第 44 卷 第 12 期 计 算 机 工 程 2018 年 12 月
Vol. 44 No. 12 Computer Engineering December 2018

·移动互联与通信技术· 文章编号：1000-3428(2018)12-0120-09 文献标志码：A 中图分类号：TP393

STP 安全通信协议设计与形式化验证

李 堃[1]，张雪松[2]

（1. 中国铁道科学研究院通信信号研究所，北京 100081；2. 中国铁道科学研究院电子信息研究所，北京 100081）

摘 要： 无线调车机车信号和监控系统（STP）是基于无线数传电台，实现车载和地面设备之间双向信息传输的实时信号监控和安全防护系统。为保证系统中车-地之间交互信息的实时性、可靠性与完整性，依据欧标 EN50159，在现有 ETCS 安全通信协议 EuroRadio 的基础上，通过增加安全连接超时重发、双序号时间戳和故障导向安全机制，设计一套适用于 STP 系统的安全通信协议，并利用分层着色 Petri 网和 ASK-CTL 时序逻辑验证语言对其进行建模和形式化验证。分析结果表明，该协议不仅满足功能安全性要求，同时还能保证系统在非理想信道环境下的故障导向安全。

关键词： 安全通信协议；时序逻辑；分层着色 Petri 网；ASK-CTL 形式化验证；故障导向安全

中文引用格式： 李 堃，张雪松. STP 安全通信协议设计与形式化验证[J]. 计算机工程，2018，44(12)：120-128.

英文引用格式： LI Kun，ZHANG Xuesong. Design and formal verification of safety communication protocol in STP[J]. Computer Engineering，2018，44(12)：120-128.

Design and Formal Verification of Safety Communication Protocol in STP

LI Kun[1]，ZHANG Xuesong[2]

（1. Institute of Communication Signals，China Academy of Railway Sciences，Beijing 100081，China；
2. Institute of Electronic Information，China Academy of Railway Sciences，Beijing 100081，China）

【Abstract】 The STP system is based on data-transmission radio. By the data transmission between the mobile equipment and the ground equipment，it can realize the real-time monitoring of train signal，which is used to ensure the safety protection of the train. To make sure that the information，which transmits by wireless to be real-time，reliable and complete，this paper designs a safety communication protocol for STP. According to En50159，it adds some extra safety measures，such as time-out，sequence-number and so on in STP safe communication protocol，which is based on EuroRadio，to ensure the communication system to be fail-safe. And it uses hierarchical Colored Petri Net（CPN） and ASK-CTL formula to build hierarchy models and make formal verification of the proposed protocol. Analysis results show that the proposed protocol for STP has the functional safety，and it also can be fail-safe，even though the transmission happens during the undesired channel.

【Key words】 safety communication protocol；temporal logic；hierarchical Colored Petri Net（CPN）；ASK-CTL formal verification；fail-safe

DOI： 10.19678/j.issn.1000-3428.0049260

0 概述

近年来，随着编组站内调车作业业务量的不断增大，为减少由于人为操作失误而造成的“冲、挤、脱”等事故的发生，无线调车机车信号和监控系统（STP）应运而生[1]。该系统是一种可有效解决上述问题并保证站内调车作业安全的控制系统[2]。当调车机车进入调车作业区时，机车接收到点式应答器的地面入网信息，并通过无线信道将该信息发送给系统的地面设备注册入网。地面设备收到入网申请后，向该机车发送确认注册信息，并分配给机车一个入网注册号，建立安全控制信息通道，系统进入调车监控状态。此后，地面设备根据调车机车所在的位置及开放的调车信号，通过无线信道向作业的调车机车发出控制命令。机车接收到控制命令后，将车列的确切位置、机车速度、设备的工作状态等有关信息送回给地面设备，实现系统的闭环控制[3]。

由此可见，安全可靠的车-地之间通信是保证

基金项目： 中国铁道科学研究院重点课题项目“基于新一代无线移动通信的调车机车控制技术研究”（2016YJ054）；中国铁路总公司科研重点课题项目“通信信号设备提升关键技术研究-STP 互联互通技术要求研究”（2017X012-B）。

作者简介： 李 堃（1993—），女，硕士研究生，主研方向为交通信息工程及控制、通信工程；张雪松，研究员。

收稿日期： 2017-11-10 **修回日期：** 2017-12-16 **E-mail：** likun_0106@126.com

STP 系统高效运行的关键所在。为此,本文设计一种基于 STP 车-地无线通信系统的安全通信协议,以保证传输信息的实时性、可靠性和完整性。由于国内对于 STP 系统安全通信协议的研发仍处于起步阶段,因此本文同时借鉴 ETCS 安全通协议 EuroRadio 接口规范[4]并对其加以创新,设计符合 STP 系统通信需求的安全通信协议。

1　STP 安全通信协议设计

STP 是基于车-地之间无线数传信道传输交互信息,进而形成闭环控制的安全防护系统。依据标准 EN50159 中给出的不同传输系统的分类分析 STP 车-地之间传输系统的特征,该通信系统为第 3 类开放式传输系统。EN50159 还给出了不同类型传输系统可能存在的通信风险,以及针对不同风险推荐的防御手段[5]。表 1 为常见的威胁-防御矩阵,其中 × 表示存在的威胁与相应的防御措施之间的对照关系。据此,在 STP 安全通信协议设计中采用序列号、超时检测、源目的标识、认证过程、安全码等防护措施,并借鉴 ETCS 安全相关系统结构[6],得出 STP 车-地无线通信系统结构,如图 1 所示。

表 1　常见威胁-防御矩阵

威胁	防御措施							
	序列号	时间戳	超时防护	源和目的标识	消息反馈	认证流程	安全码	密码术
重复	×	×						
删除	×							
插入	×			×	×	×		
重排序	×	×						
损坏							×	×
超时		×	×					
伪装					×	×		×

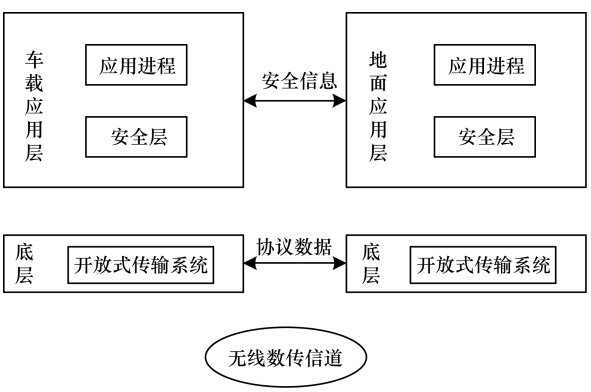


图 1　STP 车-地无线通信系统结构

1.1　序列号

STP 安全通信协议采用双序号时间戳防护机制[7],以保证车-地传输数据的实时性、完整性和有效性,即在发送数据时,协议将为其添加 2 Byte 的序列号。如用 *LS*、*LR* 分别代表车载设备的本方序列号和对方序列号(地面设备序列号),且各占 1 Byte。当车载设备每发送一包数据时,协议将会在车载接收到的上一包数据中提取本方序列号 *LS*,并加 1,对方序列号 *LR* 不变。地面设备在发送数据时以同样的方式处理序列号。

