

Controller area network (can) schedulability analysis: Refuted, revisited and revised

汇报人：张子健

汇报时间：2025.04.16

基本信息

01

作者团队

Robert I. Davis, Alan Burns, Reinder J. Bril,
Johan J. Lukkien

02

所属机构

University of York, Technische Universiteit
Eindhoven

目录

01

摘要

02

Controller Area Network (CAN)

03

响应时间分析

04

优先级分配策略

05

启示与建议

论文摘要解读

CAN总线核心

汽车电子领域的核心技术，实现实时通信与数据传输。



调度分析缺陷

指出当前分析方法的不足，无法满足高实时性需求。

修正与优化

提出改进方法，优化调度分析，提升系统可靠性与CAN性能。



论文价值

揭示现有方法局限，为实际应用提供指导，推动CAN技术发展。

介绍

背景

简要说明所介绍事物的背景信息。

01

应用

介绍该事物在实际中的应用场景。

03

02

特点

概述该事物的主要特点或优势。

04

前景

展望该事物未来的发展趋势或前景。

CAN背景介绍

01

CAN的广泛应用

在汽车行业中，CAN微控制器年产量超过4亿，自1991年梅赛德斯率先量产CAN以来，已被广泛采用。

02

传统线束的挑战

90年代中期，随着ECU数量激增至约40个，连接复杂、线缆繁重、成本高昂等问题逐渐凸显。

03

CAN的普及

面对传统线束的挑战，汽车行业广泛采用CAN，因其成本效益显著。2003年CAN节点销量超过3.4亿，欧洲几乎每辆新车都配备CAN总线。

04

CAN应用及发展

随着CAN的普及，其在汽车行业的应用不断扩展，推动了CAN技术的持续发展。

带CAN外设的微控制器销量

销量增长

1999年至2003年，带CAN外设的微控制器销量显著增长。

销量数据

2003年，带CAN外设的微控制器销量超过3.4亿。

微控制器类型

销量包含8位、16位和32位微控制器。

CAN节点重要性

带CAN外设的微控制器销量增长反映了CAN节点在市场上的重要性和需求的增加。

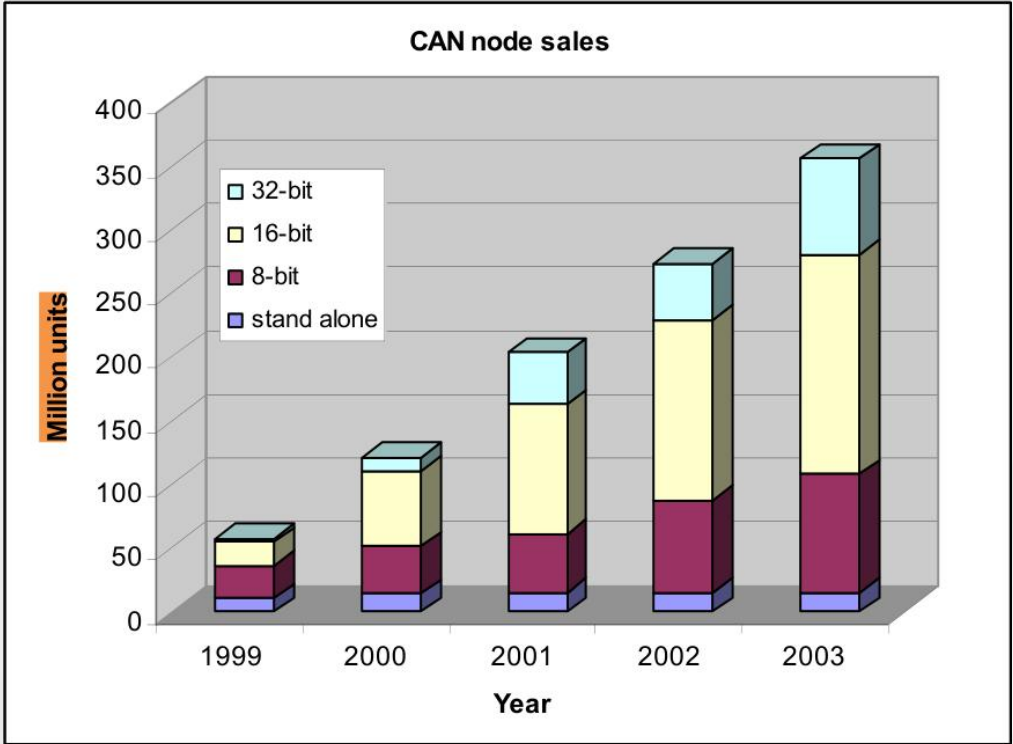


Figure 1: Sales of Microcontrollers with on-chip CAN Peripherals

汽车应用中的CAN总线

CAN总线功能

高速网络连接底盘和动力总成ECU；低速网络连接车身和舒适电子设备；网关实现不同CAN总线间数据交换。

信号传输与实时性

传输车速、油温、发动机转速等信号；制动信号需快速传输；引擎控制信号需5ms发送一次。



网络架构案例

大众帕萨特使用多个CAN总线，连接约45个ECU，LIN提供低速网络连接。

应用领域

广泛应用于汽车行业，实现车内各电子系统间的通信。

BMW 7系网络需求

	Body	Chassis	Powertrain
No. of ECUs	14-30	6-10	3-6
Bandwidth	100 Kbits/s	500 Kbits/s	500Kbits/s
No. of Messages	300	180	36
Cycle Times	50ms-2s	10ms-1s	10ms-10s

