

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Profiles –
Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2**

**Réseaux de communication industriels – Profils –
Partie 3-2: Bus de terrain de sécurité fonctionnelle – Spécifications
supplémentaires pour CPF 2**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2007 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Profiles –
Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2**

**Réseaux de communication industriels – Profils –
Partie 3-2: Bus de terrain de sécurité fonctionnelle – Spécifications
supplémentaires pour CPF 2**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XH

ICS 25.040; 35.100.05

ISBN 978-2-8322-0888-5

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	11
INTRODUCTION.....	13
1 Scope.....	16
2 Normative references	16
3 Terms, definitions, symbols, abbreviated terms and conventions	17
3.1 Terms and definitions	17
3.1.1 Common terms and definitions	17
3.1.2 CPF 2: Additional terms and definitions	21
3.2 Symbols and abbreviated terms.....	22
3.2.1 Common symbols and abbreviated terms	22
3.2.2 CPF 2: Additional symbols and abbreviated terms	22
3.3 Conventions	23
4 Overview of FSCP 2/1 (CIP Safety™).....	23
4.1 General.....	23
4.2 FSCP 2/1	23
5 General	24
5.1 External documents providing specifications for the profile.....	24
5.2 Safety functional requirements	25
5.3 Safety measures	25
5.4 Safety communication layer structure	26
5.5 Relationships with FAL (and DLL, PhL)	26
5.5.1 General	26
5.5.2 Data types.....	26
6 Safety communication layer services	27
6.1 Introduction.....	27
6.2 Connection object	27
6.2.1 General	27
6.2.2 Class attribute extensions	27
6.2.3 Service extensions	28
6.2.4 Explicit message response format for SafetyOpen and SafetyClose	28
6.3 Connection Manager object.....	29
6.3.1 General	29
6.3.2 ForwardOpen for safety	29
6.3.3 Safety network segment	31
6.3.4 Originator rules for calculating the connection parameter CRC	33
6.3.5 SafetyOpen processing flowcharts	33
6.3.6 Checks required by Multipoint producers with existing connections	36
6.3.7 Electronic key usage for safety	36
6.3.8 RPI vs. API in safety connections	36
6.3.9 Application path construction for safety	36
6.3.10 Safety Validator connection types.....	38
6.3.11 Application reply data in a successful SafetyOpen response.....	39
6.3.12 Unsuccessful SafetyOpen response	40
6.3.13 ForwardClose for safety.....	43
6.4 Identity object.....	43
6.4.1 General	43

6.4.2	Changes to common services	43
6.5	Link objects	44
6.5.1	DeviceNet object changes	44
6.5.2	TCP/IP Interface object changes	44
6.6	Safety Supervisor object.....	45
6.6.1	General	45
6.6.2	Safety Supervisor class attributes.....	45
6.6.3	Subclasses	46
6.6.4	Safety Supervisor instance attributes.....	46
6.6.5	Semantics	49
6.6.6	Subclasses	55
6.6.7	Safety Supervisor common services	55
6.6.8	Safety Supervisor behavior.....	66
6.7	Safety Validator object	73
6.7.1	General	73
6.7.2	Class attributes	73
6.7.3	Instance attributes	74
6.7.4	Class services	80
6.7.5	Instance services.....	80
6.7.6	Object behavior	81
6.8	Connection Configuration Object.....	84
6.8.1	General	84
6.8.2	Class attribute extensions	84
6.8.3	Instance attributes, additions and extensions.	84
6.8.4	Instance attribute semantics extensions or restrictions for safety.....	86
6.8.5	Special Safety Related Parameters – (Attribute 13)	91
6.8.6	Object-specific services.....	95
6.8.7	Common service extensions for safety.....	95
6.8.8	Object behavior	97
7	Safety communication layer protocol	98
7.1	Safety PDU format	98
7.1.1	Safety PDU encoding	98
7.1.2	Safety CRC	108
7.2	Communication protocol behavior.....	109
7.2.1	Sequence of safety checks	109
7.2.2	Connection termination.....	109
7.2.3	Cross checking error	109
7.3	Time stamp operation.....	110
7.4	Protocol sequence diagrams	111
7.4.1	General	111
7.4.2	Normal safety transmission.....	111
7.4.3	Lost, corrupted and delayed message transmission.....	112
7.4.4	Lost, corrupted or delayed message transmission with production repeated.....	115
7.4.5	Point-to-point ping	117
7.4.6	Multipoint ping on CP 2/3 Safety.....	118
7.4.7	Multipoint ping on CP 2/2 safety networks	119
7.4.8	Multipoint ping – retry with success	120
7.4.9	Multipoint ping – retry with timeout	121

7.5	Safety protocol definition	122
7.5.1	General	122
7.5.2	High level view of a safety device	122
7.5.3	Safety Validator object	123
7.5.4	Relationship between SafetyValidatorServer and SafetyValidatorClient	123
7.5.5	SafetyValidatorClient function definition	124
7.5.6	SafetyValidatorServer function definition	132
7.6	Safety message and protocol data specifications	142
7.6.1	Mode octet	142
7.6.2	Time Stamp Section	143
7.6.3	Time Coordination Message	143
7.6.4	Time correction message	144
7.6.5	Safety data production	144
7.6.6	Producer dynamic variables	151
7.6.7	Producer per consumer dynamic variables	153
7.6.8	Consumer data variables	155
7.6.9	Consumer input static variables	157
7.6.10	Consumer dynamic variables	157
8	Safety communication layer management	159
8.1	Overview	159
8.2	Definition of the measures used during connection establishment	160
8.3	Originator-Target relationship validation	163
8.4	Detection of mis-routed connection requests	164
8.5	SafetyOpen processing	164
8.6	Ownership management	165
8.7	Bridging different physical layers	166
8.8	Safety connection establishment	167
8.8.1	Overview	167
8.8.2	Basic facts for connection establishment	167
8.8.3	Configuring safety connections	168
8.8.4	Network time expectation multiplier	169
8.8.5	Establishing connections	171
8.8.6	Recommendations for consumer number allocation	173
8.8.7	Recommendations for connection establishment	174
8.8.8	Ownership establishment	174
8.8.9	Ownership use cases	175
8.8.10	PID/CID usage and establishment	178
8.8.11	Proper PID/CID usage in multipoint and point-to-point connections	178
8.8.12	Network supported services	180
8.8.13	FSCP 2/1 Safety device type	181
8.9	Safety configuration process	185
8.9.1	Introduction to safety configuration	185
8.9.2	Configuration goals	185
8.9.3	Configuration overview	186
8.9.4	User configuration guidelines	187
8.9.5	Configuration process SIL3 justification	188
8.9.6	Device functions for tool configuration	189
8.9.7	Password security	189

8.9.8	SNCT interface services	189
8.9.9	Configuration lock.....	189
8.9.10	Effect of configuration lock on device behavior	190
8.9.11	Configuration ownership	191
8.9.12	Configuration mode	191
8.9.13	Measures used to ensure integrity of configuration process	191
8.9.14	Download process	193
8.9.15	Verification process	196
8.9.16	Verification process	199
8.9.17	Configuration error analysis.....	200
8.10	Electronic Data Sheets extensions for safety	203
8.10.1	General rules for EDS based safety devices	203
8.10.2	EDS extensions for safety	204
9	System requirements.....	208
9.1	Indicators and switches	208
9.1.1	General indicator requirements.....	208
9.1.2	LED indications for setting the device UNID.....	208
9.1.3	Module Status LED.....	209
9.1.4	Indicator warning	209
9.1.5	Network Status LED.....	209
9.1.6	MACID determination.....	211
9.1.7	Reset switch.....	212
9.2	Installation guidelines.....	213
9.3	Safety function response time	213
9.3.1	Overview.....	213
9.3.2	Network time expectation.....	213
9.3.3	Equations for calculating network reaction times	214
9.4	Duration of demands	216
9.5	Constraints for calculation of system characteristics.....	216
9.5.1	Number of nodes	216
9.5.2	Network PFH	216
9.5.3	Bit Error Rate (BER)	218
9.6	Maintenance.....	219
9.7	Safety manual	219
10	Certification.....	219
Annex A (informative) Additional information for functional safety communication profiles of CPF 2.....		220
A.1 Hash function example code.....		220
Bibliography.....		232

Table 1 – Communications errors and detection measures matrix.....	25
Table 2 – New class attributes	27
Table 3 – Service extensions	28
Table 4 – SafetyOpen and SafetyClose response format	28
Table 5 – Safety network segment identifier.....	31
Table 6 – Safety network segment definition	31
Table 7 – Safety network segment router format	33

Table 8 – Multipoint producer parameter evaluation rules	36
Table 9 – ForwardOpen setting options for safety connections.....	38
Table 10 – Network connection parameters for safety connections	39
Table 11 – CP 2/3 Safety target application reply (size: 10 octets).....	40
Table 12 – SafetyOpen target application reply (size: 16 octets).....	40
Table 13 – New and extended error codes for safety	41
Table 14 – SafetyOpen error event guidance table.....	42
Table 15 – Identity object common service changes	43
Table 16 – New DeviceNet object instance attribute	44
Table 17 – New TCP/IP Interface object Instance Attribute	44
Table 18 – Safety Supervisor class attributes	45
Table 19 – Safety Supervisor instance attributes	46
Table 20 – Device status attribute state values	50
Table 21 – Exception status attribute format	50
Table 22 – Common exception detail attribute values	51
Table 23 – Exception detail format summary.....	52
Table 24 – Summary of device behavior for various CFUNID values	54
Table 25 – Safety Supervisor common services	56
Table 26 – Safety Supervisor object specific services	56
Table 27 – Configure_Request message structure	58
Table 28 – Validate_Configuration message structure.....	58
Table 29 – Validate_Configuration success message structure	58
Table 30 – Validate_Configuration error code	59
Table 31 – Validate_Configuration extended codes	59
Table 32 – Set_Password message structure.....	61
Table 33 – Reset_Password message structure	61
Table 34 – Configuration_Lock/Unlock message structure	62
Table 35 – Mode_Change message structure	62
Table 36 – Safety_Reset message structure	63
Table 37 – Safety Supervisor safety reset types	63
Table 38 – Attribute bit map parameter	63
Table 39 – Reset processing rules for rest types.....	64
Table 40 – Propose_TUNID service	64
Table 41 – Apply_TUNID service	65
Table 42 – Safety Supervisor events.....	67
Table 43 – State event matrix for Safety Supervisor.....	68
Table 44 – Configuration owner control vs. device state.....	71
Table 45 – State mapping of Safety Supervisor to Identity object	72
Table 46 – Safety Supervisor object event mapping	72
Table 47 – Identity object event mapping	73
Table 48 – Safety Validator class attributes	74
Table 49 – Safety Validator instance attributes	74
Table 50 – Safety Validator state assignments.....	77

Table 51 – Safety Validator type, bit field assignments	77
Table 52 – Multipoint producer SafetyOpen parameter evaluation rules	79
Table 53 – Safety Validator class services	80
Table 54 – Safety Validator instance services	80
Table 55 – Safety Validator Get_Attributes_All service data	81
Table 56 – Safety Validator state event matrix	83
Table 57 – State mapping between Safety Supervisor and Safety Validator objects	84
Table 58 – Connection configuration object class attribute extensions	84
Table 59 – Connection Configuration Object instance attribute additions/extensions	85
Table 60 – Connection flag bit definitions	87
Table 61 – O-to-T connection parameters	88
Table 62 – T-to-O connection parameters	89
Table 63 – Data map formats	90
Table 64 – Data map format 0	91
Table 65 – Data map format 1	91
Table 66 – Target device's SCCRC values	93
Table 67 – Target device's SCTS values	94
Table 68 – Time correction connection parameters for multipoint connection	94
Table 69 – Connection Configuration Object-specific services	95
Table 70 – Get_Attributes_All Response service data (added attributes)	96
Table 71 – Set_Attributes_All Request service data (added attributes)	96
Table 72 – State Mapping between Safety Supervisor and the CCO objects	97
Table 73 – Connection sections and PDU formats	99
Table 74 – Mode octet variables	100
Table 75 – Time Stamp variables	102
Table 76 – Time Coordination message variables	103
Table 77 – Time Correction Message variables	105
Table 78 – CRC polynomials used	108
Table 79 – Connection sections and message formats	109
Table 80 – Data reception - Link triggered	134
Table 81 – Time_Correction reception - Link triggered	135
Table 82 – Data reception - Application triggered	135
Table 83 – Time_Correction reception - Application triggered	135
Table 84 – Consuming application – Safety data monitoring	136
Table 85 – Producer connection status determination	145
Table 86 – Consuming safety connection status	155
Table 87 – Connection establishment errors and measures to detect errors	160
Table 88 – SNN Date/Time allocations	161
Table 89 – SNN legal range of time values	161
Table 90 – Safety connection parameters	169
Table 91 – SafetyOpen summary	171
Table 92 – Originator/Target service mapping	182
Table 93 – Unsupported originator/target service types	182

Table 94 – Configuration goals	186
Table 95 – Configuration owner control vs. device state.....	191
Table 96 – Errors and detection measures	200
Table 97 – Parameter class keywords.....	205
Table 98 – New Connection Manager section keywords for safety	205
Table 99 – Connection Manager field usage for safety	206
Table 100 – Connection parameter field settings for safety	207
Table 101 – LED indications for setting UNID	208
Table 102 – Module Status LED.....	209
Table 103 – Network status LED states	210
Table 104 – Connection reaction time type – producing/consuming applications.....	214
Figure 1 – Relationships of IEC 61784-3 with other standards (machinery).....	13
Figure 2 – Relationships of IEC 61784-3 with other standards (process).....	14
Figure 3 – Relationship of Safety Validators	24
Figure 4 – Communication layers.....	26
Figure 5 – ForwardOpen with safety network segment.....	30
Figure 6 – Safety network target format.....	32
Figure 7 – Target Processing SafetyOpen with no configuration data (Form 2 SafetyOpen)	34
Figure 8 – Target Processing for SafetyOpen with configuration data (Form 1 SafetyOpen)	35
Figure 9 – Applying device configuration.....	59
Figure 10 – Configure and Validate processing flowcharts	60
Figure 11 – UNID handling during “Waiting for TUNID”	66
Figure 12 – Safety Supervisor state diagram.....	67
Figure 13 – Configuration, testing and locked relationships.....	71
Figure 14 – Safety connection types	78
Figure 15 – Safety Validator state transition diagram	82
Figure 16 – Connection Configuration Object state diagram.....	97
Figure 17 – Connection Configuration Object data flow.....	98
Figure 18 – Format of the mode octet	99
Figure 19 – 1 or 2 octet data section.....	100
Figure 20 – 3 to 250 octet data section format	101
Figure 21 – Time Stamp section format.....	102
Figure 22 – Time Coordination message encoding.....	103
Figure 23 – Time Correction message encoding	104
Figure 24 – 1 or 2 octet point-to-point PDU encoding.....	106
Figure 25 – 1 or 2 Octet multipoint PDU encoding.....	106
Figure 26 – 1 or 2 Octet, multipoint, Format 2 safety connection format	107
Figure 27 – 3 to 250 Octet Point-to-point PDU encoding	107
Figure 28 – 3 to 248 Octet Multipoint PDU encoding	107
Figure 29 – 3 to 248 Octet, Multipoint, safety connection format	108

Figure 30 – Time stamp sequence	110
Figure 31 – Sequence diagram of a normal producer/consumer safety sequence.....	111
Figure 32 – Sequence diagram of a normal producer/consumer safety sequence (production repeated)	112
Figure 33 – Sequence diagram of a corrupted producer to consumer message	113
Figure 34 – Sequence diagram of a lost producer to consumer message	114
Figure 35 – Sequence diagram of a delayed message	115
Figure 36 – Sequence diagram of a corrupted producer to consumer message with production repeated	116
Figure 37 – Sequence diagram of a connection terminated due to delays	117
Figure 38 – Sequence diagram of a failure of safety CRC check	117
Figure 39 – Sequence diagram of a point-to-point ping - normal response	118
Figure 40 – Sequence diagram of a successful multipoint ping, CP 2/3 safety	119
Figure 41 – Sequence diagram of a successful multipoint ping, CP 2/2 safety	120
Figure 42 – Sequence diagram of a multipoint ping retry	121
Figure 43 – Sequence diagram of a multipoint ping timeout	121
Figure 44 – Safety device reference model entity relation diagram	122
Figure 45 – Two devices interchanging safety data via a SafetyValidatorClient and a SafetyValidatorServer	123
Figure 46 – Safety production data flow	124
Figure 47 – Consumer safety data monitoring	133
Figure 48 – SafetyValidatorServer - application triggered	133
Figure 49 – Target ownership	164
Figure 50 – SafetyOpen forms	165
Figure 51 – Connection ownership state chart	165
Figure 52 – SafetyOpen UNID mapping	166
Figure 53 – Common CPF 2 application layer	166
Figure 54 – End-to-End routing example	167
Figure 55 – Sources for safety related connection parameters	170
Figure 56 – Parameter mapping between originator and target	170
Figure 57 – CP 2/3 Safety connection establishment in targets for Form 2a SafetyOpen	172
Figure 58 – General sequence to detect configuration is required	173
Figure 59 – PID/CID exchanges for two originator scenarios	178
Figure 60 – Seed generation for multipoint connections	179
Figure 61 – PID/CID runtime handling	180
Figure 62 – Connection categories and supported services	183
Figure 63 – Recommended connection types	184
Figure 64 – Logic-to-logic supported services	184
Figure 65 – Recommended connection types for logic to logic	185
Figure 66 – Configuration data transfers	186
Figure 67 – Protection measures in safety devices	188
Figure 68 – Configuration, testing and locked relationships	190
Figure 69 – Originator's configuration data	192
Figure 70 – SNCT to device download process	194

Figure 71 – SNCT Downloads to originators that perform Form 1 configuration	195
Figure 72 – Protection from locking and ownership	197
Figure 73 – Example of read back and comparison of original and printout	198
Figure 74 – Diverse display without full data read back	199
Figure 75 – Verification process including all alternatives	199
Figure 76 – Safety device MACID processing logic	212
Figure 77 – Safety function response time	213
Figure 78 – Safety function response time components	215
Figure 79 – Network protocol reliability block diagram (RBD)	216
Figure 80 – Network PFH summary	218

Withdrawing
IECNORM.COM: Click to view the full PDF of IEC 61784-3-2:2007

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
PROFILES –****Part 3-2: Functional safety fieldbuses –
Additional specifications for CPF 2**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning the functional safety communication profiles for family 2 as follows, where the [xx] notation indicates the holder of the patent right:

US 6,631,476	[RA]	Safety network for industrial controller providing redundant connections on single media
US 6,701,198	[RA]	Safety network for industrial controller allowing initialization on standard networks
US 6,721,900	[RA]	Safety network for industrial controller having reduced bandwidth requirements
US 6,891,850	[RA]	Network independent safety protocol for industrial controller
US 6,915,444	[RA]	Network independent safety protocol for industrial controller using data manipulation techniques

IEC takes no position concerning the evidence, validity and scope of these patent rights.

The holders of these patents rights have assured the IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holders of these patent rights are registered with IEC.

Information may be obtained from:

[RA] Rockwell Automation, Inc.
1201 S. Second Street
Milwaukee, WI 53204
USA
Attention: Intellectual Property Dept.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61784-3-2 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial process measurement, control and automation.

This bilingual version (2013-07) corresponds to the monolingual English version, published in 2007-12.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/470/FDIS	65C/481/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The French version of this standard has not been voted upon.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

The list of all parts of the IEC 61784-3 series, under the general title *Industrial communication networks – Profiles – Functional safety fieldbuses*, can be found on the IEC website.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The IEC 61158 fieldbus standard together with its companion standards IEC 61784-1 and IEC 61784-2 defines a set of communication protocols that enable distributed control of automation applications. Fieldbus technology is now considered well accepted and well proven. Thus many fieldbus enhancements are emerging, addressing not yet standardized areas such as real time, safety-related and security-related applications.

This standard explains the relevant principles for functional safety communications with reference to IEC 61508 series and specifies several safety communication layers (profiles and corresponding protocols) based on the communication profiles and protocol layers of IEC 61784-1, IEC 61784-2 and the IEC 61158 series. It does not cover electrical safety and intrinsic safety aspects.

Figure 1 shows the relationships between this standard and relevant safety and fieldbus standards in a machinery environment.

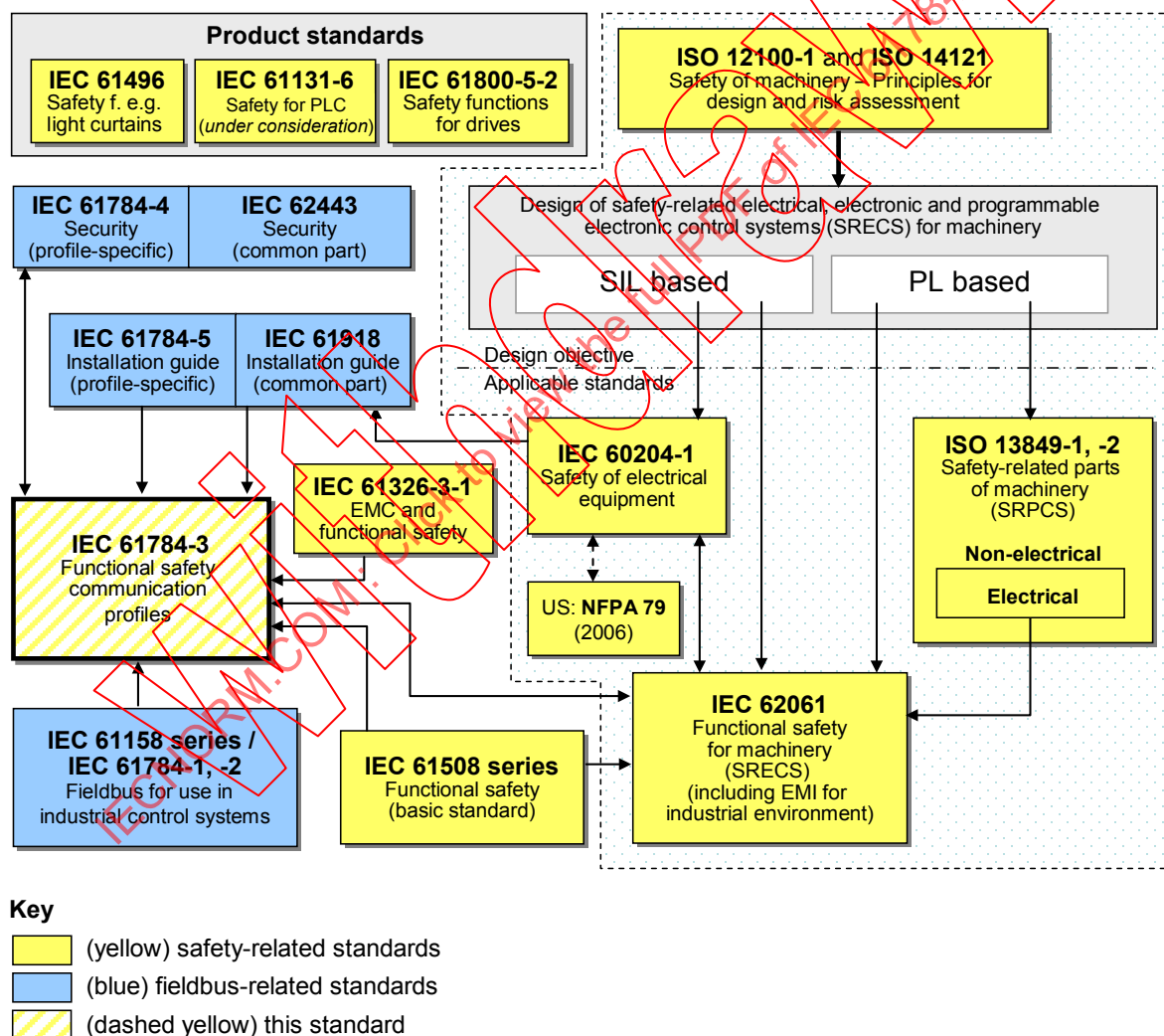
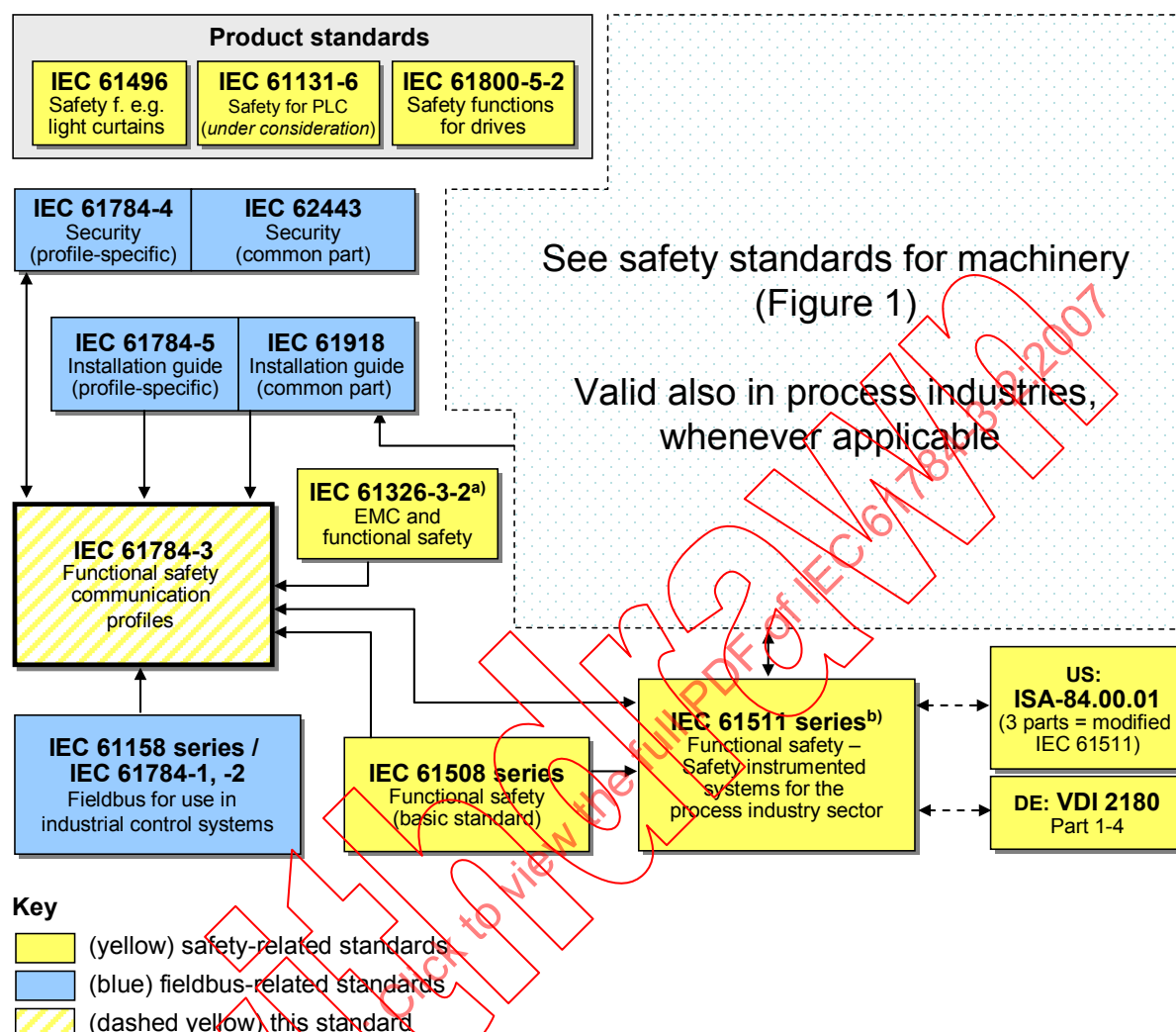


Figure 1 – Relationships of IEC 61784-3 with other standards (machinery)

Figure 2 shows the relationships between this standard and relevant safety and fieldbus standards in a process environment.



^a For specified electromagnetic environments; otherwise IEC 61326-3-1.

^b EN ratified.

Figure 2 – Relationships of IEC 61784-3 with other standards (process)

Safety communication layers which are implemented as parts of safety-related systems according to IEC 61508 series provide the necessary confidence in the transportation of messages (information) between two or more participants on a fieldbus in a safety-related system, or sufficient confidence of safe behaviour in the event of fieldbus errors or failures.

Safety communication layers specified in this standard do this in such a way that a fieldbus can be used for applications requiring functional safety up to the Safety Integrity Level (SIL) specified by its corresponding functional safety communication profile.

The resulting SIL claim of a system depends on the implementation of the selected functional safety communication profile within this system – implementation of a functional safety communication profile in a standard device is not sufficient to qualify it as a safety device.

This standard describes

- basic principles for implementing the requirements of IEC 61508 series for safety-related data communications, including possible transmission faults, remedial measures and considerations affecting data integrity;
- individual description of functional safety profiles for several communication profile families in IEC 61784-1 and IEC 61784-2;
- safety layer extensions to the communication service and protocols sections of the IEC 61158 series.

IECNORM.COM: Click to view the full PDF of IEC 61784-3-2:2007

Withdawn

INDUSTRIAL COMMUNICATION NETWORKS – PROFILES –

Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2

1 Scope

This part of the IEC 61784-3 series specifies a safety communication layer (services and protocol) based on CPF 2 of IEC 61784-1, IEC 61784-2 and IEC 61158 Type 2. It identifies the principles for functional safety communications defined in IEC 61784-3 that are relevant for this safety communication layer.

NOTE 1 It does not cover electrical safety and intrinsic safety aspects. Electrical safety relates to hazards such as electrical shock. Intrinsic safety relates to hazards associated with potentially explosive atmospheres.

This part¹ defines mechanisms for the transmission of safety-relevant messages among participants within a distributed network using fieldbus technology in accordance with the requirements of IEC 61508 series for functional safety. These mechanisms may be used in various industrial applications such as process control, manufacturing automation and machinery.

This part provides guidelines for both developers and assessors of compliant devices and systems.

NOTE 2 The resulting SIL claim of a system depends on the implementation of the selected functional safety communication profile within this system – implementation of a functional safety communication profile according to this part in a standard device is not sufficient to qualify it as a safety device.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-2, *Programmable controllers – Part 2: Equipment requirements and tests*

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC 61158-2, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-3-2, *Industrial communication networks – Fieldbus specifications – Part 3-2: Data-link layer service definition*

IEC 61158-4-2, *Industrial communication networks – Fieldbus specifications – Part 4-2: Data-link layer protocol specification*

IEC 61158-5-2, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition*

IEC 61158-6-2, *Industrial communication networks – Fieldbus specifications – Part 6-2: Application layer protocol specification*

¹ In the following pages of this standard, “this part” will be used for “this part of the IEC 61784-3 series”.

IEC 61326-3-1, *Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety related functions (functional safety) – General industrial applications*²

IEC 61326-3-2, *Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 3-2: Immunity requirements for safety-related systems and for equipment intended to perform safety related functions (functional safety) – Industrial applications with specified EM environment*²

IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

IEC 61784-3, *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions*

IEC 61784-5-2, *Industrial communication networks – Profiles – Part 5: Installation of fieldbuses – Installation profiles for CPF 2*

IEC 61918, *Industrial communication networks – Installation of communication networks in industrial premises*

IEC 62026-3, *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 3: DeviceNet*

ISO 15745-2, *Industrial automation systems and integration – Open systems application integration framework – Part 2: Reference description for ISO 11898-based control systems*

ISO 15745-3, *Industrial automation systems and integration – Open systems application integration framework – Part 3: Reference description for IEC 61158-based control systems*

ISO 15745-4, *Industrial automation systems and integration – Open systems application integration framework – Part 4: Reference description for Ethernet-based control systems*

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1 Common terms and definitions

3.1.1.1 availability

probability for an automated system that for a given period of time there are no unsatisfactory system conditions such as loss of production

NOTE Availability depends on MTBF (mean time between failure) and MDT (mean down time):
Availability = MTBF / (MTBF + MDT).

² To be published.

3.1.1.2

black channel

communication channel without available evidence of design or validation according to IEC 61508 series

3.1.1.3

bridge

abstract device that connects multiple network segments along the data link layer

3.1.1.4

communication channel

logical connection between two end-points within a *communication system*

3.1.1.5

communication system

arrangement of hardware, software and propagation media to allow the transfer of *messages* (ISO/IEC 7498 application layer) from one application to another

3.1.1.6

connection

logical binding between two application objects within the same or different devices

3.1.1.7

Cyclic Redundancy Check (CRC)

<value> redundant data derived from, and stored or transmitted together with, a block of data in order to detect data corruption

<method> procedure used to calculate the redundant data

NOTE 1 Terms "CRC code" and "CRC signature", and labels such as CRC1, CRC2, may also be used in this standard to refer to the redundant data.

NOTE 2 See also [28], [29]³

3.1.1.8

error

discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition

NOTE 1 An error can be caused by a faulty item, e.g. a computing error made by faulty computer equipment.

[IEV 191-05-24], [IEC 61508-4:1998], [IEC 61158]

NOTE 2 Errors may be due to design mistakes within hardware/software and/or corrupted information due to electromagnetic interference and/or other effects.

NOTE 3 Errors do not necessarily result in a *failure* or a *fault*.

3.1.1.9

failure

termination of the ability of a functional unit to perform a required function

NOTE 1 The definition in IEC 191-04-01 is the same, with additional notes.

[IEC 61508-4:1998], [ISO/IEC 2382-14.01.11]

NOTE 2 Failure may be due to an *error* (for example, problem with hardware/software design or message disruption)

³ Figures in square brackets refer to the bibliography.

3.1.1.10**fault**

abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function

NOTE IEC 191-05-01 defines “fault” as a state characterized by the inability to perform a required function, excluding the inability during preventative maintenance or other planned actions, or due to lack of external resources.

[IEC 61508-4:1998], [ISO/IEC 2382-14.01.10]

3.1.1.11**fieldbus**

communication system based on serial data transfer and used in industrial automation or process control applications

3.1.1.12**frame**

denigrated synonym for DLPDU

3.1.1.13**hash function**

(mathematical) function that maps values from a (possibly very) large set of values into a (usually) smaller range of values

NOTE 1 Hash functions can be used to detect data corruption.

NOTE 2 Common hash functions include parity, checksum or CRC.

[IEC 62210, modified]

3.1.1.14**hazard**

state or set of conditions of a system that, together with other related conditions will inevitably lead to harm to persons, property or environment

3.1.1.15**master**

active communication entity able to initiate and schedule communication activities by other stations which may be masters or slaves

3.1.1.16**message**

ordered series of octets intended to convey information

[ISO/IEC 2382-16.02.01, modified]

3.1.1.17**nuisance trip**

spurious trip with no harmful effect

NOTE Internal abnormal errors can be caused in communication systems such as wireless transmission, for example by too many retries in the presence of interferences.

3.1.1.18**redundancy**

existence of means, in addition to the means which would be sufficient for a functional unit to perform a required function or for data to represent information

EXAMPLE Duplicated functional components and the addition of parity bits are both instances of redundancy.

NOTE 1 Redundancy is used primarily to improve reliability or availability.

NOTE 2 The definition in IEC 191-15-01 is less complete.

[IEC 61508-4:1998], [ISO/IEC 2382-14.01.12]

3.1.1.19

reliability

probability that an automated system can perform a required function under given conditions for a given time interval (t_1, t_2)

NOTE 1 It is generally assumed that the automated system is in a state to perform this required function at the beginning of the time interval.

NOTE 2 The term "reliability" is also used to denote the reliability performance quantified by this probability.

NOTE 3 Within the MTBF or MTTF period of time, the probability that an automated system will perform a required function under given conditions is decreasing.

NOTE 4 Reliability differs from availability.

[IEC 62059-11, modified]

3.1.1.20

risk

combination of the probability of occurrence of harm and the severity of that harm
[IEC 61508-4:1998]

3.1.1.21

safety communication layer (SCL)

communication layer that includes all the necessary measures to ensure safe transmission of data in accordance with the requirements of IEC 61508

3.1.1.22

safety connection

connection that utilizes the safety protocol for communications transactions

3.1.1.23

safety data

data transmitted across a safety network using a safety protocol

NOTE The Safety Communication Layer does not ensure safety of the data itself, only that the data is transmitted safely.

3.1.1.24

safety device

device designed in accordance with IEC 61508 and which implements the functional safety communication profile

3.1.1.25

safety function

function to be implemented by an E/E/PE safety-related system, other technology *safety-related system* or external risk reduction facilities, which is intended to achieve or maintain a safe state for the EUC, in respect of a specific hazardous event
[IEC 61508-4:1998]

3.1.1.26

safety function response time

worst case elapsed time following an actuation of a safety sensor connected to a fieldbus, before the corresponding safe state of its safety actuator(s) is achieved in the presence of errors or failures in the safety function channel

NOTE This concept is introduced in IEC 61784-3, 5.2.4 and addressed by the functional safety communication profiles defined in this part.

3.1.1.27**safety integrity level (SIL)**

discrete level (one out of a possible four) for specifying the safety integrity requirements of the *safety functions* to be allocated to the E/E/PE safety-related systems, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest

NOTE The target failure measures for the four safety integrity levels are specified in Tables 2 and 3 of IEC 61508-1.

[IEC 61508-4:1998]

3.1.1.28**safety measure**

<this standard> measure to control possible communication *errors* that is designed and implemented in compliance with the requirements of IEC 61508

NOTE 1 In practice, several safety measures are combined to achieve the required safety integrity level.

NOTE 2 Communication *errors* and related safety measures are detailed in IEC 61784-3, 5.3 and 5.4.

3.1.1.29**safety-related application**

programs designed in accordance with IEC 61508 to meet the SIL requirements of the application

3.1.1.30**safety-related system**

system performing *safety functions* according to IEC 61508

3.1.1.31**slave**

passive communication entity able to receive messages and send them in response to another communication entity which may be a master or a slave

3.1.1.32**spurious trip**

trip caused by the safety system without a process demand

3.1.1.33**time stamp**

time information included in a *message*

3.1.2 CPF 2: Additional terms and definitions**3.1.2.1****ping**

request for the consumer to acknowledge the reception of a message

3.1.2.2**ping response**

response to a ping

3.1.2.3**signature**

method of uniquely identifying a set of data

NOTE This could be by revision, checksum or some other method that is agreed upon by the originating and target devices.

3.2 Symbols and abbreviated terms

3.2.1 Common symbols and abbreviated terms

CP	Communication Profile	[IEC 61784-1]
CPF	Communication Profile Family	[IEC 61784-1]
CRC	Cyclic Redundancy Check	
DLL	Data Link Layer	[ISO/IEC 7498-1]
DLPDU	Data Link Protocol Data Unit	
EMI	Electro-Magnetic Interference	
EUC	Equipment Under Control	[IEC 61508-4:1998]
FAL	Fieldbus Application Layer	[IEC 61158-5]
FSCP	Functional Safety Communication Profile	
E/E/PE	Electrical/Electronic/Programmable Electronic	[IEC 61508-4:1998]
PDU	Protocol Data Unit	[ISO/IEC 7498-1]
PFD	Average Probability of Failure on Demand	[IEC 61508-6:1998]
PFH	Probability of Failure per Hour	[IEC 61508-6:1998]
PhL	Physical Layer	[ISO/IEC 7498-1]
PLC	Programmable Logic Controller	
SCL	Safety Communication Layer	
SIL	Safety Integrity Level	[IEC 61508-4:1998]

3.2.2 CPF 2: Additional symbols and abbreviated terms

AIRS	Already In Requested Mode/State
BER	Bit Error Rate
CCO	Connection Configuration Object
CFUNID	Configuration UNID
CNP	Connection Point
CID	Consumer ID
CIP™	Common Industrial Protocol (application framework shared among CPF 2 communication profiles)
CPCRC	Connection Parameter CRC
CTM	Connection Timeout Multiplier
EDS	Electronic Data Sheet
EPI	Expected Packet Interval
LED	Light Emitting Diode
MACID	Device physical address on network
MOD	Module
NV	Non-volatile Storage
OCPUNID	Output Connection Owner UNID
OSC	Object State Conflict
O-to-T	Originator to Target
OUNID	Originator UNID
PID	Producer Identifier
PS	Power Supply
Res.	Reserved
RPI	Requested Packet Interval
SCCRC	Safety Configuration Data CRC

SCID	Safety Configuration Identified
SCTS	Safety Configuration Time Stamp
SNCT	Safety Network Configuration Tool
SNN	Safety Network Number
T-to-O	Target to Originator
TBD	To Be Defined
TNS	TUNID Not Set
TUNID	Target UNID
UNID	Unique Network Identifier
UML	Universal Modeling Language

3.3 Conventions

The conventions used for the descriptions of objects services and protocols are described in IEC 61158-3-2, IEC 61158-4-2, IEC 61158-5-2, and IEC 61158-6-2.

As appropriate this part uses flow charts, UML Sequence Diagrams and “C” pseudo-code to describe concepts.

4 Overview of FSCP 2/1 (CIP Safety™)

4.1 General

Communication Profile Family 2 (commonly known as CIP™⁴) defines communication profiles based on IEC 61158-2 Type 2, IEC 61158-3-2, IEC 61158-4-2, IEC 61158-5-2, and IEC 61158-6-2.

The basic profiles CP 2/1, CP 2/2, and CP 2/3 are defined in IEC 61784-1 and IEC 61784-2. The CPF 2 functional safety communication profile FSCP 2/1 (CIP Safety™⁴) is based on the CPF 2 basic profiles in IEC 61784-1 and IEC 61784-2 and the safety communication layer specifications defined in this part.

4.2 FSCP 2/1

FSCP 2/1 is based on the producer/consumer model of CPF 2. The pairing of producers and consumers is an important part of the relationship that provides the high integrity needed for safety-related applications.

The FSCP 2/1 safety communication layer is specified using a Safety Validator object. This object is responsible for managing the FSCP 2/1 safety connections and serves as the interface between the safety-related application objects and the link layer connections, as shown in Figure 3. The Safety Validator ensures the integrity of the safety data transfers.

⁴ CIP™ (Common Industrial Protocol) and CIP Safety™ are trade names of the non-profit organization Open DeviceNet Vendor Association, Inc (ODVA). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this part does not require use of the trade names CIP™ or CIP Safety™. Use of the trade names CIP™ or CIP Safety™ requires permission of ODVA.

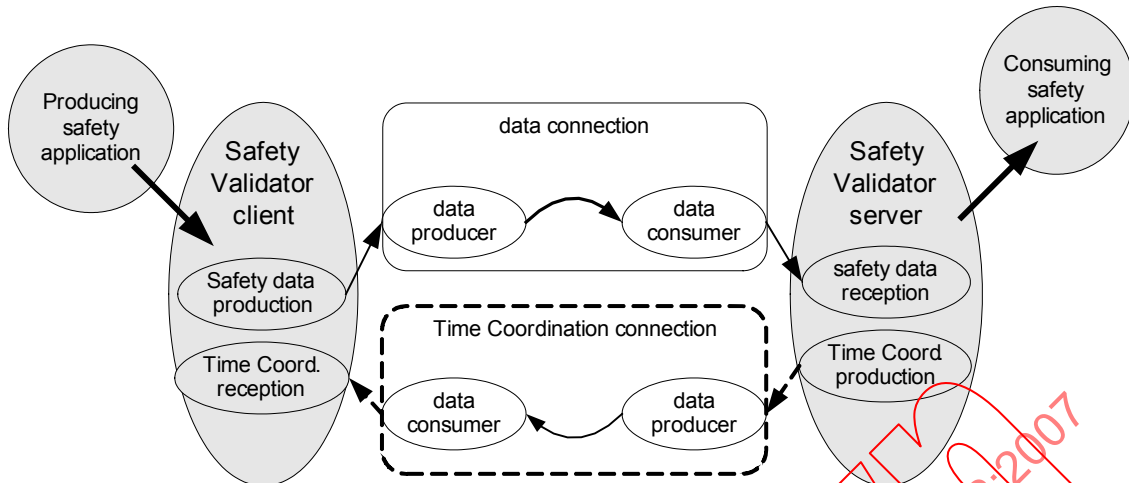


Figure 3 – Relationship of Safety Validators

The integrity of the safety data transfers is ensured as follows:

- the producing safety-related application uses an instance of a client Safety Validator to produce safety data and ensure time coordination;
- the client uses a link data producer to transmit the data and a link consumer to receive time coordination messages;
- the consuming safety-related application uses a server Safety Validator to receive and check data;
- the server uses a link consumer to receive data and a link producer to transmit time coordination messages.

FSCP 2/1 utilizes the black channel concept. The link producers and consumers have no knowledge of the safety packet and implement no safety function. The responsibility for high-integrity transfer and checking of safety data lies within the Safety Validators.

FSCP 2/1 uses the following measures to ensure the integrity of safety messaging:

- time stamp;
- connection authentication;
- data integrity assurance;
- redundancy with cross checking;
- different data integrity assurance systems.

Messages are produced with a timestamp that allows the consumer to verify the age of data being sent. Identification is encoded into each safety-related message to ensure that the correct consumer is using the message. All safety-related messages use a unique CRC. Safety-related data is sent redundantly. Diverse measures for producing safety-related messages are used to ensure that standard CPF 2 messages are not interpreted as safety messages.

5 General

5.1 External documents providing specifications for the profile

The following documents are especially useful in understanding the design of FSCP 2/1 protocols:

- GS-ET-26 [27]

5.4 Safety communication layer structure

The safety protocol is layered on top of the standard network protocol (the standard protocol stack is a black channel). Figure 4 shows how the safety communication layer is related to the existing CPF 2 layers.

The safety layers accept safety data from the safety-related application circuit. The safety layers then formulate a safety message, transmit the safety message, receive and decode the safety message. Once the safety message is decoded and verified, the data is presented to the receiving safety-related application circuit.

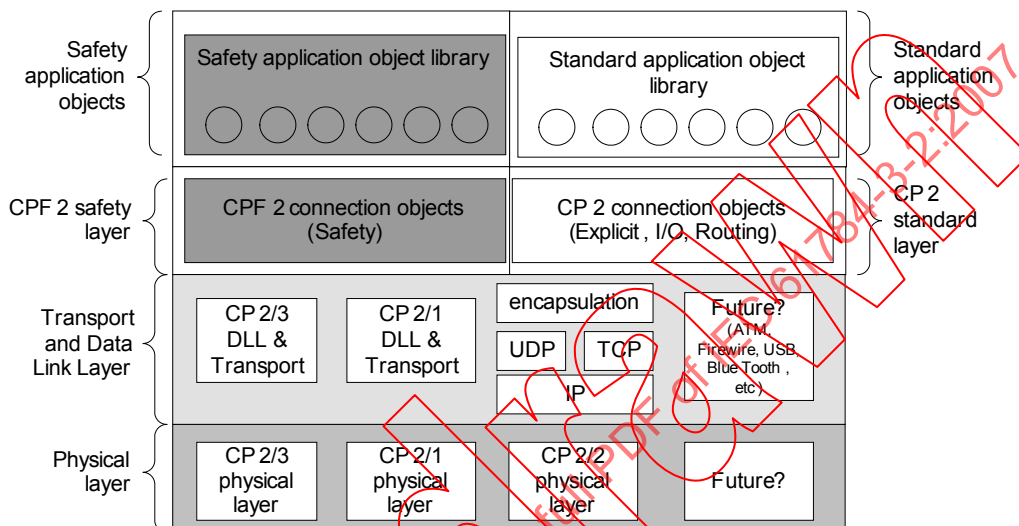


Figure 4 – Communication layers

5.5 Relationships with FAL (and DLL, PhL)

5.5.1 General

This safety communication layer shall only be used in conjunction with CPF 2 communication profiles (CP 2/3 (DeviceNet™⁵), CP 2/2 (EtherNet/IP™⁶)). There are no requirements of CPF 2 communications profiles other than those defined in this part.

5.5.2 Data types

Profiles defined in this part support all the CPF 2 data types as defined in IEC 61158-5 for Type 2.

⁵ DeviceNet™ is a trade name of the non-profit organization Open DeviceNet Vendor Association, Inc (ODVA). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this part does not require use of the trade name DeviceNet™. Use of the trade name DeviceNet™ requires permission of ODVA.

⁶ EtherNet/IP™ is a trade name of ControlNet International, Ltd. (CI) and Open DeviceNet Vendor Association, Inc (ODVA). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this part does not require use of the trade name EtherNet/IP™. Use of the trade name EtherNet/IP™ requires permission of either CI or ODVA.

6 Safety communication layer services

6.1 Introduction

This subclause defines the extensions to existing CPF 2 objects and new objects for Safety. Safety connections use Class 0 connections for I/O messages. Standard Explicit message connections are used for the establishment of Safety I/O connections.

6.2 Connection object

6.2.1 General

Several safety specific class attributes and services are defined to support safety. Safety nodes which reside on FSCP 2/1 shall implement the SafetyOpen and SafetyClose services of the Connection object.

6.2.2 Class attribute extensions

Table 2 defines the class attributes used by safety devices.

Table 2 – New class attributes

Attrib ID	Need in implementation	Access rule	N V	Name	Date type	Description of attribute
8	Optional	Get	V	Connection request error count	UINT	Diagnostic Counter that is a running count of SafetyOpen or SafetyClose faults ^a
9	Optional	Get	V	Safety connection counters	STRUCT of	Status counters
				SafetyOpen request success count	UINT	Count of successful SafetyOpen requests issued from this node to another target
				SafetyOpen request failure count	UINT	Count of unsuccessful SafetyOpen requests issued from this node to another target
				SafetyClose request success count	UINT	Count of successful SafetyClose requests issued from this node to another target
				SafetyClose request failure count ^b	UINT	Count of unsuccessful SafetyClose requests issued from this node to another target
				SafetyOpen indication success count	UINT	Count of successful SafetyOpen requests issued from an external originator to this node
				SafetyOpen indication failure count ^b	UINT	Count of unsuccessful SafetyOpen requests issued from an external originator to this node
				SafetyClose indication success count	UINT	Count of successful SafetyClose requests issued from an external originator to this node
				SafetyClose indication failure count ^b	UINT	Count of unsuccessful SafetyClose requests issued from an external originator to this node

^a Attribute 8 is the sum of the 4 failure counters in attribute 9. Having this single attribute represent connection errors can be used in monitoring software to efficiently monitor errors without polling the additional attribute 9 structures. Once attribute 8 is non-zero then the monitoring software can examine attribute 9 for more details.

^b Included in class attribute 8.

6.2.3 Service extensions

Table 3 defines services for use by safety devices on FSCP 2/1. These services shall be supported by a target device on FSCP 2/1. The SafetyOpen and SafetyClose services, used in lieu of the ForwardOpen and ForwardClose services of the Connection Manager object, inherit the identical behavior from those Connection Manager services.

Table 3 – Service extensions

Service code	Need in implementation		Service name	Description of service
	Class	Instance		
4D _{hex}	Optional	N/A	Reset Counters	This service resets class attributes 8 and 9 to zero
4E _{hex}	Required	N/A	SafetyClose	Used to close a Safety I/O connection. See the ForwardClose service of the Connection Manager object 6.3.13
54 _{hex}	Required	N/A	SafetyOpen ^a	Used to establish a safety I/O connection. See the ForwardOpen service of the Connection Manager object 6.3.2
^a The SafetyOpen service data shall contain a Safety Network Segment in the Connection_Path field.				

6.2.4 Explicit message response format for SafetyOpen and SafetyClose

The SafetyOpen and SafetyClose service responses have a common response format. This format contains status information for both success and failure. If the service response is a success, response data specific to each service is returned. If the service response is an error, a successful FSCP 2/1 status indicated by the general error code FF_{hex} is returned, along with a variable amount of extended error information.

The response format for the SafetyOpen and SafetyClose object Specific services are defined in Table 4:

Table 4 – SafetyOpen and SafetyClose response format

Parameter name	Data type	Description of response parameter
Reserved	USINT	Reserved
General status	USINT	One of the general status codes listed in IEC 61158-6-2 (status codes)
Size of additional status	USINT	Number of 16 bit words in Additional status array
Additional status	ARRAY of UINT	Additional status
Response data	ARRAY of OCTET	Response data from request or additional error data if general status indicated an error

The Response Data and General/Additional Statuses for the SafetyOpen and SafetyClose services shall match the Response Data and General/Additional Statuses for the ForwardOpen and ForwardClose services of the Connection Manager object, respectively.

6.3 Connection Manager object

6.3.1 General

Safety devices shall use a ForwardOpen with a Safety Network Segment to establish safety connections. This segment uniquely identifies the ForwardOpen service as a safety ForwardOpen. Throughout this subclause, the term “SafetyOpen” shall refer to both the ForwardOpen service of the Connection Manager object (when containing a Safety Network Segment) and the SafetyOpen service of the Connection object.

6.3.2 ForwardOpen for safety

The contents of the ForwardOpen service of the Connection Manager object, including the safety network segment, are outlined in Figure 5. The proper settings for the various parameters in the ForwardOpen are given below.

Parameter Name		Data Type / Size	Target Destination and Semantics
Priority/Time_tick		OCTET	Connection manager servicing request
Time-out_ticks		USINT	Connection manager servicing request
O-to-T Network Connection ID		UDINT	Consuming connection instance:: CIP consumed connection ID
T-to-O Network Connection ID		UDINT	Producing connection instance:: CIP producing connection ID
Connection Serial Number		UINT	Value: Unique (to originator) 16-bit number Destination: Connection manager servicing request
Originator Vendor ID		UINT	Connection manager servicing request
Originator Serial Number		UDINT	Connection manager servicing request
Connection Timeout Multiplier		USINT	Producing and consuming connection instance: Timeout Multiplier
Reserved		3 OCTETS	
O-to-T RPI		UDINT	Consuming connection instance: EPR
O-to-T Network Connection Parameters	Redundant Owner	WORD	Value: 0 (Safety Connections do not support redundancy); Destination: Connection Manager
	Connection Type		Value: Point-to-point; Destination: Connection Manager
	Priority		Value: High Priority; Destination: Connection Manager
	Fixed/Variable		Value: Fixed; Destination: Connection Manager
	Connection Size (in octets)		Value: Size of safety PDU including the Mode Octet, Actual, Complement, and Time stamp sections Destination: Consuming connection instance: Consumed connection size attribute
T-to-O RPI		UDINT	Producing connection instance: EPR
T to O Network Connection Parameters	Redundant Owner	WORD	Value: 0 (Safety connections do not support redundancy); Destination: Connection Manager
	Connection Type		Value: Point-to-point; Destination: Connection Manager
	Priority		Value: High Priority; Destination: Connection Manager
	Fixed/Variable		Value: Fixed; Destination: Connection Manager

Parameter Name		Data Type / Size	Target Destination and Semantics
	Connection Size (in octets)		Value: Size of safety PDU including the Mode Octet, Actual, Complement, and Time stamp sections Destination: Producing Connection Instance: Produced connection size
Transport Type/Trigger		SWORD	Value: Safety single or multipoint Destination: Validator instance Trigger always set to Application. Client/Server: depending on direction
Connection_Path_Size		USINT	Number of 16 bit words in the Connection_Path field – destination is Connection Manager servicing request. NOTE Value included in CRC calculation does not include the words used for Network/Port segments.
Connection_Path (Padded EPATH)	Path		Variable Consumed by intermediate bridge connection manager(s). Each Router Format-safety Network Segment immediately follows the (CPF 2 - to- CPF 2) bridge for which it is intended
	Logical Segment: Electronic Key	Vendor ID	UINT
		Prod Type	UINT
		Prod Code	UINT
		Compatible/ Major Rev	SWORD
		Minor Rev	USINT
	Application Path 1	Logical Segment: Class	Variable; 1 word typically
		Logical Segment: Instance	Variable; 1 word typically
	Application Path 2	Logical Segment: Connection Point	Variable; 1 word typically
	Application Path 3	Logical Segment: Connection Point	Variable; 1 word typically
	Config Data	Data Segment: Simple Data	Variable
	Network Segment: Safety	Safety Segment Name	50 _{hex}
		Segment Size	USINT
		Segment Format	USINT
		Network Safety Data	ARRAY of OCTET

Key

- = Connection/Safety Validator fields covered by Connection Parameters CRC
- = Configuration Data Segment covered by SCCV

Figure 5 – ForwardOpen with safety network segment

6.3.3 Safety network segment

6.3.3.1 Common format

The Safety Connection Parameters Segment Type has been added to the CPF 2 Network Segment definition as segment type 10_{hex} (resulting in a segment value of 50_{hex}). The updated Network Segment definition is defined in Table 5 below.

Table 5 – Safety network segment identifier

Network segment type	Network segment name
50 _{hex}	Safety segment

The format of the Safety Network Segment is defined in Table 6. This segment type is in the range of Network Segment types, which contains a data size field. The Safety Segment Format parameter, the first parameter in the data area, indicates which data follows.

Table 6 – Safety network segment definition

Octet offset	Field name	Data type	description
0	Segment Type	SWORD	Indicates Safety Network Segment (50 _{hex})
1	Network Segment data length	SWORD	Length of segment data to follow (in words)
2	Safety Network Segment format	USINT	Indicates type of Safety Network Segment 0 = Target format 1 = Router format 2 – 255 = Reserved
3 thru n	Safety Network Segment data	ARRAY of OCTET	Safety Network Segment data, defined by segment format.

6.3.3.2 Safety network segment: target format (00_{hex})

The target format is directed at the target of the connection. The value of the Network Segment Data Length parameter shall be 27 (1b_{hex}) which corresponds to 27 WORDS. Figure 6 defines the safety network target format.

Field name	Data type	Description
Reserved	SWORD	Pad
Configuration CRC (SCCRC)	UDINT	CRC-S4 of target Config Data Segment. A value of 0 is an indicator that SCID check can be skipped
Configuration Time Stamp (SCTS)	DATE_AND_TIME	Time and Date stamp of the configuration. A value of 0 is an indicator that SCID check can be skipped. 6-octet value (Date: 2-octets, Time: 4-octets)
Time Correction EPI	UDINT	Not used: 0 When used: Time Correction Connection Instance:: EPI
Time Correction Network Connection Parameters	WORD	Network Connection Parameters for the Time Correction connection. This field uses the same definition as the Network Connection Parameter within the Connection Manager object. When not used: 0 (Null) When used: Fixed, High Priority, Multipoint
Target_UNID (TUNID)	STRUCT 10 OCTETS	UNID of the target device. This value is a 10-octet value consisting of the Safety Network Id (6 octets) and the Node Address (4-octet address)
Originator UNID (OUNID)	STRUCT 10 OCTETS	UNID of the Originator device. This value is a 10-octet value consisting of the Safety Network Id (6 octets) and the Node Address (4-octet address)
Ping_Interval_EPI_Multiplier	UINT	Number that defines the Ping_Count_Interval for the connection. Valid range of values is given in 7.6.5.10
Time_Coord_Msg_Min_Multiplier	UINT	For Producing Connection Instance. Valid range is 0 to 7 813.
Network_Time_Expectation_Multiplier	UINT	For Consuming Connection Instance. Valid range is 0 to 42 969
Timeout_Multiplier	USINT	Used by producer and consumer to timeout connections. Valid range 1 to 4
Max_Consumer_Number	USINT	Value: 1 (Point-to-point), 2-15(Multipoint)
Connection Parameters CRC (CPCRC)	UDINT	CRC-S4 of target connection parameters in this request
Time Correction Connection ID	UDINT	Not Used: FFFFFFFF _{hex} When used: Time Correction Connection Instance:: Connection ID

Key

 = Safety Segment fields covered by Connection Parameters CRC

Figure 6 – Safety network target format

6.3.3.3 Safety network segment: router format (01_{hex})

The router format is directed at an intermediate router along the connection path when that router needs to know this is a Safety ForwardOpen. Currently, only CP 2/3 routers need this network segment directed to them. The Network Segment is applied to the subnet from which

the connection request originated. Router Format Safety Network Segments only appear in the Path portion of the Connection_Path of a Safety ForwardOpen. The total segment length is 6 words. The safety network segment router format is defined in Table 7.

Table 7 – Safety network segment router format

Field name	Data type	Description
Reserved	SWORD	Reserved, value = 0
Time Correction Connection ID	UDINT	Not Used: FFFFFFFF _{hex} When used: Time Correction connection instance:: Connection ID
Time Correction EPI	UDINT	Not used: 0 When used: Time Correction connection instance:: EPI
Time Correction Network Connection Parameters	WORD	Network Connection Parameters for the Time Correction connection. This field uses the same definition as the Network Connection Parameter within the Connection Manager object. Not used: 0 (Null) When used: Fixed, High Priority, Multipoint

6.3.4 Originator rules for calculating the connection parameter CRC

The Connection Parameter CRC (CPCRC) is only calculated when a connection is initially configured or when that connection configuration is changed. This value remains static in the originator otherwise.

The connection path size in the SafetyOpen changes value depending on whether an originator is sending configuration data or not. Since this field is included in the CPCRC calculation, originators need to calculate and maintain two values when they support the SafetyOpen configuration function. One value is used when configuring a device while opening the connection and the other value is used when just opening the connection (no configuration data is included).

Also, the path size shall be “pre-adjusted” to remove the path entries that get stripped off by routers as the Safety ForwardOpen traverses each sub-net. The requirement is that the originator is responsible for calculating the CPCRC value based on what the SafetyOpen or ForwardOpen will be when the target receives it so that the target can perform a simple calculation in all cases.

6.3.5 SafetyOpen processing flowcharts

The following flowcharts outline the required behavior of a target upon receipt of a SafetyOpen:

- Verify the Connection Parameter CRC. As depicted in Figure 6, the connection parameter CRC covers the SafetyOpen fields that will be used by the target. Routers along the path shall not modify these fields;
- The target checks that the electronic key segment matches (see 6.3.7);
- Depending on the type of safety connection, the logic is defined in Figure 7 or Figure 8 shall be followed.

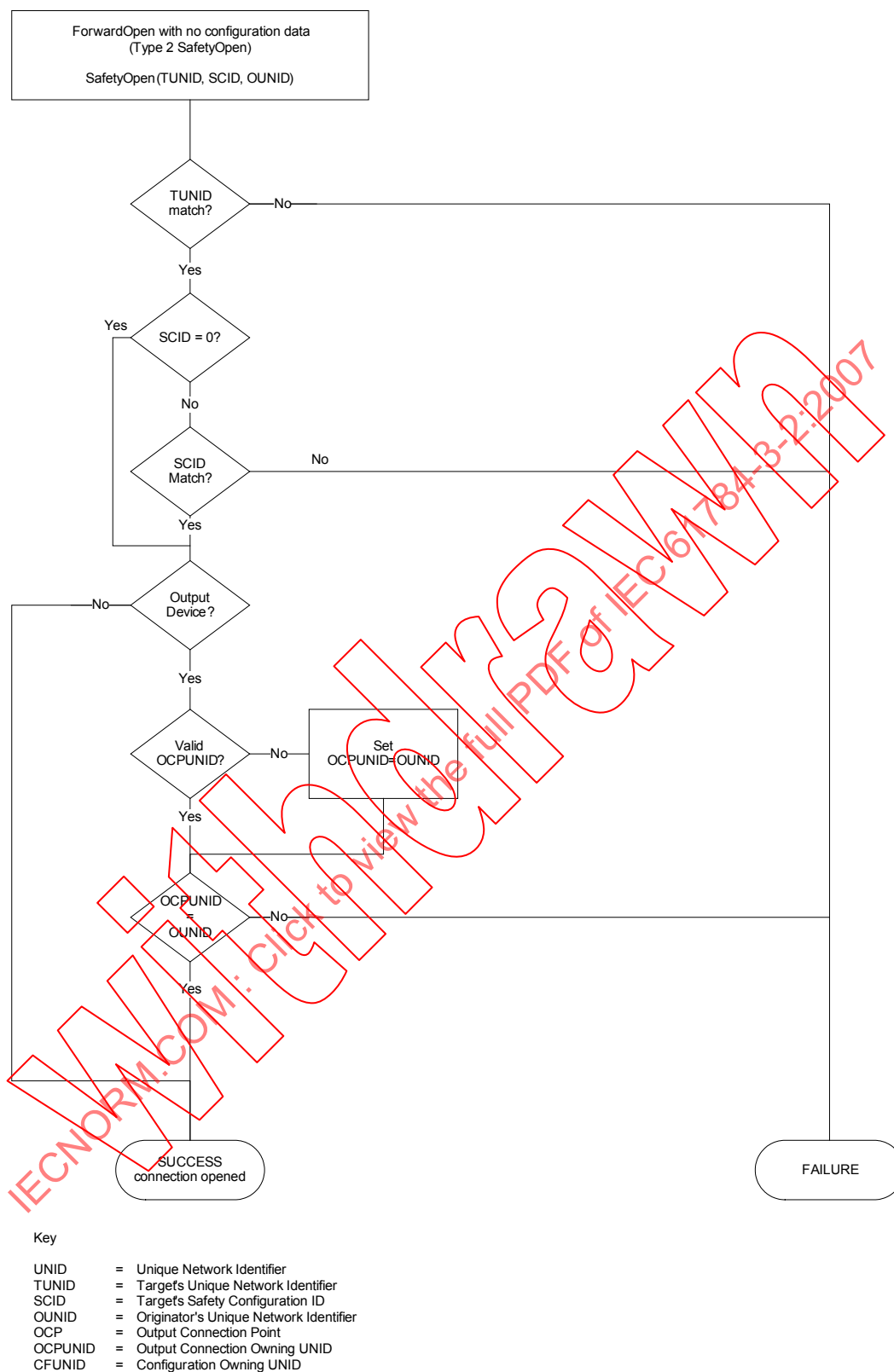


**Figure 7 – Target Processing SafetyOpen with no configuration data
(Form 2 SafetyOpen)**

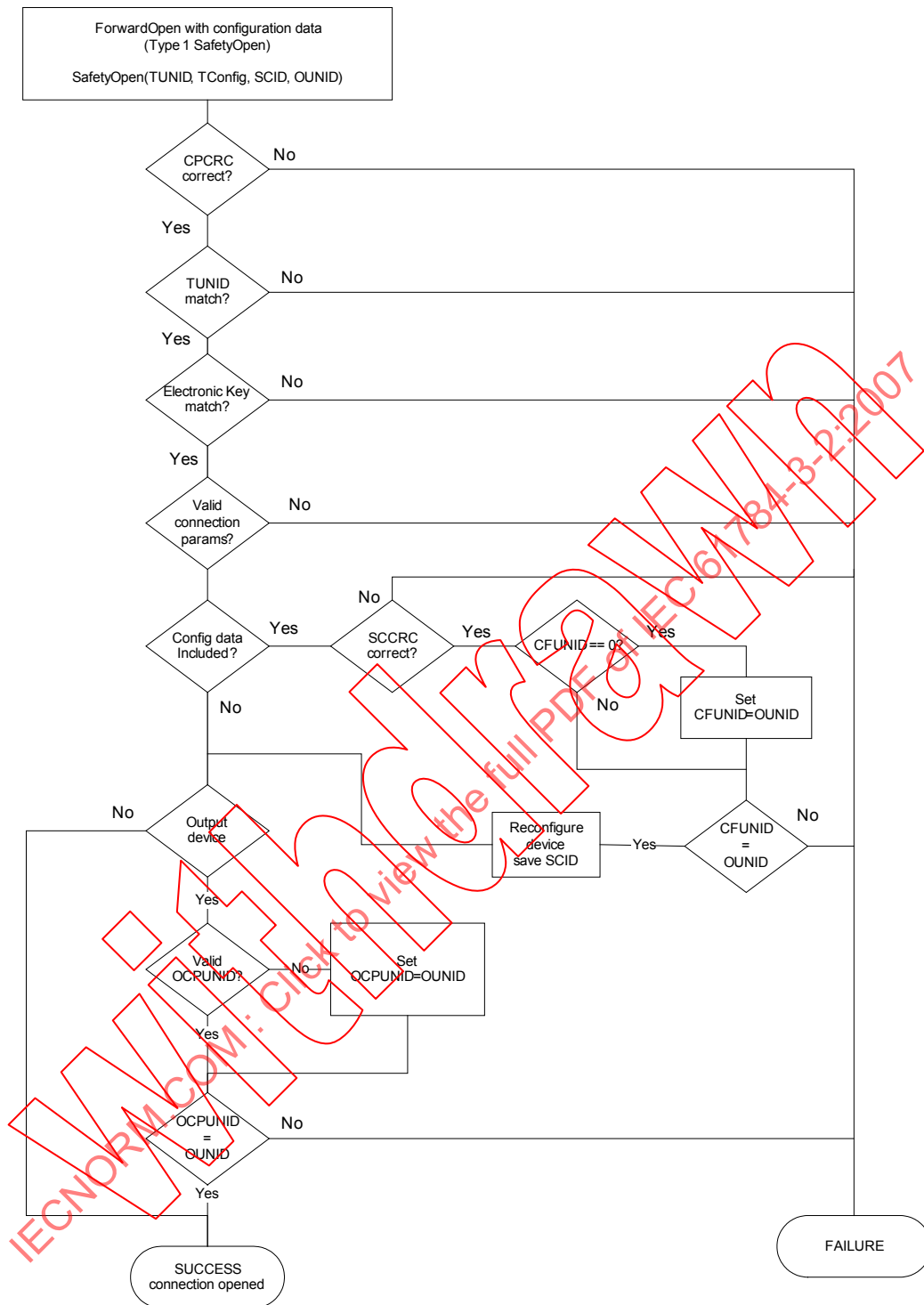


**Figure 8 – Target Processing for SafetyOpen with configuration data
(Form 1 SafetyOpen)**

6.3.6 Checks required by Multipoint producers with existing connections

There is a distinct possibility that the safety parameters used in a ForwardOpen or SafetyOpen from various consuming originators will not always be the same. A consistent set of rules shall be used for how a Multipoint producing target evaluates these parameters when they conflict with other already established consuming connections. Table 8 defines the multipoint producer parameter evaluation rules.

Table 8 – Multipoint producer parameter evaluation rules

Safety parameter	Producer evaluation rule if connection exists	Error response ^a (if appropriate)	
		General	Extended
Ping_Interval EPI Multiplier	Shall match existing P.I.E.M.	01 _{hex} 00 _{hex}	0801 _{hex} 0012 _{hex}
Time Coord Msg Min Multiplier	Accept any (all can be different)	N/A	
Network Time Expectation Multiplier	Accept any (all can be different)	N/A	
Timeout_Multiplier	Accept any (all can be different)	N/A	
Max_Consumer_Number	Shall match existing Max_Consumer_Number	00 _{hex} 01 _{hex}	0801 _{hex} 002A _{hex}
T-to-O RPI, O-to-T RPI, and Time Correction RPI	Shall match existing connection RPIs	01 _{hex}	0111 _{hex}
^a At the option of the device developer, another connection could also be opened rather than return an error response.			

6.3.7 Electronic key usage for safety

Safety originators and targets shall meet the following requirements for use of the Electronic Key segment within a SafetyOpen.

The Electronic Key segment shall be present in all SafetyOpen service requests.

The Electronic Key shall always be processed by Target devices with integrity (cannot be confirmed externally).

The compatibility bit and behavior in the Electronic Key shall be supported as defined by safety originators and targets. Wildcards (fields set to 0) for individual fields within the Electronic Key shall not be supported.

6.3.8 RPI vs. API in safety connections

Safety devices do not have the ability to negotiate the RPI value. Safety Targets either accept the requested RPI because it can be supported, or it rejects it completely. The tolerance around an acceptable RPI request shall be +/- 128 μ s.

The T-to-O API, O-to-T API, and Time Correction API in a success response shall always be the same value received in the SafetyOpen.

6.3.9 Application path construction for safety

IEC 61158-6-2 defines a method to construct an application path in a SafetyOpen for 3 references: one for configuration, one for consumed data and one for produced data.

The possible path types are identified as follows:

- A. - [Class] [Assembly] [Instance] [Configuration-instance]
- B. - [Class] [Assembly] [Instance] [Producer-instance]
- C. - [Class] [Assembly] [Instance] [Consumer-instance]
- D. - [Class] [Assembly] [Instance] [Device-defined NULL instance]
- E. - [CNP] [Consumer CNP]
- F. - [CNP] [Producer CNP]
- G. - [CNP] [Device-defined NULL CNP]

When all 3 paths referenced the same class, they are often formatted in a compressed format: (CP stands for Connection Point).

[Class] [Class #] [Instance] [Instance #] [CNP] [Consumer CP] [CNP] [Producer CNP]

Since all three paths shall reference the same class, a NULL instance value shall be reserved in that class (by the device developers) to use the compressed form for safety. The NULL value will be used for the portion of the path that references the Time Coordination connection and configuration connection (in non-configuring originators). If a common class reference is not practical, the long form shall be used.

The following rules apply to SafetyOpen path construction:

- all devices shall support the long form at a minimum;
- a Null path or Null connection point shall be used to represent the Time Coordination connection;
- a Null path or Null connection point shall be used for the configuration path when no configuration data can ever be sent on the connection;
- on any connections, which can accept a Form 1 SafetyOpen, Originators shall always send a valid configuration path even if no configuration data is present (i.e. Form 2a).

The "Null" path or connection point used shall be one of the module's choosing. The method of making this selection in an EDS file will be via the Data Path field in the Connection Manager section. Devices configured by configuration applet also shall provide this path.

Safety device designers can choose the path format to specify in their EDS files from the following options. The possible three-path combinations for Form 1, 2a SafetyOpen on connections, which will accept configuration are:

- [A, D, B] for the input connection;
- [A, G, F] compressed format for input connections;
- [A, C, D] for the output connection;
- [A, E, G] compressed format for output connections.

The possible three-path combinations for Form 2a, 2b SafetyOpen on connections, which will not accept configuration data are:

- [D, D, B] for the input connection;
- [D, G, F] compressed format for input connections;

- [D, C, D] for the output connection;
- [D, E, G] compressed format for output connections.

6.3.10 Safety Validator connection types

The Safety Validator function, depending on the safety connection type provided in the SafetyOpen, will always require between 2-3 connection resources. The SafetyOpen contains the connection information to allow a target to set up these connections appropriately. This subclause will show how the connection types are determined by how the SafetyOpen shall be filled out.

One basic assumption is that connections for Time Coordination and Time Correction messages do not require a path entry since they have an implied path to the Safety Validator object coordinating the safety connection. The various connection scenarios are described in the Validator object in 6.7.

The Safety Validator function can determine which connection resources will be required by looking at the parameter in the SafetyOpen. The various SafetyOpen parameters as well as the key parameters that the safety function will need to determine the types of connections are defined in Table 9. Connection object instances may not be available due to target identifier capacity or limitations; there may be no unallocated connection identifiers. Or the device may not have the processing throughput to handle the additional connection; etc. Safety Connection objects requires resources similar to standard I/O connections.

Table 9 – ForwardOpen setting options for safety connections

O-to-T connection type	T-to-O connection type	Transport type/ trigger	App path 1	App path 2a	App path 3a	Max consumer number	Safety Validator connection type	Key decision factors	Target role
Point-to-point	Point-to-point	Application, Class 0, Client	Vendor defined Null Path	Vendor defined Null Path	Safety Input App. object	1	Point-to-point Producer (Input)	2-point-to-point + Client	Producing connection: Safety data Consumer connection: Coordination
Point-to-point	Point-to-point	Application, Class 0, Server	Vendor defined Null Path	Safety Output App. object	Vendor defined Null Path	1	Point-to-point Consumer (Output)	2- point-to-point + Server	Consuming connection: Safety data Producing connection: coordination
Point-to-point	Multipoint	Application, Class 0, Client	Vendor defined Null Path	Vendor defined Null Path	Safety Input App. object	2-15	Multipoint Producer CP 2/3	2 – Multipoint + 1 point-to-point + Client	Producing connection: Safety data Producing connection: Correction Consuming connection: coordination
Point-to-point	Multipoint	Application, Class 0, Client	Vendor defined Null Path	Vendor defined Null Path	Safety input app. object	2-15	Multipoint Producer CP 2/2	1 – Multipoint + 1 point-to-point + Client	Producing connection: Safety data + correction Consuming connection:

O-to-T connection type	T-to-O connection type	Transport type/ trigger	App path 1	App path 2 ^a	App path 3 ^a	Max consumer number	Safety Validator connection type	Key decision factors	Target role
									coordination
Multipoint	Point-to-point	Application, Class 0, Server	Vendor defined Null Path	Safety output app. object	Vendor defined Null Path	2-15	Multipoint Consumer CP 2/3	2 – Multipoint + 1 point-to-point + Server	Consuming connection: Safety data Consuming connection: Correction Producing connection: coordination
Multipoint	Point-to-point	Application, Class 0, Server	Vendor defined Null Path	Safety output app. object	Vendor defined Null Path	2-15	Multipoint Consumer CP 2/2	1 – Multipoint + 1 point-to-point + Server	Consuming connection: Safety Data + correction Producing connection: coordination

^a Input and Output with respect to the network.

^b Devices configurable via EDS shall define an application Null Path in the EDS file.

Table 10 defines which network connection parameters to use for all three connections associated with each safety connection type.

Table 10 – Network connection parameters for safety connections

Target type	Network connection parameter	Proper parameter value	Description
Point-to-point producer	O-to-T	4406 _{hex}	Point-to-point high priority, fixed
	T-to-O	4408 _{hex}	Point-to-point high priority, fixed
	Time Correction	0000 _{hex}	Not used
Point-to-point producer	O-to-T	4408 _{hex}	Point-to-point high priority, fixed
	T-to-O	4406 _{hex}	Point-to-point high priority, fixed
	Time Correction	0000 _{hex}	Not used
Multipoint producer	O-to-T	4406 _{hex}	Point-to-point high priority, fixed
	T-to-O	240E _{hex}	Multipoint high priority, fixed
	Time Correction	2406 _{hex}	Multipoint high priority, fixed

Successful SafetyOpen responses shall return the connection identifiers used, as well as the Consumer_Number assigned to the originator (always FFFF_{hex} for point-to-point connections).

6.3.11 Application reply data in a successful SafetyOpen response

The format for a SafetyOpen response for Success is the same as the Forward_Open_Reply as defined in IEC 61158-6-2. The data return is defined in Table 11.

The target of a SafetyOpen shall return the data defined in Table 12 in the Application Reply field of the SafetyOpen response. The consumer number and PID reply data is only of significance when the originator is the consumer.

Table 11 – CP 2/3 Safety target application reply (size: 10 octets)

Parameter name	Data type	Description
Consumer_Number	UINT	
PID/CID	STRUCT of:	Target Identifier used to generate a PID or CID seed value
Target Vendor Id	UINT	Target's Vendor Id
Target Device Serial Number	UDINT	Target's Device Serial Number
Target Connection Serial Number	UINT	Connection Serial Number Target assigned to the connection. The Safety Validator instance Id shall be used

Table 12 – SafetyOpen target application reply (size: 16 octets)

Parameter name ^a	Data type	Description
Consumer_Number	UINT	
PID/CID	STRUCT of:	Target Identifier used to generate a PID or CID seed value
Target Vendor Id	UINT	Target's Vendor Id
Target Device Serial Number	UDINT	Target's Device Serial Number
Target Connection Serial Number	UINT	Connection Serial Number Target assigned to the connection. The Safety Validator instance Id shall be used
Time_Correction_Connection_ID	UDINT	Connection ID selected by the Target for the Time Correction connection. If chosen by the originator, then the value is echoed back
Time_Correction_API	UDINT	Returns same value received in the Request packet
^a These fields are always present in the Application Reply field. When a CP 2/3 Safety target receives a SafetyOpen that is not multipoint (thus no Time Correction connection is established) the Time_Correction_Connection_ID parameter shall be set to FFFF _{hex} . If the target is not a multipoint producer, the consumer number is set to FFFF _{hex} .		

6.3.12 Unsuccessful SafetyOpen response

The SafetyOpen has some special error cases that do not map to the existing set of error codes, and extends the definition of existing error codes.

Table 13 defines new extended error codes for safety and shows the extended definition of existing error codes.

Table 13 – New and extended error codes for safety

General status	Extended status	Explanation
01 _{hex}	0105 _{hex}	Ownership conflict or OUNID mismatch. The configuration is already owned by another originator
01 _{hex}	0106 _{hex}	Ownership conflict or OUNID mismatch. The output connection was already owned by another originator
01 _{hex}	0205 _{hex}	Parameter Error in Unconnected Send service or Parameter Error in SafetyOpen or SafetyClose
01 _{hex}	801 _{hex}	Incompatible Multipoint RPI. An existing connection has been established at a different RPI
01 _{hex}	802 _{hex}	Invalid Safety Connection size
01 _{hex}	803 _{hex}	Invalid Safety Connection format
01 _{hex}	804 _{hex}	Invalid Time Correction Connection Parameters
01 _{hex}	805 _{hex}	Invalid Ping Interval EPI Multiplier
01 _{hex}	806 _{hex}	Time Coordination Msg Min Multiplier
01 _{hex}	807 _{hex}	Network Time Expectation Multiplier
01 _{hex}	808 _{hex}	Timeout Multiplier
01 _{hex}	809 _{hex}	Invalid Max Consumer Number
01 _{hex}	80A _{hex}	Invalid CPCRC
01 _{hex}	80B _{hex}	Time Correction Connection Id invalid
01 _{hex}	80C _{hex}	SCID Mismatch. The SCID was non-zero and did not match the value in the target
01 _{hex}	80D _{hex}	TUNID not set. Device is out-of-box and TUNID has not been set, so connections are not allowed
01 _{hex}	80E _{hex}	TUNID Mismatch. The TUNID provided does not match. The message was likely routed to this node in error
01 _{hex}	80F _{hex}	Configuration operation not allowed

Event/Error Matrix Table 14 provides guidance for which error codes to use for possible errors detected in processing a SafetyOpen. This is not intended to be an exhaustive list.

Table 14 – SafetyOpen error event guidance table

Event	Error type	General status	Extended status
The target device received a SafetyOpen request in states of Safety Supervisor object other than Idle and Executing.	Configuration operation not allowed	01 _{hex}	080F _{hex}
	Device not configured	01 _{hex}	110 _{hex}
Safety Connection Type (name) Invalid	Safety parameter error – Invalid Connection Type	01 _{hex}	315 _{hex}
Safety Connection Size Invalid	Safety parameter error – Invalid Safety Connection Size	01 _{hex}	0802 _{hex}
Safety Connection Format Invalid	Safety parameter error – Invalid Safety Connection Format	01 _{hex}	0803 _{hex}
Time Correction Connection Parameters Invalid	Safety parameter error – Invalid Time Correction Connection Parameters	01 _{hex}	0804 _{hex}
Ping Interval EPI Multiplier was out of the range	Safety parameter error – Invalid Ping Interval EPI Multiplier	01 _{hex}	0805 _{hex}
Time Coordination Message Min Multiplier was out of the range	Safety parameter error – Time Coordination Msg Min Multiplier	01 _{hex}	0806 _{hex}
Time Expectation Multiplier was out of the range	Safety parameter error – Time Expectation Multiplier	01 _{hex}	0807 _{hex}
Timeout Multiplier was out of the range	Safety parameter error – Timeout Multiplier	01 _{hex}	0808 _{hex}
Max Consumer Number not within the limits in Multipoint-Producer	Safety parameter error – Invalid Max Consumer Number	01 _{hex}	0809 _{hex}
CPCRC value in SafetyOpen and calculated CPCRC value.	Safety parameter error – Invalid CPCRC	01 _{hex}	080A _{hex}
Time Correction Connection Id Invalid	Safety parameter error – Time Correction Connection Id invalid	01 _{hex}	080B _{hex}
RPI requested is incompatible with previously established multipoint connections	Incompatible multipoint RPI	01 _{hex}	801 _{hex}
O-to-T RPI, T-to-O RPI, or Time Correction RPI is not supported	RPI not supported	01 _{hex}	0111 _{hex}
All Safety Validator Instances are being used.	Connection Manager or connection object cannot support any more connections.	01 _{hex}	0113 _{hex}

6.3.13 ForwardClose for safety

There are no additional behaviors defined for the ForwardClose service when used to close a Safety connection. In the following Safety Requirements, the term “SafetyClose” refers to both the ForwardClose of the Connection Manager object and the SafetyClose of the Connection object.

The SafetyClose Service shall be used to close connections with a Safety Targets (and in the case of a ForwardClose all other nodes in the connection path).

The SafetyClose request shall remove a connection from all the nodes participating in the original connection.

The SafetyClose request shall cause all resources in all nodes participating in the connection to be deallocated, including connection IDs, bandwidth, and internal memory buffers.

Targets receiving the SafetyClose service request shall close the connection who's Originator Vendor ID, Connection Serial Number, and Originator Serial Number parameters match. Additional information provided by this service (ie, Connection Path) shall be ignored by the target.

A SafetyClose Success reply shall be returned after the connection has been deleted at the target.

When a SafetyClose success reply is received, the originator and each intermediate node along the path, closes the connection and releases resources associated with that connection when the success response is received. But the originator shall also close the connection if no response is received within a vendor selected wait period.

6.4 Identity object

6.4.1 General

Class Code: 01hex

This subclause describes changes to the Identity object to support safety.

All CPF 2 safety devices shall replace the Identity object state behavior with the behavior defined in the Safety Supervisor object.

6.4.2 Changes to common services

Table 15 defines the services of the Identity object that require updates.

Table 15 – Identity object common service changes

Service code	Need in implementation		Service name	Description of service
	Class	Instance		
05 _{hex}	Optional	Conditional ^a	Reset	Invokes the Reset Service for the device
^a Conditional – The Identity object Reset service shall not be supported in safety devices (reply to any requests with “Service Not Supported” error). The service shall be replaced by the Reset service defined by the Safety Supervisor.				

6.5 Link objects

6.5.1 DeviceNet object changes

6.5.1.1 Additional attribute for safety

Class Code: 03hex

This subclause defines the changes and additions required of the DeviceNet object to support safety. An attribute shall be added to allow the Safety Network Number (SNN) to be read. The definition for this attribute is defined in Table 16.

Table 16 – New DeviceNet object instance attribute

Attrib ID	Need in implem.	Access rule	Name	Data type	Description of attribute	Semantics of value
11	Required	Get	Safety Network Number	6 OCTETS	System-wide unique number for the subnet this device resides on.	Attribute shall contain the SNN portion of the TUNID in the Safety Supervisor. See 8.2 for further definition of this attribute

Devices implementing FSCP 2/1 on CP 2/3 shall support attributes 11 of the DeviceNet object.

6.5.1.2 Quick Connect restriction for safety

All CP 2/3 safety devices shall not support the Quick Connect feature. The Quick Connect attribute shall not be supported and shall respond to any Get/Set attribute service with the error code "Attribute not supported".

6.5.2 TCP/IP Interface object changes

Class Code: F5hex

This subclause defines the changes and additions required of the TCP/IP Interface object to support safety. An attribute shall be added to allow the Safety Network Number (SNN) to be read. The definition for this attribute is defined in Table 17.

Table 17 – New TCP/IP Interface object Instance Attribute

Attrib ID	Need in implem.	Access rule	Name	Data type	Description of attribute	Semantics of value
7	Required	Get	Safety Network Number	6 OCTETS	System-wide unique number for the subnet this device resides on.	Attribute shall contain the SNN portion of the TUNID in the Safety Supervisor. See 8.2 for further definition of this attribute

Devices implementing FSCP 2/1 on CP 2/2 Safety shall support attribute 7 of the TCP/IP Interface object.

6.6 Safety Supervisor object

6.6.1 General

Class Code: 39 hex

The Safety Supervisor is the core object of FSCP 2/1 Safety Devices. There are two fundamental Safety Supervisor implementations defined:

- **Baseline Supervisor** – defined as a supervisor implementation that supports all the “required” attributes and services defined by this object definition. The baseline Supervisor includes all state machine behavior excluding state event processing for services not required (see 6.6.8);
- **SNCT Supervisor** – defined as a supervisor implementation that supports all the Baseline functionality plus the attributes and services that are marked “conditional” on SNCT support.

All safety devices on a FSCP 2/1 safety network shall support the baseline Safety Supervisor definition.

Because many of the parameters in the safety supervisor object are deemed safety critical, the attributes of this object shall have a higher integrity than standard object data.

The state behavior of the Safety Supervisor object shall control the state of entire device. All objects in a safety device are required to be subservient to the states and commands of the Safety Supervisor to manage its functions and behaviors.

The Safety Supervisor object centralizes application object state definitions and related status information, exception status indications (alarms and warnings), and defines a behavior model which is assumed by objects identified as belonging to safety devices. That is, if a reset is requested of the Safety Supervisor object instance, it is performed by this object instance as well as all of its associated application objects.

The Identity object shall get its state information from the Safety Supervisor object. A reset request to the Identity object is not supported by FSCP 2/1 Safety devices (see 6.4 and shall be replaced by a reset request to the Safety Supervisor object which has additional qualifiers to execute the request. Further relationships are specified in 6.6.8.

Additionally, some instance attributes are reserved to allow for future extension to implement profiles defined for the semiconductor industry.

6.6.2 Safety Supervisor class attributes

The Class attributes used by the Safety Supervisor are defined in Table 18.

Table 18 – Safety Supervisor class attributes

Attribute ID	Need in implementation	Access rule	Name	Date type	Description of attribute
1 thru 96	These class attributes are described in IEC 61158-6-2				
97-98	Reserved by FSCP 2/1 for future use				
99	Conditional ^a	Get	Subclass	UINT	Identifies a subset of additional class attributes, services, and behaviors
^a If the value of Subclass is 00 which identifies “no subclass”, then this attribute should be in implementation, otherwise, this attribute shall be in the implementation.					

6.6.3 Subclasses

Each class level subclass defines a unique meaning for an overlapping range of class attribute IDs and/or class service IDs. The range for subclass definitions begins at ID 96 and numbers downward for attributes, and ID 63_{hex} and numbers downward for services. The subclass for a given class is identified by the value of its subclass class attribute. There are presently no subclasses defined for the class instance.

6.6.4 Safety Supervisor instance attributes

The instance attributes for the Safet Supervisor are defined in Table 19.