对于双方接收到的数据包,以车载设备为例进行序列号有效性检测说明。当车载设备接收到新数据包后,提取双序列号:地面本方序列号 *DS* 和地面的对方序列号 *DR*。通过 if ($DS = LR + 1$)语句,判断接收数据包的连续性,即检测是否出现了数据包删除、插入、乱序等问题;以 if($LS - DR < = 4$)语句保证接收到的数据包是在允许延时范围内的。在 STP 车-地通信系统中,地面以 500 ms 为周期发送控制数据给车载。当一个轮询 2 s 内,即 4 个周期内接收不到车载设备的回执时,认为数据通信超时、无效,应当丢弃[8]。

1.2　超时检测

在周期性信息传输系统中,接收信息时协议会检测 2 个消息之间的时间间隔是否超过事先确定的最大时间间隔。若超过,则信息的时效性得不到保证,就有可能会给系统带来危害。

因此,本文在 STP 安全通信协议设计过程中采用了 2 种超时检测机制。一种是数据传输过程中的双序号时间戳检测机制,该机制已在 1.1 节中给出,这里不再赘述;另一种则是安全连接建立阶段的超时重发机制。当调车机车进入调车作业区,经过入网应答器时,车载设备应用进程发起安全连接,请求地面设备允许其注册入网,此时,启动车载设备安全层的安全连接计时器。如果在规定时间内,未收到地面的确认安全连接请求信息,即分配车载入网注册号信息时,计时器溢出,认为安全连接超时。在 STP 安全通信协议中,安全连接计时器超时设定为6 s。当计时器溢出时,安全层检测出车载与地面设备之间通信超时,便会在车载设备报警之前(6.5 s)自动做出响应,向应用进程发送错误报告,保证在 STP 车-地之间通信故障时及时导向安全,并请求重新发起安全连接,从而提高系统通信可靠性。

1.3 源和目的标识

STP 车-地通信系统是一个多端通信系统。因为在大型编组站内,通常会有多台地面设备控制不同站场内的多台调车机车同时作业,所以在使用所接收到的信息之前,需要用合适的方法来检测信息的来源。

对于车-地多端通信系统而言,协议为每一个通信端都定义了一个区别于其他设备的唯一标识符。在传输信息时,每一包数据都携带了一个唯一的源标识和一个唯一的目的标识[9]。接收端通过接收到的消息中的源和目的标识判断该消息是否从既定方发出,以及是否是发给本方,以此来检验消息的合法性。

1.4 认证机制

STP 安全通信协议借鉴 EuroRadio 的 2 种安全认证机制,一种是安全连接建立阶段的对等实体认证机制,另一种是数据传输阶段的消息来源认证机制[10],其安全功能参见文献[6],限于本文篇幅,此处不再赘述。

1.5 安全码

在 STP 车-地开放式传输系统中,使用安全码的目的是为了检测出信息中的错误,以此来保证信息的完整性[11]。在本文协议中采用了 CCITT 建议的 CRC 方式,校验结果置于 CRC 和 CRC2 中,其生成多项式为:$G(X)=X^{16}+X^{12}+X^5+1$。

2 安全通信协议建模

2.1 安全通信协议时序图

安全通信协议的核心是对非安全传输系统的消息进行安全处理,通过不同功能模块完成各自的安全通信功能,然而每个功能模块的核心又是通过管理相应的消息收发时序实现其功能的[12]。STP 安全通信协议的设计借鉴了现有成熟 EuroRadio 的功能结构,并在其基础上加以创新,得到适用于 STP 车-地通信,且功能完整的安全通信协议。图 2 为 EuroRadio 安全通信协议连接建立阶段、数据传输阶段、断开连接阶段 3 个安全过程的时序图。图 3(a)~图 3(c)对应 STP 安全通信协议 3 个同样安全过程的时序图,图 3(d)则对应 STP 安全通信协议新增连接建立超时,故障导向安全功能过程的时序图。

