

DOI:10.19651/j.cnki.emt.2415409

基于时间约束的CAN网络管理一致性测试*

张 建¹ 王 辉¹ 李超超² 徐封杰² 方 菱²

(1. 安徽大学物质科学与信息技术研究院 合肥 230000; 2. 中国科学院合肥物质科学研究院 合肥 230031)

摘要: 汽车网络管理是现代汽车必不可少的基础功能。随着智能网联汽车的兴起,ECU数量的增加使得车载网络系统的需求和复杂性不断提高。为了确保网络管理的安全性和稳定性,需要全面的测试解决方案。本研究在深入理解AUTOSAR网络管理规范的基础上,提出一种针对AUTOSAR CAN网络管理的测试方法。首先基于AUTOSAR提供的标准TTCN-3测试脚本搭建测试系统,设计不定长测试报文,实现对CAN网络管理的多场景测试。其次基于巴科斯范式定义一种新的语法格式,对TTCN-3进行时间约束,提高对CAN网络报文的监测准确度。利用此方法,在基于MPC5745B平台的实际测试中,定位到3处异常信息,评估了AUTOSAR CAN网络管理的安全性和稳定性。实验结果表明,该方法能够在节省测试软硬件资源的情况下,全面测试AUTOSAR CAN网络管理的一致性。

关键词: AUTOSAR;CAN网络管理;TTCN-3;一致性测试

中图分类号: TP2 **文献标识码:** A **国家标准学科分类代码:** 510.1050

Conformance testing of CAN network management based on time constraints

Zhang Jian¹ Wang Hui¹ Li Chaochao² Xu Fengjie² Fang Ling²

(1. School of Materials Science and Information Technology, Anhui University, Hefei 230000, China;

2. Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China)

Abstract: Network management is an essential foundational function in modern automobiles. With the rise of intelligent connected vehicles, the increasing number of Electronic Control Units has led to higher demands and complexities in in-vehicle network systems. To ensure the security and stability of network management, comprehensive testing solutions are required. This study proposes a testing method specifically for AUTOSAR CAN network management, based on a deep understanding of the AUTOSAR CAN network management specification. Firstly, a testing system is constructed using the standard TTCN-3 scripts provided by AUTOSAR. The system is designed to generate variable-length test messages and conduct multi-scenario testing of CAN network management. Secondly, a new syntax format is defined based on the Backus-Naur Form to impose time constraints on TTCN-3, thereby enhancing the accuracy of monitoring CAN network messages. Using this method, three abnormal messages were identified in practical tests on the MPC5745B platform, enabling the evaluation of the security and stability of AUTOSAR CAN network management. Experimental results demonstrate that this method enables comprehensive testing of the consistency of AUTOSAR CAN network management while conserving testing software and hardware resources.

Keywords: AUTOSAR;CAN network management;TTCN-3;conformance testing

0 引 言

随着近年汽车产业的迅速发展,汽车系统的设计日益复杂,电子控制单元(electronic control unit, ECU)的数量也

在快速增加。为了实现ECU之间的信息共享,车载网络作为一种互联网络应用于汽车控制系统中。汽车开放系统架构(automotive open system architecture, AUTOSAR)网络管理为车载网络提供了标准的管理策略,通过接口和服

收稿日期:2024-01-23

* 基金项目:安徽省重点研究与开发计划项目(202004a05020041)资助

务来监控和管理汽车网络中的 ECU 节点,以确保车载网络信息传输的准确性和安全性^[1]。

AUTOSAR 标准^[2]在汽车电子开发中的广泛应用推动了行业的发展,但也带来了一些新的挑战。由于 AUTOSAR 规范以文档形式存在,存在着开发中的二义性。一旦开发者存在理解偏差,很容易导致车载网络管理协议的实现与规范不一致,对汽车安全构成潜在威胁^[3]。尤其是随着智能网联汽车的兴起,网络管理状态转换的频率不断增加,从而使异常事件发生的概率增加^[4]。因此,车载网络管理(network management, NM)的一致性测试是汽车网络管理中重要的测试环节。

一致性测试是一种黑盒测试,旨在通过模拟多样化的环境和运行条件,对被测系统的功能进行全面验证。该测试方法不依赖于被测系统的内部实现细节,专注于评估系统在不同环境下的一致性行为。通过设计适当的测试用例和数据输入,一致性测试可以有效检测系统在各种条件下的功能正确性、稳定性和可靠性,从而为系统的质量保证提供重要支持^[5]。

