Computer Science and Application 计算机科学与应用, 2018, 8(9), 1447-1451
Published Online September 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.89158

Hans汉斯

Research on the Dynamic Priority Scheduling Algorithm of CAN Bus in Application-Layer

Bowen Wang

China Coal Science and Engineering Group Chongqing Research Institute Co., Ltd., Chongqing
Email: 29681717@qq.com

Received: Sep. 7th, 2018; accepted: Sep. 22nd, 2018; published: Sep. 29th, 2018

Abstract

With the continuous development of automation technology, traditional CAN bus scheduling non-destructive bus arbitration technology can't have satisfied social industrial information field in real-time. As the complicated network structure and the load of network increasing, the traditional CAN bus arbitration technology can't guarantee the node in real-time. After the comprehensive comparison based on CAN bus technology, I put forward a based message dynamic priority scheduling algorithm which is proved high theoretic value and can effectively solve the problem of network in real-time and validity under complex conditions.

Keywords

CAN Bus, Bus Scheduling, Bus Arbitration

CAN总线在应用层实现动态优先级调度算法研究

王博文

中煤科工集团重庆研究院有限公司，重庆
Email: 29681717@qq.com

收稿日期：2018年9月7日；录用日期：2018年9月22日；发布日期：2018年9月29日

摘　要

随着自动化技术的不断发展，传统的CAN总线调度非破坏性总线仲裁技术已经不能满足社会工业信息领

文章引用: 王博文. CAN 总线在应用层实现动态优先级调度算法研究[J]. 计算机科学与应用, 2018, 8(9): 1447-1451.
DOI: 10.12677/csa.2018.89158

域的实时性要求。随着网络结构的复杂化，网络负载的不断加大，传统的CAN总线仲裁技术不能保证网络节点的实时性。本文在综合比较了CAN总线总裁技术的基础上，提出了一种基于CAN报文动态优先级调度的算法，实践证明其具有交高的理论价值，能够有效的解决网络复杂条件下的实时性和有效性问题。

关键词

CAN总线，总线调度，总线仲裁

1. 引言

控制器局域网 CAN (Controller Area Network)是目前被批准为国际标准的少数现场总线之一。它是一种有效支持分布式实时控制的串行通信网络。具有实时性强，可靠性高，连接方便，性价比高等特点。其在工业过程控制、机械工业、纺织工业、农用机械等各个领域被广泛的应用。但近年来，随着信息社会的不断发展，CAN 应用领域的不断扩展和控制网络的不断复杂化，传统的物理层和数据链路层 CAN2.0 B 协议出现了一些问题，主要表现在当处于 CAN 网络上的节点同时向总线发送数据时，高优先级的节点会在竞争中胜出，获得总线的使用权，而低优先级的节点会被屏蔽。

网络的优先级较低的站点在多次发送数据时与优先级高的站点冲突，并且在竞争中失败而不能发送数据。这样就会导致优先级较低站点数据传输延时的不确定性，时而较大，时而较小，整个网络的实时性和可靠性会大大下降。

2. CAN 总线的调度算法分析

CAN 总线利用数据帧中的标识符表示信息的来源和优先级。标识符既可以静态设定，也可以动态设定，即利用标识符可以实现静态或者动态的信息优先级。在满足可调度性的实时调度算法中，目前常用的是优先级驱动的实时调度算法，它可分为静态优先级调度算法和动态优先级调度算法[1]。

以上两种调度算法是分别是基于主从式的，和分布式的。基于主从式的调度算法是基于主从式数据结构，利用分时调度原理来避免节点数据的冲突。而基于分布式的调度算法则是在网络分布式结构的基础上，处于 CAN 总线上的各个节点根据传输的需要自行改变自己的报文优先级，从而达到网络实时性的需求。

2.1. 基于 TTCAN 技术的时间触发调度算法

时间触发 TTCAN 总线调度算法充分利用了 TDMA 的时分复用技术，时间触发 CAN 总线上的每个节点都占有一个时隙，节点在时隙内拥有总线的使用权[2]。

每个节点在总线周期内至少有一个时隙。这样每个总线周期内，每个节点都有访问总线的机会。从而不会造成优先级低的节点因为多次竞争失败而不能发送数据。

TTCAN 由时间进程驱动，其时间触发调度由顺序固定的时间窗组成。时间窗是用于交换报文的时间片断，通常有三类时间窗：专用时间窗(特定的周期性报文)、仲裁时间窗(通过仲裁访问总线的报文)和空闲时间窗(为总线扩展所保留)。专用时间窗类似于 TDMA(时分多路访问)，属于离线进行的静态调度，所