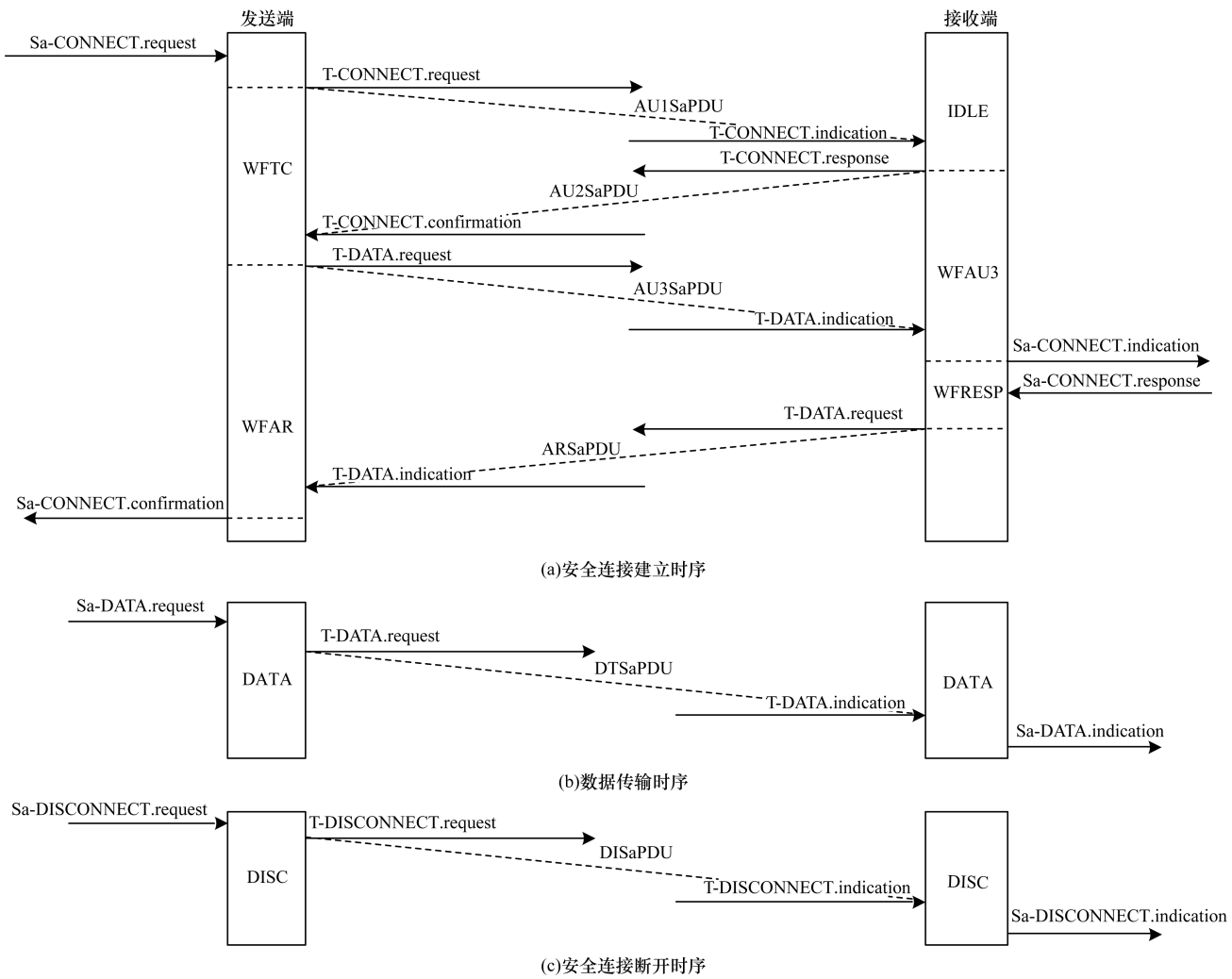


图 2 EuroRadio 协议时序图

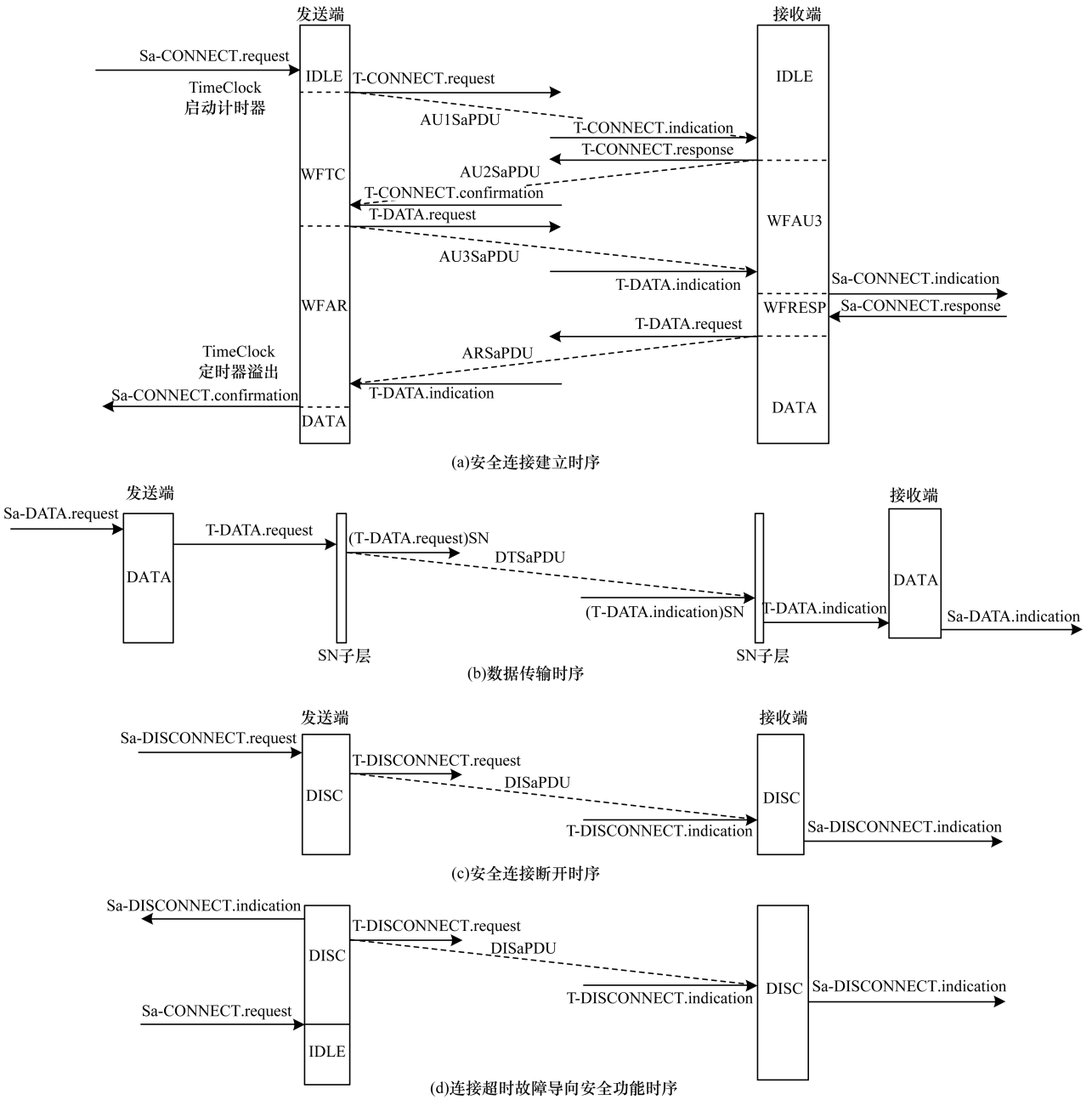


图 3　STP 安全通信协议时序图

图 3(a)所示为安全连接建立阶段,协议在 EuroRadio 安全功能的基础上,新增了一个计时器功能,用以检测安全连接请求自发起之时到计时器溢出,协议是否完成了安全连接。

图 3(b)所示为安全数据传输过程,协议新增序列号 SN 子层(序列号功能子层)。当数据发送方发送数据,经过 SN 层时,SN 层会在本方原有序列号基础上自增 1,但不改变对方序列号。随后 SN 层将更新后的双序列号添加到发送数据中,作为本方用户数据一并发送给对方。当接收方接收到数据时,SN 层首先会对接收数据所携带序列号的时序性进行检测,以保证发送给顶层应用进程的数据具有时效性和完整性。双序号时间戳安全防护机制的具体内容可参见 1.1 节。由于序列号安全防护机制的过程相对复杂,鉴于本文篇幅有限,其具体模型建立与模型功能验证本文不做讨论。