关于一致性测试,相关研究人员进行了一定的研究。周毅等^[6]在分析一致性测试执行架构的基础上,针对 AUTOSAR 测试标准,基于开源软件 Eclipse Titan 和 AUTOSAR 官方提供的测试与测试控制标记语言第 3 版(testing and test control notation version 3, TTCN-3)脚本设计一套统一的通信报文格式,保证了可移植性,解决了测试系统和被测系统在不同平台下的交互问题,并最终通过对 AUTOSAR Lin SM 模块的功能性接口及上下层通知交互进行实验,验证了本方法的有效性。然而,由于 TTCN-3 在测试具有时间参数的系统(如 AUTOSAR CanNm)时存在局限性,无法准确测试 CAN 网络管理中报文的周期性发送。Luo 等^[7]通过对 AUTOSAR NM 状态机的分析,提取 NM 一致性测试的关键因素,并据此搭建虚拟网络管理测试仿真环境。在该测试环境中,其利用模拟器模拟虚拟的 ECU 网络节点,并通过信号触发器给被测系统信号激励,同时通过数据分析仪监测网络管理报文的发送周期。然而,该方法只关注在不同网络状态下网络管理报文是否按照配置的定时器时间进行发送,仅覆盖到部分 AUTOSAR 网络管理规范,被测系统中仍有可能存在众多不符合规范的潜在问题,难以保障系统的可靠性和稳定性。此外,该方法还需要依赖于众多硬件平台,增加了测试成本。文献[8]将基于属性的测试(property-based testing, PBT)应用于 AUTOSAR 基础软件测试中,此方法针对系统功能定义无状态以及有状态属性,以此来描述系统在给定输入下应用具有的行为和特性,同时利用 Python 工具 Hypothesis 搭建测试框架,通过随机生成输入数据,对系统进行测试。显著提升了对 CAN 网络管理的测试覆盖率和异常探测能力,然而此方法在进行属性定义时存在一定挑战,需要测试人员深入理解被测系统功能和特性,并确保属

性能正确描述系统的预期行为。而且在生成测试用例时会包含大量的输入数据组合和状态变化,这会增加测试用例的维护成本,需要额外的资源和时间来执行测试。文献[9]将整个测试系统分为 PC 上位机、电源控制模块、程控电源、随机唤醒源发生模块、NM 报文监控模块以及 CAN 通信网络控制模块这六大模块。通过随机唤醒源发生模块对被测系统发出信号激励,并通过 NM 报文监测模块实现对网络管理报文的监测,最终通过所记录的数据生成网络版测试报告。此方法与文献[6]类似,仍存在测试覆盖率不够以及需要依赖于众多硬件平台的问题。

为弥补现有研究的不足,本文针对现有研究在测试 AUTOSAR CAN 网络管理所面临的灵活性不足、测试覆盖率不充分以及高昂的测试成本等问题,提出了一种全新的测试方法。该方法旨在在节约测试资源的前提下,实现对 AUTOSAR CAN 网络管理的全面测试。首先本研究选用开源软件 Eclipse Titan 作为 TTCN-3 的编译工具,通过对 AUTOSAR 提供的标准 TTCN-3 测试脚本^[10]深入分析,使用 Uart 串口建立测试通信链路,设计可变长测试通信报文。实现了在不干扰底层网络管理协议栈运行的情况下,对 AUTOSAR CAN 网络管理的功能性接口和上下层交互进行多场景化测试。其次,为了能够实现在不同网络管理状态下对网络管理报文的准确监测,本研究设计网络管理测试报文与被测系统的具体接口进行映射,以此来模拟激励信号,控制 CAN 网络管理协议处于不同状态以进行网络管理报文的发送,同时针对 TTCN-3 在测试具有时间参数的系统存在一定的缺陷,本研究基于巴克斯范式定义一种新的语法格式,对 TTCN-3 进行时间约束,从而能够准确地监测网络管理报文在不同网络状态下是否按照预定的定时器时间进行发送,无需依赖于昂贵的软硬件,同时提高了测试灵活性。

1 AUTOSAR CAN 网络管理一致性测试

1.1 AUTOSAR CAN 网络管理协议栈

AUTOSAR CAN 网络管理协议栈即基于 CAN 总线的网络管理协议栈,它是一种分布式直接网络管理,基于周期性网络管理报文,通过广播传输,由集群中所有节点接收^[11]。图 1 为符合 AUTOSAR 规范的网络管理软件架构。

ComM 模块,负责简化用户对网络管理和总线通信状态的控制。CANSM 模块负责根据 ComM 的通信请求设置 CAN 控制器和收发器的工作状态,从而同步信道的通信状态和通信设备的工作状态。NMIf 模块封装了不同类型总线网络管理的接口,向 ComM 提供网络管理的统一访问接口。

CANNM 是 AUTOSAR CAN 网络管理协议栈的核心功能模块,负责管理节点的网络状态,向 ComM 提供依赖于 CAN 总线的网络管理服务。CANIF 模块承担着与底层

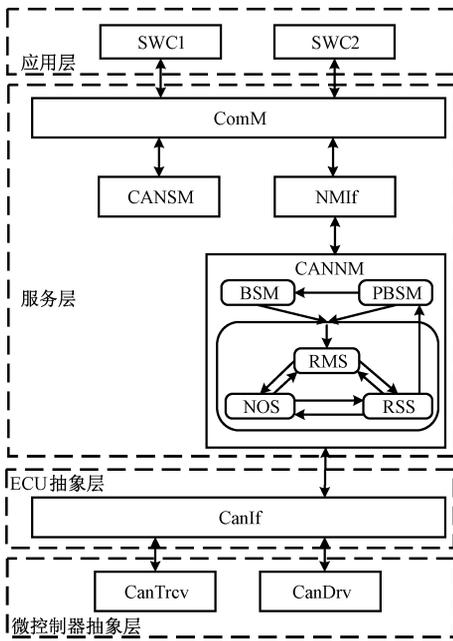


图 1 AUTOSAR 网络管理协议栈

CAN 控制器和收发器进行交互的任务,为 CANNM 模块提供通信访问接口。

根据 AUTOSAR CAN 网络管理规范^[11],网络管理状态机内部又分为 3 个模式:总线休眠模式(bus-sleep mode, BSM)、网络模式(network mode)、总线预休眠模式(prepare bus-sleep mode)。在总线休眠模式下:网络中各节点都处于休眠模式,禁止网络管理报文和应用报文的发送,不对 CAN 总线上的报文进行 ACK 应答;在网络模式下,网络中节点都处于唤醒状态,该模式又分为 3 个子状态,分别为:1)重复报文状态(repeat message state, RMS):周期发送重复网络管理报文通知其他节点自身的存在。2)正常操作状态(normal operation state, NOS):节点会周期发送网络管理报文来保持网络唤醒状态。3)等待休眠模式(ready sleep state, RSS):节点禁止发送网络管理报文但会因接收到网络管理报文而进入到 RSS 或 NOS;在总线预休眠模式下:网络中各节点处于静默模式,各节点禁止发送网络报文和应用报文,但需要对总线上的报文进行 ACK 应答。