Table 1: BMW 7 Series Network Requirements

01

ECU数量

车身14-30个，底盘6-10个，动力总成3-6个

02

带宽需求

车身100 Kbits/s，底盘/动力500 Kbits/s

03

周期时间

10ms至数秒不等

04

网络性能

确保高效、稳定的数据传输

大众帕萨特网络架构

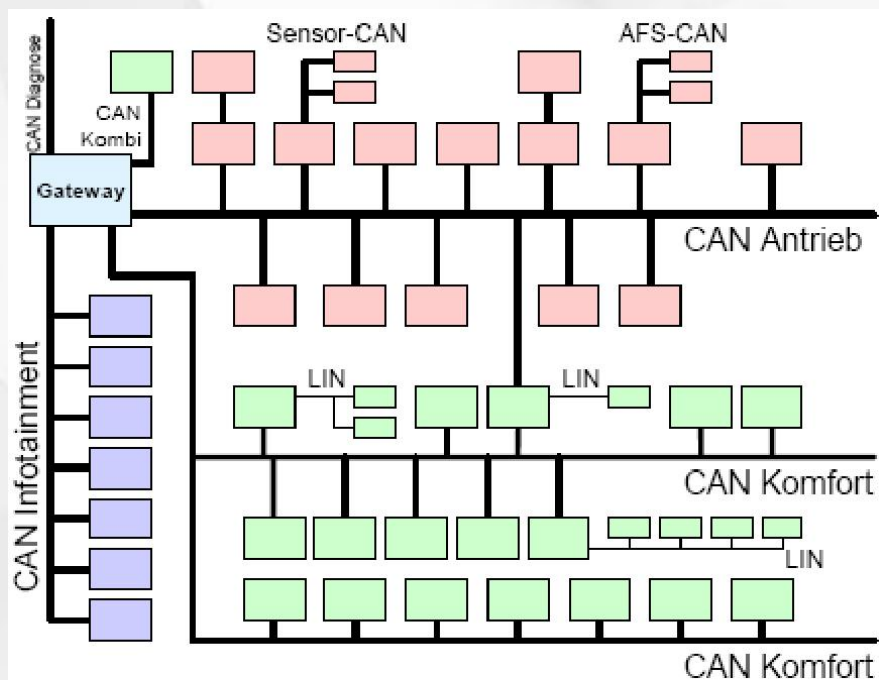


Figure 2: VW Passat Network Architecture

核心架构

以CAN总线为基础，连接约45个ECU。

辅助技术

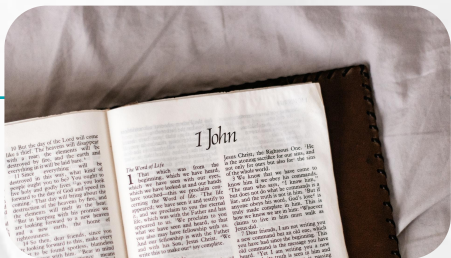
采用LIN总线，作为CAN的补充技术。

高效协同

CAN与LIN总线相互配合，实现车内网络高效协同工作。

稳定可靠

该网络架构确保车辆各项功能稳定、可靠运行。



早期CAN的误解

- 高优先级消息低延迟: 协议擅长传输
- 低优先级消息: 难以保证满足时限
- 需保证所有消息: 满足时限要求

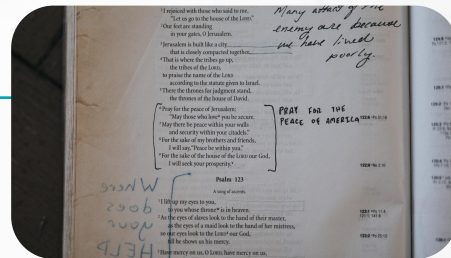
01



Tindell的贡献

- 单处理器调度理论: 应用于CAN消息调度
- 首次实现: 计算CAN消息的最坏响应时间
- 为CAN系统: 提供时序正确性保证

02



研究影响与应用

- 影响芯片设计: Motorola msCAN
- 推动理论发展: 调度理论和错误模型
- Volvo S80成功应用: 配置和分析CAN总线

03

CAN消息缺陷分析

优先级设定

消息A、B、C的优先级分别为
1、2、3。

潜在问题

高优先级消息可能频繁打断低
优先级消息，导致低优先级消
息延迟或丢失。

Message	Priority	Period	Deadline	TX time
A	1	2.5ms	2.5ms	1ms
B	2	3.5ms	3.25ms	1ms
C	3	3.5ms	3.25ms	1ms