Table 19 – Safety Supervisor instance attributes

Attr ID	Need in implementation	Access rule	NV	Name	Date type	Description of attribute
1	Optional	Get	NV	Number of Attributes	USINT	Number of Attributes supported by the object instance
2	Optional	Get	NV	Attribute List	ARRAY of USINT	List of attributes supported by the object instance
3	Reserved					
4	Reserved					
5	Optional	Get	NV	Manufacturer Name	Short String	ASCII Text, Max. 20 characters, see 6.6.5.1
6	Optional	Get	NV	Manufacturer Model Number	Short String	ASCII Text, Max. 20 characters, Manufacturer specified
7	Optional	Get	NV	Software Revision Level	Short String	ASCII Text, Max. 6 characters, see 6.6.5.2
8	Optional	Get	NV	Hardware Revision Level	Short String	ASCII Text, Max. 6 characters, see 6.6.5.3
9	Optional	Get	NV	Manufacturer Serial Number	Short String	ASCII Text, Max. 30 characters, manufacturer specified, see 6.6.5.4
10	Optional	Get	NV	Device Configuration	Short String	ASCII Text, Max. 50 Characters, manufacturer specified. Optional additional information about the device configuration
11	Required	Get	V	Device Status	USINT	see 6.6.5.5
12	Required	Get	V	Exception Status	SWORD	see 6.6.5.6
13	Conditional based on Exception Status Bit 7	Get	V	Exception Detail Alarm	STRUCT of:	A Structure of 3 structures containing a bit-mapped representation of the alarm detail
				Common Exception Detail	STRUCT of:	
				Size	USINT	Number of Common Exception Detail Octets
				Detail	ARRAY of:	see 6.6.5.7
				Detail	SWORD	see 6.6.5.7
				Device Exception Detail	STRUCT of:	

Attr ID	Need in implementation	Access rule	NV	Name	Date type	Description of attribute
				Size	USINT	Number of Device Exception Detail Octets
				Detail	ARRAY of:	Manufacturer specified
				Detail n	SWORD	Manufacturer specified
				Manufacturer Exception Detail	STRUCT of:	
				Size	USINT	Number of Manufacturer Exception Detail Octets
				Detail	ARRAY of:	Manufacturer specified
				Detail n	SWORD	Manufacturer specified
14	Conditional Based on Exception Status Bit 7	Get	V	Exception Detail Warning	STRUCT of:	A Structure of 3 structures containing a bit-mapped representation of the warning detail
				Common Exception Detail	STRUCT of:	
				Size	USINT	Number of Common Detail Octets
				Detail	ARRAY of:	see 6.6.5.8
				Detail n	SWORD	see 6.6.5.8
				Device Exception Detail	STRUCT of:	
				Size	USINT	Number of Device Detail Octets
				Detail	ARRAY of:	Manufacturer specified
				Detail n	SWORD	Manufacturer specified
				Manufacturer Exception Detail	STRUCT of:	
				Size	USINT	Number of Manufacturer Detail Octets
				Detail	ARRAY of:	Manufacturer Specified
				Detail n	SWORD	Manufacturer Specified
15	Required	Set	NV	Alarm Enable	BOOL	see 6.6.5.12
16	Required	Set	NV	Warning Enable	BOOL	see 6.6.5.12
17	Optional	Set	a	Time	DATE_ AND_ TIME	The value of the Device's internal real-time clock.
18	Optional	Get	NV	Clock Power Cycle Behavior	USINT	Describes the behavior of the device's internal real-time clock (the "Time" attribute) during a power cycle 0 = [default] clock always resets during power cycle 1 = clock value is stored in

Attr ID	Need in implementation	Access rule	NV	Name	Date type	Description of attribute
						nonvolatile memory at power down 2 = clock is battery-backed and runs without device power. 3-255 = Reserved
19	Optional	Get	NV	Last Maintenance Date	DATE	The date on which the device was last serviced
20	Optional	Get	NV	Next Scheduled Maintenance Date	DATE	The date on which it is recommended that the device next be serviced
21	Optional	Get	NV	Scheduled Maintenance Expiration Timer	INT	see 6.6.5.13
22	Conditional – Required if Calibration Expiration is supported	Set	NV	Scheduled Maintenance Expiration Warning Enable	BOOL	see 6.6.5.14
23	Optional	Get	NV	Run Hours	UDINT	An indication of the number of hours that the device has had power applied.
24	Conditional ^c	Set only by "Configuration Lock/Unlock" Service Gettable	NV	Configuration Lock	BOOL	Marks the device configuration contained within as Verified and locked, or alternatively as unverified, unlocked, and changeable.
25	Conditional ^c	Get	NV	Configuration UNID	10 octets	CFUNID - Identifies the owner of a Device Configuration. All FF _{hex} = Tool-only configuration. 0 = un-owned, accept any owner
26	Required	Get	NV	Safety Configuration Identifier	10 octets	The SCID is comprised of the Safety Configuration CRC + Safety Configuration Time Stamp. This is the signature for the Configuration
27	Required	Get	NV	Target UNID	10 octets	The current UNID of the device.
28	Conditional ^d	Get	NV	Output Connection Point Owners		
				Number of Array Entries	UINT	Number of OCPUNID STRUCT entries
				Output Owners	ARRAY of STRUCT	

Attr ID	Need in implementation	Access rule	NV	Name	Date type	Description of attribute
				OCPUNID	10 octets	The owner UNID for the output resource 0 = un-owned, accept any owner (default)
				ePath Size	USINT	Path size, number of octets
				Application Resource	Packed ePath	The path to owned resource (i.e. 20 04 24 01 - Assembly class, instance 1)
29	Conditional ^c	Get		Proposed TUNID	10 octets	The UNID value that an originator is attempting to set in the device.
97-98	Reserved by FSCP 2/1 for Future Use					
99	Conditional ^b	Get	NV	Subclass	UINT	Identifies a subset of additional instance attributes, services, and behaviors
^a The nonvolatile behavior of the Time attribute is conditional on the value of the Clock Power Cycle Behavior attribute. ^b If the value of Subclass is 00 which identifies "no subclass", then this attribute should be in implementation, otherwise, this attribute shall be in implementation. ^c Safety Devices which support the SNCT interface (i.e. not configured by the SafetyOpen or some vendor specific means) shall support this attribute. ^d Conditional – the Output Connection Owner attribute shall be supported for all safety output assemblies.						

6.6.5 Semantics

6.6.5.1 Manufacturer name

The Manufacturer Name attribute, identifies the manufacturer of the device.

The Manufacturer Name attribute is not guaranteed, by specification, to be unique. Therefore, it is not a substitute for the corresponding attribute of the Identity object and should not be used for identification purposes.

6.6.5.2 Software revision level

This is an ASCII coded text string representing the revision of the software corresponding to the specific device identified by the Identity object.

6.6.5.3 Hardware revision level

This is an ASCII coded text string representing the revision of the hardware which is identified by the Identity object. The manufacturer of the device shall control this revision such that modifications to the device hardware may be tracked.

6.6.5.4 Manufacturer's serial number

This attribute is a string representation of the vendor's serial number of the device, formatted to fit most manufacturing tracking systems. It is not required that this attribute match the Identity object's serial number which is used to uniquely identify the device in the network environment.

6.6.5.5 Device status

This attribute represents the current state of the device. Its value changes as the state of the device changes. The Device status attribute state values are defined in Table 20.

Table 20 – Device status attribute state values

Attribute value	State
0	Undefined
1	Self-Testing
2	Idle
3	Self-Test exception
4	Executing
5	Abort
6	Critical Fault
7	Configuring
8	Waiting for TUNID
9-50	Reserved by FSCP 2/1
51-99	Device specific
100-255	Vendor specific

6.6.5.6 Exception status

A single octet attribute whose value indicates that the status of the alarms and warnings for the device. This indication may be provided in one of two methods: Basic or Expanded.

For the Basic Method, bit seven of the Exception Status attribute is set to zero; all exceptions are reported exclusively through communication of this Exception Status attribute. The format of bits zero through six in this mode is device specific; the format may be further specified in an appropriate device profile specification; if it is not specified, then the format of bits zero through six is equivalent to that specified for the expanded method.

For the Expanded Method, bit seven of Exception Status attribute is set to one; exceptions are reported through the communication of this Exception Status attribute, formatted as specified in Table 21. In addition, the Exception Detail attributes are supported. The Exception Status bits are determined by a logical “OR” of the related Exception Detail bits, as indicated.

Table 21 – Exception status attribute format

Bit	Exception status bit map, Bit 7 Set to 0 Function	Exception status bit map, Bit 7 set to 1 Function
0	Device Specific definition (see Device Profile)	ALARM/Device-common ^a
1		ALARM/Device-specific
2		ALARM/Manufacturer-specific
3		Reserved – set to 0
4		WARNING/device-common ^a
5		WARNING/device-specific
6		WARNING/manufacturer-specific
7	0 = Basic Method	1 = Expanded Method
^a The alarm or warning is not specific to the device type or device type manufacturer.		

6.6.5.7 Exception detail alarm and exception detail warning

The formats of these two attributes are identical. Therefore, they are described together here:

Attributes that relate the detailed status of the alarms or warnings are associated with the device. Each attribute is a structure containing three members; these three members respectively relate the detailed status of exceptions that are common (i.e., not device-specific), device-specific but not manufacturer-specific, and manufacturer-specific. The common detail is defined below. The device-specific detail is defined in the appropriate Device Profile. The manufacturer-specific detail is defined by the manufacturer. A SIZE value of zero indicates that no detail is defined for the associated exception detail structure.

Each of the three structure members is defined as a structure containing an ordered list (i.e., array) of OCTETS of length SIZE, and an unsigned integer whose value is SIZE. Each of the octets in each array has a specific mapping. This mapping is formatted as 8 bits representing 8 independent conditions, whereas a value of 1 indicates that the condition is set (or present), and a value of 0 indicates that the condition is cleared (or not present). If a device does not support an exception detail, the corresponding bit is never set. The bitmaps for alarms and warnings in the corresponding attributes are structured in parallel so that a condition may have either alarm or warning set depending on severity. If a condition inherently cannot be both alarm and warning, then the parallel bit position corresponding to the other state shall remain "0".

The existence of an exception detail variable structure is dependent on the value of the Exception Status Attribute; the existence of an exception detail variable structure is only required if bit seven of the Exception Status attribute is set to 1 (indicating Expanded method reporting) and the bit (among bits zero through six) of the Exception Status attribute corresponding to the particular exception type is also set to 1.

6.6.5.8 Common exception detail

This structure relates exception conditions (i.e., alarms or warnings) which are common to all devices. The Detail element of the structure is an ordered list (i.e., array) of octets of length SIZE which is the value of the structure element Size. For each octet in the Detail field, all bits which are not identified are reserved for future standardization.

The first octet in this attribute is CommonExceptionDetail[0]. Additional exception details, if provided, are named CommonExceptionDetail[1], CommonExceptionDetail[SIZE]. The specific exception associated with each of the bitmaps is given in Table 22. The SIZE for this revision is two (2). The criteria details for each exception condition are outside the scope of this part.

6.6.5.9 Common exception detail attribute values

Table 22 – Common exception detail attribute values

Common Exception Detail [0] ^a	Common Exception Detail [1] ^a
Internal diagnostic exception	power supply over current
Microprocessor exception	reserved power supply
EPROM exception	power supply output voltage
EEPROM exception	power supply input voltage
RAM exception	scheduled maintenance due
Reserved	notify manufacturer
Internal real-time exception	reset exception
Reserved	Reserved
^a If a device does not support an exception detail, the corresponding bit is never set.	

6.6.5.10 Device exception detail

This structure, similar in form to Common Exception Detail, relates exception conditions which are specific to individual devices on the network and are defined in their respective device profiles. The Detail element of the structure is an ordered list (i.e., array) of octets of length SIZE which is the value of the structure element size. For a detailed description of this attribute, consult the appropriate specific device profile.

6.6.5.11 Manufacturer exception detail

This structure (see Table 23), similar in form to Common Exception Detail, relates exception conditions which are specific to the manufacturers of individual devices on the network and are defined by them in their product documentation. The Detail element of the structure is an ordered list (i.e., array) of octets of length SIZE which is the value of the structure element Size. For a detailed description of this attribute, consult the appropriate specific device manufacturer documentation.

Table 23 – Exception detail format summary

Detail component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Common exception detail size	0	0	0	0	0	0	0	0
Common exception detail 0	Res.	Real-time fault	Res.	Data memory	Non-volatile memory	Code memory	Micro-process or	Diagnostic
Common exception detail 1	Res.	Reset exception	Notify vendor	Scheduled maint. due	PS input voltage	PS output voltage	Res.	PS over current
Device exception detail size	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a
Device exception detail 0	--	--	--	--	--	--	--	--
Device exception detail n	--	--	--	--	--	--	--	--
Manufacturer exception detail size	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a
Manufacturer exception detail 0	--	--	--	--	--	--	--	--
Manufacturer exception detail n	--	--	--	--	--	--	--	--

^a Consult the appropriate specific device manufacturer documentation for complete definition.

6.6.5.12 Alarm enable and warning enable

These Boolean attributes are used to enable (1) or disable (0) the Safety Supervisor object's process of setting Exception bits. When disabled, corresponding bits are never set; and, if they were set, disabling clears them. Also, alarm and warning states are not retained; when enabled, bits shall be set only if the corresponding condition is true.

The default state for these Enable attributes is enabled (1).

This optional attribute represents the value of the time and date as maintained by the device's real-time clock with a resolution of one millisecond.

The default value for the Time attribute is zero (0), corresponding to 00:00:00, 1970-01-01.

6.6.5.13 Scheduled maintenance expiration timer

This attribute, with a resolution of one hour, is used to cause a warning which indicates that a device calibration is due. A Safety Supervisor timer decrements this attribute once per hour while power is applied. When the attribute is no longer positive and the Scheduled Maintenance Expiration Warning Enable attribute is set to enabled, a Scheduled Maintenance Expiration Warning condition is generated. This causes the Scheduled Maintenance Due Warning bit to be set.

The Scheduled Maintenance Expiration Timer attribute, when implemented, shall not wrap; when the attribute reaches its most negative value, it no longer decrements. The Scheduled Maintenance Expiration Timer, when implemented, shall continue to decrement irrespective of the state of the Scheduled Maintenance Expiration Warning Enable attribute. The Scheduled Maintenance Expiration Timer, when implemented, shall be maintained in nonvolatile memory.

6.6.5.14 Scheduled maintenance expiration warning enable

This Boolean attribute is used to enable (1) or disable (0) the Safety Supervisor object's process of setting the Scheduled Maintenance Due Exception bit. When disabled, the corresponding bit is never set; and, if it was set, disabling clears it. When enabled, the bit shall be set only if the corresponding condition is true.

The default state for this Enable attribute is enabled (1).

6.6.5.15 Configuration lock

The Configuration Lock attribute, when implemented with the SNCT interface, shall be used to mark the device configuration as verified and to lock down the configuration; 0 = Unlocked (unverified and changeable), 1 = Locked (verified and not changeable).

The default value for the Configuration Lock attribute is 0 = unlocked. Its primary use is for safety devices that are configured by the SNCT (i.e. not configured by the SafetyOpen). After the tool has configured the device and the user has validated the configuration, the software sets this flag to lock out any further changes. It is an option that this flag be protected by password which requires user verification before the configuration can be unlocked and changed.

While a device has the Configuration Lock attribute set, it shall reject Configure_Requests, Type 1 and 2 Safety Reset, and any valid Form 1 SafetyOpen.

6.6.5.16 Configuration UNID (CFUNID)

The Configuration UNID attribute shall be implemented as part of the SNCT interface. The Configuration UNID identifies the owner of the configuration in this safety device. The Configuration UNID attribute can be set by a number of means. The Configuration UNID attribute shall have the following special meanings assigned:

- All octets set to 00_{hex} = No owner, accept any,
- All octets set to FF_{hex} = Software Tool is the assigned owner.

The default for the Configuration UNID shall be all octets set to 00_{hex}.

When the Configuration UNID attribute is set to "No owner, accept any", FSCP 2/1 Safety Devices shall capture the first configuration source (Form 1 and/or Configure_Request) as the owner. (see Table 24).

The CFUNID value is used in conjunction with the OUNID value provided in either the Configure_Request or a Form 1 SafetyOpen. See Table 24 for the definition of how devices respond to the various services based on what the CFUNID is set to at the time the service is received. SNCT shall always use all FF_{hex} for the OUNID.

EXAMPLE If the CFUNID is set to all FF_{hex}, then the OUNID of the Configure_Request shall equal this value to process the command.

Table 24 – Summary of device behavior for various CFUNID values

Service	Value of CFUNID		
	FF _{hex}	00 _{hex}	OUNID
Configure_Request (FF _{hex}) from SNCT	Accept	Accept Set CFUNID = FF _{hex}	Fail
Configure_Request (OUNID)	Fail	Accept Set CFUNID = OUNID	Accept only if CFUNID = OUNID Fail otherwise
Form 1 SafetyOpen	Fail	Accept Set CFUNID = OUNID	Accept only if CFUNID = OUNID Fail otherwise

6.6.5.17 Safety Configuration Identifier (SCID)

The Safety Configuration Identifier attribute is the signature on the safety configuration. This attribute is a concatenation of the Safety Configuration CRC (SCCRC) + Safety Configuration Time Stamp (SCTS). These values are obtained from either the SafetyOpen or from the Safety Configuration Software during configuration validation.

The default for SCID attribute in the Safety Supervisor shall be 0.

When a device enters Configuring mode, the SCID attribute shall be set to 0 and maintained at this value (through power cycles) until a successful Validate has been executed.

```
typedef struct _S_SCID
{
    DWORD    dwCRC;    //calculated by device
    DWORD    dwTime;   // set by software
    WORD     wDate;
} S_SCID;
```

6.6.5.18 Target UNID (TUNID)

The Target UNID is an NV attribute that reflects the UNID saved with the configuration.

The Target UNID attribute in the Safety Supervisor shall always reflect either the default or the current Target UNID saved through the UNID setting process.

The UNID of a device shall be stored to NV as part of the UNID setting operation.

The UNID of a device shall be stored to the DeviceNet object or TCP/IP Interface object SNN attribute as part of the UNID setting operation.

A device in Manufacturers default state shall have an invalid UNID value in its NV (for example FF_{hex}, FF_{hex}, FF_{hex}, FF_{hex}, FF_{hex}, FF_{hex}).

The UNID of a device shall be read from NV during power up and loaded into an attribute of the Safety Supervisor object and Safety_Network_Number attribute of the DeviceNet object or TCP/IP Interface object.

If the UNID represents a valid Number (not equal to out-of-box), this number shall be compared to the MACID or IP address object attributes. In case of a match the device continues its startup procedure. In case of a mismatch the Device transits to Abort.

When a device faults from a Switch-NV setting mismatch, it shall allow either a reset to out-of-box, or a Recover request issued to the Safety Supervisor to clear the Abort. In both cases, the device shall move to the “Waiting for TUNID” state.

The reading of the UNID attribute in the Safety Supervisor shall be done with safety integrity.

The applying of configuration data and UNID to NV shall be done with safety integrity.

6.6.5.19 Proposed TUNID (OUNID)

The default value for the Proposed TUNID attribute shall be all FF_{hex}.

While a device is going through the “ProposeTUNID/ApplyTUNID” process, this attribute reflects the value being proposed. Once the apply operation is complete, this attribute shall be set back to the out-of-box default of all FF_{hex}.

6.6.5.20 Output Connection Owner (OCPUNID)

This attribute is only used for safety connection to outputs. Safety connections to outputs define an owner to prevent errant connections from hijacking an output resource in a validated safety system. This owner Id identifies which originator safety device has been given ownership rights to this safety output connection. This parameter is obtained from the Safety Segment Parameters in the SafetyOpen. The OCPUNID value could be the same as Attribute 25 of the Safety Supervisor (CFUNID) when the device is also configured as part of the SafetyOpen. Since device may have more than one output resource and may allow for separate owners of each, this attribute is arranged as a structure to allow for each resource to be separately maintained. The resource is identified by a Logical Segment address.

The default for all OCPUNIDs shall be 0 (accept any owner) with the ePath for each pointing to the respective resource.

6.6.6 Subclasses

Each instance level subclass defines a unique meaning for an overlapping range of instance attribute IDs and/or instance service IDs. The range for subclass definitions begins at ID 96 and numbers downward for attributes, and ID 63_{hex} and numbers downward for services. The subclass for a given instance is identified by the value of its Subclass instance attribute. There are no subclasses currently defined for the instances.

6.6.7 Safety Supervisor common services

6.6.7.1 Common services overview

The common services supported by the Safety Supervisor are defined in Table 25. The Safety Supervisor object specific services are defined in Table 26.

Table 25 – Safety Supervisor common services

Service code	Need in implementation		Service name	Description of service
	Class	Instance		
0E _{hex}	Conditional	Required	Get_Attributes_Single	Returns the contents of the specified attribute
010 _{hex}	n/a	Required	Set_Attributes_Single	Modifies an attribute value
0D _{hex}	n/a	Conditional ^a	Apply	Applies a pending configuration to NV memory, and moves the device to the Idle state.
15 _{hex}	n/a	Optional	Restore	Restores last applied configuration in devices that support it
^a Safety Devices which support the SNCT interface (i.e. not configured by the SafetyOpen or some vendor specific means) shall support this service.				

Any Safety Supervisor instance service may be requested internally by the device as specified by the manufacturer.

Table 26 – Safety Supervisor object specific services

Service code	Need in implementation		Service name	Description of service
	Class	Instance		
4B _{hex}			Reserved	
4C _{hex}	n/a	Optional	Recover	Moves the device out of the Abort State
4E _{hex}	n/a	Optional	Perform_Diagnostics	Causes the device to perform a set of diagnostic routines if appropriate
4F _{hex}	n/a	Conditional ^a	Configure_Request	Moves the device to the Configuring State if appropriate. This service requires a password be provided to be executed.
50 _{hex}	n/a	Conditional ^a	Validate_Configuration	Causes the device to perform a validation check of the pending configuration. This service requires the software to provide the SCID (CRC + Time Stamp)
51 _{hex}	n/a	Conditional ^a	Set_Password	Used to set a password attribute in the Supervisor object.
52 _{hex}	n/a	Conditional ^a	Configuration_Lock/Unlock	This service acts upon the Configuration Lock attribute in the Safety Supervisor. This service requires a password be provided to be executed.
53 _{hex}	n/a	Conditional ^c	Mode Change	Shall be in implementation for Logic devices which support "executing" independent of I/O connections. A password is required to execute mode changes in safety devices.
54 _{hex}	n/a	Type 0 Required, Type 1 and 2 Conditional ^a	Safety Reset	Resets the device in a manner similar to the Identity object, except that a password is required to execute.
55 _{hex}	n/a	Conditional ^b	Reset Password	Used to reset the password in

Service code	Need in implementation		Service name	Description of service
	Class	Instance		
				a device. Data contents of this message is vendor specific
56 _{hex}	n/a	Conditional ^d	Propose_TUNID	Used to initiate the setting of the TUNID in an out-of-box device
57 _{hex}	n/a	Conditional ^d	Apply_TUNID	Used to complete the TUNID setting operation.
^a Shall be in implementation for FSCP 2/1 safety devices which support the SNCT interface. ^b Conditional on supporting the password service as part of the SNCT interface ^c Shall be in implementation for FSCP 2/1 safety devices which support the SNCT interface and are Logic devices which support “executing” independent of I/O connections. A password is required to execute mode changes in safety devices. ^d Shall be in implementation for FSCP 2/1 safety devices which support the SNCT interface but highly recommended for devices which are configurable by Form 1 SafetyOpen but don't support the SNCT interface.				

Any Safety Supervisor instance service may be requested internally by the device as specified by the manufacturer. Generally, these requests will be generated as the result of an event such as the activation of a button or external contact closure.

6.6.7.2 Recover service

The Recover service is used to transition the device application objects, out of the recoverable fault state, to the idle state. This service request may be originated internally, from application objects.

6.6.7.3 Perform_Diagnostics service

The Perform_Diagnostics service is used to instruct the Safety Supervisor object to perform a diagnostic test. A diagnostic test is either of type common or device-dependent. Common diagnostic tests could include RAM, EPROM, non-volatile memory, and communications. The structure of common type diagnostic tests is implementation-specific. Details of device-dependent diagnostics are outside the scope of this part.

6.6.7.4 Configure_Request service

The Configure_Request service is used to transition the device application objects from the Abort, Execute or Idle state to the Configuring state.

When a Configure_Request is received, if the Configuration_Lock attribute is set, an “Object State Conflict (0Chex)” error shall be returned. The request contains three parameters; the password parameter contained in the Safety Supervisor, the Target device UNID, and the Originator UNID. These values are matched up against the Password, TUNID, and CFUNID in the device. The OUNID is needed to allow the device to determine if the source is a software tool or an ACR capable originator.

When the Configure Request command is accepted, the device shall only accept configuration messages to safety relevant objects over “this connection only”. Any attempts to write to safety-relevant objects over other connections shall be rejected.

If a configuration connection fails for any reason, another Configure_Request shall be received (establishing a new connection source) before configuration changes can be made (even after a power-cycle).

The structure of the Configure_Request is defined in Table 27.

Table 27 – Configure_Request message structure

Name	Data type	Description
Service Code	USINT	4F _{hex}
Password	ARRAY of 16 OCTETS	
TUNID	ARRAY of 10 OCTETS	Target Device UNID. Used to detect misrouted requests
OUNID	ARRAY of 10 OCTETS	Originator Device UNID. If the originator is a software tool, this parameter shall be set to all FF _{hex} . If the originator is a device with ACR, this parameter shall be the device's OUNID

If a TUNID, OUNID mismatch error occurs when processing a Safety Supervisor Configure Request, an Invalid Parameter (20_{hex}) error code shall be returned.

If a password mismatch occurs when processing a Safety Supervisor Configure Request, a Privilege Violation (0F_{hex}) error code shall be returned.

6.6.7.5 Validate_Configuration service

This service causes the device to validate a pending configuration by executing a CRC calculation over this data and comparing it to the CRC provided in the message. If the CRC does not yield the same result, an error response as defined below is returned. Otherwise, a “success” response is returned. See Figure 9 and Figure 10 for the processing details for this service. This command can only be received over the connection that put the device into the configuration mode. The device NV saves the final SCID in Attribute 26 of the Safety Supervisor.

The structure of the Validate_Configuration message is defined in Table 28.

Table 28 – Validate_Configuration message structure

Name	Data type	Description
Service Code	USINT	50 _{hex}
SCID	STRUCT of:	Safety Configuration ID
SCCRC	4 OCTETS	Proposed Safety Configuration CRC as calculated by the software. This is validated by the device
SCTS	6 OCTETS	Safety Configuration Time Stamp. This value marks the date and time of the configuration, which is used as a change detection mechanism

The structure of the Success Validate Reply is defined in Table 29.

Table 29 – Validate_Configuration success message structure

Name	Data type	Description
Service Code	USINT	50 _{hex}
SCID	STRUCT of:	Safety Configuration ID
SCCRC	4 OCTETS	Safety Configuration CRC as calculated by the device.
SCTS	6 OCTETS	Safety Configuration Time Stamp. This value is an echo of what was sent in the Validate request.

If the Safety Supervisor Validate_Configuration command does not succeed, the error response shall send a class defined "Validation Error" (D0_{hex}). The following error code defined in Table 30 is defined as a Validation Error in the Safety Supervisor, additional error codes are defined in Table 31.

Table 30 – Validate_Configuration error code

General error code	Additional error code	Error name	Description
D0 _{hex}	see Table 31	Validation Error	This error is a class specific error for the Safety Supervisor and is returned in response to a Validate_Configuration error

Table 31 – Validate_Configuration extended codes

Additional error codes	Reason code	Description
1	CRC mismatch	This is the default non-specific additional code, when no other additional information is available and the configuration CRC does not match the sent value
2	Invalid Configuration Parameter	This code can be used to indicate a parameter error if a device does specific validation of configuration data. NOTE This could happen even if the CRC's matched.
3	TUNID Not Set	This code can be used to indicate that the device is in the Waiting for TUNID state and needs to have its TUNID set

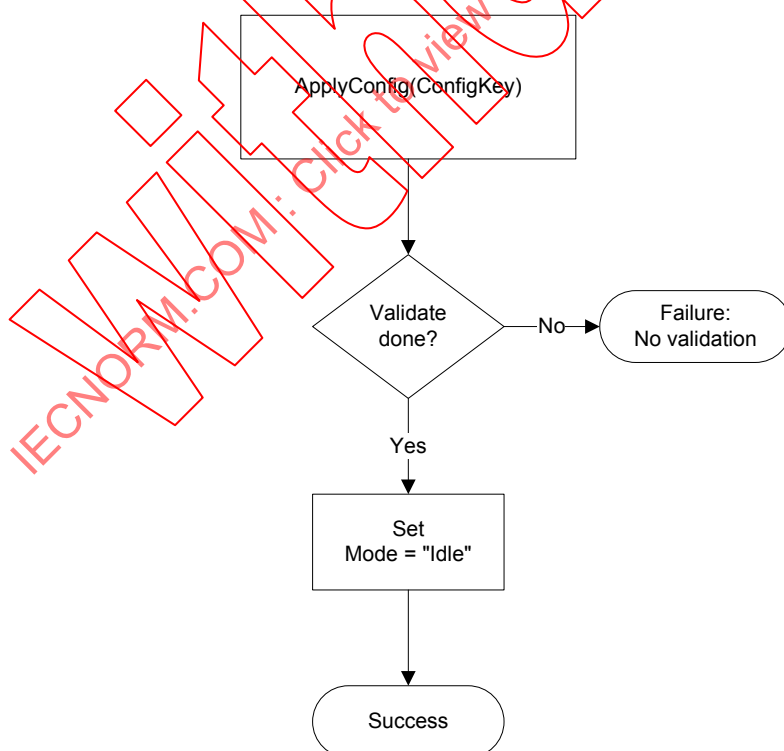


Figure 9 – Applying device configuration

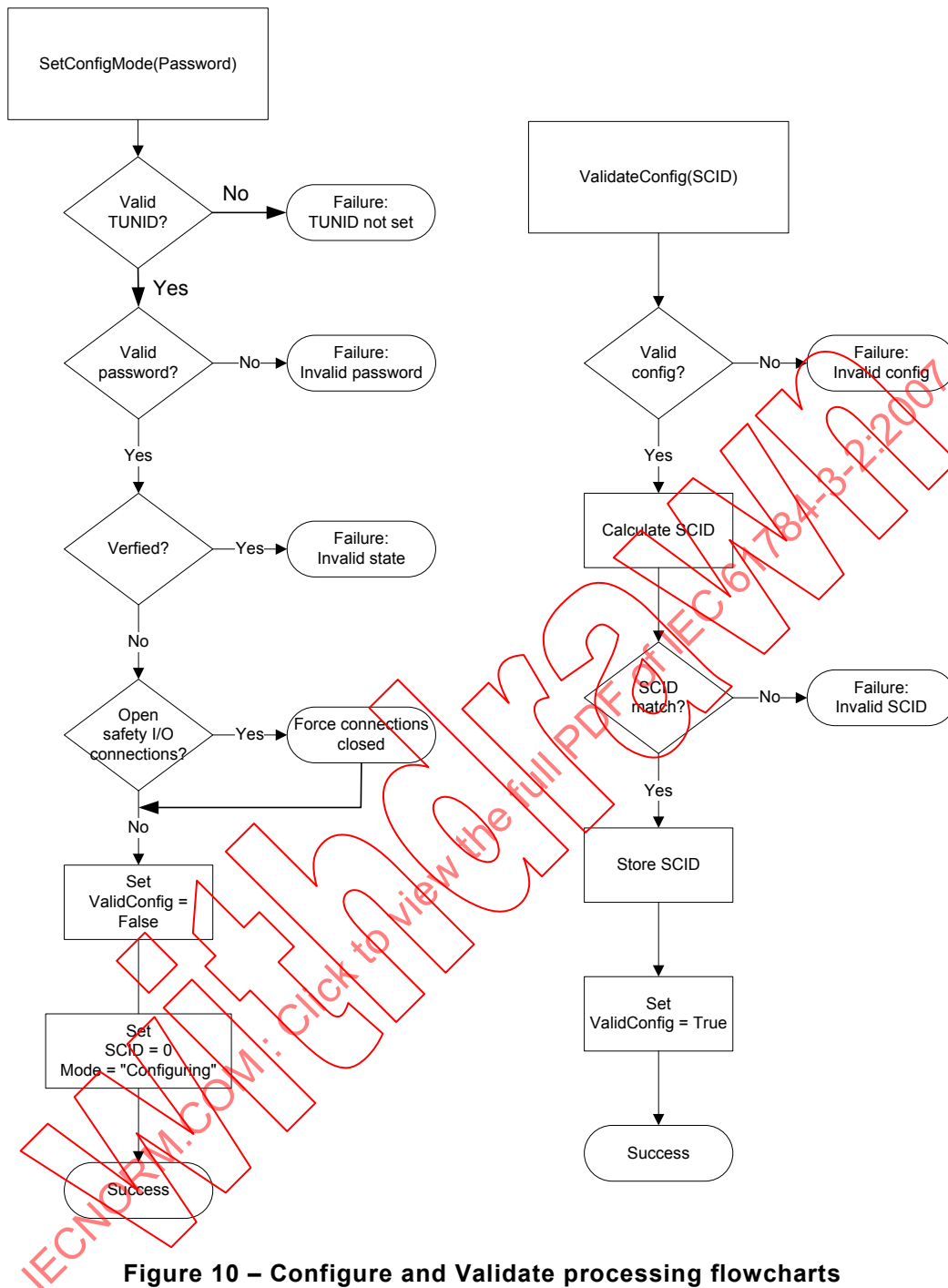


Figure 10 – Configure and Validate processing flowcharts

6.6.7.6 Set_Password service

The Set_Password service causes the device to set a Password in the Supervisor. This service takes 2 parameters. The first one is a password parameter representing the existing password, and the second is a parameter representing the new parameter. When setting a password for the first time, the Current Password parameter is set to the Out-of-box default (zero).

The structure of the Set_Password is defined in Table 32.

Table 32 – Set_Password message structure

Name	Data type	Description
Service Code	USINT	51 _{hex}
Current Password	16 OCTETS	
New Password	16 OCTETS	

If a password mismatch occurs on the current password when processing a Safety Supervisor Set Password command, a Privilege Violation (0F_{hex}) error code shall be returned.

A password which is smaller than the password parameter shall be right-aligned in little endian order to the least significant octet of the 16-octet password parameter. The unused upper octets of the password parameter shall be zero-filled.

6.6.7.7 Reset_Password service

The Reset_Password service is provided to allow passwords in a device to be reset. The exact contents and usage has been made vendor specific to protect password integrity. This service requires 1 parameter plus a vendor specific data field. The required parameter is a data size field to identify the size of the vendor specific data field. The content of vendor data is assumed to be unique to the device and the means by which this service is used shall be provided by each vendor.

The structure of the Reset_Password is defined in Table 33.

Table 33 – Reset_Password message structure

Name	Data type	Description
Service Code	USINT	55 _{hex}
Data Size	USINT	Number of vendor specific data octets
Vendor data	Data Size	Vendor data which contains data needed to cause the password to be reset

If a Data Size error occurs in the Safety Supervisor Reset_Password command, either error code 15_{hex} (too much data) or 13_{hex} (not enough data) shall be returned.

If a Vendor data mismatch occurs when processing a Safety Supervisor Reset_Password command, a Privilege Violation (0F_{hex}) error code shall be returned.

6.6.7.8 Configuration_Lock/Unlock service

The Configuration_Lock/Unlock service causes the device to modify the Configuration_Lock attribute in the Safety Supervisor. This service requires a value parameter (0 = unlocked, 1 = locked), and the password parameter. The password parameter is compared to the stored password and will only be executed if the password matches. If there is a password mismatch, a "Password Mismatch" error is returned.

The structure of the Configuration_Lock/Unlock is defined in Table 34.

Table 34 – Configuration_Lock/Unlock message structure

Name	Data type	Description
Service Code	USINT	52 _{hex}
Value	BOOL	0 = unlocked, 1 = locked
Password	16 OCTETS	
TUNID	10 OCTETS	Target Device UNID. Used to detect misrouted requests

If a TUNID mismatch occurs in the Safety Supervisor Configuration_Lock/Unlock command, the Invalid Parameter (020_{hex}) error code shall be returned.

If a password mismatch occurs when processing a Safety Supervisor Configuration_Lock/Unlock command, a Privilege Violation (0F_{hex}) error code shall be returned.

6.6.7.9 Mode Change service

The Mode Change service is used by devices which need to move between Idle and Executing states regardless of its connection status. This command requires proper authorization to execute, so the password is required to execute.

The structure of the Mode Change is defined in Table 35.

Table 35 – Mode_Change message structure

Name	Data type	Description
Service Code	USINT	53 _{hex}
Value	BOOL	0 = Idle, 1 = Executing
Password	16 OCTETS	

If a password mismatch occurs in the Safety Supervisor Mode Change command, an Privilege Violation (0F_{hex}) error code shall be returned.

6.6.7.10 Safety_Reset service

The Safety_Reset service is used to reset the device. The Safety_Reset service replaces the standard reset service defined so that it can be qualified by the password.

When a Type 1 or Type 2 Safety Reset is received, if the Configuration_Lock attribute is set, a “Object state conflict (0C_{hex})” error shall be returned.

The structure of the Safety_Reset is defined in Table 36.

Table 36 – Safety_Reset message structure

Name	Data type	Description
Service Code	USINT	54 _{hex}
Reset type	USINT	see Table 37
Password	16 OCTETS	
TUNID	10 OCTETS	Target Device UNID. Used to detect misrouted requests
Attribute_Bit_Map	USINT	Bit mapped parameter which indicates which attributes shall be preserved through the reset. This parameter is only included when the Reset Type is 2. see Table 38

If a TUNID mismatch occurs when sending the Safety Supervisor Safety Reset command, the Invalid Parameter (20_{hex}) error code shall be returned.

If a password mismatch occurs when processing a Safety Supervisor Safety Reset command, a Privilege Violation (0F_{hex}) error code shall be returned.

Table 37 – Safety Supervisor safety reset types

Value:	Type of reset
0	Emulate as closely as possible cycling power on the device
1	Return as closely as possible to the default configuration, and then emulate cycling power as closely as possible
2	Return as closely as possible to the out-of-box configuration except to preserve the parameters indicated by the Attribute Bit Map, and then emulate cycling power as closely as possible

Table 38 – Attribute bit map parameter

Bit	Attribute bit map assignments
0	When set, preserve Soft-set MACID
1	When set, preserve Soft-set Baud Rate
2	When set, preserve the TUNID
3	When set, preserve the Password
4	When set, preserve the CFUNID
5	When set, preserve the OPCUNID
6	Reserved, always 0
7	Use Extended Map (To be defined later)

Table 39 shows how a safety device shall process the various reset types based on the Lock/Unlock condition and safety connection status.

Table 39 – Reset processing rules for rest types

Reset Type	Possible Conditions when Reset is received			
	Safety Connection open / Locked	Safety Connection open / Unlocked	No Safety Connections / Locked	No Safety Connections / Unlocked
Type 0	Mode/State Error Response	Mode/State Error Response	Process and Respond	Process and Respond
Type 1	Mode/State Error Response	Mode/State Error Response	Mode/State Error Response	Process and Respond
Type 2	Mode/State Error Response	Mode/State Error Response	Mode/State Error Response	Process and Respond

6.6.7.11 Propose_TUNID service

The Propose_TUNID service establishes a proposed TUNID setting within the device. The value isn't stored in NV memory or in the TUNID attribute until the apply service is received.

The Propose_TUNID service shall only be accepted when the target device is in the Waiting_for_TUNID state.

Once the TUNID is successfully set, it can transition to Configuring.

The MACID or IP portion of the TUNID shall be matched up against the MACID of the DeviceNet object or IP attribute of the TCP/IP object.

The value in the MACID attribute of the DeviceNet object or IP Attribute of the TCP/IP object will be either what was read on the switches or what was set by a tool. In either case, the MACID/IP portion of the Proposed_TUNID shall match this.

The structure of the Propose_TUNID service is defined in Table 40.

Table 40 – Propose_TUNID service

Name	Data type	Description
Service Code	USINT	56hex
TUNID	10 OCTETS	Proposed Target Device UNID value

If an originator wants to cancel a propose/apply operation, sending a propose service with a TUNID of all FF_{hex} shall cancel the operation.

The following pseudo code defines the behavior of the Propose_TUNID service:

```

TUNIDProposedHandler()
{
  IF (Device_State=Waiting_for_TUNID)
    THEN IF (Propose_TUNID[TUNID] != all FFhex)
      THEN IF (Propose_TUNID[TUNID_MACID] == MACID attribute/IP
attribute)
        THEN
          TUNIDProposedAttr = Propose_TUNID[TUNID]
          NET LED Flash sequence is started
          Return Success Reply

```

```

ELSE

    Error Code "Invalid Parameter" is returned
    ELSE // cancel propose_apply operation, returns att. to all FFhex
        Stop NET LED Flash Sequence
        TUNIDProposedAttr = Propose_TUNID[TUNID]
        Return Success Reply
ELSE
    Error Code "Object State Conflict" is returned
}

```

6.6.7.12 Apply_TUNID service

The structure of the Apply_TUNID service is defined in Table 41 and a processing flow chart is defined in Figure 11.

Table 41 – Apply_TUNID service

Name	Data type	Description
Service Code	USINT	57 _{hex}
TUNID	10 OCTETS	Proposed Target Device UNID value

The Apply_TUNID service shall validate the TUNID against the proposed value, stops the LED flashing, saves the TUNID to nonvolatile storage, updates the TUNID attribute in the Supervisor, and sets the proposed_TUNID attribute to all FF_{hex}. The Supervisor shall then transition to Configuring.

The following pseudo code defines the behavior of the Apply_TUNID service:

```

Apply_TUNIDHandler()
{
    IF (Device_State=Waiting_for_TUNID)
        THEN IF {(Apply_TUNID[TUNID] != all FFhex)
            AND (TUNIDProposedAttr == Apply_TUNID[TUNID])
            THEN
                Stop Flash Sequence
                Target_UNID = TUNIDProposedVar[TUNID]
                //NV Memory operation
                // Set SNN attribute in DeviceNet or TCP/IP object, NV
                Memory operation
                Safety_Network_Number = TUNIDProposedVar[SNN]
                TUNIDProposedVar[TUNID] = TUNID_OUT_OF_BOX_DEFAULT
                Return Success Reply
                Device_State = CONFIGURING
            ELSE
                Error Code "Invalid Parameter" is returned
        ELSE
            Error Code "Object State Conflict" is returned
    }
}

```

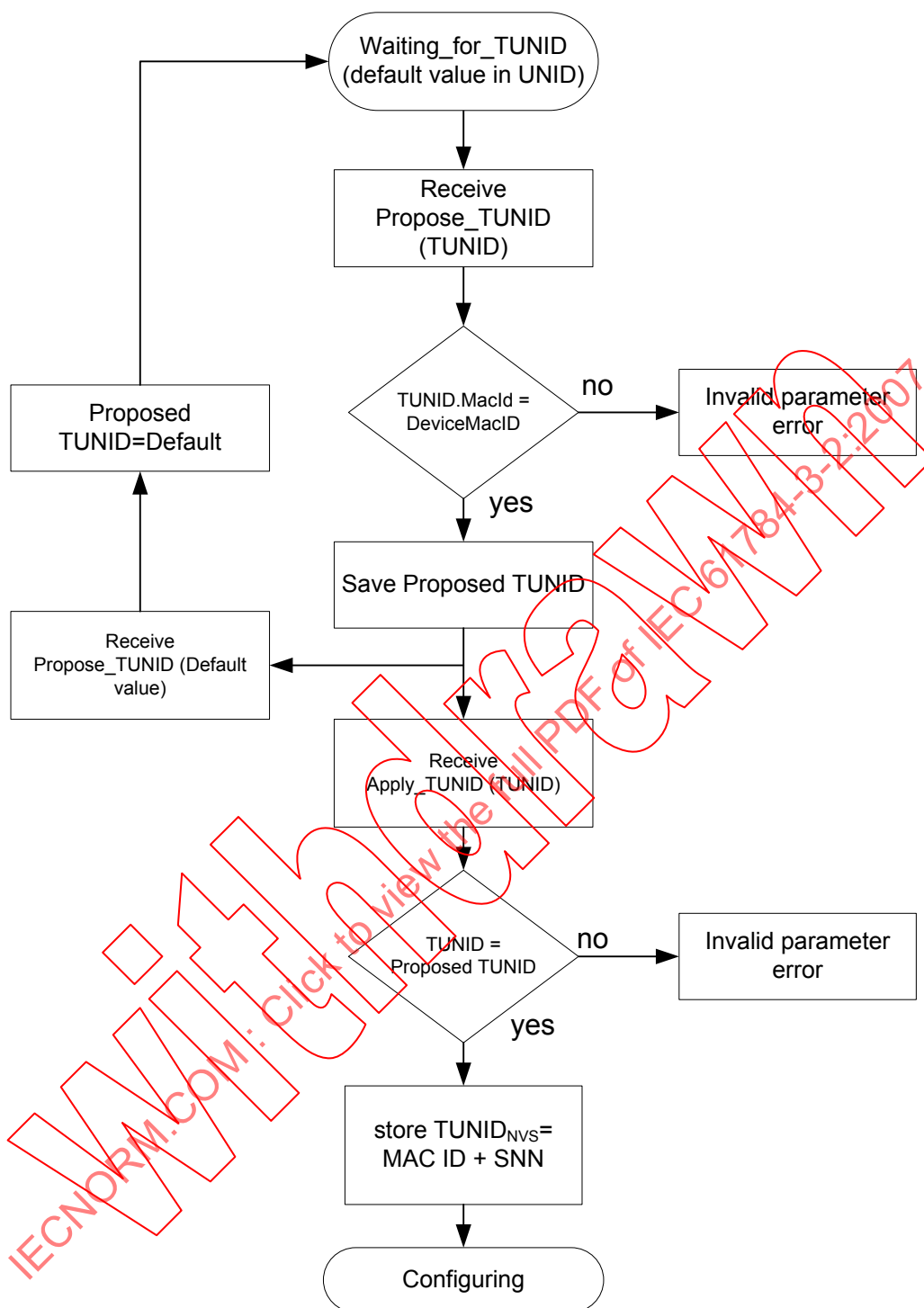


Figure 11 – UNID handling during “Waiting for TUNID”

6.6.8 Safety Supervisor behavior

6.6.8.1 Safety Supervisor object states

Figure 12 shows the state diagram defining the behavior of the Safety Supervisor object, further information on the Safety Supervisor events are given in Table 42.

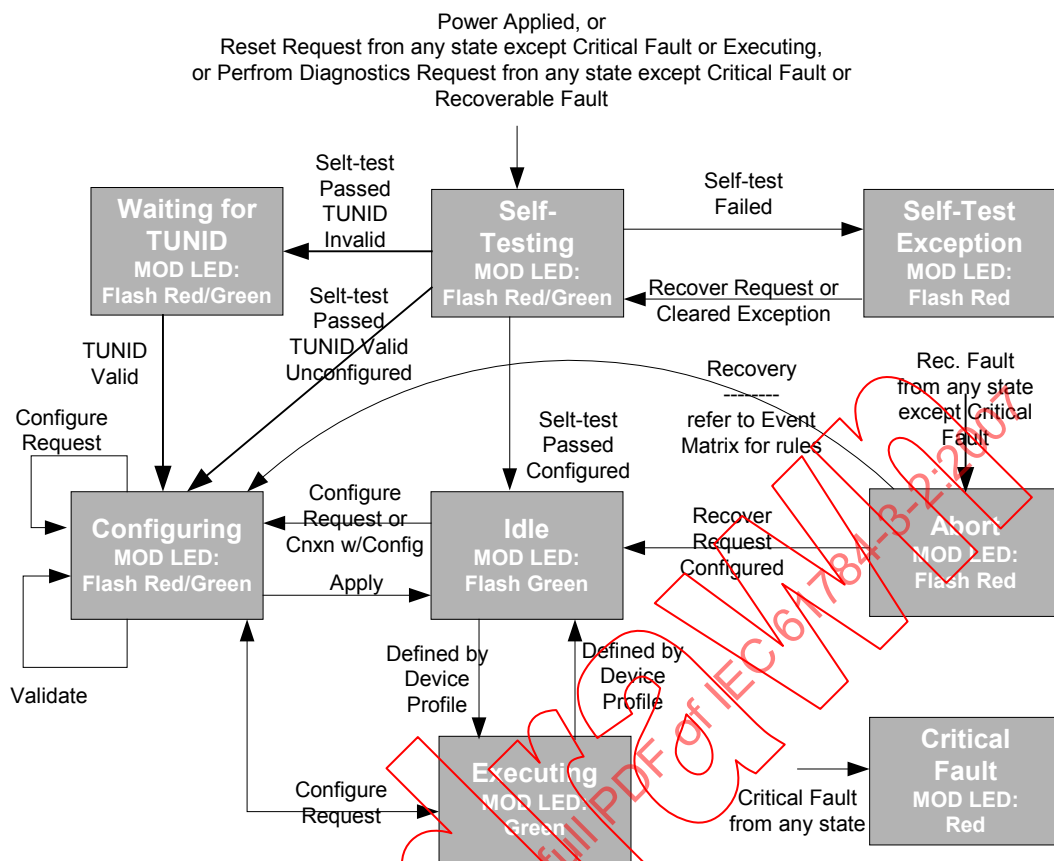


Figure 12 – Safety Supervisor state diagram

Table 42 – Safety Supervisor events

State	Description
Executing	Device is executing, (for example, is fully configured, has no errors, and has met vendor defined criteria). This state, and the transitions into or out of it, shall be further specified in an appropriate device profile specification
Self-Testing	Object instance exists and has been initialized; all attributes have appropriate initial values (as indicated herein and in applicable device profile specification). Exception Status bits have been reset. Device is performing device-specific and device type specific test to determine if it is qualified to execute its application process
Self-Test Exception	Object has detected an exception condition during self-testing. The details of the exception are stored in the appropriate attribute values of the Safety Supervisor object
Idle	Object and device have been initialized, successfully completed self-testing, and has a valid configuration. Further, the device is not executing the operational components of its device specific functions. Configuring and Idle are persistent states that are preserved through power cycles
Abort	Object instance is in a Abort state; this state can only be entered due to internally generated events. The conditions required for exit from a recoverable fault is defined in the Event Matrix
Waiting for TUNID	Device exits Self-testing and finds the NV saved TUNID is at the out-of-box default value. It remains in this state until the TUNID setting sequence (i.e. propose/apply in SNCT interface) is successfully completed. Only then can the device be configured
Configuring	Object and device have been initialized, successfully completed self-testing, but does NOT have a valid configuration. Configuring and Idle are sticky states that are preserved through power cycles
Critical Fault	The object (and device) is in a fault state from which there is no recovery. Object services cannot be processed. The conditions required for exit from a critical fault is outside the scope of this part

The Safety Supervisor Status attribute value indicates the overall state of the device. It is updated on appropriate state transitions within the Safety Supervisor object. Attribute values 1 through 8 represent valid states. A value of zero indicates that the Safety Supervisor state is unknown; conditions under which a zero value may occur are outside the scope of this part.

6.6.8.2 Safety Supervisor state event matrix

Table 43 defines the states and events for the Safety Supervisor.

Table 43 – State event matrix for Safety Supervisor

Event	State							
	Waiting for TUNID	Configuring (Safety State)	Idle (Safety State)	Self-Test (Safety State)	Self-Test Exception (Safety State)	Executing	Abort (Safety State)	Critical Fault (Safety State)
Power Applied			--	Default Entry Point performs its Self-test application process	--	--	--	Transition to SELF TEST
Self-Test Passed			Not Applicable	Transition to Last Saved State if not out-of-box Else, Wait for TUNID	Not Applicable	Not Applicable	Not Applicable	Not Applicable
Self-Test Failed			Not Applicable	Set appropriate Exception Status Bits and Transition to SELF TEST EXCEPTION	Not Applicable	Not Applicable	Not Applicable	Not Applicable
Propose TUNID	Validate Settings and Flash Net LED pattern	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b
Apply TUNID	Save TUNID to NV memory, Transition to Configuring	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b
Exception Condition Cleared			Not Applicable	Not Applicable	Set appropriate Exception Status Bits and Transition to SELF TESTING	Not Applicable	Not Applicable	Not Applicable
Critical Fault	Transition to	Transition to CRITICAL	Transition to CRITICAL	Transition to	Transition to	Transition to	Transition to	Ignore Event

Event	State							
	Waiting for TUNID	Configuring (Safety State)	Idle (Safety State)	Self-Test (Safety State)	Self-Test Exception (Safety State)	Executing	Abort (Safety State)	Critical Fault (Safety State)
	CRITICAL FAULT	FAULT	FAULT	CRITICAL FAULT	CRITICAL FAULT	CRITICAL FAULT	CRITICAL FAULT	
Configure Request	Error TNS ^c	Stay in state Validate and accept	If not Locked, Transition to Configuring	Error OSC ^b	Error OSC ^b	If not Locked, Drop Connections --- Apply Safety State Go to Configuring	If Configured: Transition to IDLE --- Unconfigured: Transition to Configuring --- If Switch/TUNID mismatch Error OSC ^b	Error OSC ^b
Validate Configuration	Error OSC ^b	Stay in state - Store Result	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b
Apply Request	Error OSC ^b	Transition to Idle if Valid	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b	Error OSC ^b
Form 1 SafetyOpen	Error TNS ^c	Apply configuration Transition to Executing if Config. Valid	Transition to Configuring, Proceed to IDLE/EXECUTING dependent on Profile	Error OSC ^b	Error OSC ^b	Drop Connections, Transition to Configuring, Proceed to IDLE EXECUTING dependent on Profile	Error OSC ^c	Error OSC ^b
Reset Request	Process type Transition to SELF TEST	Process type Transition to SELF TEST	If not Locked, Process type Transition to SELF TEST	Process type Transition to SELF TEST	Process type Transition to SELF TEST	If not Locked, Process type Transition to SELF TEST	If not Locked, Process type Transition to SELF TEST	Process type Transition to SELF TEST
Internal Abort Request	Transition to Abort	Transition to Abort	Transition to Abort	Error OSC ^b	Error OSC ^b	Transition to Abort	Error AIRS ^a	Error OSC ^b
Recover Request	Error OSC ^b	Error OSC ^b	Error OSC ^b	Restart SELF TESTING	Transition to SELF TESTING	Error OSC ^b	If Configured: Transition to IDLE --- Unconfigured:	Error OSC ^b

Event	State							
	Waiting for TUNID	Configuring (Safety State)	Idle (Safety State)	Self-Test (Safety State)	Self-Test Exception (Safety State)	Executing	Abort (Safety State)	Critical Fault (Safety State)
							Transition to Configuring --- If Switch/TUNID mismatch Error OSC ^b	
Perform Diagnostics Request	Error OSC ^b	Error OSC ^b	Transition to SELF TEST	Restart SELF TEST	Transition to SELF TEST	Error OSC ^b	Perform all device diagnostic tests	Error OSC ^b
Safety Connection Failed	Ignore Event	Ignore Event	Ignore Event	Ignore Event ^d	Ignore Event	Device Profile Defined	Ignore Event ^d	Ignore Event ^d
Safety I/O Connection established	Not Possible	Not Possible	Device Profile Defined	Ignore Event ^d	Not Possible	Device Profile Defined	Not Possible	Not Possible
Safety I/O Connection Deleted	Ignore Event ^d	Ignore Event ^d	Ignore Event ^d	Ignore Event ^d	Not Possible	Device Profile Defined	Ignore Event ^d	Error OSC ^b
Mode Change	Error OSC ^b	Error OSC ^b	Device Profile Defined	Error OSC ^b	Error OSC ^b	Device Profile Defined	Error OSC ^b	Error OSC ^b
Restore	Error OSC ^b	Restore Previous Configuration	Error OSC ^b	Error OSC ^b	Error OSC ^b	Ignore Event ^d	Error OSC ^b	Error OSC ^b
Lock/Unlock	Error OSC ^b	Error OSC ^b	if valid, Execute and reply	Error OSC ^b	Error OSC ^b	if valid, Execute and reply	if valid, Execute and reply	Error OSC ^b
LED Definitions for States								
Module Status LED	Flashing Red/Green	Flashing Red/Green	Flashing Green	Flashing Red/Green	Flashing Red	Solid Green	Flashing Red	Solid Red
^a Error AIRS = Error Response “Already in Requested Mode/State” (Code 0B_{hex}) ^b Error OSC = Error Response “Object State Conflict” (Code 0C_{hex}) when possible ^c Error TNS= Error Response “TUNID not Set”, 01_{hex} extended code or 80D_{hex} object specific extended code ^d Ignore Event:: defined as “event has no effect on the state”, no response necessary								

Any Safety Supervisor instance service may be requested internally by the device as specified by the manufacturer. Generally, these requests will be generated as the result of an event such as a diagnostic failure.

6.6.8.3 Effect of locking on device behavior

Figure 13 defines the effect that the “locked” condition has on the ability of a device to be configured; devices which are locked will reject all attempts to change any configuration data that affects the SCID.

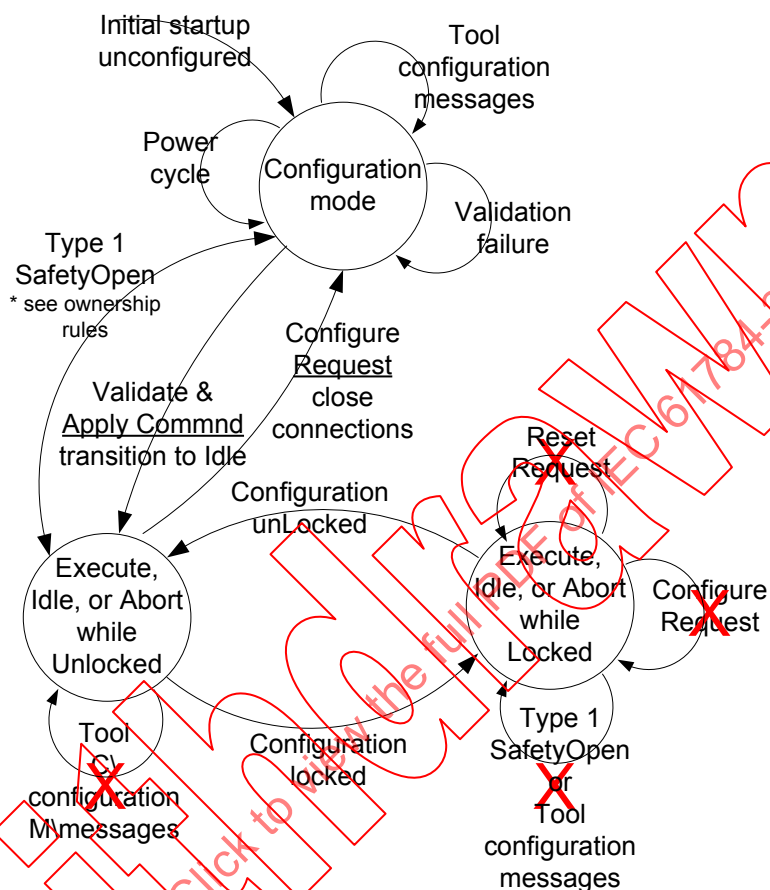


Figure 13 – Configuration, testing and locked relationships

Table 44 defines the effect that ownership has on a device’s ability to be configured. It shows that owned devices can only be Form 1 configured by the originator that has the owner UNID. A locked device can never be configured regardless of its ownership.

Table 44 – Configuration owner control vs. device state

Ownership rules	Configure mode	Execute/Idle Mode Unlocked	Execute/Idle Mode Locked
Config Owned (Config. OUNID = Owner UNID)	Form1 accepted by owner only	Form1 accepted by owner only	All Form 1 rejected
Config Un-owned (Config. OUNID = 0)	Form1 accepted from anyone	Form1 accepted from anyone	All Form 1 rejected
SW Tool Owned (Config. OUNID = all FF _{hex})	No Form 1 accepted	No Form 1 accepted	All Form 1 rejected

6.6.8.4 State mapping of Safety Supervisor object to Identity object

Table 45 defines the mapping of Safety Supervisor states to states defined for the Identity object. This state mapping defines the interface between these three objects. The identity object state attribute (if supported) shall follow this table.

Table 45 – State mapping of Safety Supervisor to Identity object

Safety Supervisor	Identity
Self-Testing	Device Self-Testing
Self-Test Exception	Major Unrecoverable
Idle	Standby
Configuring	Standby
Waiting for TUNID	Standby
Executing	Operational
Abort	Major Recoverable
Critical Fault	Major Unrecoverable

6.6.8.5 Safety Supervisor object event to Identity object event mapping

Table 46 describes the mapping of defined Safety Supervisor events to equivalent events defined for the Identity object. This event mapping defines the interface between these objects. This mapping is provided for reference purposes.

Table 46 – Safety Supervisor object event mapping

Safety Supervisor state events map to ->	Equivalent Identity event
Power Applied	Power Applied
Self-Test Failed	Failed Tests
Self-Test Passed	Passed Tests
Propose TUNID	No equivalent event
Apply TUNID	No equivalent event
Exception Condition Cleared	Fault Corrected
Critical Fault	Major Unrecoverable Fault
Configure Request	Deactivated
Validate Configuration	No equivalent event
Apply Configuration	Activated
Form 1 SafetyOpen	No equivalent event
Reset Request	Reset
Abort (Internal)	Major Recoverable Fault
Recover Request	Fault Corrected
Perform Diagnostic Request	No equivalent event
Safety Connection Failed	Deactivated
Safety Connection Established	Activated
Safety Connection Deleted	No equivalent event
Mode Change	No equivalent event
Restore	No equivalent event
Lock/Unlock	No equivalent event

6.6.8.6 Identity object event to Safety Supervisor event mapping

Table 47 describes the mapping of defined Identity objects event conditions to equivalent events defined for the Safety Supervisor object. This event mapping defines the equivalent events between these objects. This mapping is provided for reference purposes only; the Safety Supervisor object is the controlling object in safety devices and only events defined by the supervisor shall be implemented in safety devices.

Table 47 – Identity object event mapping

Identity events map to ->	Safety Supervisor events
Power applied	Power applied
Failed tests	Self-Test failed
Passed tests	Self-Test passed
Deactivated	Configure request
Activated	Apply configuration
Major recoverable fault	Abort (internal)
Major unrecoverable fault	Critical fault
Minor fault	No effect
Fault corrected	Exception condition cleared
Reset	Service not supported error

6.7 Safety Validator object

6.7.1 General

Class Code: 3Ahex

The Safety Validator object is designed for use in Safety devices on all FSCP 2/1 networks. The Safety Validator contains the information necessary to coordinate and maintain reliable safety connections between client and server safety-related applications. The primary role of the Safety Validator function is to act as a safety transport manager of multiple low-level CPF 2 connections that together form a complete safety connection.

The device containing the Safety Validator object can create several instances of the Safety Validator object for each safety-related application connection point (for example application data assembly) supported by the device.

The Safety Validator object supports two basic safety transport types: Point-to-point and Multipoint. Each safety transport type has a defined set of low-level CPF 2 connections that shall be allocated along with the Safety Validator object when a safety connection is instantiated by a SafetyOpen.

6.7.2 Class attributes

The class attributes for the safety supervisor are defined in Table 48.

Table 48 – Safety Validator class attributes

Attr ID	Need in implem.	Access rule	Name	Data type	Description of attribute	Semantics of value
1 thru 7	Optional	Get			See IEC 61158-5-2 and IEC 61158-6-2 for a description of the optional, reserved class attributes.	
8	Required	Get	Safety Connection Fault Count	UINT	Diagnostic Counter that is a running count of Safety Connection Faults	

Safety Connection Fault Count: this Safety Connection Fault Count attribute in the Safety Validator shall be a running count (with auto-rollover) of how many times any safety connection was faulted for any reason.

6.7.3 Instance attributes

6.7.3.1 Instance attributes overview

Most attribute values are derived, hard coded, or obtained from the dynamic connection establishment process. The only settable attribute value is the Max Data Age diagnostic. A complete list of the instance attributes is defined in Table 49.

Table 49 – Safety Validator instance attributes

Attr ID	Need in Implementation	Access rule	NV	Name	Data type	Description of attribute	Semantics of value
1	Required	Get		Safety Validator State	USINT	State of the safety connection	
2	Required	Get		Safety Validator Type	USINT	Safety Validator type used in this instance.	see 6.7.3.3
3	Optional	Get		Ping Interval EPI Multiplier	UINT	Number that defines the Ping_Count_Interval for a particular connection	16 to 1000
4	Optional	Get		Time Coord Msg Min Multiplier	STRUCT of:		
				Time Coord Msg Min Multiplier array size	USINT	Size of array equals Max Consumer number for multipoint producers and 1 for point-to-point and multipoint consumer	

Attr ID	Need in Implementation	Access rule	NV	Name	Data type	Description of attribute	Semantics of value
				Time Coord Msg Min Multiplier	Array of: UINT	Minimum number of 128 μ s increments it could take for a Time Coordination Message to traverse from the consumer to the producer	0 - 7813
5	Optional	Get		Network Time Expectation Multiplier	STRUCT of:		
				Network Time Expectation Multiplier array size	USINT	Size of array equals Max Consumer number for multipoint producers and 1 for point-to- point and multipoint consumer	
				Network Time Expectation Multiplier	Array of: UINT	Maximum number of 128 μ s increments that a consumer shall allow the age of the safety data to reach	0 - 45 313
6	Optional	Get		Timeout Multiplier	STRUCT of:		
				Timeout Multiplier array size	USINT	Size of array equals Max Consumer number for multipoint producers and 1 for point-to- point and multipoint consumer	
				Timeout Multiplier	Array of: USINT	Determines the number of messages that may be lost before declaring a connection error	1-4
7	Optional	Get		Max Consumer Number	USINT	Maximum number of consumers allowed for the connection	1 (Point-to- point), 2-15 (Multipoint)
8	Optional ^a	Get		Data Connection Instance	UINT	Connection Instance ID of the Safety Data Connection	

Attr ID	Need in Implementation	Access rule	NV	Name	Data type	Description of attribute	Semantics of value
9	Optional ^a	Get		Coordination Connection Instance	STRUCT of:	Connection Instance IDs of the Safety Coordination Connection	1-max number of connections in the device
				Coordination Instance array size	USINT	Size of array equals Max Consumer number for multipoint producers and 1 for point-to-point and multipoint consumer	
				Coordination Connection Instance	ARRAY of: UINT		
10	Optional ^a	Get		Correction Connection Instance	UINT	Connection Instance ID of the Time Correction Connection	0 (when not used) 1-max number of connections in the device
11	Optional ^a	Get		CCO Binding	UINT	Instance Id of a CCO which holds the connection Establishment settings for this connection. Only used by safety connection originators	0 (when not used or no binding present)
12	Required	Set		Max Data Age	UINT	Diagnostic which holds the largest Data Age detected in 128 µs increments. Attribute only used for Safety Consumers	
13	Required	Get		Application Data Path	EPATH	Points to the application data attached to this safety connection	
14	Optional	Get	V	Error Code	UINT	Reason for error within this instance	0 if no error, else implementation specific error code

^a It is optional that these parameters be made publicly visible. However, all Validator instances shall have internal parameters which represent these variables.

6.7.3.2 Safety Validator state

This attribute defines the current state of the Safety Validator object instance. Transitions in this state attribute reflect the current state of the object; the definitions are defined in Table 50. This attribute can be used to determine the condition of established safety connections.

Table 50 – Safety Validator state assignments

Value	State name	Description
0	Unallocated	In this state the Safety Validator object has either not been allocated to a connection or has been closed
1	Initializing	In this state the Safety Validator object is in the process of exchanging time coordination information across the connection. The "safety" connection is not yet fully established
2	Established	In this state the Safety Validator instance is fully established and producing/consuming safety data on behalf of the safety-related application
3	Connection_Failed	This state is entered when all connections associated with this Validator have failed for any reason. In producers, all multipoint consumer connections shall have failed. As long as any consumer is still operating, the producer would remain in the Established state
4-255	Reserved	

6.7.3.3 Safety Validator type

This attribute defines the type of Safety Validator transport that is being used, the definition is defined in Table 51. Targets shall derive this value from the Safety Connection parameters in the SafetyOpen.

Table 51 – Safety Validator type, bit field assignments

Bit 7	Bits 0 – 6
Producer(client) = 0/ Consumer(server) = 1	Safety Connection Type 0 = unallocated 1 = Point-to-point 2 = Multipoint 3-127 = Reserved

The transports for each Safety Validator Type are depicted in Figure 14. This attribute can be used to query a device to determine the type of connection that has been established.

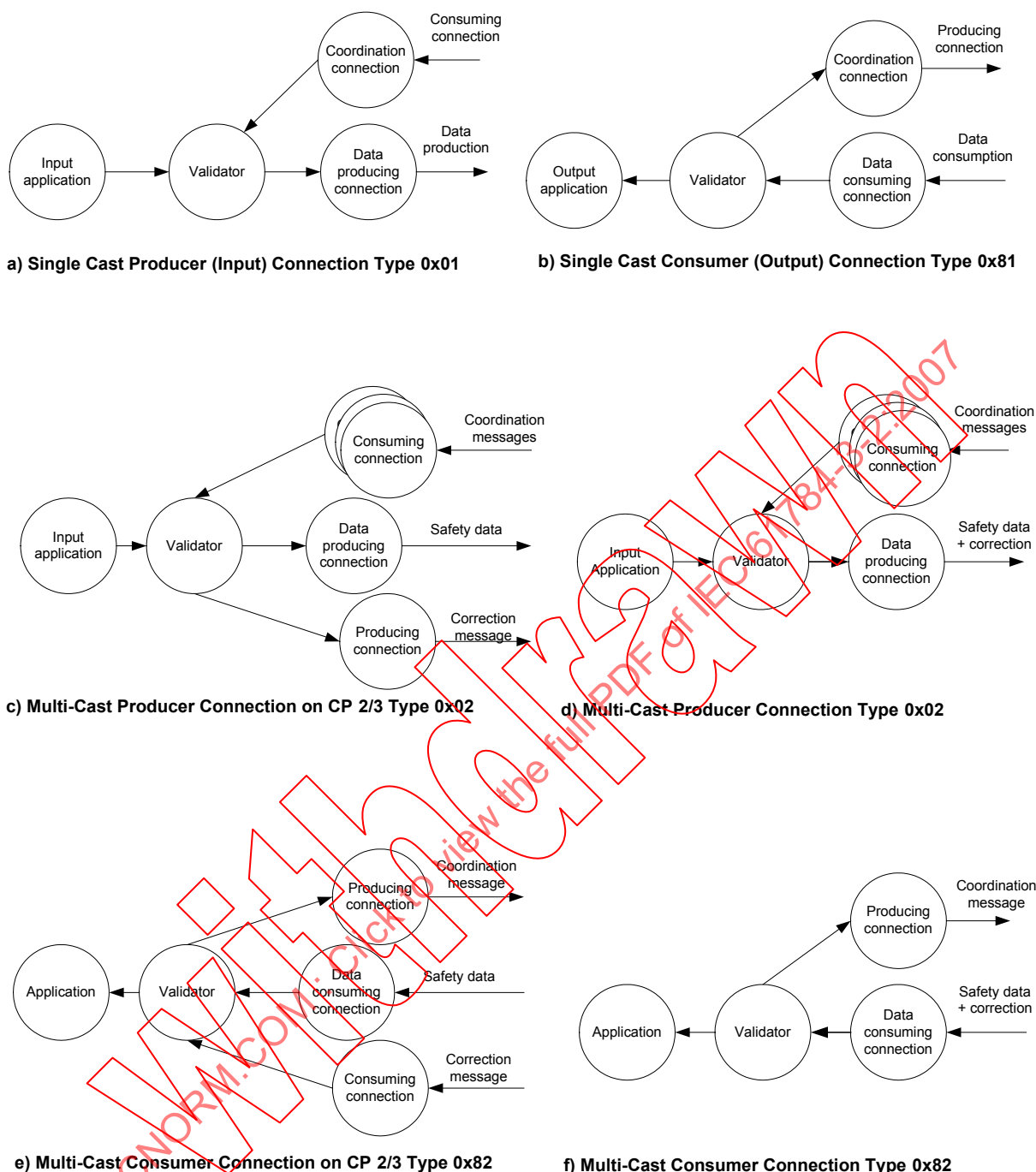


Figure 14 – Safety connection types

6.7.3.4 Ping interval EPI multiplier

This attribute is used as part of the Safety Validator function. This parameter is obtained from the Safety Segment Parameters in the SafetyOpen. The rules Multipoint producers shall follow for evaluating this parameter from multiple consumers are defined in Table 52.

6.7.3.5 Time Coord Msg min multiplier

This attribute is used as part of the Safety Validator function. This parameter is obtained from the Safety Segment Parameters in the SafetyOpen. The rules Multipoint producers shall follow for evaluating this parameter from multiple consumers are defined in Table 52.

Table 52 – Multipoint producer SafetyOpen parameter evaluation rules

Safety parameter	Producer evaluation rule	Error response (if appropriate)
Ping Interval EPI Multiplier	Shall match existing P.I.E.M. (1 st consumer sets it)	Invalid parameter
Time_Coord Msg Min Multiplier	Accept any (all can be different)	NA
Network Time Expectation Multiplier	Accept any (all can be different)	NA
Timeout Multiplier	Accept any (all can be different)	NA
Max Consumer Number	Shall match existing Max_Consumer_Number (1 st consumer sets it)	Invalid parameter

6.7.3.6 Network time expectation multiplier

This attribute defines the connection reaction time associated with this connection. This parameter is obtained from the Safety Segment Parameters in the SafetyOpen. The rules Multipoint producers shall follow for evaluating this parameter from multiple consumers are defined in Table 52.

6.7.3.7 Timeout multiplier

This attribute defines the tolerance the Safety Validator will have for lost messages before faulting a connection and taking a safety action. This parameter is obtained from the Safety Segment Parameters in the SafetyOpen. The rules Multipoint producers shall follow for evaluating this parameter from multiple consumers is defined in Table 52.

6.7.3.8 Max consumer number

This attribute defines the maximum number of consumers that are allowed to connect to a multipoint producer (Type 82). Both Multipoint Consumers and Producers have to know this value to function properly. This parameter is obtained from the Safety Segment Parameters in the SafetyOpen. The rules Multipoint producers shall follow for evaluating this parameter from multiple consumers is defined in Table 52.

6.7.3.9 Data connection instance

This attribute identifies the instance of the connection object responsible for the Safety Data connection. This attribute is assigned as part of the connection establishment process.

6.7.3.10 Coordination connection instance

This attribute identifies the instance of the connection object responsible for the Time Coordination connection. This attribute is assigned as part of the connection establishment process.

6.7.3.11 Correction connection instance

This attribute identifies the instance of the connection object responsible for the Time Correction connection if used. This attribute is assigned as part of the connection establishment process.

6.7.3.12 CCO binding

This attribute identifies the Connection Configuration instance that was used to set up this safety connection. This binding is required to notify the CCO function when the connection is closed so that it can attempt to re-establish the connection. This attribute is only used in connection originators. This attribute may either be assigned as part of the connection establishment process or hard coded at design time.

6.7.3.13 Max data age

The data age of received packet data is defined as the difference between the consumer's 128 μ s clock and received Time Stamp at the time the packet is received. The Max Data Age attribute is the maximum value this age has reached while the connection is/was established. The value can be used by diagnostic tools to fine tune the Network Time Expectation Multiplier.

The Max Data Age attribute shall always reflect the largest value the connection's data age reaches without exceeding the expectation multiplier.

If a safety connection is ever faulted and closed, the Max Data Age attribute shall be re-initialized to 0 when the connection is re-established.

6.7.4 Class services

The class services supported by the Safety Validator are defined in Table 53.

Table 53 – Safety Validator class services

Service code	Need in implementation	Name	Description of service
0E _{hex}	Required	Get_Attribute_Single	Used to read a Safety Validator object Class attribute value
4B _{hex}	Optional	Reset all error counters	Used to reset class attribute 8 in all instances

6.7.5 Instance services

6.7.5.1 Instance services overview

The instance services supported by the Safety Validator are defined in Table 54.

Table 54 – Safety Validator instance services

Service code	Need in implementation	Name	Description of service
01 _{hex}	Optional	Get_Attribute_All	Returns the values of all instance attributes
0E _{hex}	Required	Get_Attribute_Single	Used to read a Safety Validator object attribute value
10 _{hex}	Required	Set_Attribute_Single	Used to reset the Max Data Age Attribute
4B _{hex}	Optional	Reset error code	Used to reset instance attribute 14

6.7.5.2 Get_Attributes_All Response

The response data for the Get_Attributes_All service is defined in Table 55.

Table 55 – Safety Validator Get_Attributes_All service data

Attribute ID	Octet number	Name	Data type
1	0	Safety Validator State	USINT
2	1	Safety Validator Type	USINT
3	2,3	Ping Interval EPI Multiplier	UINT
4	variable	Time Coord Msg Min Multiplier	STRUCT
5	variable	Network Time Expectation	STRUCT
6	variable	Timeout Multiplier	STRUCT
7	variable	Max Consumer Number	USINT
8	variable	Data connection Instance	UINT = 0 if not supported
9	variable	Coordination Connection Instance	UINT = 0 if not supported
10	variable	Correction Connection Instance	STRUCT
11	variable	CCO Binding	UINT = 0 if not supported
12	variable	Max Data Age	UINT
13	variable	Application Path	EPATH
14		Error Code	UINT

6.7.6 Object behavior

6.7.6.1 State transition diagram

The behavior of the Safety Validator object is illustrated in Figure 15 and Table 56 below.

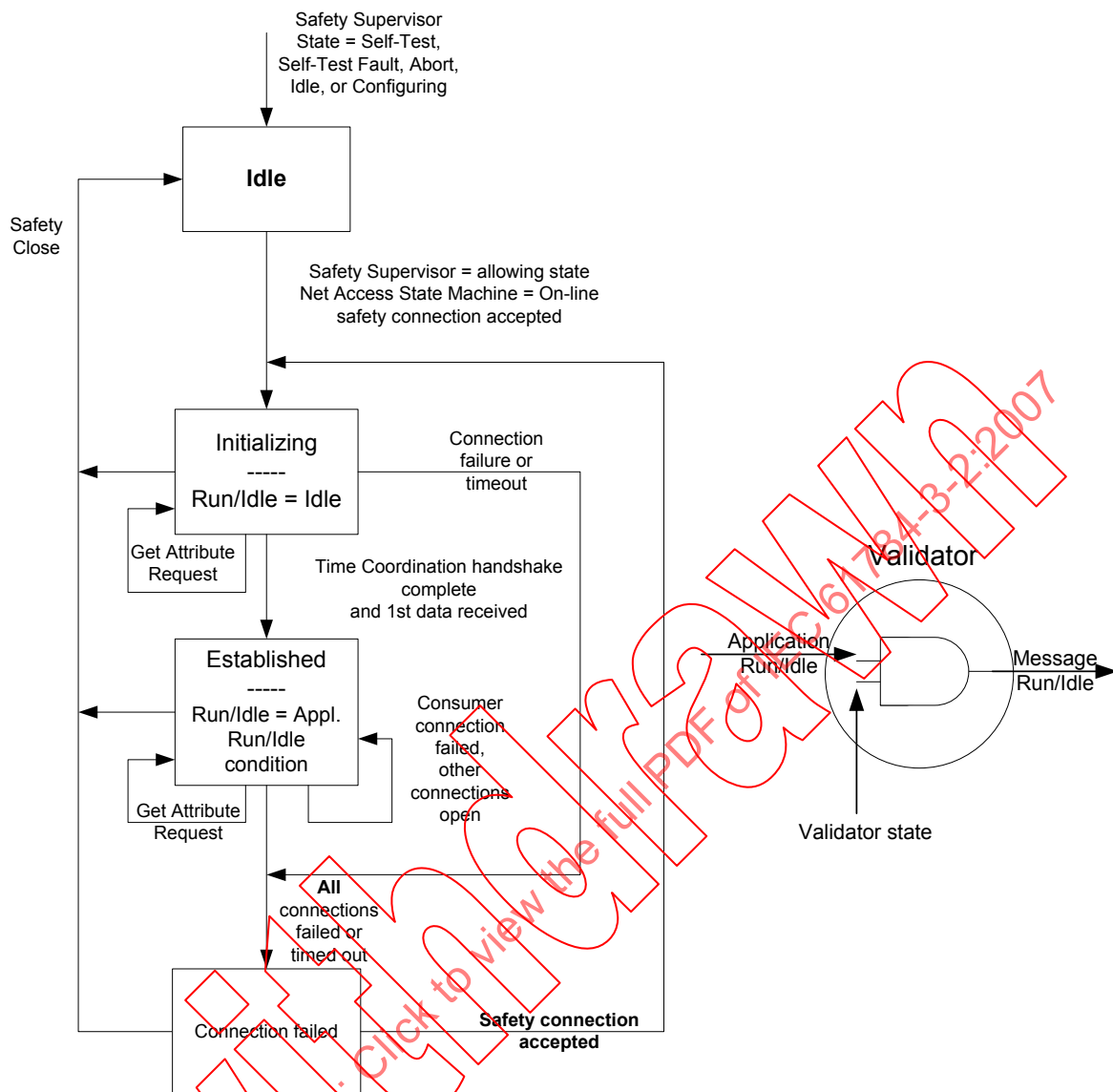


Figure 15 – Safety Validator state transition diagram

6.7.6.2 IDLE

The Idle state is initial state entered after Power-up/Self-test. This state is the normal state when no connections are open or failed.

6.7.6.3 Initializing

For multipoint producers, this state exists for each consumer connection handled by the producer. For point-to-point producers and all consumers, this state relates to the entire Safety Validator connection.

6.7.6.4 Established

For multipoint producers, this state exists for each consumer connection handled by the producer. For point-to-point producers and all consumers, this state relates to the entire Safety Validator connection.

6.7.6.5 Connection_Failed

This state is entered when all connections associated with the Safety Validator have failed. It can remain in this state until either a SafetyClose is received, or the connection is re-established from a matching originator (i.e. OUNID and Application Path match a connection resource).

6.7.6.6 State event matrix

Table 56 is the state event matrix that defines the Safety Validator behavior. Table 57 maps the Safety Validator states to the Safety Supervisor states.

Table 56 – Safety Validator state event matrix

Event	State			
	Idle	Initializing	Executing	Connection failed
Power applied and Self-Test complete	Initial state	--	--	--
Safety Supervisor = allowing state, Net Access State Machine = On-line, New safety connection requested	Validate Request and transition to Initializing	If multipoint producer, Initialize safety connection	If Multipoint producer, Initialize safety connection	Match OUNID and Application Path, If match, Transition to Initializing
Time coordination handshake complete	--	Transition to Executing	If multipoint producer, New Consumer On-line	--
Consumer fails in Multipoint producer while other consumers still on-line	--	Flash NET status LED Red	Flash NET status LED Red	--
All connections failed	--	Transition to Connection Failed	Transition to Connection Failed	Flash NET status LED Red
Matching consumer connection request (i.e. OUNID and Application path match a failed consumer connection)	--	Transition Consumer to Initializing	Transition Consumer to Initializing	Transition to Initializing
SafetyClose received	Success Reply, stay in Idle	Success Reply, Transition to Idle	Success Reply, Transition to Idle	Success Reply, Transition to Idle

Table 57 – State mapping between Safety Supervisor and Safety Validator objects

Safety Supervisor	Safety Validator
Self-Testing	Idle
Self-Test Exception	Idle
Waiting for TUNID	Idle
Idle	Idle ^a , Initializing Executing, or Connection_Failed
Configuring	Idle
Executing	Idle ^a , Initializing or Executing, or Connection_Failed
Abort	Idle
Critical Fault	Idle
^a A Safety Validator shall only accept connections when the Safety Supervisor is in the state that can accept connections which is device dependent (i.e. Executing only, or Execute/Idle). A Safety Validator instance could be Idle while another Safety Validator is still active.	

6.8 Connection Configuration Object

6.8.1 General

This sub-clause defines the extensions to the Connection Configuration Object, Class F3_{hex} for FSCP 2/1. See IEC 61158-4-2 for the complete definition.

Class Code: F3hex

6.8.2 Class attribute extensions

The class attributes for use by FSCP 2/1 devices are defined in Table 58.

Table 58 – Connection configuration object class attribute extensions

Attribute ID	Need in implem.	Access rule	Name	Data type	Description of attribute
9	Conditional ^a	Set	Edit Signature	UDINT	Created and used by configuration software to detect modifications to the instance attribute values
^a This attribute is only required in devices not using the Safety SNCT interface.					

6.8.3 Instance attributes, additions and extensions.

Table 59 defines the instance attributes and changes for use by FSCP 2/1 devices.

Table 59 – Connection Configuration Object instance attribute additions/extensions

Attr ID	Need in implem.	Access rule	NV	Name	Data type	Description of attribute	Semantics of value
9	Required	Set	NV	Data Map	STRUCT of		
				format_number	UINT	This number determines the map format used.	2-99 Reserved 100 – 199 Vendor Specific All other values are Reserved.
				Data_Map data_size	UINT	Size, in octets, of Map information	
				Data_Map_data	Array of octet	Data Map format data based on Format number	
10	Optional	Set	NV	Config #2 Data2	STRUCT of		
				config_data_size	UINT	Length of config_data in octets	
				config_data	Array of octet	Config # 2 data	
11	Optional	Set	NV	Proxy Device ID	STRUCT of		
				vendor_id	UINT	Vendor ID	
				product_type	UINT	Device Type	
				product_code	UINT	Product Code	
				major_rev	USINT	Major revision	
				minor_rev	USINT	Minor revision	
12	Conditional ^a	Set	NV	Connection Disable	BOOL		0 – connection is enabled. 1 – connection is disabled. When an Open service is received this value shall be set to 0. When a Close or Stop service is received this value shall be set to 1.
13	Conditional ^a	Set	NV	Safety Parameters	STRUCT of		
				reserved	USINT	Reserved	Shall be zero
				reserved	USINT	Reserved	Shall be zero
				Configuration CRC	UDINT	CRC-S4 over the target device configuration data in Attribute 7 and 10	This value calculated by the configuration software on the data in Attr. 7 and 10 after the configuration is applied
				Configuration Signature	DATE AND TIME		
				Time Correction EPI	UDINT		
				Time Correction Connection Parameters	WORD		
				Target UNID	10 octets		UNID of target device

Attr ID	Need in implem.	Access rule	NV	Name	Data type	Description of attribute	Semantics of value
				Reserved space for Originator UNID	10 octets	All zeros	Allows FW to easily stuff OUNID into safety segment
				Ping Int EPI Mult	UINT		
				Time Coord Msg Min Mult	UINT		
				Net Time Exp Mult	UINT		
				Timeout Multiplier	USINT		
				Max Consumer Number	USINT		
14	Conditional ^a	Get	NV	Connection Parameter CRC	UDINT	CRC-S4 over the connection parameters used in a SafetyOpen	This value is calculated by this device only once when the configuration is applied. It shall be readable by configuration software
15	Conditional ^a	Set	NV	Configuration Instance	UINT	Which object has configuration for this node	= 0 when no object assigned
16	Conditional ^a	Set	NV	Id Allocation	BOOL	When cleared, the originator will allocate the network Ids for its produced connections	0: allocate the Id for produced connections (default) 1; ask target for produced Ids
17	Conditional ^a	Get		Target Connection Serial Number	UINT	The instance # of the Target's Safety Validator. Target Connection Serial Number returned during connection establishment. Read by diagnostic tools	0 when connection is not established
^a This attribute is required for a device which supports safety connections; otherwise this attribute is optional.							

6.8.4 Instance attribute semantics extensions or restrictions for safety

6.8.4.1 Connection flags – (Attribute 2)

Connection Flags are defined in Table 60.

Table 60 – Connection flag bit definitions

Bits	Definition for safety devices
Bit 0:	Connection Direction Always 0 = Originator of connection
Bit 1-3:	O-to-T Real time transfer format Always 101 = Safety Format
Bit 4-6:	T-to-O Real time transfer format Always 101 = Safety Format
Bit 7-15:	Not used. Always 0
Value:	005A _{hex} = Originator 005B _{hex} = Target

6.8.4.2 CS Data Index Number – (Attribute 4)

CS Data Index is not used, shall be FFFF FFFF_{hex}.

6.8.4.3 Connection Timeout Multiplier – (part of Attribute 5)

Connection Timeout Multiplier for safety is used by the producer and consumer to timeout any of the 3 standard connections. It is used by the consumer of the connection to determine if the connection should timeout.

The timeout value for the connection is defined as:

$$\text{Connection RPI} \times (\text{CTM} + 1) \times 4$$

Range: 0 to 7

where

0 = (0x4);

1 = (0x8);

2 = (0x12);

3 = (0x16);

4 = (0x20);

5 = (0x24);

6 = (0x28);

7 = (0x32);

Value: Always 1 (x8)

6.8.4.4 Transport Class and Trigger – (part of Attribute 5)

The value of the Transport Class and Trigger for safety is 40_{hex} for an Input Connection or C0_{hex} for an Output Connection.

6.8.4.5 O-to-T RPI – (part of Attribute 5)

If direction is 0 (Client / T-to-O), then this is the Time Coordination RPI and the T-to-O RPI is the Data Production RPI.

If direction is 1 (Server / O-to-T), then this is the Data Production RPI and the T-to-O RPI is the Time Coordination RPI.

The Time Coordination RPI is defined as:

Range: 100 to 100 000 000 us
 Ping Interval EPI Multiplier * Data Production RPI
 Data Production RPI
 Range: 100 to 1 000 000 us

6.8.4.6 O-to-T connection parameters – (part of Attribute 5)

O-toT connection parameters are defined in Table 61.