图 3(c)为断开安全连接阶段,其为应用进程主动发起断开安全连接请求场景。如调车机车完成了站内调车作业后,不再需要接收地面发来的控制命令时,则向地面主机发起退网申请。该申请由机车主机应用进程发起,经过机车安全层,将断链请求发送至地面设备。如图 3(c)所示,通过 DISCONNCET 类4 种安全服务原语,分别断开机车主机、地面主机各自安全层与应用进程间的安全

连接,以及车载和地面设备对等实体之间的安全连接,使协议回到初始状态,等待下次的安全连接,即机车入网申请的再次发起。

图 3(d)为 STP 安全通信协议新增故障导向安全功能,即:在安全连接建立阶段,当安全层计时器溢出后,安全层检测到对等实体间仍没有成功建立安全连接时,视为故障;在数据传输阶段,接收到的数据首先经安全层新增 SN 子层的序列号检测,如果检测不满足 1.1 节中给出的防护机制,视为数据超时失效、故障。对应上述 2 种情况下的故障,安全层均做出响应,分别向本方应用进程和对方安全层发送 Sa-DISCONNECT. indication 和 T-DISCONNECT. request 2 种安全服务原语,断开机车和地面之间的安全连接,跳转到协议初始状态。随后,发起方应用进程便会自动重发安全连接。

2.2 安全通信协议 Petri 模型

本文采用着色 Petri 网(Colored Petri Net,CPN),依据“自顶向下”的建模规则建立 STP 安全通信协议分层模型。采用 CPN 建模的优点在于,它不仅可以利用库所(place)、变迁(transition)和弧(arc)的连接表示系统的静态结构,还可以通过变迁的触发和库所内令牌(token)的迁移,描述系统的动态行为[13]。同时 CPN 作为一种高级 Petri 网,被认为是基于网络复杂系统建模和分析的最佳工具[14]。用它建立的模型不但可执行,同时还有利于形式化验证和动态仿真。

图 4 显示了 STP 安全通信协议的模型层次结构,其中每一个节点代表系统中的一个模型。结构中共包含了 8 个模型。除车载和地面安全层子模型和 2 个计时器子模型外,其他模型均为安全通信协议运行的外部环境模型。而建立完备的外部环境模型的目的在于,保证所设计的安全通信协议在满足 EN50159 标准的基础上,方便验证其安全功能是否符合 STP 车-地之间通信需求,其动态仿真性能参数是否满足相应安全等级。

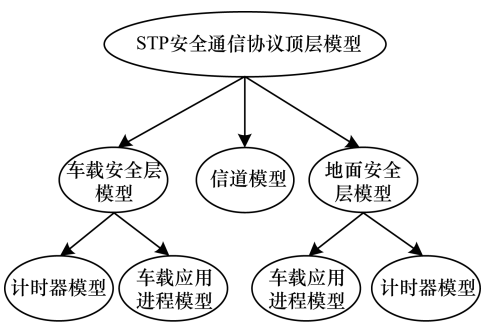


图 4 STP 安全通信协议 Petri 模型架构

在图 5 所示的顶层模型中,替代变迁(substitution-transition)Train 和替代变迁 Ground 分别对应机车与地面的安全功能层。图 6、图 7 则给出了安全通信协议的安全功能执行过程,包含从安全连接成功建立后自动进入数据传输阶段,和机车完成站场内调车作业后主动申请退网断开连接的通信过程,其时序关系与图 3(a)~图 3(c)所示的时序图保持一致。在车载和地面安全层模型中,库所 SafeStatus 和库所 Safestatus中包含的令牌值均为:IDLE | WFTC | WFAR | WFAU3 | WFRESP | DATA,依次代表其安全层的不同状态,即:初始状态 | 等待连接状态 | 等待确认状态 | 等待 AU3 原语状态 | 本方已建立安全连接等待对方打开数据链路状态 | 安全数据传输状态。图 6 和图 7 所示模型中的所有变迁分别对应安全层即将进入下一状态之前会被触发的不同事件。一旦某一变迁(事件)在某一时刻被触发,它将决定该时刻向本方应用进程和对方安全层发送什么类型的服务原语,以及将本方安全层置于相应什么状态。从而保证安全层不同状态之间的连续自动转换,和安全功能的连续有效执行。

图 8 给出了车载安全层的计时器模型。由于车载与地面安全层计时器功能一样,因此本文以车载为例,仅给出车载安全层计时器模型。如图 8 所示,库所 T 的初始令牌值为 6,它表示安全连接最大超时时间为 6 s。库所 FULL 代表计时器超时溢出状态,因此,当库所 FULL 中包含令牌值时,计时器模型此时便会检测库所 Status 中的令牌值是否为 DATA。如果是,则表示安全连接已在 6 s 内成功建立,安全层此时不发送错误报告;如果库所 Status 中的令牌值是除 DATA 外的其他任一值时,均认为安全连接超时,那么变迁 ReLink 此时将会被触发,向本方应用进程发送连接错误报告,并申请重新发起安全连接,保证系统故障导向安全,其时序关系与图 3(d)连接超时故障导向安全时序逻辑保持一致。