Table 2: CAN Messages Highlighting Flawed Analysis

传输时间

每条消息的传输时间为1ms。

解决方案探讨

需优化优先级分配和消息调度
策略，确保各类消息能够及时、
准确地传输。

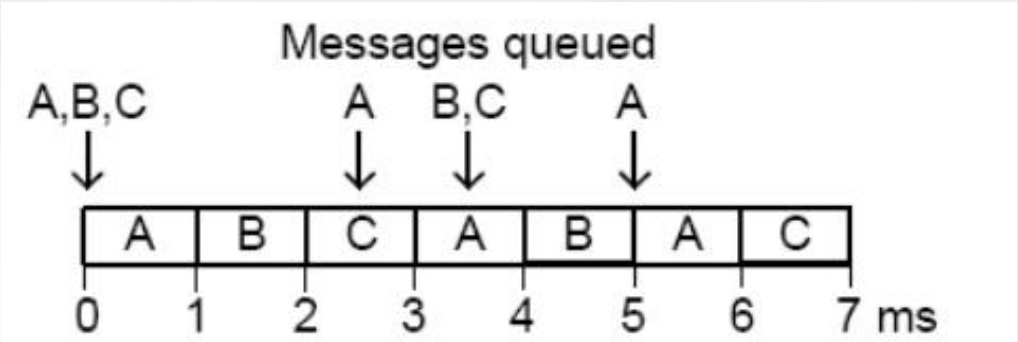


Figure 3: Worst-case Scenario for Message C

动机：CAN调度分析的缺陷



现有分析的局限性

- 依赖性：车辆网络依赖[6, 8, 9]的调度分析
- 缺陷：现有分析可能导致乐观的响应时间
- 实例：表2的消息集合突显现有分析的问题



实例分析与验证

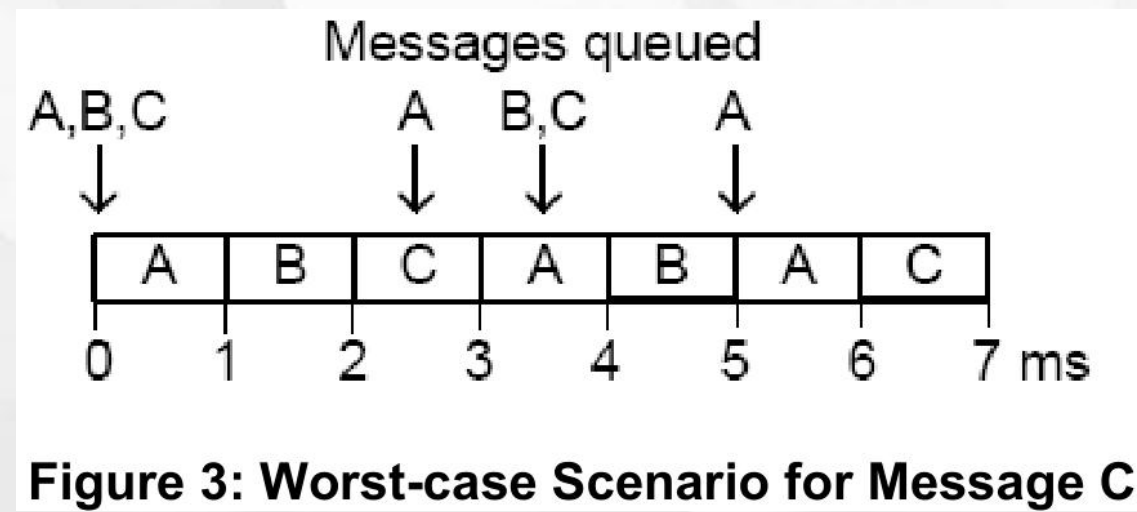
- 网络配置：125Kbit/s网络，消息携带7字节数据
- 消息长度：假设11位标识符，消息最大长度125位
- 传输时间：每条消息的最大传输时间为1ms



问题揭示与影响

- 实例C完成：t=7ms时，实例C响应时间3.5ms
- 实际情况：实例C实际最坏响应时间3.5ms
- 系统状态：违反[6, 8, 9]的保证，系统不可调度

消息C的最坏情况



01

首次调用延迟

高优先级消息A和B导致消息C首次调用被延迟。

02

非抢占性影响

消息C的非抢占性传输影响后续A和B的传输。

03

实际最坏响应时间

消息C的实际最坏情况响应时间为3.5ms。

04

系统性能考虑

需综合考虑消息优先级和传输特性以保障系统性能。

相关工作

02

Joseph算法：
针对任务截止时间
的分析方法

04

早期CAN分析局
限：未考虑错误
帧对总线影响

01

Audsley算法：
优化优先级分配
策略

03

Tindell, Burns,
Wellings研究：
CAN总线调度机
制探讨

05

现有研究改进：
引入错误帧影响，
提升模型精确度

论文组织结构

内容安排

各章节分别介绍CAN协议与术语、错误检测，RTA基础分析与实例验证，以及优先级策略的分析方法适用性。

章节概览

论文包含CAN协议基础、RTA分析核心、讨论与分析等章节。



写作特点

结合实例辅助分析，提出策略和建议，实现从理论到实践的全面探讨。

递进关系

论文结构由浅入深，从协议介绍到RTA分析，再到策略讨论，形成递进式内容安排。

01

Controller Area Network (CAN)