有流程和时间参数均需要预先指定，并可以在多级或多个 TTCAN 网络内实现同步[3]。TTCAN 的全局时间由时间主机周期发送的参考报文产生，它的总线最多可以配置 8 个具有优先级的时间主机节点，以确保总线的连续、确定性通信，优先级最高的时间主机为当前时间主机。总体来说 TTCAN 的总线调度过程如下图 1 所示。

2.2. 基于分布式的动态优先级调度

由于传统的物理层和数据链路层 CAN2.0B 协议已经不能满足复杂条件下，总线实时性的需求，所以我们为了让处于总线上的节点有效的收发数据，需要在应用层实现调度算法。而 CAN 总线中以数据帧为例如下图 2，将仲裁场的标识符分为优先级块和识别块二部分[4]。优先级部分已经不再具有协议帧的标识功能，而只是表示协议帧的优先级功能，它可以根据实际情况被系统调整。

由于 11 位 CAN ID 标识的 ID 数量有限，所以以 29 位 CAN ID 为例进行说明。

如表 1 所示，优先级块和识别块的位数分别是 n、m。用集合表示为 A_n、B_m，那么 A_n 和 B_m 中的元素个数分别是 2^n 和 2^m。n = m 的情况下，所有的识别块都有对应的优先级，但总的标识符为 29 位奇数个数，所以必须 n < m，取 v = n，集合 Cv∈Bm。所以对于系统来说未使用的报文数量应该为 $2^m - 2^v$。

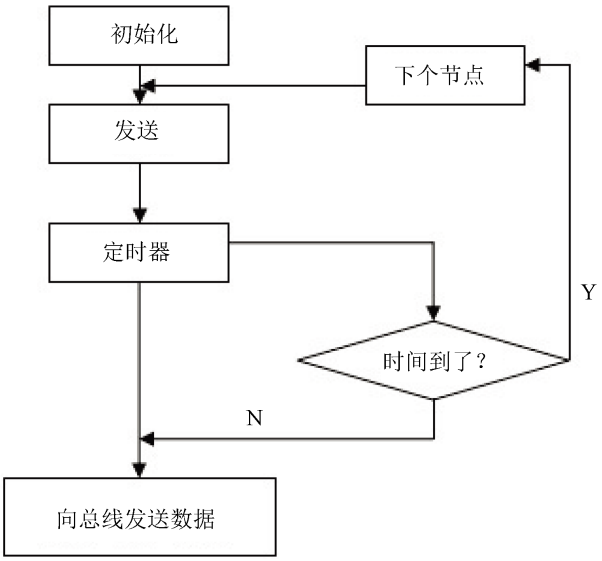

Figure 1. TTCAN Bus scheduling flow chart
图 1. TTCAN 总线调度流程图

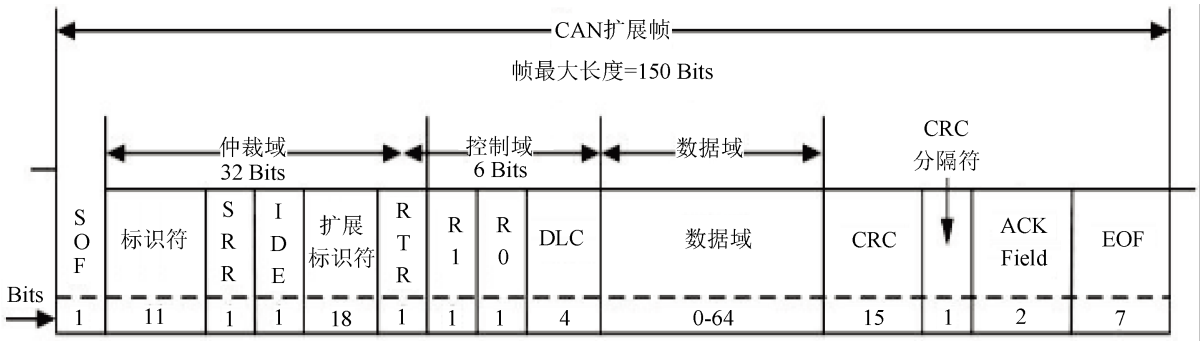

Figure 2. CAN Extended frame structure diagram
图 2. CAN 扩展帧结构图

Table 1. 29 bit CAN ID identifier partition
表 1. 29 位 CAN ID 标识符划分

名称	优先级块	识别块
bit	n	m

处于 CAN 总线上的节点向总线发送报文时，所有的待发送报文被放在一个队列 Q 内，假设单帧报文的发送时间为 t_F (包括报文准备发送时间和报文在总线的传输时间以及等待传输完成时间)，当前报文在队列中的位置为 P。