1.2 CAN 网络管理一致性测试评估

1) TTCN-3 概述

TTCN-3 作为一种强大的编程语言,在研究和实践中被广泛应用于测试领域。它具备广泛的适用性和可移植性,能够应用于多个协议层级的测试任务^[12]。

为了满足不同的测试需求,TTCN-3 同时支持基于过程和基于消息的通信。基于过程的通信是一种同步通信模型,组件之间通过直接调用彼此的过程或函数进行通信。基于消息的通信是一种异步通信模型,组件之间可以独立发送消息,不需要等待被测系统的响应^[13]。

2) TTCN-3 在一致性测试中的应用

AUTOSAR 采用基于过程的通信机制设计基于组件级的 TTCN-3 测试脚本,对 CAN 网络管理的功能性接口及上下层交互进行多场景测试。测试过程中,首先针对协议栈中标准接口,采用 TTCN-3 设计伪接口模拟其功能,并利用 Call 关键字调用协议栈中实际标准接口;针对回调接口(上下层通知接口),采用 GetCall 关键字检查回调接口是否被调用。最后,将标准接口返回值与伪接口运行结果进行比对,并对 GetCall 的返回值进行判断,以验证被测功能与测试目的是否一致。具体的一致性测试用例设计流程如图 2 所示。

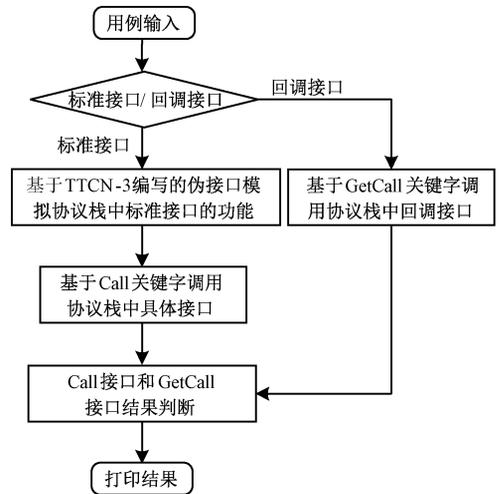


图 2 AUTOSAR 测试用例设计流程

3) TTCN-3 在网络管理测试中的缺陷

尽管 TTCN-3 在协议的功能测试方面具有显著优势,但仍存在一些局限性。作为一种测试规范语言,TTCN-3 最初是为了测试通信系统的一致性而开发^[14-15]。然而,在嵌入式系统中,仅通过功能测试需求无法全面验证系统的一致性。除了功能测试需求外,还必须进行与时间相关的非功能性需求测试,如 AUTOSAR CAN 网络管理中网络报文和应用报文的周期性发送。对于与时间相关的测试需求,TTCN-3 提供了定时器等遵守时间要求的测试手段^[16],然而在 TTCN-3 中定时器的使用受到快照语义以及接收/超时事件在 alt 语句中排列顺序的影响,所测试的时间点并不是消息接收的时间点,而是测试程序评估队列的时间点,这导致时间测量的准确性不够。图 3 描述了 TTCN-3 中触发快照拍摄时 alt 语句的评估流程,alt 语句的静态结构由一定数量的 alt 分支组成,在不同的 alt 分支中会经过解码、编码、匹配和检查等操作。这一系列操作过程的实际耗时无法准确评估。

除此之外,能够触发快照机制的不仅仅只有定时器超时事件,还有消息接收/发送事件。由事件触发的快照序列图如图 4 所示。其中 e_i 表示事件,包括发送消息事件、接收消息事件、超时事件等, S_n 表示 e_i 事件所触发的快照,

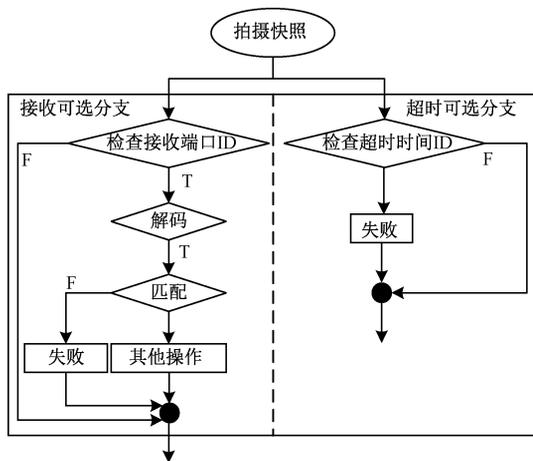


图 3 TTCN-3 中 alt 语句评估流程

S_n 和 S_{n+1} 之间的这段时间表示 S_n 的评估时间,即检查、编码、解码、匹配的耗时。在图 4 中,时间 e_0 触发快照 S_n ,而下一个快照 S_{n+1} 的触发只能等到 S_n 快照评估完成之后才能进行,在这个过程中又有可能有 e_1, e_2, e_3 等其他事件的到来,为了评估系统的实时性,只能得出 $t(e_1) < t(e_2) < t(e_3) \dots < t(e_k) < t(s_{n+1})$,而无法得出具体准确的时间。