CAN协议与术语

1

CAN总线访问

- CSMA/CR: 载波侦听多路访问/冲突解决
- 优先级仲裁: 通过标识符决定消息优先级
- 非抢占调度: 消息发送如同全局优先级队列

2

CAN错误检测

- CRC校验: 15位循环冗余校验码
- 错误帧类型: 主动错误和被动错误
- 错误帧长度: 17到31位

3

CAN位填充

- 避免错误: 防止出现'000000'和'111111'
- 填充规则: 每5个连续相同位插入反极性位
- 最大传输时间: 与数据字节和标识符有关



CAN标准格式数据帧

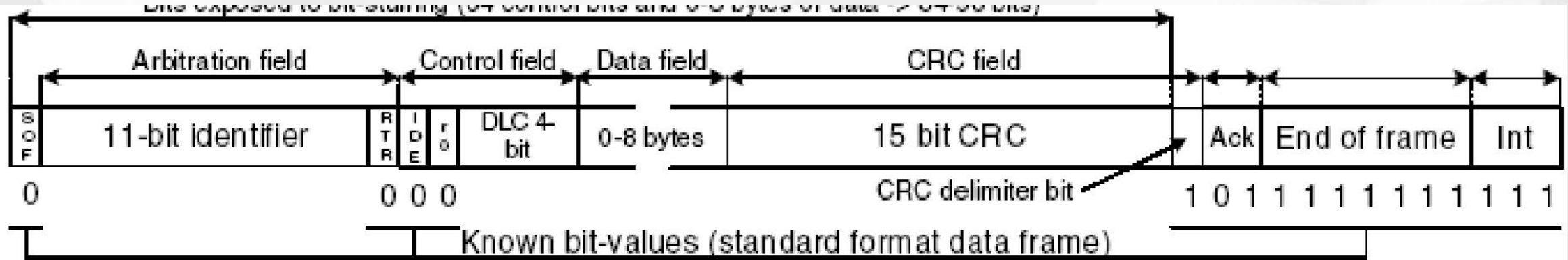


Figure 4: Standard Format Data Frame

标识符长度

11位或29位

数据字节范围

0-8个字节

结构特点

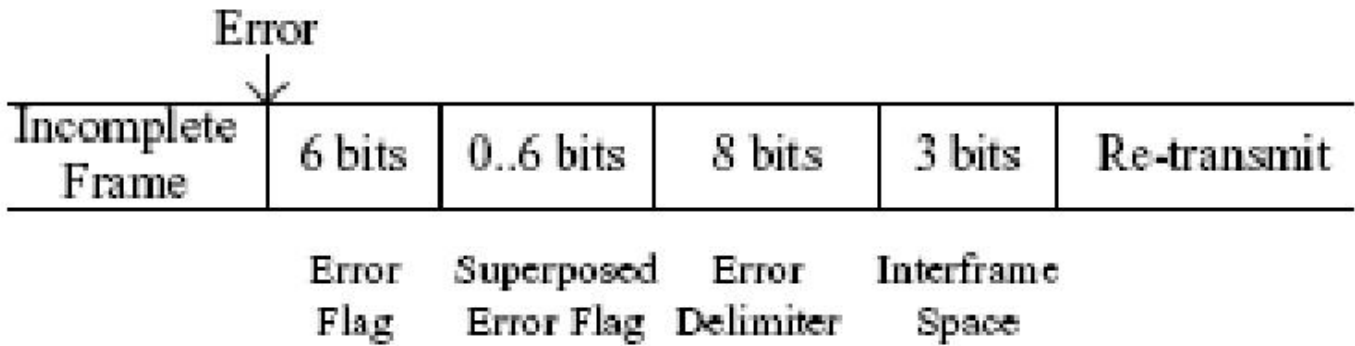
包含标识符和数据字节

应用

汽车、工业控制等领域

CAN错误帧

Active Error Frame



Passive Error Frame

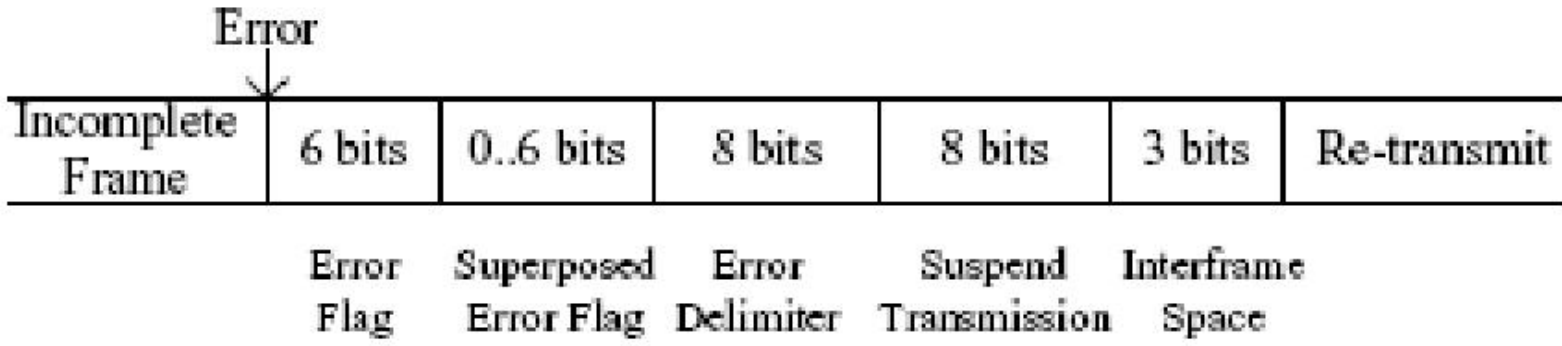


Figure 5: CAN Error Frames

结构

CAN错误帧由错误标志和错误界定符组成。

长度

错误帧的长度为17到31位。

错误恢复

错误恢复涉及开销，用于恢复正常通信。

消息重传

在错误处理后，需要重新传输受影响的消息。

CAN调度模型

01

系统模型

- 节点通过CAN连接
- 节点保证最高优先级消息进入仲裁
- 静态硬实时消息集合

02

消息特性

- 消息 m 具有唯一优先级
- 最大数据字节数 m
- 最大传输时间 C

03

时间约束

- 最小到达间隔时间 T
- 最大容忍时间 D
- $R \leq D$, 消息可调度

最坏情况下的位填充

Before stuffing 111110000111100001111....
After stuffing 11111**00000**11111**00000**11111**10**....

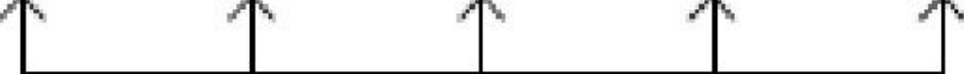

Stuff bits

Figure 6: Worst-case Bit Stuffing

位填充概念

在CAN总线中，为确保数据同步，采用位填充机制。

最坏情况描述

每个填充位启动一个5位序列，此序列本身可能需再次填充。

序列填充需求

5位序列在特定条件下，也需要进行位填充操作。

影响与考虑

这种最坏情况增加了数据传输的复杂性和时间成本。

模型应用的实践意义

CAN网络时钟要求

独立时钟源，需处理消息抖动和位时间，平衡时钟快慢以优化性能。

消息缓冲与约束

针对Motorola等芯片，需管理消息槽和缓冲，防止数据覆盖，确保Deadline合理。

消息优先级与传输

确保最高优先级消息的全局优先传输，注意不同CAN控制器的特性差异。

时钟容差处理关键

为了满足分析的前提假设，CAN控制器必须能够有效处理时钟容差。

02

响应时间分析

基本分析与停止条件

$$w_m = B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m + J_k + \tau_{bit}}{T_k} \right\rceil C_k$$