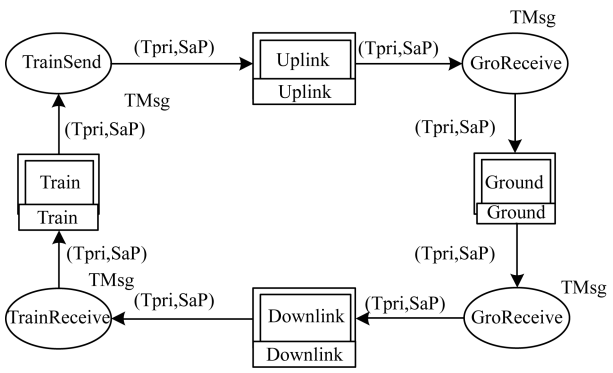


图 5 STP 安全通信协议顶层模型

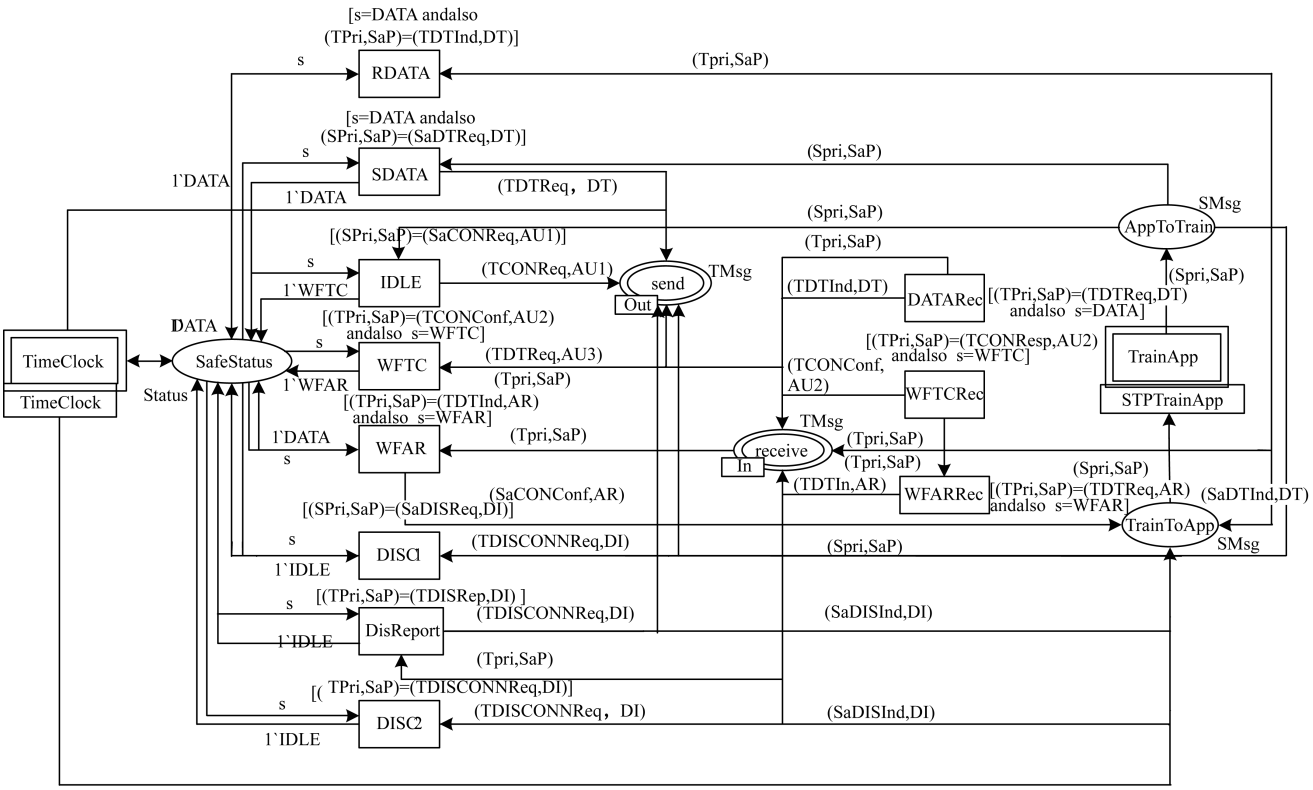


图 6　车载安全层模型

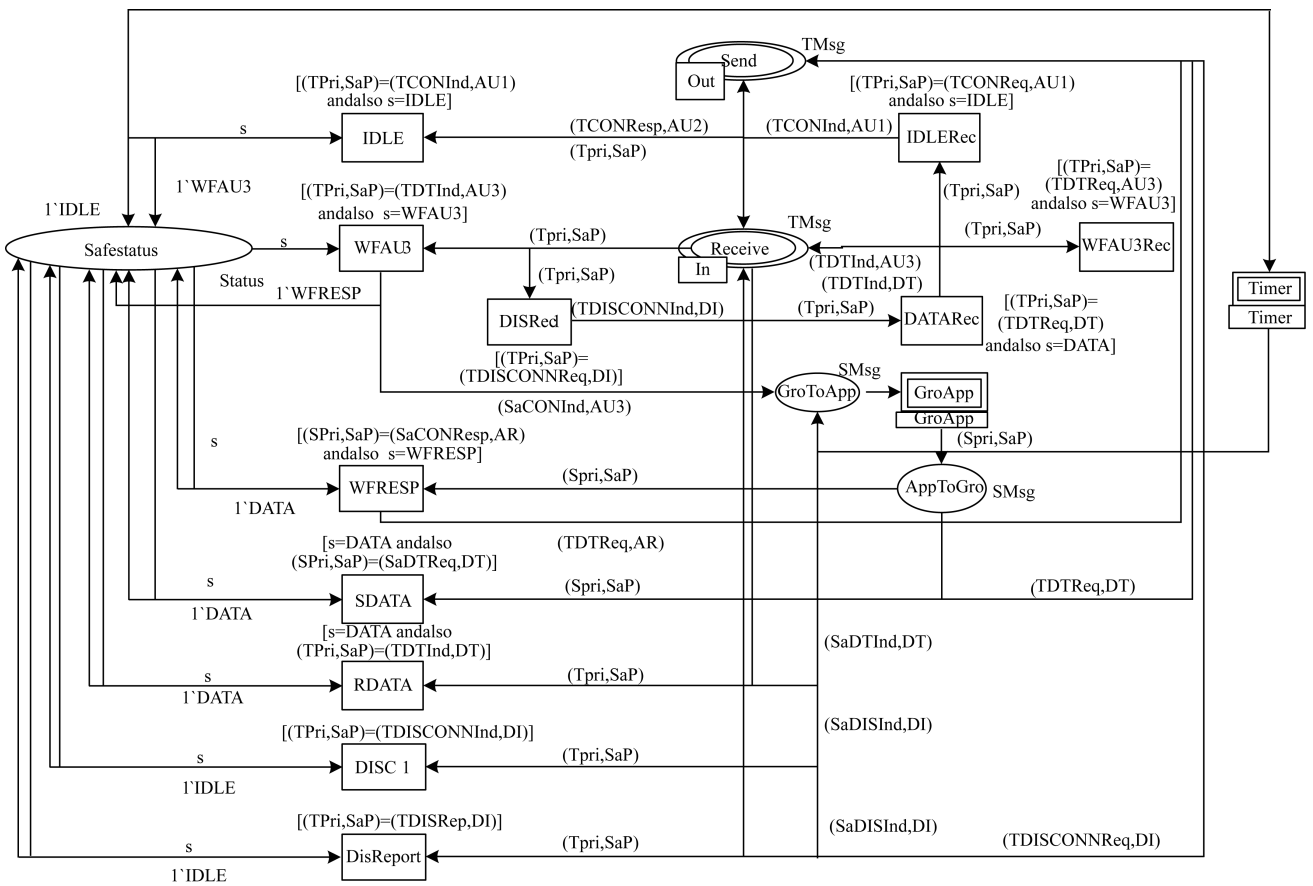


图 7　地面设备安全层模型

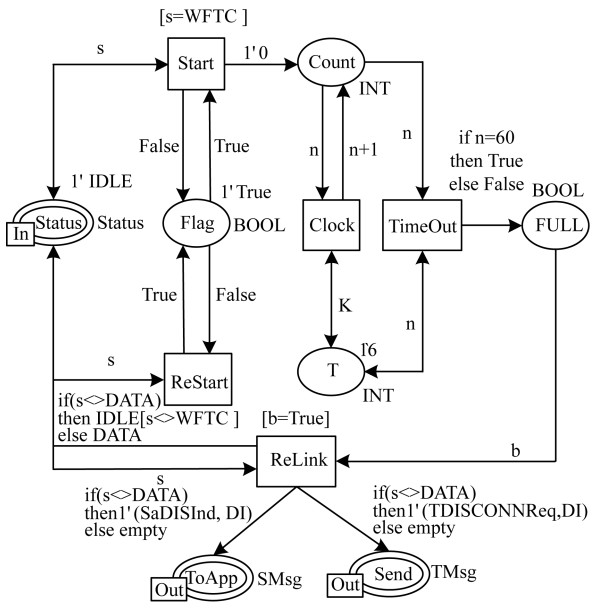


图 8 安全层计时器模型

3 基于 ASK-CTL 的形式化验证

安全通信协议功能的正确性取决于协议能否做出正确响应,且以正确的顺序响应事件。而协议的逻辑表示的是状态转换的过程,因此,安全通信协议的功能验证,即验证协议是否以正确的事件发生次序到达期望状态。时序逻辑是与模型检测紧密相关的概念,用来刻画有限状态系统中事件发生的时序关系。

据此,本节首先利用 CPN. Tools 建模工具生成第 2 节中给出的 Petri 模型的状态空间可达图,同时将待验证的安全功能性质表达成形式化时序逻辑描述语言:ASK-CTL 公式。最后通过状态空间搜索算法检验模型状态空间可达图上是否存在满足此性质的状态节点。结果为真,则证明安全通信协议满足该性质;反之,不满足,找出原因,并对安全通信协议做出相应改进。

3.1 ASK-CTL 相关定义

计算树时序逻辑(Computational Tree Logic, CTL)是一种典型的分支时序逻辑,是被作为描述有限状态系统的一种声明语言[15]。在描述系统状态转换的性质方面,CTL 具有很强的规范描述逻辑。它用时态算子描述沿计算路径的状态变化,用路径算子解释时间分支性质[16]。而 ASK-CTL 仅是 CTL 逻辑的一种扩展,因此与 CTL 具有很多共同的时序逻辑。

ASK-CTL 状态公式的语法如下:

$$A::=tt\mid\alpha\mid A\mid A1\vee A2\mid <B>EU(A1,A2)\mid AU(A1,A2)=$$
$$tt\mid\alpha\mid NOT(A)\mid A1\cup A2\mid <B>EU(A1,A2)\mid AU(A1,A2)$$

为构造复杂的系统时序逻辑表达式,ASK-CTL 还有以下扩展操作符:

POS A = EU(tt, A) = EXIST_UNTIL(tt, A):表示存在一条路径,可以到达一个状态,使得 A 成立;

INV A = POSA = NOT(POS(NOT A)):表示没有路径可以达到一个状态,使得 A 的反命题成立,即对于每一个可达状态,A 成立;

EV A = AU(tt, A) = FORALL_UNTIL(tt, A):表示对于所有路径,在有限步内使 A 成立;

ALONG A = EVA = NOT(EV(NOT A)):表示存在一条路径在有限步内使 A 的反命题成立,即存在一条无限路径,路径中的每个状态都使得 A 成立。

3.2 形式化验证

通常系统待验证的目标属性可分为两大类,即可操作性和安全性[17]。所谓可操作性,即系统期待的状态最终会出现;而安全性描述的则是危害系统的坏事情永远不会发生的属性。

结合 STP 安全通信协议所具备的安全功能属性,本文将协议待验证的属性归纳为:1)安全连接建立过程的可操作性;2)数据传输过程的可操作性;3)安全连接建立过程的安全性;4)数据传输过程的安全性;5)故障导向安全属性。前 4 个属性的验证是基于车载安全层与地面安全层 Petri 模型实现的,其模型对应图 6 和图 7。此时的验证不包含对图 8 所示的连接超时计时功能模型的检验。而最后一个属性的验证则是建立在完备安全通信协议模型的基础上实现的,即在图 6 ~ 图 8 所示原有安全层模型上添加连接超时检测子模型,具体过程如图 9 所示。

```
use (ogpath^"/ASKCTL/ASKCTLloader.sml")
fun LinkState n=(Mark.STPTrain'SafeStatus 1 n=[DATA] andalso
Mark.STPGround'Safestatus 1 n=[DATA]);
val LinkState = PredAllNodes (fn n =>(LinkState n));

fun LinkTrain n =(Mark.STPTrainApp'AppStatus 1 n  = [CONED]);