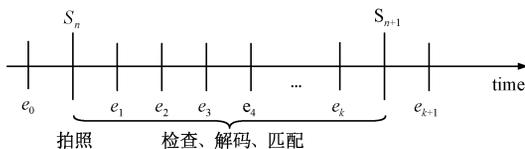


图 4 事件触发的快照序列图

2 CAN 网络管理一致性测试方法

本节将介绍 AUTOSAR CAN 网络管理一致性测试的总体设计与实现。

2.1 测试框架概述

本研究将测试系统分为两个模块,即测试管理模块和测试服务模块,测试管理模块位于上测试器 UT 中,由测试序列组成,并生成最终的测试结果;测试服务模块在上测试器 UT 和被测系统 SUT 中均有分布。在本研究中,测试服务模块起到两方面作用,一方面用于生成和解析测试通信报文,另一方面做为被测系统的应用程序,通过调用 CAN 网络管理中 API 接口,控制 SUT 的运行,并收集 SUT 中接口的运行结果,返回给上测试器 UT。本研究的测试架构如图 5 所示。

2.2 通信报文设计

通信报文包括测试码、返回码以及 CAN 网络管理测试报文,用于上测试器 UT 和被测系统 SUT 之间的通信。

测试码用于 UT 对 SUT 中具体接口的调用,需包含与 CAN 网络管理测试报文的区分位、接口编号、接口调用的参数长度以及参数编码值。参数编码规则需与 TTCN-3 的编码规则保持一致,对于整形参数,将其统一编码为 4 字

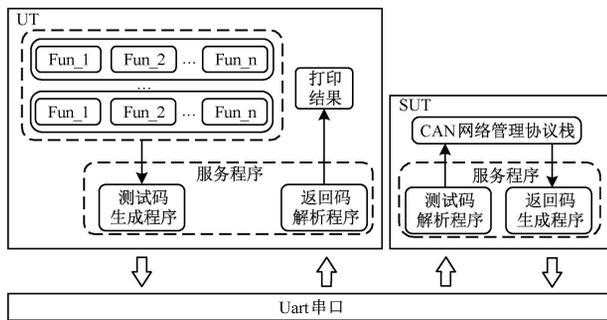


图 5 测试框架图

节大端数据;结构体类型按照参数长度依次编码,对于指针类型数据,将其指向的内容编码。不定长测试码的格式如图 6 所示,第 1 个字节表示标准测试用例,用于区分扩展的测试用例、第 2 个字节为接口编号、第 3 个字节表示参数长度、后续字节根据接口入参依次添加。

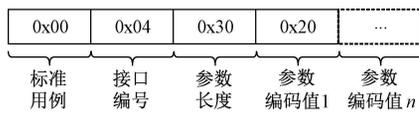


图 6 不定长测试码设计

返回码在 SUT 中产生,携带网络管理协议栈中接口的返回信息,在处理返回码时,需考虑 3 种情况:无返回值接口,有返回值接口以及需检查接口中入参是否更新。返回码格式如图 7 所示,第 1 个字节表示接口编号、第 2 个字节表示返回长度。对于无返回值接口,返回长度编码为 0 用于通知上测试器接口调用成功;有返回值接口或需检查入参是否更新的接口,根据所需返回信息进行相应编码。

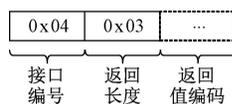


图 7 测试返回码

CAN 网络管理测试报文用于测试 CAN 网络报文的周期性发送,长度为 6 字节。在 SUT 中,会通过测试码解析程序对其重新组装,并传递给 CanIf_RxIndication 接口。当 CanIf_RxIndication 接收到网络管理测试报文时,会通过回调接口 CanNm_RxIndication 通知 CAN 网络管理层,从而触发对应通道的休眠或唤醒操作。CAN 网络管理测试报文格式如图 8 所示。第 1 个字节用于区分 CAN 网络管理测试报文和其他不定长测试报文,第 2、3 字节为 CAN Id (2 字节)表示 PDU 数据所属 CAN 标识符、第 4 个字节为硬件单元(1 字节)CAN 标识符所属硬件单元、第 5 个字节为控制器 Id,是 CanIf 层所需的控制器标识符、第 6 个字节表示唤醒或休眠方式(1 字节);用于指示唤醒或休眠操作的方式。01 表示主动唤醒,00 表示被动唤醒,02 表示休眠操作。

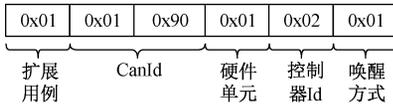


图 8 CAN 网络管理测试报文

2.3 TTCN-3 实时性扩展

对 CAN 网络管理中报文的周期性发送进行准确监测,需消除 TTCN-3 中消息评估流程的影响。为此本研究基于 TTCN-3 中的标准语法规则巴克斯范式定义一种新的语法形式为 TTCN-3 做实时性扩展。