Table 61 – O-to-T connection parameters

Bits	Parameter	Value
0-8	Connection size (in octets)	If the direction is 1 (Server / O-to-T), it is the size of the Safety Message: If Data Size is 1-2: Data Size + 6 If Data Size is 3-250: (Data Size × 2) + 8 If the direction (Client / T-to-O) it is the size of the Time Coordination Message (6 octets) Range: 7-510
9	Fixed/Variable	Always 0 (Fixed)
10-11	Priority	Always 01 (High)
12	Reserved	Always 0
13-14	Connection Type	Always 10 (Point-to-point)
15	Redundant Owner	Always 0 (safety doesn't support redundancy)

6.8.4.7 T-to-O RPI – (part of Attribute 5)

If direction is 0 (Client / T-to-O), then this is the Data Production RPI and the O-to-T RPI is the Time Coordination RPI.

If direction is 1 (Server / O-to-T), then this is the Time Coordination RPI and the O-to-T RPI is the Data Production RPI.

Value: see O-to-T RPI.

6.8.4.8 T-to-O Connection parameters – (part of Attribute 5)

The T-to-O connection parameters are defined in Table 62.

Table 62 – T-to-O connection parameters

Bits	Parameter	Value
0-8	Connection size (in octets)	If the direction is 0 (Client / T-to-O), it is the size of the Safety Message. For Point-to-point, see O-to-T Connection Parameters. For Multipoint with a Data Size of 1-2: Data Size + 12 For Multipoint with a Data Size of 3-248: (Data Size × 2) + 14 Multipoint calculations shall include the Time Correction Section. This is required for all multipoint connections to ensure the proper calculation of the CPCRC in the SafetyOpen. The value is fixed up by the Target Device. If the direction is 1 (Server / O-to-T), it is the size of the Time Coordination Message (6 octets). Range: 7-510
9	Fixed/Variable	Always 0 (Fixed)
10-11	Priority	Always 01 (High)
12	Reserved	Always 0
13-14	Connection Type	01 (Multipoint) 10 (Point-to-point) Default: 10 (Point-to-point)
15	Redundant Owner	Always 0 (safety doesn't support redundancy)

6.8.4.9 Connection Path – (Attribute 6)

6.8.4.9.1 Connection Path size

The Connection Path Size is the number of 16-bit words stored in Connection Path. This path shall be adjusted to remove the size of the bridge path elements when the CPCRC is calculated (i.e. the size used in the calculation shall be the value the target will see after the bridge(s) have stripped away their segments and appropriately adjusted this size).

6.8.4.9.2 Connection Path

The connection paths associated with the connection in the following order: (1) Bridge Path, (2) Configuration Path, (3) Consumption Path, and (4) Production Path.

The bridge path (1) shall be separated by the originator from the application path 2, 3, and 4 at the time the SafetyOpen is built and when the CPCRC is calculated.

6.8.4.9.3 Bridge Path

Valid path constructed for the route between the originator and target.

6.8.4.9.4 Configuration Path

Valid path if configuration data is present in Attributes 7 or 8, or Null path otherwise. The path in all cases comes from the Target Applet (i.e. EDS File).

6.8.4.9.5 Target Consumption Path

Valid path if direction 1 (Server / O-to-T) or Null Path otherwise. The path in all cases comes from the Target Applet (i.e. EDS File).

6.8.4.9.6 Target Production Path

Valid path if direction is 0 (Client / T-to-O) or Null Path otherwise. The path in all cases comes from the Target Applet (i.e. EDS File).

6.8.4.10 Config # 1 Data Size – (part of Attribute 7)

Usage is optional for Safety, zero if not used

6.8.4.11 Config # 1 Data – (part of Attribute 7)

Usage is optional for Safety and shall be concatenated with the contents of Config#2 data

6.8.4.12 Config # 2 Data Size – (part of Attribute 10)

Number of octets stored in Config #2 Data (below)

Range: 1 to < Max Allowable Configuration Size>

6.8.4.13 Config # 2 Data – (part of Attribute 10)

Used to store the target device's configuration data

6.8.4.14 Data Map – (Attribute 9)

6.8.4.14.1 Map format allocation

Table 63 defines a set of map formats that are currently defined in the standard range. The vendor specific range is still available to be used for special map formats.

Table 63 – Data map formats

Format number	Format name	Data Map data Size (in octets)	Data Map data	Comment
0	Open scanner format	4 octets	see Table 64	Simple format which maps data in word granularity to input and output image tables
1	Octet-Index Scanner format	4 octets	see Table 65	Simple format which maps data in octet granularity to input and output image tables
2-99	Reserved			
100-199	Vendor specific			
200 - 65535	Reserved			

6.8.4.14.2 Format 0 usage for safety scanners

Data map format 0 contains the Map Data structure using the format described in Map Format Type, defined in Table 64.

Table 64 – Data map format 0

Name	Data type	Description	Value
Originator to Target data image offset	UINT	The offset, in 16 bit units, in the originator to target image table	If the direction is 0 (Client / T-to-O): Always 0. If the direction is 1 (Server / O-to-T): Word offset into the Safety Output Data Table.
Target to Originator data image offset	UINT	The offset, in 16 bit units, in the target to originator image table	If the direction is 0 (Client / T-to-O): Word offset into the Safety Input Data Table. If the direction is 1 (Server / O-to-T): Always 0.

6.8.4.14.3 Format 1 usage for safety scanners

Data map format 1 contains the Map Data structure using the format described in Map Format Type, defined in Table 65.

Table 65 – Data map format 1

Name	Data type	Description	Value
Originator to Target data image offset	UINT	The offset, in 8 bit units, in the originator to target image table	If the direction is 0 (Client / T-to-O): Always 0. If the direction is 1 (Server / O-to-T): octet offset into the Safety Output Data Table.
Target to Originator data image offset	UINT	The offset, in 8 bit units, in the target to originator image table	If the direction is 0 (Client / T-to-O): octet offset into the Safety Input Data Table. If the direction is 1 (Server / O-to-T): Always 0.

6.8.4.15 Proxy Device ID

Proxy Device ID is never used.

6.8.4.16 Connection Disable (Attribute 12)

When the Connection Enable/Disable attribute in the CCO object is TRUE, the connection shall remain disabled through power cycles. When the Connection Disable attribute in the CCO object is FALSE, the connection shall remain enabled through power cycles.

6.8.5 Special Safety Related Parameters – (Attribute 13)**6.8.5.1 Ping Interval EPI Multiplier**

This attribute is used as part of the safety Validator function. This parameter is part of the Safety Segment Parameters in the SafetyOpen. Ping_Interval_EPI_Multiplier is the number that defines the Ping_Count_Interval for the safety connection.

Devices implementing the safety extensions to the CCO object shall perform a range check on the Ping Interval EPI Multiplier during a Validate_Configuration request.

Ping Interval EPI Multiplier usage is defined as follows:

Range: $[\text{Max\_Consumer\_Number} \times \text{Timeout\_Multiplier} + 15]$ to 1000.

Default:

If the direction is 0 (Client / T-to-O): 100

If the direction is 1 (Server / O-to-T): 19

6.8.5.2 Time Coord Msg Min Multiplier

This attribute is used as part of the Safety Validator function. This parameter is obtained from the safety segment parameters in the SafetyOpen. Time_Coord_Msg_Min_Multiplier is the minimum number of 128 μ s increments it could take for a Time Coordination Message to traverse from the consumer to the producer. The default is 0.

Devices implementing the safety extensions to the CCO object shall perform a range check on the Time_Coord_Msg_Min_Multiplier during a Validate_Configuration request.

Time Coord Msg Min Multiplier usage is defined as follows:

Range: 0 to 7 813, which equates to 0 s to 1 s.

Default: 2 for non-bridge; 3 or more when bridging is added.

6.8.5.3 Network Time Expectation Multiplier

This attribute is used as part of the Safety Validator function. This parameter is part of the safety segment parameters in the SafetyOpen. Network_Time_Expectation_Multiplier is the maximum number of 128 μ Sec increments that a consumer shall allow the age of the safety data to reach.

Device implementing the safety extensions to the CCO object shall perform a range check on the Network_Time_Expectation_Multiplier during a Validate_Configuration request.

Network Time Expectation Multiplier usage is defined as follows:

Range: 0 to 45 313, which equates to 1 s to 5,8 s.

Value: $[(\text{Timeout\_Multiplier} + 2) \times \text{Data\_Production\_RPI}] / 128$

6.8.5.4 Timeout Multiplier

This attribute is used as part of the Safety Validator function. This parameter is part of the safety segment parameters in the SafetyOpen.

Device implementing the safety extensions to the CCO object shall perform a range check on the Timeout_Multiplier during a Validate_Configuration request.

Timeout Multiplier usage is defined as follows:

Range: 1 to 4

Default: 2

6.8.5.5 Max Consumer Number

This attribute defines the maximum number of consumers that are allowed to connect to a multipoint producer (Type 2). Both Multipoint Consumers and Producers needs to know this

value to function properly. This attribute is used as part of the safety Validator function. This parameter is part of the Safety Segment Parameters in the SafetyOpen.

Device implementing the safety extensions to the CCO object shall perform a range check on the Max Consumer Number during a Validate_Configuration request.

Max Consumer Number usage is defined as follows:

Range: 1 to 15
 Default: 15
 Value: Comes from the Target Applet (i.e. EDS File)

6.8.5.6 Target Connection UNID

This attribute is the UNID of the target safety device for this connection. This parameter is part of the Safety Segment Parameters in the SafetyOpen and is checked by the target device to confirm the request didn't get misrouted. The only way to confirm this value is correct is through user testing of the connection.

Target UNID format on CP 2/3 is defined as follows:

OCTET 0-5: Safety Network Number
 OCTET 6: CP 2/3 MACID
 OCTET 7-9: Always 0 for CP 2/3

6.8.5.7 Configuration CRC (SCCRC)

The SCCRC serves as a signature for the device configuration and can be used to confirm the integrity of the configuration over time. The configuration software shall calculate the SCCRC over the configuration data in the exact order it appears in the SafetyOpen message. This same calculation is done by target devices and compared to this value.

The SCCRC shall be calculated over the configuration data by the device whenever a device configuration is validated (i.e. Validate_Configuration Request to the safety supervisor or type 1 safety connection).

The SCCRC shall be saved to NV storage along with the other NV attributes whenever a configuration is applied (i.e. apply request to the safety supervisor).

The user shall be instructed in the safety manual to test safety connection configurations after they are applied in an originator to confirm the target connection is operating as intended.

The target device's SCCRC usage is shown in Table 66.

Table 66 – Target device's SCCRC values

Safety connection type	SCCRC value
Type 1	SCCRC identifying configuration data in Config #2 Data
Type 2a	SCCRC identifying the expected target device configuration
Type 2b	0

6.8.5.8 Configuration Signature (Time Stamp)

The target device's SCTS values are shown in Table 67.

Table 67 – Target device's SCTS values

Safety connection type	SCTS value
Type 1	SCTS identifying configuration data in Config #1 Data
Type 2a	SCTS identifying the expected target device configuration
Type 2b	0

6.8.5.9 Time Correction EPI

This parameter defines the EPI for the Time Correction Message used in a multipoint connection.

The Value for a multipoint connection is :

Ping Interval EPI Multiplier × Data Production RPI;

Otherwise 0.

6.8.5.10 Time Correction Connection Parameters

Defines the Connection Parameters used in the Time Correction Message used in a multipoint connection.

The value is 0000_{hex} for a non Multipoint connections or 2406_{hex} for a Multipoint connection, defined as shown in Table 68.

Table 68 – Time correction connection parameters for multipoint connection

Bits	Parameter	Value
0-8	Connection size (in octets)	Size of the Time Correction Message which is 6 octets
9	Fixed/Variable	Always 0 (Fixed)
10-11	Priority	Always 01 (High)
12	Reserved	Always 0
13-14	Connection Type	01 (Multipoint)
15	Redundant Owner	Always 0 (safety doesn't support redundancy)

6.8.5.11 Connection Parameter CRC (CPCRC) – (Attribute 14)

The Connection Parameter CRC attribute shall be calculated by this originator device whenever the originator's configuration is validated (i.e. Validate_Configuration request to the safety supervisor).

The CPCRC shall be saved to NV storage along with the other NV attributes whenever a configuration is applied (i.e. apply request to the safety supervisor).

The user should be required to test this connection after it is applied to confirm the connection configuration. From that point on, the CPCRC serves as a signature for that connection configuration and can be used to confirm the integrity of the configuration over time. The calculation itself is done over the connection parameters in SafetyOpen in the exact order

they appear in the message. This same calculation is done by target devices and compared to this value, so the exact order shall be followed.

6.8.5.12 Configuration Instance – (Attribute 15)

This attribute provides the linkage to the instance of the object assigned to this connection. The attribute may be used when the target's configuration data cannot be sent via the SafetyOpen. A value of 0 (default) for this attribute indicates that no object is assigned.

6.8.5.13 Id Allocation

This Boolean attribute is used to give the user some flexibility in how originators allocate Ids on connection establishment. This is particularly important for FSCP 2/1 on CP 2/3. The default behavior is for the originator to provide Ids for the connections in which it is the producer. If the user sets this attribute, the originator will instead ask the Target to provide an Id for these produced connections. A user may want to do this when the originator has a large number of connections and does not have enough to do the application. This attribute only applies to non-multipoint productions (multipoint productions shall always be allocated by the producer). This function is considered an advanced function and great care should be taken when presenting this feature to a user. The user shall be advised it should only be used in limited application scenarios where connection repeatability can be guaranteed.

6.8.6 Object-specific services

The Connection Configuration Object provides the following Object Specific Services, defined in Table 69.

Table 69 – Connection Configuration Object-specific services

Service	Need in implementation		Service name	Description of service
	Class	Instance		
4B _{hex}	Conditional ¹	N/A	Kick-Timer	Kicks Edit Watchdog Timer
4C _{hex}	Optional	Conditional ^b	Open	Opens connections
4D _{hex}	Optional	Conditional ^b	Close	Closes connections
4E _{hex}	Optional	Conditional ^b	Stop	Stops connections
4F _{hex}	Conditional ^a	N/A	Change Start	Manages session editing
50 _{hex}	Required	N/A	Get Status	Get status for multiple connections
51 _{hex}	Conditional ^a	N/A	Change Complete	Completes session editing
52 _{hex}	Conditional ^a	N/A	Audit Changes	Audits pending changes
^a Safety Devices shall not implement these services since the software configuration tool will coordinate configuration through the Safety Supervisor. ^b FRS357 Safety Devices shall support the Open, Close and Stop services. These services shall only be accepted when the device is unlocked. If the device is locked, it shall respond with a "Device state conflict" error.				

6.8.7 Common service extensions for safety

6.8.7.1 Get Attribute All (Service Code 01_{hex})

The added attributes to the end of the Get_Attribute_All response is defined in Table 70. All other parameters of this service are as defined in the standard CCO object in IEC 61158-5-2 and IEC 61158-6-2.

Table 70 – Get_Attributes_All Response service data (added attributes)

Name	Data type	Description
Safety Parameters	STRUCT of	
	USINT	Pad, set to zero
	USINT	Pad, set to zero
	UDINT	Configuration CRC (SCCRC)
	DATE AND TIME	Configuration Time Stamp (SCTS)
	UDINT	Time Correction EPI
	WORD	Time Correction Connection Parameters
	10:octets	Target UNID
	10:octets	Pad, set to zero
	UINT	Ping Int EPI Mult
	UINT	Time Coord Msg Min Mult
	UINT	Net Time Exp Mult
	USINT	Timeout Multiplier
	USINT	Max Consumer Number
Connection Parameter CRC	UDINT	CRC-32 over the connection parameters used in a SafetyOpen
Configuration Instance	UINT	
ID Allocation Flag	BOOL	
Target Connection Serial Number	UINT	Safety Validator Instance returned from Target during connection establishment

6.8.7.2 Set_Attribute_All (Service Code 02_{hex})

The added attributes to the end of the Set_Attribute_All response shall be extended as defined in Table 71. All other parameters of this service are as defined in the standard CCO object in IEC 61158-5-2 and IEC 61158-6-2.

Table 71 – Set_Attributes_All Request service data (added attributes)

Name	Data type	Description
Safety parameters	STRUCT of	
	USINT	Pad, set to zero
	USINT	Pad, set to zero
	UDINT	Configuration CRC (SCCRC) (value set = 0, device does this calc)
	DATE AND TIME	Configuration Time Stamp (SCTS) (if applicable, else = 0)
	UDINT	Time Correction EPI
	WORD	Time Correction Connection parameters
	10octets	Target UNID
	10octets	Pad, set to zero
	UINT	Ping Int EPI Mult
	UINT	Time Coord Msg Min Mult
	UINT	Net Time Exp Mult
	USINT	Timeout Multiplier
	USINT	Max Consumer Number

Name	Data type	Description
Configuration instance	UINT	
ID Allocation flag	BOOL	

6.8.7.3 Restore (Service Code 15_{hex})

Safety devices shall not implement the Restore service at either class or instance level since the configuration tool will coordinate configuration through the Safety Supervisor.

6.8.8 Object behavior

The Connection Configuration object behavior for Safety Devices shall be controlled by the Safety Supervisor and follow the behavior defined in Table 72, Figure 16, and Figure 17.

Table 72 – State Mapping between Safety Supervisor and the CCO objects

Safety Supervisor	Connection Configuration
Self-Testing	Edits not allowed
Self-Test exception	Edits not allowed
Idle	Edits not allowed
Configuring	Edits allowed
Executing	Edits not allowed
Abort	Edits not allowed
Critical fault	Edits not allowed

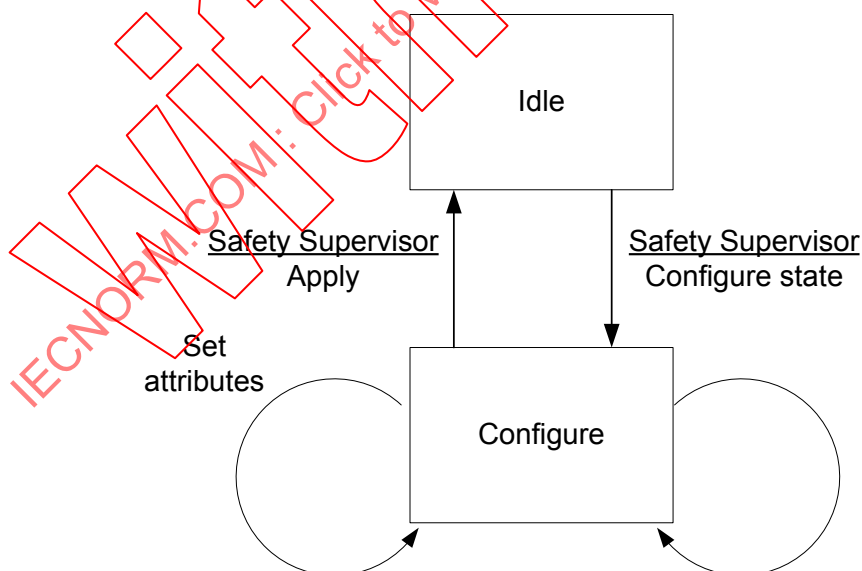


Figure 16 – Connection Configuration Object state diagram

The Data flow for the Connection Configuration object while in the Configure State is defined in Figure 17.

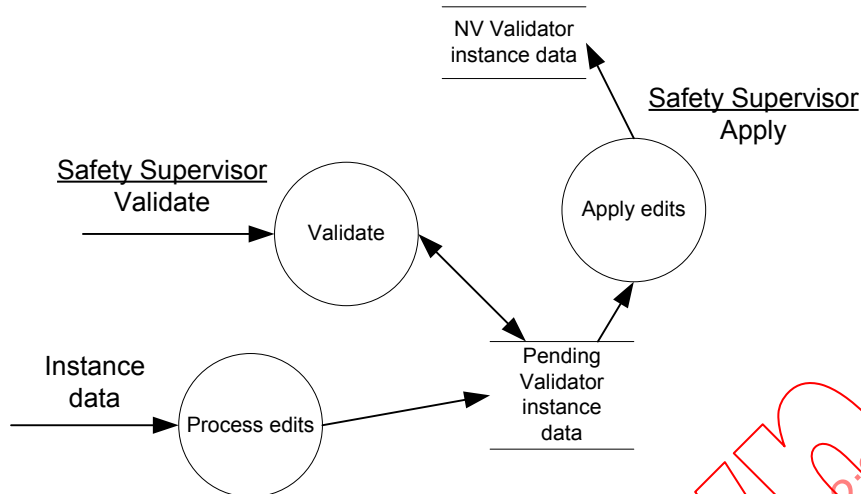


Figure 17 – Connection Configuration Object data flow

7 Safety communication layer protocol

7.1 Safety PDU format

7.1.1 Safety PDU encoding

7.1.1.1 General encoding rules

The Safety Protocol defines five section encoding formats that are used for all safety messaging. The five types of section encoding formats are:

- 1 or 2 octet data section (see 7.1.1.4),
- 3 to 250 octet data section (see 7.1.1.5),
- Time stamp section (see 7.1.1.6),
- Time coordination section (see 7.1.1.7),
- Time correction section (see 7.1.1.7).

These section encodings are used in the six types of connection formats. The six types of connection formats are:

- 1 or 2 octet point-to-point (see 7.1.1.11),
- 1 or 2 octet Multipoint Format 1 (see 7.1.1.12),
- 1 or 2 octet Multipoint Format 2 (see 7.1.1.13),
- 3 to 248 octet Multipoint Format 1 (see 7.1.1.14),
- 3 to 248 octet Multipoint Format 1 (see 7.1.1.15),
- 3 to 248 octet Multipoint Format 2 (see 7.1.1.16).

The six types of connection formats are sent via three message types. The message types are:

- Data Message,
- Time Coordination Message,
- Time Correction Message.

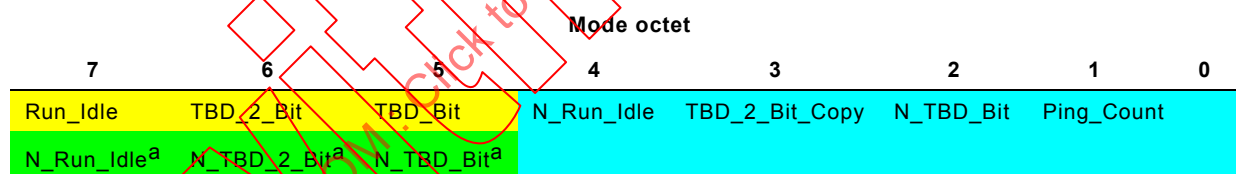
Table 73 defines which Sections are used within the various message types for the six different connection format types:

Table 73 – Connection sections and PDU formats

Safety Connection format			Message types					
			Data Message				Time Coordination message	Time Correction message
Data size	Safety Connection type	Network hop type	1 or 2 octet data section	3 to 250 octet data section	Time Stamp section	Time Correction section	Time Coordination section	Time Correction section
1 or 2 octets	Point-to-point	All	X		X		X	
	Multipoint	CP 2/3	X		X		X	X
		CP 2/2	X		X	X	X	
3 to 248 Octets	Point-to-point	All		X	X		X	
	Multipoint	CP 2/3		X	X		X	X
		CP 2/2		X	X	X	X	

7.1.1.2 Mode octet

The Mode octet is described outside of the section definitions because it is in both the data sections and the time stamp section. The layout of the Mode octets is defined in Figure 18. Bits 0 thru 4 of the mode octet are part of the time stamp section, while bits 5 thru 7 are part of the data sections. The variables used in the mode octet, brief descriptions and references to complete descriptions are given in Table 74.



Key

- = Covered by actual data CRC
- = Covered by the complemented data CRC
- = Covered by the Time Stamp Section CRC

^a The values are not sent in the data packet, there are derived by the consumer.

Figure 18 – Format of the mode octet

Table 74 – Mode octet variables

Name	Description	Reference
Run_Idle	Run_Idle shall be used to indicated the usability of the data as determined by the Producer Safety-related application	7.6.1.1
N_Run_Idle	Complement of the Run_Idle	7.6.1.2
TBD_2_Bit	Reserved for future use	7.6.1.3
TBD_2_Bit Copy	Copy of the TBD_2_Bit	7.6.1.4
TBD_Bit	Reserved for future use The TBD_Bit Shall be set to 0 by the Safety Validator	7.6.1.6
N_TBD_Bit	Complement of the TBD_Bit	7.6.1.7
Ping_Count	The ping count is used for determining when a consumer shall send a time coordination message	7.6.1.5

7.1.1.3 Mode octet CRC processing

The Mode octet processing within the CRCs shall be done as follows.

- CRC Type Mode Octet logic prior to inclusion within the CRC;
- Actual Data CRC Mode Octet AND $E0_{hex}$;
- Complement Data CRC (Mode Octet XOR FF_{hex}) AND $E0_{hex}$;
- Time Stamp Section CRC Mode Octet AND $1F_{hex}$.

7.1.1.4 1 or 2 octet data section

The 1 or 2 octet data section shall consist of the actual data octet or octets, the mode octet, the actual CRC, and the complement CRC. The layout of the 1 or 2 octet data section is defined in Figure 19.

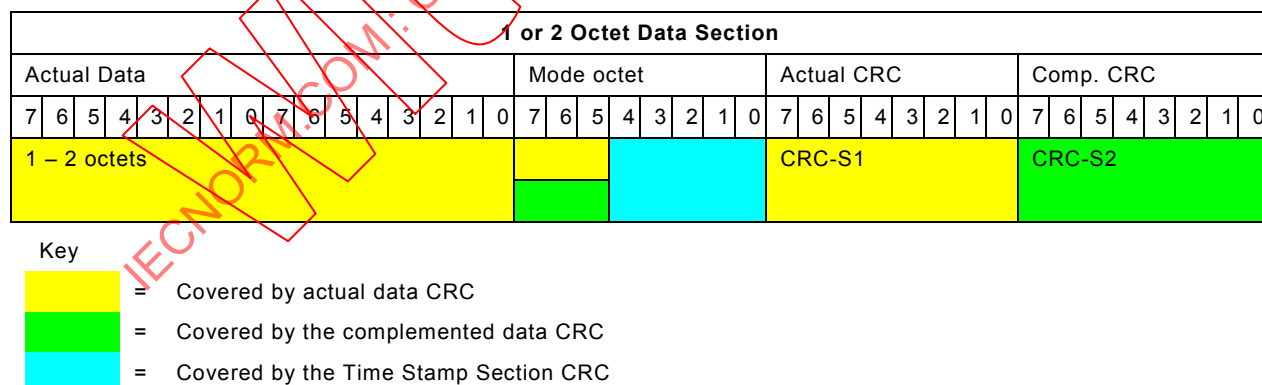


Figure 19 – 1 or 2 octet data section

The 1 or 2 Octet Actual Data CRC calculation shall include:

- Producer Identifier (PID) (by CRC-S1);
- The (Mode Octet) AND ($E0_{hex}$) (by CRC-S1);
- Actual Data octet(s) (by CRC-S1).

The 1 or 2 Octet Complement Data CRC calculation shall include:

- Producer Identifier (PID) (by CRC-S1);

- The (Mode Octet XOR FF_{hex}) AND (E0_{hex}) (by CRC-S2);
- Actual Data octets XOR FF_{hex} (by CRC-S2).

NOTE The PID uses CRC-S1 for both the actual and the complement CRC calculation.

7.1.1.5 3 to 250 octet data section

The 3 to 250 octet data section shall consist of the actual data octets, the mode octet, the actual CRC, the complement data octets, and the complement CRC. The layout of the 3 or 250 Octet Data section is defined in Figure 20.

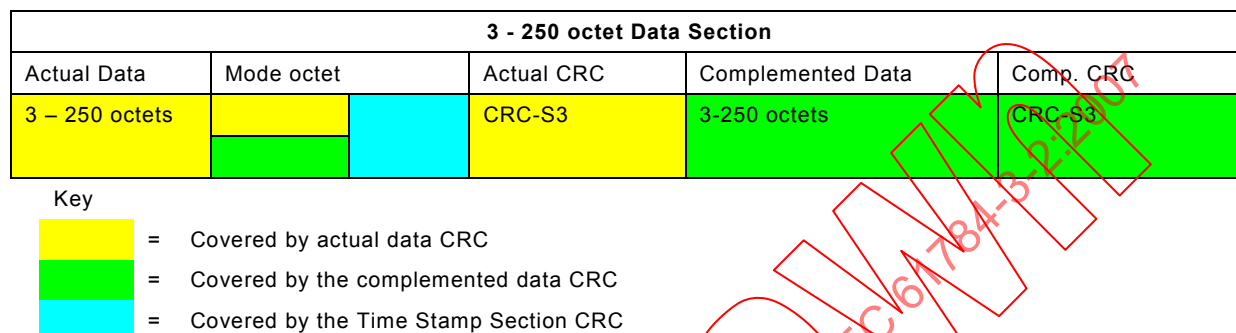


Figure 20 – 3 to 250 octet data section format

The 3-to-250 octet actual data CRC calculation shall include:

- Producer Identifier (PID) (by CRC-S3);
- The (Mode Octet) AND (E0_{hex}) (by CRC-S3);
- Actual Data octet(s) (by CRC-S3).

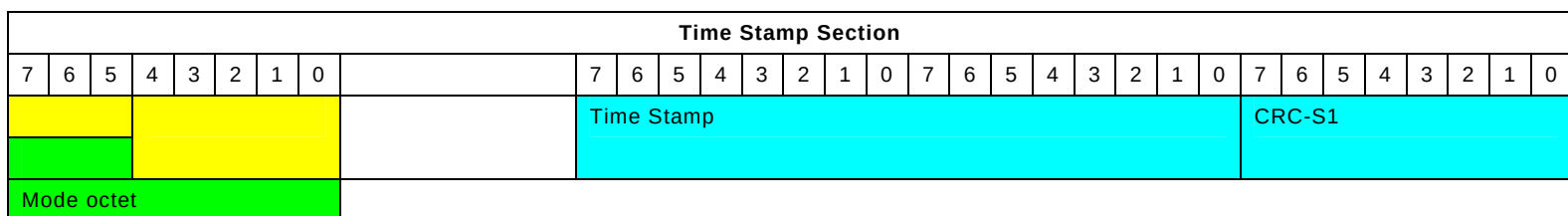
The 3-to-250 octet complement data CRC calculation shall include:

- Producer Identifier (PID) (by CRC-S3);
- The (Mode Octet XOR FF_{hex}) AND (E0_{hex}) (by CRC-S3);
- Actual Data octets XOR FF_{hex} (by CRC-S3).

7.1.1.6 Time Stamp section

The Time Stamp section is composed of the mode octet, the Time_Stamp, and the time stamp CRC. The layout of the Time Stamp section is defined in Figure 21.

Descriptions of the Time Stamp section variables is defined in Table 75.



Key

- = Covered by actual data CRC
- = Covered by the complemented data CRC
- = Covered by the Time Stamp Section CRC

Figure 21 – Time Stamp section format

Table 75 – Time Stamp variables

Name	Description	Reference
Data_Time_Stamp	Point-to-point -- Data_Time_Stamp is the time the data was sampled for production relative to the consumers clock. Multipoint -- Data_Time_Stamp is the time the data was sampled for production relative to the producers clock	7.6.2

The Time Stamp CRC calculation shall include:

- Producer Identifier octet (PID) (by CRC-S1);
- The (Mode Octet) AND (1F_{hex}) (by CRC-S1);
- Time Stamp octet(s) (by CRC-S1).

7.1.1.7 Time Coordination section

The Time Coordination section shall be sent from the consumer to the producer in response to a change in the ping count.

The Time Coordination section is used as the Time Coordination Message for all point-to-point and multipoint time coordination requests.

The encoding for the time coordination section is defined in Figure 22. The variables used in the time coordination section, brief descriptions and references to complete descriptions are given in Table 76.

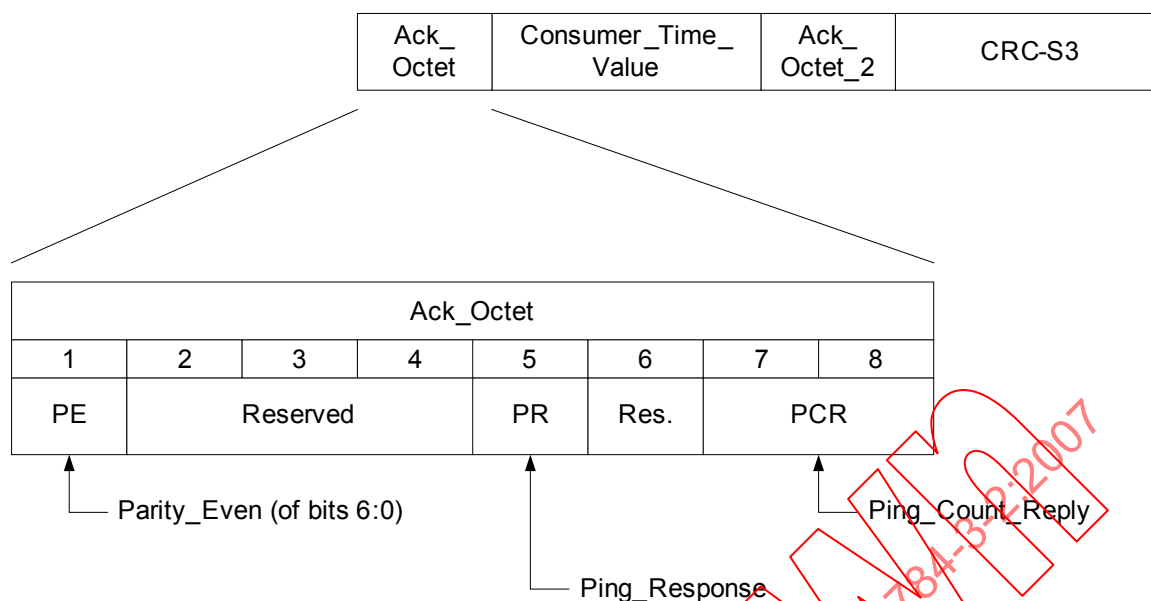


Figure 22 – Time Coordination message encoding

Table 76 – Time Coordination message variables

Name	Description	Reference
Consumer_Time_Value	Consumer_Time_Value in the Time Coordination message shall (as closely as possible) mark the time at which the consumer sends a ping response message	7.6.3.2
Ack_Octet	Consisting of:	
Parity_Even	The Parity_Even bit in the Time Coordination message shall be even parity of bits 0 through 6 of the message	
Ping_Response_Bit	The Ping_Response_Bit in the Time Coordination message shall indicate that a ping response is being sent	7.6.3.1
Ping_Count_Reply	Ping_Count_Reply in the Time Coordination message shall indicate which ping request the consumer is responding to	7.6.3.3
Reserved	Reserved, set to zero	
CRC-S3	CRC covering the data within the time coordination message	

The following rule shall be used for the calculation of the parity bit in the Time Coordination message:

Bit 7 of the Ack_Octet form even parity on the Ack_Octet.

The following rule shall be used for setting of the bits in Ack_Octet_2 of the Time Coordination message:

$$\text{Ack\_Octet\_2} = ((\text{Ack\_Octet} \text{ XOR } \text{FF}_{\text{hex}}) \text{ AND } \text{55}_{\text{hex}}) \text{ OR } (\text{Ack\_Octet} \text{ AND } \text{AA}_{\text{hex}}).$$

7.1.1.8 Time Coordination CRC calculation

The Time Coordination CRC calculation shall include:

- Consumer Identifier (PID) (by CRC-S3);
- Ack Octet (by CRC-S3);
- Consumer_Time_Value (by CRC-S3).

Ack_Octet_2 in the Time Coordination message shall not be included in the Safety CRC.

7.1.1.9 Time Correction section

The Time Correction message shall be sent from a multipoint producer to each multipoint consumer, once every ping interval. The Time Correction message sends consumer specific information to the consumer at a slower rate than the produced data. The encoding for the time correction message is in Figure 23. The variables used in the time correction section are defined in Table 77.

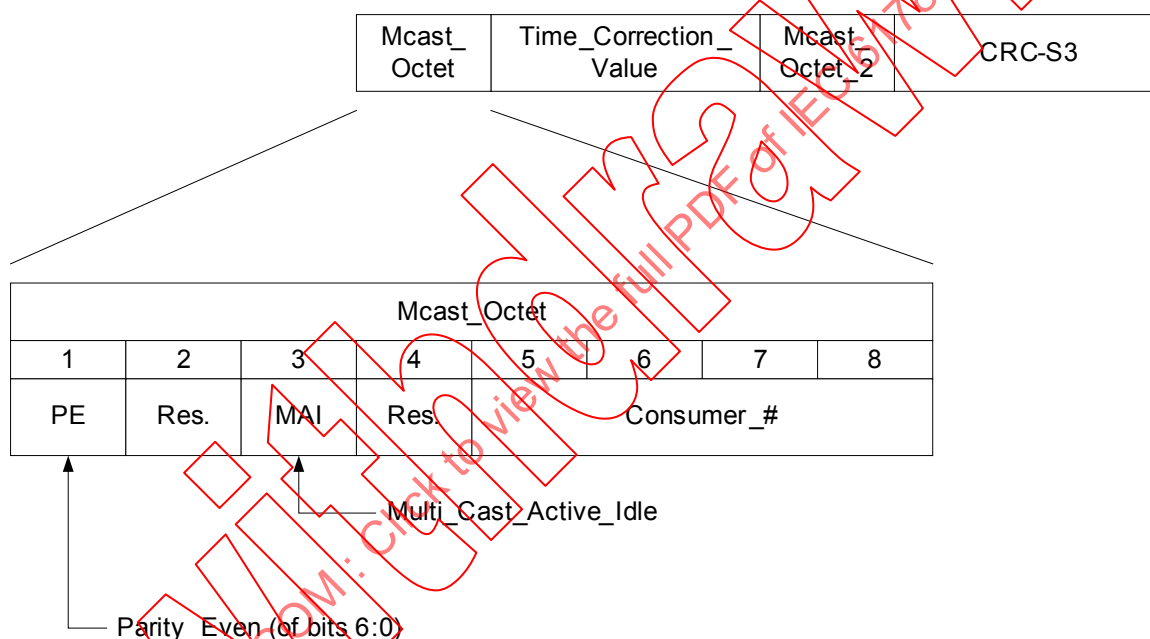


Figure 23 – Time Correction message encoding

Table 77 – Time Correction Message variables

Name	Description	Reference
Consumer_Time_Correction_Value	Consumer_Time_Correction_Value in the Time Correction message shall be used to correct the data time stamp and make the time relative to the multipoint consumers clock	7.6.4.2
MCast_Octet	Consisting of:	
Parity_Even	Parity_Even in the Time Correction message shall be the even parity of bits 0 through 6 of the Message	
Multi_Cast_Active_Idle	Multi_Cast_Active_Idle in the Time Correction message shall indicate Idle if transmitted before this consumer has responded with a valid time coordination message	7.6.4.3
Consumer_#	The Consumer_# in the Time Correction message shall indicates the number of the consumer that this message is directed to	7.6.4.1
Reserved	Reserved, set to zero	7.6.4.4
CRC-S3	CRC covering the data within the time coordination message	

The following rule shall be used for the calculation of the parity bit in the Time Correction message:

Bit 7 of the MCast_Octet form even parity on the MCast_Octet.

The following rule shall be used for setting of the bits in Mcast_Octet_2 in the Time Correction message:

$\text{MCast\_Octet\_2} = ((\text{MCast\_Octet} \text{ XOR } \text{FF}_{\text{hex}}) \text{ AND } 55_{\text{hex}}) \text{ OR } (\text{MCast\_Octet} \text{ AND } \text{AA}_{\text{hex}}).$

7.1.1.10 Time Correction CRC calculation

The Time Correction CRC calculation shall include:

- Producer Identifier (PID) (by CRC-S3);
- Mcast_Octet (by CRC-S3);
- Time_Correction_Value (by CRC-S3).

MCast_Octet_2 of the Time Correction message shall not be included in the Safety CRC.

7.1.1.11 1 or 2 Octet, point-to-point, safety connection format

Figure 24 defines the encoding for 1 or 2 octet point-to-point safety connection format.

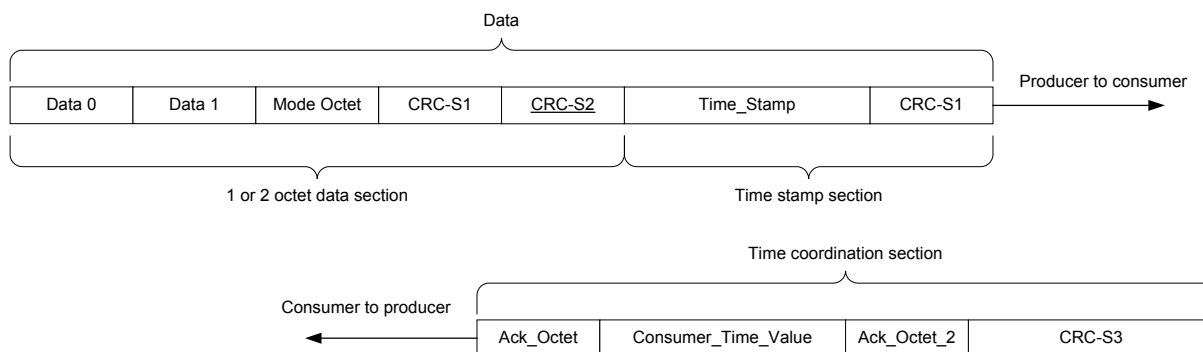


Figure 24 – 1 or 2 octet point-to-point PDU encoding

7.1.1.12 1 or 2 Octet, multipoint, Format 1 connection format

Figure 25 defines the encoding for the 1 or 2 octet Format 1 connection format.

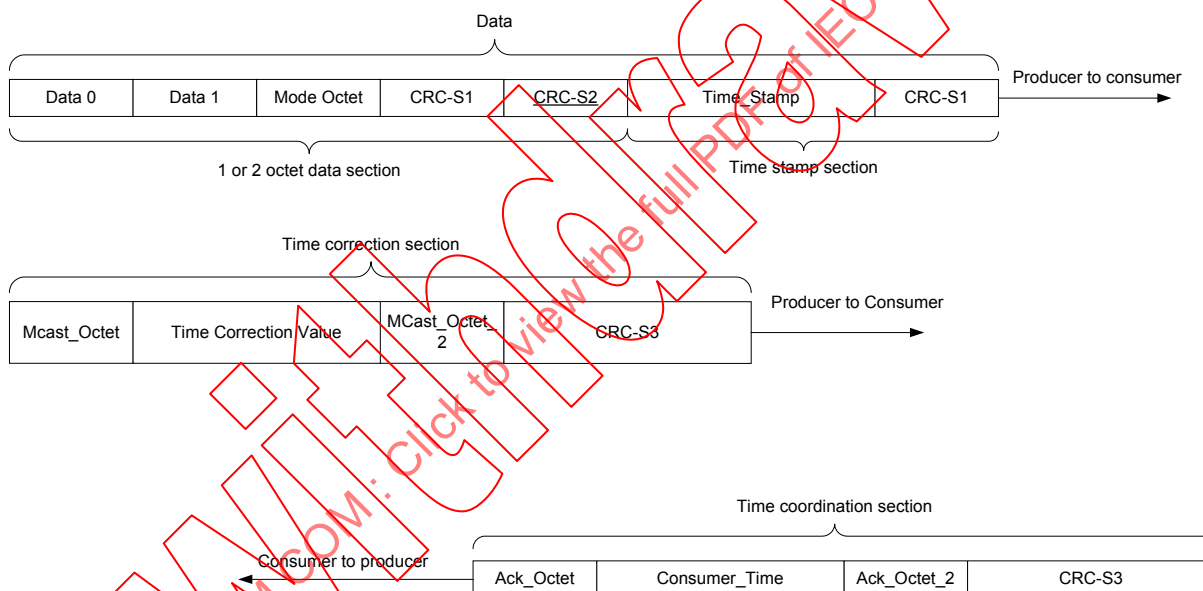


Figure 25 – 1 or 2 Octet multipoint PDU encoding

7.1.1.13 1 or 2 Octet, multipoint, Format 2, safety connection format

Figure 26 defines the encoding for 1 or 2 octet multipoint format 2 safety connection format.

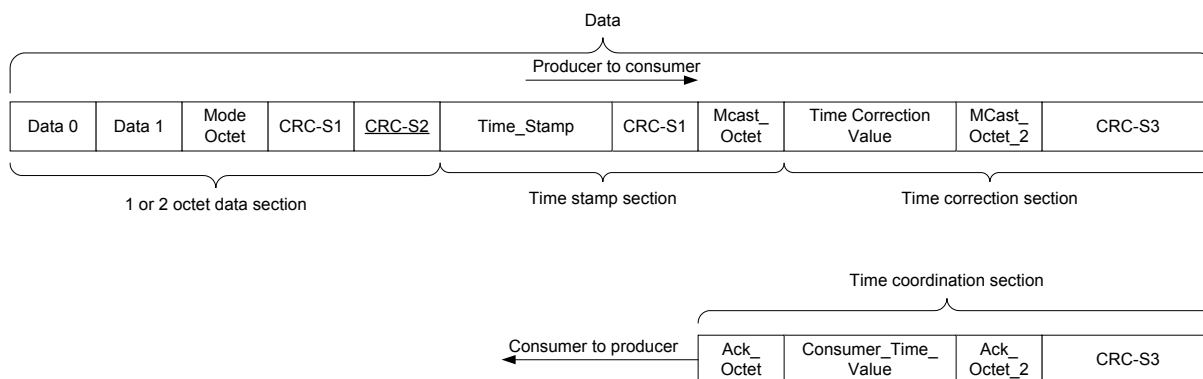


Figure 26 – 1 or 2 Octet, multipoint, Format 2 safety connection format

7.1.1.14 3 to 250 octet, point-to-point, safety connection format

Figure 27 defines the encoding for 3 to 250 octet point-to-point safety connection format.

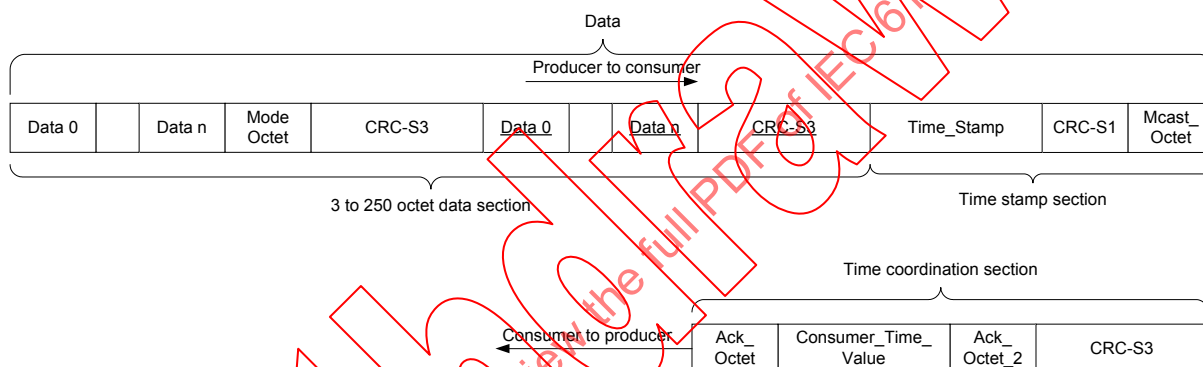


Figure 27 – 3 to 250 Octet Point-to-point PDU encoding

7.1.1.15 3 to 248 Octet, multipoint, Format 1 Safety PDU format

Figure 28 defines the encoding for 3 to 248 octet multipoint safety connection format.

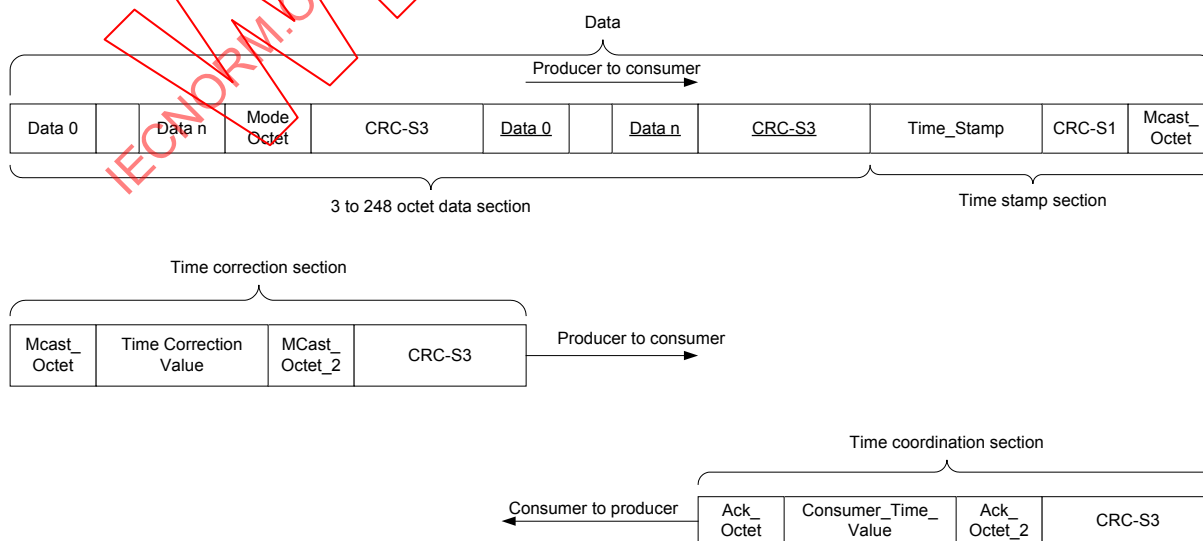


Figure 28 – 3 to 248 Octet Multipoint PDU encoding

7.1.1.16 3 to 248 Octet, multipoint, Format 2, safety PDU format

Figure 29 defines the encoding for 3 to 248 octet multipoint safety connection format.

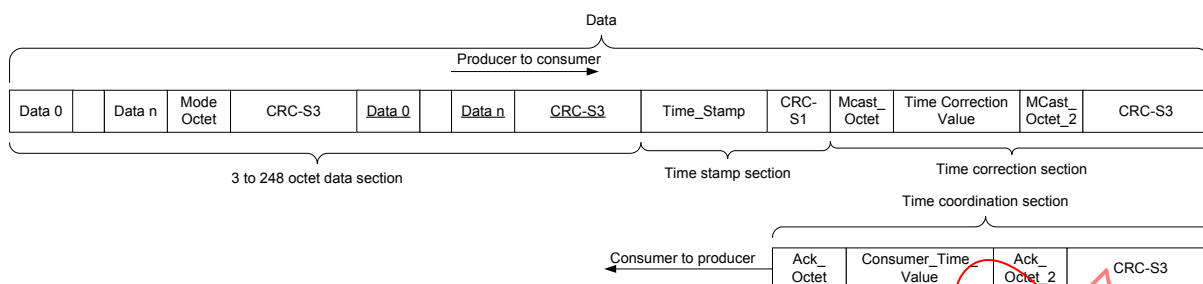


Figure 29 – 3 to 248 Octet, Multipoint, safety connection format

7.1.2 Safety CRC

7.1.2.1 Safety CRC overview

There are three safety CRCs used in FSCP 2/1. An 8-bit CRC (CRC-S1 or CRC-S2) is used for the data sections connections containing up to two octets of safety data and the time stamp sections. A 16-bit CRC (CRC-S3) is used for data section for 3 to 250 octets of data and for the time correction and time coordination messages. A fourth safety related CRC, a 32-bit CRC (CRC-S4) is used for connection establishment, configuration, and the Safety extension to the EDS file definition.

7.1.2.2 CRC polynomials

Table 78 defines the polynomials used in FSCP 2/1.

Table 78 – CRC polynomials used

Name	Generator polynomial	Hex value of polynomial
CRC-S1	$g(x) = x^8 + x^5 + x^4 + x^2 + x^1 + 1$	137 _{hex}
CRC-S2	$g(x) = x^8 + x^5 + x^4 + x^3 + x^1 + 1$	13b _{hex}
CRC-S3	$g(x) = x^{16} + x^{11} + x^3 + x^2 + x^1 + 1$	1080F _{hex}
CRC-S4	$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^4 + x^2 + x^1 + 1$	Bit-reversed to EDB88320 _{hex}

7.1.2.3 CRC usage specifications

The I/O messaging CRC usage is defined in 7.1 using CRC-S1, CRC-S2, and CRC-S3, but there are a number of other CRCs defined in FSCP 2/1 for connection establishment, configuration, and the Safety extension to the EDS file definition. Table 79 defines which algorithm is used and the starting seed value for each use.

Table 79 – Connection sections and message formats

CRC name	Algorithm used	Starting seed value	Description
SCCRC	CRC-S4 (see A.1)	FFFFFFFF _{hex}	Safety Configuration CRC
CPCRC	CRC-S4 (see A.1)	FFFFFFFF _{hex}	Connection Parameter CRC
Password CRC	CRC-S4 (see A.1)	FDFDFDFD _{hex}	Used to encrypt passwords before transmission
EDS File CRC	CRC-S4 (see A.1)	FDFDFDFD _{hex}	Used to provide integrity over EDS files used by safety devices

7.2 Communication protocol behavior

7.2.1 Sequence of safety checks

The following Sequence of checks shall be used to check messages.

- Ping count check;
- Evaluate Time stamp section CRC;
- Evaluated Time Stamps;
- Evaluate Actual Data CRC;
- Evaluate Complement Data CRC;
- Perform Cross-Check.

7.2.2 Connection termination

When a producer detects an error that requires connection termination (i.e. ping response outside of Producer Ping Interval) it shall terminate the connection, and notify the application of the action.

All consumers shall monitor the periodic transmission of data and go to a safety state if the periodic transmissions cease. If the consumer detects an error, it shall go to a safety state and terminate the consuming connection associated with that error.

When the Safety Layer detects an error requiring connection termination the termination shall be implemented using the following sequence.

- The safety layer detects an error requiring termination.
- The safety layer notifies the application program by setting the connection status to indicate a safety communications fault.
- The safety-related application shall transition data and I/O (for example set outputs to the safety state) associated with the connection to a safety state.
- The safety layer shall notify the underlying communications system of an error and request the termination of the connection by setting its status to indicate a safety communications error.
- The safety layer shall not transition from its safety state until the fault is cleared and a connection restart sequence is initiated.

7.2.3 Cross checking error

When crosschecked safety data are found to be different, the data is treated as faulty and the consumer shall terminate the connection (see 7.2.2).

7.3 Time stamp operation

The time stamp allows for the calculation of the timeliness of receipt of data in the consumer. Figure 30 shows the time stamp sequence of operations for a simple point-to-point producer consumer pair.

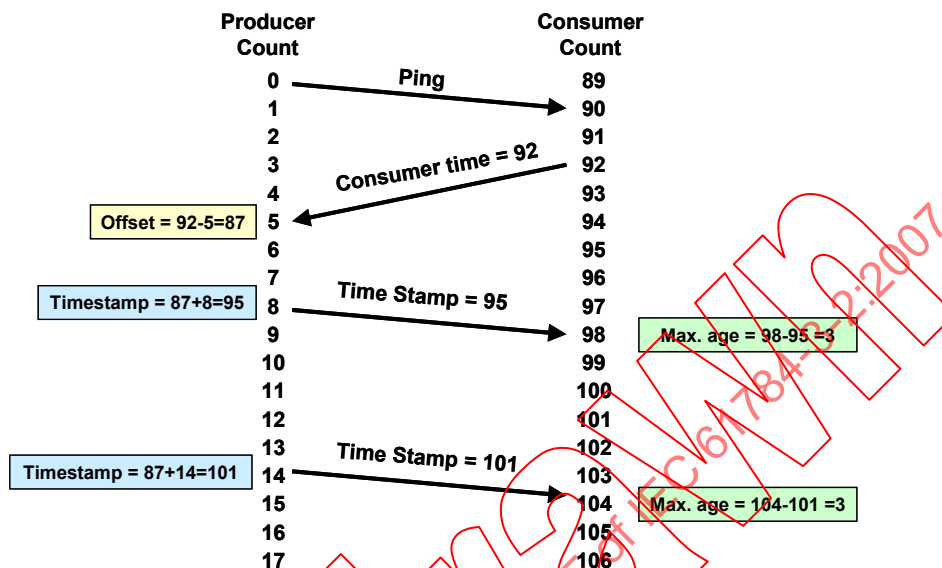


Figure 30 – Time stamp sequence

Upon connection establishment, the producer shall request an initial ping response.

When the consumer sees a change in the ping count, the consumer shall prepare a Time Coordination response message that contains the consumer's value of its clock count.

The producer shall use the clock count in the Time Coordination response to calculate an offset that will be applied to future productions.

At each scheduled production, the producer shall use the value of its own clock count and the offset to calculate the time stamp and sends it to the consumer along with the data.

The consumer shall use the value received from the producer to calculate the age of the data by subtracting the producer's time stamp from its current clock count. If the calculated delta is less than the maximum delta specified by the user, the data is ok and the data is used by the application within the device. If the delta is greater than the user specified value, the data is deemed to be too old to use and the connection is terminated.

Once each Ping Interval, the producer shall request another ping response from the consumer. This will cause a recalculation of producer offset and prevent the slow aging of data due to time errors.

A time correction factor is added to account for the maximum skew and drift of clocks in an asynchronous system. This message is used to adjust the offset in the producer.

7.4 Protocol sequence diagrams

7.4.1 General

The protocol sequence diagrams show the relationship between producers and consumers with respect to time and events. Time flows from the top of each figure to the bottom. Normal messages that are received at the producer or consumer are shown in normal text boxes. Anomalous messages are shown in bolded text boxes. Messages that were expected but not received are shown as unboxed text.

7.4.2 Normal safety transmission

Normal communication between a safety producer and consumer requires a sequence of periodic messages. When the consumer receives a valid message, the transaction is complete as defined in Figure 31.

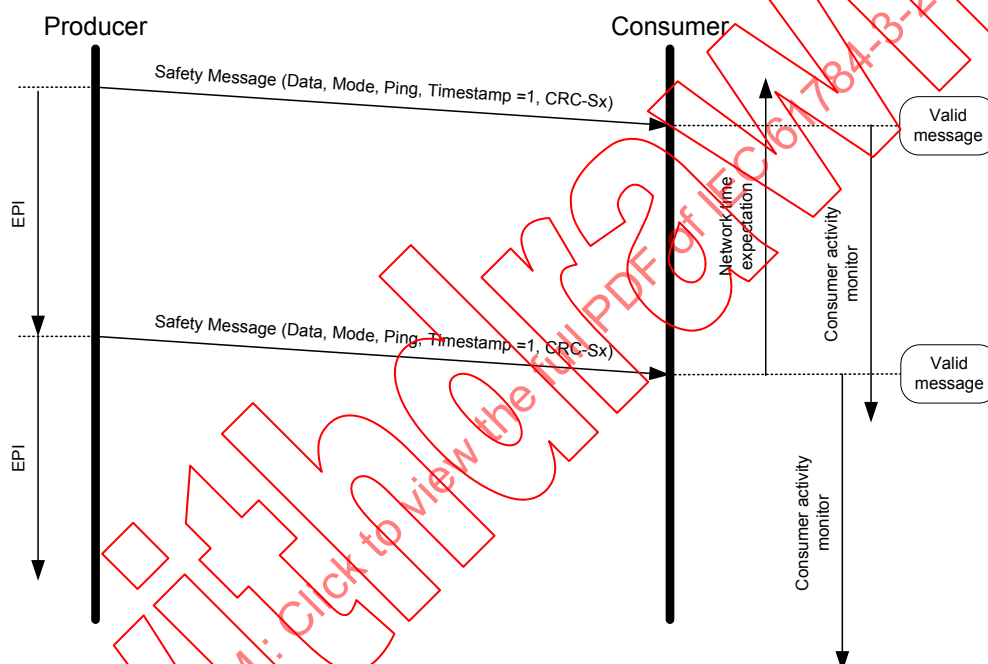


Figure 31 – Sequence diagram of a normal producer/consumer safety sequence

The safety producer produces data at a fixed rate called the Expected Packet Interval (EPI).

Link-triggered consumers shall maintain an activity timer, which is reset after receiving a valid expected message.

If a valid message has not been received before the activity timer expires, the link-triggered consumer shall terminate the connection (see 7.2.2) and go to the defined safety state.

Consumers shall check the age of the data received by checking the time stamps received.

If the age of the data is greater than the network time expectation the consumer shall terminate the connection (see 7.2.2) and go to the defined safety state.

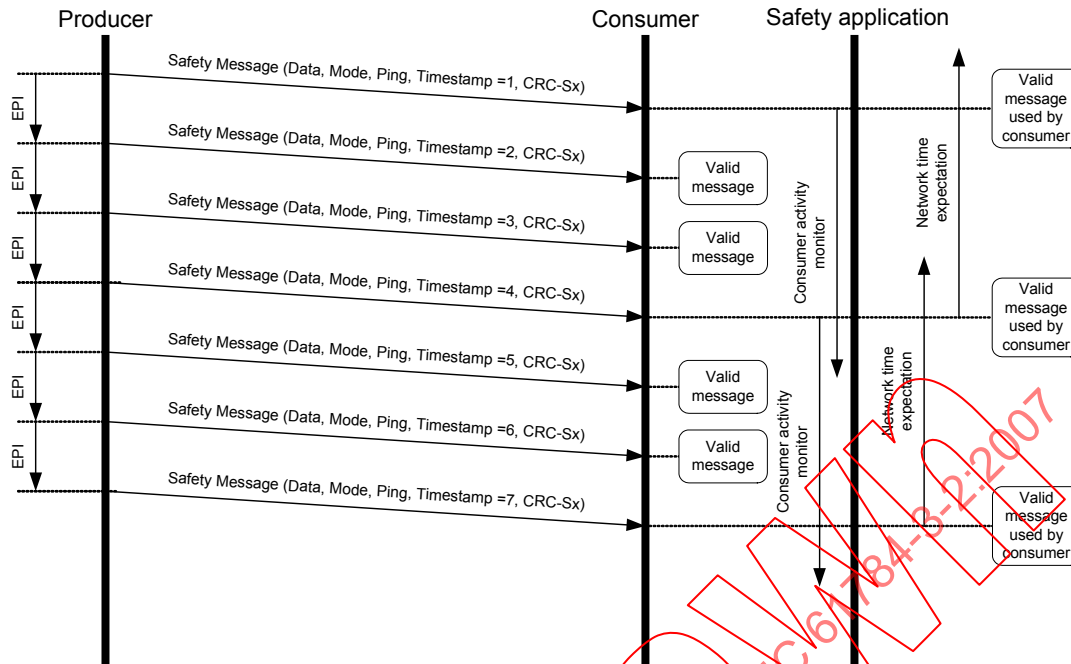


Figure 32 – Sequence diagram of a normal producer/consumer safety sequence (production repeated)

Safety communications shall be configurable with production repeated. In this case, the consumer does not use every packet that is sent to it.

When safety communication is sent with repeated production, application-triggered consumers shall use the last packet received to complete the transaction, as show in Figure 32.

The consumer shall see the Timestamp increase for each successive period, or the connection will be terminated (see 7.2.2).

Messaging with production repeated offers the additional advantage of using the most current data.

7.4.3 Lost, corrupted and delayed message transmission

Due to external disturbances it may be possible for the message to be corrupted as it is transmitted from the producer to the consumer, as defined in Figure 33. If a corrupted message is detected at the standard layer, it shall be discarded and not presented to the application. This will result in an expiration of the consumer activity monitor or the data age will exceed the network time expectation and the consumer terminates the connection (see 7.2.2) and goes to the defined safety state.

If the transmitted CRC-Sx and data are checked and found to be in error, the consumer shall conclude the underlying link is untrustworthy, terminate the connection and go to the defined safety state (see 7.2.2).

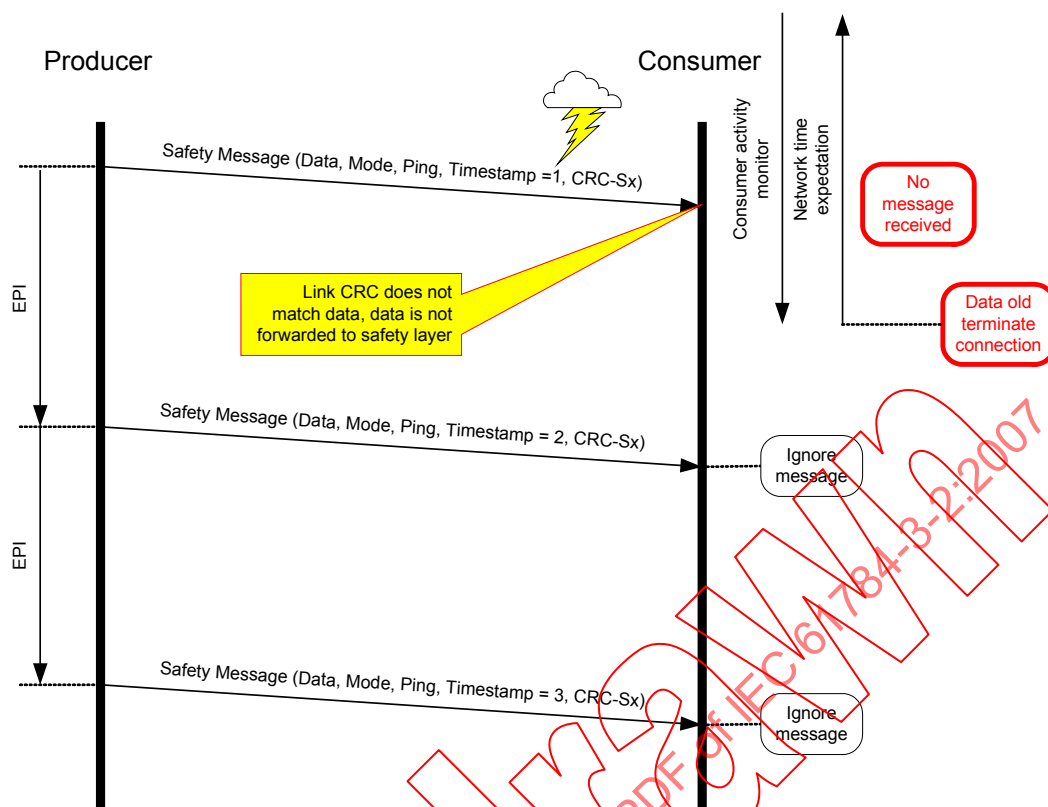


Figure 33 – Sequence diagram of a corrupted producer to consumer message

Due to external disturbances it may be possible for the message to be lost as it is transmitted from the producer to the consumer, as defined in Figure 34. This will result in an expiration of the consumer activity monitor or the data age will exceed the network time expectation and the consumer terminates the connection (see 7.2.2) and goes to the defined safety state.

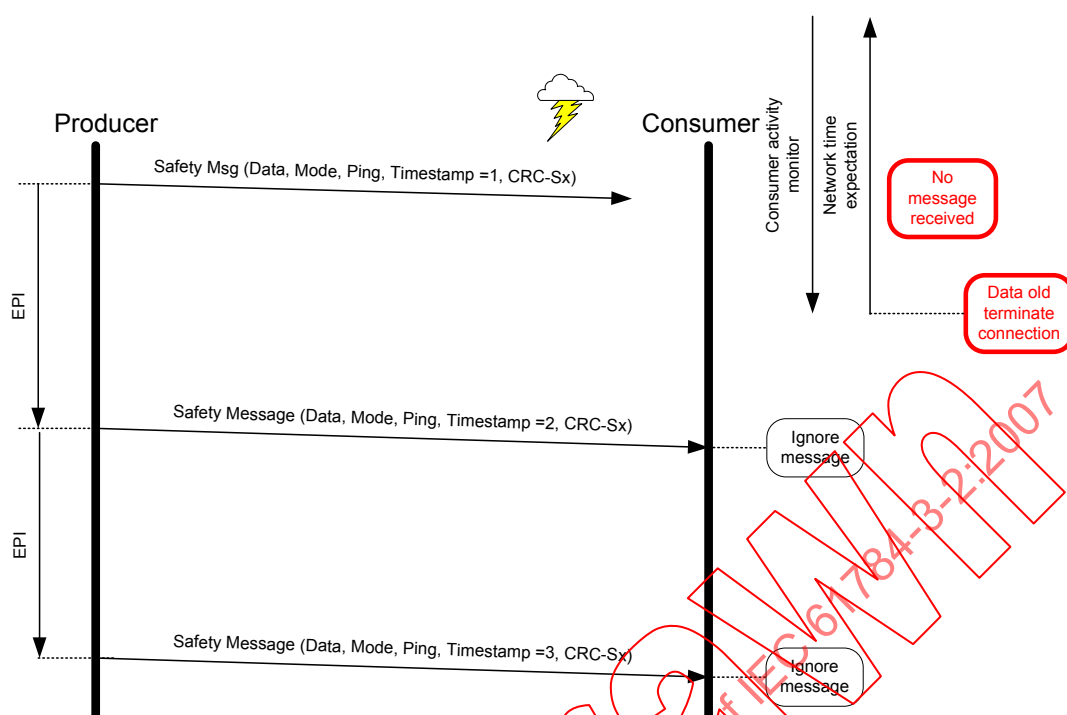


Figure 34 – Sequence diagram of a lost producer to consumer message

Messages may be delayed in transmission. If a message is delayed such that it is received

- beyond the consumer activity monitor (link-triggered application) or
- the data age exceeds the network time expectation,

Then all further messages are ignored, and the connection shall be terminated.

In both cases the consumer terminates the connection (see 7.2.2) and goes to the defined safety state. This behavior is defined in Figure 35.

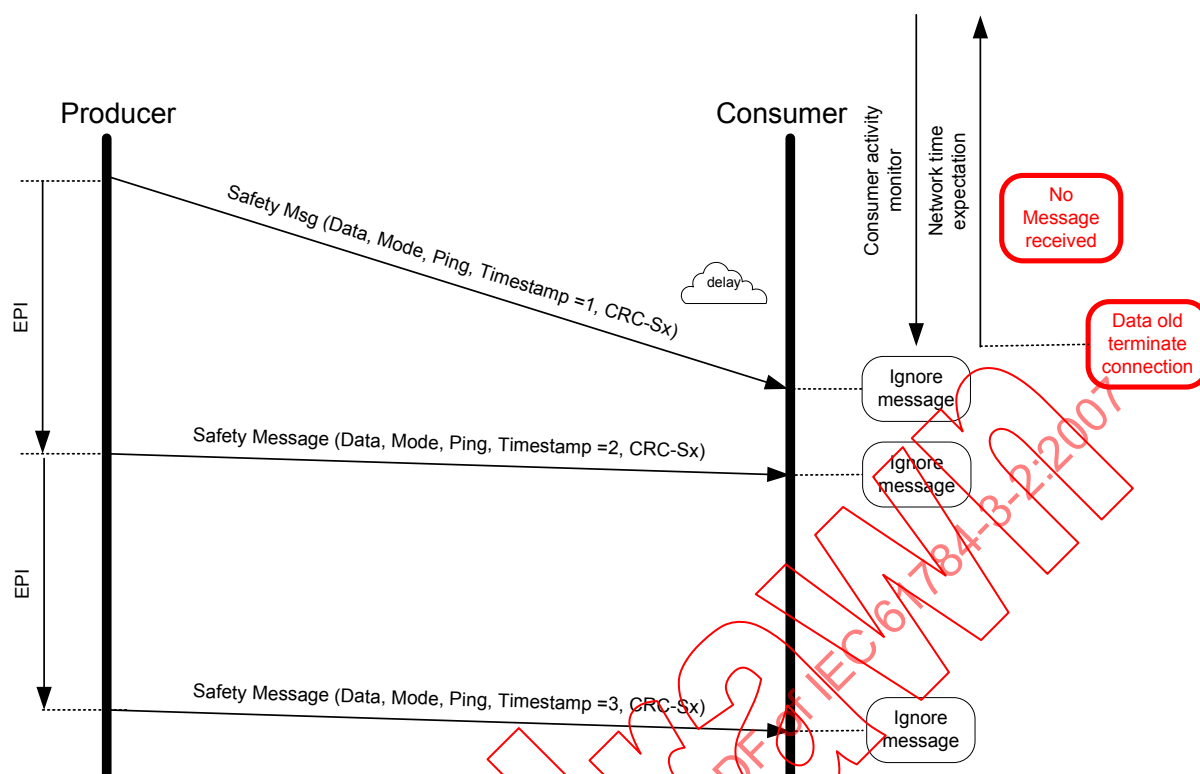


Figure 35 – Sequence diagram of a delayed message

7.4.4 Lost, corrupted or delayed message transmission with production repeated

Due to external disturbances it may be possible for the message to be corrupted as it is transmitted from the producer to the consumer, as defined in Figure 36. In order to, minimize nuisance trips and increase the availability of the overall system, production may be repeated.



Figure 36 – Sequence diagram of a corrupted producer to consumer message with production repeated

If there are errors that are detected in the standard layers, the messages can be discarded, as long as a valid and expected message is received within the consumer activity monitor and the age of the data does not exceed the network time expectation.

Valid repeated messages shall be overwritten. The most recent valid message is used by the application.

Due to external disturbances, it may be possible for a series of messages to be delayed resulting in a lost connection. This sequence is defined in Figure 37. If no valid messages are received before the consumer periodic timer expires, the connection will be terminated (see 7.2.2).

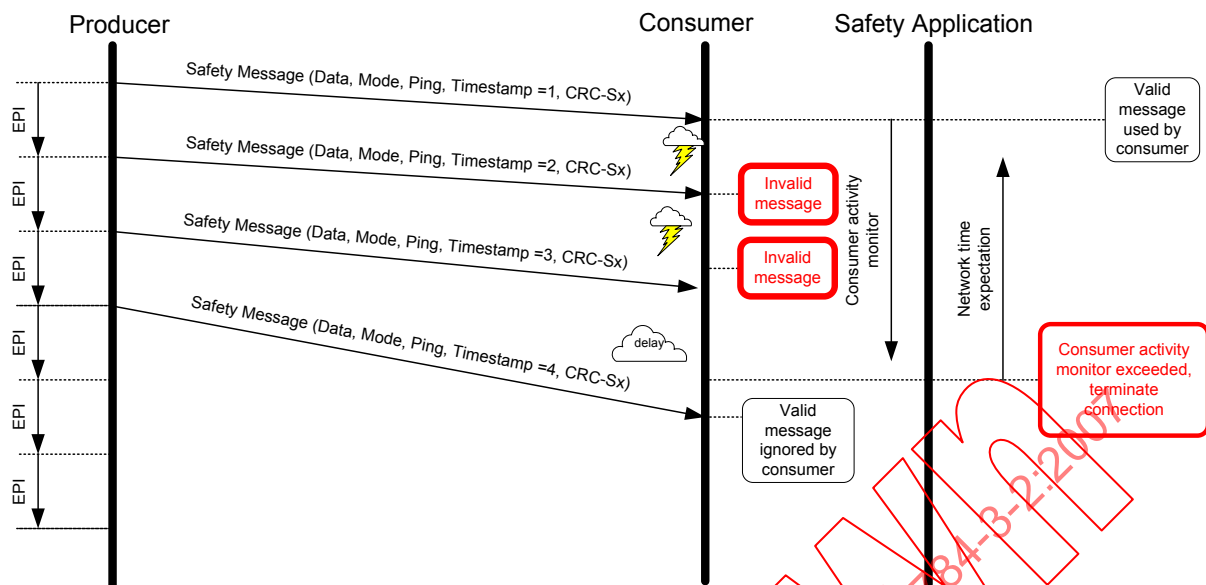


Figure 37 – Sequence diagram of a connection terminated due to delays

A valid link message that fails a Safety CRC check will cause the consumer to assume the underlying link is untrustworthy and the connection is immediately terminated. This sequence is defined in Figure 38.

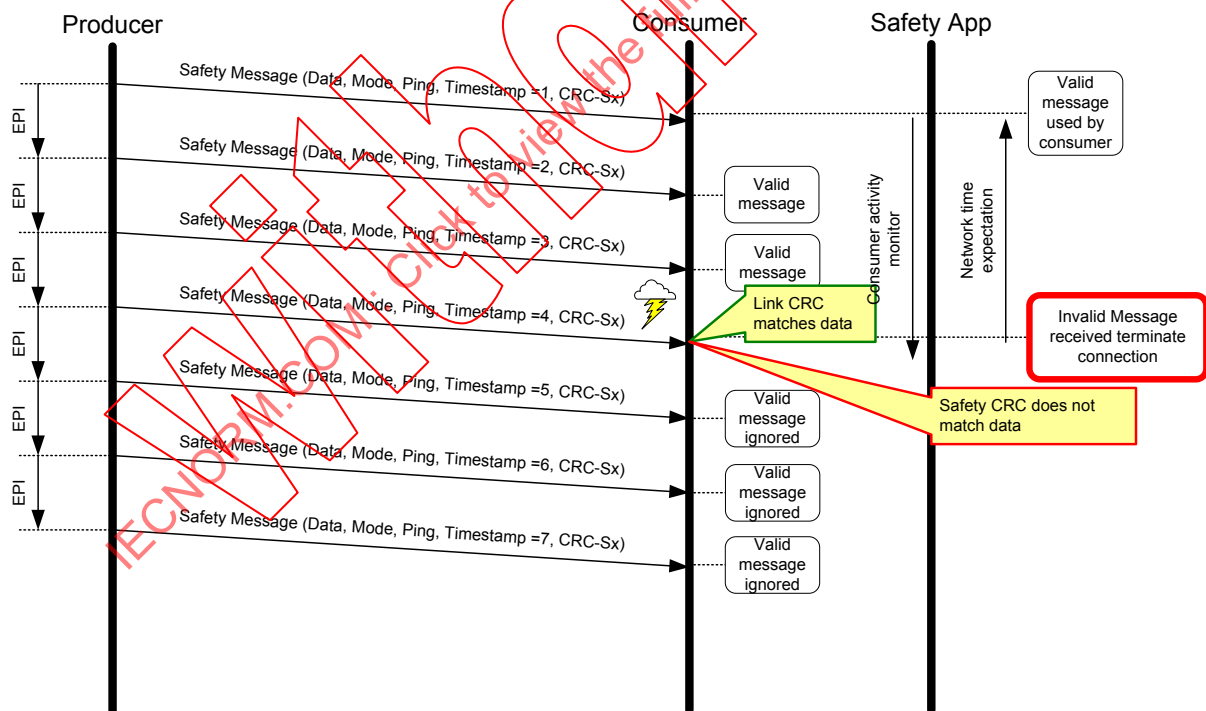


Figure 38 – Sequence diagram of a failure of safety CRC check

7.4.5 Point-to-point ping

In order to verify that messages are not being delayed in bridges a ping request and response message transaction sequence is used. The producer is fully described in 7.5.5 and the consumer is fully described in 7.5.6. This sequence of messages for a point-to-point ping is defined in Figure 39.

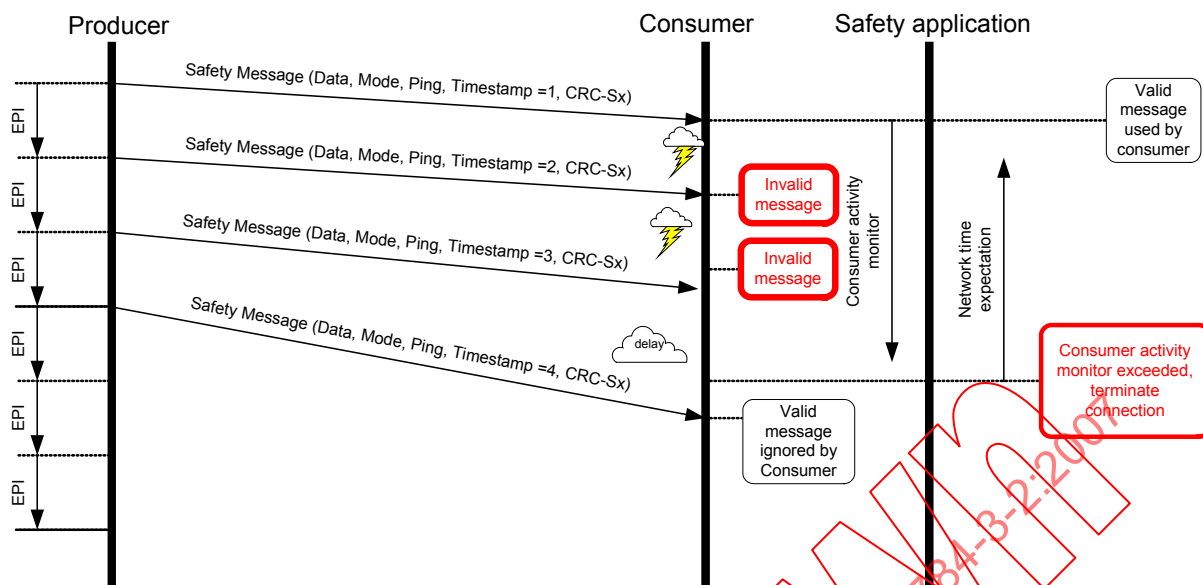


Figure 39 – Sequence diagram of a point-to-point ping - normal response

At each point-to-point producer ping interval, the ping count shall be incremented in the next available safety message header.

The point-to-point producer ping interval shall be a multiple of the EPI; ping requests can be made at rates from the EPI up to 0,5 s.

When a point-to-point consumer sees a change in the ping count the consumer generates a time coordination message.

The time coordination message (see 7.4.2) shall be received by the producer prior to the expiration of the current producer ping interval, if the time coordination message is not received within the Producer Ping Interval the connection shall be terminated by the producer (see 7.2.2).

7.4.6 Multipoint ping on CP 2/3 Safety

In order to verify that messages are not being delayed in bridges a ping request and response is used.

For CP 2/3 Safety Multipoint connections, a separate message shall be used to communicate time correction information from the produce to each multipoint consumer.

The producer is fully described in 7.5.5 and the consumer is fully described in 7.5.6. This sequence of messages for multipoint ping is defined in Figure 40. For clarity the time correction message is defined on a separate diagram.

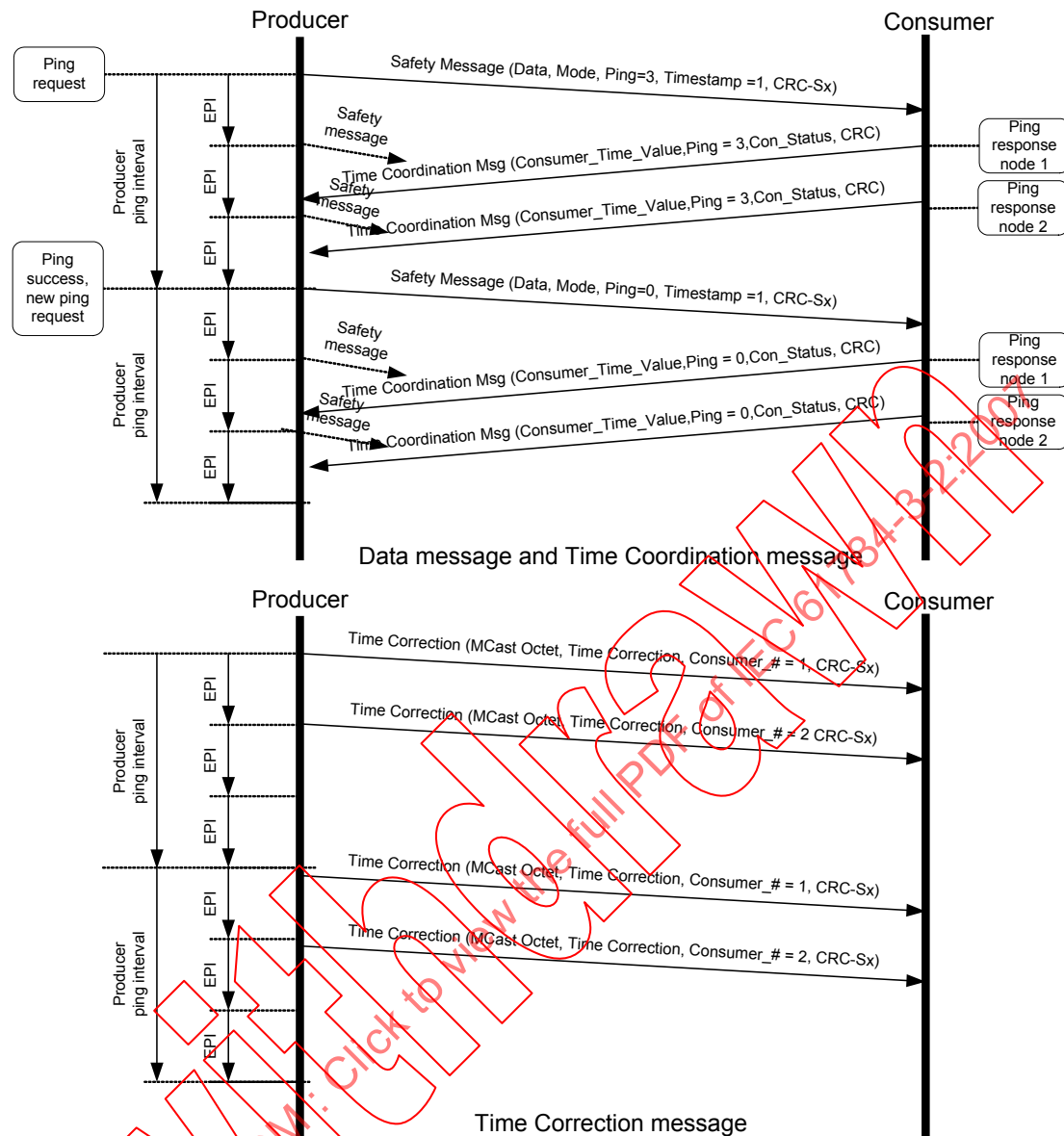


Figure 40 – Sequence diagram of a successful multipoint ping, CP 2/3 safety

During each multipoint producer ping interval, a ping request will be sent from the multipoint producers to multipoint consumers requesting a response.

All multipoint consumers shall respond to a change in the ping request value, consumers other than consumer 1 shall wait $\text{Consumer\_Number} - 1$ times the EPI to reply.

All of the multipoint consumers shall respond within a single producer ping interval.

Time correction messages shall be sent to the multipoint consumers at a rate to ensure that each consumer gets a time correction message within a single producer ping interval.

7.4.7 Multipoint ping on CP 2/2 safety networks

In order to verify that messages are not being delayed in bridges a ping request and response is used. For CP 2/2 Safety networks the time coordination message shall be concatenated with the safety data message and sent as a single message. The producer is fully described in 7.5.5 and the consumer is fully described in 7.5.6. This sequence of messages for multipoint ping is defined in Figure 41.

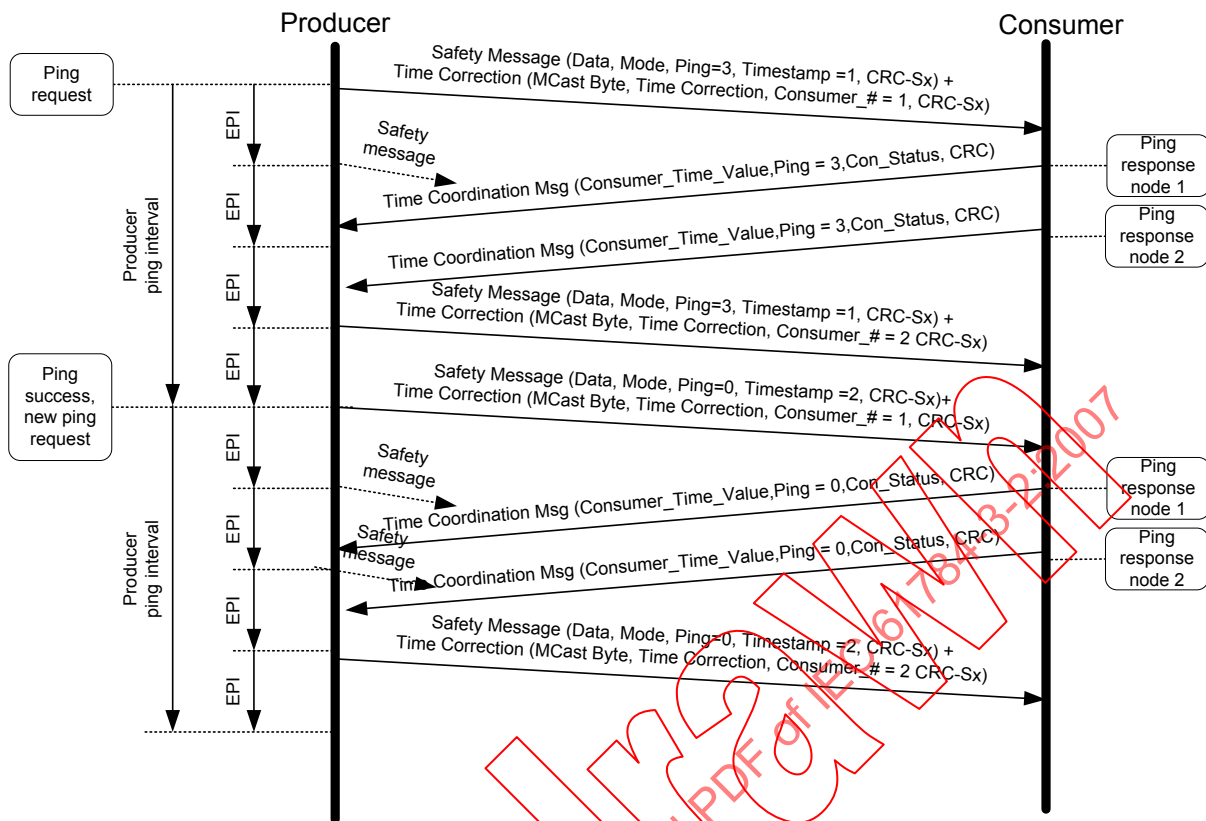


Figure 41 – Sequence diagram of a successful multipoint ping, CP 2/2 safety

During each producer ping interval, a ping request is sent from the multipoint producers to multipoint consumers requesting a response. All consumers respond to a change in the ping request value, consumers other than consumer 1 will wait Consumer_Number -1 times the EPI to reply. All of the multipoint consumers shall respond within a single producer ping interval.

