

表 1 RC 消息配置

RC 消息 序号	BAG /ms	截止期 /ms	长度 /Byte	传输 时间/ms	源终端	目的 终端
1	8	8	183	0.5	ES1	ES4
2	16	16	433	0.5	ES2	ES4
3	16	16	433	1.0	ES3	ES4
4	32	32	183	0.5	ES1	ES5
5	16	16	433	1.0	ES2	ES5
6	32	32	183	0.5	ES1	ES2
7	64	64	433	1.0	ES2	ES3

3.2 消息可调度性

表 2 所示为各消息集中可调度消息占比结果。由表 2 可知,与改进 TT-RMS 算法相比,本文算法消息调度表的可调度性大幅提高,有更多的 TT 消息和

RC 消息被调度。原因是改进 TT-RMS 算法消息调度表没有考虑 RC 消息,导致 TT 消息占用传输链路的持续时间较长,增加了部分 RC 消息的时延上界,部分优先级较低的 RC 消息处于不可调度的状态,从而降低了消息的整体可调度性。

表 2 2 种算法调度表的可调度消息占比情况 %

消息集序号	本文优化算法	改进 TT-RMS 算法
1	100.00	71.01
2	100.00	62.90
3	97.27	58.59
4	100.00	63.16
5	100.00	58.17
6	98.91	59.06
7	100.00	53.97
8	97.70	50.57
9	97.37	46.38
10	100.00	47.90
11	98.92	45.95
12	97.91	43.60

3.3 消息平均时延

消息的时延情况是评价调度表设计效果的重要指标。表 3 所示为 2 种算法中 RC 消息的平均时延上界情况。由表 3 可知,相比改进 TT-RMS 算法,本文算法 RC 消息的平均时延上界及其百分比都有所降低,且时延均在各自的截止期之内。因此,本文算法可以解决部分 RC 消息不能在最大时延内到达目的终端的问题。

表 3 2 种算法 RC 消息时延上界情况

RC 消息 序号	平均时延上界		平均时延上界百分比/%	
	本文优 化算法	改进 TT- RMS 算法	本文优 化算法	改进 TT- RMS 算法
1	3.1	29.2	76.42	92.17
2	3.2	22.6	64.74	93.17
3	3.4	29.0	78.60	98.42
4	3.3	16.8	60.21	99.24
5	4.0	17.2	76.12	97.33
6	4.1	23.8	61.75	93.87
7	4.0	12.5	66.35	95.51
8	4.3	18.3	79.29	99.34
9	4.6	27.9	75.02	94.69
10	3.3	29.5	72.26	95.46
11	3.5	24.3	78.94	98.45
12	4.1	20.5	75.97	97.39

表 4 所示为 2 种算法中 TT 消息平均时延和平均时延百分比情况。由表 4 可知,相比改进 TT-RMS 算法,本文算法 TT 消息平均时延及其百分比都有较大幅度的降低。该结果表明,本文算法能够

优化整体消息集,提高 TT 消息的实时性,保证在出现消息冲突问题时消息类型的转换具备充足的时间,即能够增强网络的健壮性。

表 4 2 种算法 TT 消息时延情况

TT 消息序号	平均时延		平均时延百分比/%	
	本文优化算法	改进 TT-RMS 算法	本文优化算法	改进 TT-RMS 算法
1	9.51	15.62	67.75	83.00
2	18.19	51.83	69.58	97.97
3	38.66	77.91	64.91	100.00
4	15.63	49.78	54.42	88.29
5	28.46	63.05	56.10	93.11
6	32.35	52.36	66.18	99.35
7	19.32	46.92	50.57	78.96
8	35.47	83.71	60.04	86.60
9	48.35	160.45	59.67	100.00
10	11.83	17.58	51.84	92.49
11	27.80	47.58	66.98	98.64
12	47.48	105.20	62.61	100.00

4 结束语

TTE 网络的实时性与健壮性均较低。为此,本文提出一种基于贪心思想的调度表优化算法。相比改进的 TT-RMS 算法,该算法解决了调度表中 RC 消息的时延上界大于其最大时延、TT 消息在错过时间窗后没有足够时间转换消息类型以继续传输的问题。仿真结果表明,该算法能够有效优化消息调度表,降低 RC 消息的平均时延上界和 TT 消息的平均时延值,提高消息调度表的合理性,增强网络健壮性。基于贪心思想的算法能够保证局部最优,但最后结果并不一定最优,因此,今后将考虑使用智能算法以提高获得最优结果的概率。

参考文献

- [1] 焦文喆,翟正军,王国庆. 时间触发 AFDX 调度设计及实时性分析[J]. 计算机工程,2016,42(7):42-48.
- [2] STEINER W, BAUER G, HALL B, et al. TTEthernet dataflow concept [C]//Proceedings of the 8th IEEE International Symposium on Network Computing and Applications. Washington D. C., USA: IEEE Press, 2009: 319-322.
- [3] 刘晚春,李峭,何峰,等. 时间触发以太网同步及调度机制的研究[J]. 航空计算技术,2011,41(4): 122-127.

- [4] STEINER W. Synthesis of static communication schedules for mixed-criticality systems [C]//Proceedings of the 14th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing. Washington D. C., USA: IEEE Press, 2011: 11-18.
- [5] TAMAS-SELICEAN D. Design of mixed-criticality applications on distributed real-time systems [D]. Copenhagen, Denmark: Technical University of Denmark, 2014.
- [6] 易娟,熊华钢,何峰,等. TTE 网络流量转换策略及其延时性能保障调度算法研究[J]. 航空学报, 2014,35(4):1071-1078.
- [7] 徐晓飞,曹晨,郭骏,等. TT-RMS: 时间触发网络通信表生成算法[J]. 北京航空航天大学学报, 2015, 41(8):1403-1408.
- [8] 张超,南建国,褚文奎,等. 改进的 TT-RMS 调度表生成算法[J]. 空军工程大学学报(自然科学版), 2016,17(6):82-87.
- [9] KOPETZ H, GRUNSTEIDL G. The time-triggered Ethernet (TTE) design [C]//Proceedings of the 8th IEEE International Symposium on Object Oriented Real-time Distributed Computing. Washington D. C., USA: IEEE Press, 2005: 22-33.
- [10] 欧海斌,尹艳,熊建新. 空基自组网中基于负载均衡的路由策略[J]. 计算机工程,2011,37(14):79-81.
- [11] SAKET R, NAVET N. Frame packing algorithms for automotive applications[J]. Journal of Embedded Computing, 2006,2(1):93-102.
- [12] WANG B, HOU J. Multicast routing and its QoS extension: problems, algorithms, and protocols [J]. IEEE Network, 2000,14(1):22-36.
- [13] TAMAS-SELICEAN D, POP P, STEINER W. Synthesis of communication schedules for TTEthernet-based mixed-criticality systems [C]//Proceedings of International Conference on Hardware/Software Codesign and System. New York, USA: ACM Press, 2012: 473-482.
- [14] STEIN S R, FILLER R L. Kalman filter analysis for real time applications of clocks and oscillators [C]// Proceedings of the 42nd Annual Frequency Control Symposium. Washington D. C., USA: IEEE Press, 1988: 447-452.
- [15] WANG Z, CROWCROFT J. Quality-of-service routing for supporting multimedia applications [J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228-1234.
- [16] TAMAS-SELICEAN D, POP P, STEINER W. Timing analysis of rate constrained traffic for the TTEthernet communication protocol [C]//Proceedings of IEEE International Symposium on Real-time Distributed Computing. Washington D. C., USA: IEEE Press, 2015: 119-126.

编辑 吴云芳