为获得消息发送/接收的准确时间点,可以将消息的接收和发送时间作为消息属性进行存储。在 TTCN-3 中,通过重定向机制可以实现类似的访问方式。时间戳重定向会将消息的到达时间(即消息入队时间)或发送时间(即消息被编码并传入到输出端口的时间)存储到指定的时间戳变量 *datetime* 中。此外,重定向机制适用于 TTCN-3 中所有通信语句类型。表 1 展示了基于时间戳的语法规则,其作用是将某特定端口或任意端口接收消息的时间点通过重定向机制存储到时间戳变量 *timestamp* 中。

表 1 基于时间戳机制语法规则

```

<ReceiveStatement> ::=
<PortOrAny> " receive" [ "(" <RceiveParameter>
">")" ]
[ <FromClause> ] [ <PortRedirect> ]
<PortRedirect> ::=
"→"
(( <ValueSpec> [ <SenderSpec> ] | <SenderSpec> )
[ <TimeStampSpec> ] | <TimeStampSpec> )
<TimeStampSpec> ::=
"timestamp" <Identifier>

```

除了要消除评估队列中不确定时间的影响,还需对不同端口的消息传入、传出时机以及消息超时处理进行控制。为此,本研究引入时间谓词控制不同端口消息传入/传出时机,并通过 break 子句限制消息超时后快照的创建。时间谓词的语法规则如表 2 所示,由时间运算符 at、after、before 和 within 组成,可以用于简单的时间条件判断。复杂的时间谓词通过逻辑运算符 or 和 and 连接多个谓词来进行组合。

表 2 时间谓词语法规则

```

<TemporalPredicate> ::=
<SimplePredicate> [ ("and" | "or") <SimplePredicate>
"> ]
<SimplePredicate> ::=
([ "not" ] <TemporalPredicate> ) | <Within> | <
BeforeAfterAt>

```

```

<Within> ::=
" within" " (" ( <DateTimeExp> | <TimeSpanExp>
"> ) ". ."
( <DateTimeExp> | <TimeSpanExp> ) )"
<BeforeAfterAt> ::=
(" before" | " after" | " at") ( <DateTimeExp> | <
TimeSpanExp> )

```

break 子句与普通的 TTCN-3 超时的主要区别体现在时间谓词的更高表达能力以及它们直接参与时间戳机制。break 字句的语法规则如表 3 所示,alt 表示一个选择语句,<AltGuardList> 是备选项的表,其中每个备选项都有一个条件。程序会依次检查每个备选项的条件,然后选择第一个满足条件的备选项执行,使用关键字 "break" 可以指定在选择指定备选项后立即退出整个 "alt" 语句。

表 3 break 子句语法规则

```

<AltStatement> ::=
"alt" <AltGuardList>
[ "break" <TemporalPredicate> ] [ <StmtBlock> ]
<AltStatement> ::=

```

2.4 消除通信链路中数据传输延迟影响

UART 串口是一种异步串行通信接口,数据按位逐个传输,每个数据位传输时间为微秒级,不会对报文的周期性发送产生影响。然后在测试服务程序中,需等到整条报文接收完毕之后,才能够对协议栈中的具体接口进行调用,当通信报文长度达到一定长度,此过程耗时就会达到毫秒级,与报文的发送周期在同一个数量级。例如在波特率为 115 200 bps 的情况下,传输 8 字节报文就会产生 0.67 ms 的误差。为了消除通信链路中数据传输延迟的影响,特别是由串口波特率引起的数据传输延迟,可以使用式(1)计算出延迟时间,并在扩展测试用例中添加传输延迟差。其中 N_{bits} 表示要传输的数据位数, R_{baund} 表示串口波特率。

$$T_{delay} = \frac{N_{bits}}{R_{baund}} \quad (1)$$

由式(1)可以得出报文 M 在通信链路的数据传输延迟时间 T_D ,如式(2)所示,其中 M_{bytes} 表示报文字节数。

$$T_D = \frac{8 \times M_{bytes}}{R_{baund}} \quad (2)$$

2.5 基于扩展 TTCN-3 的测试用例

对于 CAN 网络报文的周期性监测,本研究选用基于消息通信模型扩展测试用例。与基于过程通信相比,基于消息通信模型是异步机制,测试环境可以按照预定的时间间隔发送报文,不会因等待响应影响周期性。

以 AUTOSAR CAN 网络管理中规范 ID 为 SWS_CanNm_00335 为例,AUTOSAR 所提供的一致性测试文

档并未设计相关测试用例。基于本研究扩展 TTCN-3 对该条规范所设计的测试用例如表 4 所示。其测试目的是在 CAN 网络管理协议栈主动唤醒后,将经历一段持续 100 ms 的快发阶段,快发周期为 10 ms,共发送 10 次快发报文。其中 T_d 表示报文传输延迟总时间, T_s 表示主动唤醒报文 T_m 到达端口 PT_NM 的时间点, INTERLEAVE 为可选步, T_r 为端口 PT_NM 接收到 CAN 网络管理报文 R_m 时间点, 端口每次接收到 R_m 的时间点均存储到时间戳数组 TSP 中, 同时将接收次数 Count 加 1。在时间 $T_r=100$ ms 之内, 对接收到的相邻两条报文 CAN 网络报文 R_m 的时间间隔进行判断, T_{sd} 为 SUT 向 UT 传输单条网络报文的传输延迟时间, T_{rd} 为 UT 向 SUT 发送 CAN 网络管理测试报文的传输延迟时间。当时间到达 $T_s + T_d + T_r$ 之后, 将通过 BREAK AFTER 操作中断可选步, 并判断所接收到的 CAN 网络报文的次数。