公式(6)的解读

- w_m : 最坏情况排队延迟
- B_m : 阻塞时间
- $hp(m)$: 优先级高于m的消息集合

$$w_m^n = B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m^{n+1} + J_k + \tau_{bit}}{T_k} \right\rceil C_k$$

公式(7)的迭代

- $w_{\{m\}}^{n+1}$: 下次迭代的延迟
- $w_{\{m\}}^n$: 当前迭代的延迟
- 迭代至 w_m 收敛或超限

$$t_m^{n+1} = B_m + \sum_{\forall k \in hp(m) \cup m} \left\lceil \frac{t_m^n + J_k}{T_k} \right\rceil C_k$$

公式(8)的修正

- $t_{\{m\}}^{n+1}$: 下次迭代的忙碌期长度
- $t_{\{m\}}^n$: 当前迭代的忙碌期长度
- $hp(m) \cup J$: 优先级高于等于m的消息集合

多实例消息检查

实例数量计算

- 公式(10):计算繁忙期内消息m的实例数

$$Q_m = \left\lceil \frac{t_m + J_m}{T_m} \right\rceil$$

- 确定消息m的最坏响应时间

01

最长传输时间

- 公式(11):计算实例q的传输起始时间

$$w_m^{n+1}(q) = B_m + qC_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m^n + J_k + \tau_{bit}}{T_k} \right\rceil C_k$$

- 从繁忙期开始到实例q成功传输的时间

02

响应时间计算

- 公式(12):计算实例q的响应时间

$$R_m(q) = J_m + w_m(q) - qT_m + C_m$$

- 公式(13):消息m的最坏响应时间

$$R_m = \max_{q=0..Q_m-1} (R_m(q))$$

03

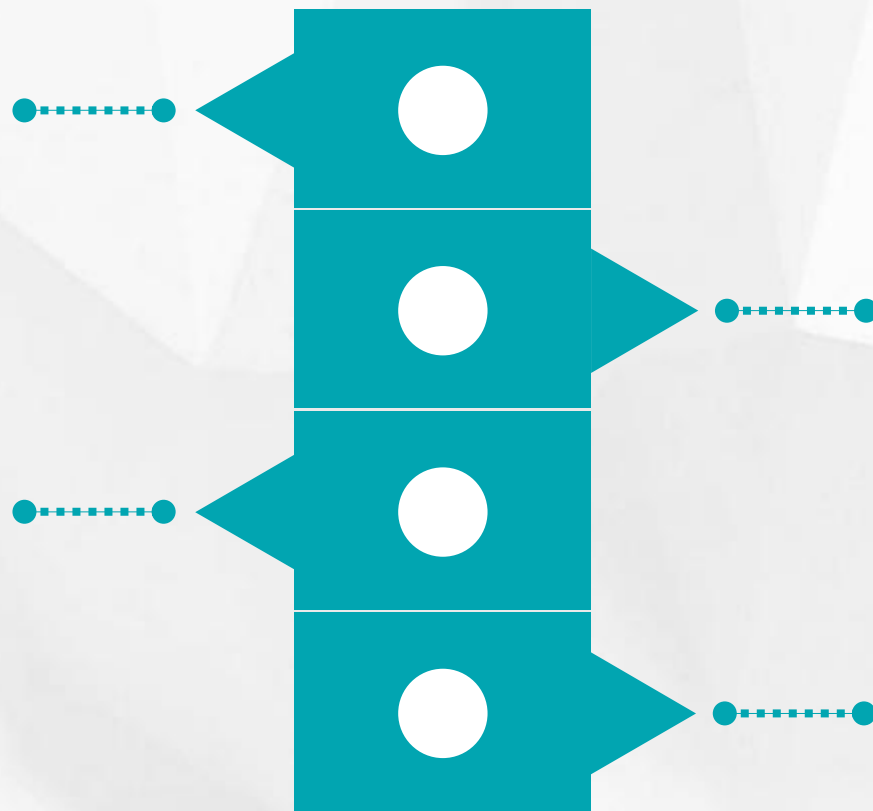
响应时间分析：实例分析

消息参数概述

消息A、B、C具有不同的优先级和周期，传输时间均为1ms。其中，消息C优先级最低，周期为3.5ms。

最坏响应时间计算

消息C的最坏响应时间为3.5ms，需通过迭代计算确定繁忙期长度，以考虑多个消息实例的并发情况。



响应时间问题

传统分析方法可能提供乐观的响应时间估计，无法保证消息满足截止时间。需考虑多个消息实例的影响。

新分析方法需求

传统分析存在局限，需采用新的分析方法进行验证，确保响应时间分析的准确性和可靠性。

CAN消息参数表

Message	Priority	Period	Deadline	TX time
A	1	2.5ms	2.5ms	1ms
B	2	3.5ms	3.25ms	1ms
C	3	3.5ms	3.25ms	1ms