Time correction messages are sent to the consumer at a rate to ensure that each consumer get a time correction message within a single producer ping interval.

7.4.8 Multipoint ping – retry with success

In order to verify that messages are not being delayed in bridges a ping request and response is used. The producer is fully described in 7.5.5 and the consumer is fully described in 7.5.6. This sequence of messages for a multipoint ping with retries is defined in Figure 42. The time correction connection is not defined for clarity.

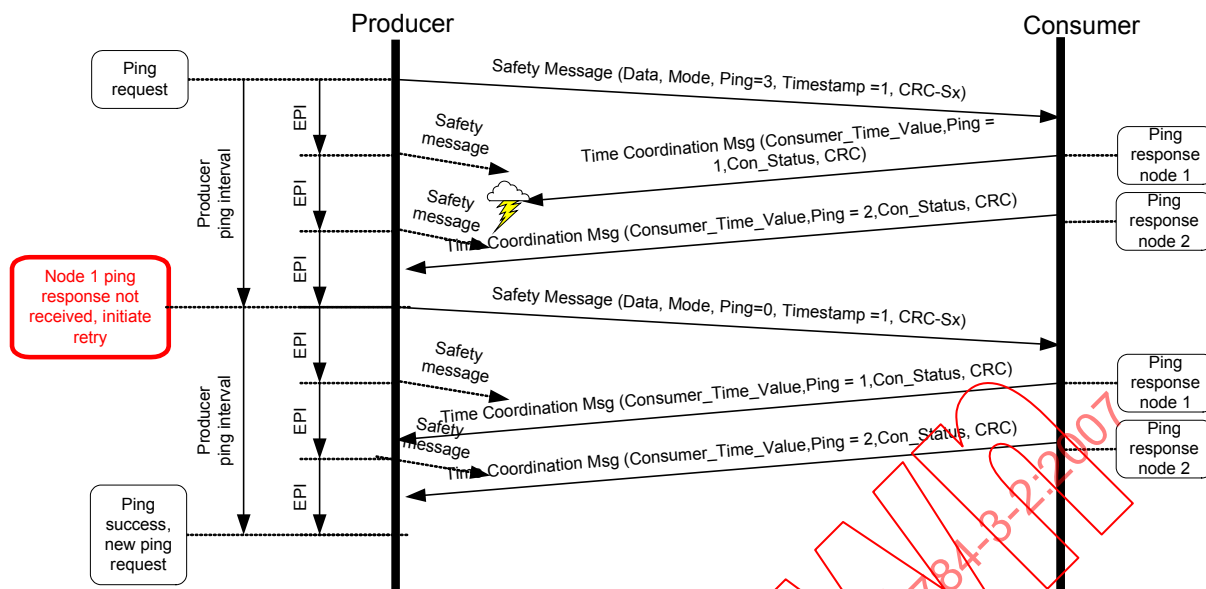


Figure 42 – Sequence diagram of a multipoint ping retry

If a multipoint time coordination message is lost or delayed past the Timeout_Multiplier + 1 number of producer ping response timeouts, an error shall be generated and the connection closed.

7.4.9 Multipoint ping – retry with timeout

In order to verify that messages are not being delayed in bridges, a producer using a ping request and response is used. The producer is fully described in 7.5.5 and the consumer is fully described in 7.5.6. This sequence of messages for a multipoint ping with retries and timeout is defined in Figure 43. The time correction connection is not defined for clarity.

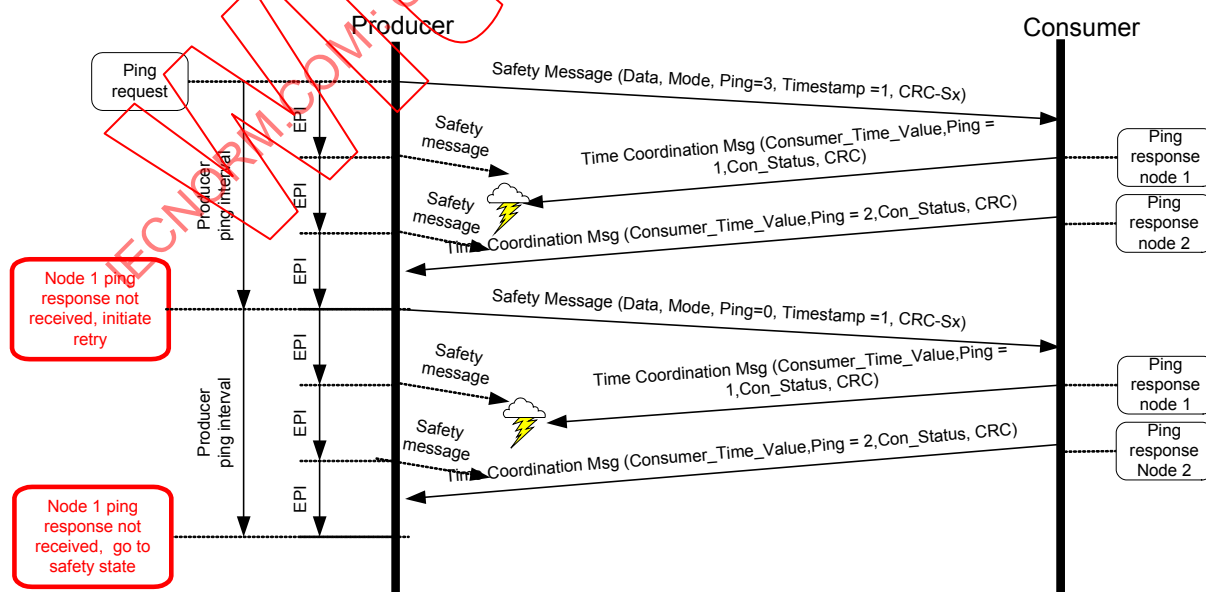


Figure 43 – Sequence diagram of a multipoint ping timeout

If the ping response continues to be delayed, the producer will indicate a fault on in the producer status for the specific consumer and the consumer will go to the defined safety state. If the ping was delayed due to a fault in the consumer, the consumer will have already gone to a safety state.

7.5 Safety protocol definition

7.5.1 General

The safety layer is a layer of service residing between the standard CPF 2 transport layer and the safety-related application layer as defined in Figure 44. This safety layer need not be directly addressable in the CPF 2 framework; rather it is a behavior associated with a connection between applications exchanging data in a SIL 3 application. This behavior is implemented as the Safety Validator object as described in 6.7.

NOTE There may be multiple methods of accomplishing this desired behavior. The objects in this subclause are to be considered a reference model. Individual products and applications may choose to implement this behavior, as they deem necessary. However, this behavior needs to be expressed here to form a basis for product development.

7.5.2 High level view of a safety device

Figure 44 shows a reference model of a safety device containing one or more safety-related application objects. Each safety-related application objects functions with a pair of Safety Validator objects to exchange SIL 3 data between the CPF 2 transport layer and the safety I/O electronic circuitry. These electronics are designed to provide a level of error detection and fault isolation suitable for SIL 3 applications. Each of the two Safety Validator objects is associated with a network communication channel such as CP 2/3 or CP 2/2. This model assumes a 1oo2 Safety Topology in which two independent CPUs participate in the safety protocol.

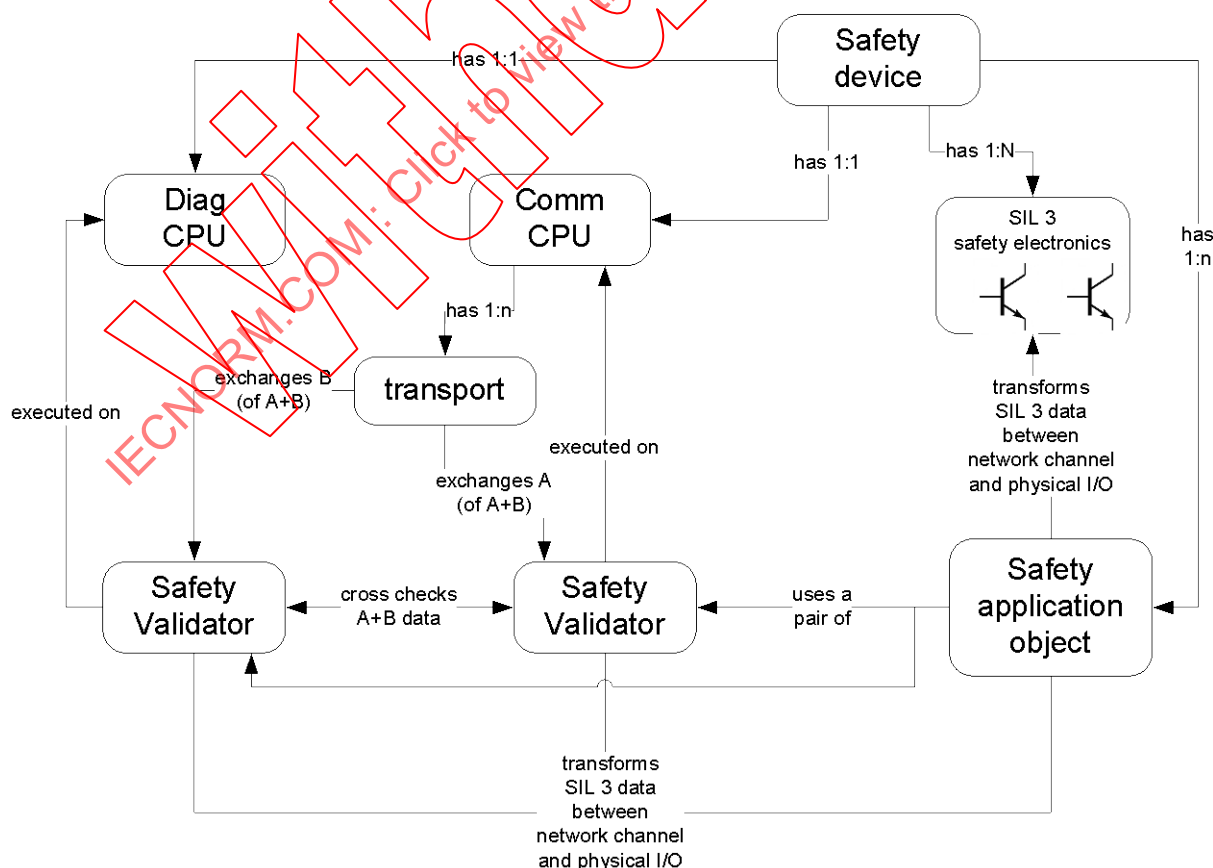


Figure 44 – Safety device reference model entity relation diagram

7.5.3 Safety Validator object

Subclause 6.7 defines a Safety Validator object from which both the SafetyValidatorServer and SafetyValidatorClient objects are implemented.

The general principles and common attributes of these two functions within the object are defined in 7.5.4.

7.5.4 Relationship between SafetyValidatorServer and SafetyValidatorClient

Figure 45 shows a high level view of two devices interchanging data via a SafetyValidatorClient and a SafetyValidatorServer. A safety producing application uses a SafetyValidatorClient to send safety data to a safety consuming application that uses a SafetyValidatorServer.

NOTE Only one of the redundant paths interfaces to the CPF 2 transport layer. The A+B (actual and complement) data is encapsulated in a single transport frame and only one processor is involved in the reception and transmission of the data frames.

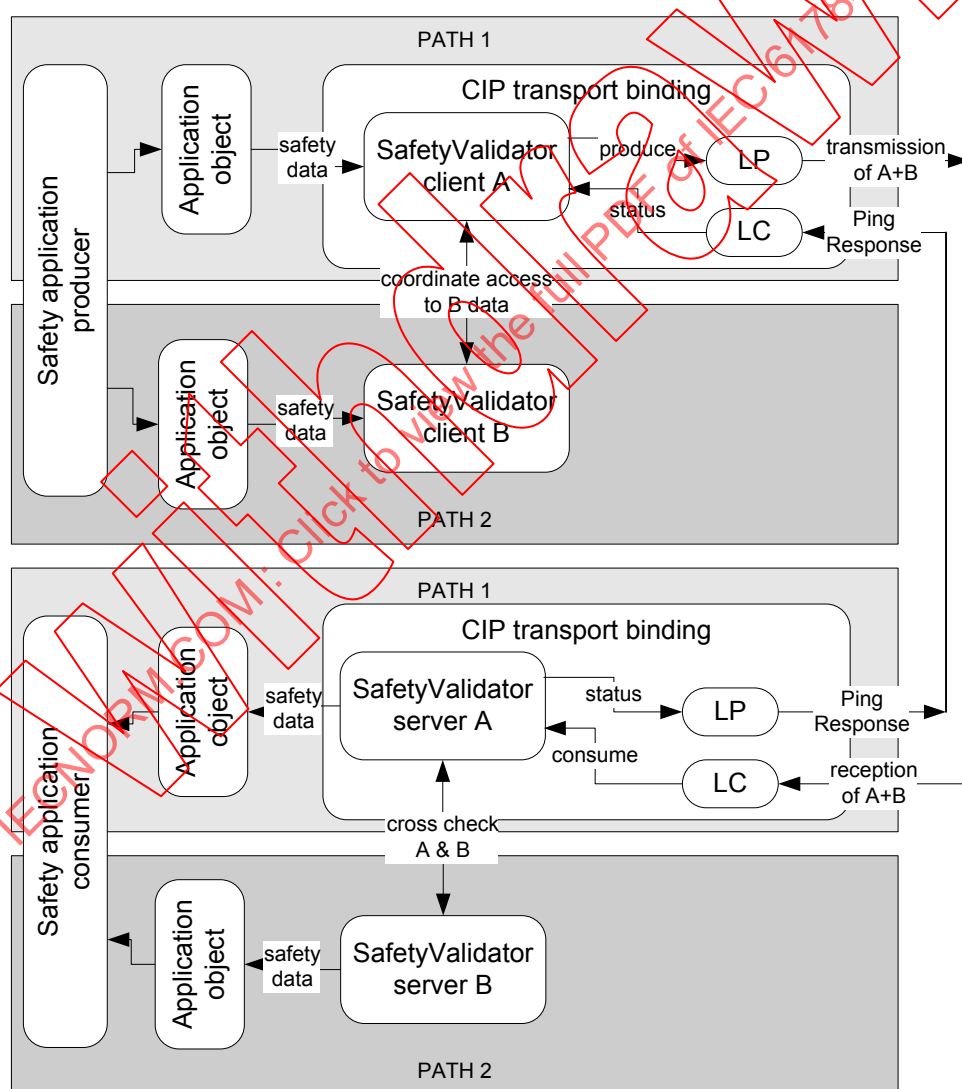


Figure 45 – Two devices interchanging safety data via a SafetyValidatorClient and a SafetyValidatorServer

7.5.5 SafetyValidatorClient function definition

7.5.5.1 General

The SafetyValidatorClient Class is the embodiment of the safety layer that augments a standard connection on CP 2/3 or CP 2/2 to transmit SIL3 safety data. It coordinates the production of application data with a peer instance of the SafetyValidatorClient class.

7.5.5.2 Safety production

This subclause describes the production of data and the interfaces between the application, the SafetyValidatorClient and the underlying link layer. The focus is on the SafetyValidatorClient run-time behavior and its interfaces. The application itself shall be of high integrity and this can be accomplished in many ways and is ultimately a vendor design decision.

Figure 46 graphically shows the data flow between the SafetyValidatorClient and a producing safety-related application. Safety_Data_Out is produced at a periodic rate, referred to as the Expected Packet Interval (EPI). The SafetyValidatorClient shall sample, capture, and time stamp the data to be sent every EPI time period.

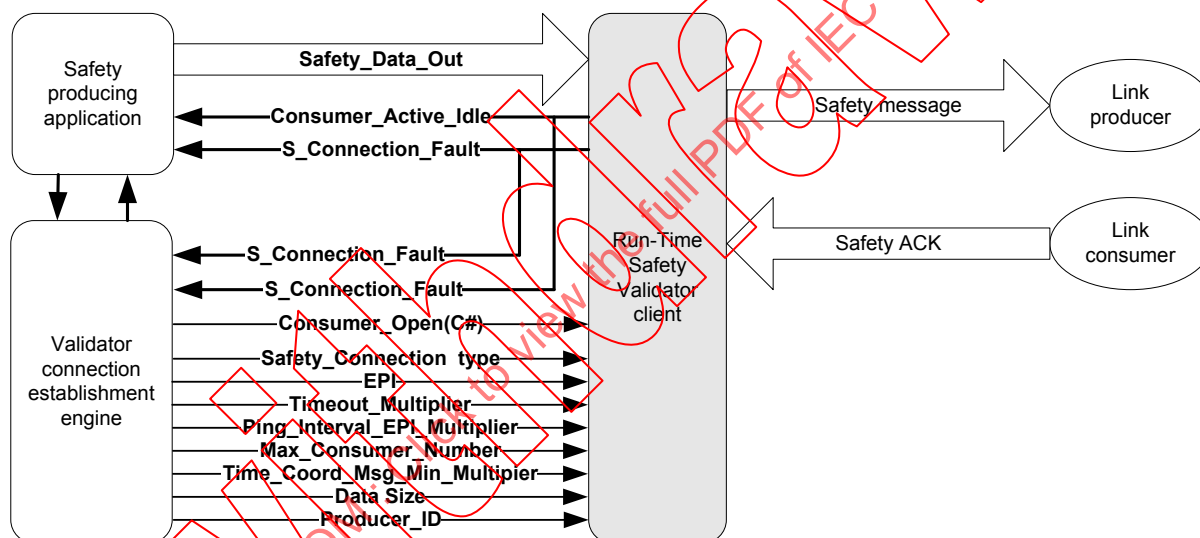


Figure 46 – Safety production data flow

7.5.5.3 Producing application interface

The producing application provides Safety_Data_Out to the SafetyValidatorClient. The SafetyValidatorClient shall build the Mode_Octet, Actual_Data, and Complement_Data, and Time stamp section.

The SafetyValidatorClient shall provide safety connection status back to the producing application for each consumer.

7.5.5.4 Safety data production logic

This subclause describes the logic that shall be followed by all safety data producers. The actual implementation of the logic may vary, but equivalent results shall be obtained. The logic in this part assumes an asynchronous producing application that makes the safety data available to the SafetyValidatorClient.

Definitions for the variables used in the safety data production logic can be found in 7.6.

7.5.5.5 Example safety data production cold start logic

The production logic to initiate a cold start of a connection is:

```

////////////////////////////////////
// Cold start after connection establishment processing
////////////////////////////////////
// This Logic should be executed at the transition of the
// producing connection from closed to open.
// For Point-to-point connections this logic may
// be performed any time the connection is Opened or Re-Opened.
////////////////////////////////////

Ping_Interval_EPI_Count = 0,
RR_Con_Num_Index_Pntr = Max_Consumer_Number,

// Initialize the ping count in the safety message to 0
Mode_Octet.Ping_Count = 0,

// Time_Drift_Per_Ping_Interval, the minimum value is 1
Time_Drift_Per_Ping_Interval =
Roundup(EPI × Ping_Interval_EPI_Multiplier / 320000,

FOR (Consumer_Num = 1 to Max_Consumer_Number),
{
    // Producer Dynamic Variables
    Consumer_Active_Idle[Consumer_Num-1] = Idle,
    S_Connection_Fault[Consumer_Num-1] = OK,
    Producer_Rcvd_Time_Value[Consumer_Num-1] = 0x0000,
    Consumer_Time_Correction_Value[Consumer_Num-1] = 0x0000,
    Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1] =
0x0000,
    Consumer_Time_Value[Consumer_Num-1] = 0x0000

    // Producer Derived Variables

    // Time_Drift_Constant, the minimum value is 1.
    (Time_Drift_Constant is not
    // saved but is used in the calculation of the
    // Connection_Correction_Constant below.)
    Time_Drift_Constant =
Roundup((Timeout_Multiplier[Consumer_Num-1] +1)×EPI ×
Ping_Interval_EPI_Multiplier / 320000),

    // Connection_Correction_Constant
    Connection_Correction_Constant[Consumer_Num-1] =
Time_Drift_Constant + 1 - Time_Coord_Msg_Min_Multiplier [Consumer_Num-
1],

    // Time_Coord_Response_EPI_Limit
    Time_Coord_Response_EPI_Limit[Consumer_Num-1]= Roundup((5000000 +
(Time_Coord_Msg_Min_Multiplier[Consumer_Num-1]×128) +
(EPI × ( Consumer_Num - 1))) / EPI),

    // Time_Coord_Response_EPI_Limit has a maximum value of 1000
    IF (Time_Coord_Response_EPI_Limit[Consumer_Num-1] > 1000),
    THEN
    {
        Time_Coord_Response_EPI_Limit[Consumer_Num-1] = 1000,
    }
}

```

```

ENDIF
}
ENDFOR

```

```

////////////////////////////////////
// end, Cold start after connection establishment processing
////////////////////////////////////

```

7.5.5.6 Example safety data production multipoint consumer re-start logic

The logic allows individual multipoint consumers to be stopped and restarted while the multipoint production continues to other consumers.

This subclause defines the logic needed to restart an individual consumer, but it does not represent an efficient implementation of how to detect that the individual connection has been restarted.

```

////////////////////////////////////
// Re-initialization of Multipoint production to individual consumers
////////////////////////////////////
// This Logic should be executed after a multipoint consumer performs
// a successful open to a multipoint producer while the production
// is in process to re-initialize production to that consumer number
////////////////////////////////////
FOR (Consumer_Num = 1 to Max_Consumer_Number),
{
IF (Consumer_Open(Consumer_Num-1)transitions from Closed to Open)
THEN
{
// Producer Dynamic Variables
Consumer_Active_Idle[Consumer_Num-1] = Idle,
S_Connection_Fault[Consumer_Num-1] = OK,
Producer_Rcvd_Time_Value[Consumer_Num-1] = 0x0000,
Consumer_Time_Correction_Value[Consumer_Num-1] = 0x0000,
Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1] = 0x0000,

// Producer Derived Variables
// Time_Drift_Constant, the minimum value is 1
//(Time_Drift_Constant is not
// saved but is used in the calculation of the
// Connection_Correction_Constant below.)
Time_Drift_Constant =
Roundup((Timeout_Multiplier[Consumer_Num-1] +1) × EPI ×
Ping_Interval_EPI_Multiplier / 320000),
// Connection_Correction_Constant
Connection_Correction_Constant[Consumer_Num-1] =
Time_Drift_Constant + 1 - Time_Coord_Msg_Min_Multiplier
[Consumer_Num-1],
// Time_Coord_Response_EPI_Limit
Time_Coord_Response_EPI_Limit[Consumer_Num-1] = Roundup((5000000 +
(Time_Coord_Msg_Min_Multiplier[Consumer_Num-1]×128) +
(EPI × ( Consumer_Num - 2))) / EPI),

// Time_Coord_Response_EPI_Limit has a maximum value of 1000
IF (Time_Coord_Response_EPI_Limit[Consumer_Num-1] > 1000),
THEN
{
Time_Coord_Response_EPI_Limit[Consumer_Num-1] = 1000,

```

```

    }
    ENDIF
}
ENDIF
}

```

7.5.5.7 Example combined data production

The following is the common required logic of the SafetyValidatorClient for all of safety connections.

```

////////////////////////////////////
// start of EPI Safety Data production processing
////////////////////////////////////
// This logic assumes that the application has passed the Safety
// Data and Application_Run_Idle to the Safety Validator.
//
////////////////////////////////////
// Capture the time to be used for the time stamp
Producer_Safe_Data_TS = Producer_Clk_Count,

// Check if it is time to increment the Ping Count
Ping_Interval_EPI_Count = Ping_Interval_EPI_Count + 1,

IF (Ping_Interval_EPI_Count >= Ping_Interval_EPI_Multiplier),
THEN
{
Increment Mode_Octet.Ping_Count,
Ping_Interval_EPI_Count = 0,
}
ENDIF

// Set the remainder of the Mode_Octet bits
Mode_Octet.Run_Idle = Application_Run_Idle,
Mode_Octet.TBD_Bit = 0,
Mode_Octet.TBD_2_Bit = 0,

// Execute the safety producer, connection type specific functions
IF (Connection_Type == multipoint),
THEN
{
multi_cast_producer_function(),
}
ELSE
{
single_cast_producer_function(),
}
ENDIF

// set Mode_Octet redundant bits (2:4) to the correct values
Mode_Octet = (Mode_Octet AND 0xe3) OR ((Mode_Octet >> 3) AND 0x1c XOR
0x14),
Calculate Data Section CRC,
Calculate Time_Stamp Section CRC,
Trigger the sending of the Data Message,

// If on FSCP 2 on CP 2/3 and Multipoint, send the Time Correction
// Message if it is time.
IF ((Connection_Type == CP 2/3) AND
(Send_Time_Correction_Message == 1)),

```

```

THEN
{
Trigger the sending of the Time Correction Message,
Send_Time_Correction_Message = 0,
}
ENDIF
/////////////////////////////////////////////////////////////////
// end of safety data production processing
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
// Start of point-to-point production function
/////////////////////////////////////////////////////////////////
single_cast_producer_function()
// Check if a Time Coordination message has been received in
// the allotted time.
IF (Ping_Interval_EPI_Count == 8),
THEN
{
Increment Ping_Int_Since_Last_Time_Coord_Msg_Count[0],
IF (Ping_Int_Since_Last_Time_Coord_Msg_Count[0]
>= (Timeout_Multiplier[0] + 2)),
THEN
{
S_Connection_Fault[0]=Faulted,
}
ENDIF
}
ENDIF

//Hold Data_Time_Stamp = 0 until time coordination msg received
IF(Consumer_Active_Idle[0] == Idle),
THEN
{
Mode_Octet.Run_Idle = Idle,
Time_Stamp_Section.Data_Time_Stamp = 0x0000,
}
ELSE
{
Time_Stamp_Section.Data_Time_Stamp = Producer_Safe_Data_TS +
Consumer_Time_Correction_Value[0],
}
ENDIF

END single_cast_producer_function()
/////////////////////////////////////////////////////////////////
// end of point-to-point processing
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
// Start of multipoint production function
/////////////////////////////////////////////////////////////////
multi_cast_producer_function()
// Check if its time to start sending Time Correction messages
// Time Correction messages are sent starting at the 8th EPI
production
// within the Ping interval
IF (Ping_Interval_EPI_Count == 8),
THEN
{

```

```

RR_Con_Num_Index_Pntr = 0,
}
ENDIF

// Check if a Time Coordination message has been received in
// the allotted time.
IF (RR_Con_Num_Index_Pntr < Max_Consumer_Number),
THEN
{
Increment
Ping_Int_Since_Last_Time_Coord_Msg_Count[RR_Con_Num_Index_Pntr],
IF (Ping_Int_Since_Last_Time_Coord_Msg_Count[RR_Con_Num_Index_Pntr]
    >= (Timeout_Multiplier[RR_Con_Num_Index_Pntr] + 2)),
THEN
{
    Set S_Connection_Fault[RR_Con_Num_Index_Pntr]=Faulted,
}
ENDIF
ENDIF

// Check if its time to send an active non-error Time Correction
// message
IF ((RR_Con_Num_Index_Pntr < Max_Consumer_Number) AND
    (Consumer_Active_Idle[RR_Con_Num_Index_Pntr] == Active)), AND
    (Consumer_Open[RR_Con_Num_Index_Pntr] == Open) AND
    (S_Connection_Fault[RR_Con_Num_Index_Pntr] == OK)
THEN
{
Send_Time_Correction_Message = 1,
Mcast_Octet.Consumer_# = RR_Con_Num_Index_Pntr + 1,
Mcast_Octet.Multi_Cast_Active_Idle =
Consumer_Active_Idle[RR_Con_Num_Index_Pntr],
Time_Correction_Section.Time_Correction_Value =
Consumer_Time_Correction_Value[RR_Con_Num_Index_Pntr],
Set Mcast_Octet.Parity_Even to the correct value,
Mcast_Octet_2 = ((Mcast_Octet XOR 0xFF) AND 0x55) OR
(Mcast_Octet AND 0xAA)
Calculate CRC of time correction message,
}
ELSE
{
//load null TIME CORRECTION Section,
Mcast_Octet.Consumer_# = 0,
Mcast_Octet.Multi_Cast_Active_Idle = Idle,
Time_Correction_Section.Time_Correction_Value = 0,
Set Mcast_Octet.Parity_Even to the correct value,
Mcast_Octet_2 = ((Mcast_Octet XOR 0xFF) AND 0x55) OR
(Mcast_Octet AND 0xAA)
Calculate CRC of time correction section,
}
ENDIF

// If Time Correction messages are being sent, increment to the
// next consumer
IF (RR_Con_Num_Index_Pntr < Max_Consumer_Number),
THEN
{
RR_Con_Num_Index_Pntr = RR_Con_Num_Index_Pntr + 1,
}
ENDIF

// Set the time stamp value
Time_Stamp_Section.Data_Time_Stamp = Producer_Safe_Data_TS,

```

```
END multi_cast_producer_function()
////////////////////////////////////
// end of multipoint processing
////////////////////////////////////
```

7.5.5.8 Example time coordination message reception logic

The SafetyValidatorClient shall complete the following steps for each time coordination message received. For point-to-point and multipoint safety connections, a Consumer_Time_Value is received in the time coordination message.

Producers shall allow Time Coordination responses to arrive during the Ping_Interval or Ping_Interval +1.

```
////////////////////////////////////
//Time Coordination message reception processing
////////////////////////////////////
// Consumer_Num equals 1 for point-to-point
// Consumer_Num equals the consumer the message
// was received from for multipoint
////////////////////////////////////
// This Logic should only be executed If Ack_Octet.Ping_Response
// of the message is equal to 1 AND Consumer_Open[Consumer_Num-1]
// is equal to Open. If Ack_Octet.Ping_Response is equal to 0 OR
// Consumer_Open[Consumer_Num-1] is equal to Closed, this
// message should be ignored.
////////////////////////////////////
// reject Time Coordination packets if CONSUMER_TIMESTAMP has
// previously been captured AND this is not the first time coordination
// message from this consumer
IF ( (Consumer_Time_Value[Consumer_Num-1] ==
Time_Coordination_Section.Consumer_Time_Value) AND
(Consumer_Active_Idle[Consumer_Num-1] == Active) )
THEN
{
    // the only time this would be the case is if that Time
    // Coordination was already received
    Return
}
ELSE
{
    // Capture the received CONSUMER time
    Consumer_Time_Value[Consumer_Num-1] =
Time_Coordination_Section.Consumer_Time_Value;
}
ENDIF

// Capture the received time
Producer_Rcvd_Time_Value[Consumer_Num-1] = Producer_Clk_Count,
}
ENDIF

// Perform the integrity checks
Check the CRC of the Time Coordination Section,
IF ((Ack_Octet.parity incorrect) OR (Ack_Octet_2 !=
(((Ack_Octet XOR 0xFF) AND 0x55) OR (Ack_Octet AND 0xAA))))
THEN
{
    Ack_Octet_Error = true,
}
ELSE
```



```

{
Ack_Octet_Error = false,
}
ENDIF
IF ((CRC_Error) OR (Ack_Octet_Error = true) OR
// Ensure that the Time Coordination message returned
// with the same ping interval or the next ping interval.
// The next ping interval is allowed for the case that a
// multipoint consumer connects to an existing producer.
// If the new consumer receives its first message near the end
// of the ping interval, the time coordination may arrive back
// at the producer during the next ping interval.
    ((Ack_Octet.Ping_Count_Reply != Ping_Count) AND
    (Ack_Octet.Ping_Count_Reply != Ping_Count-1)) OR
// Ensure that the Time Coordination message returned
// within the approximately 5 second limit
((Consumer_Active_Idle[Consumer_Num-1] == Active) AND
(Ping_Interval_EPI_Count >
    Time_Coord_Response_EPI_Limit[Consumer_Num-1])),
THEN
{
S_Connection_Fault[Consumer_Num-1] = Faulted,
}
ENDIF

// Determine the worst case Consumer_Time_Correction_Value based on
the
// received Time_Coordination section
Worst_Case_Consumer_Time_Correction_Value[Consumer_Num-1] =
    Time_Coordination_Section.Consumer_Time_Value
    - Producer_Rcvd_Time_Value[Consumer_Num-1]
- Connection_Correction_Constant[Consumer_Num-1],

// Determine the worst case time drift since the last
// Time Coordination information was received. This is equal to the
// number of Ping Intervals since the last Time Coordination message
// times the Time Drift per Ping Interval plus 1 for the asynchronous
// clocks. This value will not be used until the second
// Time_Coordination message is received.

Time_Drift_Since_Last_Time_Coord =
({Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1]+ 1}
    × Time_Drift_Per_Ping_Interval) + 1,

// If this Time Coordination Transport delay is greater than the
// last Time_Coordination Transport delay by more than the
appropriate
// number of Time_Drift_Per_Ping_Intervals, then, use the previous
// Correction value minus the appropriate number of
// Time_Drift_Per_Ping_Intervals. This action is not taken until the
// second Time_Coordination message is received.
// The math for this check should be 16 bit or greater
IF ((Consumer_Active_Idle[Consumer_Num-1] = Active) AND
((Consumer_Time_Correction_Value[Consumer_Num-1]-
    Worst_Case_Consumer_Time_Correction_Value[Consumer_Num-1] -
    Time_Drift_Since_Last_Time_Coord) AND 0x8000 == 0)),
THEN
{
Consumer_Time_Correction_Value[Consumer_Num-1] =
Consumer_Time_Correction_Value[Consumer_Num-1]
Time_Drift_Since_Last_Time_Coord,
}
ELSE

```

```
{
Consumer_Time_Correction_Value[Consumer_Num-1] =
Worst_Case_Consumer_Time_Correction_Value[Consumer_Num-1],
}
ENDIF

// Set the flag indicating a Time_Coordination message has been
received
Consumer_Active_Idle[Consumer_Num-1] = Active,

// Reset the Time_Coordination message timer
Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1] = 0,
////////////////////////////////////
// end producer time coordination information reception processing
////////////////////////////////////
```

7.5.6 SafetyValidatorServer function definition

7.5.6.1 General

The SafetyValidatorServer function is the safety layer that augments a standard connection on CP 2/3 or CP 2/2 to transmit SIL3 data. It coordinates the reception of safety data with a peer instance of the SafetyValidatorClient class and delivers this data to the consuming safety-related application.

7.5.6.2 Safety consumption

This subclause describes the consumption of data and the interfaces between the application, the SafetyValidatorServer and the underlying link layer.

There are two versions of safety data consumer monitoring. They are:

- SafetyValidatorServer - Link Triggered.
This is appropriate for consuming applications that are synchronized to the reception of the data, or consuming applications that present a continuous delay from the reception of data.
- SafetyValidatorServer - Application Triggered.
This is appropriate for consuming applications that are periodic in nature and not synchronized to the reception of the data.

In both versions, the SafetyValidatorServer shall perform the following aspects of the safety data monitoring:

- Time coordination with the producer;
- Time stamp section integrity checking;
- Data integrity checking;
- Network time expectation checking.

7.5.6.3 SafetyValidatorServer - link triggered

Figure 47 shows the data flows for the SafetyValidatorServer in the link triggered mode.

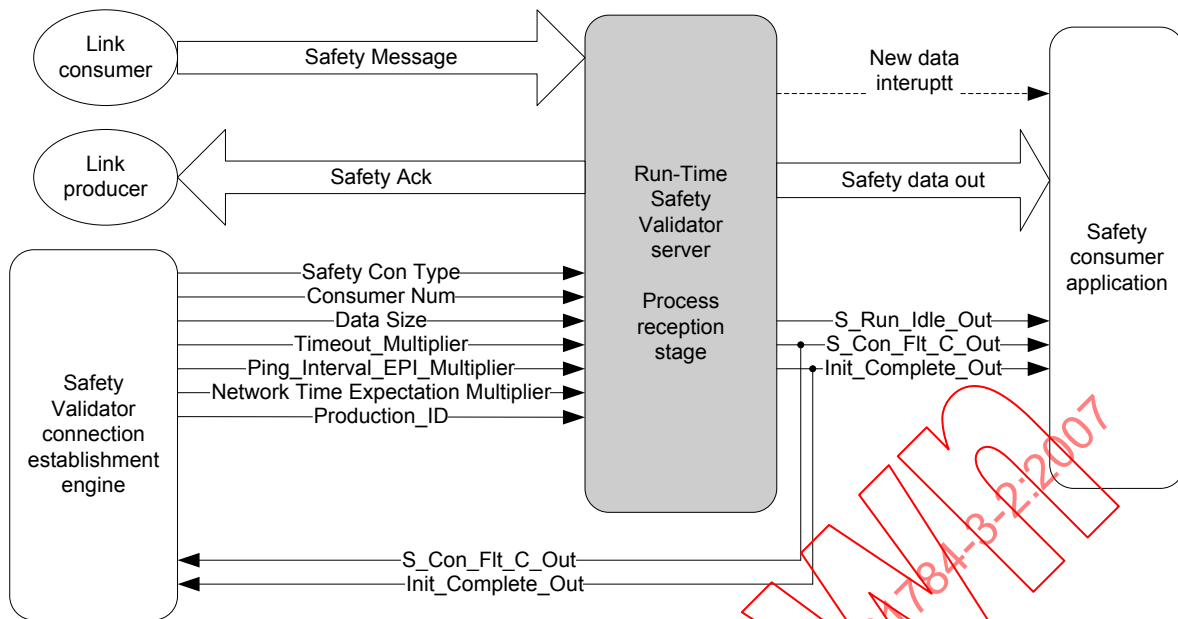


Figure 47 – Consumer safety data monitoring

The Link Triggered SafetyValidatorServer shall perform all aspects of the safety protocol on each message received.

For multipoint, the Link Triggered SafetyValidatorServer shall also perform all aspects of the safety protocol on each Time_Correction section received.

The SafetyValidatorServer shall insure that the safety data presented to the consuming application has been crosschecked and that the network time expectation is being met.

The Link Triggered SafetyValidatorServer shall set the new data flag each time data is ready for consumption.

7.5.6.4 SafetyValidatorServer - application triggered

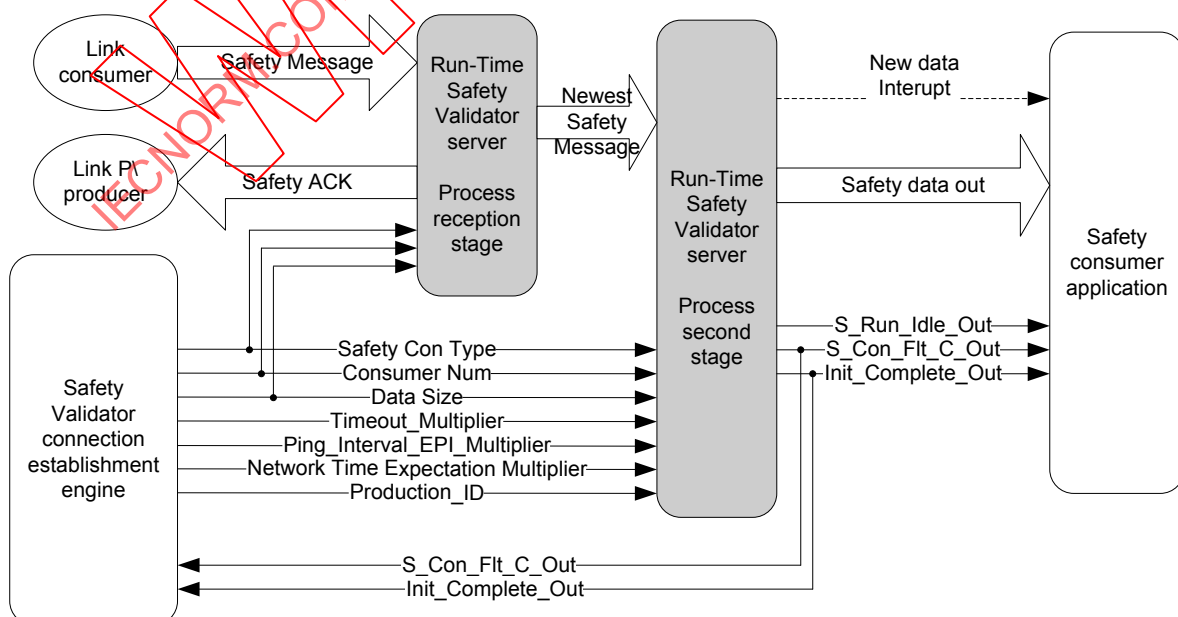


Figure 48 – SafetyValidatorServer - application triggered

The Application Triggered SafetyValidatorServer does not fully process each message.

The process reception stage of the application triggered SafetyValidatorServer shall check the ping count for all incoming data messages.

For multipoint consumption in application triggered SafetyValidatorServers, the latest Time_Correction section received shall be forwarded to the consuming application.

The consuming safety-related application shall request new data from the SafetyValidatorServer in order to cause the SafetyValidatorServer to complete processing of a message.

For periodic applications that are not synchronized to the reception of the data, this reduces the processing burden, since the data section is only crosschecked (in Process Second Stage in Figure 48 for the safety data messages that are used. This is particularly useful for the case where the network EPIs are at a faster rate than the consuming application's periodic rate.

The consuming application shall ensure that it gets new data at least once every 2 s to ensure that the time stamp does not rollover.

7.5.6.5 Safety data reception logic

The safety data reception processing consists of:

- Ping_Count check,
- Time Stamp section checking,
- Data integrity checking,
- Time stamp checking, and
- Deriving Run_Idle and fault status.

The definitions of the variables used in the safety data reception logic can be found in 7.6.8

Table 80 defines the safety data processing steps that are performed by the Link Triggered SafetyValidatorServer.

Table 80 – Data reception - Link triggered

	Point-to-point	Multipoint
Check Ping_Response	No	No
Check Ping_Count	Yes	Yes
Check Time stamp section CRC	Yes	Yes
Check Run_Idle	Yes	Yes
Check TBD_Bit	Yes	Yes
Check Data CRCs	Yes	Yes
Check Data Actual/Complement	Yes	Yes
Derive Data_Time_Stamp	Yes	Yes

Table 81 defines the safety data processing steps that shall be performed by the Link Triggered SafetyValidatorServer for the reception of each Time Correction section.

Table 81 – Time_Correction reception - Link triggered

Multipoint	Multipoint
Check MCast_Octet and MCast_Octet_2	Yes, if consumer_# match
Check Time_Correction CRC	Yes, if consumer_# match
Get Time_Correction_Value	Yes, if consumer_# match
Check Multi_Cast_Active_Idle	Yes, if consumer_# match

Table 82 defines the safety data processing steps that are performed on each message reception by the application triggered SafetyValidatorServer.

Table 82 – Data reception - Application triggered

	Point-to-point	Multipoint
Check Ping_Response	No	No
Check Ping_Count	Yes	Yes
Check Time stamp section CRC	No	No
Check Run_Idle	No	No
Check TBD_Bit	No	No
Check Data CRCs	No	No
Check Data Actual/Complement	No	No
Derive Data_Time_Stamp	No	No

Table 83 defines the safety data processing steps that shall be performed on each Time_Correction section reception by the application triggered SafetyValidatorServers.

Table 83 – Time_Correction reception - Application triggered

	Multipoint
Check MCast_Octet and MCast_Octet_2	Forward message if consumer_# match
Check Time_Correction CRC	No
Get Time_Correction_Value	no
Check Multi_Cast_Active_Idle	No

Table 84 defines the safety data processing steps that are performed by the application triggered SafetyValidatorServer when safety data is requested.

Table 84 – Consuming application – Safety data monitoring

	Point-to-point	Multipoint
Check Ping_Response	NA	NA
Check Ping_Count	NA	NA
Check Time_stamp section CRC	Yes	Yes
Check MCast_Octet and MCast_Octet_2	NA	Yes (from last time correction)
Check Time_Correction CRC	NA	Yes (from last time correction)
Get Time_Correction_Value	NA	Yes (from last time correction)
Check Multi_Cast_Active_Idle	NA	Yes (from last time correction)
Check Run_Idle	Yes	Yes
Check TBD_Bit	Yes	Yes
Check Data CRCs	Yes	Yes
Check Data Actual/Complement	Yes	Yes
Derive Data_Time_Stamp	Yes	Yes

7.5.6.6 Ping count checking

The SafetyValidatorServer shall perform Ping_Count checking on every message. Ping_Count checking checks the ping count to see if it has changed.

If a SafetyValidatorServer detects that the ping count has changed, it shall produce a time coordination message within that Ping Interval or within 5 s, whichever is less.

Ping_Count checking errors should not be treated as dangerous failures. The time stamp section CRC does not need to be checked to perform the Ping_Count checking.

7.5.6.7 Data and network time expectation checking maximum interval

The data integrity shall be checked at least once every 2 s.

The Time Stamp and Network Time Expectation (NTE) shall be checked at least once every 2 s.

7.5.6.8 Example cold start initialization

```

////////////////////////////////////
// Cold start after connection establishment processing
////////////////////////////////////
// This Logic should be executed at the transition of the
// consuming connection from closed to open.
////////////////////////////////////

```

```

S_Con_Flt_C_Out = OK,
Init_Complete_Out = 0,
S_Run_Idle_Out = Idle,

```

```

Last_Ping_Count = 3,
Time_Coordination_Count_Down = 0x0000,

```

```

Corrected_Data_Time_Stamp = 0x0000,
Last_Data_Time_Stamp = 0x0000,
Last_Rcvd_Multi_Cast_Active_Idle = Idle,
Last_Rcvd_Time_Correction_Value = 0x0000,
Time_Correction_Ping_Interval_Count = 0x0000,

```

```

Time_Correction_Received_Flag = 0,
Data_Age = 0x0000,
Max_Data_Age = 0x0000,
////////////////////////////////////
// end, Cold start after connection establishment processing
////////////////////////////////////

```

7.5.6.9 SafetyValidatorServer - link triggered logic

Subclauses 7.5.6.10 and 7.5.6.11 describe the consumer safety data reception logic for the SafetyValidatorServer - Link Triggered option. The term Link Triggered means that all the logic is executed on every reception.

Subclause 7.5.6.10 describes the combined logic associated with the Safety data packet reception for the point-to-point and multipoint safety connection types.

Subclause 7.5.6.11 describes the logic associated with the Time Correction Message reception for the multipoint safety connection type.

This logic shall only be executed if the connection is open with no errors.

7.5.6.10 Example combined reception logic - link triggered logic

The following is the required SafetyValidatorServer reception logic for the safety data packet reception of the point-to-point or multipoint safety connection type, for the SafetyValidatorServer – Link Triggered Logic.

Multipoint connections shall send a Time Coordination Message (Consumer_Num – 1) receptions after the ping count change.

Multipoint-consumers connecting to an existing producer shall send the Time_Coordination message immediately upon the first valid reception.

```

////////////////////////////////////
// Safety Data Consumption - link triggered logic
////////////////////////////////////

// Check if its time to send a Time Coordination Message
// Multipoint connections send a Time Coordination Message
// (Consumer_Num - 1) receptions after the ping count change.
IF (received_Mode_Octet.Ping_Count != Last_Ping_Count),
THEN
{
    Time_Coordination_Count_Down = Consumer_Num,
    Increment Time_Correction_Ping_Interval_Count,
}
ENDIF

// For multipoint-consumers connecting to an existing producer,
send
// the Time_Coordination message immediately upon the 1st reception.
IF (It is the first data reception),
THEN
{
    Produce Time_Coordination_Message,
    Time_Coordination_Count_Down = 0,
}
ENDIF
// Check if a ping count change has been detected and the Time

```

```
// Coordination message has not been sent yet.
IF (Time_Coordination_Count_Down > 0),
// Send the Time_Coordination_message if the decremented
// Time_Coordination_Count_Down transitions to 0
THEN
{
decrement Time_Coordination_Count_Down,
IF (Time_Coordination_Count_Down == 0),
THEN
{
Produce Time_Coordination_Message,
}
ENDIF
}
// Save ping count for next reception check
Last_Ping_Count = Mode_Octet.Ping_Count,

// check for data integrity faults
Temp_Fault_Flag = OK,
IF ((Time_Stamp_Section_CRC != OK) OR (Actual_Data_Section_CRC != OK)
OR (Complement_Data_Section_CRC != OK)
OR (Mode_Octet.Run_Idle != not Mode_Octet.N_Run_Idle)
OR (Mode_Octet.TBD_2_Bit !=
Mode_Octet.TBD_2_Bit_Copy)
OR (Mode_Octet.TBD_Bit != not Mode_Octet.N_TBD_Bit)
OR (Actual_Data != not Complement_Data) [for > 2 octets only]
),
THEN
{
Temp_Fault_Flag = Faulted,
}
ENDIF

// execute the safety connection type specific functions
IF (Connection_Type == Multipoint),
THEN
{
multi_point_consumer_function(),
}
ELSE
{
single_point_consumer_function(),
}
ENDIF

// Set up Time_Stamp_Delta, to be used to check for out of sequence
// messages Time_Stamp_Delta is not calculated until the EPI
// reception after Init_Complete_Out is set to 1.
IF(Init_Complete_Out == 0),
THEN
{
Time_Stamp_Delta = 1,
}
ELSE
{
Time_Stamp_Delta = Time_Stamp_Section.Data_Time_Stamp -
Last_Data_Time_Stamp,
}
ENDIF

// Save Time stamp for next reception sequence check
Last_Data_Time_Stamp = Time_Stamp_Section.Data_Time_Stamp,
```



```

// Calculate the worst case age
IF (Init_Complete_Out_Temp == 1),
THEN
{
    Data_Age = Consumer_Clk_Count - Corrected_Data_Time_Stamp,
}
ENDIF

// Calculate the maximum worst case age
IF (Data_Age > Max_Data_Age)
THEN
{
    Max_Data_Age = Data_Age,
}
ENDIF

// Check for Age, Integrity, and out of sequence faults
IF (( unsigned_int_16( Data_Age > Network_Time_Expectation_Multiplier)
AND (Init_Complete_Out_Temp == 1))
OR (Temp_Fault_Flag == Faulted)
OR (Time_Stamp_Delta < 0)
OR (Time_Stamp_Delta > Network_Time_Expectation_Multiplier))
THEN
{
    S_Con_Flt_C_Out_Temp = Faulted,
}
ENDIF

////////////////////////////////////
// The following information is passed on to the consuming application
//   with block integrity: Safety
Data, Init_Complete_Out, S_Run_Idle_Out
//   and S_Con_Flt_C_Out.
// The time between the checking of the age of the data and passing
// the data off to the consuming application shall be protected under
// the time stamp check of the consuming application time checks.
////////////////////////////////////
Make Safety Data Available to Consuming Application
Init_Complete_Out = Init_Complete_Out_Temp,
S_Run_Idle_Out    = S_Run_Idle_Out_Temp,
S_Con_Flt_C_Out   = S_Con_Flt_C_Out_Temp,
////////////////////////////////////
// end Safety Data Consumption - link triggered logic
////////////////////////////////////

////////////////////////////////////
// Start of point-to-point consumer function
////////////////////////////////////
single_cast_consumer_function()
// For point-to-point multipoint, no time stamp correction is required
Corrected_Data_Time_Stamp = Time_Stamp_Section.Data_Time_Stamp,

// Determine the Init_Complete_Out state to be passed on to the
// consuming application.
// For point-to-point or bi-dir, a time_stamp other than 0 or a Run
// indication, indicate that the Time_Coordination_Message has been
// received by the Client and the corrected time stamp is being used.
IF ((Corrected_Data_Time_Stamp != 0) OR (Mode_Octet.Run_Idle == Run)),
THEN
{
    Init_Complete_Out_Temp = 1,
}
ENDIF

```

```
// Determine the Run_Idle indication to be passed on to the consuming
// application.
IF ((Mode_Octet.Run_Idle == Idle) OR (Init_Complete_Out_Temp == 0))
THEN
{
    S_Run_Idle_Out_Temp = Idle,
}
ELSE
{
    S_Run_Idle_Out_Temp = Run,
}
ENDIF

END single_cast_consumer_function()
////////////////////////////////////
// end of point-to-point processing
////////////////////////////////////

////////////////////////////////////
// Start of multipoint consumer function
////////////////////////////////////
multi_cast_consumer_function()
// If multipoint, check that a time correction message has been
// received within the last Timeout_Multiplier + 1 Ping Intervals
IF (Time_Correction_Ping_Interval_Count > (Timeout_Multiplier + 1))
THEN
{
    Temp_Fault_Flag = Faulted,
}
ENDIF

// If multipoint, add correction value to time stamp
Corrected_Data_Time_Stamp = Time_Stamp_Section.Data_Time_Stamp +
    Last_Rcvd_Time_Correction_Value
// Determine the Init_Complete_Out state to be passed on to the
// consuming application.
// For multipoint, the reception of a Time_Coorection Message,
// indicate
// that data may be used.
IF (Time_Correction_Received_Flag == 1)
THEN
{
    Init_Complete_Out_Temp = 1,
}
ENDIF
// Determine the Run_Idle indication to be passed on to the consuming
// application.
IF ((Mode_Octet.Run_Idle == Idle) OR
    (Init_Complete_Out_Temp == 0) OR
    (Time_Correction_Received_Flag == 0) OR    //used only on
multipoint
    (Last_Rcvd_Multi_Point_Active_Idle==Idle))//used only on
multipoint
THEN
{
    S_Run_Idle_Out_Temp = Idle,
}
ELSE
{
    S_Run_Idle_Out_Temp = Run,
}
}
```

```

ENDIF
end multi_point_consumer_function()
/////////////////////////////////////////////////////////////////
// end of multipoint processing
/////////////////////////////////////////////////////////////////

```

7.5.6.11 Example time correction message reception - link triggered logic

The following is the required SafetyValidatorServer logic associated with the Time Correction Message reception for the multipoint safety connection type, for the Consumer – Link Triggered.

```

/////////////////////////////////////////////////////////////////
// Consumer processing - Time Correction Message reception
/////////////////////////////////////////////////////////////////
// Is the Time Correction message for this Consumer ?
IF (MCast_Octet.Consumer_Num = Consumer_Num)
    // Perform integrity checks.
    THEN
    {
        IF (Time_Coordination_message_CRC_incorrect)
        OR (MCast_Octet_parity_incorrect)
        OR {MCast_Octet_2 != ((MCast_Octet XOR 0xFF) AND 0x55)
            OR (MCast_Octet AND 0xAA)}
            OR {MCast_Octet.Multi_Cast_Active_Idle == Idle}
        AND (Last_Rcvd_Multi_Cast_Active_Idle == Active)},
        // only one transition from connection idle to active will
        // be allowed, to prevent an erroneous idle indication,
        // a transition from active to idle will be latched into
        // a fault state
        THEN
        {
            S_Con_Flt_C_Out = Faulted,
        }
    }
ENDIF

    // Save the Time_Correction_Value.
    Last_Rcvd_Time_Correction_Value =
    Time_Correction_Section.Time_Correction_Value,

    // Save the Multi_Cast_Active_Idle bit.
    Last_Rcvd_Multi_Cast_Active_Idle =
    MCast_Octet.Multi_Cast_Active_Idle,

    // Set the Time_Correction_Received_Flag.
    Time_Correction_Received_Flag = 1,

    // Reset the Time_Correction_timeout_counter
    Time_Correction_Ping_Interval_Count = 0,
ENDIF
/////////////////////////////////////////////////////////////////
// end - Time Correction Message reception
/////////////////////////////////////////////////////////////////

```

7.5.6.12 SafetyValidatorServer - application triggered logic

For SafetyValidatorServer - Application Triggered, the same logic described above shall be performed by the combination of the SafetyValidatorServer, but only when requested by the consuming application. Application Triggered implementations only process the safety packets when needed by the application. This implementation strategy is useful for applications that usually run slower than a typical EPI rate. The functions can be divided as defined in Table 81 and Table 84.

The protocol processing steps defined in Table 84 only need to be performed on the messages used by the consuming application.

7.5.6.13 Example time coordination message production logic

The SafetyValidatorServer shall perform the following steps each time a time coordination message is sent. For point-to-point and multipoint safety connections, time coordination data is sent in the time coordination message. This logic is also described in the single_cast_function() of the combined production logic section.

```

////////////////////////////////////
//Time Coordination message production processing
////////////////////////////////////
// Set the Ping_Count_Reply to the value captured at the ping request
Ack_Octet.Ping_Count_Reply = Mode_Octet.Ping_Count,

// Set the Ping_Response bit to 1
Ack_Octet.Ping_Response = 1,

// Set the Consumer_Time_Value to the producers clock value
Time_Coordination_section.Consumer_Time_Value = Producer_Clk_Count,

// Add data integrity measures
Set Ack_Octet.Reserved(6:4) = 0,
Set Ack_Octet.Reserved(2) = 0,
Set Ack_Octet.Parity_Even bit to the correct value,
Ack_Octet_2 !=(((Ack_Octet XOR 0xFF)AND 0x55)OR
               (Ack_Octet AND 0xAA)),
Calculate CRC on Time Coordination message,
////////////////////////////////////
// end time coordination message send logic
////////////////////////////////////

```

7.6 Safety message and protocol data specifications

7.6.1 Mode octet

7.6.1.1 Mode_Octet.Run_Idle

The Run_Idle bit value of 0 shall indicate Idle, and the Run_Idle bit value of 1 shall indicate Run.

For point-to-point connections, the Run_Idle bit shall be set to idle if the producing application is Idle or if the producer is waiting for Time Coordination Message information to be received at the initiation of data production.

If Time Coordination Message information has been received, the Run_Idle bit shall be set to the Run_Idle status indicated by the producing application.

For multipoint connections, the Run_Idle bit shall be set to the Run_Idle status indicated by the producing application.

7.6.1.2 Mode_Octet.N_Run_Idle

The N_Run_Idle complement bit shall always be the complement of the Run_Idle bit.

7.6.1.3 Mode_Octet TBD_2_Bit

The Mode_Octet Reserved TBD_2_bit shall be set to 0 by the producer.

7.6.1.4 Mode_Octet.TBD_2_Copy

The TBD_2_Copy bit shall always equal to the TBD_2 bit.

7.6.1.5 Mode_Octet.Ping_Count

The Ping_Count is used by the consumer to know when it should send a Time Coordination Message.

7.6.1.6 Mode_Octet.TBD_Bit

The Mode_Octet Reserved TBD_bit shall be set to 0 by the producer. The Reserved bits shall be ignored by the consumer except for CRC checking and Actual/Complement checking of the TBD_Bit/N_TBD_Bit pair.

7.6.1.7 Mode_Octet.N_TBD_Bit

The N_TBD_Bit complement bit shall always be the complement of the TBD_Bit.

7.6.2 Time Stamp Section

Time_Stamp_Section.Data_Time_Stamp: the Data_Time_Stamp is the time that the Data was sampled for this production.

For point-to-point, the Data_Time_Stamp is relative to the timer that is on the consumer. For multipoint, the Data_Time_Stamp is relative to the timer that is on the producer.

The Data_Time_Stamp is always in 128 μ s increments.

7.6.3 Time Coordination Message

7.6.3.1 Ack_Octet.Ping_Response bit

The Ping_Response bit shall indicate that a ping response is being sent on this data production from the safety data consumer on this node.

If the Ping_Response bit is 1, it shall indicate that the Consumer_Time_Value, and Ping_Count_Reply, are valid on this production.

If the Ping_Response bit is 0, it shall indicate that the Consumer_Time_Value, and Ping_Count_Reply, are not valid on this production and should be ignored.

7.6.3.2 Ack_Octet.Consumer_Time_Value

The Consumer_Time_Value shall be the safety data consumer's clock value at the time that the message is sent.

The Consumer Time Value is used to derive the Data_Time_Stamp for the other directions data production.

7.6.3.3 Ack_Octet.Ping_Count_Reply

The Ping_Count_Reply is used by each consumer to indicate the ping request that is being responded to.

The Ping_Count_Reply is equal to the value of the Ping_Count that is being responded to.

7.6.3.4 Ack_Octet.Time Coordination Reserved bits

The Time Coordination Reserved bits shall be set to 0 by the consumer.

The Reserved bits shall be ignored by the producer except for CRC and Ack_Octet_2 checking.

7.6.4 Time correction message

7.6.4.1 Mcast_Octet.Consumer_#

For multipoint produced data the Multi_Cast_Active_Idle, and Consumer_Time_Correction_Value shall be directed to the consumer indicated by the Consumer_# field of the message.

The Consumer_# points to a specific consumer (1 through 15).

The Multi_Cast_Active_Idle, and the Consumer_Time_Correction_Value shall be applied by the consumer indicated by the Consumer_#.

A Consumer_# of 0 shall be used to indicate Multi_Cast_Active_Idle, and the Consumer_Time_Correction_Value are not being transferred if needed.

7.6.4.2 Time_Correction_Section.Consumer_Time_Correction_Value

The Consumer_Time_Correction_Value is used by the multipoint consumer to correct the Producer_Time_Stamp so the resultant time stamp is relative to the consumers' clock.

7.6.4.3 Mcast_Octet.Multi_Cast_Active_Idle

The Multi_Cast_Active_Idle bit value of 0 shall indicate Idle, and the Multi_Cast_Active_Idle bit value of 1 shall indicate Active.

Once the Multi_Cast_Active_Idle bit has been set to Active after 1st data production, this bit shall not be set back to idle until the safety connection is re-initialized.

If the consumer sees the Multipoint_Active_Idle bit transition from Active to Idle, it shall stay in the idle state until the safety connection has been re-initialized.

For multipoint connections, the Run_Idle bit of the Mode_Octet shall be set to the Run_Idle status indicated by the producing application.

The consumer will need to look at both the Run_Idle bit and the Multi_Cast_Active_Idle bit.

7.6.4.4 Mcast_octet.time correction reserved bits

The Time Correction Reserved bits shall be set to 0 by the producer.

The Reserved bits shall be ignored by the consumer except for CRC checking and MC_Octet_2 checking.

7.6.5 Safety data production

7.6.5.1 Producing connection status

The producer connection status shall be determined by the combination of the S_Connection_Fault and the Consumer_Active_Idle flags (see Table 85 for definition).

Table 85 – Producer connection status determination

S_Connection_Fault	Consumer_Active_Idle	Producer Connection status	Comment
Fault	Don't Care	Status = Faulted.	Restart will be required
OK	Idle	Status = Idle	The safety connection has not been activated
OK	Active	Status = Active	The safety connection is active

7.6.5.2 Consumer_Open

For Safety Data producers, Consumer_Open from the Validator Connection Establishment Engine to the Run Time Validator shall be provided to indicate whether or not a particular consumer of the produced data has an active open connection.

If the producer is multipoint, Consumer_Open shall exist for each multipoint consumer, 1 through Max_Consumer_Number (indexed from 0 to Max_Consumer_Number – 1).

The Consumer_Open values are Open and Closed.

The Consumer_Open indication shall act as an enable to the Run Time Validator on a connection by connection basis. After successful connection establishment, Consumer_Open flags shall transition from Closed to Open.

The Validator Connection Establishment engine shall provide an array of consumer connection status resources for multipoint producers that shall be allocated when a consumer makes a connection.

The consumer numbers allocated to a Run-time Validator shall be the number sent to the consumer in the connection establishment reply.

The consumer connection numbers can be treated as an allocation pool that can be re-used and re-allocated as needed.

The Validator Connection Establishment Engine shall ensure that the Consumer_Open resource for any allocated consumer number is set to Closed for a minimum time of $(\text{Timeout\_Multiplier} + 2) \times \text{Ping Interval}$ prior to re-allocating that Consumer number to another connection. This is to ensure that any previously established connections have timed out prior to allocating the number to a new connection.

In other words, if the Validator Connection Establishment Engine sees S_Connection_Fault set, it shall set Consumer_Active to Closed for a minimum of $(\text{Timeout\_Multiplier} + 2) \times \text{Ping Interval}$ or optionally until that same consumer tries to re-establish the same connection.

7.6.5.3 Application_Run_Idle

If the producing application is actively controlling data the Run_Idle indication shall indicate Run. If the producing application is not actively controlling the data and the data should be put in the safety state, the Run_Idle indication shall indicate Idle.

7.6.5.4 Consumer_Active_Idle

Consumer_Active_Idle is a run time flag to indicating if valid Time Coordination Message information has been received for this consumer.

If a safety data producer is producing data and valid Time Coordination information has been received without errors the Consumer_Active_Idle flag shall be equal to Active, otherwise it will be Idle.

For multipoint producers, a Consumer_Active_Idle flag shall exist for each multipoint consumer, numbered 1 through Max_Consumer_Number.

The Consumer_Active_Idle values are idle and active.

At connection establishment, all Consumer_Active_Idle flags shall be initialized to Idle.

7.6.5.5 S_Connection_Fault

S_Connection_Fault is a run time flag that indicates whether the safety connection to the consumer is OK or Faulted.

An S_Connection_Fault indication of Fault, shall indicate that the producer has detected a problem with the safety connection to this consumer or the consumer has indicated an error back to the producer.

If the safety data producer has received invalid Time Coordination information, or a Time Coordination information timeout, the S_Connection_Fault flag shall be equal to Fault, otherwise it will be OK.

For multipoint producers, an S_Connection_Fault flag shall exist for each multipoint consumer, numbered 1 through Max_Consumer_Number.

The S_Connection_Fault values are OK and Faulted.

At connection establishment, all S_Connection_Fault variables shall be initialized to OK.

7.6.5.6 Producer input static variables

Producer input static variables are the configuration time parameters that define the operation of the safety protocol producer.

7.6.5.7 EPI

EPI is the expected packet interval. Safety data shall be produced at a rate not exceeding the EPI. The EPI is expressed as an integer value in microseconds. The legal range of values is from 100 to 1000 000 microseconds.

For all safety connections, the producer and all consumers shall all have the same EPI.

7.6.5.8 Timeout_Multiplier [per consumer]

The Timeout_Multiplier indicates the number of data production retries to use in the connection failure detection equations.

The Timeout_Multiplier shall be used to determine how many ping intervals (Timeout_Multiplier+1) a producer will wait before faulting a consumer who has failed to send a Time Coordination response to a Ping request.

The legal range of Timeout_Multiplier values shall be 1, 2, 3, or 4.

For any producer, each consumer may have a different Timeout_Multiplier.

7.6.5.9 Connection_Type

The Connection_Type specifies whether the connection is point-to-point or multipoint.

For any producer, each producer connection and the corresponding consumer connection(s) shall be of the same Connection Type.

7.6.5.10 Ping_Interval_EPI_Multiplier

Ping_Interval_EPI_Multiplier is the number that defines the Ping_Count_Interval for a particular connection.

The Ping_Count_Interval shall be equal to the EPI times the Ping_Interval_EPI_Multiplier.

The Ping_Count_Interval shall define the rate at which Time Coordination Messages will be requested and sent.

The Ping_Count_Interval shall define the rate at which Time Correction sections are sent to each consumer.

A legal Ping_Interval_EPI_Multiplier shall have a minimum value determined from the equation $\text{Max\_Consumer Number} \times \text{Timeout\_Multiplier} + 15$, and a maximum less than 1 000. The default values shall be 100 for multipoint and 19 for point-to-point.

The Ping_Interval_EPI_Multiplier should be set large enough to allow the ping request to be recognized by the server(s), the server(s) to send the Time Coordination section(s), and all Time_Coordination section(s) to arrive and be processed by the client. All these steps shall occur before the Ping_Interval time expires.

All Time_Correction sections shall be sent out by the client within the Ping_Interval.

The Ping_Interval_EPI_Multiplier shall be set small enough to ensure that the Ping_Count_Interval is less than or equal to 100 s.

A ping count interval of greater than 100 s may contribute greater than 0,5 s to the clock drift, depending upon the Timeout_Multiplier for that connection. To prevent clock size wrap around, 0,5 s has been set at the maximum value that can be allocated to clock drift.

The following variables will be used to calculate the Ping_Interval_EPI_Multiplier.

- C_TO_S_Delay_Multiplier equals the time as a multiplier of the EPI time that is equal to the time for message initiation, transport delay, and reception processing delay of the Data Message or the Time_Correction Message from the client to the server.
- S_TO_C_Delay_Multiplier equals the time as a multiplier of the EPI time that is equal to the time for message initiation, transport delay, and reception processing delay of the Time_Coordination Message from the server to the client.
- Ping_Rcvd_TO_Time_Coord_Sent_Multiplier equals the time as a multiplier of the EPI time that is allowed from the time that the EPI reception that should trigger the sending of the Time Coordination message until the message should be sent.
- Time_Coord_Rcvd_TO_Time_Coord_Processed_Multiplier equals the time as a multiplier of the EPI time that is allowed from the time that the Time Coordination message is received until the Time Coordination message is processed.

For a single network it is reasonable to expect that C_TO_S_Delay_Multiplier and S_TO_C_Delay_Multiplier are less than 2 times the EPI.

For a bridged network depending upon the number of hops, it is reasonable to expect that C_TO_S_Delay_Multiplier and S_TO_C_Delay_Multiplier are less than 4 times the EPI. 4 will be used as the minimum value for C_TO_S_Delay_Multiplier and S_TO_C_Delay_Multiplier when calculating the minimum value for Ping_Interval_EPI_Multiplier.

The C_TO_S_Delay_Multiplier and S_TO_C_Delay_Multiplier EPI multipliers should not have to consider messages being lost on the network. The Timeout_Multiplier will account for lost messages.

Ping_Rcvd_TO_Time_Coord_Sent_Multiplier should be less than 3 times the EPI.

Time_Coord_Rcvd_TO_Time_Coord_Processed_Multiplier should be less than 3 times the EPI.

The equation to determine the minimum Ping_Interval_EPI_Multiplier is:

$$\begin{aligned} & [\{ \text{the maximum Timeout\_Multiplier(C\#) for all consumers} \} \times \text{Max\_Consumer\_Num}] \\ & + \text{C\_TO\_S\_Delay\_Multiplier} \\ & + \text{S\_TO\_C\_Delay\_Multiplier} \\ & + \text{Ping\_Rcvd\_TO\_Time\_Coord\_Sent\_Multiplier} \\ & + \text{Time\_Coord\_Rcvd\_TO\_Time\_Coord\_Processed\_Multiplier} + 1 \end{aligned} \quad (1)$$

where

C# = Consumer_Number

Using 4 as the constant used for C_TO_S_Delay_Multiplier and S_TO_C_Delay_Multiplier and 3 as the constant used for Ping_Rcvd_TO_Time_Coord_Sent_Multiplier and Time_Coord_Rcvd_TO_Time_Coord_Processed_Multiplier:

The Ping_Interval_EPI_Multiplier shall be greater than or equal to:

$$[\{ \text{the Max Timeout\_Multiplier(C\#) for all consumers} \} \times \text{Max\_Consumer\_Num}] + 15 \quad (2)$$

This relationship will be relied upon by the producer when sending Time Correction messages. For C_TO_S_Delay_Multiplier and S_TO_C_Delay_Multiplier values greater than 4, and Ping_Rcvd_TO_Time_Coord_Sent_Multiplier and Time_Coord_Rcvd_TO_Time_Coord_Processed_Multiplier greater than 3, the Ping_Interval_EPI_Multiplier should be increased accordingly.

The Ping_Interval_EPI_Multiplier shall be less than: 100 s / EPI.

The Ping_Interval_EPI_Multiplier range of 75 to 100 will meet the restrictions defined above for all legal values defined below:

- EPI, 0,000 1 to 1 s;
- Timeout_Multiplier, 1 to 4;
- Max_Consumer_Num, 1 to 15;
- C_TO_S_Delay_Multiplier and S_TO_C_Delay_Multiplier, 1 to 4;
- Ping_Rcvd_TO_Time_Coord_Sent_Multiplier, 1 to 3;
- Time_Coord_Rcvd_TO_Time_Coord_Processed_Multiplier, 1 to 3.

For any producer, the producer and all consumers shall have the same Ping_Interval_EPI_Multiplier.

7.6.5.11 Max_Consumer_Number

Max_Consumer_Number is the maximum number of consumers that may be configured.

If the connection type is point-to-point, the Max_Consumer_Number shall be equal to 1.

For multipoint connections, the legal range of Max_Consumer_Number values shall be from 1 through 15.

For any producer, the producer and all consumers shall have the same Max_Consumer_Number.

7.6.5.12 Time_Coord_Msg_Min_Multiplier [per consumer]

Time_Coord_Msg_Min_Multiplier is the minimum number of 128 microseconds increments it could take for a Time Coordination Message to traverse from the consumer to the producer. A value other than 0 for this variable will allow the time stamp to be more accurate. The default for the Time_Coord_Msg_Min_Multiplier shall be 0, and the legal range of values shall be from 0 through 7 813, which equates to 0 through 1 second.

This is the minimum number of 128 μ s increments it could take:

From:

The consumer capturing Consumer_Time_Value from the Consumer_Clk_Count for the Time Coordination Message,

To:

The producer capturing the Producer_Rcvd_Time_Value upon reception of the Time Coordination Message

For any producer, each consumer may have a different Time_Coord_Msg_Min_Multiplier.

7.6.5.13 Producer connection derived static variables

These static variables are derived from the input static variables. These static variables are derived at configuration time and are used during producer processing.

7.6.5.14 Time_Drift_Per_Ping_Interval

The Time_Drift_Per_Ping_Interval is a value in 128 μ s increments that represents the worst-case time drift due to crystal inaccuracy during one Ping Interval.

The clocks in safety devices shall be accurate to within 0,02 % (0,000 2). Considering positive and negative drift the clocks will remain within $0,000\ 2 \times 2$ of each other or 0,004 (0,04 %).

The legal range of values is from 1 through 313, since the Ping Interval is limited to 100 s (100 000 000 μ s).

The Time_Drift_Per_Ping_Interval calculation is done with time references in microseconds (EPI is the EPI in microsecond units). The Time_Drift_Per_Ping_Interval calculation is done in 32 bit integer math with the result being a 16 bit integer. The divide by 320 000 factor in the Time_Drift_Per_Ping_Interval calculation is equivalent to multiplying by the worst case clock drift of 0,000 4 and dividing by 128 μ s.

Time_Drift_Per_Ping_Interval (C#) =
 Roundup [EPI $\times$ Ping_Interval_EPI_Multiplier / 320 000]
 OR 1 whichever is greater

(3)

where

C# = Consumer_Number

EXAMPLE If EPI = 8 000 µs and Ping_Interval_EPI_Multiplier = 100, then Time_Drift_Per_Ping_Interval(C#) = 3 (384 µs)

7.6.5.15 Connection_Correction_Constant [per consumer]

Connection_Correction_Constant is a value in 128 µs increments that is subtracted from the time stamp to represent the worst case error due to:

- Time drift;
- The asynchronous nature of the producer and consumer clocks;
- The minimum time for the Time Coordination Message to traverse from the consumer to the producer.

$$\text{Connection\_Correction\_Constant [C\#]} = \text{Time\_Drift\_Constant} + 1 - \text{Time\_Coord\_Msg\_Min\_Multiplier [C\#]} \quad (4)$$

where

C# = Consumer_Number

The Time_Drift_Constant (which is part of the Connection_Correction_Constant calculation) is a value in 128 µs increments that is subtracted from the time stamp to represent the worst-case time drift due to crystal inaccuracy. The producer and consumer clocks shall be accurate to within 0,02 % (0,000 2). Considering positive and negative drift the clocks will remain within 0,000 2 × 2 of each other or 0,000 4 (0,04 %).

The legal range of values for the Time_Drift_Constant is from 1 through 1 563.

The Time_Drift_Constant calculation is done with time references in microseconds (EPI is the EPI in microsecond units).

The Time_Drift_Constant calculation shall be done using at least 32 bit integer math with the result being a 16 bit integer.

The 1/320 000 factor in the Time_Drift_Constant calculation is equivalent to multiplying by the worst case clock drift of 0,000 4 and dividing by 128.

$$\begin{aligned} \text{Time\_Drift\_Constant (in 128 µs counts)} = \\ \text{Roundup [(Timeout\_Multiplier} \\ \text{(C\#) + 1) × EPI\_in\_µs × Ping\_Interval\_EPI\_Multiplier / 320 000]} \\ \text{OR 1 whichever is greater.} \end{aligned} \quad (5)$$

EXAMPLE Timeout_Multiplier = 2, EPI(in microseconds) = 8 000, Ping_Interval_EPI_Multiplier = 100, Time_Drift_Constant = 8, Time_Coord_Msg_Min_Multiplier = 2, then Connection_Correction_Constant = 7

7.6.5.16 Time_Coord_Response_EPI_Limit [per consumer]

The Time_Coord_Response_EPI_Limit is used to ensure that a Time Coordination message that returns to the producer requesting it, arrives within a time sufficient to prevent the possibility of a 16 bit time stamp math rollover. There are other checks to insure that the Time_Coordination message returns within the Ping Interval and that not too many Time Coordination messages are dropped.

Time_Coord_Response_EPI_Limit (a value in EPI increments) is used to determine if the time between the incrementing of the Ping Count and the Time Coordination response is less than a time value that is equal to:

$$\{5 + [\text{Time\_Coord\_Msg\_Min\_Multiplier} \times 0,000\ 128] + [\text{EPI} \times (\text{Consumer\_Num} - 1)]\}$$

in seconds. (6)

To keep the reaction time less than 5,8 s, the transport time of the safety message plus the transport time of the Time Coordination message minus the minimum Time Coordination message $[\text{Time\_Coord\_Msg\_Min\_Multiplier}/0,000\ 128]$ shall be less than 5,0 s if the worst case drift of 0,5 s is considered. This procedure keeps the Connection_Correction_Constant within range to prevent rollover at the age check.

The legal range of values for Time_Coord_Response_EPI_Limit is between 5 to 1 000.

If the Consumer_Active_Idle[Consumer_Num-1] == Active, Producers shall check that Time Coordination Messages are received within the Time_Coord_Response_EPI_Limit and fault the connection if they are not.

The Time_Coord_Response_EPI_Limit calculation is done with time references in microseconds (EPI is the EPI in microsecond units).

The Time_Coord_Response_EPI_Limit calculation shall be done in 32 bit integer math with the result being a 16 bit integer.

$$\begin{aligned} \text{Time\_Coord\_Response\_EPI\_Limit [C\#]} = \\ \text{Roundup } (\{5\ 000\ 000 + [\text{Time\_Coord\_Msg\_Min\_Multiplier}[C\#] \times 128] \\ + [\text{EPI} \times (\text{C\#} - 1)]\} / \text{EPI}) \\ \text{OR } 1\ 000 \text{ whichever is lower.} \end{aligned} \quad (7)$$

where

C# = Consumer_Number

EXAMPLE If Time_Coord_Msg_Min_Multiplier = 2 (256 µs), EPI = 8 000 µs, and Consumer_Number = 1, then Time_Coord_Response_EPI_Limit = 626

7.6.6 Producer dynamic variables

7.6.6.1 Producer_Clk_Count

Each product that produces or consumes safety data shall derive a periodic timer that increments a counter (Producer_Clk_Count or Consumer_Clk_Count) every 128 µs.

Producer_Clk_Count shall have tolerances and stability within +/- 0,02 %.

Producer_Clk_Count (and Consumer_Clk_Count) shall be 16 bits in length.

7.6.6.2 Producer_Safe_Data_TS

Producer_Safe_Data_TS is a variable that shall be set equal to Producer_Clk_Count at the point in time that the Safety_Data for a particular EPI data production is sampled and captured from the producing application.

This location that is sampled is the point where the producing applications time monitoring responsibilities end and the safety protocols time monitoring responsibilities begin. This value is used as the time stamp for the data that is relative to the producer's clock. The legal range of values is from 0 through 65 535.

7.6.6.3 Data_Time_Stamp

Data_Time_Stamp is the time that the safety data was sampled relative to the consumer's clock.

The Data_Time_Stamp value is derived by the producer for a point-to-point production.

The Data_Time_Stamp value is derived by the consumer in multipoint connections.

The legal range of values is from 0 through 65 535.

For point-to-point:

$$\text{Data\_Time\_Stamp} = \text{Producer\_Safe\_Data\_TS} + \text{Consumer\_Time\_Correction\_Value} \quad (8)$$

For multipoint:

$$\text{Data\_Time\_Stamp} = \text{Producer\_Safe\_Data\_TS} \quad (9)$$

7.6.6.4 Ping_Interval_EPI_Count

Ping_Interval_EPI_Count is a count of the number of EPI time periods between the incrementing of the Ping_Count.

Ping_Interval_EPI_Count is used to control Ping Intervals and to control the sending of Time Correction messages for multipoint producers.

The Ping_Interval_EPI_Count is incremented at every data production. When Ping_Interval_EPI_Count reaches the Ping_Interval_EPI_Multiplier, it is reset to 0, and the Ping count is incremented.

For multipoint producers, Time Correction messages shall be sent to the n consumers (where n = 1 thru Max_Consumer_Number), on the 9th thru n+9 productions of the Ping Interval.

The Time Correction message of consumer 1 shall be sent first, followed by 2, 3,...,n.

The intent of this logic is to distribute the Time Correction messages at a point within the Ping Interval, when the Time Coordination messages should have already arrived and been processed for this Ping Interval.

The legal range of values is from 0 through Ping_Interval_EPI_Multiplier.

At connection establishment, the Ping_Interval_EPI_Count variable should be set to 0000_{hex}.

7.6.6.5 RR_Con_Num_Index_Pntr

RR_Con_Num_Index_Pntr is a variable used by the multipoint producer to control the checking of the timeout on the reception of Time Coordination messages, and to control the sending of Time Correction messages.

A value greater than or equal to Max_Consumer_Number, indicates that the Time Coordination timeouts are not being checked on this EPI, and no Time Corrections are being sent based on this EPI.

A value of 0 thru (Max_Consumer_Number -1) indicates which Time Coordination timeouts will be checked (Consumer_Number = RR_Con_Num_Index_Pntr + 1) on this EPI and which

Time Correction message (Consumer_Number = RR_Con_Num_Index_Pntr + 1) may be sent based on this EPI production.

7.6.6.6 Time_Drift_Since_Last_Time_Coord

Time_Drift_Since_Last_Time_Coord is a variable that contains the maximum time drift that could have occurred between the producer and the consumer since the last time that a Time Coordination message was received. This value is equal to the Ping_Int_Since_Last_Time_Coord_Msg_Count times Time_Drift_Per_Ping_Interval. The Ping_Int_Since_Last_Time_Coord_Msg_Count is the number of Ping Intervals since the last Time Coordination message.

The Time_Drift_Since_Last_Time_Coord shall be used to indirectly determine if the delay of the Time Coordination message received is significantly greater than the delay for the Time Coordination message previously being used.

The Delay of the Time Coordination message has an impact on the Consumer_Time_Correction_Value that will be used. Greater delays in the Time Coordination message cause the Time_Stamp to be compensated in a negative direction, causing the age of the data to be greater.

If the difference between the delays of two consecutive Time_Coordination messages is greater than the EPI time, the later Consumer_Time_Correction_Value may cause the Time_Stamp of the next EPI production to actually be less than the Time_Stamp of the previous EPI production. This would result in a sequence error at the consumer of the Time_Stamp.

Since the maximum time drift is controlled, either the Last Time_Correction_Value minus the Time_Drift_Since_Last_Time_Coord value, or the New Time_Correction_Value value, shall be sent to the consumer, whichever is better.

This prevents long delays in the delivery of a Time Correction messages from causing sequence errors in the Time Stamp checks.

For multipoint producers, a Time_Drift_Since_Last_Time_Coord shall exist for each multipoint consumer, 1 through Max_Consumer_Number.

The legal range of values is from 0 through 3 910.

7.6.6.7 Worst_Case_Consumer_Time_Correction_Value

This a temporary value used to determine the Consumer_Time_Correction_Value.

7.6.7 Producer per consumer dynamic variables

7.6.7.1 General

The following dynamic variables are producer variables that exist for each consumer. For point-to-point connections there will be only one variable for each consumer. For multipoint there will be Max_Consumer_Number of variables, each representing an individual consumer.

7.6.7.2 Consumer_Time_Value[per consumer]

Consumer_Time_Value is the time value sent to the producer of a time stamp from the consumer of a time stamp or in the Time Coordination Message. A Consumer_Time_Value is received for each multipoint consumer, 1 through Max_Consumer_Number.

The legal range of values is from 0 through 65 535.

At connection establishment, all Consumer_Time_Value variables shall be set to 0000_{hex}.

When a Time Coordination Message is received; the producer shall check that the Time_Coordination_Section.Consumer_Time_Value is not the same as the last received Consumer_Time_Value. If it is, the Time Coordination message shall not be processed.

This check will prevent a producer from processing network generated duplicate Time Coordination messages.

7.6.7.3 Producer_Rcvd_Time_Value[per consumer]

Producer_Rcvd_Time_Value shall be set equal to Producer_Clk_Count at the point in time that the Time Coordination Message information is received from the time stamp consumer.

A Producer_Rcvd_Time_Value shall exist for each multipoint consumer, 1 through Max_Consumer_Number.

The legal range of values is from 0 through 65 535.

At connection establishment, all Producer_Rcvd_Time_Value variables shall be set to 0000_{hex}.

7.6.7.4 Consumer_Time_Correction_Value[per consumer]

The Consumer_Time_Correction_Value exists only for multipoint safety connections.

The Consumer_Time_Correction_Value is added to the producer's data time stamp to allow the result to indicate the worst-case earliest time that the data could have been sampled relative to the consumer's clock value.

A Consumer_Time_Correction_Value exists for each multipoint consumer, 1 through Max_Consumer_Number.

Consumer_Time_Correction_Value [Consumer_Number] =

$$\begin{aligned} & \text{Consumer\_Time\_Value [Consumer\_Number]} \\ & - \text{Producer\_Rcvd\_Time\_Value [Consumer\_Number]} \\ & - \text{Connection\_Correction\_Constant [Consumer\_Number]} \end{aligned} \quad (10)$$

The legal range of values is from 0 through 65 535.

At connection establishment, all Consumer_Time_Correction_Value variables shall be set to 0000_{hex}.

7.6.7.5 Ping_Int_Since_Last_Time_Coord_Msg_Count [per consumer]

Ping_Int_Since_Last_Time_Coord_Msg_Count is a run time count of the number of Ping Intervals between Time Coordination Messages for a particular consumer.

For point-to-point connections, the Ping_Int_Since_Last_Time_Coord_Msg_Count shall be incremented at the 8th EPI production of the Ping Interval.

For multipoint connections, Ping_Int_Since_Last_Time_Coord_Msg_Count shall be incremented every (8 + n)th EPI production of the Ping Interval, where n equals the consumer number - 1.

The Ping_Int_Since_Last_Time_Coord_Msg_Count for any particular consumer shall be reset upon the reception of a valid Time Coordination Message from that consumer.

If the Ping_Int_Since_Last_Time_Coord_Msg_Count ever exceeds the Timeout_Multiplier plus 1 for a connection, that consumer shall be set to the faulted state.

If the producer is multipoint, Ping_Int_Since_Last_Time_Coord_Msg_Count exists for each multipoint consumer, 1 through Max_Consumer_Number.

The legal range of values is from 0 through 5.

At connection establishment, all Ping_Int_Since_Last_Time_Coord_Msg_Count variables shall be set to 0000_{hex}.

7.6.8 Consumer data variables

7.6.8.1 General

This subclause describes the data variables that will be used by a safety consumer to receive safety data. The internal variables are defined in an attempt to define the behavior of the safety data consumer upon the reception of safety data.

7.6.8.2 Consuming connection status

The status of the safety data and the safety connections can be derived from the S_Con_Flt_C_Out, the S_Run_Idle_Out flags, and the Init_Complete_Out. Table 86 summarizes the logic that the consuming application can use to drive the safety data and status.

Table 86 – Consuming safety connection status

S_Con_Flt_C_Out	S_Run_Idle_Out	Init_Complete_Out	Connection Status	Description
Fault	Don't care	Don't care	Faulted	The safety connection is faulted. Restart will be required. Put the data in safety state
OK	Idle	0	Idle	The safety connection has not been activated. Put the data in safety state
OK	Idle	1	Idle	The safety connection has been activated and the producing application is in Idle Mode. Put the data in safety state
OK	Active	0	N/A	Illegal condition – can not occur
OK	Active	1	Run	The safety connection is active and the producing application is in Run Mode. Use the data

The Connection Status of consuming safety connections shall be indicated as defined in Table 86.

7.6.8.3 S_Con_Flt_C_Out

S_Con_Flt_C_Out indicates to the consuming application if it should put the safety data in a safety state because the safety validators have detected a safety connection fault. S_Con_Flt_C_Out_Temp is used as a temporary location for this variable.

The S_Con_Flt_C_Out values are OK and Fault.

At connection establishment, all S_Con_Flt_C_Out flags shall be set to OK.

7.6.8.4 S_Run_Idle_Out

S_Run_Idle_Out shall be used by a consuming application to determine if it should put the safety data in a safety state.

The S_Run_Idle_Out shall indicate Idle when either the producing application is Idle or the Safety Validators are not active. S_Run_Idle_Out_Temp is used as a temporary location for this variable.

The S_Run_Idle_Out values are Idle and Run.

At connection establishment, all S_Run_Idle_Out flags shall be set to Idle.

7.6.8.5 Init_Complete_Out

Init_Complete_Out indicates if the time stamp initialization is complete or not.

S_Run_Idle_Out shall be in Idle until Init_Complete_Out is set to a 1.

After the connection is first established, the consuming application shall close its connection if initialization is not completed within 10 s.

The S_Run_Idle_Out will indicate Idle, regardless of this timeout. If the timeout is incorporated, the error shall be latched until the safety connection is closed and a re-open attempt is made.

The Init_Complete_Out flag indicates whether the time stamp initialization is complete or not.

For Point-to-point, initialization complete shall indicate that the first time coordination message has been received by the producer, and the producer is producing data with the time stamp relative to the consumers clock.