表 4 基于扩展 TTCN-3 一致性测试用例

测试用例 1. 主动唤醒时报文的周期性发送

发送: UT 发送主动唤醒报文 T_m

接收: UT 每 10 ms 接收到 CAN 网络报文 R_m , 维持时间为 100 ms

1. PTNM_MSG = {CANNM 端口中的 T_m 和 R_m }
2. TSP = {端口接收或发送报文的时间戳数组}
3. PT_NM.SEND(T_m)
4. → 存储发送时间点到变量 T_s
5. END SEND
6. INTERLEAVE
7. PT_NM.RECEIVE(R_m)
8. → 存储接收时间点到变量 T_r
9. END RECEIVE
10. TSP[Count++] = T_r
11. IF Count = 0
12. IF TSP[0] - $T_s - T_{sd} - T_{rd} = 10 * ms$
13. SETVERDICT(PASS)
14. END IF
15. ELSE
16. SETVERDICT(FAIL)
17. END ELSE
18. END IF
19. ELSE
20. IF TSP[Count] - TSP[Count-1] = 10 * ms
21. SETVERDICT(PASS)
22. END IF
23. ELSE
24. SETVERDICT(FAIL)
25. END ELSE

26. END ELSE
27. END INTERLEAVE
28. BREAK AFTER($T_s + T_d + T_r$)
29. IF Count = 10
30. SETVERDICT(PASS)
31. END IF
32. ELSE
33. SETVERDICT(FAIL)
34. END ELSE
35. END BREAK AFTER

2.6 基于 CAN 通信协议栈确定报文发送点

由图 1 可知, CAN 网络管理协议栈在向总线发送 CAN 网络管理报文时, 需要通过 CanIf 层提供的 CanIf_Transmit 接口去访问底层的 CAN 驱动。报文的最终发送点通过 CAN 驱动层的 Can_Write 接口实现, 因此在 CAN 驱动层选取测试点插入桩函数。由于 CAN 网络管理报文在不仅需要携带 PDU 信息, 同时需要携带帧 Id 信息, 为此在确定插桩点之后, 本研究会通过桩函数将 CAN 网络报文重新组装并通过串口发送到上测试 UT 进行解析。为了区分与标准测试用例的区别, 本研究在测试码解析程序中通过测试码的第 1 个字节控制桩函数的开启和禁用。

3 实验与分析

本研究基于实验室自主研发的 AUTOSAR CAN 网络管理协议栈进行一致性测试, 参照的 AUTOSAR 官方规范文档版本为 4.0.3, 被测系统基于 MPC5745B 芯片平台, Uart 串口波特率选用 115 200 bps。在实验进行之前, 首先需要将上测试器 UT 和下测试器 SUT 中的配置保持一致, 从而确保二者能够处于同一条件下, 使得上测试器 UT 中的 TTCN-3 测试脚本能够准确的模拟下测试器 SUT 中协议栈接口的具体功能。本研究通过对 AUTOSAR CAN 网络管理协议栈中初始化、PDU 数据、状态转换、被动唤醒、远程休眠、发送接收、启用禁用通信、重复报文和 CAN 网络管理报文的周期性发送等九个关键功能进行一致性测试, 验证了本方法的有效性。具体的实验结果与实验分析, 将会在第 3.1 节和第 3.2 节展开讲解。

3.1 实验结果

本研究对 CAN 网络管理一致性测试时共使用 67 条测试用例, 覆盖了 AUTOSAR CAN 网络管理 121 条功能规范, 成功定位到 3 处异常信息。所有被测功能一致性测试结果如表 5 所示。实验结果表明通过本方法能够在节省测试软硬件资源的同时, 对 CAN 网络管理中的功能性接口、上下层交互以及报文的周期性发送进行多场景化测试。为确保 AUTOSAR CAN 网络管理的安全性和稳定性提供了有效的解决方案。本研究以测试用例中所覆盖的有效功能规范为标准, 由文献[11]可知 AUTOSAR CAN 网络管理

共包含 145 条功能规范。通过本方法可以验证其中 121 条规范的正确性和符合性,能够确保 CAN 网络管理协议栈的功能在实际场景中按照规范要求进行操作和运行。

表 5 一致性测试结果

被测功能	测试序列	通过数	未通过数
初始化	5	5	0
PDU 数据	5	4	1
状态转换	12	12	0
被动唤醒	2	2	0
远程休眠	6	6	0
发送接收	8	8	0
启禁通信	3	2	1
重复报文	16	16	0
周期发送	10	9	1
合计	67	64	3
百分比/%	100	95.55	4.45

设文献[7]、文献[6]和文献[9]的一致性测试方法分别为 M_1 、 M_2 和 M_3 ,本方法为 M_4 。表 6 展示了本方法与这 3 种方法实验对比结果。由表 6 可知,本研究在测试覆盖率和异常定位方面均占优势。

表 6 不同方法测试对比结果

测试结果	M_1	M_2	M_3	M_4
测试序列	24	57	38	67
所覆盖规范数	55	97	72	121
覆盖率/%	37.93	66.90	49.65	83.45
Bug 异常数	1	2	1	3