$$P \le 2^n - 1 \tag{1}$$

对于总数为 2^n-1 的报文，假设对应每个 ID 的报文在队列 Q 中的个数为 D_d，那么任一报文所需要发送至总线需要的等待时间为：

$$T_w = t_f \times \sum_{d=1}^{P-1} D_d \tag{2}$$

公式(2)是固定优先级调度模式下，队列 Q 中报文的等待时间，可以看出队列中当前报文之前报文的数量越多，所等待的时间就越长。队列 Q 尾报文发送所等待的时间最长，大约为：

$$T_w = t_f \times \sum_{d=1}^{2^n-2} D_d \tag{3}$$

可以看出 n 和 D_d 越大，队列 Q 中的报文越多，队尾报文的等待发送时间就越长，在网络拥堵的情况下，系统的实时性和可靠性将大幅度下降。

为了保障系统的可靠性和实时性，提高报文的传输效率，就必须有效控制报文的等待时间 T。除了优化系统的效率以及网络的传输效率外，可以调节的参数就是报文在队列 Q 中的位置 P。位置 P 是报文在队列中的优先级，这里引入动态优先级调度通过不断改变队列的位置 P 来实现报文的实时传输。

$$\begin{aligned} \eta &= \{D_d\} \\ P &= 2^n - 1 - f(t_\lambda, t_f, \eta) \end{aligned} \tag{4}$$

函数 $f(t_\lambda, t_f, \eta)$ 包含了当前报文的最长响应时间 t_λ，每帧报文的传输时间 t_f，以及队列 Q 中报文的个数集合 η。

初始条件下，每个节点的优先级各不相同。在没有发生冲突的情况下，各个节点按初始固定优先级(优先级块的集合为 A_n)完成数据的发送。

当发生冲突后，总线对节点的发送进行仲裁，优先级高的节点在竞争中胜出发送数据。为了保证当前报文传输的可靠性，为了能让竞争失败的节点在下次的竞争中有更大的概率胜出，可以把竞争失败节点的优先级提高一个等级来参加下一次的竞争。

优先级晋升的原则是保证当前节点的最长响应时间大于其在队列 Q 中的等待时间 t_w (即 $t_\lambda > t_w$)只有这样该报文才会及时发送至总线，保证时效。

最坏的情况，网络最大的负荷情况下，CAN 总线下所有的节点处于一种拥堵的状态，这时候的优先级调度的算法如下：

① 已知总线上各报文的最长响应时间，找出响应时间最小的报文 P_1,其最长响应时间为 t_1，此时 $t_1 = \min\{t_\lambda\}$，$t_\lambda \in \{1, 2^n\text{-}1\}$，$S_0 = W_1$。

② 找出总线上响应时间次小的报文 P_2,此时自由时间 $S_1 = T_2 - D_1 \times t_f$。若 $S_1 > 0$，则表明 P_2 待 P_1 发出后不会超时，有 S_1 的时间可以缓冲等待。此时 P_i 的优先级设置为 1，P_2 的优先级暂定为 2。若 $S_1 < 0$，

则表明 P_2 待 P_1 发出后会超时，没有时间可以缓冲等待。此时 P_1 的优先级设置为 2，P_2 的优先级暂定为 1。$S_1 = t_1 - D_2 \times t_f$，若此时 $S_j < 0$，则报文发生死锁，表明那个报文先发送都会导致超时，优先级配置退出。