For Multipoint, initialization complete shall indicate that the first time coordination message has been received by the producer for this consumer, and the producer has received the 1st time correction message based on the time coordination information.

For Point-to-point, the Data_Time_Stamp value shall be set to 0 by the producer until the first time coordination section is received.

If the consumer receives the Data_Time_Stamp as a value other than 0, the time stamp checking shall be enabled.

For Multipoint, the Data_Time_Stamp value shall always be set equal to the producers clock.

The Multipoint consumer shall enable time stamp checking upon the reception of the first time correction section.

For all consumed safety connections, time stamp checking shall be enabled if the mode of the data indicates Run.

Init_Complete_Out_Temp is used as a temporary location for this variable.

7.6.9 Consumer input static variables

7.6.9.1 Connection_Type

Connection_Type specifies whether the connection is point-to-point or multipoint. For any producer, the producer and all consumers shall have the same Connection_Type.

7.6.9.2 Consumer_Num

Consumer_Num is the number a Multipoint Consumer is assigned (1 through 15).

7.6.9.3 Network_Time_Expectation_Multiplier

Network_Time_Expectation_Multiplier is the maximum number of 128 μ s increments that a consumer should allow the age of the safety data to reach.

If consumed safety data becomes older than Network_Time_Expectation_Multiplier, the safety data shall be put in the safety state.

The legal range of values is from 0 through 45313, which equates to 0 through 5,8 s. The 5,8 s limit, the requirement to check the Network_Time_Expectation at least once every 2 s, and the maximum Time_Correction value change (or maximum clock drift) for a Ping interval is 0,5 s, will ensure that the clock base value of 8,388 s ($FFFF_{\text{hex}} \times 0.000\ 128$) does not wrap around between checks.

The actual network reaction time may be this value in time plus 1 EPI if the producing application is asynchronous to the data production, since in this case the data may be one EPI old at the time that the time stamp was placed on the produced data.

7.6.9.4 Timeout_Multiplier [per consumer]

Timeout_Multiplier is the number of data production retries to use for in network fault detection equations. This is the same value used by the producer.

7.6.9.5 Ping_Interval_EPI_Multiplier

Ping_Interval_EPI_Multiplier is the number that defines the Ping_Count_Interval for a particular connection. The Ping_Count_Interval is equal to the EPI times the Ping_Interval_EPI_Multiplier.

This is the same value used by the producer.

7.6.9.6 Consumer connection derived static variables

These static variables are derived from the input static variables. These static variables are derived at configuration time and are used during producer processing.

7.6.10 Consumer dynamic variables

7.6.10.1 Consumer_Clk_Count

Each product that consumes safety data will derive a periodic timer that increments a counter (Consumer_Clk_Count) every 128 μ s.

Consumer_Clk_Count will have tolerances and stability within +/- 0,02 % because safety devices are required to have clocks with this tolerance.

Consumer_Clk_Count shall be 16 bits in length.

If the difference between the Consumer_Clk_Count at the time that the data is given to the consuming application for use and the Data_Time_Stamp sent with the data exceeds the Network_Time_Expectation_Multiplier, the S_Con_Flt_C_Out flag shall be set to indicate that the Reaction Time for the network has not been met.

7.6.10.2 Last_Ping_Count

A consumer of safety data shall be able to detect when the Ping_Count has changed. This 2 bit variable is used to hold the Ping_Count for comparison on the next reception.

Since the initial production of safety data should have a Ping_Count of 00, the Last_Ping_Count shall be initialized to 11.

7.6.10.3 Time_Coordination_Count_Down

The Time_Coordination_Count_Down shall be set equal to the Consumer_Number upon the reception of a multipoint ping request.

The consumer shall decrement the Time_Coordination_Count_Down value on subsequent reception of consumed data until the value is equal to 0.

If the value is equal to 1 during any reception of consumed data, the ping response for that consumer shall be sent.

The intent of this logic is to distribute the multipoint ping responses.

When a connection is first established, the Time_Coordination_Count_Down shall be initialized to 0.

7.6.10.4 Corrected_Data_Time_stamp

This value is the consumer copy of the received Data_Time_Stamp.

For multipoint, the Corrected_Data_Time_Stamp shall be a sum of the received time stamp and the Last_Rcvd_Time_Correction_Value.

7.6.10.5 Last_Data_Time_stamp

This value is the saved copy of the received Data_Time_Stamp.

The Last_Data_Time_Stamp shall be used to detect out of sequence messages.

When a connection is first established, the Last_Data_Time_Stamp shall be initialized to 0.

7.6.10.6 Last_Rcvd_Multi_Cast_Active_Idle

This value is the saved copy of the received Multi_Cast_Active_Idle.

The Last_Rcvd_Multi_Cast_Active_Idle shall be used to detect a change from active to idle, which would be considered a fault.

When a connection is first established, the Last_Rcvd_Multi_Cast_Active_Idle shall be initialized to 0.

7.6.10.7 Last_Rcvd_Time_Correction_value

This value is the saved copy of the received Time_Correction_value.

The Last_Rcvd_Time_Correction_Value shall be used to correct the Time Stamp for each multipoint consumer.

When a connection is first established, the Last_Rcvd_Time_Correction_Value shall be initialized to 0.

7.6.10.8 Time_Correction_Ping_Interval_Count

This value indicates the number of Ping Intervals that have occurred since the last reception of a Time Correction message.

If the Time_Correction_Ping_Interval_Count is greater than the Timeout_Multiplier + 1, the connection shall be faulted.

7.6.10.9 Time_Correction_Received_Flag

This value indicates that a Time Correction message has been received for this multipoint consumer. This value indicates that a time correction message has been received and the time stamp can be corrected and checked.

When the Time_Correction_Received_Flag is set, the consumer shall begin the process of correcting and checking the time stamp.

7.6.10.10 Data_Age

This value indicates the worst case age of the data in 128 μ s increment.

The age serves as a measure from when the time the Producer_Clk_Count was captured until the Time_Stamp was checked.

7.6.10.11 Max_Data_Age

This value indicates maximum value of Data_Age since the last time that it was reset.

8 Safety communication layer management

8.1 Overview

This subclause does not attempt to provide a tutorial on every aspect of CPF 2, but rather a high-level overview of how CPF 2 connections are established on each network and across network types, highlighting the features used and extensions required to support Safety Connections.

All FSCP 2/1 connections shall be established using a Forward Open with a safety network segment (CP 2/2) or a SafetyOpen (CP 2/3). There are measures for detection of both system and user errors defined. Table 87 defines the errors and the measures to detect these errors.

Table 87 – Connection establishment errors and measures to detect errors

Connection establishment errors	Measures to detect connection establishment errors				
	User testing	OUNID verification	TUNID verification	CPCRC CRC	Configuration ID (SCID)
User misroutes connection request	X	X1	X		
System misroutes connection request			X		
Corrupted SafetyOpen				X	X
Corrupted or changed application configuration					X
Reconfiguration from wrong originator		x ^a			
^a Only applies if ownership has been established.					

8.2 Definition of the measures used during connection establishment

System-Wide Unique "Safety Network Number" (SNN) – Uniquely identifies a network across all networks in the safety system. The user safety manual shall contain an advisory that states "The user should assign SNN numbers for each safety network or safety sub-net that are unique system-wide", or words to that effect.

The Safety Network Number makes up part of the UNID. This value is a 6-octet value that uses the CPF 2 DATE_AND_TIME format.

The DATE values of 1 thru 11 687 decimal (1972-01-01 thru 2002-12-31) are past time values that are reserved for Manual Setting values of the SNN.

The SNN may be one of two types:

- Manual Setting, or
- Software Tool Generated.

The Type of SNN can be determined from the DATE section, see Table 88 for details.

Table 88 – SNN Date/Time allocations

From DATE (Decimal)	To DATE (Decimal)	From DATE (Day)	To DATE (Day)	SNN Type
0	0	1972-01-01	1972-01-01	Null SNN (illegal value)
1	1	1972-01-02	1972-01-02	Manual setting - backplane
2	2	1972-01-03	1972-01-03	Manual setting – CP 2/1
3	3	1972-01-04	1972-01-04	Reserved for future use
4	4	1972-01-05	1972-01-05	Manual setting – CP 2/2
5	5	1972-01-06	1972-01-06	Manual setting – CP 2/3
6	1 460	1972-01-07	1975-12-31	Reserved for future use
1 461	11 687	1976-01-01	2003-12-31	Vendor specific range
11 688	65 534	2004-01-01	2151-06-05	Software tool generated
65 535	65 535	2151-06-06	2151-06-06	Target indication that no SNN has been set

If a software tool reads a reserved SNN Type, it shall display it as a Hex value.

The legal Range of TIME values for the Type of SNN are listed in Table 89.

Table 89 – SNN legal range of time values

SNN type	From time hex	To time hex	From time d/h/min/s	To time d/h/min/s
Null SNN	00000000	00000000	00d00h00min00,000s	00d00h00min00,000s
Manual setting (DATE = 1 thru 5)	00000001	0000270F (9 999 decimal)	00d00h00min00,001s	00d00h00min09,999s
Reserved for future use			-	-
Software tool generated	00000000	05265BFF	00d00h00min00,000s	00d23h59min59,999s
Target indication that no SNN has been set	FFFFFFFF	FFFFFFFF	-24d20h31min23,648s	-24d20h31min23,648s

System-Wide "Unique Node Identifier" (UNID) – Uniquely identifies a node across all networks in the safety system. This value is made up of the Safety Network Number (SNN) and the Node Address of the device. This value is a structure consisting of the 6-octet SNN and a 4-octet Node Address. The 4-octet node address is sized to accommodate all forms of CPF 2 network addresses.

```
struct UNID
{
    S_SNN SNN;
    UDINT    MACID;
};
```

MACIDs smaller than 4-octets (i.e., CP 2/3) shall be aligned to the least significant octet in little endian order. The unused octets shall be zero filled.

Safety Function – The UNID is used to uniquely identify originators and/or targets during the connection establishment process.

Other uses of UNID:

TUNID = Target's Unique Network Identifier

Safety Function – Identifies the target in the SafetyOpen

OUNID = Originator's Unique Network Identifier

Safety Function – Identifies the originator in the SafetyOpen

OCPUNID = Output Connection Owning UNID

Safety Function – Identifies the owner of the output connection in the target, this shall be equal to the OUNID in normal operation

CFUNID = Configuration Owning UNID

Safety Function – Identifies the owner of the safety configuration in the target, this shall be equal to the OUNID in normal operation

Producer Identifier (PID) – Identifies the producer of the runtime data and Time Correction data.

The Producer Identifier (PID) shall be used as an initial seed value in the CRCs in the runtime protocol.

The Consumer shall obtain the Producer Identifier from either the SafetyOpen or SafetyOpen response (depending upon originator) and use the value as an initial seed in runtime checking the safety CRCs.

The value of the PID is Vendor ID + Device Serial Number + CIP Connection Serial Number.

Safety Function – The PID is used to ensure the correct routing of the produced/safety connection data and Time Correction data.

Consumer Identifier (CID) – Identifies the producer of the Time Coordination message.

The Consumer Identifier (CID) shall be used as an initial seed value in the CRC in the Time Coordination message.

The Producer shall obtain the Consumer Identifier from either the SafetyOpen or the SafetyOpen Response (depending upon originator) and use the value as an initial seed in runtime checking the safety CRCs.

The value of the CID is Vendor ID + Device Serial Number + CIP Connection Serial Number.

Safety Function – The CID is used to ensure the correct routing of the Time Coordination data.

Connection Parameters CRC (CPCRC) – This is a safety related CRC covering the CPF 2 connection establishment data in the SafetyOpen. This CRC will also cover the additional parameters required by the safety protocol. The CRC used for the CPCRC is CRC-S4 and is defined in 7.1.2.

Safety Function – The CPCRC is used to ensure the integrity of the connection data contained within the SafetyOpen.

Safety Configuration CRC (SCCRC) – This is a CRC that covers the device configuration data that is contained in a SafetyOpen. The parameters covered by the CPCRC are not covered by the SCCRC. The CRC used for the SCCRC is CRC-S4 and is defined in 7.1.2.

The Safety Configuration Data CRC (SCCRC) shall be calculated over the entire application data section that is sent to the device being configured by the tool.

The SCCRC itself shall be included in the calculation of the CPCRC.

Safety Function – The SCCRC is used to ensure configuration data integrity while being transferred and to ensure configuration data integrity in storage.

Safety Configuration Time Stamp (SCTS) – This is a safety related signature that identifies the revision of the device configuration contained in either a SafetyOpen or in a safety device.

The SCTS shall be a Time/Date stamp marking the Time and Date of the configuration creation or change.

The SCTS parameter in the SafetyOpen shall be included in the calculation of the CPCRC.

The configuration software shall follow the CPF 2 defined Date and Time format (see IEC 61131-3) for setting the signature.

Safety Function – The SCTS is used to ensure the identity of the safety-related application data.

Safety Configuration Identifier (SCID) – The SCID is a combination of the SCCRC and the SCTS. The SCTS and SCCRC together serve to uniquely identify a configuration to the user and originator.

The SCID = 0 in a Form 2b SafetyOpen shall have special meaning to indicate that the SCID shall not be checked by a target before accepting the connection.

This special SCID value allows originators who are not concerned about the configuration of a device to make connections without knowing the SCID value.

The safety manual shall contain user instructions that state “If you choose to configure safety connections with an SCID = 0, you are responsible for ensuring that originators and targets have the correct configurations”, or similar equivalent wording.

8.3 Originator-Target relationship validation

In the CPF 2 framework, nodes are either originators (nodes that request connections) or targets (nodes that receive connection requests). Controllers are typically the connection originators. I/O modules (as well as other controllers or monitors) are typically the targets. Targets may support more than one application connection point.

Consider the system depicted in Figure 49. For the purposes of illustration, assume that all targets are the exact same.

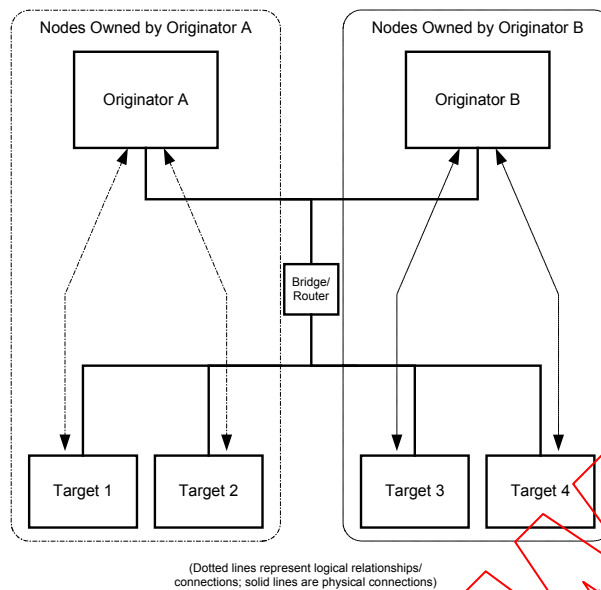


Figure 49 – Target ownership

Use of standard bridges/routers and standard lower-layer protocol stacks within the end-nodes offers the opportunity for additional errors that require protection measures that are more stringent than those applied in standard CPF 2 connection establishment. For instance, the bridge/router could mis-direct (for example corrupt the destination MACID of) a connection request from originator A intended for Target 1 to Target 2, 3, or 4 instead. Or the source of the error could be the user mis-configuring originator B to connect to Target 1 while originator A is off-line and Target 1 is powered. Without a second level of comparison above the identical electronic key, the connection from originator B to Target 1 would be allowed; at which point originator B's application could place Target 1 (assuming it's an output device) into an incorrect operational state with respect to originator A's commissioned application.

8.4 Detection of mis-routed connection requests

All SafetyOpen requests will contain the UNID of the target device. Upon the receipt of a SafetyOpen the target shall compare the SafetyOpen target UNID (TUNID) with its own UNID. If the TUNID to UNID comparison is successful then the SafetyOpen destination is deemed to be correct and processing shall continue.

8.5 SafetyOpen processing

There are two ways of processing the SafetyOpen (see Figure 50):

- Destination, Ownership, Configuration data processing and SCID checking;
- Destination, Ownership and SCID Checking (Optional).

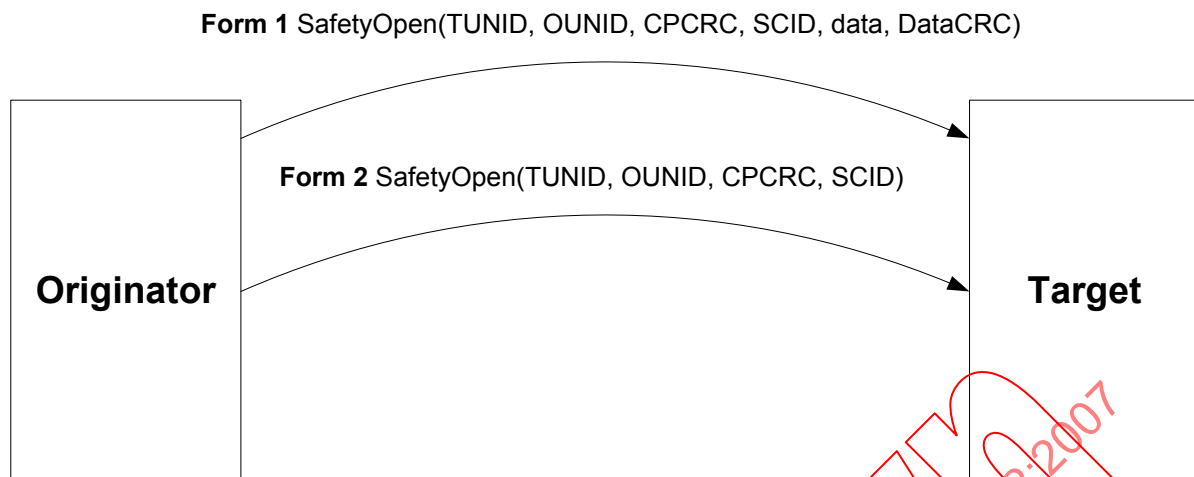


Figure 50 – SafetyOpen forms

8.6 Ownership management

There are two methods of establishing configuration and/or connection ownership, through the configuration tools or from the SafetyOpen. The tool can set the mode to be tool configuration only or originator configuration only.

Any attempts to reconfigure a device via the SafetyOpen shall be rejected if the tool only configuration mode is selected.

If the originator configured mode is used, the OUNID of the selected originator can be set by the software tool during target device commissioning for added security. Figure 51 defines the ownership states the methods of transitioning from ownership state to another ownership state.

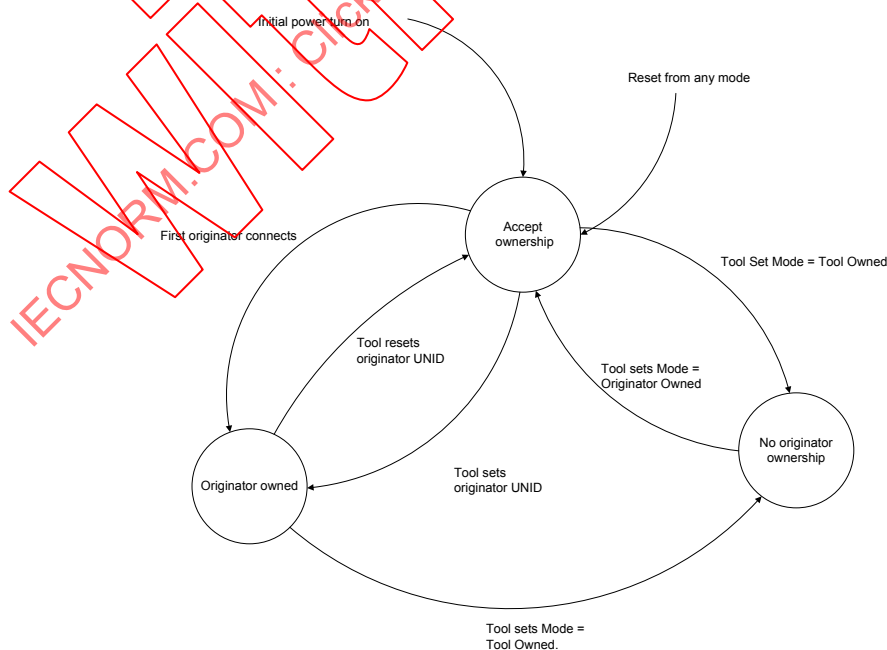


Figure 51 – Connection ownership state chart

To establish ownership of the configuration (and the connection in outputs), in targets the OUNID of the originating device shall be passed to the target in the SafetyOpen service.

The OUNID of the originating device shall be stored in non-volatile storage with the configuration to which it is associated.

Outputs devices shall store the OUNID associated with safety outputs to prevent other originators from hijacking outputs inadvertently. Keeping the originator OUNID associated with the configuration will allow multiple devices to connect to an input device and still maintain configuration ownership. Figure 52 defines the mapping of the SafetyOpen UNIDS to the target device originator ownership locations.

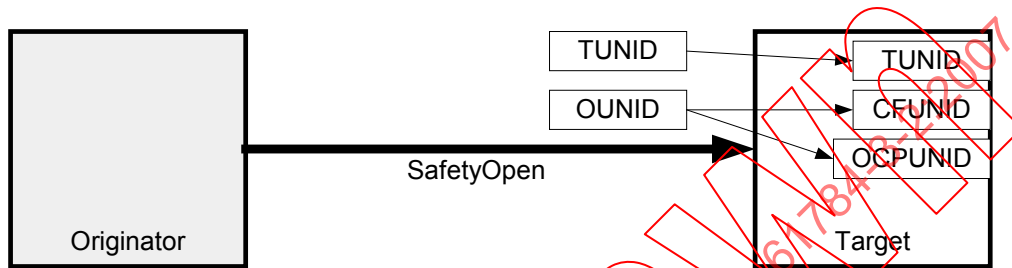


Figure 52 – SafetyOpen UNID mapping

8.7 Bridging different physical layers

CPF 2 nodes share a common application and user layer as defined in Figure 53 below.

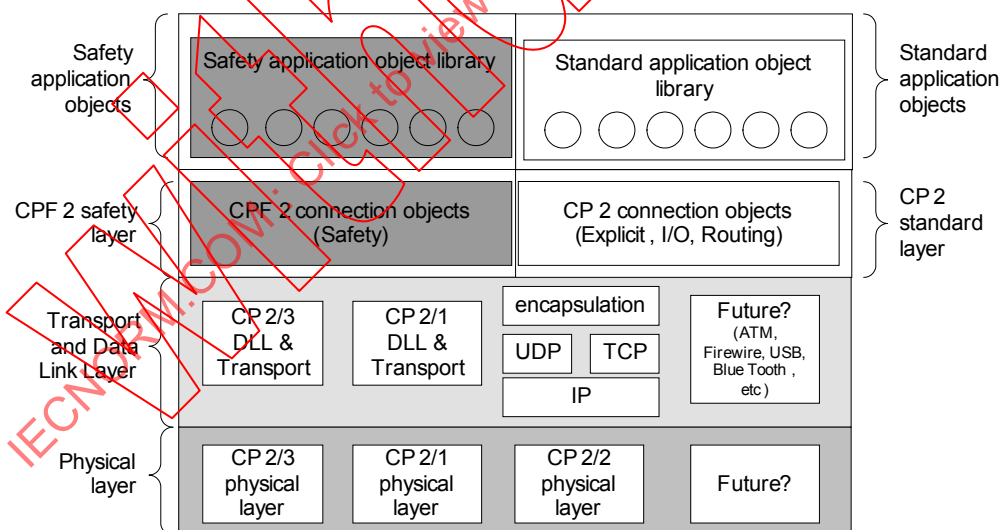


Figure 53 – Common CPF 2 application layer

The CPF 2 connection objects are the entities that reconcile the inherent differences between the capabilities and services provided by the underlying networks, exchanging SIL 3 data with the application objects above. For instance, CP 2/3 physical layer (i.e. CAN) data frames are limited to a maximum of 8 octets each. The CPF 2 connection object in a CP 2/3 node transports I/O packets larger than 8 octets via CP 2/3's I/O fragmentation protocol; where as on the higher-capacity networks like CP 2/2, the maximum connection size of 511 octets fits

into one physical layer data frame. Once a safety connection is established, the safety-related application layer exchanges safety data with the safety connection objects.

The safety protocol has been designed to enable end-to-end transport of the safety data without requiring specific knowledge of the safety protocol within or SIL 3 certification of intermediate devices. Figure 54 depicts an end-to-end connection between a safety node on CP 2/3 and a safety node on CP 2/1 routed across a CP 2/2 network. The safety connection endpoints enable the application level peers to share SIL 3 data.

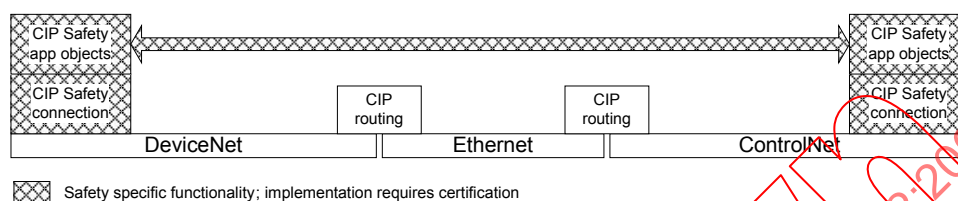


Figure 54 – End-to-End routing example

Safety connection establishment is dynamic on every network. CP 2/2 originators employ the ForwardOpen service of the connection manager object. The connection path of a ForwardOpen service's data includes a safety network segment for the end node. This safety network segment informs the recipient that a connection object capable of supporting a safety connection shall be available in order to accept the request. It also includes data used to specify Connection object attributes related to its safety behavior (for example expectation timers).

On CP 2/3 Safety, the originator shall establish an explicit messaging connection with the router or end-node to deliver the ForwardOpen or SafetyOpen service requests. The other CPF 2 networks (typically) deliver the ForwardOpen to the UCMM without a connection.

8.8 Safety connection establishment

8.8.1 Overview

Establishing safety connections is a key system function. There are two safety elements that need to be addressed; setting up a connection relationship in an originator, and setting up a run-time connection. This subclause will describe the system requirements to perform both of these operations with integrity.

8.8.2 Basic facts for connection establishment

The basic facts for safety connection establishment are listed below.

- SafetyOpen is a service defined in the Connection object.
 - An explicit messaging connection is required with a path that targets the Connection object.
- ForwardOpen is a service defined in the Connection Manager object.
 - This is an unconnected message with a path that targets the Connection Manager object.
 - When originating from a CP 2/3 Safety originator, an explicit messaging connection is required with a path that targets the Connection Manager object.
- One additional set of connection parameters are required in the SafetyOpen to define the 3rd connection point in CP 2/3 Safety nodes when the connection type is Multipoint. These parameters are contained in the Safety Segment.
- A null application path is defined to mean either that a connection has no application data, or the SafetyOpen contains no configuration data. Each safety module vendor shall define a null path that is appropriate for their device. A null path shall always use a full path

reference (i.e. 20 04 24 [vendor specific instance]). For EDS-based devices, the selected null instance shall be disclosed with an entry in the connection manager section of the EDS file (refer to ISO 15745-4 Annex C).

- The application path for the Time Coordination connection shall always be a null path.
- Form 2 SafetyOpen (no configuration data) shall include a null application path for the configuration path.
- There is no path required for the Time Correction connection.
- The extra parameters of the 3rd connection point are named and have values as follows:
 - Time Correction RPI;
 - Time Correction Network Connection ID;
 - Connection Type = Multipoint;
 - Priority = High Priority;
 - Fixed/Variable = Fixed;
 - Connection Size = 6 (fixed Time Correction size).
- The Time Correction Network Connection ID is not part of the CPCRC.
- The standard O-to-T and T-to-O connection sizes shall always reflect the 2-connection size for each. In other words, the connection that has the Data and Time Correction packets concatenated will have a size that reflects this combination. It will be the responsibility of bridges and the CP 2/3 Safety nodes to calculate the appropriate sizes for the connection resources on CP 2/3 Safety (this is made easier in that the size of the 3rd connection point, provided in the SafetyOpen message, can be added or subtracted).

8.8.3 Configuring safety connections

This subclause defines the parameters needed to perform the safety protocol. It also defines the Safety Segment parameters needed to be passed in the SafetyOpen to properly set up a safety connection. This subclause will describe how these parameters get assigned and suggest reasonable defaults for a safety system.

The parameters that need to be configured for a safety connection are:

- Target UNID;
- Originator UNID;
- Ping Interval EPI Multiplier;
- Time Coord Msg Min Multiplier;
- Timeout Multiplier;
- Max Consumer Number;
- Network Time Expectation Multiplier;
- ASYNC.

All but the ASYNC parameter are included in the SafetyOpen service. The configuration Software for a Safety Connection Originator shall provide a means for the following parameters to either be configured or given default values: Expected Packet Interval, Target UNID, Originator UNID, Timeout Multiplier, Ping Interval EPI Multiplier, Time Coord Msg Min Multiplier, Max Consumer Number, Network Time Expectation Multiplier, and ASYNC. Table 90 summarizes how these parameters will be obtained.

Table 90 – Safety connection parameters

Parameter	Suggested configuration method	Suggested default value
Data Expected Packet Interval (EPI)	User assigned or tool-based estimation	None, user shall provide one
Time Coordination EPI	Data EPI × Ping Interval EPI Multiplier	Software calculated
Time Correction EPI	Data EPI × Ping Interval EPI Multiplier	Software calculated
Target UNID	User assigned or obtained by software	None, user shall provide one
Originator UNID	User assigned	None, user shall provide one
Timeout Multiplier	User assigned	2
Ping Interval EPI Multiplier	Default value for inputs or outputs	100 for inputs 19 for outputs
Time Coord Msg Min Multiplier	Default value on path	2 for non-bridged 3 for bridged
Max Consumer Number	Fixed value	15
Network Time Expectation Multiplier	Calculation	Calculated
ASYN	EDS file input	1 (asynchronous)

8.8.4 Network time expectation multiplier

One of the primary calculations required of a Safety Configuration Tool or Connection Originator is the calculation of the Network Time Expectation Multiplier. This subclause provides a simplified way to do this calculation.

Subclause 9.3.2 defines the Network Time Expectation Multiplier as:

$$\text{Network\_Time\_Expectation\_Multiplier} > \frac{[\text{EPI\_in\_sec} \times \text{Timeout\_Multiplier} + \text{Safety\_Message\_Time(max)} + \text{Time\_Coord\_Message\_Time(max)}] / 128 \mu\text{s} + \text{Connection\_Correction\_Constant}}{1} \quad (11)$$

Some simplifying assumptions can be made as follows:

- Safety Message Time (max) = 1 EPI;
- Time Coordination Msg Time = Safety Message Time (max);
- Effect of Time_Drift_Constant is negligible (Connection Correction Constant can be ignored).

Under these assumptions, the generalized equation can be stated:

$$\text{Network Time Expectation Multiplier} = [\text{EPI\_in\_sec} \times (\text{Timeout Multiplier} + 2)] / 128 \mu\text{s} \quad (12)$$

The Network Time Expectation Multiplier parameter can now be determined by a common set of user inputs and one EDS file parameter.

Figure 55 shows where the various parameters defined in Table 90 are obtained. It also shows how these parameters are used by the software to construct the safety segment of a SafetyOpen.

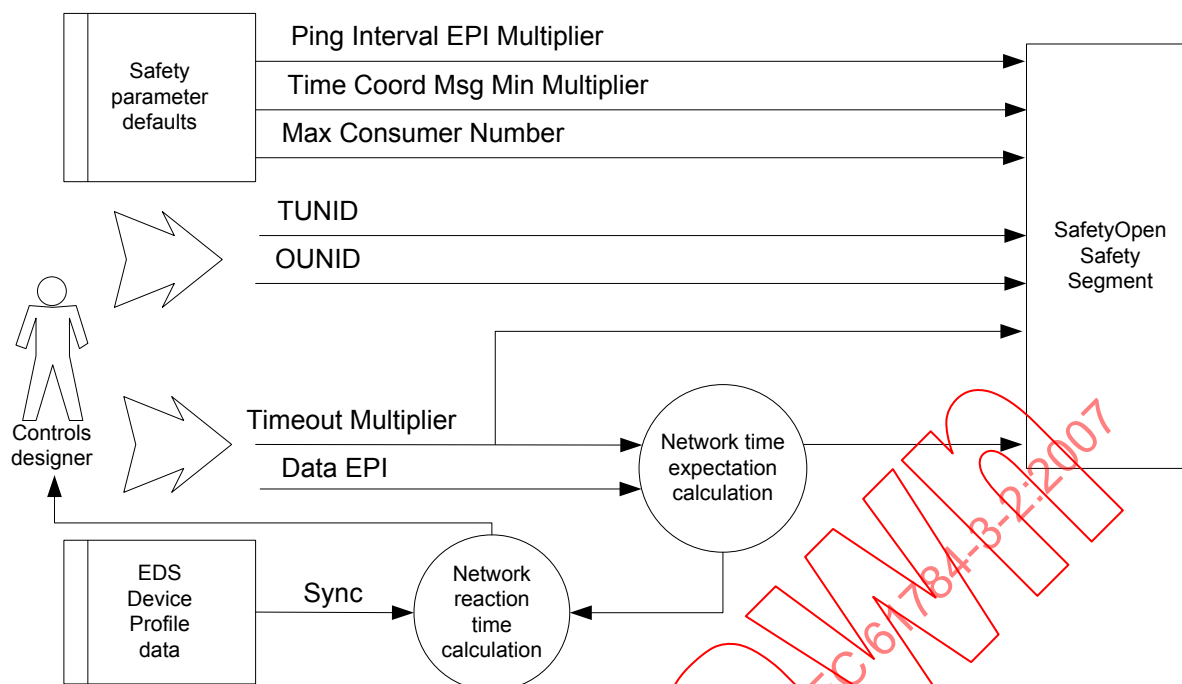


Figure 55 – Sources for safety related connection parameters

Figure 56 defines how the SafetyOpen parameters are mapped in Originators and Targets of a safety connection.

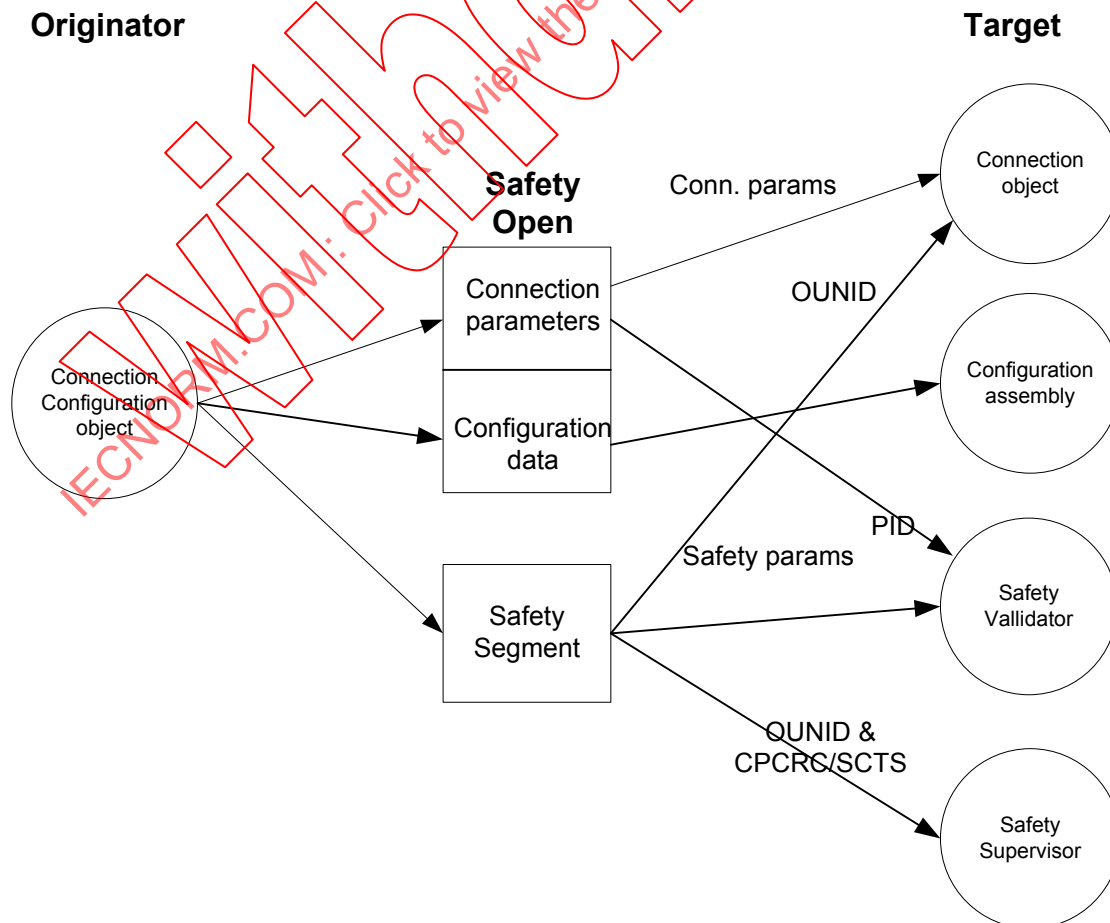


Figure 56 – Parameter mapping between originator and target

8.8.5 Establishing connections

When configuration data is transmitted within the SafetyOpen this is defined as a Form 1 SafetyOpen. When no configuration data is transmitted within the SafetyOpen this is defined as a Form 2 SafetyOpen. This is defined in Figure 57.

If a connection has configuration data, this configuration data shall only be sent when it is needed. Unless the configuration has been changed, this shall be accomplished by sending a Form 2 SafetyOpen, and only followed with a Form 1 SafetyOpen if the target responds with Device Not Configured error.

When a device receives a Form 1 SafetyOpen containing configuration data and an SCID that matches its existing configuration, it shall re-apply the configuration. All existing rules checks (i.e. OUNID and TUNID checks) shall still apply.

Since all safety nodes are required to have NV storage for configuration data, this is only likely on device replacement, initial start-up or when the configuration has changed. This process is illustrated in Figure 58.

Targets shall respond to a Form 2 SafetyOpen with an error code 01_{hex} additional code 0110_{hex} (device not configured), when the device is not configured.

When an originator detects that the configuration data held for a device has changed, it shall always issue a Form 1 connection request the first time it connects. The SCID returned shall be validated against the value held.

There are two kinds of Form 2 SafetyOpens: with and without the SCID.

- Form 2a: When a target receives a Form 2 SafetyOpen (no configuration data) accompanied by a non-zero Safety Configuration ID (SCID = SCCRC + SCTS), it shall compare it against the SCID of the configuration and only accept the connection if they match.
- Form 2b: When a target receives a Form 2 SafetyOpen (no configuration data) accompanied by a zero value SCID, it skips the check.

Table 91 provides a summary of the Forms of SafetyOpen.

Table 91 – SafetyOpen summary

	Form 1 SafetyOpen with data	Form 2a SafetyOpen with SCID check	Form 2b SafetyOpen without SCID check
TUNID	X	X	X
OUNID	X	X	X
UPID ^a	X	X	X
CPCRC	X	X	X
SCID	X	X	= 0
Config. Data in SafetyOpen	Yes	No	No
^a UPID is derived from components in the SafetyOpen or from application data returned in the SafetyOpen response. See 6.3.12.			

The processing of SafetyOpens in Targets will differ depending on the Form of SafetyOpen message that is received. Figure 58 defines the message and event sequence for the Form 1 SafetyOpen and the objects that are responsible for the various processing steps. Figure 58

defines that the primary difference between processing Form 1 and Form 2 is the step where configuration data is passed to a configuration assembly and an owner assigned to that configuration.

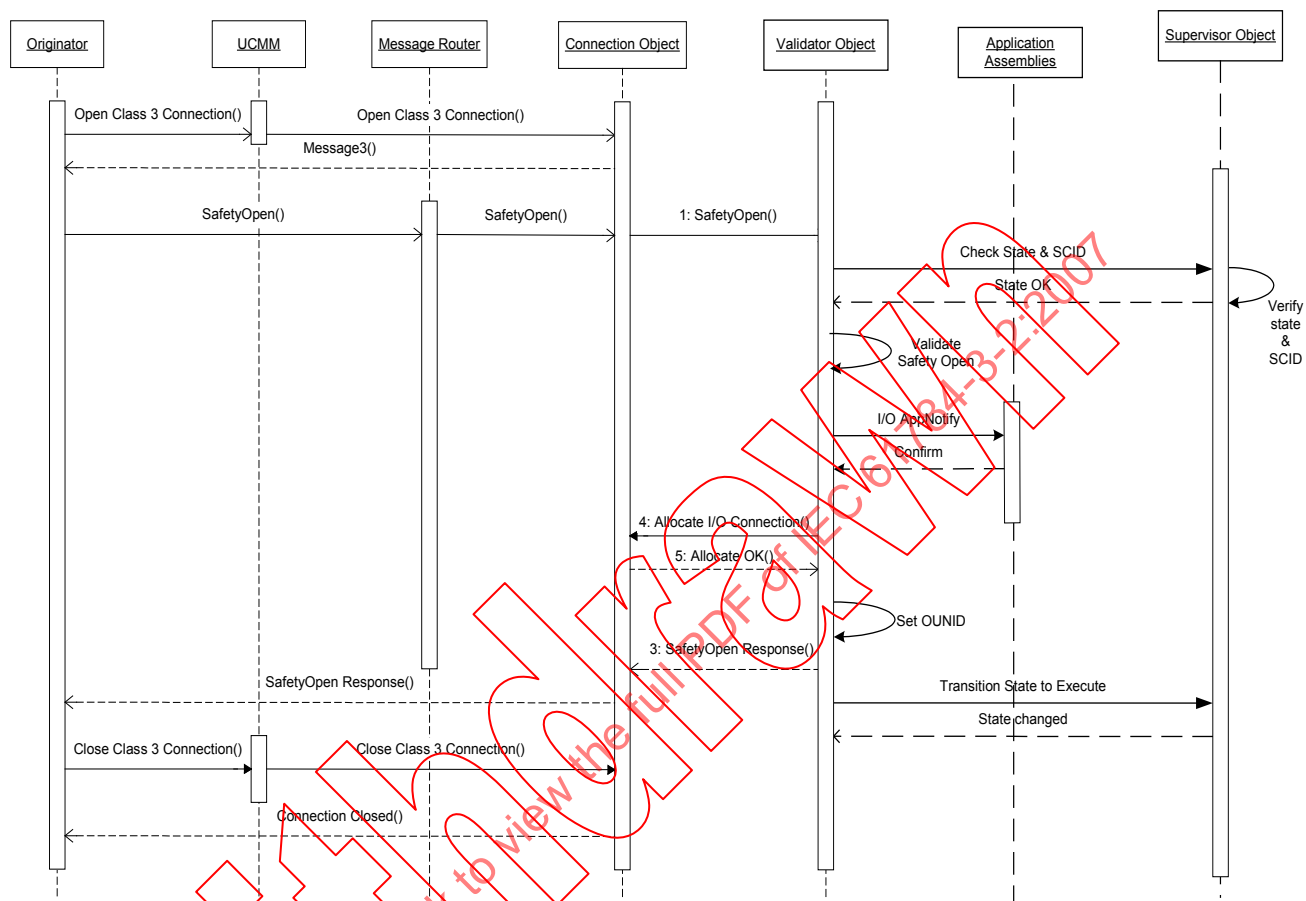


Figure 57 – CP 2/3 Safety connection establishment in targets for Form 2a SafetyOpen

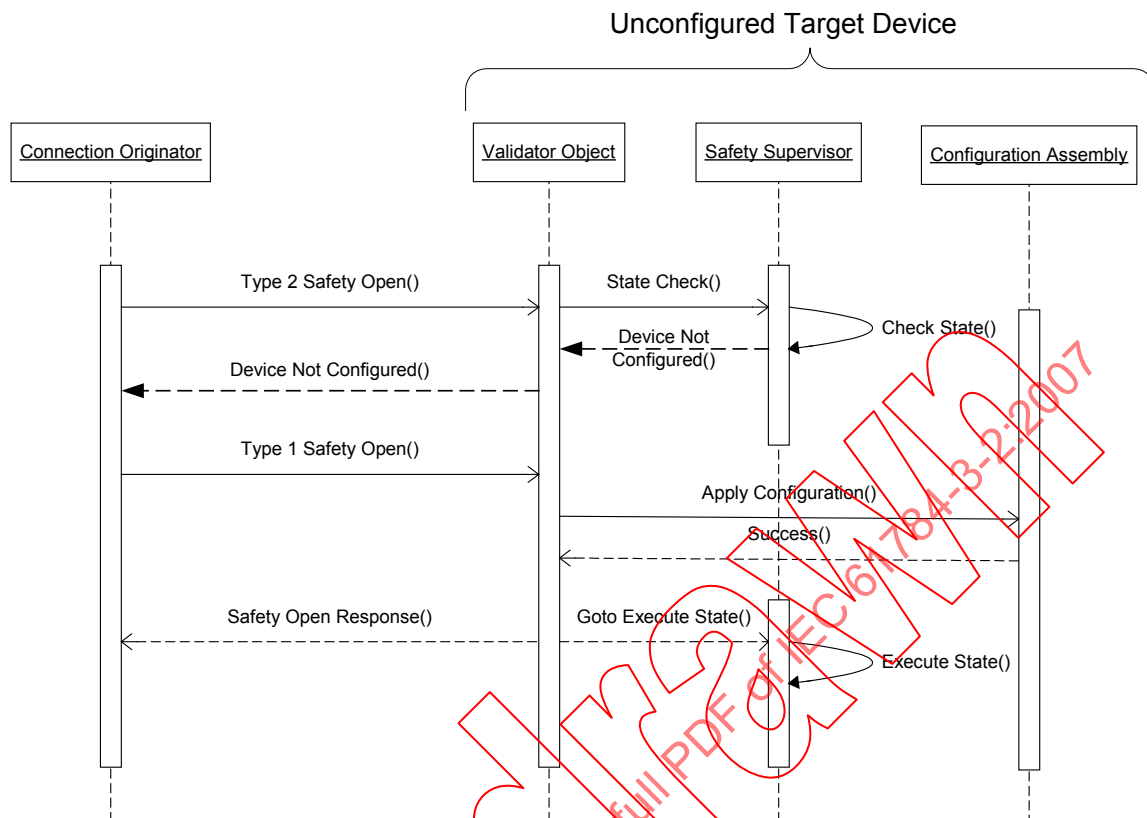


Figure 58 – General sequence to detect configuration is required

8.8.6 Recommendations for consumer number allocation

When a Multipoint connection is established, the producer of that connection shall allocate a unique consumer number to the consumer. The main difficulty with this process is not in the initial allocation, but in dealing with the case where a producer has closed a connection for some reason and now needs to decide when it is acceptable to re-assign that consumer number again.

Since multipoint producers will continue to produce data even though a connection to a particular consumer was closed (i.e. assumes other consumer connections still open), the only mechanism available for a multipoint consumer to detect the loss of the connection is by detecting that it has not gotten a Time Correction message within $\text{Timeout\_Multiplier}+1$ ping intervals. During this period, the Consumer still thinks the connection is open, is still processing received data, and is still sending Time Coordination responses (even though the producer ignores them).

If a producer were to get a new connection request from another consumer and re-allocate this consumer number, there would be two consumers using the same number during this period. The producer would again start processing Time Coordination responses and sending Time Correction messages with that consumer number. In other words, if the timing were just right, the old consumer might never detect the connection has failed and there would exist two consumers with the same number sending conflicting clock information and using conflicting correction values. While this case will ultimately be detected by the producer through PID seeding, a means shall be employed to minimize the occurrence.

Multipoint producers shall quarantine consumer numbers that have been taken off-line for $\text{timeout\_multiplier}+2$ ping intervals before it can be re-allocated to another connection. The

only exception to this is when a Consumer issues a ForwardClose to notify the producer that the connection is being closed. In this case, the consumer number can be re-used immediately.

8.8.7 Recommendations for connection establishment

When a multipoint consumer decides to unilaterally close a connection, the only way for a producer to detect this event (other than getting a Forward Close from the consumer) is when it no longer receives a Time Coordination response for Timeout_Multiplier+2 Ping Intervals. During that time interval, the consumer number is still allocated to that consumer. There is a strong likelihood that while this connection is still open, that consumer will issue a new SafetyOpen. Producers have two options in this case:

- Search through the existing connections and see if a connection to this same consumer is already established and if so, re-issue the consumer number again.
- Allocate a new consumer and allow the old consumer number to time-out and go through quarantine.

The first option is the more complicated to code, but is the recommended option since it offers a more robust re-allocation scheme for consumer numbers. The advantage of the second option is that it is simpler to implement, but allows for consumer numbers to languish in quarantine longer than necessary. The second option can also make connection startup problematic for applications where the number of consumers is > 8 .

If option one is used, the recommended scheme to match up connections should be to match up:

- OUNID, and
- Application data path.

If these match up with the parameters of an existing connection, then that device's consumer number can be re-issued.

8.8.8 Ownership establishment

Subclause 8.9 discussed two methods of establishing connection ownership, through the software tools or from the SafetyOpen. It indicated that devices are required to support a facility to allow "tool only" configuration or originator configuration. It also implied that attempts to reconfigure a device via the SafetyOpen will be rejected if the configuration is locked. If the "tool only" state is not set, the OUNID of the selected originator can also be set by the software tool during target device commissioning for added security.

This subclause will define how this capability has been defined in the new Safety objects. The new Safety objects are required in devices that will be configured by the Safety Network Configuration Tool (SNCT).

To establish ownership of the configuration (and the connection in outputs), the OUNID of the originating device is passed to the target in the SafetyOpen message.

Safety Devices shall NV-store the OUNID for the configuration in the device. Also, Safety Devices shall NV-store the signature for the configuration (SCID) in the device. The SCID is derived by concatenating the SCCRC and the SCTS in the SafetyOpen.

Output devices shall NV-store the OUNID associated with the safety output connection.

When the user has tested and verified the tool-generated configuration, they can lock the configuration from changes and flag it as "Locked". The "Configuration Lock" flag is an attribute of the Safety Supervisor. To modify it, a password can be provided which provides user defined security.

Devices which are configured by the SafetyOpen cannot have a “Configuration Lock” condition. The assumption is the Lock condition is set and controlled at the connection originator.

8.8.9 Ownership use cases

8.8.9.1 General

This subclause will present a series of connection scenarios where configuration is also being attempted. Each scenario will provide a brief description of how the device should deal with each case. Most of these cases utilize the SafetyOpen to configure the device. As such, the tool-based configuration features such as the “Configuration Lock” attribute shall not be in use.

8.8.9.2 The user-designated owner connects and configures an un-owned, un-configured input device – (OUNID assigned at first connection)

The Configuration Lock attribute in the target shall be “unlocked” and the Safety Supervisor shall be in Configure mode; device has no configuration. The SafetyOpen is validated by the target device as being correct and the originator OUNID is stored in the target and related to the configuration. A point-to-point or multipoint connection is established. Normal operation starts.

The originator of a Form 1 SafetyOpen that configures a previously unconfigured and unowned input device shall become the owner of the configuration.

8.8.9.3 The user-designated owner connects and configures an un-owned, un-configured output device – (OUNID assigned at first connection)

The Configuration Locked attribute in the target shall be “unlocked” and the Safety Supervisor shall be in Configure mode; device has no configuration. The SafetyOpen is validated by the target device as being correct and the OUNID is stored in the target and related to the configuration and the connection. A point-to-point connection is established. Normal operation starts.

A Form 1 SafetyOpen that configures a previously unconfigured and unowned output device shall become the owner of both the connection and the configuration.

8.8.9.4 Owner connects and configures an owned, un-configured input device – (OUNID assigned by tool)

The Configuration Lock attribute in the target shall be “unlocked” and the Safety Supervisor shall be in Configure mode; device has no configuration. The SafetyOpen is validated by the target device as being correct and the SafetyOpen OUNID matches the stored OUNID. A point-to-point or multipoint connection is established. Normal operation starts.

8.8.9.5 Owner connects and configures an owned, unconfigured output device – (OUNID assigned by tool)

The Configuration Lock attribute in the target shall be “unlocked” and the Safety Supervisor shall be in Configure mode; device has no configuration. The SafetyOpen is validated by the target device as being correct and the SafetyOpen OUNID matches the stored OUNID for the configuration and the connection. The target stores the configuration. A point-to-point connection is established. Normal operation starts.

A second device attempts to connect to an owned point-to-point output connection

If no connections are open the SafetyOpen will be evaluated by the target device. If a Form 1 SafetyOpen is sent where the originator OUNID is different than the OUNID that is stored with

the connection the SafetyOpen shall be rejected and an error message, “O_UNID Mismatch” is returned

8.8.9.6 Owner connects and attempts to re-configure a tool-owned, configured device – (Configuration Lock set)

If an originator attempts to configure a device with the Configuration Lock attribute set, the device shall reject the SafetyOpen and return an error message “Configuration Not Allowed” is returned.

8.8.9.7 Device reconfiguration – Input Device

The owner of a device sends a SafetyOpen with configuration data plus an SCID (SCCRC and SCTS) to the target device. The data may or may not be the same as what exists in the device and the SCID may or may not match the existing SCID. Regardless, as long as the following process is followed, it should treat the message as a re-configuration.

A target input device that has an existing configuration shall (at a minimum) do the following when a Form 1 SafetyOpen is received:

- Verify that the TUNID in the SafetyOpen matches the device TUNID;
- If configuration data is included (Form 1), verify that the OUNID in the SafetyOpen matches the stored OUNID;
- Verify the Electronic Key matches the device;
- Calculate the SCCRC over the received data and confirm that it obtains the same value;
- verify the CPCRC over the configuration is correct;
- start its reconfiguration process (enters Config State), close and inhibit connections during the reconfiguration (respond to connection requests with an error response, “Invalid Mode/State”);
- update the configuration and SCID in NV Memory;
- verify and validate the connection parameters;
- Validate the application path;
- Confirm the safety connection requested is supported;
- Safety parameters are within valid ranges;
- transition the connection to the established state, and the device to the executing state;
- New connections can now be accepted.

A connection of the requested type is established. While new connections can again be accepted, these other devices shall now either have the new SCID, or connect with an SCID = 0.

8.8.9.8 Device reconfiguration – Output Device

The owner of a device sends a SafetyOpen with configuration plus an SCID to the target device. The data may or may not be the same as what exists in the device and the SCID may or may not match the existing SCID. Regardless, as long as the following process is followed, it should treat the message as a re-configuration.

A target output device that has an existing configuration shall (at a minimum) do the following when a Form 1 SafetyOpen is received:

- If the connection path is to an output resource, verifies that the OUNID in the SafetyOpen matches the OUNID for the connection;
- Verify that the TUNID in the SafetyOpen matches the device TUNID;

- Verify the Electronic Key matches the device;
- If configuration data is included (Form 1), verify that the OUNID in the SafetyOpen matches the stored OUNID for the configuration;
- Calculate the SCCRC over the received data and confirm that it obtains the same value
- verify the CPCRC over the configuration is correct;
- start its reconfiguration process by closing and inhibiting connections during the reconfiguration (respond to connection requests with an error response, “Invalid Mode/State”);
- update the configuration and SCID in NV memory;
- verify and validate the connection parameters;
- Validate the application path;
- Confirm the safety connection requested is supported;
- Safety parameters are within valid ranges;
- transition the connection to the established state, and the device to the executing state.

A connection of the requested type should now be established.

8.8.9.9 Connection establishment to a previously owned device (Changing OUNID)

When attempting to configure a device that has another owner, a SafetyOpen configuration request (i.e., Form 1) without a matching OUNID will be rejected with an error response, “OUNID Mismatch”.

To clear an existing OUNID, safety targets shall support a “reset to out-of-box” using a reset switch on the device or using the special reset command defined in the Safety Supervisor.

If the device has a Safety I/O connection when a reset command is received, the reset command shall be rejected and an error message returned, “Invalid Mode/State”. In order for a device to be reset by command it shall not have Safety I/O connections established.

A second device attempts to connect to an owned multipoint connection

A second originator sends a SafetyOpen (i.e., Form 2a, or Form 2b) to a device that has a multipoint connection owned by another originator. If the SCID in the SafetyOpen is non-zero, the SCID is compared to the saved SCID. If the signatures match, the connection is established. If the signature does not match, the error response “SCID Mismatch” is returned. The owner does not need to be active during this process.

A target device that has an existing configuration shall (at a minimum) do the following when a Form 2a or 2b SafetyOpen is received:

- If the connection path is to an output resource, verifies that the OUNID in the SafetyOpen matches the OUNID for the connection;
- If the SCID is non-zero, confirm the SCID matches the device SCID;
- Verify that the TUNID in the SafetyOpen matches the device TUNID;
- Verify the Electronic Key matches the device;
- verify the CPCRC over the configuration is correct;
- verify and validate the connection parameters;
- Validate the application path;
- Confirm the safety connection requested is supported;
- Safety parameters are within valid ranges;

- transition the connection to the established state.

8.8.10 PID/CID usage and establishment

The PID/CID seeding mechanism has been defined to detect message insertion from an invalid source. This insertion can occur from another producer sending Safety Data, another producer sending Time Correction messages, or an invalid consumer sending Time Coordination messages. All these cases will be covered by the PID/CID scheme in the Safety Protocol. The key to this is the exchange of PID/CID information during the connection establishment process. The Producer will provide its PID to all its consumers, and all the consumers shall provide their respective CID back to the producer. This exchange occurs regardless of who originates the connection, or what type of safety connection is being established.

During connection establishment the producing and consuming nodes shall exchange their respective PID/CID information (Vendor ID + Device Serial Number + Connection Serial Number). The originator sends its ID in the SafetyOpen request, and the target returns its ID as part of the SafetyOpen Success response. The two cases are defined in Figure 59.

The PID shall be used to seed Data production CRCs and Time Correction CRC, and the CID shall be used to seed the Time Coordination CRC.

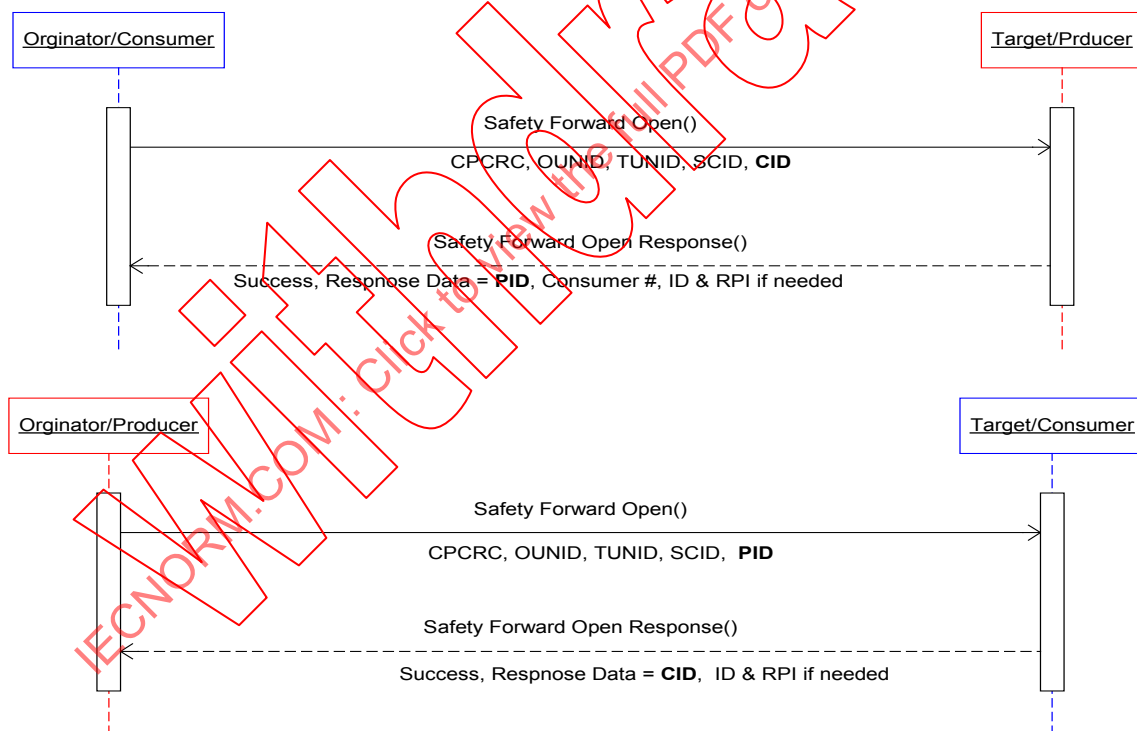


Figure 59 – PID/CID exchanges for two originator scenarios

8.8.11 Proper PID/CID usage in multipoint and point-to-point connections

The diagram shown in Figure 60 illustrates how the Producer and Consumer ID parameters are to be used to generate the CRC seed values. This diagram defines the multipoint case, but it also applies to Point-to-point connections as well (this is done simply for consistency). The producer seeds are generated using only the S1 and S3 generators (i.e. same seed used for S1 and S2 CRC calculations) and the CID seed is generated with the S3 generator.

Multi-cast connection establishment PID/CID handling

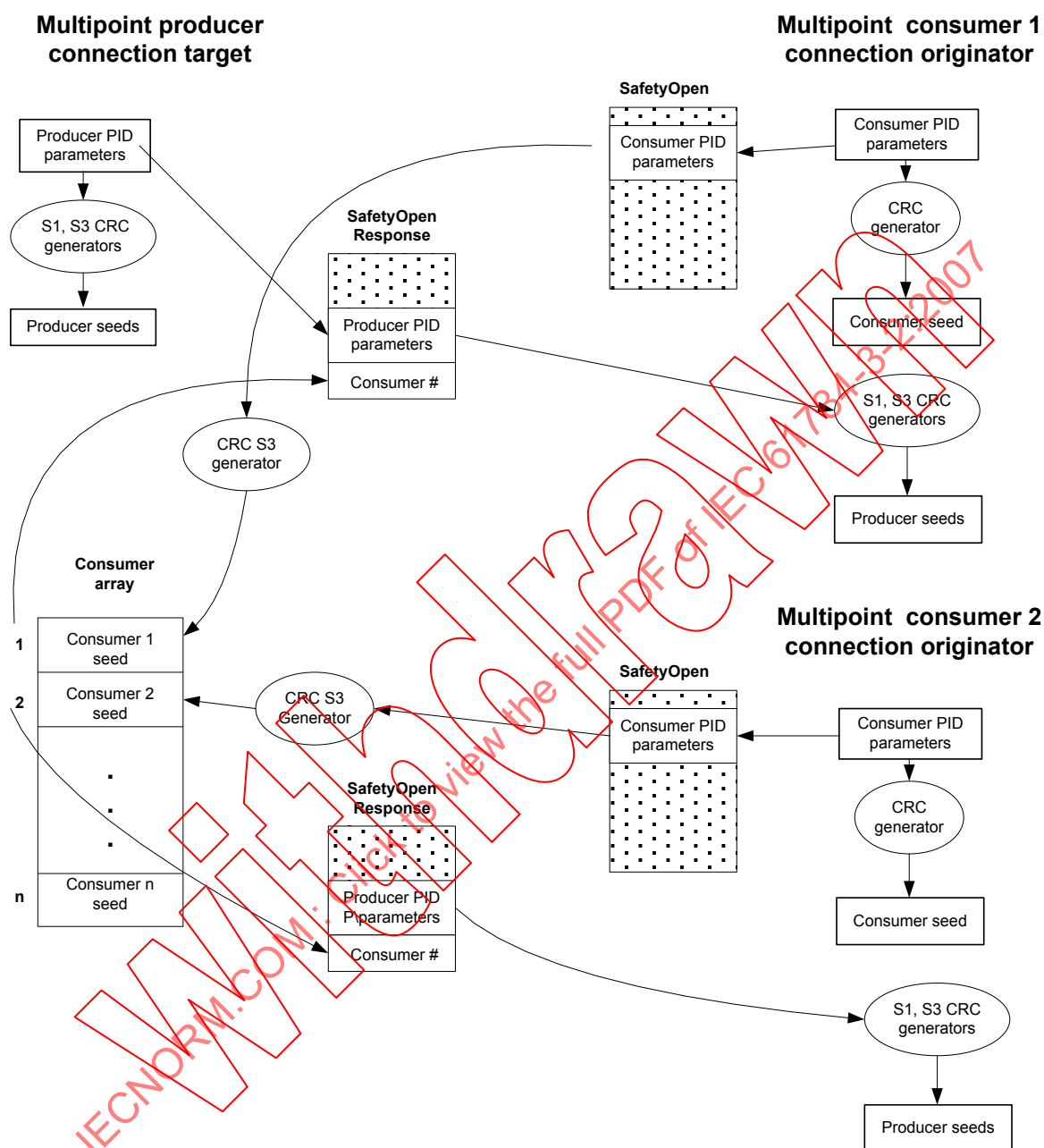


Figure 60 – Seed generation for multipoint connections

PID parameters shall be used to seed the CRCs for Produced Data messages and Time Correction messages, and CID parameters shall be used to seed the CRC for the Time Coordination responses.

As shown in the figure, multipoint producers will need to maintain a consumer array to keep track of the seed value for each allocated consumer number. The scheme shall also apply to Point-to-point connections except that only one consumer seed value needs to be maintained.

The diagram shown in Figure 61 illustrates how the seed values, set up during connection establishment, get used in the run-time data production.

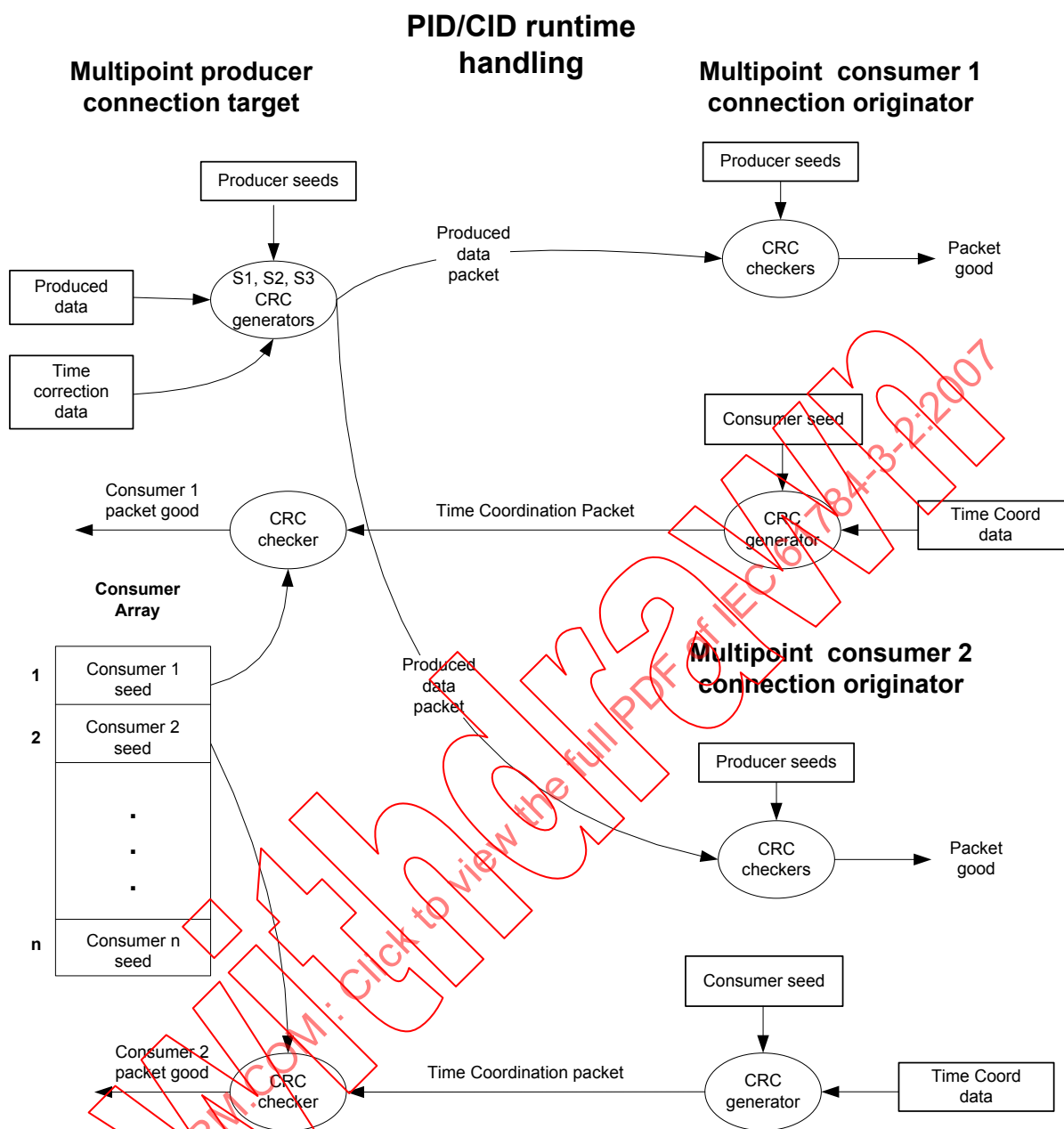


Figure 61 – PID/CID runtime handling

8.8.12 Network supported services

The purpose of this subclause is to promote compatibility between devices developed to exchange safety data over the FSCP 2/1 Safety Networks. It defines guidelines on which safety services should be supported by the different safety device types.

The goal is to allow the development of input and output devices which range from very simple discrete I/O to smart high functionality specialty I/O modules. When an input or output module reach a certain level of functional requirements, they may need to be viewed as embedding a safety Logic Device within the module.

8.8.13 FSCP 2/1 Safety device type

8.8.13.1 General

To simplify the exchange of information and presentation of Safety Data to the users of multiple systems, the devices that participate on the safety network are divided into 4 device types. The device types are:

- Input Devices,
- Logic Devices,
- Output Devices, and
- I/O Devices.

Input Devices are devices that translate user signals into safety data and send the safety data to Logic Devices. Input Devices do not receive safety data other than the safety connection status. If a device with input information needs to receive safety data from the other side of the safety connection, it should be considered an I/O Device or a Logic Device, and follow the guidelines of the I/O or Logic Safety Device Type.

Logic Devices are devices that process Safety Input Data and Produce Output Data. Direct communication between Input Devices and Output Devices is not supported. A Logic Device, no matter how simple, has to be between an Input Device and an Output Device.

Output Devices are devices that present some signal to the user's environment based on the safety data consumed by the device. Output Devices may have status that is produced back to the Logic Device.

I/O Devices are devices that have both Safety Inputs and Safety Outputs.

8.8.13.2 Safety connection category

All safety connections can be placed in one of four categories. They are:

- Input_Device_To_Logic_Device;
- Logic_Device_To_Output_Device;
- Logic_Device_To_I/O_Device (an Output device with Status is considered an I/O Device);
- Logic_Device_To_Logic_Device

Logic devices may choose to support any or all of the safety connection categories. If a device does support a safety connection category, it should support all the recommended services of that category.

8.8.13.3 Safety connection services

Table 92 defines all the defined Safety Validator Services that Connection Targets or Connection Originators can support.

Table 92 – Originator/Target service mapping

Connection originator or target	Safety connection types		Validator type
	Type	Use	
Target	Point-to-point	Producer	Type 1
Target	Point-to-point	Consumer	Type 2
Target	Multipoint	Producer	Type 3
Originator	Point-to-point	Producer	Type 1
Originator	Point-to-point	Consumer	Type 2
Originator	Multipoint	Consumer	Type 4

Table 93 defines the Safety Validator Services that Connection Targets or Connection Originators should not support. Multipoint producers are not required to initiate connections to its consumers; the consumers shall connect to it; Multipoint Consumers should be the connection Originator, and Multipoint Producers should be connection targets.

Table 93 – Unsupported originator/target service types

Connection originator or target	Safety connection types		Validator type
	Type	Use	
Target	Multipoint	Consumer	N/A
Originator	Multipoint	Producer	N/A

8.8.13.4 Services supported for each category

Figure 62 and Figure 64 list the preferred Safety Connection Services for various Connection Categories (see Figure 63 and Figure 65 for further clarification). For each Safety Connection Category, the different columns represent preferred methods of establishing safety connections between the two devices. For any particular device, only one column would be supported at one time.

					Supported Services for Input or Output Type					
Safety Connection Category					1		2	3		
Safety Connection Service					Input Device To Logic Device		Logic Device To Output Device	Logic Device To I/O Device (Output with Status)		
Originator or Target	Safety Connection Type									
					Op 1	Op 2		Op 1	Op 2	Op 3
Input and Output Devices	Target	Point-to-point	Producer			O			O	
	Target	Point-to-point	Consumer				R	R	O	
	Target	Multipoint	Producer		R			R		
Logic Devices	Originator	Point-to-point	Producer				R	R	O	
	Originator	Point-to-point	Consumer			O			O	
	Originator	Multipoint	Consumer		R			R		

Key

Op1, 2, 3 = Optional Combinations

R = Recommended Services for compatibility

O = Optional Service

Figure 62 – Connection categories and supported services

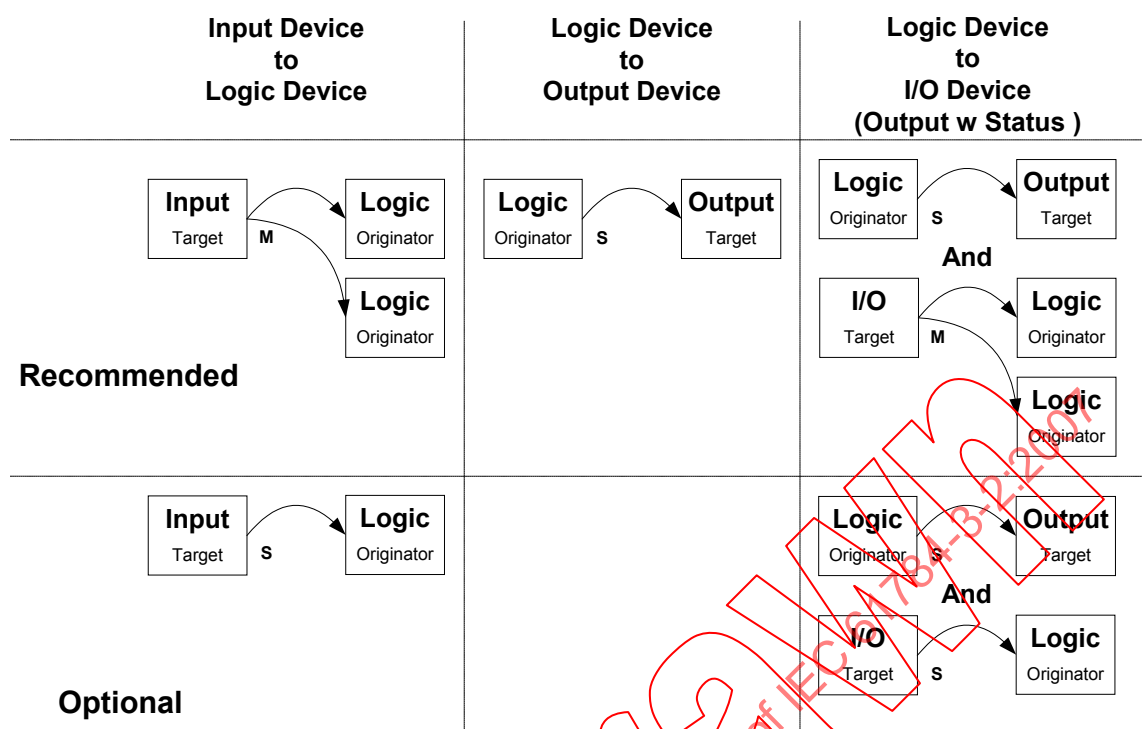


Figure 63 – Recommended connection types

Safety connection service				Safety connection Category 4			
	Originator or Target	Safety Validator type		Logic device to logic device			
				Supported service combinations			
				Op 1	Op 2	Op 3	Op 4
Logic device target	Target	Point-to-point	Producer			O	
	Target	Point-to-point	Consumer		O		
	Target	Multipoint	Producer	R			
Logic device originator	Originator	Point-to-point	Producer		O		
	Originator	Point-to-point	Consumer			O	
	Originator	Multipoint	Consumer	R			

Key

Op1, 2, 3 = Optional Combinations

R = Recommended services for compatibility

O = Optional service

Figure 64 – Logic-to-logic supported services

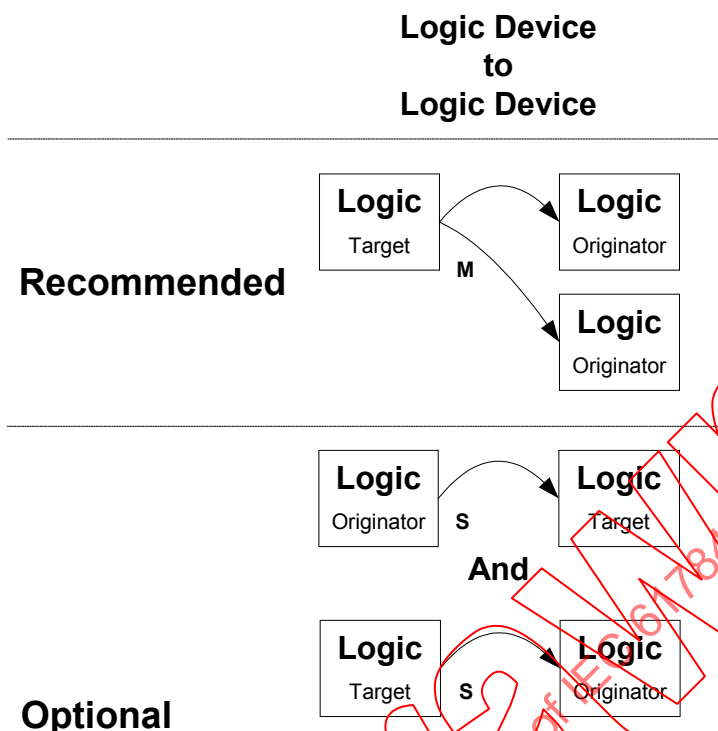


Figure 65 – Recommended connection types for logic to logic

8.9 Safety configuration process

8.9.1 Introduction to safety configuration

The purpose of this subclause is to identify the Safety Network Configuration process and the configuration data flows through the safety system. This subclause defines the system and the software requirements of the Safety Network Configuration interface and how data is protected to ensure the safety system meets SIL3.

The configuration process is comprised of two parts: 1) download and 2) testing and verification.

Download Process:

During the download portion of the configuration process the configuration data is transferred from one of the following two sources:

- The Safety Network Configuration Tool (SNCT) to the device being configured;
- From the originator to the target device.

Verification Process:

- During the verification stage the user will functionally test the device in the system, and visually verify that the correct configuration data has been transferred to the device being configured.
- Together the protections in these two parts ensure that devices are suitably configured for use in a SIL3 environment.

8.9.2 Configuration goals

The safety related goals for the configuration process for FSCP 2/1 safety devices are shown in Table 94.

Table 94 – Configuration goals

Goal	Description
Goal 1	To ensure with high integrity that the intended configuration created by the user with the software was correctly sent to and saved in the safety device
Goal 2	To have a common download process that is the same for the transfer of configuration data to all safety devices
Goal 3	To have a common download process that is the same for the transfer of configuration data from a software tool or an originating device during configuration
Goal 4	To ensure with SIL3 integrity that a target device has the intended configuration prior to opening a safety connection. The measures used during configuration download should be reused in the safety connection establishment process
Goal 5	To ensure unique identification of configuration with or without data being included within a connection establishment

8.9.3 Configuration overview

Safety configuration begins with the downloading of configuration data into a safety device. Figure 66 shows the possible configuration data transfers into safety devices. The numbers in the figure are references to the discussion numbering below.

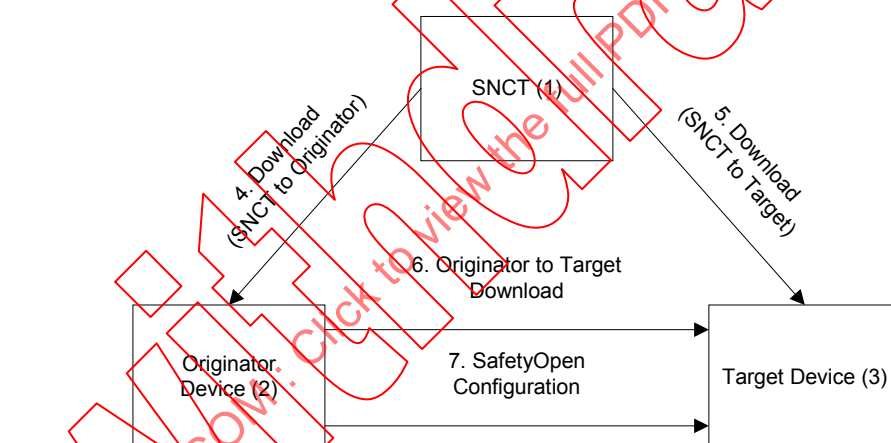


Figure 66 – Configuration data transfers

The devices involved in the configuration process are shown in Figure 66. The functions of items 1, 2, and 3 are described as follows:

- The SNCT (1) Safety Network Configuration Tool and the process described in this part allows this tool to be non-safety critical;
- The Originator (2) is a device that is capable of configuring Target devices (3) while initiating connections to them. Because these devices can hold configuration data for Targets (for targets that are not configured by the SNCT directly), they have additional procedures to complete the configuration process;
- The Target device (3) is one that can be either configured directly by the SNCT or can be configured by an Originator (2) owner. The ownership concept plays an important role in assuring the integrity of “originator configured” targets.

There are two basic configuration procedures (see Figure 66). The configuration procedures are described as follows:

- SNCT-to-Originator (flow 4) and SNCT-to-Target (flow 5). This procedure is one that configures a single device and these devices hold no configuration data for other devices;

- SNCT-to-Originator-to-Target (flow 4 plus flow 6 or 7). This special procedure is for configuring an originator device that holds configuration data for a target;
- Flow 6 is a procedure that duplicates the sequence of operations that would otherwise be done in flow 5;
- Flow 7 is a procedure by which the configuration is performed as part of the processing of a SafetyOpen.

There are two ways that a replacement device can be reconfigured from the originator and they are described as follows:

- Flow 6 is a procedure that duplicates the sequence of operations that would otherwise be done in flow 5;
- Flow 7 is a procedure by which the configuration is performed as part of the processing of a SafetyOpen.

8.9.4 User configuration guidelines

This subclause describes user configuration guidelines that assure a FSCP 2/1 safety system is properly configured and protected. These guidelines are used to define the safety manual requirements.

The user configuration data flows are defined in Figure 67.

- a) The FSCP 2/1 safety system user shall assure that SNCT-configured-devices (2) are locked after verification.

Devices that can be configured via the SNCT interface shall include a safety manual advisory that instructs the user to lock the device after verification has been completed.

- b) The FSCP 2/1 safety system user shall assure that all originator-configured-devices (3) have ownership (7) assigned to the intended originator.

Devices that can be configured by a Form 1 SafetyOpen shall include a safety manual advisory that instructs the user to verify that all originator-configured safety devices have their ownership assignments as part of the final verification process.

- c) The user shall visually confirm (4) the sent configuration data and confirm it was downloaded correctly.

All FSCP 2/1 safety devices shall include a safety manual advisory that instructs the user to visually verify that all configuration data was downloaded correctly.

- d) The user shall functionally test their application.
- e) The user shall assure the SCID (5) check by target and originator is done.

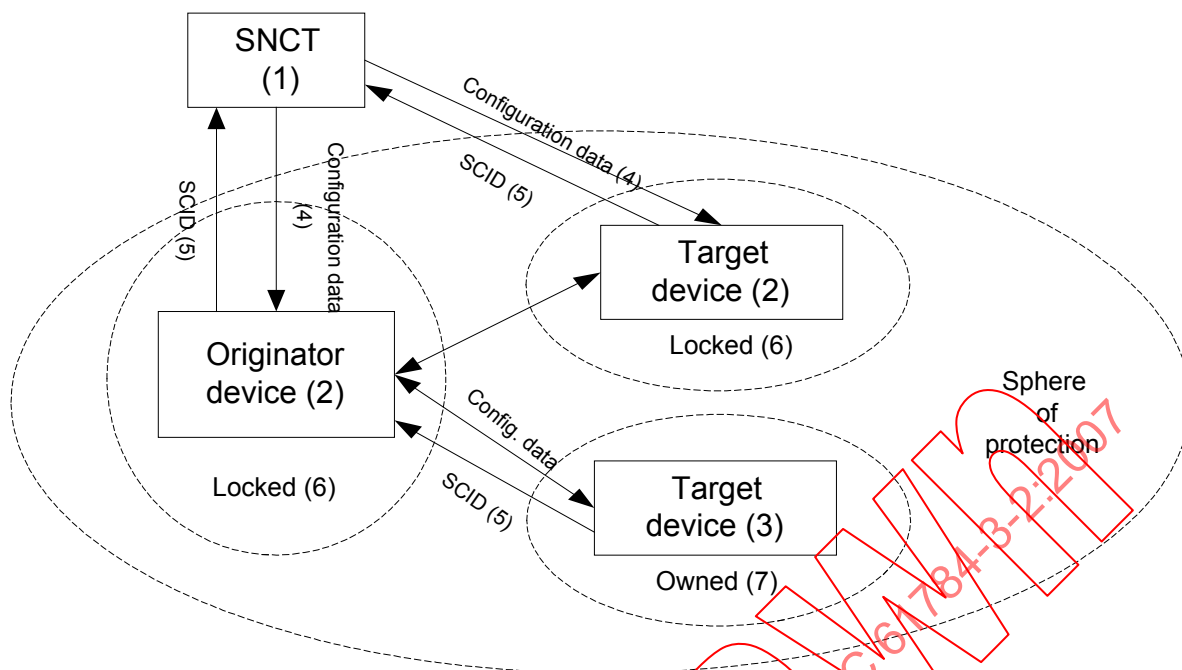


Figure 67 – Protection measures in safety devices

8.9.5 Configuration process SIL3 justification

The FSCP 2/1 safety device being downloaded has safety measures defined which ensures integrity of data and data transfers as specified by IEC 61508 SIL3.

The SNCT on the other hand, will not be certified due to the limitations of personal computer hardware, the operating system utilized and the development tools. Because of these factors, the techniques employed to configure a safety device require that the safety device and user procedures, not the workstation software, enforce the protection measures.

SIL3 integrity is achieved by meeting the following requirements:

- A receiving SIL3 device shall calculate a SCCRC with SIL3 integrity and compare it to the SCCRC within the SCID;
- For Form 1 configured or tool configured devices, configuration validation of the downloaded safety-relevant data from the software (or originator) shall be performed with SIL3 integrity;
- When a SIL3 device is configured directly from a workstation, the device safety manual shall instruct the user to compare the transferred SCID and configuration data with the SCID and configuration data originally viewed in the workstation;
- The SNCT software shall provide a facility to allow the user to confirm downloaded configurations are correct;
- SIL3 devices shall store/restore its configuration data with SIL3 integrity;
- SIL3 devices shall handle all device state transitions with SIL3 integrity;
- Sub-systems that are required to be examined and certified to SIL3;
- Target device configuration processing;
- Target device configuration storage;
- Target device configuration validation;
- Target device modes.

8.9.6 Device functions for tool configuration

The process by which the SNCT interacts with originators and targets are controlled by facilities defined in the Safety Supervisor object. Features such as Password protection and the Configuration Lock attribute combine to provide security for this interface. Since the Safety Supervisor is provided in the baseline profile (see 6.6) for both safety targets and originators, the interface details are common. This subclause will define these features and how they are used by the SNCT.

8.9.7 Password security

The SNCT interface defines the following functions to optionally provide password security:

- Password functionality added.
Support for one password shall be implemented by devices supporting the SNCT interface.
Passwords submitted to devices supporting the SNCT interface shall be values obtained from running the encryption algorithm defined in 7.1.2.3.
The default password value in devices supporting the SNCT interface shall be zero
Software shall transparently insert “zero” in all services which require one until a user sets it to something different.
- The SNCT interface shall support a Password service to set passwords which takes 2-password parameters; Old PW and New PW
- The SNCT interface shall provide a Reset Password service which takes a vendor specific field to reset the password.
Each vendor shall choose a method to reset passwords and instruct the user what to fill in to cause the device to reset the password. The software shall only insert whatever value the user enters into the reset field (see the Reset_Password service in the Safety Supervisor object definition in 6.6).
The safety supervisor commands that have a password parameter define the processing requirements and error responses for these parameters.

8.9.8 SNCT interface services

The SNCT shall provide the following services.

- The SNCT interface shall provide a Configure Request service which requires password, TUNID, and OUNID to execute.
- In the SNCT interface, an unowned device shall capture the OUNID as the device owner when the first Configure_Request is received.
- The SNCT interface shall provide a Validate Configuration service which requires the SCID, SCCRC, and the SCTS to execute.
- The SNCT interface shall provide an Apply service that causes the device to save the configuration to NV memory.
- The SNCT interface shall provide a special Reset service which requires the password and TUNID to execute and supports the 2 common Reset types along with one that can reset to the default but preserve the password.
- The SNCT interface defines an optional a Mode Change service which, if implemented, shall require a password to execute.

8.9.9 Configuration lock

The SNCT interface defines the following functions for Configuration Lock. Its intent is to provide a way for the user to flag that a tool-generated configuration has been tested and verified as valid. From that point, the signature (SCID) reflects a verified configuration. The configuration lock has the following characteristics.

- The SNCT interface shall provide a Configuration Lock service which requires the password and TUNID to execute.
- The SNCT Configuration Lock Attribute shall have a default value of “Unlocked”.
- Safety devices which support the SNCT interface shall require that the “Configuration Lock” attribute be cleared before any command to change a safety device to the Configure State (Configure_Request) will be accepted.
- The Configuration Lock has two purposes:
 - Serves to indicate that the user has tested and confirmed the Tool-based configuration in this device is correct.
 - “Tool Only” mode requirement. When the Configuration Lock attribute is set, the device shall reject any Form 1 SafetyOpen messages.