fun LinkGround n =(Mark.STPGroundApp'Appstatus 1 n = [CONED]);

val myASKCTLformula =AND(POS(MODAL(NF("Train is normal",LinkTrain))),
                         POS(MODAL(NF("Ground is normal",LinkGround))));


fun verify n= eval_arc myASKCTLformula n ;
val results = map verify LinkState;
let
val fid=TextIO.openOut"The establishment of the protocol.txt"
val _ = if(cf(true,results)>0)
then TextIO.output(fid,"The connection establishment is normal")
else TextIO.output(fid,"The connection establishment is abnormal")
in
TextIO.closeOut(fid)
end
```

```
val LinkState = fn : Node -> bool
val LinkState =
  [9999,9998,9997,9996,9995,9994,9993,9992,9991,9990,999,9989,9988,9987,9986,
   9985,9984,9983,9982,9981,9980,998,9979,9978,9977,9976,9975,9974,9973,9972,
   9971,9970,997,9969,9968,9967,9966,9965,9964,9963,9962,9961,9960,996,9959,
   9958,9957,9956,9955,9954,9953,9952,9951,9950,995,9949,9948,9947,9946,9945,
   9944,9943,9942,9941,9940,994,9939,9938,9937,9936,9935,9934,9933,9932,9931,
   9930,993,9929,9928,9927,9926,9925,9924,9923,9922,9921,9920,992,9919,9918,
   9917,9916,9915,9914,9913,9912,9911,9910,991,9909,...] : Node list
val LinkTrain = fn : Node -> bool
val LinkGround = fn : Node -> bool
val myASKCTLformula =
  NOT
    (OR
       (NOT (EXIST_UNTIL (TT,MODAL (NF ("Train is normal",fn)))),
        NOT (EXIST_UNTIL (TT,MODAL (NF ("Ground is normal",fn)))))) : A
val verify = fn : Arc -> bool
val results =
  [true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
   true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
   true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
   true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
   true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
   true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
   true,true,true,true,true,true,true,true,true,true,true,...] : bool list
```