③ 继续找出总线上响应时间次小的报文 P_3，并且比较 $S_2 = t_3 - (D_1 + D_2) \times t_f$，若 $S_k < 0$,则需要对前面报文的优先级进行调整，调整的原则是 $t_2 < \max\{S_0,S_1\}$。若 $t_2 > S_1$，则表明该报文插入队列后会导致 P_2 延时，该调整无效。若 $t_2 < \min\{S_0,S_1\}$，则报文 P_3 插入 P_1 前面，P_1，P_2 的优先级依次递降 1，$S_0 = S_0 - t_2$，$S_1 = S_1 - t_2$。若 $t_2 > S_0$ 且 $t_2 < S_1$，则 P_3 插入 P_2 前面。此时，P_2 之前由于插入了 P_3，相应的自由时间 $S_1 = S_1 - t_2$。

④ 按照上述方法②，③找出总线上响应时间次小的报文 P_i，$S_{i-1} = t_i - (D_1 + D_2 + \ldots + D_i) \times t_f$，若 $S_{i-1} < 0$，则需要对前面报文优先级进行调整，调整的原则是 $t_i < \max\{S_0,S_1,\ldots,S_{i-1}\}$。若 $t_i > S_{i-1}$，则表明该报文插入队列后会导致 P_{i-1} 延时，该调整无效。若 $t_i < \min\{S_0,S_1,\ldots,S_{i-1}\}$，$P_i$ 报文插入 P_1 之前，后续报文的时间自由度做调整，$S_0 = S_0 - t_i$，$S_1 = S_1 - t_i,\ldots, S_{i-1} = S_{i-1} - t_{i-1}$。若 $t_i < S_j$，且 $t_i > S_{j-1}$，$S_j \in \{S_0,S_1,\ldots,S_{i-1}\}$，则 P_i 插入 P_{j+1} 前面，后续报文的时间自由度做如下调整，$S_0 = S_0 - t_i$，$S_1 = S_1 - t_i,\ldots, S_{i-1} = S_{i-1} - t_{i-1}$。

报文因超时的紧迫性需要加速发送时，通过调整优先级插入到队列之中。保证了前面的报文优先级不变，后面的报文优先级依次递降 1 个单位。

3. 对调度算法的实现分析

由于在 TT-CAN 的时间触发调度算法中，每个节点都是按照总线周期分给自己的时隙访问总线的。在实际的应用中，优先级高的节点相对于优先级低的节点相对于一个数据集合Mi (M由 i 节点发送的所有数据帧组成)的发送，并不占有时间上的优势。也就是说不能按照实际的需要控制各个节点访问总线次数[5]。

而本文所动态优先级调度能够有效的解决这个问题，可以根据实际的需要将各个节点的报文赋于一定的优先级，然后竞争失败的节点在失败后有效调整自己的优先级，从而在下次的竞争中获得有利的地位。并且可以根据报文的最大响应时间科学合理的调整优先级，极大提高了总线的利用效率。

4. 结束语

本文详细讨论了 Can 网络总线调度的实现方法，主要涉及 TT-CAN 调度算法、动态优先级调度算法等关键技术。从 Can 动态调度算法入手，分析其算法原理，提出了在本文所讨论的算法中影响网络实时性和可靠性的一些因素，并采用 MATLAB 软件对算法进行仿真，论证了了动态优先级算法的有效性和实用性。随着工业自动化的不断进步，网络技术的不断发展和硬件的不断更新换代，Can 网络将会在工业领域有更为广阔的应用前景。Can 总线也将在电子、航天、消费等各个领域发挥越来越重要的作用。

参考文献

[1] 汪庆, 黄振卫. 基于 CAN 总线的高可靠测控单元设计[J]. 太赫兹科学与电子信息学报, 2015(4): 12-14.

[2] 牛跃听. CAN 总线嵌入式开发[M]. 北京: 北京航空航天大学出版社, 2016.

[3] 陶震宇, 夏继强, 满庆丰. TTCAN 非周期信息概率延时分析及改进[J]. 仪表技术与传感器, 2014(12): 21-25.

[4] 王秋苹. 基于 CAN 总线船舶数据采集模块的设计与实现[D]: [硕士学位论文]. 大连: 大连海事大学, 2017.

[5] 张新福, 黄海波, 李江江, 范超. 基于 CAN-bus 的组网技术研究[J]. 自动化技术与应用, 2015(10): 30-31.

知网检索的两种方式：

1. 打开知网页面 http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD
   下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询
2. 打开知网首页 http://cnki.net/
   左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：http://www.hanspub.org/Submission.aspx
期刊邮箱：csa@hanspub.org