These two features make it clear that for the SNCT to configure any safety node, it shall support features to set and clear the Configuration Lock attribute, and shall present the proper password to do so. These features are illustrated in the sequence diagram defined in Figure 70.

8.9.10 Effect of configuration lock on device behavior

Figure 68 defines the effect that the “configuration locked” condition has on the ability of a device to be configured; there are requirements defined in the Safety Supervisor object that cause safety devices to reject all attempts to change a locked configuration.

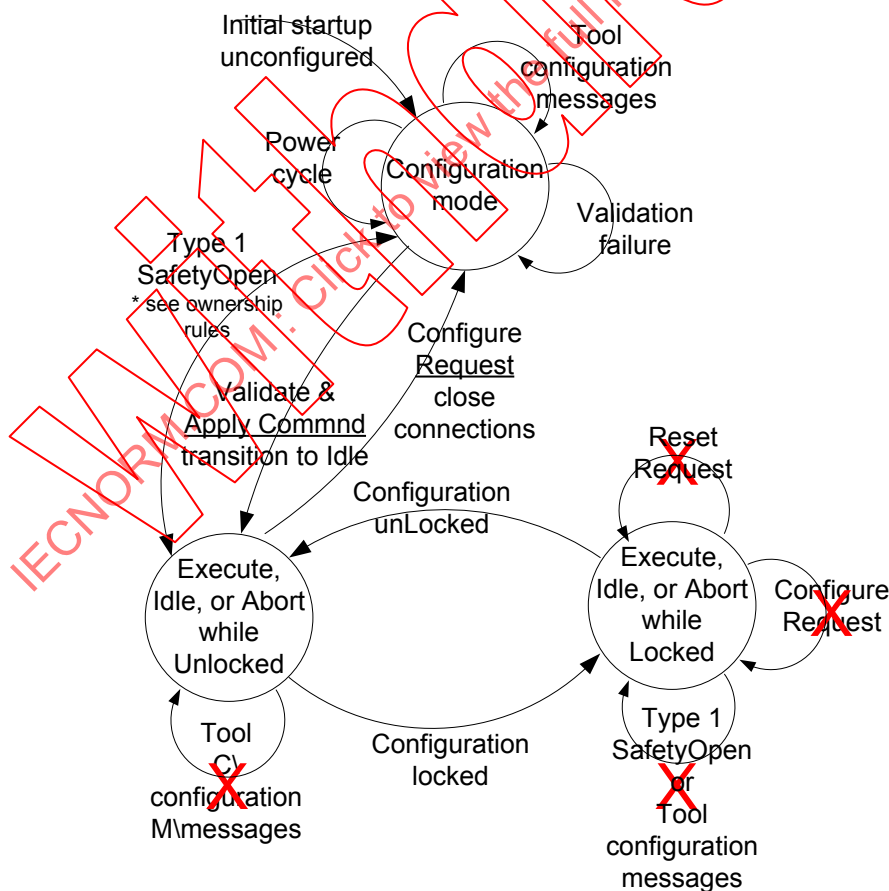


Figure 68 – Configuration, testing and locked relationships

Table 95 defines the effect that ownership has on a device's ability to be configured. It shows that owned devices can only be Form 1 configured by the originator that has the owner UNID. A locked device can never be configured regardless.

Table 95 – Configuration owner control vs. device state

Ownership rules	Configure mode	Execute/Idle Mode Unlocked	Execute/Idle Mode Locked
Configuration Un-owned	Tool configuration or Form 1 acceptance by anyone	Tool configuration or Form 1 acceptance by anyone	No Tool configuration, No Form 1 accepted
Originator Owned	Tool configuration or Form1 by owner only	No Tool configuration (shall reset first) or Form1 by owner only	No Tool configuration, No Form 1 accepted
Software Tool Owned	Tool configuration only No Form 1 accepted	Tool configuration only No Form 1 accepted	No Tool configuration, No Form 1 accepted

8.9.11 Configuration ownership

The SNCT interface defines functionality for handling the configuration owner identifier for a device. This function has been defined to allow a device to provide exclusive configuration rights to a particular configuration source. The source could also be assigned exclusively to a configuration tool. The following functions are provided:

- Configuration UNID attribute
The default is to allow any device to be an owner
Device are required to capture the first configuration source as the owner
Attributes and services in the Safety Supervisor are defined which allow Tools to be the exclusive configuration source
- Even though an owner has been assigned as a configuration source, the SNCT tool has the ability to override a configuration. The SNCT shall provide protection measures to assure the user cannot perform an override without warnings and the proper password.

8.9.12 Configuration mode

All safety devices shall support a non-volatile configuration mode; this means that once a device is commanded to enter the configuration mode it shall remain in the configuration mode until the configuration is successfully validated.

8.9.13 Measures used to ensure integrity of configuration process

8.9.13.1 Safety Configuration Identifier (SCID)

In safety systems it is necessary to clearly identify the configuration. The identification is used during several operations:

Download from tools –This gives the user the ability to check that the tool and device agree on the information downloaded.

Device Replacement – If the tool is used for device replacement, the user has the ability to verify that the configuration is the correct one to download. If the originator is used to

automatically reconfigure a device, the SCID is used to indicate that a reconfiguration is necessary and ensure the integrity of the operation.

Connection Establishment – The originator and target use the SCID to ensure that both devices are using the same configuration.

The SCID is a concatenation of the SCTS and the SCCRC described in 8.9.13.3 and 8.9.13.4 respectively.

8.9.13.2 Originator and Target SCID coverage

When originators are used to configure targets, Figure 69 illustrates how the SCIDs provide coverage of the various configurations. The SCID for targets that are configured by originators are not calculated in the originators, the calculation is performed in the workstation. The target configurations are protected by their respective SCIDs.

Originator Config		Originator SCID
Target 1 Config	Target 1 SCID	
Target 2 Config	Target 2 SCID	
Target 3 Config	Target 3 SCID	

Figure 69 – Originator's configuration data

8.9.13.3 Safety Configuration CRC (SCCRC)

This is a CRC that covers the device configuration data that is contained in a SafetyOpen. It is part of the SCID for the device. The SCCRC is described in 8.2.

The safety function of the SCCRC is used to ensure the correctness of the safety-related application data.

8.9.13.4 Safety Configuration Timestamp (SCTS)

The Safety Configuration Time Stamp is a safety related signature that identifies the revision of the device configuration contained in either a SafetyOpen or in a safety device. It is part of the SCID for the device configuration.

The safety function of the SCTS is to ensure the uniqueness of the safety-related application data.

8.9.13.5 System-wide unique "Safety Network Number" (SNN)

The Safety Network Number uniquely identifies a network across all networks in the safety system. The end user is responsible for assigning a unique number to each safety network or safety sub-net within a system.

The safety function of the UNID is to uniquely identify originators and/or targets during the connection establishment process.

8.9.13.6 System-wide "Unique Node Identifier" (UNID)

Uniquely identifies a node across all networks in the safety system. This value is made up of the Safety Network Number (SNN) and the Node Address of the device. This value is a structure consisting of the 6-octet SNN and the 4-octet Node Address. The 4-octet node address is sized to accommodate all forms of CPF 2 network addresses.

```
struct UNID
{
    S_SNN SNN;
    UDINT MACID;
};
```

Maclds smaller than 4-octets (i.e. CP 2/3) shall be aligned to the least significant octet in little endian order. The unused octets shall be zero filled.

The safety function of the UNID is to uniquely identify originators and/or targets during the connection establishment process.

Other uses of UNID:

- TUNID = Target's Unique Network Identifier. The safety function of the TUNID is to identify the target in the SafetyOpen and Configuration mode message
- OUNID = Originator's Unique Network Identifier. The safety function of the OUNID is to identify the originator in the SafetyOpen.
- OCPUNID = Output Connection Point Owning UNID. The safety function of the OCPUNID is to identify the owner of the output connection in the target. Set to the OUNID of the originator in the SafetyOpen that obtains ownership of the output connection point. Used to prevent an unauthorized originator to obtain ownership of an output connection point.
- CFUNID = Configuration Owning UNID. The safety function of the CFUNID is to identify the owner of the safety configuration in the target, this should be equal to the OUNID in normal operation. This is set to the OUNID of the originator in the SafetyOpen that includes configuration data and is used to prevent an unauthorized originator from modifying the configuration.

8.9.13.7 Connection Parameters CRC (CPCRC)

The Connection Parameter CRC is a safety related CRC covering the CPF 2 connection establishment data in the SafetyOpen. This CRC will also cover the additional parameters required by the safety protocol. The CRC is defined in 7.1.2.3

The safety function of the CPCRC is to ensure the integrity of the connection data contained within the SafetyOpen.

8.9.14 Download process

8.9.14.1 Overview

The safety download process ensures that the configuration data is transferred with high integrity from the SNCT or the Originator to the device being configured. At the successful completion of the download process the configuration is considered valid. There are two download processes used with safety devices:

- SNCT downloads to originators and targets;
- Originator downloads to targets using a Form 1 SafetyOpen.

The following subclauses will describe the details of the processes that are used for the download of safety devices.

8.9.14.2 SNCT download to originators and targets

Figure 70 show the process of downloading a safety device from the SNCT. The download process between the SNCT and the Configured Device assures that the data is instantiated in the Configured Device's memory with high integrity.

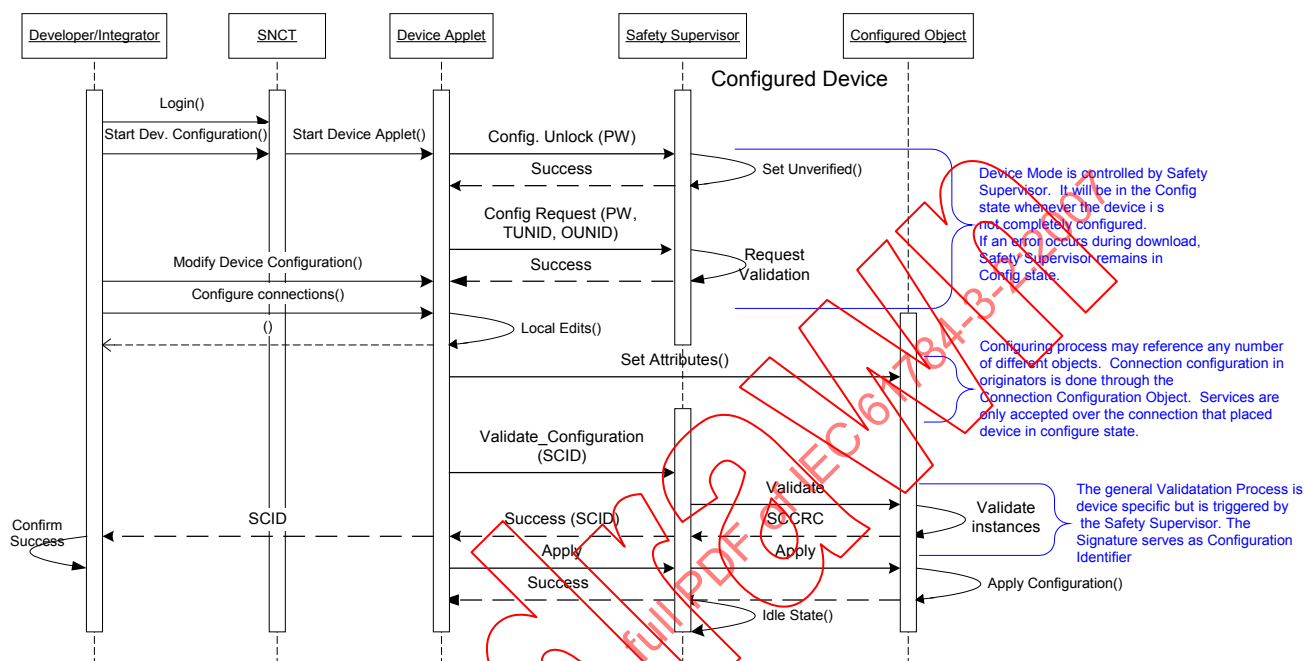


Figure 70 – SNCT to device download process

8.9.14.3 SNCT to device download process steps

The following are the process steps for the SNCT to device download process as shown in Figure 70.

- a) Clear Configuration Lock Attribute
 - i) Device: Configuration Lock Attribute cleared.
- b) Enter configuration state with password, TUNID, OUNID
 - i) Device: No safety I/O connections can be active.
 - ii) Device: Device saves Config mode to NV storage
 - A) Until “Validate” and “Apply”, power cycles go back to config mode
- c) Transfer of configuration data
 - i) Tool: Calculate “proposed SCCRC” for data (SCCRC part of SCID)
 - ii) Device: Any explicit messages are allowed to set configuration in device, but device shall only accept safety-related explicit messages over the connection that put it into the configuration mode.
- d) Validate configuration data
 - i) Tool: Send Validate_Request with SCID (which contains proposed SCCRC)
 - ii) Device: Analyze configuration data
 - iii) Device: Calculates SCCRC value and compares to “proposed SCCRC”
 - iv) Device: send SCID in Validate_Reply
 - v) Tool: Compares returned SCCRC to “proposed SCCRC”

e) Apply Configuration

- i) Device: Save configuration to NV storage with SCID
- ii) Device: Mark configuration as valid.
- iii) Device: Set mode to Idle
- iv) Tool: Capture and store the SCID with the configuration file.

8.9.14.4 SNCT downloads to originators which do Form 1 target configuration

Figure 71 show the process by which Originators are initially configured and how the Form 1 SafetyOpen is used to download Target devices. Once the user finishes defining the Originator and Target configurations in the SNCT, it is sent to the Originator. At this point the originator SCID can be confirmed by the user. This process ensures that the configuration data in the SNCT and originator match with a high degree of integrity.

Then the user shall place the Originator into execute mode to get the Target devices downloaded. If the connection is successful, the Target configuration/SCID was accepted by the target and confirmed by the originator. This action also established the Originator as the Target's owner and prevents the Target from being downloaded by any other device except the SNCT.

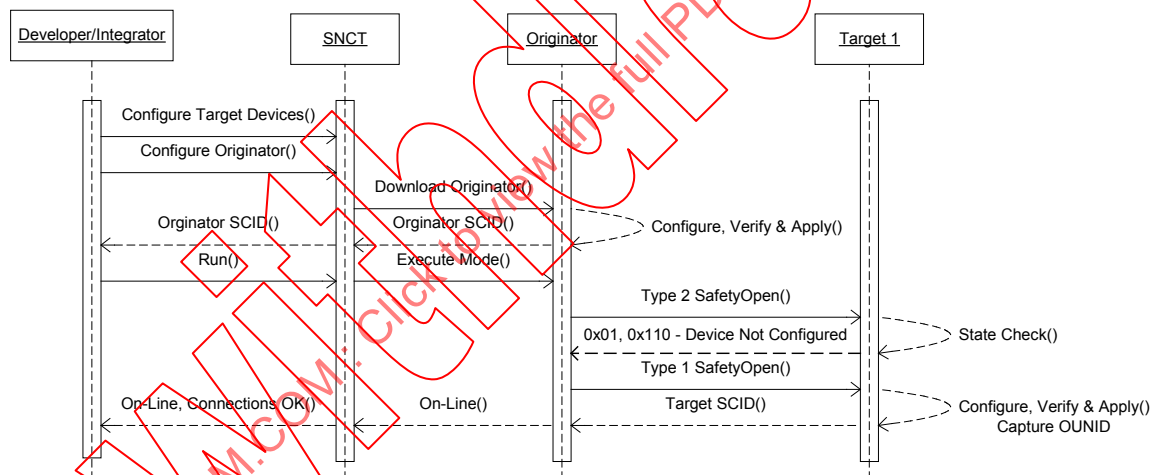


Figure 71 – SNCT Downloads to originators that perform Form 1 configuration

8.9.14.5 SNCT downloads to originators that perform Form 1 configuration

The following are the process steps for SNCT Downloads to originators that perform Form 1 configuration as shown in Figure 71.

- a) Clear Configuration Lock Attribute on originator
 - i) Device: Configuration Lock Attribute cleared.
- b) Enter configuration state with password, TUNID, OUNID
 - i) Device: No safety I/O connections can be active.
 - ii) Device: Device saves “Configuring” mode to NV storage
 - iii) Until “Validate” and “Apply”, power cycles go back to “Configuring mode
- c) Transfer of configuration data
 - i) Tool: Calculate “proposed SCCRC” for data (SCCRC part of SCID)

- ii) Tool: Any explicit messages are allowed to set configuration in device, but device shall only accept safety-related explicit messages over the connection that put it into the configuration mode.
- d) Validate configuration data
 - i) Tool: Send Validate_Request with SCID (which contains proposed SCCRC)
 - ii) Device: Analyze configuration data
 - iii) Device: Calculates SCCRC value and compares to “proposed SCCRC”
 - iv) Device: send SCID in Validate_Reply
 - v) Tool: Compares returned SCCRC to “proposed SCCRC”
- e) Apply Configuration
 - i) Device: Save configuration to NV storage with SCID
 - ii) Device: Mark configuration as valid.
 - iii) Device: Set mode to Idle
 - iv) Tool: Capture and store the SCID with the configuration file.
- f) Set Mode to Run
 - i) Originator: Sends Form 2 SafetyOpens (see 8.8.5)
 - ii) Target returns one of the following errors:
 - A) Unconfigured: 01_{hex}, 0110_{hex} – Device Not Configured
 - B) Prior configuration: 01_{hex}, 80C_{hex} - SCID Mismatch error
 - iii) Originator: Sends Form 1 SafetyOpen
 - iv) Targets: Verify and Apply the configuration data
 - v) Targets: Return target SCID# to originators
 - vi) Originator: Compare sent and received SCID#’s
 - vii) Originator: Return On-Line status to SNCT

8.9.15 Verification process

8.9.15.1 Verification process overview

The verification process uses the data that was transferred during any of the download processes to allow the user to correctly verify that the downloaded data is correct, the system operation is correct and the system is ready for SIL 3 operation. There are three main parts to the configuration process:

- Configuration Data Verification;
- Functional testing;
- Configuration Locking.

Key requirements for the safety manual:

- The device Safety Manual shall contain an advisory that user testing is the means by which all downloads are validated;
- The device Safety Manual shall contain an advisory that the signature should only be considered “verified” (and configuration locked) after user testing.

The device User manual shall contain an advisory that configuring an originator with connection data and/or target configuration data shall be downloaded to the target so it can be tested and verified. Only then can SCIDs from the target be confirmed.

As shown in Figure 72, when originators are tool-configured and targets are originator configured, only the originator is “locked” after verification and testing. The target in this case uses ownership assignments for protection. Also shown in Figure 72 is that Targets can also be tool-configured as well. In this case, these targets are also “locked” after verification and testing.

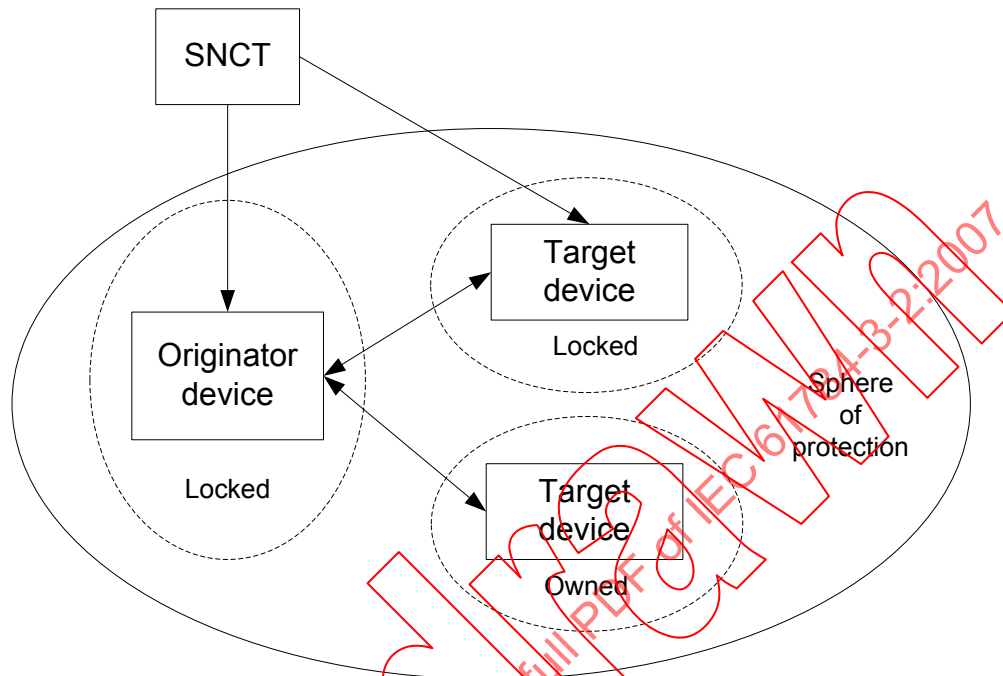


Figure 72 – Protection from locking and ownership

8.9.15.2 User configuration verification and alternatives

8.9.15.2.1 Overview

The user needs to visually verify that the downloaded configuration is correct and is the intended configuration for the device. This subclause presents four alternatives that will fulfill the requirements for user verification of the device configuration data. In all cases the user needs to verify the functionality of the devices being configured through some kind of functional testing.

One or a combination of the following options can be used to fulfill the user verification requirement: user comparison of the original software display, and results of the read back that have been displayed by a diverse means.

8.9.15.2.2 Alternative 1 – immediate read back and diverse comparison

The device configuration data is read back by the work station and presented to the user utilizing a diverse execution path in the workstation, this is shown in Figure 73.

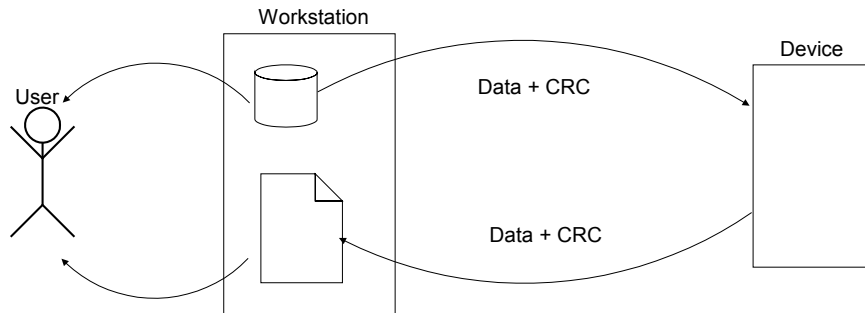


Figure 73 – Example of read back and comparison of original and printout

Reading back the entire configuration, comparing it to what the software has and providing a printout of the device configuration for the user to compare with what was previously entered. The user shall examine and compare the printout to the original data to satisfy the integrity requirements. Any automated comparison by the software tools is a diagnostic and shall not be relied upon in and of itself to ensure integrity.

The read back and display using a diverse means reduces the possibility of a systemic error in the workstation causing the same data corruption on data entry and user checking, which could cause an undetected error. If, for instance, a corruption occurred prior to the generation of the CRC the data would be interpreted as correct by the device and the same data and CRC would be sent back to the workstation. Automated checks of the original and read back data would not indicate an error, because the data matches the CRC. If the data was displayed using the same mechanism as the original display mechanism the error could still be masked and would be undetected. The diverse display paths should reduce this common cause of error.

8.9.15.2.3 Alternative 2 – local diverse display

It is possible to achieve the same level of integrity as defined in 8.9.15.2.2 since it can be shown that the configuration data on the workstation storage is identical to the configuration data in the device. The verification of configuration data is accomplished by writing the configuration data and CRC to the local drive in the workstation. The configuration data and CRC from the local drive are then sent to the device. The device then verifies the correct transfer by comparing the data and CRC. If the transfer to the device is correct, the device sends its calculated CRC back to the workstation to confirm that the data was sent correctly. At that point the workstation compares the device CRC to the stored CRC. At this time it can be assumed that the data in the workstation was correctly sent to the device. It is important that the CRC is only calculated once by the workstation and stored on the local drive. It is not recalculated for the purpose of transfers. This also ensures the integrity of the data on the local drive. At this point it can be assumed that the data in the workstation was correctly sent to the device.

For verification, the user shall view the “sent” configuration data using a diverse means to confirm that the correct configuration was sent to the device. Since the data on the workstation storage includes a CRC, a diverse means of display will allow the user to uncover any errors caused by the workstation when the data was entered into storage. This effectively duplicates the functionality of the full read back method without the overhead of doing a full data read back from the device to the workstation. Figure 74 graphically depicts this process.

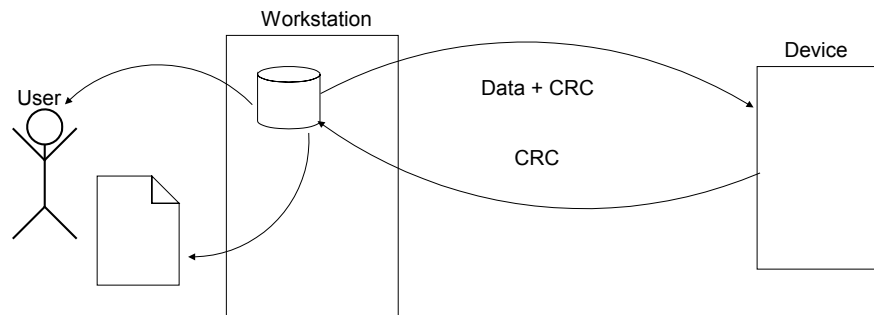


Figure 74 – Diverse display without full data read back

8.9.16 Verification process

Figure 75 defines the complete process for verifying the configuration of safety devices. It is assumed that the data has been downloaded using the process described in 8.9.14, ensuring that the SNCT and the device have the same configuration data. One alternative shall be selected for the user verification of the configuration data. Upon completion of this process the devices and system are ready for SIL 3 operation.

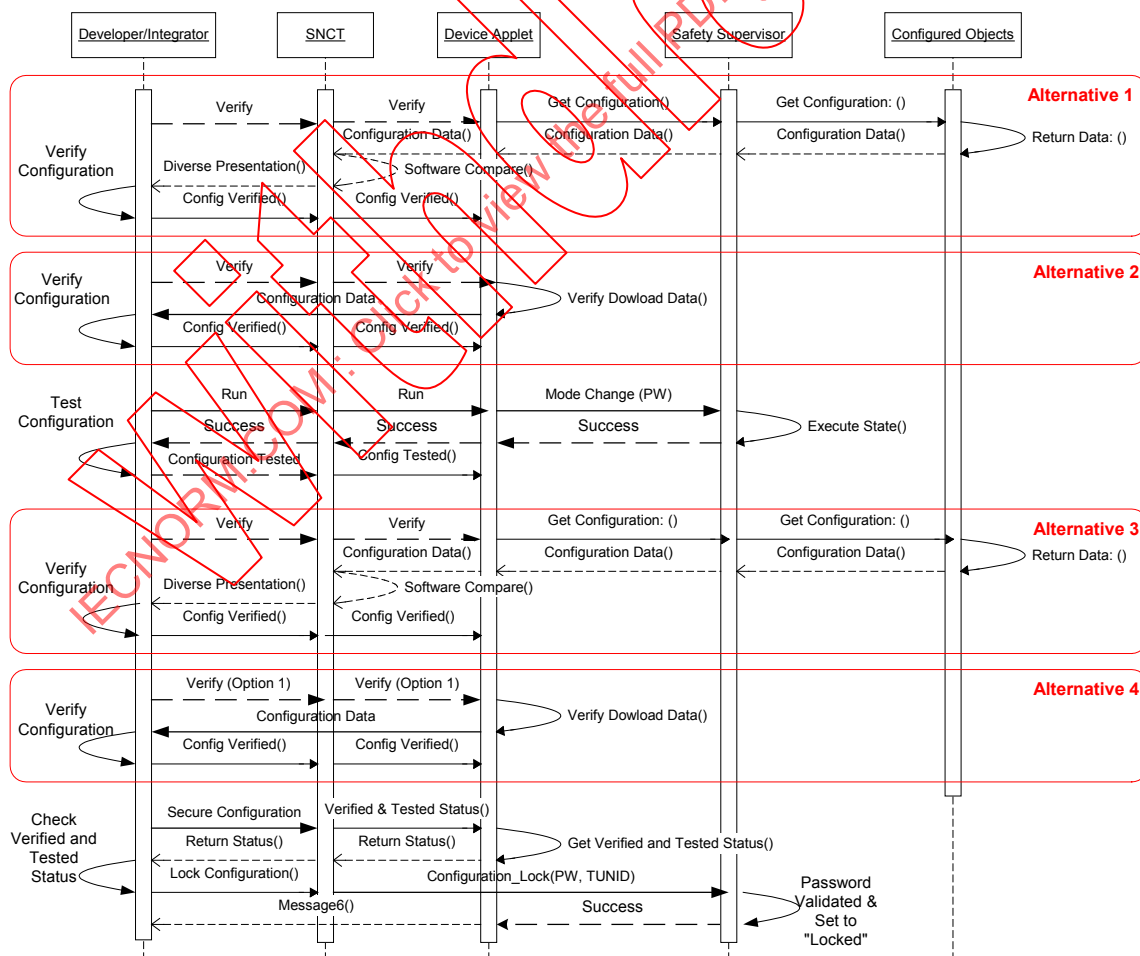


Figure 75 – Verification process including all alternatives

8.9.17 Configuration error analysis

8.9.17.1 Configuration errors and detection measures relationships

Table 96 defines the errors and detection measures that are considered for the download and configuration process defined by the SNCT interface.

Table 96 – Errors and detection measures

Errors	Detection measures							
	User Authentication ^a	Identification of configuration	Identification of target	SCCRC protection on configuration (Part of SCID)	Configuration Locking or Device Ownership	Configuration session control	User Functional Testing	Read back of the device configuration
User misdirected configuration	X				X		X	
System misrouted configuration	X		X		X		X	X
Lost Configuration Message				X			X	X
Corrupted configuration				X			X	X
Configuration at inappropriate time						X		
User loads wrong configuration		X					X	X
Configuration process is interrupted						X		
^a User Authentication is an optional feature and only applies if the feature is used by the end user.								

8.9.17.2 Configuration errors

8.9.17.2.1 User misdirects configuration

The user is required to functionally test devices that are configured so user misdirection can be discovered during testing.

If the user configures a node unintentionally and it is not noticed, the originator connecting to the node will not have the matching signature and will either reconfigure the device with the correct signature or reject the connection. Either of these two actions will result in the system going to a safety state.

If the end user enables authentication features, users cannot configure nodes for which they do not have authorization. User authentication is an optional feature.

8.9.17.2.2 System misroutes configuration

There are two different cases for a system to misroute a configuration: configuration performed during connection establishment by a Form 1 SafetyOpen, and configuration from a workstation or O-to-T configuration process. Both of these cases are explained in the following paragraphs.

- If a routing error occurs during the establishment of a configuration session the configuration mode message will be rejected by the device because of the wrong TUNID or OUNID in the mode message.
- If a configuration message is misrouted within a configuration session, it will be rejected by the receiving device if the device is not in the correct mode or is received on a connection other than the connection used to establish the configuration mode. If the device is in the correct mode the configuration message would have to be addressed to the correct class, instance, attribute to be accepted and would be discovered during the SCID validation steps. The device that was supposed to correctly receive the configuration message will also fail the SCID validation. The user will also discover this during the check of the configuration read back from the device.

8.9.17.2.3 Lost configuration message

The detections for a lost message are the same as those for a misrouted message.

8.9.17.2.4 The configuration is corrupted

If for any reason the configuration is corrupted, it will be detected by the SCID checks, the user functional testing, and the user configuration examination.

8.9.17.2.5 Configuration at an inappropriate time

Configuration of operating devices is prevented by the user authentication and setting of the configuration mode. If a user attempts to configure a device that the user is not authorized for the mode request will be rejected. If the user sends configuration messages to a device without the configuration mode set the device will reject the changes.

8.9.17.2.6 User loads the wrong configuration

Each configuration is uniquely identified by the SCID. Users are required to implement procedures to verify that the configuration being loaded is the correct one for the application.

8.9.17.2.7 Configuration process is interrupted

If the configuration process is interrupted for any reason the device will remain in the configuration mode until a valid configuration is loaded and the mode is changed.

When the configuration process is interrupted the process will need to be restarted on a new connection with a new configuration request.

8.9.17.3 Detection measures

8.9.17.3.1 User authentication

All safety devices have optional password protection. The Configuration Lock, Configuration Request, Mode Change, and Reset commands all require the use of a password to change the state.

8.9.17.3.2 Identification of configuration file

All safety configuration files are uniquely identified by the SCID.

8.9.17.3.3 Identification of target to be configured

The configuration mode message and the verify message both use the TUNID for identification of the target.

8.9.17.3.4 CRC protection on configuration

Part of the SCID is the SCCRC. The SCCRC is 32 bit CRC that is calculated over the safety configuration data.

8.9.17.3.5 Configuration ownership

When an out-of-box target device is configured by an originator, the device captures the OUNID and subsequently prevents any other originator from configuring the device. Thus the ownership assignment (OUNID) provides a means of locking this device to a single configuration source.

8.9.17.3.6 Configuration session control

Prior to configuring a device it shall be placed into a configuration mode. The configuration mode is persistent in the device and remains in place until the device is set to another mode by a command. The configuration mode in safety devices implementing the SNCT interface shall persist through power cycles.

Upon entering the configuration mode, devices implementing the SNCT interface shall:

- Store the configuration mode in NV;
- Mark the existing configuration invalid;
- Lock out all configuration messages for connections other than the one that set the configuration mode;

To enter the configuration mode, devices implementing the SNCT interface shall:

- confirm User Password in the configuration mode command;
- confirm TUNID in the configuration mode command.

8.9.17.3.7 User functional testing

The user is required to functionally test new and replacement devices prior to utilizing the system in a SIL3 environment.

8.9.17.3.8 Configuration protection

After configuration, testing, and user verification has been completed by one of the procedures defined in 8.9.16, all tool-configured devices are locked. These are Targets, Originators, or both. This locked state can be read to provide an indication that the configuration has been verified and also guards against inadvertent configuration changes by requiring the user to first unlock the device.

As a result of user testing, originator-configured target devices have ownership assignments which binds these devices to their owners; effectively locking these devices from being inadvertently configured by some other source. As shown in Figure 72, the combination of locked tool-configured originators and owned targets provides a secure environment to protect the verified system from unauthorized changes.

8.9.17.3.9 Diagnostic software protections (not safety related)

The software may also implement data protection measures of the device configuration data as a data integrity measure and for the early error detection mechanism. It is important to understand that these measures are not safety related and therefore do not need to achieve SIL3 or be certified. These areas are defined here:

Storage of Configuration Data to Hard Disk, CD, etc shall include the SCID and be checked during a file load operation.

The transfer of Configuration Data across software interfaces shall include the SCID and be checked by the receiving software component.

8.9.17.4 Device memory architecture considerations

It is important to understand the memory architecture that target devices could use to know how it could behave during configuration. All these architectures are supported by this part but will change the behavior for the user.

There are 3 distinct data areas defined with respect to configuration:

- NV stored copy – this is a valid configuration stored in Non-volatile memory and is what is used to set up a safety device on power-up;
- Working RAM copy – This is a valid configuration that is an exact copy of the configuration in NV memory, but is what is used during run-time operation;
- Editable RAM Copy – This is a data space that can be modified by software in the configuration mode, but has yet to be applied.

Product developers may choose to implement one or more of these data areas to increase user flexibility.

NV Copy Only – This design approach keeps configuration only in NV memory. When the configuration mode is entered, the entire configuration is invalidated and there is no possibility to restore the configuration. The software shall successfully download a complete and valid configuration to the device. Once it is validated and applied the NV copy is marked as good. The restore command is not supported in this type of device.

NV and Working Copy – This design approach keeps configuration in NV memory and a working copy in RAM. When the device enters the configuration state, the working copy is modified, but until the configuration is validated and applied, the old valid configuration is still available to be restored. A power cycle before the Apply would revert to the last configuration in NV memory. The restore command is supported and would re-copy the NV version into working RAM.

NV and Working Copy with Edit buffers – This design approach keeps a valid working configuration at all times until a “validate and apply” is done. If the configuring device quits without applying the changes, the restore command would allow the device to simply exit the configure state and run with what is in memory. The restore command is supported, but no copy from NV RAM is needed, the working copy can be used.

8.10 Electronic Data Sheets extensions for safety

8.10.1 General rules for EDS based safety devices

8.10.1.1 General

The structure and rules for EDS are defined in ISO 15745–3 Annex C and ISO 15745-4 Annex C. The extensions for safety are defined in the following subclauses.

8.10.1.2 Safety configuration assembly definition

Safety devices can be configured from an EDS file definition. Safety devices that are configured via EDS file shall define a Configuration Assembly so that the software knows how to calculate the SCID. A standard EDS assembly section shall be used to define the various parameters contained in the Safety Configuration Assembly. A field and keyword has been defined in the [Parameter Class] section to identify the instance Id of the assembly. A connection manager entry shall also be used to reference this assembly if it is to be configurable with a Form1 SafetyOpen.

8.10.1.3 Configuration CRC

The SNCT and EDS based safety device shall use the same method for calculating the SCCRC. EDS based safety devices shall use CRC-S4 with the seed value defined in 7.1.2.3. The data input to the CRC algorithm is defined by the Configuration Assembly. Any padding in the configuration assembly will be assumed to be zero for purposes of calculating the SCCRC.

8.10.1.4 Password encryption for EDS devices

The SNCT and any other configuration software implementing the SNCT interface shall have a standardized method for encrypting passwords prior to presenting them to a safety device. Software tools implementing the SNCT interface shall use CRC-S4 with the seed value defined in 7.1.2.3.

When an encrypted password is smaller than the fixed-size password parameter, the method described under the Set_Password service in the Safety Supervisor object definition shall be used.

8.10.2 EDS extensions for safety

8.10.2.1 Extension to [File] section for safety

EDS Files for safety device shall be protected by a CRC. CRC-S4 with the seed value defined in 7.1.2.3 shall be used.

The EDS File CRC value shall be entered in the [file] section with the entry keyword "EDSFileCRC=". The EDSFileCRC entry keyword shall be contained on a single line.

The SNCT shall confirm the EDS file CRC prior to using the file contents as described below:

- a) Open file as a binary file. Load the entire file into memory;
- b) Remove the entire line including the carriage return that has the EDSFileCRC= entry;
- c) Calculate the CRC over the remaining contents of the file;
- d) Compare the calculated CRC to the EDSFileCRC= entry's value.

An "EDSFileCRC=" keyword and value shall be included in all safety device EDS files. If the [Device Classification] entry describes the device as having any Safety capabilities and the EDSFileCRC= entry keyword does not exist, then the SNCT shall not use the file.

8.10.2.2 Extension to [Device Classification] section for safety

The EDS file for safety devices shall include a [Device Classification] section which shall contain the line:

Classx = Safety;

where

x is a decimal value.

8.10.2.3 Extension to [Parameter Class] Section for Safety

The SafetyCfgAssembly Keyword shall be used in the safety devices to identify the configuration assembly object instance. The keyword is defined in Table 97.

Table 97 – Parameter class keywords

Entry name	Entry keyword	Field number	Data Type	Required/Optional
Safety Assembly Instance	SafetyCfgAssembly	1	UINT	Conditional ^a
^a Shall be in implementation for Safety devices that are configured via EDS file				

SafetyCfgAssembly – Identifies the configuration assembly object instance number for safety-related parameters in this device.

A “SafetyCfgAssembly=” keyword entry shall be included for Safety devices that are configured via EDS file. This value shall also match an AssemN entry keyword specified in the [Assembly] section.

8.10.2.4 Extension to [Connection Manager] section for safety

8.10.2.4.1 General

There are four new keywords defined for safety that are described in this subclause. The new keywords are summarized in Table 98.

Table 98 – New Connection Manager section keywords for safety

Entry name	Entry keyword	Field number	Data type	Required/Optional
Maximum Safety Connections	MaxSafetyConnections	1	USINT	Optional
Maximum Safety Input Connections	MaxSafetyInputCnxns	1	USINT	Optional
Maximum Safety Output Connections	MaxSafetyOutputCnxns	1	USINT	Optional
Default Safety Connections				
	Connection Instance 1	1	UINT	Required
	Connection Instance n	n	UINT	Conditional
^a Shall be in implementation for Safety devices that are configured via EDS file				

8.10.2.4.2 Max safety connections

Safety devices will typically support a large number of possible assemblies that mix and match different combinations of I/O data and/or status. At the same time, these devices have a limit on how many safety I/O connections can be established at any one time. These keywords in the [Connection Manager] section define these limits. If no entry exists in the EDS file, no limits exist.

8.10.2.4.3 Default safety connections

There will often be a set of connections that are most commonly used. Product developers can optionally define a default set of connections that allow the user to quickly and easily add

these connections to an originator configuration. The DefaultSafetyConnections entry keyword specifies which connection manager entries are these “default” connections.

8.10.2.4.4 Changes and additions to Connection Manager fields

8.10.2.4.4.1 Overview of changes and additions

Connection manager entry shall be defined for each safety component to which a connection can be established. The safety connections are defined using the existing Connection Manager section defined in ISO 15745-2 with the following extensions. The SNCT shall use each entry to present the user with a set of connection options that can be connected to from an originator.

Table 99 lists the fields currently defined for the Connection Manager section.

Table 99 – Connection Manager field usage for safety

Field name	Field number	Data type	Usage for safety
Trigger and transport	1	DWORD	Class: shall be 0 Trigger: shall be application Transport Application type: all 0 producers shall be Input Only consumers shall be Exclusive Owner EXAMPLE producers (clients): 02040001 _{hex} consumers (server): 84040001 _{hex}
Connection parameters	2	DWORD	O-to-T size: shall only be fixed T-to-O size: shall only be fixed Octets per slot: always ignored Real Time Transfer format: 5 = Safety Priority: always high Connection Type: Inputs: T-to-O Multipoint or Point-to-point O-to-T Point-to-point Outputs: O-to-T and T-to-O Point-to-Point EXAMPLE Multipoint Input (producer): 22645505 _{hex} Point-to-point Output (consumer): 22445505 _{hex}
O-to-T RPI	3	Param	Either leave blank or provide Param reference to define device capability
O-to-T size	4	UINT or Param	IF blank, will either assign the size from the associated ASSEM reference or the software will use the Time Coordination size
O-to-T format	5	Assem	For Safety output connections, this field indicates the Safety output assembly to which the connection is associated. For safety input connections, this field shall be empty
T-to-O RPI	6	Param	Either leave blank or provide Param reference to define device capability
T-to-O size	7	UINT or Param	If blank, will either assign the size from the ASSEM reference or the software will use the Time Coordination size
T-to-O format	8	Assem	For Safety input connections, this field shall indicate the Safety input assembly to which the connection is associated. For safety output connections, this field shall be empty

Field name	Field number	Data type	Usage for safety
config #1 size	9	UINT or Param	Can optionally be used
config #1 format	10	Param or Assem	Can optionally be used
config #2 size	11	UINT or Param	Size of configuration assembly when device supports configuration via Form 1 SafetyOpen
config #2 format	12	Param or Assem	This field can indicate which assembly will hold the Form 1 SafetyOpen configuration data
Connection name string	13	Eds_Char Array	Unique name to allow a user to identify the application assembly to which this instance is associated
Help string	14	Eds_Char Array	User help message related to using the connection
Data Path	15	Eds_Char Array	Complete path comprising of Config, Consumed, and Produced path. Null paths are inserted where no path exists (see possible formats below)
ASYN1	16	USINT	This is a FSCP 2/1 exclusive field Only applies to producing connections. Field should be empty for consuming connections. Used to calculate Network Reaction Time
Max Consumer Number ^a	17	USINT	This is a FSCP 2/1 exclusive field When safety devices wish to define multipoint connections and need to restrict the max number of consumers to a value less than the default maximum of 15, this field can define the product limit. If this field is empty, the SNCT shall always use the default value of 15 for the maximum number of multipoint connections. This field can be left empty for point-to-point connections
^a This field is specified only for safety related specifications and shall either be empty or omitted for non-safety related connection manager instances.			

8.10.2.4.4.2 Trigger and Transport field

Safety devices define whether a connection is a producing (client) or consuming (server) connections by filling out this field. The other parts of the field will have fixed definitions as follows:

- Class: always 0;
- Trigger: always application;
- Transport type: all 0.

8.10.2.4.4.3 Connection Parameter field

Safety Devices define what types of connections are possible for a connection by using this field. All the other fields will have fixed values. Table 100 provides the only valid values for Safety devices.

Table 100 – Connection parameter field settings for safety

Connection Tltype	Connection parameter setting
Producing (client) input connection	22645505 _{hex} or 22445505 _{hex} (no multipoint)
Consuming (server) output connection	22445505 _{hex}

8.10.2.4.4.4 Data Path field

IEC 61158-6-2 defines a method to construct an application path in a Forward Open for 3 references: one for configuration, one for consumed data, and one for produced data. With the introduction of NULL paths for safety, there are some additional requirements for safety.

8.10.2.4.4.5 ASYNC field

This parameter is necessary to properly calculate the Network Reaction Time for a safety connection. This parameter is generally different for input connection and output connections but is design dependent. This keyword is required in the Connection Manager section for producing (client) connections. Devices which produce on input connections shall define a ASYNC value for those connections, and devices which produce on output connections (i.e. logic devices or scanners) shall define a ASYNC value for these connections. Any value provided on consuming (server) connections will be ignored by the software. If an ASYNC parameter is not defined in a producing connection, the software shall use a worst-case default value of 1. The values of the ASYNC parameter are:

- ASYNC = 0 = synchronous;
- ASYNC = 1 = asynchronous.

9 System requirements

9.1 Indicators and switches

9.1.1 General indicator requirements

FSCP 2/1 safety devices are required to have module and network indicators.

Device developers are free to have additional indicators on their products if they so desire.

Important: FSCP 2/1 safety devices shall provide a separate Module Status LED and a Network Status LED. In this case, these indicators provide important additional safety-related information to the user that cannot be accomplished with a combined Net/Mod Status indicator.

Important: If there is a conflict between turning an LED on Red versus Green, the LED should be turned on Red.

9.1.2 LED indications for setting the device UNID

Table 101 defines the way the LED indications are used for setting the UNID.

Table 101 – LED indications for setting UNID

Safety Supervisor State/Event	MOD LED	NET LED
Self-test	Flash Red/Green	Off
Waiting for TUNID	Flash Red/Green	Connection dependent
Waiting for TUNID Proposed UNID Rcvd	Flash Red/Green	Flash Red/Green 250 ms / 250 ms Reuse CP 2/3 Group 4 pattern
Configuring	Flash Red/Green	Connection dependent
Idle	Flashing Green	Connection dependent
Executing	Green	Connection dependent

9.1.3 Module Status LED

This bi-color (green/red) LED provides device status. It indicates whether or not the device has power and is operating properly. Table 102 defines the Module Status LED states. Table 102 defines the Module Status LED conditions for Safety Devices (which has to use the Safety Supervisor object). The states shown reflect the device states specified in the Identity object. These states are also mapped to the states defined in Safety Supervisor object.

For safety devices which use the SNCT interface, the states of the Safety Supervisor shall be what determine the Module LED status.

Table 102 – Module Status LED

For Safety Supervisor state:	LED is:	To indicate:
No Power	Off	There is no power applied to the device
Executing	Green	The device is operating in a normal condition
Idle state	Flashing Green ^a	The Device is in the Idle or Standby State. Reference the Identity object(6.4) or Safety Supervisor object(6.6) ^b
Abort	Flashing Red ^a	Recoverable Fault
Critical Fault	Red	The device has an unrecoverable fault, may need replacing
Device Self Testing, Waiting_for_TUNID or Configuring ²	Flashing Red-Green ¹	The Device is in Self Test or the Device needs commissioning due to configuration or UNID missing, incomplete or incorrect ^c . Reference the Identity object(6.4) or Safety Supervisor object(6.6) ^b
^a The flash rate for is approximately 1 flash per second. The LED should be on for 0,5 s on off for approximately 0,5 s. ^b The state of Safety Devices is determined by the Safety Supervisor object. ^c An Unconfigured/Unowned Safety Device should not be allowed to persist while connected to an active safety system.		

9.1.4 Indicator warning

The User Safety Manual shall provide a warning that states “LEDs are NOT reliable indicators and cannot be guaranteed to provide accurate information. They should ONLY be used for general diagnostics during commissioning or troubleshooting. Do not attempt to use LEDs as operational indicators”

9.1.5 Network Status LED

This bi-color (green/red) LED indicates the status of the communication link. The indications for this LED are defined in Table 103.

Table 103 – Network status LED states

For Safety Supervisor state:	LED is:	To indicate:
Not Powered/Not On-line	Off	Device is not on-line. The device has not completed the Dup_MAC_ID test yet. The device may not be powered, look at Module Status LED.
On-line, Not Connected	Flashing Green ^a	Device is on-line but has no connections in the established state The device has passed the Dup_MAC_ID test, is on-line, but has no established connections to other nodes. For a Group 2 Only device it means that this device is not allocated to a master. For a UCMM capable device it means that the device has no established connections. For Safety Devices, a connection may be established, but the Validator has not completed an initial Time Coordination exchange.
Link OK, On-line, Connected	Green	The device is on-line and has connections in the established state.
Connection Time-out	Flashing Red ^a	One or more I/O Connections are in the Timed-Out state.
Critical Link Failure	Red	Failed communication device. The device has detected an error that has rendered it incapable of communicating on the network
Communication Faulted and Received an Identify Comm Fault Request – Long Protocol or Propose_TUNID service received	Flashing Red-Green	A specific Communication Faulted device. The device has detected a Network Access error and is in the Communication Faulted state. The device has subsequently received and accepted an Identify Communication Faulted Request - Long Protocol message. Safety devices use this indication to allow maintenance personnel to commission the UNID in a safety device.
^a The flash rate for is approximately 1 flash per second. The LED should be on for 0,5 s on off for approximately 0,5 s.		

The following algorithm can also be applied to the handling of the Baud Rate switch if implemented. When hardware switches are used for setting MACID and/or Baud Rate there are some differences in how these values are processed in safety devices. This subclause will outline the algorithm.

There is direct relationship between the MACID setting and the MACID portion of the Target UNID (TUNID) in the safety supervisor. See the Safety Supervisor (6.6) object for more information.

9.1.6 MACID determination

During device initialization, the MACID switches are read by the device firmware.

The MACID attribute (if supported) in the DeviceNet object shall always reflect the value in use.

The following logic is applied:

- When the switches are read, if the switches specify a valid MACID, (a value from 0 to 63 for MACID), this value shall be used as the MACID;
- When the switches are read, if the specified MACID differs from the value stored in the DeviceNet object's MACID Attribute (NVRAM values), the appropriate attribute/s shall be updated and saved to NVRAM;
- When the switches are read, if the switches specify an invalid MACID, (a value greater than 63), the device should follow one of the recommended options in IEC 62026-3;
- When the switches are read, if the switches used to specify the MACID are set to a valid value (a value from 0 to 63 for the MACID), the MACID attribute shall have Get Only access;
- When the switches are read, if the switches are set to the software programmable mode (>63 for the MACID), the MACID attribute shall have Set access;
- If a device has no switches it shall use default values for MACID in its out-of-box configuration (63 for the MACID);
- If the device is out-of-box it shall use the current MACID switch settings if these are valid values.

Safety devices shall monitor and detect changes in the MACID and Baud Rate switch settings. Developers are free to determine an appropriate sample period for this detection.

When a MACID switch setting change is detected, the device shall update the MACID_Switch_Value attribute. If the MACID switch equals the MACID in use or the MACID Switches are greater than 63 and software settable clear the MACID changed attribute and make no change in the state of the device, otherwise set the MACID changed attribute and transition to the ABORT state (see flow chart in (Figure 76).

When a Baud Rate switch setting change is detected, the device shall update the Baud_Rate_Switch_Changed attribute and the Baud_Rate_Switch_Value attribute, then transition to ABORT.

If a FSCP 2/1 safety device supports the Set_Attribute service to the MACID attribute, and the MACID switches are set to enable the set service, Safety devices shall only accept the Set_Attribute while in the "Waiting for TUNID" state. Otherwise, it shall reply with a "Device State Conflict".

If a FSCP 2/1 safety device supports the Set_attribute service to the Baud Rate attribute, Safety devices shall only accept the Set_Attribute while in the "Waiting for TUNID" state. Otherwise, it shall reply with a "Device State Conflict".

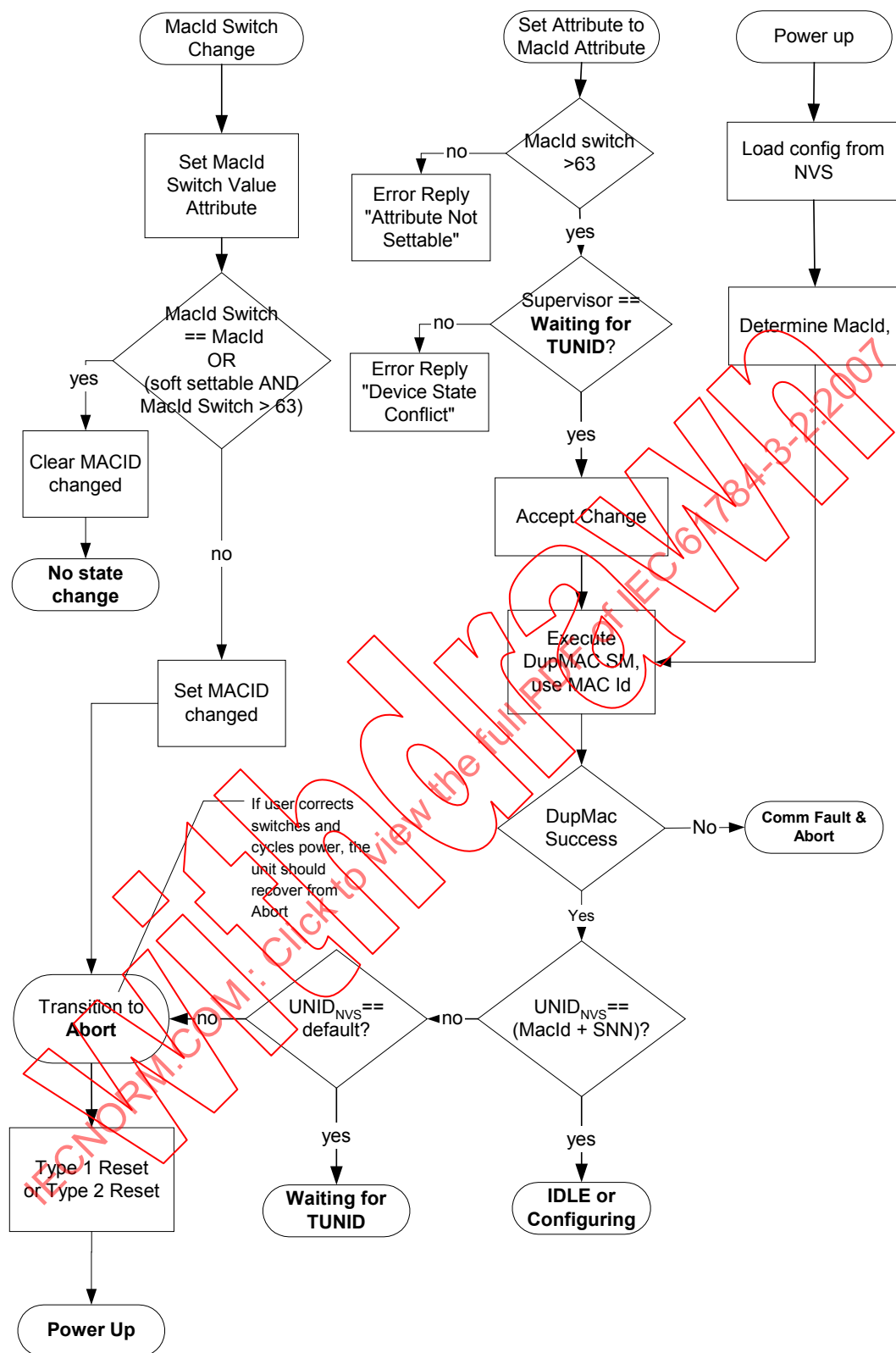


Figure 76 – Safety device MACID processing logic

9.1.7 Reset switch

If a reset switch is provided on a safety device, the safety device shall execute a Safety Supervisor Type 2 reset (i.e. reset to out-of-box, but preserve the password).

9.2 Installation guidelines

The installation guidelines of IEC 61918 and IEC 61784-5-2 for CPF 2 shall apply.

9.3 Safety function response time

9.3.1 Overview

The safety function response time is the worst-case time from a safety related event as input to the system or as a fault within the system, until the time that the system is in the safety state. The scope of the reaction time is defined in Figure 77.

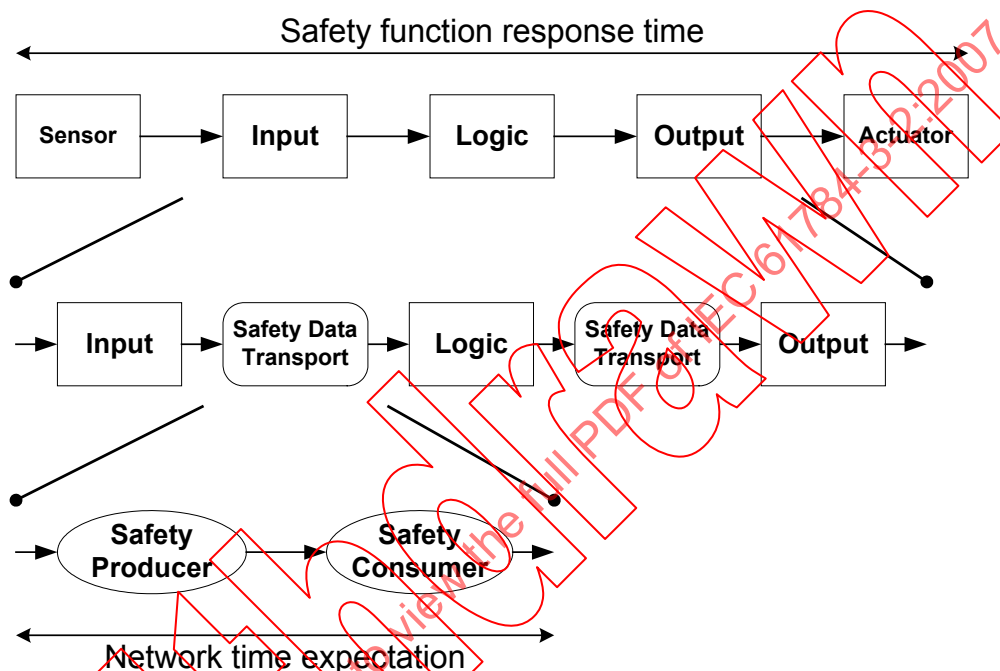


Figure 77 – Safety function response time

To determine the safety function time of any control chain the user shall add up the components of the safety chain.

Using the example in Figure 77, the safety function response time would be:

$$\begin{aligned} \text{Safety function response time} = & \text{Sensor reaction time} \\ & + \text{Input reaction time} \\ & + \text{Network reaction time} \\ & + \text{Controller reaction time} \\ & + \text{Network reaction time} \\ & + \text{Output reaction time} \\ & + \text{Actuator reaction time} \end{aligned} \quad (13)$$

9.3.2 Network time expectation

The Network Time Expectation is a portion of the Safety function response time. The Network Time Expectation is the worst case time, from the time the data is captured by the safety data producer, until the consuming application recognizes a safety state. This also includes errors during production and consumption.

The Network_Time_Expectation_Multiplier used in the previous subclauses of this part, represents the worst case measured time from the time the data is captured by the safety data producer until the time that the consuming application recognizes a safety state indication.

The `Network_Time_Expectation_Multiplier` value necessary to ensure a timeout error is not detected is:

$$\text{Network\_Time\_Expectation\_Multiplier} > \frac{\text{EPI} \times \text{Timeout\_Multiplier} + \text{Safety\_Message\_Time}(\text{max}) + \text{Time\_Coord\_Message\_Time}(\text{max})}{128 \mu\text{s}} + \text{Connection\_Correction\_Constant} \quad (14)$$

where

`Safety_Message_Time(max)`, is the actual time from the data being captured by the safety data producer until the time that the safety data is passed to the consuming application for use.

`Time_Coord_Message_Time(max)`, is the maximum time it could take for the time coordination information to be sent from the consumer to the producer. This is measured from the time the `Consumer_Clk_Count` is captured to `Consumer_Time_Value` until the time `Producer_Clk_Count` is captured to `Producer_Rcvd_Time_Value`.

`Timeout_Multiplier`, is a parameter that is needed by the safety protocol processing. This value determines the number of messages that may be lost before declaring a connection error. A `Timeout_Multiplier` of 1 indicates that no messages may be lost. A `Timeout_Multiplier` of 2 indicates that 1 message may be lost, etc. In addition to the `Timeout_Multiplier` checks within the protocol, reaction time checks are also performed. A reasonable default for the `Timeout_Multiplier` would be 2. A reasonable limit would be for configuration software to restrict the `Timeout_Multiplier` to integer values from 1 to 4.

`Connection_Correction_Constant`, is a value in 128 μs increments that is subtracted from the time stamp to represent the worst case error due to Time drift, the asynchronous nature of the producer and consumer clocks, and the minimum time for the Time Coordination Message to traverse from the consumer to the producer.

`Timeout_Multiplier`, and `Connection_Correction_Constant`, are described in 7.6.5.8 and 7.6.5.15 respectively.

9.3.3 Equations for calculating network reaction times

The most logical place for the user to set and determine the Network reaction times is at the logic device since it is very likely both the configuration holder and connection originator.

There are two cases of safety connection reaction times for most logic devices. They are:

- Input Connection reaction time;
- Output Connection reaction time.

Table 104 defines the producing application and consuming application for the Network reaction time Cases above.

Table 104 – Connection reaction time type – producing/consuming applications

Connection reaction time type	Producing application	Consuming application
Safety Input	Input module	Controller
Safety Output	Controller	Output module

Figure 78 shows the analysis of the Safety Function Response Time components. The time enforcement analysis is done from the Output signal back to the Input signal. Doing the analysis from the Output signal back to the Input signal allows the effects of asynchronous productions to be seen.

The analysis assumes that the output module will have an internal cycle that checks the age of the last received output data. The age of the output data will be maintained at less than the Network_Time_Expectation_Multiplier (NTEM) value.

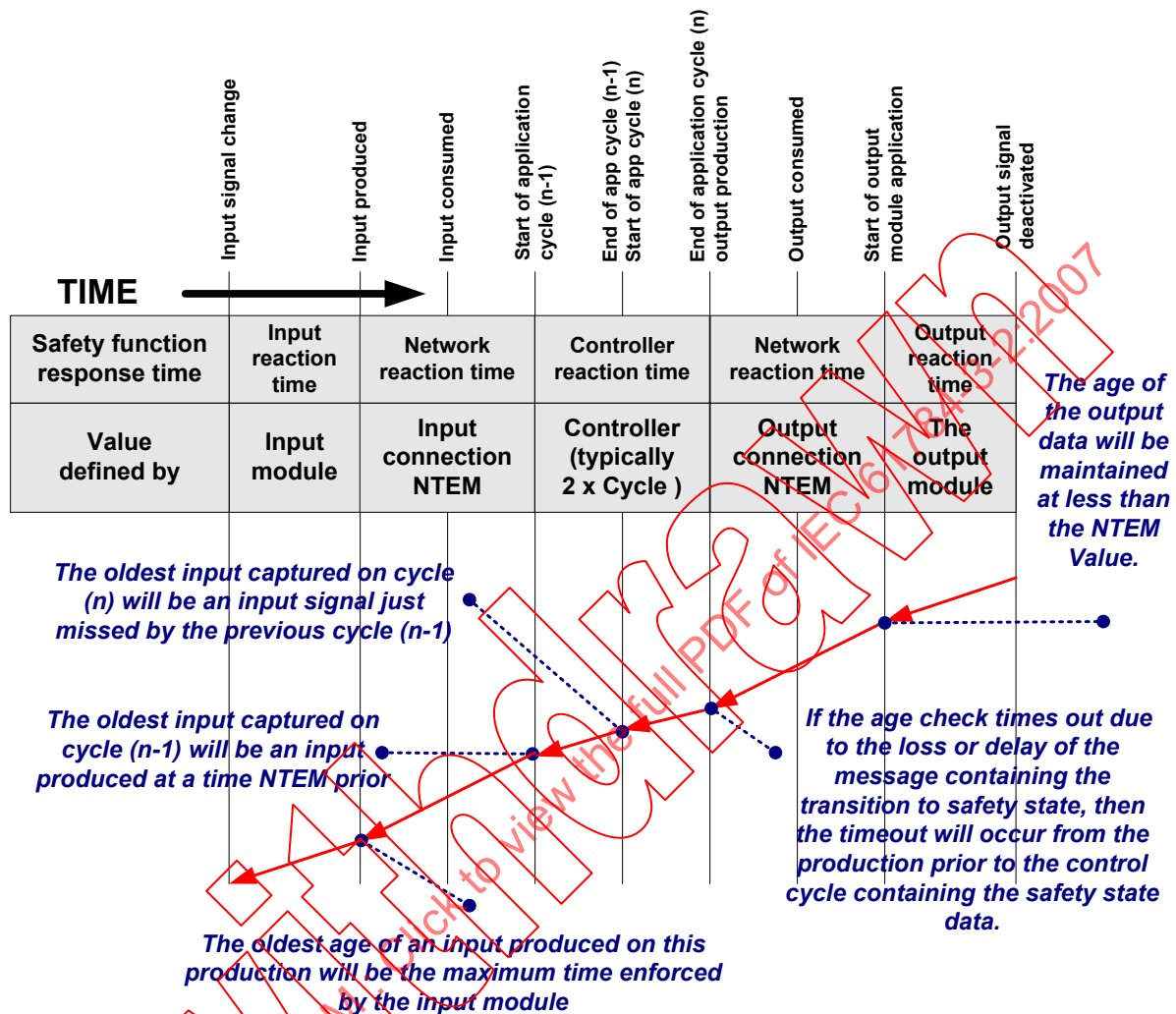


Figure 78 – Safety function response time components

From an analysis point of view, a controller to controller connection would be treated the same as an input connection.

For an Input or Controller to Controller Connection:

$$\text{Network reaction time} = \text{Network\_Time\_Expectation\_Multiplier} \times 128 \mu\text{S} \quad (15)$$

For a Connection to an Output Module:

If the Output Production is Asynchronous to the Controller Cycle:

$$\text{Network reaction time} = \text{Network\_Time\_Expectation\_Multiplier} \times 128 \mu\text{S} \quad (16)$$

If the Output Production is Synchronous to the Controller Cycle:

$$\text{Network reaction time} = \text{Network\_Time\_Expectation\_Multiplier} \times 128 \mu\text{S} - \text{EPI\_in\_Sec} \quad (17)$$

The Output Connection cases can be simplified to:

$$\text{Network reaction time} = \text{Network Time Expectation Multiplier} \times 128 \mu\text{S} + (\text{EPI\_in\_Sec} \times (\text{ASYNC} - 1)) \quad (18)$$

where the ASYNC parameter is defined as follows:

ASYNC parameter

If the producing application is synchronous with data production, ASYNC = 0. If the producing application is asynchronous to the data production, ASYNC = 1. The net effect of this difference is that one additional EPI is added to the Network reaction time for asynchronous devices. This ASYNC parameter should be provided to the Safety Configuration software via and EDS file entry. The default value in cases where it is not provided shall be 0.

9.4 Duration of demands

The duration of demand by the safety-related application to the safety communication layer shall be sufficient in duration such that demand is detected by the longest EPI selected by the application.

9.5 Constraints for calculation of system characteristics

9.5.1 Number of nodes

CP 2/3 based implementations are limited to a maximum of 32 producing and 32 consuming nodes for this safety profile. CP 2/2 based implementations have no limitation on the number of producers and consumers for this safety profile.

9.5.2 Network PFH

This subclause will describe the calculations used for the determination of the network PFH. The network is targeted at SIL 3 systems therefore the PFH shall be $<10^{-9}$ (or $<10^{-7}$ if a 1 % factor is included in the calculations).

The safety network protocol consists of four parts, safety data, time stamp information, the multipoint time correction information, and the time coordination ping response. Each of these parts has its own protection that contributes to the PFH.

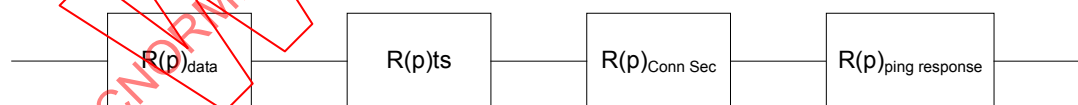


Figure 79 – Network protocol reliability block diagram (RBD)

From the RBD in Figure 79, the equation for the overall network residual error rate is

$$R(p)T = R(p) \text{ data} + R(p)ts + R(p)\text{time coord} + R(p) \text{ time\_corr} \quad (19)$$

When applying the residual error rates in the calculation of the PFH, it is important to consider that the time coordination and the multipoint time correction information are transmitted at rates other than the rate of the time stamp and data. Using the different transmission rates and the information from the RBD, for the overall PFH equation for point-to-point connection is:

$$\Lambda = 3\,600 \times ((R(p)ts + R(p)data) \times vmsg) + (R(p)t\_coord \times vping) \times (NoProd - 1) \times 100 \quad (20)$$

where

vmsg	Transmission rate in messages/second
vping	Transmission rate in messages/second
NoP	Number of Producers
3 600	conversion of hours to seconds
100	1 % factor

The PFH equation for multipoint PFH is:

$$\Lambda = 3600 \times (((R(p)t_s + R(p)\text{data}) \times v\text{msg} \times \text{NoP}) + ((R(p)t_{\text{coord}} + R(p)t_{\text{corr}}) \times v\text{ping} \times \text{NoC/P})) \times 100 \quad (21)$$

where

vmsg	Transmission rate in messages/second
vping	Transmission rate in messages/second
NoC/P	Number of Consumers per Producer
NoP	Number of Producers
3 600	conversion of hours to seconds
100	1 % factor

Figure 80 summarizes the worst-case result. The numbers are obtained from and the formulas above.

Inputs			
Expected Packet Interval (EPI)		0,010	Seconds
Ping Interval		0,524	Seconds
Number of Nodes (m)		63	
Multipoint Consumers		16	
Bit Error Probability		0,001	
Results			
		Point-to-point	Multipoint
R(p)data <= 2 octets	R(p) data section - 2 octets of data	$7,02 \times 10^{-20}$	$7,02 \times 10^{-20}$
R(p)data >= 3 octets	R(p) data section - 250 octets of data	$2,85 \times 10^{-24}$	$2,85 \times 10^{-24}$
R(p)data	R(p) data section	$7,02 \times 10^{-20}$	$7,02 \times 10^{-20}$
R(p)ts	R(p)time stamp section	$3,18 \times 10^{-18}$	$3,18 \times 10^{-18}$
R(p)multi	R(p) Time correction message		$1,40 \times 10^{-18}$
R(p)reply	R(p) Time coordination message	$1,40 \times 10^{-18}$	$1,40 \times 10^{-18}$
Msg_per_sec	Messages per second = $1 / \text{EPI}$	100	100
Rpl_per_sec	Replies per second = $1 / \text{Ping\_Interval}$	1,91	1,91
		$7,32 \times 10^{-09}$	$9,17 \times 10^{-09}$
% of 1,00E-07 goal, up to 100 % is OK		7,32 %	9,17 %
PFH = $\lambda / 100$		$7,32 \times 10^{-11}$	$9,17 \times 10^{-11}$
% of SIL 3 (1,00E-07) PFD limit, up to 1 % is OK		0,073 2 %	0,091 7 %

Figure 80 – Network PFH summary

9.5.3 Bit Error Rate (BER)

IEC 61784-3 requires a bit error rate (BER) of 0,01 be used in PDF/PFH calculations unless evidence is given that this BER is not practical. A BER of 0,01 means that 1 of every 100 bits transmitted is transmitted incorrectly. This of course, is an unrealistic condition for an industrial control system network where availability needs to be maintained.

9.5.3.1 CP 2/3 BER

The minimum packet size used in FSCP 2/1 on CP 2/3 is 108 bits. DeviceNet has extensive error confinement and all nodes on a CP 2/3 system check all messages at every transmission. In the presence of persistent errors a CP 2/3 device will bus-off (i.e. the node ceases all network activity) and remain in the bus-off state until power cycled. Because of the extensive and diverse checking and the mandatory error handling a CP 2/3 system will stop operating in the presence of high bit errors. It can be shown that a CP 2/3 network will not operate with a BER above 0,001. The inability to operate is true for both CP 2/3 and FSCP 2/1.

9.5.3.2 CP 2/2 BER

The minimum packet size for a FSCP 2/1 on CP 2/2 is 720 bits, which is much larger than the 108 bits on CP 2/3. Like CP 2/3, CP 2/2 will not operate with a BER above 0,001, though the checking mechanism is different in the case of CP 2/2. CP 2/2 uses the time expectation mechanism inherent in the FSCP 2/1 protocol to ensure messages are delivered on time. If a correct message is not delivered, within the expectation period, the connection will be closed and the device will go to the safety state. If high BER (above 0,001) are present, the safety connection will be closed and the network will not be able to operate. It simply will not be available.

Rational for setting maximum BER at 0,001 for CP 2/2:

- a) If the percent of packets not delivered correctly was high enough, some messages would not be delivered before the connection would time-out
- b) If the connections timed out frequently, the network would not be available and could not be used.
- c) If a network is not available, it is in a safe condition.

9.5.3.3 Summary

A BER < 0,001 will cause the safety protocol to exceed the maximum 7 RPI time limit frequently enough for the network not to be available for practical use, therefore BER = 0,001 was used for all calculations.

9.6 Maintenance

There are no special maintenance requirements for this protocol.

9.7 Safety manual

Implementers of this part shall supply a safety manual with following information at a minimum:

- a) The safety manual shall inform the users of constraints for calculation of system characteristics, see 9.5.
- b) The safety manuals shall inform the users of their responsibilities in the proper parameterization of the device, see 8.9.
- c) The safety manuals shall provide warning on the proper usage of LEDs, see 9.1.4.

In addition to the requirements of this clause the safety manual shall follow all requirements in IEC 61508.

10 Certification

It is highly recommended that implementers of FSCP 2/1 obtain verification from an independent competent body for all functional safety aspects of the product, both the protocol and any application. It is highly recommended that implementers of FSCP 2/1 obtain proof that a suitable conformance test has been performed by an independent competent body.

Annex A (informative)

Additional information for functional safety communication profiles of CPF 2

A.1 Hash function example code

```
#include "stdio.h"          //for printf
#include "memory.h"         //for memcmp

#define CRCS1_POLYNOMIAL    0x37
#define CRCS2_POLYNOMIAL    0x3b
#define CRCS3_POLYNOMIAL    0x080F
#define CRC32_POLYNOMIAL    0xedb88320 // not 0x04C11DB7

// CRC32 (CRC-S4)
// polynomial 0xedb88320 = standard CRC-32 bit reversed from the
// regular
// published polynomial of 0x04c11db7. This table should be used to
// replicate standard Ethernet CRC hardware algorithm. Thus the
// software
// implementation of the Ethernet CRC should produce the same result
// as the
// hardware. This table was generated via right shifting the LSB of
// each
// byte
const unsigned long CRC32Table[256] =
{
0x00000000, 0x77073096, 0xEE0E612C, 0x990951BA,
0x076DC419, 0x706AF48F, 0xE963A535, 0x9E6495A3,
0x0EDB8832, 0x79DCB8A4, 0xE0D5E91E, 0x97D2D988,
0x09B64C2B, 0x7EB17CBD, 0xE7E82D07, 0x90BF1D91,
0x1DB71064, 0x6AB020F2, 0xF3B97148, 0x84BE41DE,
0x1ADAD47D, 0x6DDDE4EB, 0xF4D4B551, 0x83D385C7,
0x136C9856, 0x646BA8C0, 0xFD62F97A, 0x8A65C9EC,
0x14015C4F, 0x63066CD9, 0xFA0F3D63, 0x8D080DF5,
0x3B6E20C8, 0x4C69105E, 0xD56041E4, 0xA2677172,
0x3C03E4D1, 0x4F06D447, 0xD60785FD, 0xA30AB56B,
0x35B5A8FA, 0x42B2986C, 0xDBBBC9D6, 0xACBCF940,
0x32D86CE3, 0x45DF5C75, 0xDCD60DCF, 0xABD13D59,
0x26D930AC, 0x51DE003A, 0xC8D75180, 0xBFDD06116,
0x21B4F4B5, 0x56B3C423, 0xCFBA9599, 0xB8BDA50F,
0x2802B89E, 0x5F058808, 0xC60CD9B2, 0xB10BE924,
0x2F6F7C87, 0x58684C11, 0xC1611DAB, 0xB6662D3D,
0x76DC4190, 0x01DB7106, 0x98D220BC, 0xEFD5102A,
0x71B18589, 0x06B6B51F, 0x9FBBE4A5, 0xE8B8D433,
0x7807C9A2, 0x0F00F934, 0x9609A88E, 0xE10E9818,
0x7F6A0DBB, 0x086D3D2D, 0x91646C97, 0xE6635C01,
0x6B6B51F4, 0x1C6C6162, 0x856530D8, 0xF262004E,
0x6C0695ED, 0x1B01A57B, 0x8208F4C1, 0xF50FC457,
0x65B0D9C6, 0x12B7E950, 0x8BBEB8EA, 0xFCB9887C,
0x62DD1DDF, 0x15DA2D49, 0x8CD37CF3, 0xFBD44C65,
0x4DB26158, 0x3AB551CE, 0xA3BC0074, 0xD4BB30E2,
0x4ADFA541, 0x3DD895D7, 0xA4D1C46D, 0xD3D6F4FB,
0x4369E96A, 0x346ED9FC, 0xAD678846, 0xDA60B8D0,
0x44042D73, 0x33031DE5, 0xAA0A4C5F, 0xDD0D7CC9,
0x5005713C, 0x270241AA, 0xBE0B1010, 0xC90C2086,
0x5768B525, 0x206F85B3, 0xB966D409, 0xCE61E49F,
0x5EDEF90E, 0x29D9C998, 0xB0D09822, 0xC7D7A8B4,
0x59B33D17, 0x2EB40D81, 0xB7BD5C3B, 0xC0BA6CAD,
```

```

0xEDB88320, 0x9ABFB3B6, 0x03B6E20C, 0x74B1D29A,
0xEAD54739, 0x9DD277AF, 0x04DB2615, 0x73DC1683,
0xE3630B12, 0x94643B84, 0x0D6D6A3E, 0x7A6A5AA8,
0xE40ECF0B, 0x9309FF9D, 0x0A00AE27, 0x7D079EB1,
0xF00F9344, 0x8708A3D2, 0x1E01F268, 0x6906C2FE,
0xF762575D, 0x806567CB, 0x196C3671, 0x6E6B06E7,
0xFED41B76, 0x89D32BE0, 0x10DA7A5A, 0x67DD4ACC,
0xF9B9DF6F, 0x8EBEEFF9, 0x17B7BE43, 0x60B08ED5,
0xD6D6A3E8, 0xA1D1937E, 0x38D8C2C4, 0x4FDDFF252,
0xD1BB67F1, 0xA6BC5767, 0x3FB506DD, 0x48B2364B,
0xD80D2BDA, 0xAF0A1B4C, 0x36034AF6, 0x41047A60,
0xDF60EFC3, 0xA867DF55, 0x316E8EEF, 0x4669BE79,
0xCB61B38C, 0xBC66831A, 0x256FD2A0, 0x5268E236,
0xCC0C7795, 0xBB0B4703, 0x220216B9, 0x5505262F,
0xC5BA3BBE, 0xB2BD0B28, 0x2BB45A92, 0x5CB36A04,
0xC2D7FFA7, 0xB5D0CF31, 0x2CD99E8B, 0x5BDEAE1D,
0x9B64C2B0, 0xEC63F226, 0x756AA39C, 0x026D930A,
0x9C0906A9, 0xEB0E363F, 0x72076785, 0x05005713,
0x95BF4A82, 0xE2B87A14, 0x7BB12BAE, 0x0CB61B38,
0x92D28E9B, 0xE5D5BE0D, 0x7CDCEFB7, 0x0BDBDF21,
0x86D3D2D4, 0xF1D4E242, 0x68DDB3F8, 0x1FDA836E,
0x81BE16CD, 0xF6B9265B, 0x6FB077E1, 0x18B74777,
0x88085AE6, 0xFF0F6A70, 0x66063BCA, 0x11010B5C,
0x8F659EFF, 0xF862AE69, 0x616BFFD3, 0x166CCF45,
0xA00AE278, 0xD70DD2EE, 0x4E048354, 0x3903B3C2,
0xA7672661, 0xD06016F7, 0x4969474D, 0x3E6E77DE,
0xAED16A4A, 0xD9D65ADC, 0x40DF0B66, 0x37D83BF0,
0xA9BCAE53, 0xDEBB9EC5, 0x47B2CF7F, 0x30B5FFE9,
0xBDBDF21C, 0xCABAC28A, 0x53B39330, 0x24B4A3A6,
0xBAD03605, 0xCDD70693, 0x54DE5729, 0x23D967BF,
0xB3667A2E, 0xC4614AB8, 0x5D681B02, 0x2A6F2B94,
0xB40BBE37, 0xC30C8EA1, 0x5A05DE1B, 0x2D02EF8D
};

```

```

//CRCS3_POLYNOMIAL = 0x080F
//This table was generated via left shifting the MSB of each byte
const unsigned short CRC16Table[256] =
{
0x0000, 0x080F, 0x101E, 0x1811, 0x203C, 0x2833, 0x3022, 0x382D,
0x4078, 0x4877, 0x5066, 0x5869, 0x6044, 0x684B, 0x705A, 0x7855,
0x80F0, 0x88FF, 0x90EE, 0x98E1, 0xA0CC, 0xA8C3, 0xB0D2, 0xB8DD,
0xC088, 0xC887, 0xD096, 0xD899, 0xE0B4, 0xE8BB, 0xF0AA, 0xF8A5,
0x09EF, 0x01E0, 0x11F1, 0x11FE, 0x29D3, 0x21DC, 0x39CD, 0x31C2,
0x4997, 0x4198, 0x5189, 0x5186, 0x69AB, 0x61A4, 0x79B5, 0x71BA,
0x891F, 0x8110, 0x9101, 0x910E, 0xA923, 0xA12C, 0xB93D, 0xB132,
0xC967, 0xC168, 0xD979, 0xD176, 0xE95B, 0xE154, 0xF945, 0xF14A,
0x13DE, 0x1BD1, 0x03C0, 0x0BCF, 0x33E2, 0x3BED, 0x23FC, 0x2BF3,
0x53A6, 0x5BA9, 0x43B8, 0x4BB7, 0x739A, 0x7B95, 0x6384, 0x6B8B,
0x932E, 0x9B21, 0x8330, 0x8B3F, 0xB312, 0xBB1D, 0xA30C, 0xAB03,
0xD356, 0xDB59, 0xC348, 0xCB47, 0xF36A, 0xFB65, 0xE374, 0xEB7B,
0x1A31, 0x123E, 0x0A2F, 0x0220, 0x3A0D, 0x3202, 0x2A13, 0x221C,
0x5A49, 0x5246, 0x4A57, 0x4258, 0x7A75, 0x727A, 0x6A6B, 0x6264,
0x9AC1, 0x92CE, 0x8ADF, 0x82D0, 0xBAFD, 0xB2F2, 0xAAE3, 0xA2EC,
0xDAB9, 0xD2B6, 0xCA7, 0xC2A8, 0xFA85, 0xF28A, 0xEA9B, 0xE294,
0x27BC, 0x2FB3, 0x37A2, 0x3FAD, 0x0780, 0x0F8F, 0x179E, 0x1F91,
0x67C4, 0x6FCB, 0x77DA, 0x7FD5, 0x47F8, 0x4FF7, 0x57E6, 0x5FE9,
0xA74C, 0xAF43, 0xB752, 0xBF5D, 0x8770, 0x8F7F, 0x976E, 0x9F61,
0xE734, 0xEF3B, 0xF72A, 0xFF25, 0xC708, 0xCF07, 0xD716, 0xDF19,
0x2E53, 0x265C, 0x3E4D, 0x3642, 0x0E6F, 0x0660, 0x1E71, 0x167E,
0x6E2B, 0x6624, 0x7E35, 0x763A, 0x4E17, 0x4618, 0x5E09, 0x5606,
0xAEA3, 0xA6AC, 0xBEBD, 0xB6B2, 0x8E9F, 0x8690, 0x9E81, 0x968E,
0xEEDB, 0xE6D4, 0xFECD, 0xF6CA, 0xCEE7, 0xC6E8, 0xDE9, 0xD6F6,
0x3462, 0x3C6D, 0x247C, 0x2C73, 0x145E, 0x1C51, 0x0440, 0x0C4F,

```

```
0x741A, 0x7C15, 0x6404, 0x6C0B, 0x5426, 0x5C29, 0x4438, 0x4C37,
0xB492, 0xBC9D, 0xA48C, 0xAC83, 0x94AE, 0x9CA1, 0x84B0, 0x8CBF,
0xF4EA, 0xFCE5, 0xE4F4, 0xECFB, 0xD4D6, 0xDCD9, 0xC4C8, 0xCCC7,
0x3D8D, 0x3582, 0x2D93, 0x259C, 0x1DB1, 0x15BE, 0x0DAF, 0x05A0,
0x7DF5, 0x75FA, 0x6DEB, 0x65E4, 0x5DC9, 0x55C6, 0x4DD7, 0x45D8,
0xBD7D, 0xB572, 0xAD63, 0xA56C, 0x9D41, 0x954E, 0x8D5F, 0x8550,
0xFD05, 0xF50A, 0xED1B, 0xE514, 0xDD39, 0xD536, 0xCD27, 0xC528
};
```

```
//CRCS1_POLYNOMIAL=0x37
```

```
//This table was generated via left shifting the MSB of each byte
```

```
const unsigned char CRC8Table37[256] =
```

```
{
0x00, 0x37, 0x6e, 0x59, 0xdc, 0xeb, 0xb2, 0x85,
0x8f, 0xb8, 0xe1, 0xd6, 0x53, 0x64, 0x3d, 0x0a,
0x29, 0x1e, 0x47, 0x70, 0xf5, 0xc2, 0x9b, 0xac,
0xa6, 0x91, 0xc8, 0xff, 0x7a, 0x4d, 0x14, 0x23,
0x52, 0x65, 0x3c, 0x0b, 0x8e, 0xb9, 0xe0, 0xd7,
0xdd, 0xea, 0xb3, 0x84, 0x01, 0x36, 0x6f, 0x58,
0x7b, 0x4c, 0x15, 0x22, 0xa7, 0x90, 0xc9, 0xfe,
0xf4, 0xc3, 0x9a, 0xad, 0x28, 0x1f, 0x46, 0x71,
0xa4, 0x93, 0xca, 0xfd, 0x78, 0x4f, 0x16, 0x21,
0x2b, 0x1c, 0x45, 0x72, 0xf7, 0xc0, 0x99, 0xae,
0x8d, 0xba, 0xe3, 0xd4, 0x51, 0x66, 0x3f, 0x08,
0x02, 0x35, 0x6c, 0x5b, 0xde, 0xe9, 0xb0, 0x87,
0xf6, 0xc1, 0x98, 0xaf, 0x2a, 0x1d, 0x44, 0x73,
0x79, 0x4e, 0x17, 0x20, 0xa5, 0x92, 0xcb, 0xfc,
0xdf, 0xe8, 0xb1, 0x86, 0x03, 0x34, 0x6d, 0x5a,
0x50, 0x67, 0x3e, 0x09, 0x8c, 0xbb, 0xe2, 0xd5,
0x7f, 0x48, 0x11, 0x26, 0xa3, 0x94, 0xcd, 0xfa,
0xf0, 0xc7, 0x9e, 0xa9, 0x2c, 0x1b, 0x42, 0x75,
0x56, 0x61, 0x38, 0x0f, 0x8a, 0xbd, 0xe4, 0xd3,
0xd9, 0xee, 0xb7, 0x80, 0x05, 0x32, 0x6b, 0x5c,
0x2d, 0x1a, 0x43, 0x74, 0xf1, 0xc6, 0x9f, 0xa8,
0xa2, 0x95, 0xcc, 0xfb, 0x7e, 0x49, 0x10, 0x27,
0x04, 0x33, 0x6a, 0x5d, 0xd8, 0xef, 0xb6, 0x81,
0x8b, 0xbc, 0xe5, 0xd2, 0x57, 0x60, 0x39, 0x0e,
0xdb, 0xec, 0xb5, 0x82, 0x07, 0x30, 0x69, 0x5e,
0x54, 0x63, 0x3a, 0x0d, 0x88, 0xbf, 0xe6, 0xd1,
0xf2, 0xc5, 0x9c, 0xab, 0x2e, 0x19, 0x40, 0x77,
0x7d, 0x4a, 0x13, 0x24, 0xa1, 0x96, 0xcf, 0xf8,
0x89, 0xbe, 0xe7, 0xd0, 0x55, 0x62, 0x3b, 0x0c,
0x06, 0x31, 0x68, 0x5f, 0xda, 0xed, 0xb4, 0x83,
0xa0, 0x97, 0xce, 0xf9, 0x7c, 0x4b, 0x12, 0x25,
0x2f, 0x18, 0x41, 0x76, 0xf3, 0xc4, 0x9d, 0xaa,
};
```

```
//CRCS2_POLYNOMIAL = 0x3b
```

```
//This table was generated via left shifting the MSB of each byte
```

```
const unsigned char CRC8Table3b[256] =
```

```
{
0x00, 0x3b, 0x76, 0x4d, 0xec, 0xd7, 0x9a, 0xa1,
0xe3, 0xd8, 0x95, 0xae, 0x0f, 0x34, 0x79, 0x42,
0xfd, 0xc6, 0x8b, 0xb0, 0x11, 0x2a, 0x67, 0x5c,
0x1e, 0x25, 0x68, 0x53, 0xf2, 0xc9, 0x84, 0xbf,
0xc1, 0xfa, 0xb7, 0x8c, 0x2d, 0x16, 0x5b, 0x60,
0x22, 0x19, 0x54, 0x6f, 0xce, 0xf5, 0xb8, 0x83,
0x3c, 0x07, 0x4a, 0x71, 0xd0, 0xeb, 0xa6, 0x9d,
0xdf, 0xe4, 0xa9, 0x92, 0x33, 0x08, 0x45, 0x7e,
0xb9, 0x82, 0xcf, 0xf4, 0x55, 0x6e, 0x23, 0x18,
0x5a, 0x61, 0x2c, 0x17, 0xb6, 0x8d, 0xc0, 0xfb,
0x44, 0x7f, 0x32, 0x09, 0xa8, 0x93, 0xde, 0xe5,
0xa7, 0x9c, 0xd1, 0xea, 0x4b, 0x70, 0x3d, 0x06,
};
```

```

0x78, 0x43, 0x0e, 0x35, 0x94, 0xaf, 0xe2, 0xd9,
0x9b, 0xa0, 0xed, 0xd6, 0x77, 0x4c, 0x01, 0x3a,
0x85, 0xbe, 0xf3, 0xc8, 0x69, 0x52, 0x1f, 0x24,
0x66, 0x5d, 0x10, 0x2b, 0x8a, 0xb1, 0xfc, 0xc7,
0x49, 0x72, 0x3f, 0x04, 0xa5, 0x9e, 0xd3, 0xe8,
0xaa, 0x91, 0xdc, 0xe7, 0x46, 0x7d, 0x30, 0x0b,
0xb4, 0x8f, 0xc2, 0xf9, 0x58, 0x63, 0x2e, 0x15,
0x57, 0x6c, 0x21, 0x1a, 0xbb, 0x80, 0xcd, 0xf6,
0x88, 0xb3, 0xfe, 0xc5, 0x64, 0x5f, 0x12, 0x29,
0x6b, 0x50, 0x1d, 0x26, 0x87, 0xbc, 0xf1, 0xca,
0x75, 0x4e, 0x03, 0x38, 0x99, 0xa2, 0xef, 0xd4,
0x96, 0xad, 0xe0, 0xdb, 0x7a, 0x41, 0x0c, 0x37,
0xf0, 0xcb, 0x86, 0xbd, 0x1c, 0x27, 0x6a, 0x51,
0x13, 0x28, 0x65, 0x5e, 0xff, 0xc4, 0x89, 0xb2,
0x0d, 0x36, 0x7b, 0x40, 0xe1, 0xda, 0x97, 0xac,
0xee, 0xd5, 0x98, 0xa3, 0x02, 0x39, 0x74, 0x4f,
0x31, 0x0a, 0x47, 0x7c, 0xdd, 0xe6, 0xab, 0x90,
0xd2, 0xe9, 0xa4, 0x9f, 0x3e, 0x05, 0x48, 0x73,
0xcc, 0xf7, 0xba, 0x81, 0x20, 0x1b, 0x56, 0x6d,
0x2f, 0x14, 0x59, 0x62, 0xc3, 0xf8, 0xb5, 0x8e,
};