图 9 安全连接建立过程可操作性验证

如图 9 所示,安全连接建立过程的可操作性形式化验证步骤如下:

步骤 1　函数 LinkTrain 和函数 LinkGround 分别描述机车和地面设备应用层被期望到达的状态[CONED],即安全连接成功建立状态。用函数 LinkState 描述车载与地面安全层被期望到达的数据链路打开状态[DATA]。

步骤 2　利用 ASK-CTL 公式 myA SKCTLformula 描述到达该期望状态的时序逻辑。

步骤 3　利用 eval_arc myASKCTLformula 状态空间查询语句,验证在其状态空间内是否存在满足该性质的节点。

图 9 中的验证结果表明,其状态空间内存在多个满足该性质的节点。即协议期待的安全连接成功建立状态可达,安全连接过程具有可操作性。

图 10 为数据传输过程可操作性验证过程,其验证步骤与安全连接建立过程可操作性属性验证步骤一致,这里不再赘述。验证结果表明,安全通信协议建立安全连接后,最终会进入数据传输阶段,使 STP 车-地之间的数据安全可靠传输,因此,满足此阶段的可操作性。

```
fun initial n = (Mark.STPTrain'TrainToApp 1 n =[]);
fun final n = (Mark.STPTrain'TrainToApp 1 n =[(SaDTInd,DT)]);
val myASKCTLformula =INV(OR(NF("initial",initial),POS(NF("final",final))));

fun verify n= eval_node myASKCTLformula n ;
val results = verify InitNode;
let
val fid=TextIO.openOut"The operation during the data transfer.txt"
val _ = if(results=true)
then TextIO.output(fid,"The data transfer is normal")
else TextIO.output(fid,"The data transfer is abnormal")
in
TextIO.closeOut(fid)
end

val initial = fn : Node -> bool
val final = fn : Node -> bool
val myASKCTLformula =
  NOT
    (EXIST_UNTIL
      (TT,NOT (OR (NF ("initial",fn),EXIST_UNTIL (TT,NF ("final",fn))))))) : A
val verify = fn : Node -> bool
val results = true : bool
val it = () : unit
```

图 10　数据传输过程可操性验证

安全连接建立过程和数据传输过程的安全性验证如图 11 和图 12 所示。验证结果表明 STP 安全通信协议在上述 2 个过程中,均满足功能安全性。即在信道环境理想的条件下,协议不会进入非安全状态,其状态空间不存在死节点。

```
fun safe n = (Mark.STPTrain'SafeStatus 1 n =[DATA]
              andalso Mark.STPTrainApp'AppStatus 1 n =[CONED]
              andalso Mark.STPGround'Safestatus 1 n =[DATA]
              andalso Mark.STPGroundApp'Appstatus 1 n =[CONED]);

val myASKCTLformula = EV(NF("safe",safe));
eval_node myASKCTLformula InitNode;


val safe = fn : Node -> bool
val myASKCTLformula = FORALL_UNTIL (TT,NF ("safe",fn)) : A
val it = true : bool
```

图 11　安全连接建立过程安全性验证

```
fun start n =(Mark.STPTrain'TrainToApp 1 n =[]
              andalso Mark.STPGround'GroToApp 1 n =[]);
fun goal n =(Mark.STPTrain'TrainToApp 1 n =[(SaDTInd,DT)]
              orelse Mark.STPGround'GroToApp 1 n =[(SaDTInd,DT)]);
val myASKCTLformula=INV(OR(NF("start",start),EV(NF("goal",goal))));


eval_node myASKCTLformula InitNode;
val start = fn : Node -> bool
val goal = fn : Node -> bool
val myASKCTLformula =
  NOT
    (EXIST_UNTIL
      (TT,NOT (OR (NF ("start",fn),FORALL_UNTIL (TT,NF ("goal",fn)))))) : A
val it = true : bool
```

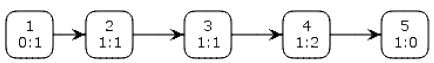
图 12　数据传输过程安全性验证

然而上述的验证均是在信道理想状态,即信道不存在干扰和丢包等情况下的结果,仅保证安全通协议自身具有安全功能属性。因此,为验证协议在非理想信道环境下也能保证 STP 车-地间通信系统的安全性,本文在图 4 所示的信道子模型内引入一定丢包率,造成了数据包在无线信道传输过程中的丢失现象。在此基础上,本文再次验证了上述2 个过程的安全性,验证结果如图 13 所示。结果中的“false”表明,此时协议不再满足连接建立过程的安全性。又通过状态空间可达树和 CPN ML 语句:OutArcs5 = [ ],得出状态节点 5 为“死节点”,即协议非安全状态。在节点 4 至节点 5 的变迁绑定(Binding)中,可以得知 Trans = FALSE,由于信道的丢包现象,造成安全连接建立过程的连接数据包传输失败,出现了非安全状态。数据传输过程的非安全状态节点出现原因与上述同理。由此可见,在不包含连接超时计时子模型的安全通信协议中,其安全功能仅能保证系统在理想信道环境下的安全通信。一旦引入信道丢包率,协议在连接建立过程便会出现非安全状态。

```
fun safe n = (Mark.STPTrain'SafeStatus 1 n =[DATA]
              andalso Mark.STPTrainApp'AppStatus 1 n =[CONED]
              andalso Mark.STPGround'Safestatus 1 n =[DATA]
              andalso Mark.STPGroundApp'Appstatus 1 n =[CONED]);

val myASKCTLformula = ALONG((NF("safe",safe)));
eval_node myASKCTLformula InitNode;

val safe = fn : Node -> bool
val myASKCTLformula = NOT (FORALL_UNTIL (TT,NOT (NF ("safe",fn)))) : A
val it = false : bool
```




```
OutArcs 5 = [];        val it = true : bool

4:4->5 TOP'Uplink 1: {TPri=TCONReq,LS=0,LR=0,SaP=AU1,Trans=FALSE}
```