Table 3: CAN Messages

01

优先级设定

消息A、B、C的优先级分别为1、2、3。

02

周期配置

A消息周期为2.5ms，B和C消息周期为3.5ms。

03

传输时间

所有消息的传输时间统一为1ms。

04

消息概述

详述了CAN网络中三条消息的关键参数。

充分可调度性测试

01

充分性定义

测试确认可调度的系统实际可调。

02

非必要性定义

测试判定的不可调度系统不一定实际不可调。

03

简化测试目标

提出更简单保守的调度性测试方法，降低复杂性。

04

推迟干扰上限

消息传输推迟干扰最多为其传输时间 C_m 。

响应时间分析：错误模型

错误假设前提

分析基于无错误发生的前提，但CAN总线实际易错，必须考虑错误带来的影响。

错误类型

包括位错误（单个位翻转）、帧错误（帧格式问题）和确认错误（如ACK缺失）。

错误影响

错误会导致延迟增加、重传，可能引发优先级反转和调度复杂性增加。

错误影响量化

需要量化各类错误对响应时间的具体影响。



讨论

讨论主题

针对特定议题或问题进行深入交流。

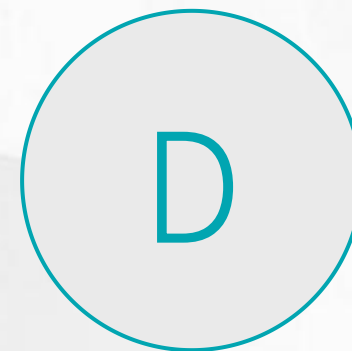
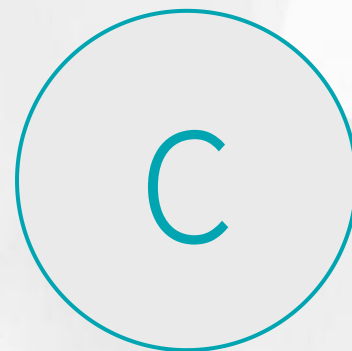


讨论目的

明确讨论的目标，达成共识或提出解决方案。

讨论流程

制定有效的讨论步骤，确保高效且有序的交流。



参与人员

确定参与讨论的人员及其角色，促进多元观点碰撞。

受影响消息的优先级分析

01

忙期计算与实例调度问题

现有分析方法存在实例调度难题，特别是消息X的第二个实例。

02

优先级影响范围

从第三高到最低优先级的消息响应时间可能计算不准确。

03

错误后果

不准确的响应时间计算可能导致系统故障。

04

优先级分类及影响

高优先级消息H对其他消息产生影响，中间优先级I和低优先级L具有特定传输特点。

消息X的繁忙期

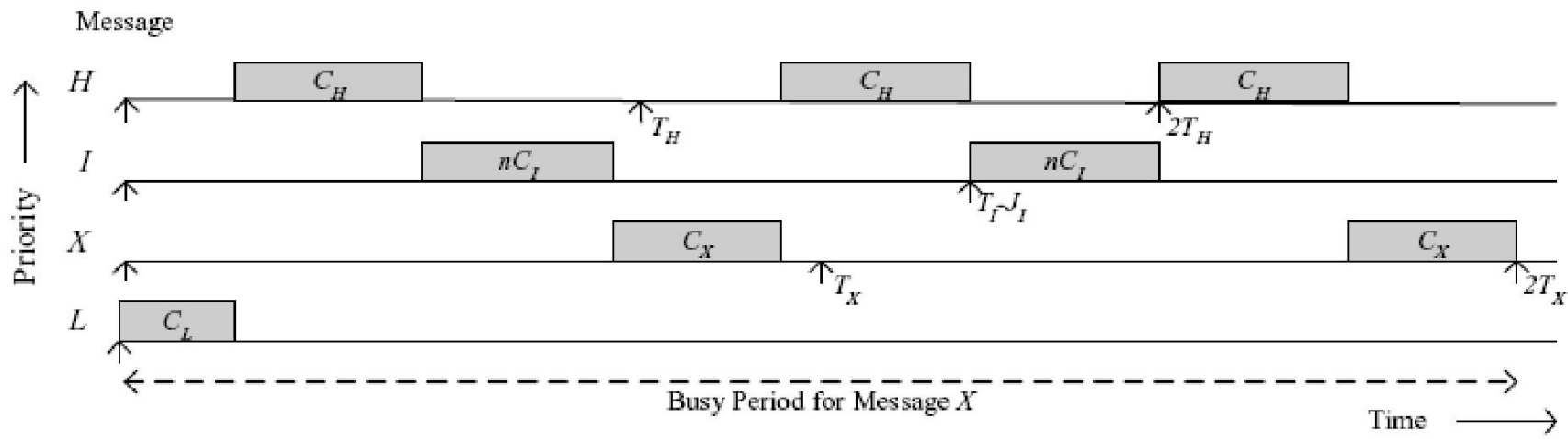


Figure 7: Busy Period for Message X.