```

```

/** GetCRC32Ethernet

```

```

    \par Purpose:
        CRC32 calculation routine for polynomial 0xedb88320 which is bit
        reversed
        from the regular polynomial of 0x04c11db7 thus the (crc >> 8)
        below
        is right shifted.

    \note
        Doing incremental crc calculation is done with providing the
        result
        from the previous block with preset.

    \note
        Restrictions:
            len > 0
            pStart points to valid memory with at least len bytes in size

    \param Input: \b pStart      Starting address to compute CRC over
    \param Input: \b len        Number of bytes
    \param Input: \b preset     Preset value for CRC

```

```

*/
unsigned long GetCRC32Ethernet(const void *pStart, int len, unsigned
long preset)
{
    unsigned long crc = preset;
    unsigned char *buf = (unsigned char *)pStart;

    while (len-- > 0)
    {
        unsigned char data8 = *buf++;
        crc = CRC32Table[(crc ^ data8) & 0xff] ^ (crc >> 8);
    }
    return crc;
} //GetCRC32Ethernet

```

```

/** ComputeCRCS1

```

```

\par Purpose:
CRC8 calculation routine for polynomial 0x37.
Note that the CRC8Table37 was computed via left shift operations.

\note
Doing incremental crc calculation is done with providing the
result
from the previous block with preset.

\note
Restrictions:
    len > 0
    pStart points to valid memory with at least len bytes in size

\par Scope: public

\param Input: \b pStart      Starting address to compute CRC over
\param Input: \b len        Number of bytes
\param Input: \b preset     Preset value for CRC

\par Edit History:
*/
unsigned char ComputeCRCS1(const void *pStart, int len, unsigned char
preset)
{
    unsigned char crc = preset;
    unsigned char *buf = (unsigned char *)pStart;

    while (len-- > 0)
    {
        crc = CRC8Table37[(crc ^ *buf++)];
    }
    return (crc);
} //ComputeCRCS1

/** ComputeCRCS2

\par Purpose:
CRC8 calculation routine for polynomial 0x3b.
Note that the CRC8Table3b was computed via left shift operations.

\note
Doing incremental crc calculation is done with providing the
result
from the previous block with preset.

\note
Restrictions:
    len > 0
    pStart points to valid memory with at least len bytes in size

\par Scope: public

\param Input: \b pStart      Starting address to compute CRC over
\param Input: \b len        Number of bytes
\param Input: \b preset     Preset value for CRC

\par Edit History:
*/

```

```

unsigned char ComputeCRCS2(const void *pStart, int len, unsigned char
preset)
{
    unsigned char crc = preset;
    unsigned char *buf = (unsigned char *)pStart;

    while (len-- > 0)
    {
        crc = CRC8Table3b[(crc ^ *buf++)];
    }
    return (crc);
} //ComputeCRCS2

```

```

/xx ComputeCRCS3

```

```

\par Purpose:
CRC16 calculation routine for polynomial 0x080F.
The CRC16Table was computed via left shift operations and thus
will do
a (crc << 8).

\note
Doing incremental crc calculation is done with providing the
result
from the previous block with preset.

\note
Restrictions:
    len > 0
    pStart points to valid memory with at least len bytes in size

\param Input: \b pStart      Starting address to compute CRC over
\param Input: \b len        Number of bytes
\param Input: \b preset     Preset value for CRC

```

```

x/
unsigned short ComputeCRCS3(const void *pStart, int len, unsigned
short preset)
{
    unsigned short crc = preset;
    unsigned char *buf = (unsigned char *)pStart;

    while (len-- > 0)
    {
        unsigned char data = *buf++;
        crc = CRC16Table[((crc >> 8) ^ data)] ^ (crc << 8);
    }
    return (crc);
} //ComputeCRCS3

```

```

/xx ComputeCRCS3Ref4

```

```

\par Purpose:
Compute crc given 4 bytes of data and a preset for the crc.
This assumes the input data is stored in native (little or big)
endian format.

\par Scope: public

\param Input: \b data      Value to compute CRC on
\param Input: \b preset    Preset value for CRC

```



```

\par Edit History:
*/
unsigned short ComputeCRCS3Ref4(unsigned long data, unsigned short
preset)
{
    unsigned short crc = preset;
    crc = CRC16Table[(crc >> 8) ^ (data&0xff)] ^ (crc << 8);
    crc = CRC16Table[(crc >> 8) ^ ((data>>8)&0xff)] ^ (crc << 8);
    crc = CRC16Table[(crc >> 8) ^ ((data>>16)&0xff)] ^ (crc << 8);
    crc = CRC16Table[(crc >> 8) ^ ((data>>24)&0xff)] ^ (crc << 8);
    return crc;
}

/** ComputeCRCS3Ref2

\par Purpose:
Compute crc given 2 bytes of data and a preset for the crc.
This assumes the input data is stored in native (little or big)
endian format.

\par Scope: public

\param Input: \b data      Value to compute CRC on
\param Input: \b preset    Preset value for CRC

\par Edit History:
*/
unsigned short ComputeCRCS3Ref2(unsigned short data, unsigned short
preset)
{
    unsigned short crc = preset;
    crc = CRC16Table[(crc >> 8) ^ (data&0xff)] ^ (crc << 8);
    crc = CRC16Table[(crc >> 8) ^ ((data>>8)&0xff)] ^ (crc << 8);
    return crc;
}

/** ComputeCRCS3Ref1

\par Purpose:
Compute crc given 1 bytes of data and a preset for the crc.

\par Scope: public

\param Input: \b data      Value to compute CRC on
\param Input: \b preset    Preset value for CRC

\par Edit History:
*/
unsigned short ComputeCRCS3Ref1(unsigned char data, unsigned short
preset)
{
    unsigned short crc = preset;
    crc = CRC16Table[(crc >> 8) ^ data] ^ (crc << 8);
    return crc;
}

//The following routines were used to generate the above tables.
//Just as a sanity check, rerun the calculations and test that the
//tables are correct
unsigned char CRC8Table37Ref[256];
unsigned char CRC8Table3bRef[256];

```



```

unsigned short CRC16TableRef[256];
unsigned long CRC32TableRef[256];

//left shift MSB
static void InitSafetyCRC8Table()
{
    unsigned short j;
    unsigned char i;
    unsigned char carry8=0;
    unsigned char entry8=0;

    //CRCS1_POLYNOMIAL
    for (j = 0; j < 256; j++)
    {
        entry8 = static_cast<unsigned char>(j);
        for (i = 0; i < 8; ++i)
        {
            carry8 = entry8 & 0x80;    // carry8 is the most
significant bit
            entry8 = entry8 << 1;
            if (carry8)                // if carry8 is nonzero
            {
                entry8 = entry8 ^ CRCS1_POLYNOMIAL;    // XOR by CRC
polynom
            }
            CRC8Table37Ref[j] = entry8;
        }
    }
    //CRCS2_POLYNOMIAL
    for (j = 0; j < 256; j++)
    {
        entry8 = static_cast<unsigned char>(j);
        for (i = 0; i < 8; ++i)
        {
            carry8 = entry8 & 0x80;    // carry8 is the most
significant bit
            entry8 = entry8 << 1;
            if (carry8)                // if carry8 is nonzero
            {
                entry8 = entry8 ^ CRCS2_POLYNOMIAL;    // XOR by CRC
polynom
            }
            CRC8Table3bRef[j] = entry8;
        }
    }
} // end of initSafetyCRCTable8bit

//left shift MSB
static void InitSafetyCRC16Table()
{
    unsigned short j;
    unsigned char i;
    unsigned short carry16=0;
    unsigned short entry16=0;

    for (j = 0; j < 256; j++)
    {
        entry16 = j;
        for (i = 0; i < 16; ++i)
        {
            carry16 = entry16 & 0x8000;    // carry16 is the lowest
// significant bit
            entry16 = entry16 << 1;

```

```

        if (carry16)                // if carry16 is nonzero
        {
            entry16 = entry16 ^ CRCS3_POLYNOMIAL;    // XOR by CRC
polynom
        }
    }
    CRC16TableRef[j] = entry16;
}
} // end of initSafetyCRC16Table()

//right shift LSB
static void InitSafetyCRC32Table()
{
    unsigned long j;
    unsigned char i;
    unsigned long carry32=0;
    unsigned long entry32=0;

    //right shift
    for (j = 0; j < 256; j++)
    {
        entry32 = j;
        for (i = 0; i < 8; ++i)
        {
            carry32 = entry32 & 1;    // carry16 is the lowest
                                     // significant bit
            entry32 = entry32 >> 1;
            if (carry32)              // if carry16 is nonzero
            {
                entry32 = entry32 ^ CRC32_POLYNOMIAL;    // XOR by
                                                         // CRC
polynom
            }
        }
        CRC32TableRef[j] = entry32;
    }
} // end of initSafetyCRC32Table()

//This is the reference model that the table driven routine shall
match
unsigned char Crc8ComputeSlow(const void* buffer, unsigned int count,
unsigned char poly, unsigned char preset)
{
    const unsigned char* ptr = (const unsigned char *) buffer;
    unsigned char crc8=preset;
    unsigned char value=0;

    while (count--)
    {
        value = *ptr++;
        crc8 ^= value;
        for (int i = 0; i < 8; i++)
        {
            if (crc8 & 0x80)
            {
                crc8 = (crc8 << 1) ^ poly;
            }
            else
            {
                crc8 <<= 1;
            }
        }
    }
}

```

```

    }
    return crc8;
} //Crc8ComputeSlow

```

//This is the reference model that the table driven routine shall match

```

unsigned short Crc16ComputeSlow(const void* buffer, unsigned int
count, unsigned short poly, unsigned short preset)

```

```

{
    const unsigned char* ptr = (const unsigned char *) buffer;
    unsigned short crc16=preset;
    unsigned short value=0;

    while (count--)
    {
        value = *ptr++;
        crc16 ^= (value << 8);
        for (int i = 0; i < 8; i++)
        {
            if (crc16 & 0x8000)
            {
                crc16 = (crc16 << 1) ^ poly;
            }
            else
            {
                crc16 <<= 1;
            }
        }
    }
    return crc16;
} //Crc16ComputeSlow

```

//This is the reference model that the table driven routine shall match

```

unsigned long Crc32ComputeSlow(const void* buffer, unsigned int count,
unsigned long poly, unsigned long preset)

```

```

{
    const unsigned char* ptr = (const unsigned char *) buffer;
    unsigned long crc32=preset;
    unsigned long value=0;

    while (count--)
    {
        value = *ptr++;
        crc32 ^= (value);
        for (int i = 0; i < 8; i++)
        {
            if (crc32 & 1)
            {
                crc32 = (crc32 >> 1) ^ poly;
            }
            else
            {
                crc32 >>= 1;
            }
        }
    }
    return crc32;
} //Crc32ComputeSlow

```

//define a structure such that it is 9 bytes long in packed format

```
//so as to equal the crc test string length (in DoCrcTest below)
struct SomeStruct
{
    unsigned char  aByte1;
    unsigned char  aByte2;
    unsigned char  aByte3;
    unsigned short aShort;
    unsigned long  aLong;
};

int DoCrcTest()
{
    const char *string = "123456789";
    int len = 9;          //length of "123456789"
    unsigned long  crc32;
    unsigned short crc16;
    unsigned char  crc8;
    SomeStruct aStruct;

    //fill aStruct with equivalent to "123456789" in
    //little endian memory order
    aStruct.aByte1  = 0x31;
    aStruct.aByte2  = 0x32;
    aStruct.aByte3  = 0x33;
    aStruct.aShort  = 0x3534;
    aStruct.aLong   = 0x39383736;
    int crcMatch = 0;

    //the slow methods serve as the reference implementation
    //all other optimizations (ie table driven) shall match these
    crc8 = Crc8ComputeSlow(string, len, CRCS1_POLYNOMIAL, 0xff);
    if (crc8 != 0x4C)
    {
        crcMatch |= 1;
    }
    crc8 = Crc8ComputeSlow(string, len, CRCS2_POLYNOMIAL, 0xff);
    if (crc8 != 0xBF)
    {
        crcMatch |= 2;
    }
    crc16 = Crc16ComputeSlow(string, len, CRCS3_POLYNOMIAL, 0xffff);
    if (crc16 != 0x9516)
    {
        crcMatch |= 4;
    }

    crc32 = Crc32ComputeSlow(string, len, CRC32_POLYNOMIAL,
0xffffffff);
    if (crc32 != 0x340bc6d9)
    {
        crcMatch |= 8;
    }

    crc8 = ComputeCRCS1(string, len, 0xff);
    if (crc8 != 0x4C)
    {
        crcMatch |= 0x10;
    }
    crc8 = ComputeCRCS2(string, len, 0xff);
    if (crc8 != 0xBF)
    {

```

```

        crcMatch |= 0x20;
    }
    crc16 = ComputeCRCS3(string, len, 0xffff);
    if (crc16 != 0x9516)
    {
        crcMatch |= 0x40;
    }

    crc32 = GetCRC32Ethernet(string, len, 0xffffffff);
    if (crc32 != 0x340bc6d9)
    {
        crcMatch |= 0x80;
    }

    //retest that the tables are correct
    InitSafetyCRC8Table();
    InitSafetyCRC16Table();
    InitSafetyCRC32Table();

    if (memcmp(&CRC8Table37Ref[0], &CRC8Table37[0], 256) != 0)
    {
        crcMatch |= 0x200;
    }
    if (memcmp(&CRC8Table3bRef[0], &CRC8Table3b[0], 256) != 0)
    {
        crcMatch |= 0x400;
    }
    if (memcmp(&CRC16TableRef[0], &CRC16Table[0], 512) != 0)
    {
        crcMatch |= 0x800;
    }
    if (memcmp(&CRC32TableRef[0], &CRC32Table[0], 1024) != 0)
    {
        crcMatch |= 0x1000;
    }

    //should work on both big and little endian machine
    crc16 = 0xffff;
    crc16 = ComputeCRCS3Ref1(aStruct.aByte1, crc16);
    crc16 = ComputeCRCS3Ref1(aStruct.aByte2, crc16);
    crc16 = ComputeCRCS3Ref1(aStruct.aByte3, crc16);
    crc16 = ComputeCRCS3Ref2(aStruct.aShort, crc16);
    crc16 = ComputeCRCS3Ref4(aStruct.aLong, crc16);
    if (crc16 != 0x9516)
    {
        crcMatch |= 0x2000;
    }

    return crcMatch;
}

int main(int argc, char* argv[])
{
    int result = DoCrcTest();
    if (result==0)
        printf("Tests passed!\n");
    else
        printf("Tests failed, result = 0x%x\n",result);
    return result;
}

```

Bibliography

- [1] IEC 60050 (all parts), *International Electrotechnical Vocabulary*
NOTE See also the IEC Multilingual Dictionary – Electricity, Electronics and Telecommunications (available on CD-ROM and at <<http://domino.iec.ch/iev>>).
- [2] IEC 60204-1, *Safety of machinery – Electrical equipment of machines – Part 1: General requirements*
- [3] IEC 61131-6: *Programmable controllers – Part 6: Functional safety*⁷
- [4] IEC 61158 (all parts), *Industrial communication networks – Fieldbus specifications*
- [5] IEC 61496, *Safety of machinery – Electro-sensitive protective equipment*
- [6] IEC 61508-4:1998, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*
- [7] IEC 61508-6:1998, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3*
- [8] IEC 61511 (all parts), *Functional safety – Safety instrumented systems for the process industry sector*
- [9] IEC 61784-4, *Industrial communication networks – Profiles – Part 4: Secure communications for fieldbuses*⁸
- [10] IEC 61784-5 (all parts), *Industrial communication networks – Profiles – Part 5: Installation of fieldbuses – Installation profiles for CPF x*
- [11] IEC 61800-5-2, *Adjustable speed electrical power drive systems – Part 5-2: Safety Requirements – Functional*
- [12] IEC/TR 62059-11, *Electricity metering equipment – Dependability – Part 11: General concepts*
- [13] IEC 62061, *Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems*
- [14] IEC 62280-1:2002, *Railway applications – Communication, signalling and processing systems – Part 1: Safety-related communication in closed transmission systems*
- [15] IEC 62280-2, *Railway applications – Communication, signalling and processing systems – Part 2: Safety-related communication in open transmission systems*
- [16] IEC 62443, *Security for industrial process measurement and control – Network and system security*⁸
- [17] ISO/IEC 2382-14, *Information technology – Vocabulary – Part 14: Reliability, maintainability and availability*
- [18] ISO/IEC 2382-16, *Information technology – Vocabulary – Part 16: Information theory*
- [19] ISO/IEC 7498 (all parts), *Information technology – Open Systems Interconnection – Basic Reference Model*
- [20] ISO 12100-1, *Safety of machinery – Basic concepts, general principles for design – Part 1: Basic terminology, methodology*
- [21] ISO 13849-1, *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*
- [22] ISO 13849-2, *Safety of machinery – Safety-related parts of control systems – Part 2: Validation*
- [23] ISO 14121, *Safety of machinery – Principles of risk assessment*
- [24] EN 954-1:1996, *Safety of machinery – Safety related parts of control systems – General principles for design*

⁷ Under consideration.

⁸ In preparation.

- [25] ANSI/ISA-84.00.01-2004 (all parts), *Functional Safety: Safety Instrumented Systems for the Process Industry Sector*
- [26] VDI/VDE 2180 (all parts), *Safeguarding of industrial process plants by means of process control engineering*
- [27] GS-ET-26; *Grundsatz für die Prüfung und Zertifizierung von Bussystemen für die Übertragung sicherheitsrelevanter Nachrichten*, May 2002. HVBG, Gustav-Heinemann-Ufer 130, D-50968 Köln ("*Principles for Test and Certification of Bus Systems for Safety relevant Communication*")⁹
- [28] Andrew S. Tanenbaum, *Computer Networks*, 2nd Edition, Prentice Hall, N.J., ISBN 0-13-162959-X
- [29] W. Wesley Peterson, *Error-Correcting Codes*, 2nd Edition 1981, MIT-Press, ISBN 0-262-16-039-0
- [30] Bruce P. Douglass, *Doing Hard Time*, 1999, Addison-Wesley, ISBN 0-201-49837-5
- [31] *New concepts for safety-related bus systems*, 3rd International Symposium "Programmable Electronic Systems in Safety Related Applications", May 1998, from Dr. Michael Schäfer, BG-Institute for Occupational Safety and Health.
- [32] Dieter Conrads, *Datenkommunikation*, 3rd Edition 1996, Vieweg, ISBN 3-528-245891
- [33] German IEC subgroup DKE AK 767.0.4: *EMC and Functional Safety*, Spring 2002
- [34] NFPA79 (2002): *Electrical Standard for Industrial Machinery*
- [35] Guy E. Castagnoli, *On the Minimum Distance of Long Cyclic Codes and Cyclic Redundancy-Check Codes*, 1989, Dissertation No. 8979 of ETH Zurich, Switzerland
- [36] Guy E. Castagnoli, Stefan Bräuer, and Martin Herrmann, *Optimization of Cyclic Redundancy-Check Codes with 24 and 32 Parity Bits*, June 1993, IEEE Transactions On Communications, Volume 41, No. 6

⁹ This document has been one of the starting points for this part. It is currently undergoing a major revision.

SOMMAIRE

AVANT-PROPOS	243
INTRODUCTION.....	245
1 Domaine d'application	249
2 Références normatives.....	249
3 Termes, définitions, symboles, abréviations et conventions	251
3.1 Termes et définitions	251
3.1.1 Termes et définitions communs	251
3.1.2 CPF 2: Termes et définitions supplémentaires	255
3.2 Symboles et abréviations	255
3.2.1 Symboles et abréviations communs.....	255
3.2.2 CPF 2: Symboles et abréviations supplémentaires	255
3.3 Conventions	256
4 Présentation générale du FSCP 2/1 (CIP Safety™)	257
4.1 Généralités.....	257
4.2 FSCP 2/1	257
5 Généralités.....	258
5.1 Documents externes de spécifications applicables au profil.....	258
5.2 Exigences fonctionnelles de sécurité.....	259
5.3 Mesures de sécurité	259
5.4 Structure de la couche de communication de sécurité	260
5.5 Relations avec la FAL (et DLL, Phb).....	261
5.5.1 Généralités.....	261
5.5.2 Types de données.....	261
6 Services de la couche de communication de sécurité	261
6.1 Introduction.....	261
6.2 Objet de connexion.....	261
6.2.1 Généralités.....	261
6.2.2 Extensions des attributs des classes	261
6.2.3 Extensions de services	262
6.2.4 Format de réponse de messages explicites pour SafetyOpen et SafetyClose.....	263
6.3 Objet Gestionnaire de connexion.....	263
6.3.1 Généralités.....	263
6.3.2 ForwardOpen pour la sécurité.....	264
6.3.3 Segment de réseau de sécurité	266
6.3.4 Règles du point d'origine pour le calcul du CRC de paramètre de connexion.....	268
6.3.5 Organigrammes de traitement de SafetyOpen	268
6.3.6 Contrôles requis par les producteurs multipoint avec les connexions existantes.....	272
6.3.7 Utilisation de la clé électronique de sécurité	273
6.3.8 RPI en fonction de API dans les connexions de sécurité.....	273
6.3.9 Élaboration du chemin d'application de sécurité	273
6.3.10 Types de connexion de l'objet de validation de sécurité.....	275
6.3.11 Données de réponse d'application dans une réponse SafetyOpen satisfaisante	277

6.3.12	Réponse SafetyOpen non satisfaisante	277
6.3.13	Service ForwardClose de sécurité	280
6.4	Objet Identité	280
6.4.1	Généralités.....	280
6.4.2	Modifications apportées aux services communs.....	280
6.5	Objets de liaison	281
6.5.1	Modifications apportées à l'objet DeviceNet.....	281
6.5.2	Modifications apportées à l'objet Interface TCP/IP	281
6.6	Objet Programme de contrôle de sécurité.....	282
6.6.1	Généralités.....	282
6.6.2	Attributs de classe de programme de contrôle de sécurité	282
6.6.3	Sous-classes	283
6.6.4	Attributs instance de programme de contrôle de sécurité	283
6.6.5	Sémantique	286
6.6.6	Sous-classes	293
6.6.7	Services communs du programme de contrôle de sécurité	294
6.6.8	Comportement du Programme de contrôle de sécurité	307
6.7	Objet de validation de sécurité	316
6.7.1	Généralités.....	316
6.7.2	Attributs de classe	316
6.7.3	Attributs instance.....	317
6.7.4	Services de classes	323
6.7.5	Services instance	323
6.7.6	Comportement d'objet	324
6.8	Objet de configuration de connexion.....	328
6.8.1	Généralités.....	328
6.8.2	Extensions d'attributs de classe.....	328
6.8.3	Attributs, ajouts et extensions instance.....	328
6.8.4	Extensions ou restrictions de sémantique des attributs instance pour des raisons de sécurité.....	331
6.8.5	Paramètres spéciaux relatifs à la sécurité – (Attribut 13)	336
6.8.6	Services spécifiques à l'objet.....	340
6.8.7	Extensions des services communs pour la sécurité	341
6.8.8	Comportement d'objet	342
7	Protocole de couche de communication de sécurité.....	344
7.1	Format PDU de sécurité	344
7.1.1	Codage PDU de sécurité	344
7.1.2	CRC de sécurité	355
7.2	Comportement du protocole de communication	356
7.2.1	Séquence des contrôles de sécurité	356
7.2.2	Terminaison de connexion.....	356
7.2.3	Erreur de contre vérification	357
7.3	Opération de datation.....	357
7.4	Diagrammes séquentiels de protocoles	358
7.4.1	Généralités.....	358
7.4.2	Transmission de sécurité normale	358
7.4.3	Transmission de messages perdus, corrompus et retardés.....	360
7.4.4	Transmission de messages perdus, corrompus ou retardés avec production répétée	363

7.4.5	Demande Ping point à point.....	366
7.4.6	Demande Ping multipoint avec sécurité CP 2/3.....	367
7.4.7	Demande Ping multipoint sur les réseaux de sécurité CP 2/2	369
7.4.8	Demande Ping multipoint – Retransmission aboutie	371
7.4.9	Demande Ping multipoint – Retransmission avec temporisation.....	371
7.5	Définition du protocole de sécurité	372
7.5.1	Généralités.....	372
7.5.2	Vue de haut niveau d'un dispositif de sécurité	373
7.5.3	Objet de validation de sécurité	374
7.5.4	Relations entre SafetyValidatorServer et SafetyValidatorClient.....	374
7.5.5	Définition de la fonction SafetyValidatorClient	376
7.5.6	Définition de la fonction SafetyValidatorServer	384
7.6	Spécifications des messages de sécurité et des données de protocole.....	395
7.6.1	Octet de mode.....	395
7.6.2	Section de datation.....	396
7.6.3	Message de coordination temporelle	396
7.6.4	Message de correction de temps	396
7.6.5	Production des données de sécurité	397
7.6.6	Variables dynamiques du producteur.....	404
7.6.7	Variables dynamiques de producteur par consommateur	407
7.6.8	Variables de données du consommateur	408
7.6.9	Variables statiques d'entrée du consommateur.....	410
7.6.10	Variables dynamiques du consommateur.....	411
8	Gestion de la couche de communication de sécurité.....	413
8.1	Présentation générale.....	413
8.2	Définition des mesures utilisées lors de l'établissement d'une connexion	413
8.3	Validation de la relation point d'origine-cible	417
8.4	Détection des demandes de connexion mal acheminées	418
8.5	Traitement de SafetyOpen.....	418
8.6	Gestion de propriété.....	418
8.7	Pontage de différentes couches physiques.....	420
8.8	Établissement de connexion de sécurité.....	421
8.8.1	Présentation générale	421
8.8.2	Faits de base pour l'établissement d'une connexion	422
8.8.3	Configuration des connexions de sécurité.....	422
8.8.4	Multiplicateur de délai de réseau	423
8.8.5	Établissement de connexions	425
8.8.6	Recommandations pour l'attribution d'un nombre de consommateurs	428
8.8.7	Recommandations pour l'établissement d'une connexion	429
8.8.8	Établissement de propriété	430
8.8.9	Cas d'utilisation de la propriété	430
8.8.10	Utilisation et établissement de la relation PID/CID	433
8.8.11	Utilisation PID/CID correcte dans les connexions multipoint et point à point.....	434
8.8.12	Services pris en charge par le réseau.....	438
8.8.13	Type de dispositif de sécurité FSCP 2/1	438
8.9	Processus de configuration de sécurité	443
8.9.1	Introduction à la configuration de sécurité	443
8.9.2	Objectifs de configuration	443

8.9.3	Présentation générale de la configuration	444
8.9.4	Lignes directrices pour la configuration de l'utilisateur	445
8.9.5	Justification de niveau SIL3 du processus de configuration	446
8.9.6	Fonctions de dispositifs pour configuration par outil	447
8.9.7	Sécurité par mot de passe	447
8.9.8	Services d'interface SNCT	447
8.9.9	Verrouillage de configuration	448
8.9.10	Influence du verrouillage de configuration sur le comportement des dispositifs	448
8.9.11	Propriété de configuration	450
8.9.12	Mode de configuration	450
8.9.13	Mesures d'assurance de l'intégrité du processus de configuration	450
8.9.14	Processus de téléchargement aval	453
8.9.15	Processus de vérification	457
8.9.16	Processus de vérification	460
8.9.17	Analyse des erreurs de configuration	462
8.10	Extensions de fiches techniques électroniques à des fins de sécurité	466
8.10.1	Règles générales applicables aux dispositifs de sécurité EDS	466
8.10.2	Extensions EDS à des fins de sécurité	467
9	Exigences système	471
9.1	Voyants et commutateurs	471
9.1.1	Exigences générales concernant les voyants	471
9.1.2	Indications DEL pour le paramétrage de l'UNID des dispositifs	471
9.1.3	DEL Etat du module	472
9.1.4	Avertissement lié aux voyants	472
9.1.5	DEL Etat de réseau	472
9.1.6	Détermination du MACID	474
9.1.7	Commutateur de réinitialisation	476
9.2	Lignes directrices d'installation	476
9.3	Temps de réponse de la fonction de sécurité	476
9.3.1	Présentation générale	476
9.3.2	Délai du réseau	478
9.3.3	Equations de calcul des temps de réaction du réseau	478
9.4	Durée des demandes	481
9.5	Contraintes liées au calcul des caractéristiques du système	481
9.5.1	Nombre de nœuds	481
9.5.2	PFH de réseau	481
9.5.3	Taux d'erreurs sur les bits (BER)	483
9.6	Maintenance	484
9.7	Manuel de sécurité	484
10	Certification	484
Annexe A (informative) Informations supplémentaires pour les profils de communication de sécurité fonctionnelle de la CPF 2		485
A.1 Exemple de code de fonction de hachage		485
Bibliographie		497

Tableau 1 – Erreurs de communication et matrice de mesures de détection..... 259

Tableau 2 – Nouveaux attributs de classe..... 261

Tableau 3 – Extensions de services	263
Tableau 4 – Format de réponse SafetyOpen et SafetyClose	263
Tableau 5 – Identifiant de segment de réseau de sécurité.....	266
Tableau 6 – Définition du segment de réseau de sécurité	266
Tableau 7 – Format de routage de segment de réseau de sécurité	268
Tableau 8 – Règles d'évaluation des paramètres de production multipoint.....	273
Tableau 9 – Options de paramétrage du service ForwardOpen pour les connexions de sécurité.....	275
Tableau 10 – Paramètres de connexion de réseau pour les connexions de sécurité.....	276
Tableau 11 – Réponse d'application de cible de sécurité CP 2/3 (taille: 10 octets)	277
Tableau 12 – Réponse d'application de cible SafetyOpen (taille: 16 octets)	277
Tableau 13 – Nouveaux codes d'erreurs étendus de sécurité.....	278
Tableau 14 – Tableau de recommandations concernant les événements d'erreur SafetyOpen.....	279
Tableau 15 – Modifications apportées aux services communs de l'objet Identité	280
Tableau 16 – Nouvel attribut instance de l'objet DeviceNet.....	281
Tableau 17 – Nouvel attribut instance de l'objet Interface TCP/IP	281
Tableau 18 – Attributs de classe de programme de contrôle de sécurité	283
Tableau 19 – Attributs instance de programme de contrôle de sécurité.....	283
Tableau 20 – Valeurs d'indication de l'attribut Etat du dispositif	287
Tableau 21 – Format de l'attribut Etat d'exception	288
Tableau 22 – Valeurs communes de l'attribut Détail d'exception	289
Tableau 23 – Synthèse du format de détail d'exception.....	290
Tableau 24 – Synthèse du comportement des dispositifs pour diverses valeurs CFUNID	292
Tableau 25 – Services communs du programme de contrôle de sécurité.....	294
Tableau 26 – Services spécifiques de l'objet Programme de contrôle de sécurité	294
Tableau 27 – Structure de message Configure_Request.....	296
Tableau 28 – Structure de message Validate_Configuration	297
Tableau 29 – Structure de message de réussite Validate_Configuration	297
Tableau 30 – Code d'erreur Validate_Configuration	297
Tableau 31 – Codes étendus Validate_Configuration	298
Tableau 32 – Structure de message Set_Password	300
Tableau 33 – Structure de message Reset_Password.....	301
Tableau 34 – Structure de message Configuration_Lock/Unlock	301
Tableau 35 – Structure de message Mode_Change	302
Tableau 36 – Structure de message Safety_Reset.....	302
Tableau 37 – Types de réinitialisation de sécurité du Programme de contrôle de sécurité.....	302
Tableau 38 – Paramètre d'attribut Mode point.....	303
Tableau 39 – Règles de traitement applicables aux types de réinitialisation.....	303
Tableau 40 – Service Propose_TUNID	304
Tableau 41 – Service Apply_TUNID	304
Tableau 42 – Événements liés au Programme de contrôle de sécurité	308

Tableau 43 – Matrice d'événements d'état du Programme de contrôle de sécurité	309
Tableau 44 – Contrôle du propriétaire de configuration par rapport à l'état du dispositif.....	314
Tableau 45 – Mise en correspondance des états du Programme de contrôle de sécurité et de l'objet Identité	314
Tableau 46 – Mise en correspondance des événements de l'objet Programme de contrôle de sécurité	315
Tableau 47 – Mise en correspondance des événements d'objet Identité	316
Tableau 48 – Attributs de classe de l'Objet de validation de sécurité	316
Tableau 49 – Attributs instance de l'Objet de validation de sécurité	317
Tableau 50 – Attributions d'états de l'Objet de validation de sécurité	319
Tableau 51 – Type d'Objet de validation de sécurité, attributions de champs de bits	319
Tableau 52 – Règles d'évaluation du paramètre SafetyOpen des producteurs multipoint.....	321
Tableau 53 – Services de classes de l'Objet de validation de sécurité	323
Tableau 54 – Services instance de l'Objet de validation de sécurité	323
Tableau 55 – Données du service Get_Attributes_All de l'objet de validation de sécurité.....	324
Tableau 56 – Matrice d'événements d'état de l'Objet de validation de sécurité	327
Tableau 57 – Mise en correspondance des états des objets Programme de contrôle de sécurité et Validation de sécurité	327
Tableau 58 – Extensions d'attributs de classe d'objet de configuration de connexion.....	328
Tableau 59 – Ajouts/extensions d'attributs instance d'Objet de configuration de connexion	329
Tableau 60 – Définitions des bits de drapeaux de connexion	331
Tableau 61 – Paramètres de connexion O-à-T	333
Tableau 62 – Paramètres de connexion T-à-O	334
Tableau 63 – Formats de mise en correspondance des données	335
Tableau 64 – Format 0 de mise en correspondance des données	336
Tableau 65 – Format 1 de mise en correspondance des données	336
Tableau 66 – Valeurs SCCRC du dispositif cible	339
Tableau 67 – Valeurs SCTS du dispositif cible.....	339
Tableau 68 – Paramètres de connexion de correction de temps pour une connexion multipoint.....	339
Tableau 69 – Services spécifiques à l'objet Configuration de connexion	340
Tableau 70 – Données du service de réponse Get_Attributes_All (attributs ajoutés)	341
Tableau 71 – Données du service de Demande Set_Attributes_All (attributs ajoutés)	342
Tableau 72 – Mise en correspondance des états des objets Programme de contrôle de sécurité et CCO	343
Tableau 73 – Sections de connexion et formats PDU	345
Tableau 74 – Variables de l'octet Mode	346
Tableau 75 – Variables de datation.....	348
Tableau 76 – Variables du message de coordination temporelle	349
Tableau 77 – Variables du message de correction de temps.....	350
Tableau 78 – Polynômes de CRC utilisés	355
Tableau 79 – Sections de connexion et formats de messages.....	356
Tableau 80 – Réception des données – Déclenchée par la liaison	387

Tableau 81 – Réception Time_Correction – Déclenchée par la liaison	387
Tableau 82 – Réception des données – Déclenchée par l'application.....	388
Tableau 83 – Réception de Time_Correction – Déclenchée par l'application	388
Tableau 84 – Application de consommation – Contrôle des données de sécurité	388
Tableau 85 – Détermination de l'état de la connexion du producteur.....	397
Tableau 86 – État de la connexion de sécurité de consommation.....	409
Tableau 87 – Erreurs d'établissement de connexion et mesures de détection des erreurs.....	413
Tableau 88 – Attributions de Date/Heure SNN	414
Tableau 89 – Plage légale SNN des valeurs temporelles	414
Tableau 90 – Paramètres de connexion de sécurité	423
Tableau 91 – Récapitulatif SafetyOpen	426
Tableau 92 – Mise en correspondance des services point d'origine/cible	439
Tableau 93 – Types de service point d'origine/cible non pris en charge	439
Tableau 94 – Objectifs de configuration	443
Tableau 95 – Contrôle du propriétaire de configuration par rapport à l'état du dispositif.....	450
Tableau 96 – Erreurs et mesures de détection	462
Tableau 97 – Mot clés de classes de paramètres	467
Tableau 98 – Nouveaux mots clés de la section Gestionnaire de connexion à des fins de sécurité.....	468
Tableau 99 – Utilisation du champ Gestionnaire de connexion à des fins de sécurité.....	469
Tableau 100 – Paramétrages du champ de paramètres de connexion à des fins de sécurité.....	470
Tableau 101 – Indications DEL pour le paramétrage de l'UNID	471
Tableau 102 – DEL Etat du module.....	472
Tableau 103 – Etats de la DEL Etat du réseau.....	473
Tableau 104 – Type de temps de réaction de connexion – applications de production/consommation	479
Figure 1 – Relations entre la CEI 61784-3 et d'autres normes (machines).....	246
Figure 2 – Relations entre la CEI 61784-3 et d'autres normes (transformation)	248
Figure 3 – Relation des objets de validation de sécurité.....	258
Figure 4 – Couches de communication	260
Figure 5 – ForwardOpen avec segment de réseau de sécurité	266
Figure 6 – Format cible de réseau de sécurité	267
Figure 7 – Service SafetyOpen de traitement de cible sans données de configuration (SafetyOpen de forme 2).....	270
Figure 8 – Service SafetyOpen de traitement de cible avec données de configuration (SafetyOpen de forme 1).....	272
Figure 9 – Application de configuration de dispositif	298
Figure 10 – Organigrammes de traitement de Configuration et Validation	300
Figure 11 – Traitement UNID pendant l'état "Waiting for TUNID".....	307
Figure 12 – Diagramme d'états du Programme de contrôle de sécurité	308
Figure 13 – Relations entre configuration, essais et verrouillage.....	313

Figure 14 – Types de connexion de sécurité	321
Figure 15 – Diagramme de transition d'état de l'Objet de validation de sécurité	326
Figure 16 – Diagramme d'états de l'objet Configuration de connexion.....	343
Figure 17 – Flux de données de l'objet Configuration de connexion	344
Figure 18 – Format de l'octet Mode.....	345
Figure 19 – Section de données à 1 ou 2 octets.....	346
Figure 20 – Format de section de données de 3 à 250 octets	347
Figure 21 – Format de section de datation	347
Figure 22 – Codage du message de coordination temporelle	348
Figure 23 – Codage du message de correction de temps	350
Figure 24 – Codage PDU point à point à 1 ou 2 octets	351
Figure 25 – Codage PDU multipoint à 1 ou 2 octets	352
Figure 26 – Format de connexion de sécurité de format 2 multipoint à 1 ou 2 octets.....	353
Figure 27 – Codage PDU point à point de 3 à 250 octets	353
Figure 28 – Codage PDU multipoint de 3 à 248 octets	354
Figure 29 – Format de connexion de sécurité multipoint de 3 à 248 octets.....	355
Figure 30 – Séquence de datation	357
Figure 31 – Diagramme d'une séquence de sécurité producteur/consommateur normale	359
Figure 32 – Diagramme d'une séquence de sécurité producteur/consommateur normale (production répétée).....	360
Figure 33 – Diagramme séquentiel d'un message producteur vers consommateur corrompu	361
Figure 34 – Diagramme séquentiel d'un message producteur vers consommateur perdu	362
Figure 35 – Diagramme séquentiel d'un message retardé	363
Figure 36 – Diagramme séquentiel d'un message producteur vers consommateur corrompu avec production répétée	364
Figure 37 – Diagramme séquentiel d'une terminaison de connexion occasionnée par des retards	365
Figure 38 – Diagramme séquentiel d'un contrôle CRC de sécurité non satisfait	366
Figure 39 – Diagramme séquentiel d'une demande Ping point à point – réponse normale	367
Figure 40 – Diagramme séquentiel d'une demande Ping multipoint satisfaisante, sécurité CP 2/3	369
Figure 41 – Diagramme séquentiel d'une demande Ping multipoint satisfaisante, sécurité CP 2/2	370
Figure 42 – Diagramme séquentiel d'une retransmission de demande Ping multipoint	371
Figure 43 – Diagramme séquentiel d'une temporisation de demande Ping multipoint	372
Figure 44 – Diagramme de relations des entités de modèle de référence d'un dispositif de sécurité.....	374
Figure 45 – Deux dispositifs échangeant des données de sécurité via un SafetyValidatorClient et un SafetyValidatorServer.....	375
Figure 46 – Flux de données de production de sécurité	376
Figure 47 – Contrôle des données de sécurité de consommation	385
Figure 48 – SafetyValidatorServer – déclenché par l'application	386

Figure 49 – Propriété cible.....	417
Figure 50 – Formes SafetyOpen	418
Figure 51 – Diagramme d'états de la propriété de connexion	419
Figure 52 – Mise en correspondance de l'UNID de service SafetyOpen	420
Figure 53 – Couche application CPF 2 commune	420
Figure 54 – Exemple d'acheminement de bout en bout	421
Figure 55 – Sources des paramètres de connexion relative à la sécurité	424
Figure 56 – Mise en correspondance des paramètres entre le point d'origine et la cible.....	425
Figure 57 – Établissement d'une connexion de sécurité CP 2/3 dans les cibles pour un service SafetyOpen de forme 2a	427
Figure 58 – Séquence générale de détection de la nécessité d'une configuration	428
Figure 59 – Echanges PID/CID pour deux scénarios de point d'origine	434
Figure 60 – Génération des valeurs de marquage pour les connexions multipoint	436
Figure 61 – Traitement d'exécution PID/CID	438
Figure 62 – Catégories de connexion et services pris en charge	440
Figure 63 – Types de connexion recommandés	441
Figure 64 – Services pris en charge logique à logique	442
Figure 65 – Types de connexion recommandés pour les services logique à logique.....	443
Figure 66 – Transferts de données de configuration	444
Figure 67 – Mesures de protection des dispositifs de sécurité.....	446
Figure 68 – Relations entre configuration, essais et verrouillage.....	449
Figure 69 – Données de configuration du point d'origine.....	451
Figure 70 – Processus de téléchargement aval du SNCT au dispositif	455
Figure 71 – Téléchargements aval du SNCT vers les points d'origine qui réalisent la configuration de forme 1	456
Figure 72 – Protection contre le verrouillage et propriété	458
Figure 73 – Exemple de relecture et de comparaison du point d'origine et de l'exemplaire	459
Figure 74 – Affichage divers sans relecture complète des données.....	460
Figure 75 – Processus de vérification comprenant toutes les méthodes alternatives	462
Figure 76 – Logique de traitement du MACID des dispositifs de sécurité	476
Figure 77 – Temps de réponse de la fonction de sécurité	477
Figure 78 – Composantes du temps de réponse de la fonction de sécurité	480
Figure 79 –Schéma de principe de fiabilité de protocole de réseau (RBD)	481
Figure 80 – Synthèse de la PFH de réseau	483

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉSEAUX DE COMMUNICATION INDUSTRIELS – PROFILS –

Partie 3-2: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 2

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

La commission électrotechnique internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité avec le présent document peut impliquer l'utilisation de brevets concernant les profils de communication de sécurité fonctionnelle pour la famille 2, où la notation [xx] désigne le détenteur des droits de propriété.

US 6,631,476	[RA]	Safety network for industrial controller providing redundant connections on single media
US 6,701,198	[RA]	Safety network for industrial controller allowing initialization on standard networks
US 6,721,900	[RA]	Safety network for industrial controller having reduced bandwidth requirements
US 6,891,850	[RA]	Network independent safety protocol for industrial controller
US 6,915,444	[RA]	Network independent safety protocol for industrial

contrôler using data manipulation techniques

La CEI ne prend pas position eu égard à la preuve, la validité et la portée de ces droits de propriété.

Les détenteurs de ces droits de propriété ont donné l'assurance à la CEI qu'ils consentent à négocier des licences avec des demandeurs du monde entier, en des termes et à des conditions raisonnables et non discriminatoires. A ce propos, la déclaration des détenteurs de ces droits de propriété est enregistrée à la CEI.

Des informations peuvent être obtenues auprès de:

[RA] Rockwell Automation, Inc.
1201 S. Second Street
Milwaukee, WI 53204
USA
Contact: Département de la propriété intellectuelle.

L'attention est attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux mentionnés ci-dessus. La CEI ne saurait être tenue pour responsable de ne pas avoir dûment signalé tout ou partie de ces droits de propriété.

La Norme internationale CEI 61784-3-2 a été établie par le sous-comité 65C: Réseaux de communication industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

La présente version bilingue (2013-06) correspond à la version anglaise monolingue publiée en 2007-12.

Le texte anglais de cette norme est issu des documents 65C/470/FDIS et 65C/481/RVD.

Le rapport de vote 65C/481/RVD donne toute information sur le vote ayant abouti à l'approbation de cette norme.

La version française de cette norme n'a pas été soumise au vote.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

La liste de toutes les parties de la série CEI 61784-3, présentées sous le titre général *Réseaux de communication industriels – Profils – Bus de terrain de sécurité fonctionnelle*, peut être consultée sur le site Web de la CEI.

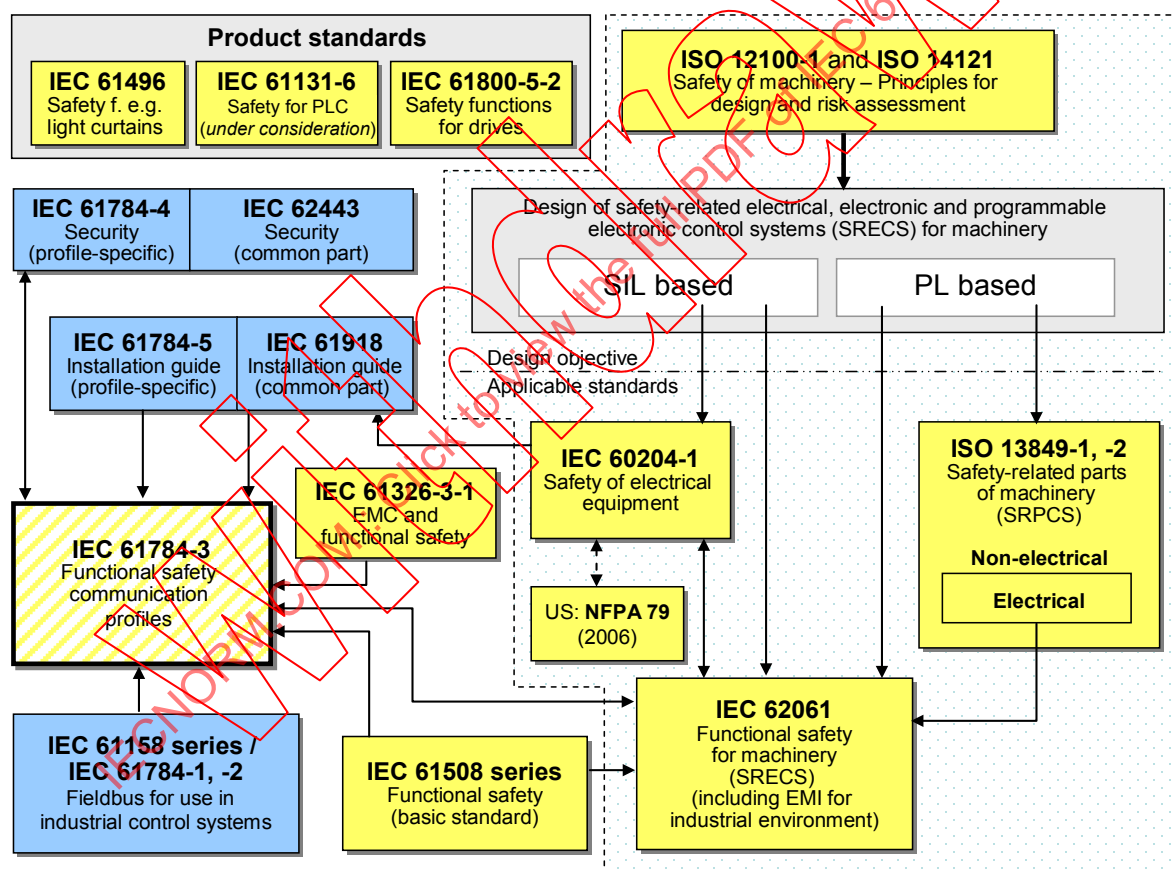
IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La CEI 61158 relative aux bus de terrain, ainsi que ses normes associées CEI 61784-1 et CEI 61784-2, définissent un ensemble de protocoles de communication qui assurent la commande répartie d'applications automatisées. La technologie de bus de terrain est désormais reconnue et bien éprouvée. Ainsi de nombreuses améliorations des bus de terrain se développent pour traiter de domaines non encore normalisés tels que les applications en temps réel relatives à la sécurité et à la sûreté.

La présente norme définit les principes pertinents applicables aux communications en termes de sécurité fonctionnelle en référence à la série CEI 61508, et spécifie plusieurs couches de communication de sécurité (profils et protocoles correspondants) basées sur les profils de communication et les couches de protocoles de la CEI 61784-1, de la CEI 61784-2 et de la série CEI 61158. Elle ne couvre pas les aspects relatifs à la sécurité électrique et à la sécurité intrinsèque.

La Figure 1 illustre les relations entre la présente norme et les normes pertinentes relatives à la sécurité et au bus de terrain dans un environnement machines.



Key

-  (yellow) safety-related standards
 (blue) fieldbus-related standards
 (dashed yellow) this standard

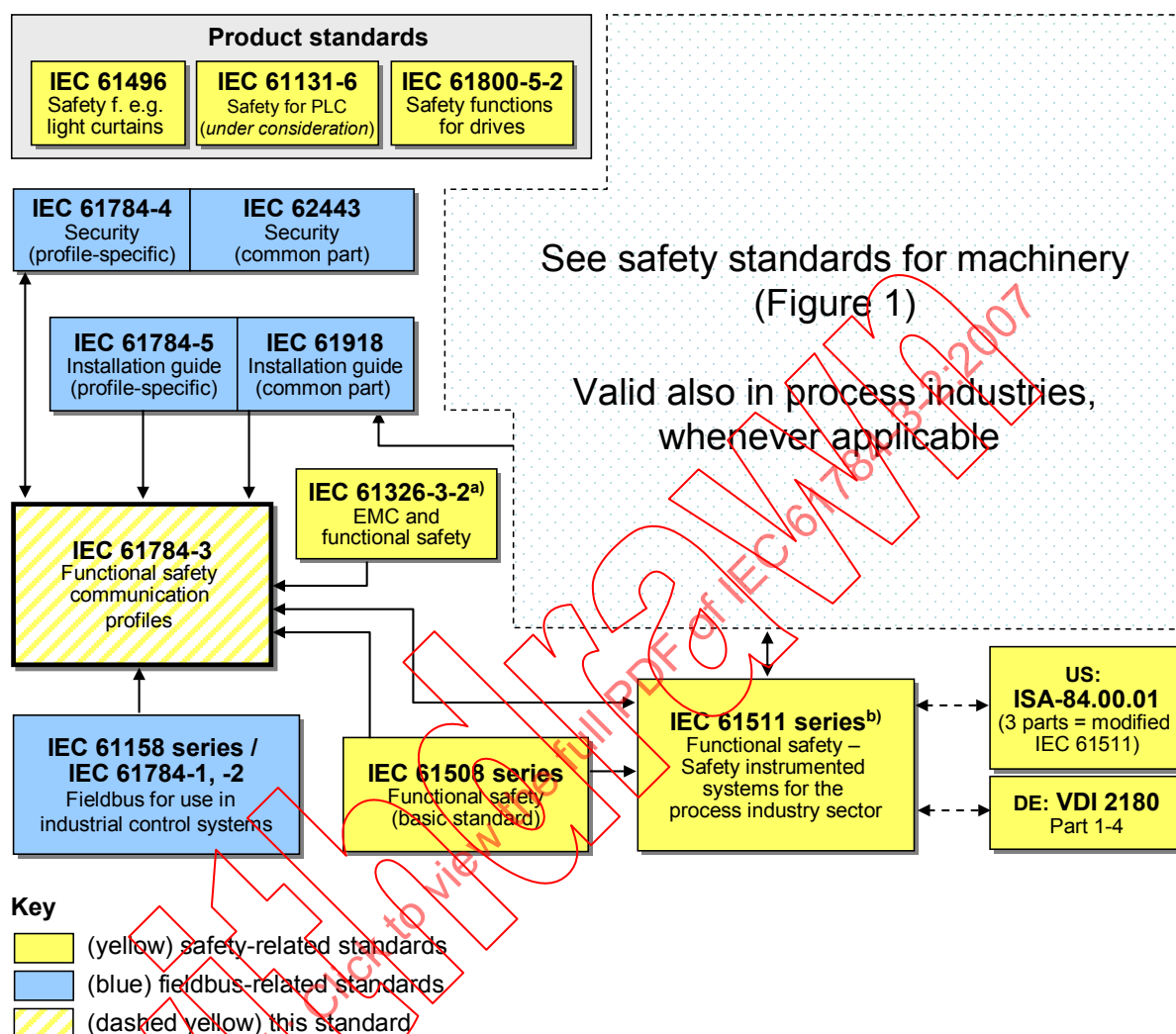
Légende

Anglais	Français
Product standards	Normes de produits

Anglais	Français
Safety function, e.g. light curtains	Fonction de sécurité, par exemple barrières photo électriques
Safety for PLC (under consideration)	Sécurité relative aux automates programmables (à l'étude)
Safety functions for drives	Fonctions de sécurité applicables aux entraînements
Safety requirements for robots	Exigences de sécurité applicables aux robots
Safety of machinery – Principles for design and risk assessment	Sécurité des machines – Principes généraux de conception et d'appréciation du risque
Security (profile-specific)	Sûreté (spécifique au profil)
Security (common part)	Sûreté (partie commune)
Design of safety-related electrical, electronic and programmable electronic control systems (SRECS) for machinery	Conception des systèmes de commande électriques, électroniques et électroniques programmables relatifs à la sécurité pour les machines
SIL based	Basé sur SIL
PL based	Basé sur PL
Installation guide (profile-specific)	Guide d'installation (spécifique au profil)
Installation guide (common part)	Guide d'installation (partie commune)
Design objective	Objectif de conception
Applicable standards	Normes applicables
Safety of electrical equipment	Sécurité des équipements électriques
Safety-related parts of machinery (SRPCS)	Parties des systèmes de commande relatives à la sécurité
Non-electrical	Non électrique
Electrical	Électrique
EMC and functional safety	Compatibilité électromagnétique et sécurité fonctionnelle
Functional safety communication profiles	Profils de communication de sécurité fonctionnelle
IEC 61158 series / IEC 61784-1,-2 Fieldbus for use in industrial control systems	Série CEI 61158 / CEI 61784-1,-2 Bus de terrain pour utilisation dans des systèmes de commande industriels
IEC 61508 series, Functional safety (basic standard)	Série CEI 61508 Sécurité fonctionnelle (norme de base)
Functional safety for machinery (SRECS) (including EMI for industrial environment)	Sécurité fonctionnelle des systèmes de commande électriques, électroniques et électroniques programmables (y compris les perturbations électromagnétiques dans l'environnement industriel)
Key	Légende
(yellow) safety-related standards	(jaune) normes relatives à la sécurité
(blue) fieldbus-related standards	(bleu) normes relatives au bus de terrain
(dashed) yellow) this standard	(jaunestrié) la présente norme

Figure 1 – Relations entre la CEI 61784-3 et d'autres normes (machines)

La Figure 2 illustre les relations entre la présente norme et les normes pertinentes relatives à la sécurité et au bus de terrain dans un environnement de transformation.



Légende

Anglais	Français
Product standards	Normes de produits
Safety f. e.g. light curtains	Fonction de sécurité, par exemple barrières photo électriques
Safety for PLC (under consideration)	Sécurité relative aux automates programmables (à l'étude)
Safety functions for drives	Fonctions de sécurité applicables aux entraînements
Safety requirements for robots	Exigences de sécurité applicables aux robots
Security (profile-specific)	Sûreté (spécifique au profil)
Security (common part)	Sûreté (partie commune)
Installation guide (profile-specific)	Guide d'installation (spécifique au profil)
Installation guide (common part)	Guide d'installation (partie commune)
See safety standards for machinery (Figure 1)	Voir normes de sécurité pour les machines (Figure 1)
Valid also in process industries, whenever applicable	Valable également dans les industries de transformation, le cas échéant
Functional safety communication profiles	Profils de communication de sécurité fonctionnelle

Anglais	Français
IEC 61326-3-2 ^{a)} EMC and functional safety	CEI 61326-3-2 ^{a)} CEM et sécurité fonctionnelle
IEC 61158 series/ IEC 61784-1,-2, Fieldbus for use in industrial control systems	Série CEI 61158/ CEI 61784-1,-2 Bus de terrain pour utilisation dans des systèmes de commande industriels
IEC 61508 series, Functional safety (basic standard)	Série CEI 61508 Sécurité fonctionnelle (norme de base)
IEC 61511 series ^{b)} Functional safety–safety instrumented systems for the process industry sector	Série CEI 61511 ^{b)} Sécurité fonctionnelle – Systèmes instrumentés de sécurité pour le secteur des industries de transformation
US: ISA-84.00.01 (3 parts = modified IEC 61511)	US: ISA-84.00.01 (3 parties = CEI 61511 modifiée)
DE: VDI 2180 Part 1 –4	DE: VDI 2180 Parties 1 à 4
Key	Légende
(yellow) safety-related standards	(jaune) normes relatives à la sécurité
(blue) fieldbus-related standards	(bleu) normes relatives au bus de terrain
(dashed) yellow) this standard	(jaune strié) la présente norme

^a Pour des environnements électromagnétiques spécifiés, sinon CEI 61326-3-1

^b EN ratifiée.

Figure 2 – Relations entre la CEI 61784-3 et d'autres normes (transformation)

Les couches de communication de sécurité mises en œuvre dans le cadre de systèmes relatifs à la sécurité conformément à la série CEI 61508, assurent la confiance nécessaire à accorder à la transmission de messages (information) entre deux participants ou plus sur un bus de terrain dans un système relatif à la sécurité, ou une fiabilité suffisante dans le comportement de sécurité en cas d'erreurs ou de défaillances du bus de terrain.

Les couches de communication de sécurité spécifiées dans la présente norme permettent de garantir cette assurance en utilisant un bus de terrain dans des applications nécessitant une sécurité fonctionnelle jusqu'au niveau d'intégrité de sécurité (SIL) spécifié par son profil de communication de sécurité fonctionnelle correspondant.

La revendication du SIL qui en résulte pour un système dépend de la mise en œuvre du profil de communication de sécurité fonctionnelle retenu au sein du système – la mise en œuvre du profil de communication de sécurité fonctionnelle dans un dispositif normal ne suffit pas à le qualifier de dispositif de sécurité.

La présente norme décrit:

- les principes de base de mise en œuvre des exigences de la série CEI 61508 pour les communications de données relatives à la sécurité, y compris les défauts de transmission potentiels, les mesures correctives et les considérations concernant l'intégrité des données;
- la description individuelle des profils de sécurité fonctionnelle pour plusieurs familles de profils de communication dans la CEI 61784-1 et la CEI 61784-2;
- les extensions de la couche de sécurité aux sections relatives au service et aux protocoles de communication de la série CEI 61158.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – PROFILS –

Partie 3-2: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 2

1 Domaine d'application

La présente partie de la série CEI 61784-3 spécifie une couche de communication relative à la sécurité (services et protocole) fondée sur la CPF 2 de la CEI 61784-1, la CEI 61784-2 et le Type 2 de la CEI 61158. Elle identifie les principes applicables aux communications de sécurité fonctionnelle définies dans la CEI 61784-3, et appropriées à cette couche de communication de sécurité.

NOTE 1 Elle ne couvre pas les aspects relatifs à la sécurité électrique et à la sécurité intrinsèque. La sécurité électrique concerne les dangers tels que les chocs électriques. La sécurité intrinsèque concerne les dangers associés aux atmosphères explosibles.

La présente partie¹ définit les mécanismes de transmission des messages relatifs à la sécurité entre les participants d'un réseau réparti, en utilisant la technologie de bus de terrain conformément aux exigences de la série CEI 61508 concernant la sécurité fonctionnelle. Ces mécanismes peuvent être utilisés dans diverses applications industrielles, telles que la commande de processus, l'usinage automatique et les machines.

La présente partie fournit des lignes directrices tant pour les développeurs que pour les évaluateurs de dispositifs et systèmes conformes.

NOTE 2 La revendication du SIL qui en résulte pour un système dépend de la mise en œuvre du profil de communication de sécurité fonctionnelle retenu au sein du système – la mise en œuvre du profil de communication de sécurité fonctionnelle, conforme à la présente partie, dans un dispositif normal ne suffit pas à le qualifier de dispositif de sécurité.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61131-2, *Programmable controllers – Part 2: Equipment requirements and tests* (disponible uniquement en anglais)

IEC 61131-3, *Programmable controllers – Part 3: Programming languages* (disponible uniquement en anglais)

CEI 61158-2, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 2: Spécification de couche physique et définition des services*

IEC 61158-3-2, *Industrial communication networks – Fieldbus specifications – Part 3-2: Data-link layer service definition* (disponible uniquement en anglais)

IEC 61158-4-2, *Industrial communication networks – Fieldbus specifications – Part 4-2: Data-link layer protocol specification* (disponible uniquement en anglais)

¹ Dans les pages suivantes de la présente norme, "la présente partie" se substitue à "cette partie de la série CEI 61784-3".

IEC 61158-5-2, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition* (disponible uniquement en anglais)

IEC 61158-6-2, *Industrial communication networks – Fieldbus specifications – Part 6-2: Application layer protocol specification* (disponible uniquement en anglais)

CEI 61326-3-1, *Matériel électrique de mesure, de commande et de laboratoire – Exigences relatives à la CEM – Partie 3-1: Exigences d'immunité pour les systèmes relatifs à la sécurité et pour les matériels destinés à réaliser des fonctions relatives à la sécurité (sécurité fonctionnelle) – Applications industrielles générales*²

CEI 61326-3-2, *Matériel électrique de mesure, de commande et de laboratoire – Exigences relatives à la CEM – Partie 3-2: Exigences d'immunité pour les systèmes relatifs à la sécurité et pour les matériels destinés à réaliser des fonctions relatives à la sécurité (sécurité fonctionnelle) – Applications industrielles dont l'environnement électromagnétique est spécifié*²

CEI 61508 (toutes les parties), *Sécurité fonctionnelle des systèmes électriques / électroniques / électroniques programmables relatifs à la sécurité*

CEI 61784-1, *Réseaux de communications industriels – Profils – Partie 1: Profils de bus de terrain*

CEI 61784-2, *Réseaux de communication industriels – Profils – Partie 2: Profils de bus de terrain supplémentaires pour les réseaux en temps réel basés sur l'ISO/CEI 8802-3*

CEI 61784-3, *Réseaux de communication industriels – Profils – Partie 3: Bus de terrain de sécurité fonctionnelle – Règles générales et définitions de profil*

IEC 61784-5-2, *Industrial communication networks – Profiles – Part 5: Installation of fieldbuses – Installation profiles for CPF 2* (disponible uniquement en anglais)

IEC 61918, *Industrial communication networks – Installation of communication networks in industrial premises* (disponible uniquement en anglais)

CEI 62026-3, *Appareillage à basse tension – Interfaces appareil de commande-appareil (CDI) – Partie 3: DeviceNet*

ISO 15745-2, *Systèmes d'automatisation industrielle et intégration – Cadres d'intégration d'application pour les systèmes ouverts – Partie 2: Description de référence pour les systèmes de contrôle fondés sur l'ISO 11898*

ISO 15745-3, *Systèmes d'automatisation industrielle et intégration – Cadres d'intégration d'application pour systèmes ouverts – Partie 3: Description de référence pour les systèmes de contrôle fondés sur la CEI 61158*

ISO 15745-4, *Systèmes d'automatisation industrielle et intégration – Cadres d'intégration d'application pour les systèmes ouverts – Partie 4: Description de référence pour les systèmes de contrôle fondés sur Ethernet*

² A publier.

3 Termes, définitions, symboles, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

3.1.1 Termes et définitions communs

3.1.1.1

disponibilité

probabilité, pour un système automatisé, qu'il ne se produise pas de conditions opérationnelles non satisfaisantes, telles que la perte de production, pendant une période donnée

NOTE La disponibilité dépend de la MTBF (moyenne des temps de bon fonctionnement) et du TMI (temps moyen d'indisponibilité)

Disponibilité = $MTBF / (MTBF + TMI)$.

3.1.1.2

canal noir

canal de communication sans preuve existante de conception ou de validation conformément à la série CEI 61508

3.1.1.3

pont

dispositif abstrait qui relie des segments de réseau multiples le long de la couche de liaison de données

3.1.1.4

canal de communication

connexion logique entre deux points limites d'un *système de communication*

3.1.1.5

système de communication

disposition de matériels, logiciels et vecteurs de propagation destinée à permettre la transmission de *messages* (ISO/CEI 7498, couche d'application) d'une application à une autre

3.1.1.6

connexion

liaison logique entre deux objets d'application de dispositifs identiques ou différents

3.1.1.7

contrôle de redondance cyclique (CRC)

<valeur> donnée redondante déduite, et enregistrée ou transmise simultanément, d'un bloc de données afin de détecter toute corruption des données

<méthode> procédure utilisée pour calculer les données redondantes

NOTE 1 Les termes « code CRC » et « signature CRC », et les étiquettes telles que CRC1, CRC2, peuvent également être utilisés dans la présente norme pour se référer aux données redondantes.

NOTE 2 Voir également [28], [29]³.

3.1.1.8

erreur

écart ou discordance entre une valeur ou une condition calculée, observée ou mesurée, et la valeur ou la condition vraie, prescrite ou théoriquement correcte

³ Les chiffres entre crochets se réfèrent à la bibliographie.

NOTE 1 Une erreur peut être causée par une entité en panne, par exemple une erreur de calcul faite par un ordinateur en panne.

[VEI 191-05-24], [CEI 61508-4:1998], [CEI 61158]

NOTE 2 Les erreurs peuvent être causées par des erreurs de conception du matériel/logiciel et/ou des informations altérées du fait de perturbations électromagnétiques et/ou autres effets.

NOTE 3 Les erreurs ne produisent nécessairement pas une *défaillance* ou une *panne*.

3.1.1.9

défaillance

cessation de l'aptitude d'une unité fonctionnelle à accomplir une fonction requise

NOTE 1 La définition du VEI 191-04-01 est identique avec des notes complémentaires.

[CEI 61508-4:1998], [ISO/CEI 2382-14.01.11]

NOTE 2 Une défaillance peut être causée par une *erreur* (par exemple, problème de conception matérielle/logicielle ou rupture de message).

3.1.1.10

panne

condition anormale susceptible de provoquer la réduction, ou la perte, de capacité d'une unité fonctionnelle à accomplir une fonction requise

NOTE Le VEI 191-05-01 définit la « panne » comme un état caractérisé par l'incapacité à accomplir une fonction requise, à l'exclusion de l'incapacité au cours de la période de maintenance préventive ou autres actions planifiées, ou du fait de l'absence de ressources externes.

[CEI 61508-4:1998], [ISO/CEI 2382-14.01.10]

3.1.1.11

bus de terrain

système de communication basé sur le transfert de données en série et utilisé dans des applications d'automatisation industrielle ou de commande de processus

3.1.1.12

trame

synonyme discrédité de DLPDU

3.1.1.13

fonction de hachage

fonction (mathématique) de mise en correspondance des valeurs d'un ensemble (éventuellement) très grand de valeurs en une plage de valeurs (habituellement) plus petite

NOTE 1 Les fonctions de hachage peuvent être utilisées pour déterminer l'altération des données.

NOTE 2 Les fonctions de hachage courantes incluent la parité, la somme de contrôle ou le CRC.

[CEI 62210, modifiée]

3.1.1.14

danger

état ou ensemble de conditions d'un système qui, avec d'autres conditions associées, entraîne inévitablement un préjudice pour les personnes, les biens ou l'environnement

3.1.1.15

maître

entité de communication active capable d'initier et de programmer des activités de communication effectuées par d'autres stations qui peuvent être des maîtres ou des esclaves

3.1.1.16**message**

série ordonnée d'octets destinée à communiquer des informations
[ISO/CEI 2382-16.02.01, modifiée]

3.1.1.17**déclenchement de nuisance**

déclenchement parasite sans effet préjudiciable

NOTE Les erreurs anormales internes peuvent être générées dans des systèmes de communication tels que des systèmes de transmission par ondes radioélectriques, par exemple, du fait d'un trop grand nombre de nouvelles tentatives en présence de perturbations.

3.1.1.18**redondance**

existence de moyens, outre les moyens qui se révéleraient suffisants pour qu'une unité fonctionnelle accomplisse une fonction requise ou que des données représentent une information

EXEMPLE La duplication de composants fonctionnels et l'ajout de bits de parité sont tous deux des instances de redondance.

NOTE 1 La redondance sert essentiellement à améliorer la fiabilité ou la disponibilité.

NOTE 2 La définition du VEI 191-15-01 est moins complète.

[CEI 61508-4:1998], [ISO/CEI 2382-14.01.12]

3.1.1.19**fiabilité**

probabilité qu'un système automatisé puisse accomplir une fonction requise, dans des conditions données, pendant un intervalle de temps donné (t_1 , t_2)

NOTE 1 On suppose en général que le système automatisé est en état d'accomplir la fonction requise au début de l'intervalle de temps donné.

NOTE 2 Le terme "fiabilité" est aussi employé pour désigner l'aptitude caractérisée par cette probabilité.

NOTE 3 Au cours de la période MTBF ou MTTF, la probabilité qu'un système automatisé exécute une fonction requise dans les conditions données décroît.

NOTE 4 La fiabilité est différente de la disponibilité.

[CEI 62059-11, modifiée]

3.1.1.20**risque**

combinaison de la probabilité d'occurrence d'un dommage ou préjudice et de la gravité de ce dernier

[CEI 61508-4:1998]

3.1.1.21**couche de communication de sécurité (SCL)**

couche de communication qui comprend toutes les mesures nécessaires permettant d'assurer la transmission de données en toute sécurité conformément aux exigences de la CEI 61508

3.1.1.22**connexion de sécurité**

connexion qui utilise le protocole de sécurité pour des transactions de communications

3.1.1.23**données de sécurité**

données transmises par un réseau de sécurité utilisant un protocole de sécurité

NOTE La couche de communication de sécurité ne garantit pas la sécurité des données proprement dites, mais uniquement la transmission en toute sécurité de ces dernières.

3.1.1.24

dispositif de sécurité

dispositif conçu conformément à la CEI 61508 et qui met en œuvre le profil de communication de sécurité fonctionnelle

3.1.1.25

fonction de sécurité

fonction qu'un système E/E/PE relatif à la sécurité, un autre *système relatif à la sécurité* ou d'autres mesures externes de réduction du risque sont tenus de mettre en œuvre, et qui est destinée à atteindre ou à maintenir un état de sécurité de l'équipement commandé, compte tenu d'un événement dangereux spécifique

[CEI 61508-4:1998]

3.1.1.26

temps de réponse de la fonction de sécurité

temps écoulé du cas le plus défavorable suite à l'activation d'un capteur de sécurité relié à un bus de terrain, avant que ne soit atteint l'état de sécurité correspondant de son (ses) actionneur(s) de sécurité, du fait d'erreurs ou de défaillances avérées dans le canal de fonction de sécurité

NOTE Ce concept, introduit dans la CEI 61784-3, 5.2.4, est traité par les profils de communication de sécurité fonctionnelle définis dans la présente partie.

3.1.1.27

niveau d'intégrité de sécurité (SIL)

niveau discret (un sur quatre niveaux possibles) permettant de spécifier les exigences d'intégrité de sécurité des *fonctions de sécurité* à affecter aux systèmes E/E/PE relatifs à la sécurité, où le niveau d'intégrité de sécurité 4 est le niveau le plus élevé et le niveau d'intégrité de sécurité 1 est le niveau le plus faible

NOTE Les mesures de défaillance cible applicables aux quatre niveaux d'intégrité de sécurité sont spécifiées dans les Tableaux 2 et 3 de la CEI 61508-1.

[CEI 61508-4:1998]

3.1.1.28

mesure de sécurité

<la présente norme> mesure permettant de contrôler les *erreurs* de communication éventuelles, qui est conçue et mise en œuvre conformément aux exigences de la CEI 61508

NOTE 1 Dans la pratique, plusieurs mesures de sécurité sont combinées pour atteindre le niveau d'intégrité de sécurité requis.

NOTE 2 Les *erreurs* de communication et les mesures de sécurité associées sont détaillées dans la CEI 61784-3, 5.3 et 5.4.

3.1.1.29

application relative à la sécurité

programmes conçus conformément à la CEI 61508 pour satisfaire aux exigences SIL de l'application

3.1.1.30

système relatif à la sécurité

système qui exécute les *fonctions de sécurité* conformément à la CEI 61508

3.1.1.31

esclave

entité de communication passive capable de recevoir des messages et de les envoyer en réponse à une autre entité de communication qui peut être maître ou esclave

3.1.1.32**déclenchement parasite**

déclenchement provoqué par le système de sécurité sans injonction du processus

3.1.1.33**datation (horodatage)**

information temporelle incluse dans un *message*

3.1.2 CPF 2: Termes et définitions supplémentaires**3.1.2.1****demande ping (demande d'acquittement)**

demande d'acquittement de la réception d'un message par le consommateur

3.1.2.2**réponse à une demande ping**

réponse à une demande ping

3.1.2.3**signature**

méthode d'identification unique d'un ensemble de données

NOTE Ceci peut prendre la forme d'une révision, d'une somme de contrôle ou d'une toute autre méthode convenue entre les dispositifs d'origine et cibles.

3.2 Symboles et abréviations**3.2.1 Symboles et abréviations communs**

CP	Profil de communication (<i>Communication Profile</i>)	[CEI 61784-1]
CPF	Famille de profils de communication (<i>Communication Profile Family</i>)	[CEI 61784-1]
CRC	Contrôle de redondance cyclique (<i>Cyclic Redundancy Check</i>)	
DLL	Couche de liaison de données (<i>Data Link Layer</i>)	[ISO/CEI 7498-1]
DLPDU	Ensemble (Unité) de données de protocole de liaison de données (<i>Data Link Protocol Data Unit</i>)	
EMI	Perturbation électromagnétique (<i>Electro-Magnetic Interference</i>)	
EUC	Équipement commandé (<i>Equipment Under Control</i>)	[CEI 61508-4:1998]
FAL	Couche Application de bus de terrain (<i>Fieldbus Application Layer</i>)	[CEI 61158-5]
FSCP	Profil de communication de sécurité fonctionnelle (<i>Functional Safety Communication Profile</i>)	
E/E/PE	Électrique/électronique/électronique programmable (<i>Electrical/Electronic/Programmable Electronic</i>)	[CEI 61508-4:1998]
PDU	Ensemble (unité) de données de protocole (<i>Protocol Data Unit</i>)	[ISO/CEI 7498-1]
PFD	Probabilité moyenne de défaillance sur sollicitation (<i>Average Probability of Failure on Demand</i>)	[CEI 61508-6:1998]
PFH	Fréquence moyenne de défaillance dangereuse par heure (<i>Probability of Failure per Hour</i>)	[CEI 61508-6:1998]
PhL	Couche physique (<i>Physical Layer</i>)	[ISO/CEI 7498-1]
PLC	Automate programmable (<i>Programmable Logic Controller</i>)	
SCL	Couche de communication de sécurité (<i>Safety Communication Layer</i>)	
SIL	Niveau d'intégrité de sécurité (<i>Safety Integrity Level</i>)	[CEI 61508-4:1998]

3.2.2 CPF 2: Symboles et abréviations supplémentaires

AIRS Déjà en Mode/État sollicité (*Already In Requested Mode/State*)

BER	Taux d'erreurs sur les bits (<i>Bit Error Rate</i>)
CCO	Objet de configuration de connexion (<i>Connection Configuration Object</i>)
CFUNID	UNID de configuration (<i>Configuration UNID</i>)
CNP	Point de connexion (<i>Connection Point</i>)
CID	ID consommateur (<i>Consumer ID</i>)
CIP™	Protocole industriel commun (<i>Common Industrial Protocol</i>) (cadre d'application partagé parmi les profils de communication CPF-2)
CPCRC	CRC de paramètre de connexion (<i>Connection Parameter CRC</i>)
CTM	Multiplicateur de temporisation de connexion (<i>Connection Timeout Multiplier</i>)
EDS	Fiche technique électronique (<i>Electronic Data Sheet</i>)
EPI	Intervalle de paquets attendu (<i>Expected Packet Interval</i>)
DEL/LED	Diode électroluminescente (<i>Light Emitting Diode</i>)
MACID	Adresse physique de dispositif sur réseau (<i>Device physical address on network</i>)
MOD	Module
NV	Non volatil (<i>Non-Volatile</i>)
OCPUNID	Propriétaire de connexion de sortie UNID (<i>Output Connection Owner UNID</i>)
OSC	Conflit d'état d'objet (<i>Object State Conflict</i>)
O-à-T	Point d'origine vers Cible (<i>Originator to Target</i>)
OUNID	Point d'origine UNID (<i>Originator UNID</i>)
PID	Identifiant de producteur (<i>Producer Identifier</i>)
PS	Alimentation (<i>Power Supply</i>)
Res.	Réservé (<i>Reserved</i>)
RPI	Intervalle de paquets sollicités (<i>Requested Packet Interval</i>)
SCCRC	CRC de données de configuration de sécurité (<i>Safety Configuration Data CRC</i>)
SCID	Configuration de sécurité identifiée (<i>Safety Configuration Identified</i>)
SCTS	Datation de configuration de sécurité (<i>Safety Configuration Time Stamp</i>)
SNCT	Outil de configuration de réseau de sécurité (<i>Safety Network Configuration Tool</i>)
SNN	Numéro de réseau de sécurité (<i>Safety Network Number</i>)
T-à-O	Cible vers Point d'origine (<i>Target to Originator</i>)
TBD	A définir (<i>To Be Defined</i>)
TNS	TUNID non défini (<i>TUNID Not Set</i>)
TUNID	UNID Cible (<i>Target UNID</i>)
UNID	Identifiant de réseau unique (<i>Unique Network Identifier</i>)
UML	Langage de modélisation unifié (<i>Universal Modeling Language</i>)

3.3 Conventions

Les conventions utilisées pour les descriptions des objets, services et protocoles sont décrites dans la CEI 61158-3-2, dans la CEI 61158-4-2, dans la CEI 61158-5-2 et dans la CEI 61158-6-2.

Le cas échéant, la présente partie utilise des organigrammes, des diagrammes séquentiels UML et un pseudo-code C pour la description des concepts.

4 Présentation générale du FSCP 2/1 (CIP Safety™)

4.1 Généralités

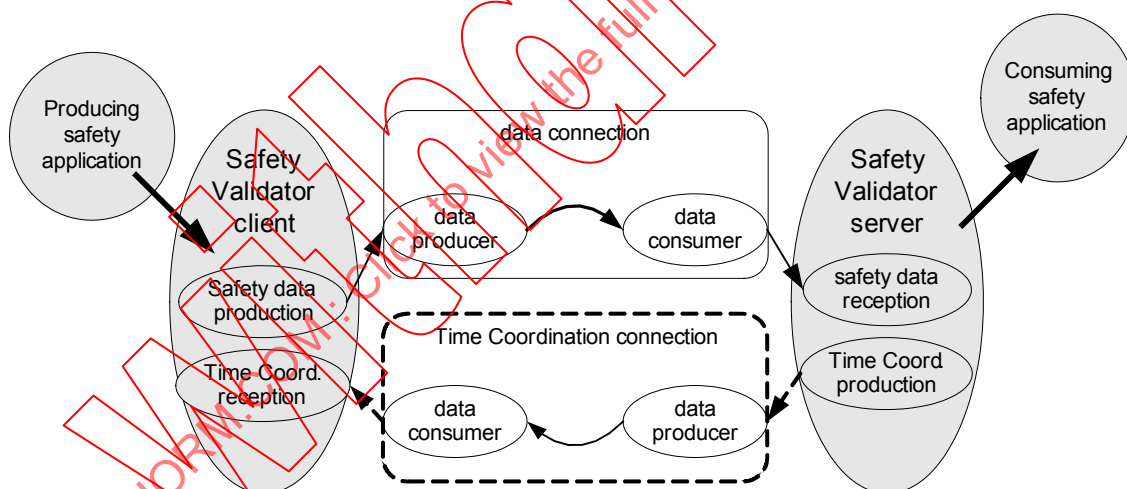
La famille de profils de communication 2 (communément appelée CIP™⁴) définit des profils de communication sur la base du type 2 de la CEI 61158-2, de la CEI 61158-3-2, de la CEI 61158-4-2, de la CEI 61158-5-2 et de la CEI 61158-6-2.

Les profils de base CP 2/1, CP 2/2 et CP 2/3 sont définis dans la CEI 61784-1 et la CEI 61784-2. Le profil de communication de sécurité fonctionnelle CPF 2 FSCP 2/1 (CIP Safety™⁴) est fondé sur les profils de base CPF 2 définis dans la CEI 61784-1 et la CEI 61784-2, ainsi que les spécifications de la couche de communication de sécurité définies dans la présente partie.

4.2 FSCP 2/1

Le FSCP 2/1 est basé sur le modèle producteur/consommateur de CPF 2. L'association des producteurs et des consommateurs constitue un élément important de la relation qui garantit la haute intégrité nécessaire aux applications relatives à la sécurité.

La couche de communication de sécurité FSCP 2/1 est spécifiée à l'aide d'un Objet de validation de sécurité. Cet objet est chargé de la gestion des connexions de sécurité FSCP 2/1 et constitue l'interface entre les objets d'application relatifs à la sécurité et les connexions de couches de liaison, tel qu'illustré à la Figure 3. L'objet de validation de sécurité garantit l'intégrité des transferts de données de sécurité.



Légende

Anglais	Français
Producing safety application	Production d'application de sécurité
Safety Validator client	Objet de validation de sécurité client
Safety data production	Production des données de sécurité
Time Coordination reception	Réception du signal de coordination temporelle
data connection	connexion de données

⁴ CIP™ (Common Industrial Protocol) et CIP Safety™ désignent les appellations commerciales de l'organisme à but non lucratif Open DeviceNet Vendor Association, Inc.(ODVA). Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que la CEI approuve ou recommande le détenteur de la marque ou de l'un quelconque de ses produits. La conformité à la présente norme n'exige pas l'emploi des appellations CIP™ ou CIP Safety™. L'emploi des appellations CIP™ ou CIP Safety™ exige l'autorisation de ODVA.

Anglais	Français
data producer	producteur de données
data consumer	consommateur des données
Time Coordination production	Production du signal de coordination temporelle
Safety Validator server	Objet de validation de sécurité serveur
Safety data reception	Réception des données de sécurité
Time Coordination production	Production de la coordination temporelle
Consuming safety application	Application de sécurité consommatrice

Figure 3 – Relation des objets de validation de sécurité

L'intégrité des transferts de données de sécurité est assurée comme suit:

- l'application relative à la sécurité productrice utilise une instance d'objet de validation de sécurité client pour générer des données de sécurité et assurer la coordination temporelle;
- le client utilise un producteur de données de liaison pour transmettre les données et un consommateur de liaison pour recevoir des messages de coordination temporelle;
- l'application relative à la sécurité consommatrice utilise un objet de validation de sécurité serveur pour recevoir et vérifier les données;
- le serveur utilise un consommateur de liaison pour recevoir les données et un producteur de liaison pour transmettre des messages de coordination temporelle.

Le FSCP 2/1 utilise le concept de canal noir. Les producteurs et les consommateurs de données de liaison n'ont aucune connaissance sur le paquet de données de sécurité et ne mettent en œuvre aucune fonction de sécurité. Les objets de validation de sécurité sont responsables du transfert à haute intégrité et du contrôle des données de sécurité.

Le FSCP 2/1 applique les mesures suivantes pour assurer l'intégrité des messages de sécurité:

- datation;
- authentification de connexion;
- assurance d'intégrité des données;
- redondance avec contre-vérification;
- différents systèmes d'assurance d'intégrité des données.

Les messages sont produits avec une datation qui permet au consommateur de vérifier l'âge des données transmises. L'identification fait l'objet d'un codage dans chaque message relatif à la sécurité afin de s'assurer que le bon consommateur utilise le message. Tous les messages relatifs à la sécurité utilisent un CRC unique. La transmission des données relatives à la sécurité est redondante. Diverses mesures de production de messages relatifs à la sécurité permettent de s'assurer que les messages CPF 2 standards ne sont pas interprétés comme des messages de sécurité.

5 Généralités

5.1 Documents externes de spécifications applicables au profil

Le document suivant est particulièrement utile pour bien comprendre la conception des protocoles FSCP 2/1:

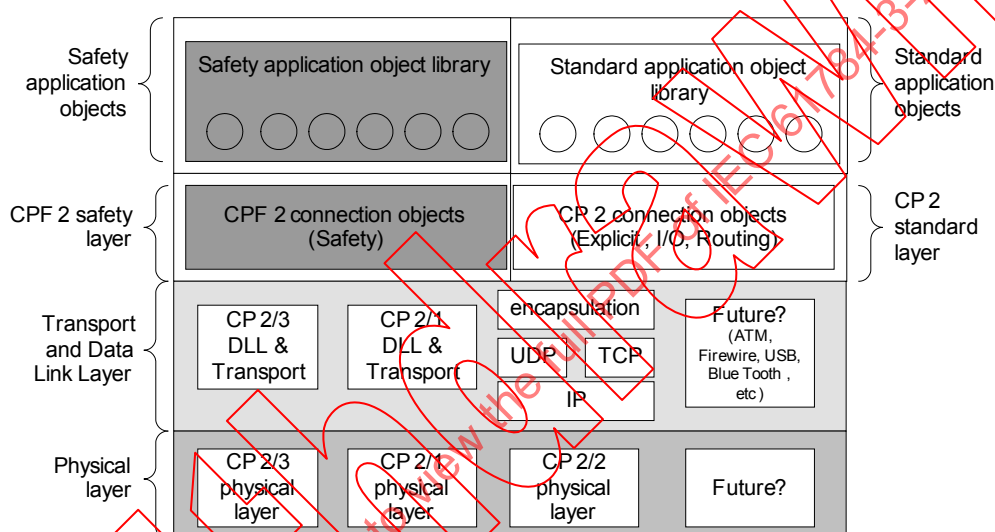
- GS-ET-26 [27]

- ^e Non revendiqué pour le profil FSCO 2/1.
^f Il est vraisemblable que la mascarade ne comporte pas de données redondantes et qu'il ne satisfera pas à la contre-vérification.

5.4 Structure de la couche de communication de sécurité

Le protocole de sécurité est structuré en couches au haut du protocole de réseau standard (la pile de protocoles standard est un canal noir). La Figure 4 illustre la relation entre la couche de communication de sécurité et les couches CPF 2 existantes.