3.2 实验分析

本节将对 4.1 节中异常定位过程进行具体描述。

测试禁用通信时,对于 TC_CANNM_0116 条测试用例,要求在禁用通信后禁止释放网络,具体测试流程如图 9 所示。

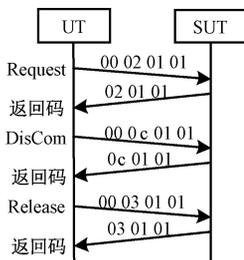


图 9 TC_CANNM_0116 测试流程图

步骤 1)UT 发送四字节测试码"00 02 01 01"。其各部

分表示标准测试用例、CanNm_NetworkRequest 接口编号、参数长度以及 01 网络通道。被测系统 SUT 在接收到该测试码后,会调用协议栈中 CanNm_NetworkRequest 接口,并将该"02 01 01"返回给上测试器 UT,通知 UT 唤醒成功。

步骤 2)UT 调用 CanNm_DisableCommunication,发送测试码为"00 0c 01 01"。在 SUT 中,返回码为"0c 01 01"表示禁用成功。

步骤 3)UT 调用 CanNm_NetworkRelease 接口释放网络。发送测试码"00 03 01 01",SUT 返回码为"03 01 01"。表示释放成功。

步骤 4)UT 中伪接口 CanNm_NetworkRelease 的期望值为"03 01 00",而步骤 3)中协议栈中 CanNm_NetworkRelease 接口的返回值是"03 01 01"。

步骤 5)经分析协议栈源代码发现 CanNm_NetworkRelease 接口异常,在第 572 行代码中缺少对禁用通知标志的 if 判断。违反了规范 SWS_CanNm_00294。

PDU 数据测试时,对于 TC_CANNM_0103 条测试用例,要求获得的报文是最后一次接收到的报文。具体测试步骤如图 10 所示。

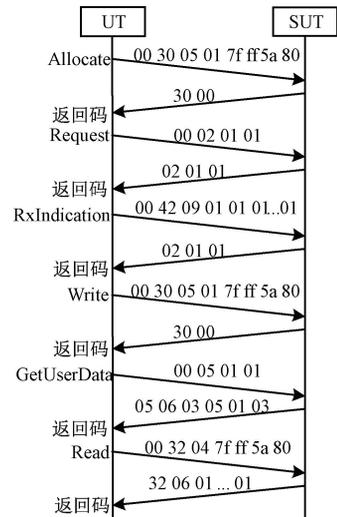


图 10 TC_CANNM_0103 测试流程图

步骤 1)UT 发送测试码"00 30 04 7f ff 5a 80",该测试码对应 UT 中 Test_Allocate 接口。其功能是在 SUT 中开辟一段缓冲区,此测试码前 3 个字节分别表示测试用例、接口编号和所获取的通道,最后 4 个字节表示在 SUT 中要分配的地址。之后,SUT 将返回"30 00"以通知上层测试器 UT 分配成功。

步骤 2)UT 发送"00 02 01 01"唤醒 01 网络通道,对应 CanNm_NetworkRequest 接口。SUT 在调用相应接口后会返回"02 01 01"。

步骤 3)UT 发送"00 45 09 01 01 01 01 01 01 01 01 01

01",对应 CanNm_RxIndication 接口,此测试码中后 6 个字节表示用户报文,返回码为"42 00"。

步骤 4)SUT 通过 Test_Write 接口更新步骤 1)分配的缓冲区数据,发送测试码"00 31 10 7f ff 5a 80 01 01 01 01 01 01"。返回码为"31 00",通知 UT 更新成功。

步骤 5)UT 发送"00 05 01 01",表示调用 CanNm_GetUserData 接口获取用户报文。返回码值为"05 06 03 05 02 01 00 01",其中后 6 个字节表示用户报文。

步骤 6)UT 发送"00 32 04 7f ff 5a 80"对应 Test_Read 接口,以读取步骤 4)中更新的数据。返回码值为"32 06 01 01 01 01 01 01",后 6 个字节表示 UT 在 Buffer 中更新的用户数据报文。

步骤 7)步骤 6)中的期望值与步骤 5)中的实际值不匹配,经分析协议栈源代码发现 CanNm_GetUserData 接口异常,定位到 CanNm_GetUserData 在第 859 行代码中返回的数据为 TxData 中的数据。违反了规范 SWS_CanNm_00294。

报文周期性发送测试中,从 Ready Sleep 状态到 Repeat Message 状态的测试时,需要以 100 ms 为周期发送 CAN 网络管理报文。具体测试流程如图 11 所示。

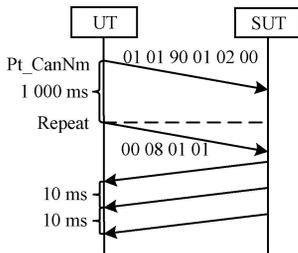


图 11 TC_CANNM_Repeat 测试流程图

步骤 1)UT 通过 pt_CanNm 端口发送网络管理测试报文"01 01 90 01 02 00",同时将发送时间点记录在时间戳变量 t_{start} 中。此 6 字节报文分别表示扩展测试用例、0x190CanId、硬件单元、控制器 Id 和被动唤醒。SUT 在接收到该网络管理测试报文后,将进入 Repeat Message 状态。

步骤 2)在时间戳 t_{start} 之后等待 1 000 ms,SUT 将进入到 Ready Sleep 状态。

步骤 3)UT 发送测试码"00 08 01 01",此测试码对应 CanNm_RepeatMessageRequest 接口,控制 SUT 重新进入 Repeat Message 状态。