消息优先级

H为高优先级，I为中等，X低于I组，L为低优先级。

消息组成

包括H、n个I消息、X及k个L消息。

繁忙期定义

指X消息在处理过程中，由于其他优先级更高的消息（H, I）的插入而导致的处理延时。

影响因素

高优先级和中等优先级消息的数量及到达率会显著影响X消息的繁忙期。

利用率崩溃分析

消息类型考量

仅关注H和X消息，中间及低优先级消息长度趋零。

低利用率崩溃现象

现有分析在极低利用率时失效。

实际应用挑战

现有分析方法失效，难以实现最低利用率。

01

02

03

04

05

病态消息特征

中间优先级消息周期长、截止时间大。

排队抖动影响

实际系统中消息排队抖动大，导致利用率指标评估失准。

打破现有分析的消息集利用率上限

Number of Messages	Utilisation
3	45.5%
5	21.4%
10	9.2%
25	3.4%
100	0.82%

消息数量与利用率关系

随着消息数量的增多，消息集的利用率逐渐降低。

3条消息利用率上限

当只有3条消息时，利用率上限为45.5%。

100条消息利用率上限

当消息数量增至100条时，利用率上限大幅下降至0.82%。

挑战与改进方向

需打破现有分析的限制，提高消息集利用率，以适应大规模消息处理需求。

误差容限分析

典型误码率

极低，范围在 10^{-6} 到 10^{-11} ，主要由外部电磁干扰引起，与消息传输无直接关联。

现有分析缺陷

消息配置可能错过截止时间，但误差率在误差模型参数范围内，可通过模型进行补偿。



误差模型应用

通过公式(21)计算排队延迟，公式(16)进行无误差调度测试，误差模型可弥补现有分析的缺陷。

CAN总线误差影响

对系统鲁棒性有重要影响，需通过详细的误差容限分析来确保系统稳定性和可靠性。

消息遗漏的影响分析

01

诊断消息特性

长度为8数据字节，仅在诊断模式下传输，具有最大阻塞因子。

02

现有分析评估

通过等同充分性测试，得出非乐观响应时间，无截止时间风险。

03

干扰情况说明

至少达到MAXB干扰水平，对比公式(7)和(17)后，确认不影响截止期。

04

优先级消息影响

高优先级消息和低优先级消息的干扰分析已进行。

消息保证不受风险的情况

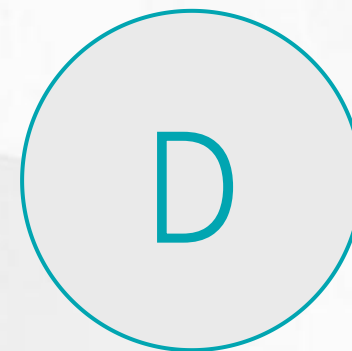
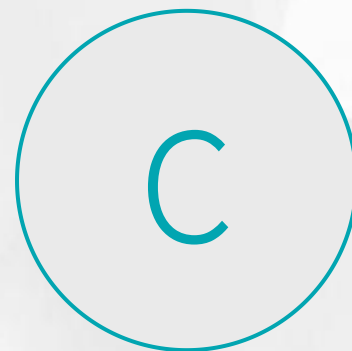
首个实例特性

具有最长响应时间，为后续实例提供排队延迟上界。



响应时间因素

低优先级消息传输时间影响正确响应时间。



正确结果条件

在deadline不超过周期时，现有分析给出正确结果。

最长响应时间

现有分析可保证在特定条件下的最长响应时间。

03

优先级分配策略

优先级分配策略

传统策略失效

Deadline Monotonic在某些情况下不适用，如CAN网络中的反例。

算法复杂度与优化

最多需进行 $n(n-1)/2$ 次可调度性测试，初始顺序对结果影响较大。

Audsley算法适用

该算法在响应时间不依赖高优先级消息时，适用于CAN优先级分配。

启发式优化建议

(D-J)降序排列，相同时按消息长度降序，可提升系统调度性能。

CAN消息非最优优先级分配

Message	Period	Deadline	Number of bits	TX time
A	3.0ms	3.0ms	135	1.08ms
B	4.0ms	4.0ms	135	1.08ms
C	4.5ms	4.5ms	65	0.52ms