图 13　非理想信道下协议安全性验证

为使上述“非安全状态”导向“安全状态”,即故障导向安全,本文在图 6 ~ 图 8 现有安全功能基础上,新增替代变迁 TimerClock 和 Timer,均对应图 6 所示计时器子模型,并在同样的非理想信道环境下,验证改进后的安全通信协议连接建立过程的安全性,其 ASK-CTL 验证结果如图 14 所示。验证结果中的“true”值表明新增连接超时重发机制的功能安

全性,保证了改进后的协议在安全连接建立阶段,即使信道存在丢包现象,导致在一定时间限度内接收不到下一包连接数据的通信故障状态下,也能够导向安全侧,从而提高了 STP 车-地之间通信的安全性和可靠性。

```
fun count n= (Mark.Timerclock'Count 1 n = [6]);

fun dis n = (Mark.STPTrainApp'TrainAppIn 1 n = [(SaDISInd,DI)]);;

fun data n = (Mark.Timerclock'Status 1 n =[DATA]);

val myASKCTLformula=EV(OR(AND(NF("count",count),NF("security1",dis)),
                          AND(NF("count",count),NF("security2",data))));
eval_node myASKCTLformula InitNode;

val count = fn : Node -> bool
val dis = fn : Node -> bool
val data = fn : Node -> bool
val myASKCTLformula =
  FORALL_UNTIL
    (TT,
     OR
       (NOT (OR (NOT (NF ("count",fn)),NOT (NF ("security1",fn)))),
        NOT (OR (NOT (NF ("count",fn)),NOT (NF ("security2",fn)))))) : A
val it = true : bool
```

图 14 故障导向安全性验证

4 结束语

本文依据欧标 EN50159,在现有成熟 ETCS 安全通信协议 EuroRadio 安全功能的基础上,新增双序号时间戳、连接超时重发和故障导向安全防护机制,设计了一个适用于 STP 车-地之间通信的安全协议。利用层次 Petri 网(CPN)对该协议建立模型,通过 ASK-CTL 时序逻辑对协议进行形式化验证。仿真结果表明,无论是在信道理想情况还是在有一定信道干扰条件下,该通信协议均具有功能安全性且其安全功能完整。后续工作将在本文 Petri 模型的基础上添加时间元素,即模型时间戳,并对所设计的 STP 安全通信协议的时间性能做进一步仿真分析。

参考文献

[1] 申旭光,郭其一,成 远. 无线调车机车信号和监控系统车地通信的模型设计[J]. 机车电传动,2007(5):34-36.

[2] 曹桂均,林通源. 调车机车信号和监控系统[J]. 铁路工程,2006,3(2):7-10.

[3] 林炳跃. 基于 UPPAAL 的无线调车机车信号和监控系统的建模与分析[D]. 北京:中国铁道科学研究院,2009.

[4] ERTM S. EuroRadio FIS:class 1 requirements:SUBSET-037V225[S]. 2003.

[5] CENELE C. Railway application-communication, signaling and processing systems-part2:safety-related communication in open transmission systems:EN50159-2[S]. 2001.

[6] 陈黎洁,单振宇,唐 涛. 列车运行控制系统中安全通信协议的形式化分析[J]. 铁道学报,2012,34(7):70-76.

[7] 单振宇. CTCS-3 级车地通信协议设计与验证[D]. 北京:北京交通大学,2009.

[8] 曹桂均,冯 军,杨华昌. 无线调车机车信号和监控系统暂行技术规范:TJ/DW035-2014[S]. 2014.

[9] 胡晓辉,陈慧丽,石广田,等. CTCT-4 级安全通信协议的形式化建模与验证[J]. 计算机工程与应用,2014,50(4):81-85.

[10] 于宏博,钟章队. ETCS 中安全通信的研究与探讨[J]. 铁道通信信号,2005,41(1):30-33.

[11] 杨 明,邱昕夕,陈祥献,等. CBTC 中安全通信协议的设计与仿真研究[J]. 计算机工程,2012,38(19):241-246.

[12] 陈黎洁. 列车运行控制系统安全通信协议验证方法的研究[D]. 北京:北京交通大学,2013.

[13] JENSEN K. A brief introduction to colored Petri nets[M]. Berlin,Germany:Springer,1997.

[14] 徐田华,赵红礼,唐 涛. 基于有色 Petri 网的 ETCS 无线通信可靠性分析[J]. 铁道学报,2008,30(1):38-42.

[15] 徐 亮,余建平. 改进的验证正确性 ACTL 性质的限界模型检测方法[J]. 计算机科学,2013,40(6A):99-102.

[16] 苏开乐,骆翔宇,吕关锋. 符号化模型检测 CTL*[J]. 计算机学报,2005,28(11):1798-1806.

[17] GIRAULT C. Petri nets for systems engineering[M]. Berlin,Germany:Springer,2010.

编辑 金胡考

(上接第 119 页)

[4] 宋 巍,窦万春. 基于资源和时间限制的多工作流网及可调度性验证[J]. 计算机应用与软件,2007,24(7):5-8.

[5] 宋 巍,窦万春,刘茜萍. 时间约束 Petri 网及其可调度性分析与验证[J]. 软件学报,2007,18(1):11-21.

[6] 庞 辉,方宗德,赵 勇. 时间约束工作流模型的简化分析与可调度性验证[J]. 计算机集成制造系统,2008,14(11):2217-2223.

[7] 顾春琴,衣 杨,常会友,等. 时间约束工作流模型的可调度性验证研究[J]. 系统仿真学报,2009,21(7):2112-2117.

[8] 冯复剑. 时间约束工作流的可调度性分析[J]. 计算机工程与应用,2016,52(12):26-30.

[9] 徐 超. 基于 PaRSEC 的并行计算工作流调度算法研究[D]. 厦门:厦门大学,2016.

[10] 徐 飞,张曦煌. 基于集群拓扑结构的工作流调度研究[J]. 计算机应用与软件,2016,33(1):291-294.

[11] 程 相,孙继银,李琳琳,等. 军事网格工作流调度算法研究[J]. 计算机工程,2012,38(10):45-47.

[12] 胡志刚,胡周君. 基于多 QoS 目标的工作流任务调度算法[J]. 计算机工程,2008,34(10):126-128.

[13] 郑继川,胡正国. 基于 Petri 网的工作流建模与分析[J]. 计算机工程,2003,29(5):26-27.

[14] 付丽娜,郝克刚. 工作流流程动态仿真技术的研究[J]. 计算机工程与应用,2009,45(13):29-33.

[15] 杨启亮,李决龙,邢建春,等. 需求自感知的军港岸基保障物联网系统[J]. 国防科技,2014,35(2):56-62.

编辑 顾逸斐