步骤 4)UT 将接收到每条报文的时间点记录在时间戳数组中,检查相邻时间戳之间的时间间隔。发现相邻时间戳之间的时间间隔为 10 ms,期望值为 100 ms。经分析协议栈源代码发现在 1 718 行代码中将快发标志位置为 1,导致进入到了快发状态,此异常会导致总线带宽的浪费。违反了规范 SWS_CanNm_005。

3.3 性能分析

本研究的性能指标基于在预设的定时器配置下,确保

在 AUTOSAR CAN 网络管理规范所规定的容差范围内准确监测 CAN 网络管理报文,并实现从一个网络管理状态到另一个网络管理状态的转换。以报文发送定时器为例,AUTOSAR CAN 网络管理开发规范,即文献[11]规定 Time_ImmediateNM_Cycle(在重复报文状态下快速发送报文阶段的定时器)的默认配置时间为 50 ms,其容差范围为上下 10%;而 Time_NormalNM_Cycle(在正常操作状态下发送报文的定时器)的默认配置时间为 640 ms,容差范围为上下 5%。

通过多次实验的结果表明,本研究所提出的方法对于定时器 Time_ImmediateNM_Cycle 的监测时间稳定在 48 ms 至 51.5 ms 之间,而对于 Time_NormalNM_Cycle 的监测时间保持在 636 ms 至 643 ms 之间。这些结果表明,本研究的方法能够符合 AUTOSAR 规范的要求,确保对 CAN 网络管理报文的准确监测和网络管理状态的转换,为 CAN 网络管理协议栈在实际应用中的可靠性和稳定性提供可信赖的依据。

4 结 论

本研究提出一种基于 AUTOSAR CAN 网络管理的一致性测试方法,详细阐述了测试系统的体系结构、测试方案、测试通信报文设计格式,并通过巴科斯范式定义一种新的语法规则,为 TTCN-3 做实时性扩展,提高对 CAN 网络报文监测准确度。最后,在 MPC5745B 平台上对 AUTOSAR CAN 网络协议栈的初始化、PDU 数据、状态转换、被动唤醒、远程休眠、发送接收、启用禁用通信、重复报文和网络报文周期性发送等 9 个关键功能进行一致性测试。通过该方法,在节省测试软硬件资源的同时,有效提高了对 AUTOSAR CAN 网络管理协议栈的一致性测试深度。对于确保基于 AUTOSAR 规范网络管理的可靠性和稳定性具有重要意义。

参考文献

- [1] AUTOSAR GbR. Specification of network management interface R4.0.3[S]. AUTOSAR GbR,2013.
- [2] AutoSAR [EB/OL]. [2021-12-07]. <http://www.AutoSAR.org>.
- [3] 唐华亮,方红萍,谢斌.汽车仪表网络管理一致性测试系统设计[J].仪表技术与传感器,2019,433(2):66-71.
- [4] 高永凡,李超超,徐封杰,等. AUTOSAR 标准的 CAN 网络休眠异常诊断机制研究[J].计算机工程与应用,2024,60(7):248-257.
- [5] 陈灿,杨兴达,方菱.基于决策表的 AUTOSAR 操作系统一致性测试研究[J].计算机工程与科学,2023,

- 45(4):622-629.
- [6] 周毅,张晓先,陈丽蓉. AUTOSAR 符合性测试适配器的实现方法[J]. 计算机工程,2014,40(1):291-294.
- [7] LUO F, XIE Y. Conformance test of AUTOSAR network management [J]. International Journal of Wireless Information Networks, 2018, 25: 279-288.
- [8] BOSE A. Property-based testing: Evaluating its applicability and effectiveness for AUTOSAR basic software[D]. Stuttgart: University of Stuttgart,2020.
- [9] 一汽奔腾轿车有限公司. CAN 网络 AUTOSAR 网络管理参数测试方法及系统:CN202310713572.3[P]. 2023-09-05.
- [10] AUTOSAR GbR. Conformance test specification of CAN network management R4.0.3[S]. AUTOSAR GbR,2013.
- [11] AUTOSAR GbR. Specification of CAN Network Management R4.0.3[S]. AUTOSAR GbR,2013.
- [12] GRABOWSKI J, HOGREFE D, RÉTHY G, et al. An introduction to the testing and test control notation (TTCN-3)[J]. Computer Networks, 2003, 42(3): 375-403.
- [13] 蒋凡,范秀萍. TTCN-3 语言中基于过程通信的编译器实现[J]. 计算机工程,2016,42(8):9-13.
- [14] DAI Z R, GRABOWSKI J, NEUKIRCHEN H. Timed TTCN-3—a real-time extension for TTCN-3[C]. Berlin: Springer US, 2002.
- [15] SERBANESCU D A, MOLOVATA V, DIN G, et al. Real-time testing with TTCN-3 [C]. International Workshop on Formal Approaches to Software Testing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008: 283-301.
- [16] 董宏成,张宁,牛勇清,等. TD-LTE 系统 RRC 一致性测试设计与 TTCN-3 验证[J]. 计算机应用研究,2014, 31(8):2414-2417.

作者简介

张建,硕士研究生,主要研究方向为车载软件测试技术。

E-mail: Q21301398@stu.ahu.edu.cn

王辉,硕士研究生,主要研究方向为车载软件测试技术。

E-mail: Q22201340@stu.ahu.edu.cn

方菱(通信作者),副研究员,博士,主要研究方向为嵌入式系统安全与测试。

E-mail: fangl@hfcas.ac