Table 5: CAN Messages Highlighting Non-optimal Priority Assignment

01

消息特性

消息A、B含8字节数据，C含1字节。

02

传输环境

网络负载含其他低优先级消息。

03

问题核心

非最优的优先级分配可能影响实时性。

04

潜在风险

大数据量消息若优先级低，可能引发延迟。

消息响应时间与“最优”优先级分配

消息类型与优先级

消息A、B、C及低优先级消息L在队列中的传输情况。

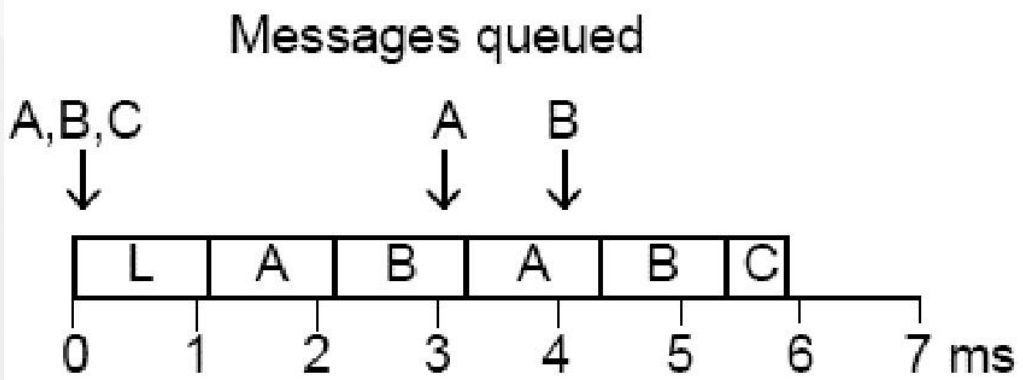


Figure 8: Message Response Times with “Optimal” Priority Assignment

延迟问题

消息C在传输前经历了长时间的延迟。

优先级影响

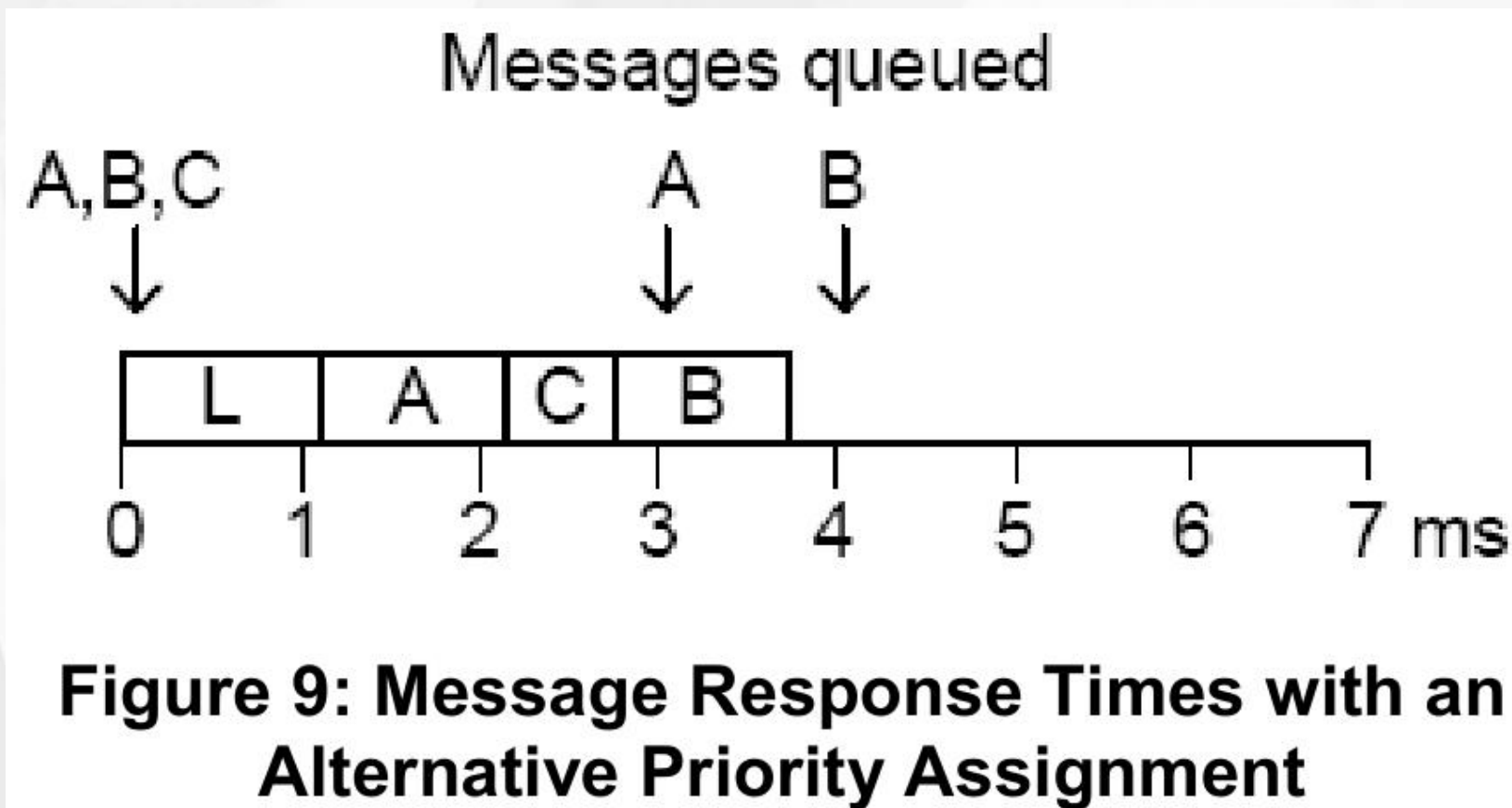
不同消息的优先级分配直接影响其响应时间和传输效率。

优化需求

为实现“最优”优先级分配，需考虑消息的重要性、紧急程度及系统资源等因素。

消息响应时间与优先级分配

1. 展示了消息A、C、B在特定优先级下的响应时间
2. $A=2.16\text{ms}$, $C=2.68\text{ms}$, $B=3.76\text{ms}$
3. 所有消息均满足 deadline



04

启示与建议

CAN调度分析工具的影响与建议

充分性测试

快速修正但结果偏保守，第三节修订分析更优，采用充分非必要测试可快速修复。

工具分析更新

需考虑本文发现，避免错误乐观，保证检查并更新分析，确保结果可靠。



D-J单调性

良好启发式，但可能错过方案，Audsley算法更可靠，确保找到可调度优先级。



优先级分配策略

关键影响因素，需细致考虑，以优化调度效果。



商用CAN应用配置建议

01

预留低优先级消息

为开发和测试预留低优先级通道，确保系统灵活性。

02

正确配置上限

设定合理的信息截止时间，以保障系统的实时性。

03

谨慎验证配置

通过详细分析最坏情况响应时间，确保系统调度的可行性和鲁棒性。

04

评估计算准确性

对响应时间计算的准确性进行评估，以减少潜在错误。

已部署系统中的故障分析



诊断消息影响

诊断消息长度影响系统响应时间，错误计算可导致响应异常。



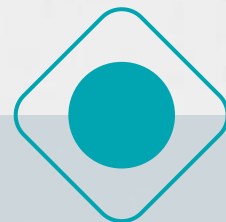
错误模型与消息丢失

商用CAN配置含错误模型，系统设计需考虑消息丢失对安全关键系统的影响。



总线错误分析

高错误率易导致总线故障，最坏情况下需考虑消息相位与位填充。



现有分析局限与容错

实际系统具备容错机制，分析显示失效情况极不可能发生。

总结

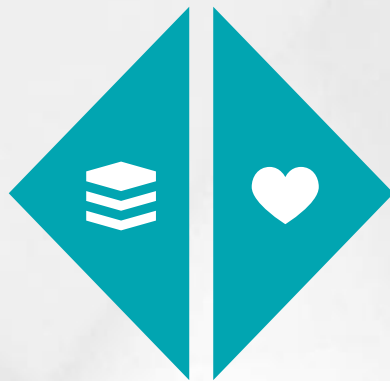
点一

核心内容回顾



点二

重点成果展示



点三

未来方向展望

点四

感谢与期待反馈

总结

现有分析缺陷

乐观计算最坏情况响应时间，可能引发截止时间错失，破坏保障。

01

02

影响范围

主要涉及优先级第三高及以下的消息，导致低总线利用率和系统保障破坏。

03

优先级分配策略评估

Deadline Monotonic和(D-J)-monotonic均非最优，Audsley算法适用于CAN，可实现可调度的优先级排序。

04

核心问题

CAN调度分析存在缺陷，需修正以准确计算响应时间。



THANKS

感谢观看