Les couches de sécurité acceptent les données de sécurité provenant du circuit d'application relatif à la sécurité. Les couches de sécurité élaborent ensuite un message de sécurité, transmettent ce message, et reçoivent et décodent ce même message. Une fois le message de sécurité décodé et vérifié, les données sont présentées au circuit d'application relatif à la sécurité récepteur.



Légende

Anglais	Français
Safety application objects	Objets d'application de sécurité
Safety application object library	Bibliothèque d'objets d'application de sécurité
Standard application objects	Objets d'application normalisés
CPF2 safety layer	Couche de sécurité CPF2
CPF 2 connection objects (safety)	Objets de connexion CPF 2 (sécurité)
CP2 connection objects (explicit, I/O, routing)	Objets de connexion CP2 (explicite, E/S, acheminement)
CP2 standard layer	Couche normalisée CP2
Transport and data link layer	Couche de liaison de données et de transport
Encapsulation	Encapsulation
Future?(ATM, Firewire, USB, Blue Tooth, etc.)	Futur ? (ATM, Firewire, USB, Blue Tooth, etc.)
Physical layer	Couche physique

Figure 4 – Couches de communication

5.5 Relations avec la FAL (et DLL, PhL)

5.5.1 Généralités

Cette couche de communication de sécurité doit être utilisée uniquement avec les profils de communication CPF 2 (CP 2/3 (DeviceNet™⁵), CP 2/2 (EtherNet/IP™⁶)). Il n'existe aucune exigence concernant les profils de communication CPF 2 autres que celles définies dans la présente partie.

5.5.2 Types de données

Les profils définis dans la présente partie prennent en charge tous les types de données CPF 2 définis dans la CEI 61158-5 pour le Type 2.

6 Services de la couche de communication de sécurité

6.1 Introduction

Le présent paragraphe définit les extensions des objets CPF 2 existants, ainsi que les nouveaux objets de sécurité. Les connexions de sécurité utilisent les connexions de classe 0 pour les messages E/S. Les connexions de messages explicites standards sont utilisées pour l'établissement des connexions E/S de sécurité.

6.2 Objet de connexion

6.2.1 Généralités

Plusieurs attributs et services de classes spécifiques à la sécurité sont définis pour prendre en charge la sécurité. Les nœuds de sécurité contenus dans le protocole FSCP 2/1 doivent mettre en œuvre les services SafetyOpen et SafetyClose de l'objet Connexion.

6.2.2 Extensions des attributs des classes

Le Tableau 2 définit les attributs de classe utilisés par les dispositifs de sécurité.

Tableau 2 – Nouveaux attributs de classe

ID d'attribut	Mise en œuvre nécessaire	Règles d'accès	NV	Nom	Type de données	Description de l'attribut
8	Facultatif	Obtenir	V	Comptage d'erreurs de demandes de connexion	UINT	Compteur de diagnostic qui représente le cumul des pannes SafetyOpen ou SafetyClose ^a
9	Facultatif	Obtenir	V	Compteur des connexions de sécurité	STRUCT de:	Compteurs d'états

⁵ DeviceNet™ désigne l'appellation commerciale de l'organisme à but non lucratif Open DeviceNet Vendor Association, Inc (ODVA). Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que la CEI approuve ou recommande le détenteur de la marque ou de l'un quelconque de ses produits. La conformité à la présente norme n'exige pas l'emploi de l'appellation commerciale DeviceNet™. L'emploi de l'appellation commerciale DeviceNet™ exige l'autorisation de ODVA.

⁶ EtherNet/IP™ désigne l'appellation commerciale de ControlNet International, Ltd. (CI) et de Open DeviceNet Vendor Association, Inc. (ODVA). Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que la CEI approuve ou recommande le détenteur de la marque ou de l'un quelconque de ses produits. La conformité à la présente norme n'exige pas l'emploi de l'appellation commerciale EtherNet/IP™. L'emploi de l'appellation commerciale EtherNet/IP™ exige l'autorisation de CI ou de ODVA.

ID d'attribut	Mise en œuvre nécessaire	Règles d'accès	NV	Nom	Type de données	Description de l'attribut
				Comptage de demandes SafetyOpen abouties	UINT	Comptage de demandes SafetyOpen abouties émises de ce nœud vers une autre cible
				Comptage de demandes SafetyOpen non abouties	UINT	Comptage de demandes SafetyOpen non abouties émises de ce nœud vers une autre cible
				Comptage de demandes SafetyClose abouties	UINT	Comptage de demandes SafetyClose abouties émises de ce nœud vers une autre cible
				Comptage de demandes SafetyClose non abouties ^b	UINT	Comptage de demandes SafetyClose non abouties émises de ce nœud vers une autre cible
				Comptage des indications SafetyOpen abouties	UINT	Comptage de demandes SafetyOpen abouties émises d'un point d'origine externe vers ce nœud
				Comptage des indications SafetyOpen non abouties ^a	UINT	Comptage de demandes SafetyOpen non abouties émises d'un point d'origine externe vers ce nœud
				Comptage des indications SafetyClose abouties	UINT	Comptage de demandes SafetyClose abouties émises d'un point d'origine externe vers ce nœud
				Comptage des indications SafetyClose non abouties ^b	UINT	Comptage de demandes SafetyClose non abouties émises d'un point d'origine externe vers ce nœud

^a L'attribut 8 est la somme des 4 compteurs de défaillance dans l'attribut 9. La représentation des erreurs de connexion par cet attribut unique peut être utilisée dans les logiciels de surveillance afin de contrôler de manière efficace les erreurs sans interroger les structures supplémentaires de l'attribut 9. Une fois que l'attribut 8 a été défini comme non nul, le logiciel de surveillance peut examiner l'attribut 9 pour plus de détails.

^b Inclus dans l'attribut de classe 8.

6.2.3 Extensions de services

Le Tableau 3 définit les services pouvant être utilisés par les dispositifs de sécurité sur le FSCP 2/1. Ces services doivent être pris en charge par un dispositif cible sur ce même profil. Les services SafetyOpen et SafetyClose, utilisés en lieu et place des services ForwardOpen et ForwardClose de l'objet Gestionnaire de connexion héritent du même comportement que ces services de Gestionnaire de connexion.

Tableau 3 – Extensions de services

Code de service	Mise en œuvre nécessaire		Nom du service	Description du service
	Classe	Instance		
4D _{hex}	Facultatif	N/A	Compteurs de réinitialisation	Ce service remet à zéro les attributs de classes 8 et 9
4E _{hex}	Requis	N/A	SafetyClose	Utilisé pour fermer une connexion E/S de sécurité. Voir le service ForwardClose de l'objet Gestionnaire de connexion en 6.3.13
54 _{hex}	Requis	N/A	SafetyOpen ^a	Utilisé pour établir une connexion E/S de sécurité. Voir le service ForwardOpen de l'objet Gestionnaire de connexion en 6.3.2
^a Les données du service SafetyOpen doivent contenir un segment de réseau de sécurité dans le champ Connection_Path.				

6.2.4 Format de réponse de messages explicites pour SafetyOpen et SafetyClose

Les réponses des services SafetyOpen et SafetyClose ont un format commun. Ce format contient des informations d'état pour des conditions de réussite et d'échec. Des données de réponse spécifiques à chaque service sont transmises si la réponse du service est positive. Si la réponse du service est une erreur, un état FSCP 2/1 satisfaisant indiqué par le code d'erreur général FF_{hex} est transmis, accompagné d'une quantité variable d'informations portant sur l'erreur étendue.

Le format de réponse pour les services spécifiques aux objets SafetyOpen et SafetyClose est défini dans le Tableau 4:

Tableau 4 – Format de réponse SafetyOpen et SafetyClose

Nom de paramètre	Type de données	Description du paramètre de réponse
Réservé	USINT	Réservé
État général	USINT	Un des codes d'état général énumérés dans la CEI 61158-6-2 (codes d'état)
Taille de l'état supplémentaire	USINT	Nombre de mots de 16 bits dans le tableau d'états supplémentaires
État supplémentaire	MATRICE de UINT	Etat supplémentaire
Données de réponse	MATRICE d'OCTETS	Données de réponse émises par la demande ou données d'erreurs supplémentaires si l'état général a indiqué une erreur

Les données de réponse et les états généraux/supplémentaires pour les services SafetyOpen et SafetyClose doivent correspondre aux données de réponse et états généraux/supplémentaires pour les services ForwardOpen et ForwardClose de l'objet Gestionnaire de connexion, respectivement.

6.3 Objet Gestionnaire de connexion

6.3.1 Généralités

Les dispositifs de sécurité doivent utiliser un service ForwardOpen avec un segment de réseau de sécurité pour établir des connexions de sécurité. Ce segment identifie de manière unique le service ForwardOpen comme un service ForwardOpen de sécurité. Dans l'ensemble du présent paragraphe, le terme "SafetyOpen" doit faire référence tant au service

ForwardOpen de l'objet Gestionnaire de connexion (lorsqu'il contient un segment de réseau de sécurité) qu'au service SafetyOpen de l'objet Connexion.

6.3.2 ForwardOpen pour la sécurité

Le contenu du service ForwardOpen de l'objet Gestionnaire de connexion, y compris le segment de réseau de sécurité, est présenté à la Figure 5. Les réglages appropriés des divers paramètres du service ForwardOpen sont indiqués ci-dessous.

Nom de paramètre		Type de donnée / Taille	Destination cible et sémantique
Priorité/Time_tick		OCTET	Demande de maintenance du gestionnaire de connexion
Time-out_ticks		USINT	Demande de maintenance du gestionnaire de connexion
ID de connexion de réseau O-à-T		UDINT	Instance de connexion de consommation:: ID de connexion consommée CIP
ID de connexion de réseau T-à-O		UDINT	Instance de connexion de production:: ID de connexion de production CIP
Numéro de série de connexion		UINT	Valeur: Numéro à 16 bits unique (pour le point d'origine). Destinataire: Demande de maintenance du gestionnaire de connexion
ID du fournisseur d'origine		UINT	Demande de maintenance du gestionnaire de connexion
Numéro de série du point d'origine		UDINT	Demande de maintenance du gestionnaire de connexion
Multiplicateur de temporisation de connexion		USINT	Instance de connexion de production et de consommation: Multiplicateur de temporisation
Réservé		3 OCTETS	
RPI O-à-T		UDINT	Instance de connexion de consommation: EPR
Paramètres de connexion de réseau O-à-T	Propriétaire redondant	WORD	Valeur: 0 (les connexions de sécurité ne prennent pas en charge la redondance); Destinataire: Gestionnaire de connexion
	Type de connexion		Valeur: Point à point; Destinataire: Gestionnaire de connexion
	Priorité		Valeur: Priorité élevée; Destinataire: Gestionnaire de connexion
	Fixe/Variable		Valeur: Fixe; Destinataire: Gestionnaire de connexion
	Taille de connexion (en octets)		Valeur: Taille du PDU de sécurité, y compris les sections Mode Octet, Actual (Réel), Complement (Complémentaire) et Time stamp (Datation); Destinataire: Instance de connexion de consommation: Attribut de taille de connexion consommée
RPI T-à-O		UDINT	Instance de connexion de production: EPR
Paramètres de connexion de réseau T-à-O	Propriétaire redondant	WORD	Valeur: 0 (les connexions de sécurité ne prennent pas en charge la redondance); Destinataire: Gestionnaire de connexion
	Type de connexion		Valeur: Point à point; Destinataire: Gestionnaire de connexion
	Priorité		Valeur: Priorité élevée; Destinataire: Gestionnaire de connexion
	Fixe/Variable		Valeur: Fixe; Destinataire: Gestionnaire de connexion

Nom de paramètre		Type de donnée / Taille	Destination cible et sémantique
	Taille de connexion (en octets)		Valeur: Taille du PDU de sécurité, y compris les sections Mode Octet, Actual (Réel), Complement (Complémentaire) et Time stamp (Datation). Destinataire: Instance de connexion de production; Taille de connexion produite
Type de transport/Déclencheur		SWORD	Valeur: Valeur de sécurité simple ou multipoint. Destinataire: Instance d'objet de validation. Déclencheur toujours réglé sur Application. Client/serveur: selon la direction
Connection_Path_Size		USINT	Nombre de mots de 16 bits dans le champ Connection_Path - le destinataire est la demande de maintenance du gestionnaire de connexion. NOTE La valeur incluse dans le calcul du CRC ne comprend pas les mots employés pour les segments Réseau/Port.
Connection_Path (EPATH complété)	Chemin		Variable
	Segment logique: Clé électronique	ID du fournisseur	UINT
		Type de prod.	UINT
		Code de prod.	UINT
		Compatible/ Rev. principale	SWORD
		Rev. secondaire	USINT
	Chemin d'application 1		Variable; 1 mot généralement
	Segment logique: Classe	Segment logique: Instance	Variable; 1 mot généralement
	Chemin d'application 2		Variable; 1 mot généralement
	Chemin d'application 3		Variable; 1 mot généralement
	Données de configuration		Variable
	Segment de données: Données simples		Données de configuration à utiliser pour l'objet d'application relatif à la sécurité identifié par le chemin de configuration
Segment de réseau: de Sécurité	Nom de segment de sécurité		50 _{hex}
	Taille de segment		USINT
	Format de segment		USINT
	Données de sécurité de réseau		MATRICE D' OCTETS

Légende

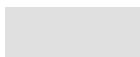
	= Champs Connexion/Objet de validation de sécurité couverts par le CRC des paramètres de connexion
	= Segment de données de configuration couvert par SCCV

Figure 5 – ForwardOpen avec segment de réseau de sécurité

6.3.3 Segment de réseau de sécurité

6.3.3.1 Format commun

Le type de segment de paramètres de connexion de sécurité a été ajouté à la définition du segment de réseau CPF 2 comme type de segment 10_{hex} (produisant une valeur de segment de 50_{hex}). La définition actualisée de Segment de réseau est donnée dans le Tableau 5.

Tableau 5 – Identifiant de segment de réseau de sécurité

Type de segment de réseau	Nom de segment de réseau
50_{hex}	Segment de sécurité

Le format du segment de réseau de sécurité est défini dans le Tableau 6. Ce type de segment se situe dans la plage des types de segment de réseau, qui comporte un champ de taille de données. Le paramètre de format de segment de sécurité, à savoir le premier paramètre dans la zone de données, indique les données qui suivent.

Tableau 6 – Définition du segment de réseau de sécurité

Décalage d'octet	Nom de champ	Type de données	Description
0	Type de segment	SWORD	Indique le segment de réseau de sécurité (50_{hex})
1	Longueur des données du segment de réseau	SWORD	Longueur des données de segment à suivre (en mots)
2	Format de segment de réseau de sécurité	USINT	Indique le type de segment de réseau de sécurité 0 = Format cible 1 = Format de routage 2 – 255 = Réserve
3 - n	Données du segment de réseau de sécurité	MATRICE D'OCTETS	Données de segment de réseau de sécurité, définies par le format de segment

6.3.3.2 Segment de réseau de sécurité: Format cible (00_{hex})

Le format cible est orienté vers la cible de la connexion. La valeur du paramètre Longueur des données de segment de réseau doit être 27 ($1b_{\text{hex}}$), ce qui correspond à 27 MOTS. La Figure 6 définit le format cible de réseau de sécurité.

Nom de champ	Type de données	Description
Réservé	SWORD	Remplissage
CRC de configuration (SCCRC)	UDINT	CRC-S4 du segment de données de configuration cible. Une valeur de 0 indique que le contrôle SCID peut être omis
Datation de configuration(SCTS)	DATE_AND_TIME	Datation de la configuration. Une valeur de 0 indique que le contrôle SCID peut être omis. Valeur de 6 octets (Date: 2 octets, Heure: 4 octets)
EPI de correction de temps	UDINT	Non utilisé: 0 Utilisé: Instance de connexion de correction de temps: EPI
Paramètres de connexion de réseau de correction de temps	WORD	Paramètres de connexion de réseau pour la connexion de correction de temps. Ce champ utilise la même définition que le paramètre de connexion de réseau contenu dans l'objet Gestionnaire de connexion. Non utilisé: 0 (Nul). Utilisé: Fixe, priorité élevée, multipoint
Target_UNID (TUNID)	STRUCT. de: 10 OCTETS	UNID du dispositif cible. Cette valeur est une valeur de 10 octets constituée de l'identifiant de réseau de sécurité (6 octets) et de l'adresse de nœud (adresse de 4 octets)
UNID du point d'origine (OUNID)	STRUCT. de: 10 OCTETS	UNID du dispositif d'origine. Cette valeur est une valeur de 10 octets constituée de l'identifiant de réseau de sécurité (6 octets) et de l'adresse de nœud (adresse de 4 octets)
Ping_Interval_EPI_Multiplier	UINT	Nombre qui définit le Ping_Count_Interval pour la connexion. La plage de valeurs valide est donnée en 7.6.5.10
Time_Coord_Msg_Min_Multiplier	UINT	Pour l'instance de connexion de production. Plage valide comprise entre 0 et 7 813
Network_Time_Expectation_Multiplier	UINT	Pour l'instance de connexion de consommation. Plage valide comprise entre 0 et 42 969
Timeout_Multiplier	USINT	Utilisé par le producteur et le consommateur pour les connexions de temporisation. Plage valide comprise entre 1 et 4
Max_Consumer_Number	USINT	Valeur: 1 (Point à point), 2-15(Multipoint)
CRC des paramètres de connexion (CPCRC)	UDINT	CRC-S4 des paramètres de connexion cible dans cette demande
ID de connexion de correction de temps	UDINT	Non utilisé: FFFFFFFF _{hex} Utilisé: Instance de connexion de correction de temps:: ID de connexion

Légende

 = Champs de segment de sécurité couverts par le CRC des paramètres de connexion

Figure 6 – Format cible de réseau de sécurité**6.3.3.3 Segment de réseau de sécurité: Format de routage (01_{hex})**

Le format de routage est orienté vers un routeur intermédiaire le long du chemin de connexion, lorsque ce routeur nécessite de savoir qu'il s'agit d'un service Safety ForwardOpen. Actuellement, seuls les routeurs CP 2/3 nécessitent que ce segment de réseau soit orienté vers eux. Le segment de réseau est appliqué au sous-réseau duquel la demande de connexion a été émise. Les segments de réseau de sécurité du format de routage

apparaissent uniquement dans la partie du chemin du Connection_Path d'un service Safety ForwardOpen. La longueur de segment totale est de 6 mots. Le format de routage de segment de réseau de sécurité est défini dans le Tableau 7.

Tableau 7 – Format de routage de segment de réseau de sécurité

Nom de champ	Type de données	Description
Réservé	SWORD	Réservé, valeur = 0
ID de connexion de correction de temps	UDINT	Non utilisé: FFFFFFFF _{hex} Utilisé: Instance de connexion de correction de temps:: ID de connexion
EPI de correction de temps	UDINT	Non utilisé: 0 Utilisé: Instance de connexion de correction de temps:: EPI
Paramètres de connexion de réseau de correction de temps	WORD	Paramètres de connexion de réseau pour la connexion de correction de temps. Ce champ utilise la même définition que le paramètre de connexion de réseau dans l'objet Gestionnaire de connexion. Non utilisé: 0 (Null). Utilisé: Fixe, priorité élevée, multipoint

6.3.4 Règles du point d'origine pour le calcul du CRC de paramètre de connexion

Le CRC de paramètre de connexion (CPCRC) est calculé uniquement lors de la première configuration d'une connexion ou lors de la modification de cette configuration. Dans les autres cas, cette valeur reste statique dans le point d'origine.

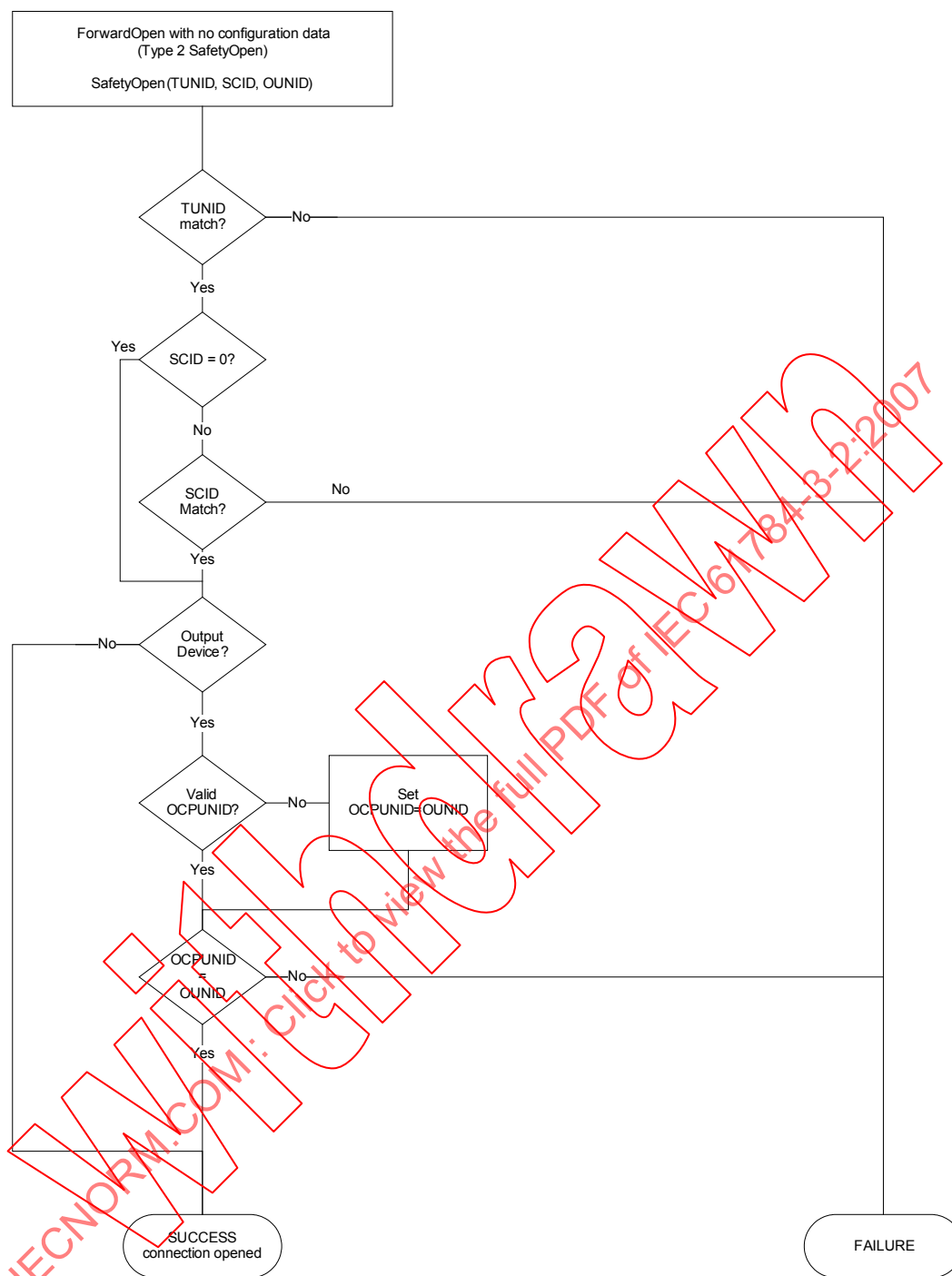
La taille du chemin de connexion dans le service SafetyOpen change de valeur selon qu'un point d'origine envoie ou non des données de configuration. Dans la mesure où ce champ est inclus dans le calcul du CPCRC, les points d'origine nécessitent d'être calculés et conservent deux valeurs lorsqu'ils prennent en charge la fonction de configuration SafetyOpen. Une valeur permet de configurer un dispositif tout en ouvrant la connexion, l'autre valeur permettant l'ouverture strictement dite de la connexion (aucune donnée de configuration n'est incluse).

La taille du chemin doit également être « pré-ajustée » afin de supprimer les entrées de chemin « dépouillées » par les routeurs au moment où le service Safety ForwardOpen transite par chaque sous-réseau. L'exigence qui s'applique impose que le point d'origine soit chargé de calculer la valeur CPCRC sur la base de l'état réel du service SafetyOpen ou ForwardOpen au moment où la cible le reçoit, de sorte que cette dernière puisse effectuer un calcul simple dans tous les cas.

6.3.5 Organigrammes de traitement de SafetyOpen

Les organigrammes suivants présentent le comportement requis d'une cible à réception d'un service SafetyOpen:

- Vérifier le CRC de paramètre de connexion. Comme l'illustre la Figure 6, le CRC de paramètre de connexion couvre les champs SafetyOpen utilisés par la cible. Les routeurs disposés le long du chemin ne doivent pas modifier ces champs;
- La cible vérifie l'adaptation du segment de clé électronique (voir 6.3.7);
- Selon le type de connexion de sécurité, la logique définie à la Figure 7 ou à la Figure 8 doit être observée.



Key

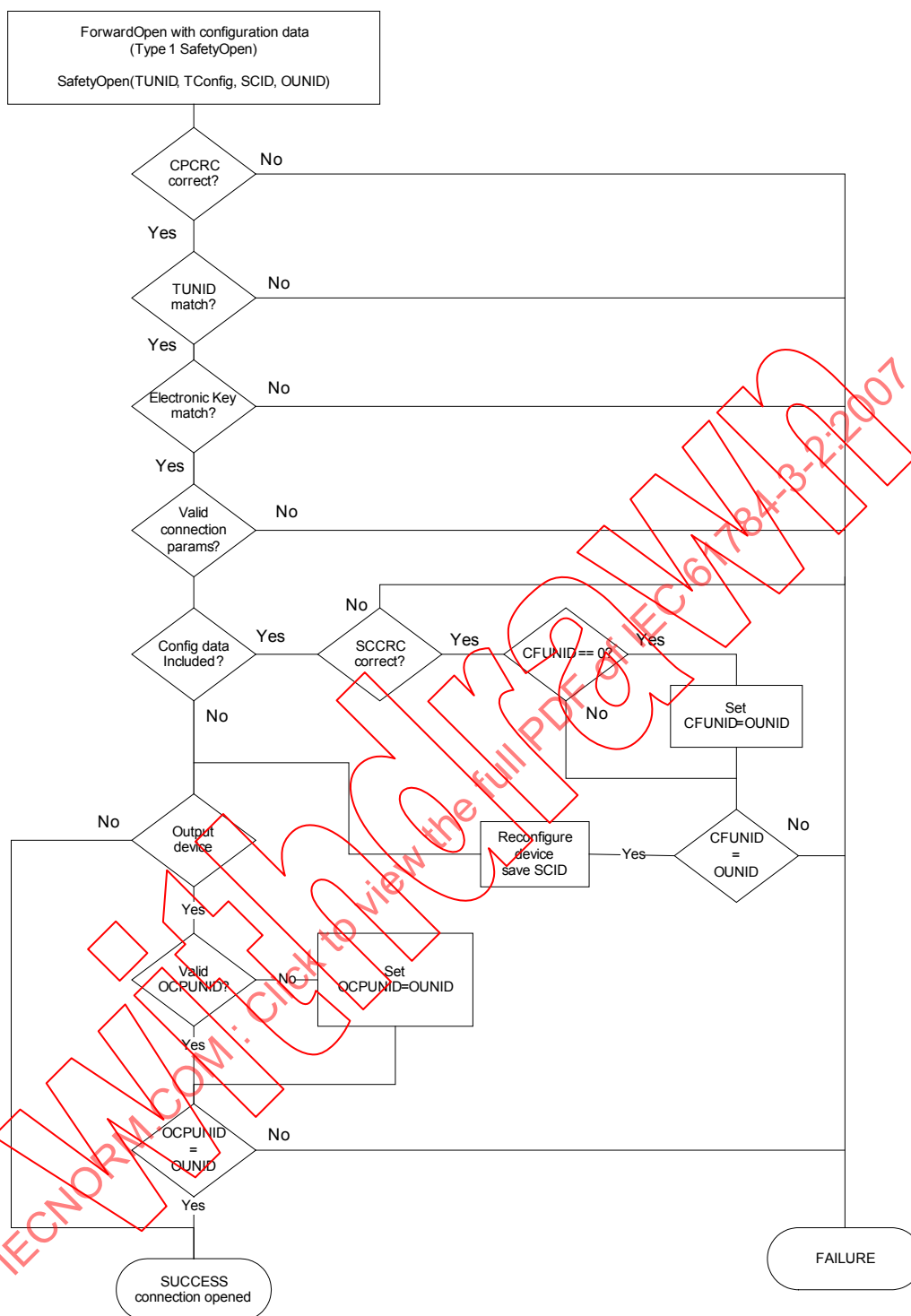
UNID	= Unique Network Identifier
TUNID	= Targets Unique Network Identifier
SCID	= Targets Safety Configuration ID
OUNID	= Originator's Unique Network Identifier
OCP	= Output Connection Point
OCPUNID	= Output Connection Owning UNID
CFUNID	= Configuration Owning UNID

Légende

Anglais	Français
ForwardOpen with no configuration data	ForwardOpen sans données de configuration
TUNID match?	TUNID adapté ?
No	Non

Anglais	Français
Yes	Oui
SCID match?	SCID adapté ?
Output device	Dispositif de sortie
Valid OCPUNID?	OCPUNID valide ?
Set	Régler
SUCCESS Connection Opened	RÉUSSITE Connexion Ouverte
FAILURE	ÉCHEC
Key	Légende
UNID = unique network identifier	identifiant de réseau unique
TUNID = target's unique network identifier	identifiant de réseau unique de la cible
SCID = target's safety configuration ID	ID de configuration de sécurité de la cible
OUNID = originator's unique network identifier	identifiant de réseau unique du point d'origine
OCP = output connection point	point de connexion de sortie
OCPUNID = output connection owning UNID	UNID de propriétaire de la connexion de sortie
CFUNID = configuration owning UNID	UNID de propriétaire de la configuration

Figure 7 – Service SafetyOpen de traitement de cible sans données de configuration (SafetyOpen de forme 2)



Key

UNID	= Unique Network Identifier
TUNID	= Target's Unique Network Identifier
CPCRC	= Connection Parameter CRC
SCID	= Target's Safety Configuration ID
SCCRC	= CRC portion of the SCID
OUNID	= Originator's Unique Network Identifier
OCPUNID	= Output Connection Owning UNID
CFUNID	= Configuration Owning UNID

Légende

Anglais	Français
ForwardOpen with configuration data	ForwardOpen avec données de configuration
CPCRC Correct?	CPCRC Correct ?

Anglais	Français
TUNID match?	TUNID adapté ?
No	Non
Yes	Oui
Electronic key match?	Clé électronique adaptée ?
Valid connection params?	Paramètres de connexion valides ?
Config data included?	Données de config. incluses ?
SCCRC correct?	SCCRC correct ?
Set	Régler
Output device	Dispositif de sortie
Reconfigure device save SCID	Reconfigurer SCID de sécurité de dispositif
Valid OCPUNID?	OCPUNID valide ?
SUCCESS connection opened	RÉUSSITE connexion ouverte
FAILURE	ÉCHEC
Key	Légende
UNID = unique network identifier	identifiant de réseau unique
TUNID = target's unique network identifier	identifiant de réseau unique de la cible
CPCRC = connection parameter CRC	CRC de paramètre de connexion
SCID = target's safety configuration ID	ID de configuration de sécurité de la cible
SCCRC = CRC portion of the SCID	partie CRC de SCID
OUNID = originator's unique network identifier	identifiant de réseau unique du point d'origine
OCPUNID = output connection owning UNID	UNID de propriétaire de la connexion de sortie
CFUNID = configuration owning UNID	UNID de propriétaire de la configuration

Figure 8 – Service SafetyOpen de traitement de cible avec données de configuration (SafetyOpen de forme 1)

6.3.6 Contrôles requis par les producteurs multipoint avec les connexions existantes

Il est visiblement possible que les paramètres de sécurité utilisés dans un service ForwardOpen ou SafetyOpen de divers points d'origine de consommation ne soient pas toujours identiques. Un ensemble cohérent de règles doit permettre de déterminer la méthode d'évaluation de ces paramètres par une cible de production multipoint, lorsque lesdits paramètres sont en contradiction avec d'autres connexions de consommation déjà établies. Le Tableau 8 définit les règles d'évaluation des paramètres de production multipoint.

Tableau 8 – Règles d'évaluation des paramètres de production multipoint

Paramètre de sécurité	Règle d'évaluation de production si la connexion existe	Réponse d'erreur ^a (le cas échéant)	
		Générale	Étendue
Multiplicateur EPI Ping_Interval	Doit correspondre au P.I.E.M. existant	01 _{hex} 00 _{hex}	0801 _{hex} 0012 _{hex}
Multiplicateur minimal de messages de coordination temporelle	Accepte n'importe lequel (peuvent être tous différents)	N/A	
Multiplicateur de délai du réseau	Accepte n'importe lequel (peuvent être tous différents)	N/A	
Timeout_Multiplier	Accepte n'importe lequel (peuvent être tous différents)	N/A	
Max_Consumer_Number	Doit s'adapter au Max_Consumer_Number existant	00 _{hex} 01 _{hex}	0801 _{hex} 002A _{hex}
RPI T-à-O, RPI O-à-T et RPI de correction de temps	Doit s'adapter aux RPI de connexion existants	01 _{hex}	0111 _{hex}
^a Une autre connexion peut également être ouverte en lieu et place de la transmission d'une réponse d'erreur, au choix du développeur de dispositifs.			

6.3.7 Utilisation de la clé électronique de sécurité

Les points d'origine et cibles de sécurité doivent satisfaire aux exigences suivantes d'utilisation du segment Clé électronique dans un service SafetyOpen.

Le segment Clé électronique doit être présent dans toutes les demandes de service SafetyOpen.

La Clé électronique doit toujours être traitée par les dispositifs cibles avec intégrité (ne peut être confirmée de manière externe).

Le bit de compatibilité et le comportement de la Clé électronique doivent être pris en charge par les points d'origine et cibles de sécurité, tel que défini. Les caractères de remplacement (champs réglés sur 0) pour chaque champ interne de la clé électronique ne doivent pas être pris en charge.

6.3.8 RPI en fonction de API dans les connexions de sécurité

Les dispositifs de sécurité n'ont pas la capacité de négocier la valeur RPI. Les cibles de sécurité acceptent le RPI demandé dans la mesure où il peut être pris en charge, ou le rejettent dans son intégralité. La tolérance relative à une demande RPI acceptable doit être de $\pm 128 \mu\text{s}$.

Les intervalles API T-à-O, O-à-T et de correction de temps dans une réponse satisfaisante doivent toujours avoir la même valeur que le service SafetyOpen reçoit.

6.3.9 Élaboration du chemin d'application de sécurité

La CEI 61158-6-2 définit une méthode d'élaboration d'un chemin d'application dans un service SafetyOpen pour les 3 références suivantes: une méthode pour configuration, une méthode pour les données consommées et une méthode pour les données produites.

Les types de chemin possibles sont identifiés comme suit:

A. - [Classe] [Ensemble] [Instance] [Configuration-instance]

- B. - [Classe] [Ensemble] [Instance] [Producteur-instance]
- C. - [Classe] [Ensemble] [Instance] [Consommateur-instance]
- D. - [Classe] [Ensemble] [Instance] [Instance NULLE définie par le dispositif]
- E. - [CNP] [CNP consommateur]
- F. - [CNP] [CNP producteur]
- G. - [CNP] [CNP NUL défini par le dispositif]

Lorsque la classe de référence des 3 chemins est identique, leur format est souvent comprimé: (CP désigne le Point de connexion).

[Class] [Class #] [Instance] [# Instance] [CNP] [CP Consommateur] [CNP] [CNP Producteur]

Dans la mesure où la classe de référence des trois chemins doit être identique, une valeur instance NULLE doit être réservée dans cette classe (par les développeurs de dispositifs) afin d'utiliser le format comprimé de sécurité. La valeur NULLE est utilisée pour la partie du chemin de référence de la connexion de coordination temporelle et de la connexion de configuration (dans les points d'origine de non configuration). Lorsque la pratique ne permet pas l'utilisation d'une référence de classe commune, la forme longue doit être utilisée.

Les règles suivantes s'appliquent à l'élaboration du chemin SafetyOpen:

- tous les dispositifs doivent au minimum prendre en charge la forme longue;
- un chemin Nul ou un point de connexion Nul doit permettre de représenter la connexion de coordination temporelle;
- un chemin Nul ou un point de connexion Nul doit être utilisé pour le chemin de configuration lorsque aucune donnée de configuration ne peut être transmise sur la connexion;
- les points d'origine doivent toujours transmettre un chemin de configuration valide sur toutes les connexions qui peuvent accepter un service SafetyOpen de forme 1, et ce, même en l'absence de toute donnée de configuration (c'est-à-dire forme 2a).

Le chemin ou le point de connexion «Nul» utilisé doit relever du choix du module. La méthode qui permet d'effectuer ce choix dans un fichier EDS utilise le champ «Chemin de données» dans la section Gestionnaire de connexion. Les dispositifs configurés au moyen d'un applet de configuration doivent également fournir ce chemin.

Les concepteurs de dispositifs de sécurité peuvent choisir le format de chemin à spécifier dans leurs fichiers EDS, parmi les options suivantes. Les combinaisons de trois chemins possibles pour les services SafetyOpen des formes 1, 2a utilisés sur les connexions, qui acceptent les données de configuration, sont les suivantes:

- [A, D, B] pour les connexions d'entrée;
- format comprimé [A, G, F] pour les connexions d'entrée;
- [A, C, D] pour les connexions de sortie;
- format comprimé [A, E, G] pour les connexions de sortie.

Les combinaisons de trois chemins possibles pour les services SafetyOpen des formes 2a, 2b utilisés sur les connexions, qui n'acceptent pas les données de configuration, sont les suivantes:

- [D, D, B] pour les connexions d'entrée;
- format comprimé [D, G, F] pour les connexions d'entrée;
- [D, C, D] pour les connexions de sortie;
- format comprimé [D, E, G] pour les connexions de sortie.

6.3.10 Types de connexion de l'objet de validation de sécurité

La fonction Objet de validation de sécurité, selon le type de connexion de sécurité prévu dans le service SafetyOpen, exige toujours entre 2 et 3 ressources de connexion. Le service SafetyOpen comporte les informations de connexion qui permettent à une cible d'établir ces connexions de manière appropriée. Le présent paragraphe montre de quelle manière sont déterminés les types de connexion, ainsi que les éléments de remplissage nécessaires du service SafetyOpen.

Une hypothèse de base repose sur le fait que les connexions pour les messages de coordination et de correction de temps n'exigent aucune entrée de chemin, dans la mesure où elles comportent un chemin implicite d'accès à l'objet de validation de sécurité qui coordonne la connexion de sécurité. Les divers scénarios de connexion sont décrits dans l'objet de validation en 6.7.

La fonction Objet de validation de sécurité peut déterminer le type de ressources de connexion requis en observant le paramètre du service SafetyOpen. Les divers paramètres de service SafetyOpen, ainsi que les paramètres clés dont a besoin la fonction de sécurité pour déterminer les types de connexions sont définis dans le Tableau 9. Les instances d'objet de connexion peuvent ne pas être disponibles en raison de la capacité ou des limites de l'identifiant de cible; il peut par ailleurs n'y avoir aucun identifiant de connexion qui ne soit pas attribué. Le dispositif peut également ne pas avoir la capacité nécessaire au traitement de la connexion supplémentaire; etc. Les objets Connexion de sécurité exigent des ressources similaires aux connexions E/S standard.

Tableau 9 – Options de paramétrage du service ForwardOpen pour les connexions de sécurité

Type de connexion O-à-T	Type de connexion T-à-O	Type de transport/déclencheur	Chemin d'application 1	Chemin d'application 2 ^a	Chemin d'application 3 ¹	Nombre maximal de consommateurs	Type de connexion de l'objet de validation de sécurité	Facteurs de décision clés	Rôle cible
Point à point	Point à point	Application, classe 0, client	Chemin Nul défini par le fournisseur ^b	Chemin Nul défini par le fournisseur	Objet d'application d'entrée de sécurité	1	Point à point Producteur (Entrée)	2-point à point + Client	Connexion de production: Données de sécurité Connexion de consommation: Coordination
Point à point	Point à point	Application, classe 0, serveur	Chemin Nul défini par le fournisseur	Objet d'application de sortie de sécurité	Chemin Nul défini par le fournisseur	1	Point à point Consommateur (Sortie)	2-point à point + Serveur	Connexion de consommation: Données de sécurité Connexion de production: Coordination
Point à point	Multipoint	Application, classe 0, client	Chemin Nul défini par le fournisseur	Chemin Nul défini par le fournisseur	Objet d'application d'entrée de sécurité	2-15	Producteur multipoint CP 2/3	2 – Multipoint + 1 point à point + Client	Connexion de production: Données de sécurité Connexion de production: Correction Connexion de consommation: coordination

Type de connexion O-à-T	Type de connexion T-à-O	Type de transport/déclencheur	Chemin d'application 1	Chemin d'application 2 ^a	Chemin d'application 3 ^a	Nombre maximal de consommateurs	Type de connexion de l'objet de validation de sécurité	Facteurs de décision clés	Rôle cible
Point à point	Multipoint	Application, Classe 0, Client	Chemin Nul défini par le fournisseur	Chemin Nul défini par le fournisseur	Objet d'application d'entrée de sécurité	2-15	Producteur multipoint CP 2/2	1 – Multipoint + 1 point à point + Client	Connexion de production: Données de sécurité + correction Connexion de consommation: coordination
Multipoint	Point à point	Application, Classe 0, serveur	Chemin Nul défini par le fournisseur	Objet d'application de sortie de sécurité	Chemin Nul défini par le fournisseur	2-15	Consommateur multipoint CP 2/3	2 – Multipoint + 1 point à point + Serveur	Connexion de consommation: Données de sécurité Connexion de consommation: Correction Connexion de production: coordination
Multipoint	Point à point	Application, Classe 0, serveur	Chemin Nul défini par le fournisseur	Objet d'application de sortie de sécurité	Chemin Nul défini par le fournisseur	2-15	Consommateur multipoint CP 2/2	1 – Multipoint + 1 point à point + Serveur	Connexion de consommation: Données de sécurité + correction Connexion de production: coordination

^a Entrée et sortie par rapport au réseau.

^b Les dispositifs configurables via EDS doivent définir un chemin Nul d'application dans le fichier EDS.

Le Tableau 10 définit les paramètres de connexion de réseau à utiliser pour l'ensemble des trois connexions associées à chaque type de connexion de sécurité.

Tableau 10 – Paramètres de connexion de réseau pour les connexions de sécurité

Type de cible	Paramètre de connexion de réseau	Valeur de paramètre correcte	Description
Producteur point à point	O-à-T	4406 _{hex}	Priorité élevée point à point, fixe
	T-à-O	4408 _{hex}	Priorité élevée point à point, fixe
	Correction de temps	0000 _{hex}	Non utilisé
Producteur point à point	O-à-T	4408 _{hex}	Priorité élevée point à point, fixe
	T-à-O	4406 _{hex}	Priorité élevée point à point, fixe
	Correction de temps	0000 _{hex}	Non utilisé
Producteur multipoint	O-à-T	4406 _{hex}	Priorité élevée point à point, fixe
	T-à-O	240E _{hex}	Priorité élevée, multipoint, fixe
	Correction de temps	2406 _{hex}	Priorité élevée multipoint, fixe

Les réponses SafetyOpen satisfaisantes doivent renvoyer les identifiants de connexion utilisés, ainsi que le Consumer_Number attribué au point d'origine (toujours FFFF_{hex} pour les connexions point à point).

6.3.11 Données de réponse d'application dans une réponse SafetyOpen satisfaisante

Le format d'une réponse SafetyOpen satisfaisante est identique à celui de la Forward_Open_Reply défini dans la CEI 61158-6-2. La transmission des données en retour est définie dans le Tableau 11.

La cible d'une fonction SafetyOpen doit renvoyer les données, définies dans le Tableau 12, dans le champ Réponse d'application de la réponse SafetyOpen. Le nombre de consommateurs et les données de réponse PID sont importants uniquement lorsque le point d'origine est le consommateur.

Tableau 11 – Réponse d'application de cible de sécurité CP 2/3 (taille: 10 octets)

Nom de paramètre	Type de données	Description
Consumer_Number	UINT	
PID/CID	STRUCT de:	Identifiant de cible permettant de générer une valeur de départ PID ou CID
Id du fournisseur de cible	UINT	Id du fournisseur de cible
Numéro de série du dispositif de cible	UDINT	Numero de série du dispositif de cible
Numéro de série de connexion de cible	UINT	Numéro de série de connexion de cible attribué à la connexion. L'identifiant instance de l'objet de validation de sécurité doit être utilisé

Tableau 12 – Réponse d'application de cible SafetyOpen (taille: 16 octets)

Nom de paramètre ^a	Type de données	Description
Consumer_Number	UINT	
PID/CID	STRUCT de:	Identifiant de cible permettant de générer une valeur de départ PID ou CID
Id du fournisseur de cible	UINT	Id du fournisseur de cible
Numéro de série du dispositif de cible	UDINT	Numéro de série du dispositif de cible
Numéro de série de la connexion de cible	UINT	Numéro de série de la connexion de cible attribué à la connexion. L'identifiant instance de l'objet de validation de sécurité doit être utilisé
Time_Correction_Connection_ID	UDINT	Identifiant de connexion choisi par la cible pour la connexion de correction de temps. La valeur est retournée si elle est choisie par le point d'origine
Time_Correction_API	UDINT	Renvoie la même valeur reçue dans le paquet Demande.
^a Ces champs sont toujours présents dans le champ Réponse d'application. Lorsqu'une cible de sécurité CP 2/3 reçoit une fonction SafetyOpen qui n'est pas multipoint (aucune connexion de correction de temps n'est ainsi établie), le paramètre Time_Correction_Connection_ID doit être réglé sur FFFF _{hex} . Si la cible n'est pas un producteur multipoint, le nombre de consommateurs est réglé sur FFFF _{hex} .		

6.3.12 Réponse SafetyOpen non satisfaisante

La fonction SafetyOpen comporte certains cas d'erreurs spéciaux qui ne sont pas mis en correspondance avec l'ensemble existant de codes d'erreurs, et étend la définition des codes d'erreurs existants.

Le Tableau 13 définit de nouveaux codes d'erreurs étendus de sécurité et présente la définition étendue des codes d'erreurs existants.

Tableau 13 – Nouveaux codes d'erreurs étendus de sécurité

Etat général	Etat étendu	Explication
01 _{hex}	0105 _{hex}	Conflit de propriété ou défaut d'adaptation OUNID. La configuration est déjà propriété d'un autre point d'origine
01 _{hex}	0106 _{hex}	Conflit de propriété ou défaut d'adaptation OUNID. La connexion de sortie était déjà propriété d'un autre point d'origine
01 _{hex}	0205 _{hex}	Erreur de paramètre dans le service Unconnected Send (Envoi non connecté) ou Erreur de paramètre dans le service SafetyOpen ou SafetyClose
01 _{hex}	801 _{hex}	RPI multipoint incompatible. Une connexion existante a été établie à un RPI différent
01 _{hex}	802 _{hex}	Taille de connexion de sécurité invalide
01 _{hex}	803 _{hex}	Format de connexion de sécurité invalide
01 _{hex}	804 _{hex}	Paramètres de connexion de correction de temps invalides
01 _{hex}	805 _{hex}	Multiplicateur EPI d'intervalles Ping invalide
01 _{hex}	806 _{hex}	Multiplicateur minimal de messages de coordination temporelle
01 _{hex}	807 _{hex}	Multiplicateur de délai du réseau
01 _{hex}	808 _{hex}	Multiplicateur de temporisation
01 _{hex}	809 _{hex}	Nombre maximal de consommateurs invalide
01 _{hex}	80A _{hex}	CPRC invalide
01 _{hex}	80B _{hex}	ID de connexion de correction de temps invalide
01 _{hex}	80C _{hex}	Défaut d'adaptation SCID. La valeur de SCID était non nulle et ne correspondait pas à la valeur de la cible
01 _{hex}	80D _{hex}	TUNID non défini. Le dispositif est out-of-box (hors connexion) et TUNID n'a pas été défini, les connexions ne sont donc pas admises
01 _{hex}	80E _{hex}	Défaut d'adaptation TUNID. Le TUNID fourni n'est pas adapté. Le message a été vraisemblablement acheminé vers ce nœud par erreur
01 _{hex}	80F _{hex}	Opération de configuration non admise

Le Tableau 14 de matrice d'événements/erreurs fournit des recommandations concernant les codes d'erreurs à appliquer pour les erreurs potentielles détectées lors du traitement d'un service SafetyOpen. Ces recommandations ne sont pas censées constituer une liste exhaustive.

Tableau 14 – Tableau de recommandations concernant les événements d'erreur SafetyOpen

Événement	Type d'erreur	Etat général	Etat étendu
Le dispositif cible a reçu une demande SafetyOpen dans des états de l'objet Programme de contrôle de sécurité autres que l'état de repos et l'état d'exécution	Opération de configuration non admise	01 _{hex}	080F _{hex}
	Dispositif non configuré	01 _{hex}	110 _{hex}
Type (nom) de connexion de sécurité invalide	Erreur de paramètre de sécurité – Type de connexion invalide	01 _{hex}	315 _{hex}
Taille de connexion de sécurité invalide	Erreur de paramètre de sécurité – Taille de connexion de sécurité invalide	01 _{hex}	0802 _{hex}
Format de connexion de sécurité invalide	Erreur de paramètre de sécurité – Format de connexion de sécurité invalide	01 _{hex}	0803 _{hex}
Paramètres de connexion de correction de temps invalides	Erreur de paramètre de sécurité – Paramètres de connexion de correction de temps invalides	01 _{hex}	0804 _{hex}
Multiplicateur EPI d'intervalles Ping hors plage	Erreur de paramètre de sécurité – Multiplicateur EPI d'intervalles Ping invalide	01 _{hex}	0805 _{hex}
Multiplicateur Min. de messages de coordination temporelle hors plage	Erreur de paramètre de sécurité – Multiplicateur minimal de messages de coordination temporelle	01 _{hex}	0806 _{hex}
Multiplicateur de délai hors plage	Erreur de paramètre de sécurité – Multiplicateur de délai	01 _{hex}	0807 _{hex}
Multiplicateur de temporisation hors plage	Erreur de paramètre de sécurité – Multiplicateur de temporisation	01 _{hex}	0808 _{hex}
Nombre maximal de consommateurs non compris dans les limites du producteur multipoint	Erreur de paramètre de sécurité – Nombre maximal de consommateurs invalide	01 _{hex}	0809 _{hex}
Valeur CPCRC dans SafetyOpen et valeur CPCRC calculée	Erreur de paramètre de sécurité – CPCRC invalide	01 _{hex}	080A _{hex}
ID de connexion de correction de temps invalide	Erreur de paramètre de sécurité – ID de connexion de correction de temps invalide	01 _{hex}	080B _{hex}
Le RPI demandé n'est pas compatible avec les connexions multipoint établies précédemment	RPI multipoint incompatible	01 _{hex}	801 _{hex}
RPI O-à-T, RPI T-à-O, ou RPI de correction de temps non pris en charge	RPI non pris en charge	01 _{hex}	0111 _{hex}
Toutes les instances Objet de validation de sécurité sont utilisées	Le Gestionnaire de connexion ou l'objet de connexion ne peut plus prendre en charge de connexions	01 _{hex}	0113 _{hex}

6.3.13 Service ForwardClose de sécurité

Aucun comportement supplémentaire n'est défini pour le service ForwardClose lorsqu'il est utilisé pour fermer une connexion de sécurité. Dans les exigences de sécurité suivantes, le terme « SafetyClose » fait référence tant au service ForwardClose de l'objet Gestionnaire de connexion, qu'au service SafetyClose de l'objet Connexion.

Le service SafetyClose doit être utilisé pour fermer les connexions avec une cible de sécurité (et tous les autres nœuds du chemin de connexion, dans le cas d'un service ForwardClose).

La demande SafetyClose doit supprimer une connexion parmi tous les nœuds qui participent à la connexion d'origine.

La demande SafetyClose doit occasionner la désaffectation de toutes les ressources dans l'ensemble des nœuds qui participent à la connexion, y compris les ID de connexion, la largeur de bande et les mémoires tampons internes.

Les cibles qui reçoivent la demande de service SafetyClose doivent fermer la connexion dont les paramètres de l'ID du fournisseur du point d'origine, du numéro de série de connexion et du numéro de série du point d'origine sont adaptés. Les informations supplémentaires fournies par ce service (c'est-à-dire le chemin de connexion) doivent être ignorées par la cible.

Une réponse satisfaisante SafetyClose doit être transmise après suppression de la connexion au niveau de la cible.

A réception d'une réponse satisfaisante SafetyClose, le point d'origine et chaque nœud intermédiaire le long du chemin, ferment la connexion et libèrent les ressources associées à cette connexion. Toutefois, le point d'origine doit également fermer la connexion si aucune réponse n'est reçue au cours d'une période d'attente sélectionnée par le fournisseur.

6.4 Objet Identité

6.4.1 Généralités

Code de classe: 01hex

Le présent paragraphe décrit les modifications apportées à l'objet Identité pour prendre en charge la sécurité.

Tous les dispositifs de sécurité CPF 2 doivent substituer le comportement défini dans l'objet Programme de contrôle de sécurité au comportement d'état de l'objet Identité.

6.4.2 Modifications apportées aux services communs

Le Tableau 15 définit les services de l'objet Identité qui exigent des actualisations.

Tableau 15 – Modifications apportées aux services communs de l'objet Identité

Code de service	Mise en œuvre nécessaire		Nom de service	Description du service
	Classe	Instance		
05 _{hex}	Facultatif	Conditionnel ^a	Réinitialisation	Appelle le service Réinitialisation pour le dispositif
^a Conditionnel – Le service Réinitialisation de l'objet Identité ne doit pas être pris en charge dans les dispositifs de sécurité (réponse aux demandes avec erreur « Service non pris en charge »). Le service doit être remplacé par le service Réinitialisation défini par le programme de contrôle de sécurité.				

6.5 Objets de liaison

6.5.1 Modifications apportées à l'objet DeviceNet

6.5.1.1 Attribut supplémentaire de sécurité

Code de classe: 03hex

Le présent paragraphe définit les modifications et ajouts nécessaires de l'objet DeviceNet pour prendre en charge la sécurité. Un attribut doit être ajouté pour permettre la lecture du numéro de réseau de sécurité (SNN). La définition de cet attribut est donnée dans le Tableau 16.

Tableau 16 – Nouvel attribut instance de l'objet DeviceNet

ID de l'attribut	Mise en œuvre nécessaire	Règle d'accès	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
11	Requis	Obtenir	Numéro de réseau de sécurité	6 OCTETS	Numéro unique pour l'ensemble du système pour le sous-réseau sur lequel repose ce dispositif	L'attribut doit contenir la partie SNN du TUNID dans le programme de contrôle de sécurité. Voir 8.2 pour une définition ultérieure de cet attribut

Les dispositifs qui mettent en œuvre le FSCP 2/1 sur le CP 2/3 doivent prendre en charge l'attribut 11 de l'objet DeviceNet.

6.5.1.2 Limitation de connexion rapide de sécurité

Tous les dispositifs de sécurité CP 2/3 ne doivent pas prendre en charge la caractéristique Quick Connect (Connexion rapide). L'attribut Connexion Rapide ne doit pas être pris en charge et doit répondre à tout service d'attribut Get/Set (Obtenir/Définir) avec le code d'erreur «Attribut non pris en charge».

6.5.2 Modifications apportées à l'objet Interface TCP/IP

Code de classe: F5hex

Le présent paragraphe définit les modifications et ajouts nécessaires de l'objet Interface TCP/IP pour prendre en charge la sécurité. Un attribut doit être ajouté pour permettre la lecture du numéro de réseau de sécurité (SNN). La définition de cet attribut est donnée dans le Tableau 17.

Tableau 17 – Nouvel attribut instance de l'objet Interface TCP/IP

ID de l'attribut	Mise en œuvre nécessaire	Règle d'accès	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
7	Requis	Obtenir	Numéro de réseau de sécurité	6 OCTETS	Numéro unique pour l'ensemble du système pour le sous-réseau sur lequel repose ce dispositif	L'attribut doit contenir la partie SNN du TUNID dans le programme de contrôle de sécurité. Voir 8.2 pour une définition ultérieure de cet attribut

Les dispositifs qui mettent en œuvre le FSCP 2/1 sur le CP 2/2 Sécurité doivent prendre en charge l'attribut 7 de l'objet Interface TCP/IP.

6.6 Objet Programme de contrôle de sécurité

6.6.1 Généralités

Code de classe: 39 hex

Le programme de contrôle de sécurité constitue l'objet central des dispositifs de sécurité FSCP 2/1. Deux mises en œuvre fondamentales du programme de contrôle de sécurité sont définies:

- Programme de contrôle de référence – défini comme une mise en œuvre de contrôle qui prend en charge tous les attributs et services « requis » définis par cette définition d'objet. Ce programme de référence inclut tous les comportements de diagrammes d'états, à l'exclusion du traitement des événements d'état pour les services non requis (voir 6.6.8);
- Programme de contrôle SNCT – défini comme une mise en œuvre de contrôle qui prend en charge la fonctionnalité de référence dans son intégralité, ainsi que les attributs et services portant le marquage « conditionnel » sur le support SNCT.

Tous les dispositifs de sécurité installés sur un réseau de sécurité FSCP 2/1 doivent prendre en charge la définition du Programme de contrôle de sécurité de référence.

De nombreux paramètres de l'objet Programme de contrôle de sécurité étant considérés critiques pour la sécurité, les attributs de cet objet doivent avoir une intégrité plus élevée que les données d'objet normalisées.

Le comportement d'état de l'objet Programme de contrôle de sécurité doit contrôler l'état du dispositif complet. Tous les objets d'un dispositif de sécurité sont tenus d'être asservis aux états et commandes du programme de contrôle de sécurité afin de gérer ses fonctions et ses comportements

L'objet Programme de contrôle de sécurité centralise les définitions d'état d'objets d'application et les informations d'état associées, les indications d'état d'exception (alarmes et avertissements), et définit un modèle de comportement supposé par les objets identifiés comme appartenant à des dispositifs de sécurité. En d'autres termes, si l'instance d'objet Programme de contrôle de sécurité est soumise à une demande de réinitialisation, la réinitialisation est effectuée par cette instance d'objet, ainsi que par l'ensemble de ses objets d'application associés.

L'objet Programme de contrôle de sécurité doit fournir à l'objet Identité ses informations d'état. Une demande de réinitialisation à l'intention de l'objet Identité n'est pas prise en charge par les dispositifs de sécurité FSCP 2/1 (voir 6.4) et doit être remplacée par une demande de réinitialisation à l'intention de l'objet Programme de contrôle de sécurité qui comporte des qualificatifs supplémentaires capables d'exécuter la demande. D'autres relations sont spécifiées en 6.6.8.

De plus, certains attributs instance sont réservés pour permettre une extension future destinée à mettre en œuvre les profils définis pour l'industrie des semi-conducteurs.

6.6.2 Attributs de classe de programme de contrôle de sécurité

Les attributs de classe utilisés par le programme de contrôle de sécurité sont définis dans le Tableau 18.

Tableau 18 – Attributs de classe de programme de contrôle de sécurité

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	Nom	Type de données	Description de l'attribut
1 - 96	Ces attributs de classe sont décrits dans la CEI 61158-6-2.				
97-98	Réservé par le FSCP 2/1 pour une utilisation future				
99	Conditionnel ^a	Obtenir	Sous-classe	UINT	Identifie un sous-ensemble d'attributs de classe, de services et de comportements supplémentaires
^a Si la valeur de sous-classe est 00, qui identifie « aucune sous-classe », alors il convient d'appliquer cet attribut. Dans le cas contraire, cet attribut doit être appliqué.					

6.6.3 Sous-classes

Chaque sous-classe de niveau de classe donne une signification unique pour un intervalle de recouvrement d'identifiants d'attributs de classe et/ou de services de classe. L'intervalle des définitions de sous-classes commence à la valeur ID 96 avec une numérotation descendante pour les attributs, et à la valeur ID 63_{hex} avec une numérotation descendante pour les services. La sous-classe d'une classe donnée est identifiée par la valeur de son attribut de classe de sous-classe. Aucune sous-classe n'est actuellement définie pour l'instance de classe.

6.6.4 Attributs instance de programme de contrôle de sécurité

Les attributs instance utilisés par le programme de contrôle de sécurité sont définis dans le Tableau 19.

Tableau 19 – Attributs instance de programme de contrôle de sécurité

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	NV	Nom	Type de données	Description de l'attribut
1	Facultatif	Obtenir	NV	Nombre d'attributs	USINT	Nombre d'attributs pris en charge par l'instance d'objet
2	Facultatif	Obtenir	NV	Liste d'attributs	MATRICE de USINT	Liste d'attributs pris en charge par l'instance d'objet
3	Réservé					
4	Réservé					
5	Facultatif	Obtenir	NV	Nom du fabricant	Chaîne courte	Texte ASCII, 20 caractères max., voir 6.6.5.1
6	Facultatif	Obtenir	NV	Numéro de modèle du fabricant	Chaîne courte	Texte ASCII, 20 caractères max., spécifié par le fabricant
7	Facultatif	Obtenir	NV	Niveau de révision logicielle	Chaîne courte	Texte ASCII, 6 caractères max., voir 6.6.5.2
8	Facultatif	Obtenir	NV	Niveau de révision matérielle	Chaîne courte	Texte ASCII, 6 caractères max., voir 6.6.5.3
9	Facultatif	Obtenir	NV	Numéro de série du fabricant	Chaîne courte	Texte ASCII, 30 caractères max., spécifié par le fabricant, voir 6.6.5.4
10	Facultatif	Obtenir	NV	Configuration de dispositif	Chaîne courte	Texte ASCII, 50 caractères max., spécifié par le fabricant. Informations supplémentaires facultatives concernant la configuration du dispositif

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	NV	Nom	Type de données	Description de l'attribut
11	Requis	Obtenir	V	État du dispositif	USINT	Voir 6.6.5.5
12	Requis	Obtenir	V	Etat d'exception	SWORD	Voir 6.6.5.6
13	Conditionnel, basé sur le bit 7 d'état d'exception	Obtenir	V	Alarme de détail d'exception	STRUCT. de:	Structure de 3 éléments contenant une représentation en mode point du détail de l'alarme
				Détail d'exception commun	STRUCT. de:	
				Taille	USINT	Nombre d'octets de détail d'exception commun
				Détail	MATRICE de:	Voir 6.6.5.7
				Détail	SWORD	Voir 6.6.5.7
				Détail d'exception du dispositif	STRUCT. de:	
				Taille	USINT	Nombre d'octets de détail d'exception du dispositif
				Détail	MATRICE de:	Spécifié par le fabricant
				Détail n	SWORD	Spécifié par le fabricant
				Détail d'exception du fabricant	STRUCT. de:	
				Taille	USINT	Nombre d'octets de détail d'exception du fabricant
				Détail	MATRICE de:	Spécifié par le fabricant
				Détail n	SWORD	Spécifié par le fabricant
14	Conditionnel, basé sur le bit 7 d'état d'exception	Obtenir	V	Avertissement de détail d'exception	STRUCT. de:	Structure de 3 éléments contenant une représentation en mode point du détail d'avertissement
				Détail d'exception commun	STRUCT. de:	
				Taille	USINT	Nombre d'octets de détail commun
				Détail	MATRICE de:	Voir 6.6.5.8
				Détail n	SWORD	Voir 6.6.5.8
				Détail d'exception du dispositif	STRUCT. de:	
				Taille	USINT	Nombre d'octets de détail du dispositif
				Détail	MATRICE de:	Spécifié par le fabricant
				Détail n	SWORD	Spécifié par le fabricant
				Détail d'exception du fabricant	STRUCT. de:	
				Taille	USINT	Nombre d'octets de détail du fabricant
				Détail	MATRICE de:	Spécifié par le fabricant
				Détail n	SWORD	Spécifié par le fabricant
15	Requis	Définir	NV	Activation alarme	BOOL	Voir 6.6.5.12

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	NV	Nom	Type de données	Description de l'attribut
16	Requis	Définir	NV	Activation avertissement	BOOL	Voir 6.6.5.12
17	Facultatif	Définir	a	Temps	DATE_AND_TIME	Valeur de l'horloge temps réel interne du dispositif
18	Facultatif	Obtenir	NV	Comportement de l'horloge pendant un cycle d'alimentation	USINT	Décrit le comportement de l'horloge temps réel interne du dispositif (attribut « Time ») pendant un cycle d'alimentation: 0 = [par défaut] l'horloge se réinitialise toujours pendant un cycle d'alimentation 1 = la valeur de l'horloge est archivée dans la mémoire non volatile hors tension 2 = horloge à pile de secours fonctionnant hors alimentation du dispositif 3-255 = réservé
19	Facultatif	Obtenir	NV	Date de dernière maintenance	DATE	Date de dernier entretien du dispositif
20	Facultatif	Obtenir	NV	Date de prochaine maintenance programmée	DATE	Date de recommandation du prochain entretien du dispositif
21	Facultatif	Obtenir	NV	Temporisateur d'expiration de maintenance programmée	INT	Voir 6.6.5.13
22	Conditionnel – Requis si l'expiration d'étalonnage est prise en charge	Définir	NV	Activation d'avertissement de l'expiration de maintenance programmée	BOOL	Voir 6.6.5.14
23	Facultatif	Obtenir	NV	Heures de fonctionnement	UDINT	Indication du nombre d'heures de mise sous tension du dispositif
24	Conditionnel °	Défini uniquement par le service « Verrouillage /Déverrouillage de configuration » Obtention possible	NV	Verrouillage de configuration	BOOL	Marque la configuration de dispositif interne comme vérifiée et verrouillée, ou alternativement comme non vérifiée, non verrouillée et modifiable
25	Conditionnel °	Obtenir	NV	Configuration UNID	10 octets	CFUNID – Identifie le propriétaire de la configuration d'un dispositif: Ensemble des FF _{hex} = configuration par outil uniquement. 0 = sans propriétaire, accepte tout propriétaire

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	NV	Nom	Type de données	Description de l'attribut
26	Requis	Obtenir	NV	Identifiant de configuration de sécurité	10 octets	Le SCID est constitué du CRC de configuration de sécurité + datation de la configuration de sécurité. Il s'agit de la signature de la configuration
27	Requis	Obtenir	NV	UNID Cible (Target UNID)	10 octets	UNID courant du dispositif
28	Conditionnel ^d	Obtenir	NV	Propriétaires des points de connexion de sortie		
				Nombre d'entrées de matrice	UINT	Nombre d'entrées OCPUNID STRUCT
				Propriétaires de sorties	MATRICE de STRUCT	
				OCPUNID	10 octets	UNID du propriétaire des ressources de sortie. 0 = sans propriétaire, accepte tout propriétaire (par défaut)
				Taille du e-chemin (ePath)	USINT	Taille du chemin, nombre d'octets
				Ressource d'application	e-chemin compact	Chemin vers la ressource « avec propriétaire » (c'est-à-dire 20 04 24 01 – classe d'ensemble, instance 1)
29	Conditionnel ^c	Obtenir	V	Proposition de TUNID	10 octets	Valeur UNID qu'un point d'origine tente de définir dans le dispositif
97-98	Réservé par le FSCP 2/1 pour une utilisation future					
99	Conditionnel ^b	Obtenir	NV	Sous-classe	UINT	Identifie un sous-ensemble d'attributs d'instance, de services et de comportements supplémentaires
^a Le comportement non volatil de l'attribut Time dépend de la valeur de l'attribut Comportement de l'horloge pendant un cycle d'alimentation. ^b Si la valeur de sous-classe est 00, qui identifie « aucune sous-classe », alors il convient d'appliquer cet attribut. Dans le cas contraire, cet attribut doit être appliqué. ^c Les dispositifs de sécurité qui prennent en charge l'interface SNCT (c'est-à-dire non configurée par la fonction SafetyOpen ou tout moyen spécifique au fournisseur) doivent prendre en charge cet attribut. ^d Conditionnel – l'attribut Propriétaire de Connexion de sortie doit être pris en charge pour tous les ensembles de sortie de sécurité.						

6.6.5 Sémantique

6.6.5.1 Nom du fabricant

L'attribut Nom du fabricant, identifie le fabricant du dispositif.

Cet attribut n'est pas garanti, par spécification, d'être unique. Il ne constitue par conséquent pas un attribut de substitution à l'attribut correspondant de l'objet Identité, et il convient de ne pas l'utiliser à des fins d'identification.

6.6.5.2 Niveau de révision logicielle

Il s'agit d'une chaîne de textes à code ASCII qui représente la révision du logiciel correspondant au dispositif spécifique identifié par l'objet Identité.

6.6.5.3 Niveau de révision matérielle

Il s'agit d'une chaîne de textes à code ASCII qui représente la révision du matériel identifié par l'objet Identité. Le fabricant du dispositif doit contrôler cette révision de manière à pouvoir assurer la traçabilité des modifications apportées au matériel du dispositif.

6.6.5.4 Numéro de série du fabricant

Cet attribut est une représentation de chaîne du numéro de série du fournisseur du dispositif, qui est formatée pour s'ajuster à la plupart des systèmes de suivi de fabrication. Il n'est pas nécessaire que cet attribut soit adapté au numéro de série de l'objet Identité qui permet d'identifier de manière unique le dispositif dans l'environnement réseau.

6.6.5.5 Etat du dispositif

Cet attribut représente l'état courant du dispositif. Sa valeur change conjointement au changement de l'état du dispositif. Les valeurs d'indication de l'attribut Etat du dispositif sont définies dans le Tableau 20.

Tableau 20 – Valeurs d'indication de l'attribut Etat du dispositif

Valeur d'attribut	Etat
0	Non défini
1	Auto-vérification
2	Repos
3	Exception d'auto-vérification
4	Exécution
5	Abandon
6	Panne critique
7	Configuration
8	Attente de TUNID
9-50	Réservé par FSCP 2/1
51-99	Spécifique au dispositif
100-255	Spécifique au fournisseur

6.6.5.6 Etat d'exception

Attribut à un seul octet dont la valeur indique l'état des alarmes et des avertissements pour le dispositif. Cette indication peut être fournie par l'une des deux méthodes suivantes: de base ou étendue.

Dans le cas de la méthode de base, le bit sept de l'attribut Etat d'exception est réglé sur zéro; la communication de cet attribut permet de consigner de manière exclusive toutes ces exceptions. Le format des bits zéro à six dans ce mode est spécifique au dispositif; il peut toutefois être précisé davantage dans une spécification de profil du dispositif appropriée. Si le format n'est pas spécifié, il est alors équivalent à celui spécifié pour la méthode étendue.

Dans le cas de la méthode étendue, le bit sept de l'attribut Etat d'exception est réglé sur un; la communication de cet attribut permet de consigner ces exceptions, le format étant spécifié dans le Tableau 21. De plus, les attributs Détail d'exception sont pris en charge. Les bits de

l'attribut Etat d'exception sont déterminés par un élément logique « OR » des bits associés de l'attribut Détail d'exception, tel qu'indiqué.

Tableau 21 – Format de l'attribut Etat d'exception

Bit	Mode point de l'état d'exception, bit 7 réglé sur 0 fonction	Mode point de l'état d'exception, bit 7 réglé sur 1 fonction
0	Définition spécifique au dispositif (voir Profil du dispositif)	ALARME/commune au dispositif ^a
1		ALARME/spécifique au dispositif
2		ALARME/spécifique au fabricant
3		Réservé – réglé sur 0
4		AVERTISSEMENT/commun au dispositif ^a
5		AVERTISSEMENT/spécifique au dispositif
6		AVERTISSEMENT/spécifique au fabricant
7	0 = méthode de base	1 = méthode étendue
^a L'alarme ou l'avertissement n'est pas spécifique au type de dispositif ou au fabricant de ce dernier.		

6.6.5.7 Alarme et avertissement relatifs au détail d'exception

Les formats de ces deux attributs sont identiques. Ils sont ainsi décrits conjointement dans le présent paragraphe.

Les attributs relatifs à l'état détaillé des alarmes ou des avertissements sont associés au dispositif. Chaque attribut est une structure contenant trois membres, chacun de ces membres étant relatif respectivement à l'état détaillé des exceptions communes (c'est-à-dire non spécifiques au dispositif), spécifiques au dispositif mais non spécifiques au fabricant, et spécifiques au fabricant. Le détail commun est défini ci-dessous. Le détail spécifique au dispositif est défini dans le profil de dispositif approprié. Le détail spécifique au fabricant est défini par ce dernier. Une valeur de SIZE (TAILLE) nulle indique qu'aucun détail n'est défini pour la structure du détail d'exception associée.

Chacun des trois membres de la structure est défini comme une structure contenant une liste ordonnée (c'est-à-dire une matrice) d'OCTETS de longueur SIZE (TAILLE), et un entier non signé dont la valeur est SIZE (TAILLE). Chaque octet de chaque matrice a une mise en correspondance spécifique. Cette mise en correspondance est formatée sous la forme de 8 bits qui représentent 8 conditions indépendantes, tandis que une valeur de 1 indique l'établissement (ou l'existence) de la condition, et une valeur de 0 indique la suppression (ou l'absence) de la condition. Si un dispositif ne prend pas en charge un détail d'exception, le bit correspondant n'est jamais établi. Les modes points dédiés aux alarmes et avertissements dans les attributs correspondants sont structurés en parallèle, de sorte que l'établissement d'une alarme ou d'un avertissement d'une condition peut être fonction de la gravité. Si une condition ne peut pas être intrinsèquement à la fois une alarme et un avertissement, la position de bits parallèle correspondant à l'autre état doit rester « nulle ».

L'existence d'une structure variable du détail d'exception dépend de la valeur de l'attribut Etat d'exception. L'existence d'une structure variable du détail d'exception est requise uniquement si le bit sept de l'attribut Etat d'exception est réglé sur 1 (indiquant la consignation de la méthode étendue), le bit (parmi les bits zéro à six) dudit attribut correspondant au type d'exception particulier étant également réglé sur 1.

6.6.5.8 Détail d'exception commun

Cette structure se rapporte aux conditions d'exception (c'est-à-dire alarmes ou avertissements) communes à tous les dispositifs. L'élément de détail de la structure est une liste ordonnée (c'est-à-dire une matrice) d'octets de longueur SIZE qui constitue la valeur de

l'élément de structure Size. Pour chaque octet du champ Détail, tous les bits non identifiés sont réservés pour une normalisation future.

Le premier octet de cet attribut est CommonExceptionDetail[0]. Les détails d'exception supplémentaires, s'ils sont fournis, sont appelés CommonExceptionDetail[1] et CommonExceptionDetail[SIZE]. L'exception spécifique associée à chacun des modes point est donnée dans le Tableau 22. La SIZE pour cette révision est de deux (2). Les détails des critères propres à chaque condition d'exception ne relèvent pas du domaine d'application de la présente partie.

6.6.5.9 Valeurs communes de l'attribut Détail d'exception

Tableau 22 – Valeurs communes de l'attribut Détail d'exception

Détail d'exception commun [0] ^a	Détail d'exception commun [1] ^a
Exception de diagnostic interne	Surintensité d'alimentation
Exception de microprocesseur	Alimentation réservée
Exception EPROM	Tension de sortie d'alimentation
Exception EEPROM	Tension d'entrée d'alimentation
Exception RAM	Date de maintenance programmée
Réservé	Notification au fabricant
Exception temps réel interne	Réinitialisation de l'exception
Réservé	Réserve
^a Si un dispositif ne prend pas en charge un détail d'exception, le bit correspondant n'est jamais établi.	

6.6.5.10 Détail d'exception du dispositif

Cette structure, de forme similaire au détail d'exception commun, se rapporte aux conditions d'exception spécifiques aux dispositifs individuels installés sur le réseau, et qui sont définies dans leurs profils de dispositif respectifs. L'élément de détail de la structure est une liste ordonnée (c'est-à-dire une matrice) d'octets de longueur SIZE qui constitue la valeur de l'élément de structure Size. Pour une description détaillée de cet attribut, consulter le profil de dispositif spécifique approprié.

6.6.5.11 Détail d'exception du fabricant

Cette structure (voir Tableau 23), de forme similaire au détail d'exception commun, se rapporte aux conditions d'exception spécifiques aux fabricants des dispositifs individuels installés sur le réseau, et qui sont définies par ces derniers dans leur documentation de produit. L'élément de détail de la structure est une liste ordonnée (c'est-à-dire une matrice) d'octets de longueur SIZE qui constitue la valeur de l'élément de structure Size. Pour une description détaillée de cet attribut, consulter la documentation du fabricant de dispositifs spécifiques appropriée.

Tableau 23 – Synthèse du format de détail d'exception

Composante de détail	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Taille du détail d'exception commun	0	0	0	0	0	0	0	0
Détail d'exception commun 0	Rés.	Panne en temps réel	Rés.	Mémoire données	Mémoire non volatile	Mémoire codage	Micro-processeur	Diagnostic
Détail d'exception commun 1	Rés.	Réinitialisation de l'exception	Notification au fournisseur	Date de maintenance programmée	Tension d'entrée d'alimentation	Tension de sortie d'alimentation	Rés.	Surintensité d'alimentation
Taille du détail d'exception du dispositif	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a
Détail d'exception du dispositif 0	--	--	--	--	--	--	--	--
Détail d'exception du dispositif n	--	--	--	--	--	--	--	--
Taille du détail d'exception du fabricant	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a	TBD ^a
Détail du fabricant 0	--	--	--	--	--	--	--	--
Détail du fabricant n	--	--	--	--	--	--	--	--

^a Consulter la documentation appropriée du fabricant de dispositifs spécifiques pour une définition complète.

6.6.5.12 Activation des alarmes et avertissements

Ces attributs booléens permettent d'activer (1) ou de désactiver (0) le processus de paramétrage des bits d'exception de l'objet Programme de contrôle de sécurité. La désactivation signifie l'absence d'établissement des bits correspondants, cette désactivation contribuant à la suppression de ces bits s'ils avaient été établis. Les états d'alarme et d'avertissement ne sont également pas conservés; lorsqu'ils sont activés, les bits doivent être réglés uniquement si la condition correspondante est vraie.

L'état par défaut de ces attributs Activer est activé (1).

Cet attribut facultatif représente la valeur de l'heure et de la date conservée par l'horloge temps réel du dispositif avec une résolution d'une milliseconde.

La valeur par défaut de l'attribut Time est zéro (0), ce qui correspond à 00:00:00, 1970-01-01.

6.6.5.13 Temporisateur d'expiration de maintenance programmée

Cet attribut, comportant une résolution d'une heure, permet d'engendrer un avertissement signifiant qu'il est nécessaire d'étalonner le dispositif. Un temporisateur du programme de contrôle de sécurité décrémente cet attribut une fois par heure sous tension. Lorsque l'attribut n'est plus positif et l'attribut Activer Avertissement de l'expiration de maintenance programmée est réglé sur activé, cette situation génère la condition Avertissement de l'expiration de maintenance programmée. Ceci entraîne le réglage du bit Avertissement de maintenance programmée.

L'attribut Temporisateur d'expiration de maintenance programmée, lorsqu'il est mis en œuvre, ne doit pas retomber à zéro; l'attribut ne décrémente plus lorsqu'il atteint sa valeur la plus négative. Le temporisateur d'expiration de maintenance programmée, lorsqu'il est mis en œuvre, doit poursuivre sa décrémentation indépendamment de l'état de l'attribut Activer Avertissement de l'expiration de maintenance programmée. Ce même temporisateur, toujours lorsqu'il est mis en œuvre, doit être maintenu dans la mémoire non volatile.

6.6.5.14 Activation d'avertissement de l'expiration de maintenance programmée

Cet attribut booléen permet d'activer (1) ou de désactiver (0) le processus de paramétrage du bit d'exception de maintenance programmée de l'objet Programme de contrôle de sécurité. La désactivation signifie l'absence d'établissement du bit correspondant, cette désactivation contribuant à la suppression de ce bit s'il avait été établi. Lorsqu'il est activé, le bit doit être réglé uniquement si la condition correspondante est vraie.

L'état par défaut de cet attribut Activer est activé (1).

6.6.5.15 Verrouillage de configuration

L'attribut Verrouillage de configuration, lorsqu'il est mis en œuvre avec l'interface SNCT, doit être utilisé pour marquer la configuration du dispositif comme étant vérifiée, et pour verrouiller cette dernière; 0 = Non verrouillé (non vérifié et modifiable), 1 = Verrouillé (vérifié et non modifiable).

La valeur par défaut de l'attribut Verrouillage de configuration est 0 = non verrouillé. Elle est utilisée principalement pour les dispositifs de sécurité configurés par l'interface SNCT (c'est-à-dire non configurés par SafetyOpen). Après configuration du dispositif par outil et sa validation par l'utilisateur, le logiciel règle ce drapeau afin de « neutraliser » les modifications ultérieures éventuelles. Il est possible de protéger ce drapeau par un mot de passe qui nécessite une vérification de l'utilisateur préalablement au déverrouillage et à la modification de la configuration.

Le réglage de l'attribut Verrouillage de configuration d'un dispositif doit entraîner parallèlement le refus des Configure_Requests, de la fonction Réinitialisation de sécurité (Safety Reset) de types 1 et 2 et de toute fonction SafetyOpen de forme 1 valide.

6.6.5.16 Configuration UNID (CFUNID)

L'attribut Configuration UNID doit être mis en œuvre comme partie intégrante de l'interface SNCT. La configuration UNID identifie le propriétaire de la configuration dans ce dispositif de sécurité. L'attribut Configuration UNID peut être réglé par différents moyens. Les significations spéciales suivantes doivent être attribuées à l'attribut Configuration UNID:

- Tous les octets réglés sur 00_{hex} = Pas de propriétaire, accepte tout propriétaire,
- Tous les octets réglés sur FF_{hex} = l'outil logiciel est le propriétaire attribué.

La valeur par défaut de la configuration UNID doit correspondre au réglage de tous les octets sur 00_{hex}.

Lorsque l'attribut Configuration UNID est réglé sur « Pas de propriétaire, accepte tout propriétaire », les dispositifs de sécurité FSCP 2/1 doivent saisir la première source de configuration (Forme 1 et/ou Configure_Request) comme étant le propriétaire (voir Tableau 24).

La valeur CFUNID est utilisée conjointement à la valeur OUNID qui accompagne la Configure_Request ou une fonction SafetyOpen de forme 1. Voir le Tableau 24 pour la définition de la méthode utilisée par les dispositifs pour répondre aux divers services basés

sur la valeur de réglage de l'attribut CFUNID au moment de la réception du service. L'interface SNCT doit toujours utiliser toutes les valeurs FF_{hex} pour l'attribut OUNID.

EXEMPLE Si l'attribut CFUNID est réglé sur toutes les valeurs FF_{hex}, l'attribut OUNID de Configure_Request doit être égal à cette valeur pour traiter la commande.

Tableau 24 – Synthèse du comportement des dispositifs pour diverses valeurs CFUNID

Service	Valeur de CFUNID		
	FF _{hex}	00 _{hex}	OUNID
Configure_Request (FF _{hex}) de SNCT	Acceptation	Acceptation Définir CFUNID = FF _{hex}	Échec
Configure_Request (OUNID)	Échec	Acceptation Définir CFUNID = OUNID	Accepter uniquement si CFUNID = OUNID Échec à défaut
SafetyOpen de forme 1	Échec	Acceptation Définir CFUNID = OUNID	Accepter uniquement si CFUNID = OUNID Échec à défaut

6.6.5.17 Identifiant de configuration de sécurité (SCID)

L'attribut Identifiant de configuration de sécurité est la signature de la configuration de sécurité. Cet attribut est une concaténation du CRC de la configuration de sécurité (SCCRC) + la datation de la configuration de sécurité (SCTS). Ces valeurs sont obtenues à partir de la fonction SafetyOpen ou du logiciel de configuration de sécurité au cours de la validation de configuration.

La valeur par défaut de l'attribut SCID du programme de contrôle de sécurité doit être 0.

Lorsqu'un dispositif adopte le mode Configuration, l'attribut SCID doit être réglé sur 0 et maintenu à cette valeur (par l'intermédiaire de cycles d'alimentation) jusqu'à exécution satisfaisante de Valider (validation).

```
typedef struct _S_SCID
{
    DWORD    dwCRC;    //calculated by device
    DWORD    dwTime;   // set by software
    WORD     wDate;
} S_SCID;
```

6.6.5.18 UNID Cible(TUNID)

Il s'agit d'un attribut NV qui reflète l'UNID que la configuration a sauvegardé.

L'attribut UNID Cible du programme de contrôle de sécurité doit toujours refléter soit l'UNID cible par défaut ou courant que le processus de paramétrage UNID a sauvegardé.

L'UNID d'un dispositif doit être archivé dans la mémoire non volatile NV comme partie intégrante de la procédure de paramétrage UNID.

L'UNID d'un dispositif doit être archivé dans l'objet DeviceNet ou l'attribut SNN de l'objet Interface TCP/IP, comme partie intégrante de la procédure de paramétrage UNID.

La mémoire non volatile d'un dispositif présentant un état par défaut du fabricant doit comporter une valeur UNID non valide (par exemple FF_{hex}, FF_{hex}, FF_{hex}, FF_{hex}, FF_{hex}, FF_{hex}).

L'UNID d'un dispositif doit être lu à partir de la mémoire non volatile lors de la mise sous tension, puis téléchargé sur un attribut de l'objet Programme de contrôle de sécurité et l'attribut Safety_Network_Number de l'objet DeviceNet ou de l'objet Interface TCP/IP.

Si l'UNID représente un Nombre valide (non égal à la valeur du dispositif à l'état "out-of-box"), ce nombre doit être comparé aux attributs d'objet d'adresse MACID ou d'adresse IP. En cas d'adaptation, la procédure de démarrage du dispositif se poursuit. En cas de défaut d'adaptation, le dispositif passe à l'état Abandon (Abort).

Lorsque la défaillance d'un dispositif a pour origine le défaut d'adaptation du paramétrage Commutation mémoire non volatile (Switch-NV), ce dispositif doit permettre soit une réinitialisation sur l'état out-of-box, soit la transmission d'une demande Récupération (Recover) au programme de contrôle de sécurité afin de supprimer la fonction Abandon. Dans les deux cas, le dispositif doit passer à l'état « Attente de TUNID ».

La lecture de l'attribut UNID dans le programme de contrôle de sécurité doit être effectuée avec une intégrité de sécurité.

L'application des données de configuration et de l'UNID à la mémoire non volatile (NV) doit s'effectuer avec une intégrité de sécurité.

6.6.5.19 Proposition de TUNID (OUNID)

La valeur par défaut de l'attribut Proposition de TUNID doit correspondre à toutes les valeurs FF_{hex}.

Cet attribut reflète la valeur proposée lorsque le dispositif suit le processus "ProposeTUNIG/ApplyTUNID". Une fois l'application achevée, cet attribut doit être réglé de nouveau sur la valeur par défaut out-of-box de toutes les valeurs FF_{hex}.

6.6.5.20 Propriétaire de connexion de sortie (OCPUNID)

Cet attribut n'est utilisé que pour les connexions de sécurité sur les sorties. Les connexions de sécurité aux sorties définissent un propriétaire afin d'éviter que des connexions errantes ne détournent une ressource de sortie dans un système de sécurité validé. Cet identifiant de propriétaire identifie le dispositif de sécurité d'origine auquel ont été conférés les droits de propriété pour cette connexion de sortie de sécurité. Ce paramètre provient des paramètres du Segment de sécurité de SafetyOpen. La valeur OCPUNID peut être identique à celle de l'attribut 25 du programme de contrôle de sécurité (CFUNID) lorsque le dispositif est également configuré comme partie intégrante de SafetyOpen. Dans la mesure où un dispositif peut comporter deux ressources de sortie ou plus, et peut admettre l'existence de propriétaires distincts de chaque ressource, cet attribut prend la forme d'une structure permettant le maintien séparé de chaque ressource. La ressource est identifiée par une adresse de Segment logique.

La valeur par défaut pour tous les OCPUNID doit être 0 (accepte tout propriétaire), chaque ePath étant dirigé vers la ressource respective.

6.6.6 Sous-classes

Chaque sous-classe de niveau d'instance donne une signification unique pour un intervalle de recouvrement d'identifiants d'attributs instance et/ou de services instance. L'intervalle des définitions de sous-classes commence à la valeur ID 96 avec une numérotation descendante pour les attributs, et à la valeur ID 63_{hex} avec une numérotation descendante pour les services. La sous-classe d'une instance donnée est identifiée par la valeur de son attribut instance Sous-classe. Aucune sous-classe n'est actuellement définie pour les instances.

6.6.7 Services communs du programme de contrôle de sécurité

6.6.7.1 Présentation générale des services communs

Les services communs pris en charge par le programme de contrôle de sécurité sont définis dans le Tableau 25. Les services spécifiques de l'objet Programme de contrôle de sécurité sont définis dans le Tableau 26.

Tableau 25 – Services communs du programme de contrôle de sécurité

Code de service	Mise en œuvre nécessaire		Nom de service	Description du service
	Classe	Instance		
0E _{hex}	Conditionnel	Requis	Get_Attributes_Single	Renvoie le contenu de l'attribut spécifié
010 _{hex}	n/a	Requis	Set_Attributes_Single	Modifie une valeur d'attribut
0D _{hex}	n/a	Conditionnel ^a	Application	Applique une configuration en attente à la mémoire non volatile, et dirige le dispositif vers l'état Repos
15 _{hex}	n/a	Facultatif	Restauration	Restaure la dernière configuration appliquée dans les dispositifs qui la prennent en charge
^a Les dispositifs de sécurité qui prennent en charge l'interface SNCT (c'est-à-dire non configurée par SafetyOpen ou tout moyen spécifique au fournisseur) doivent prendre en charge ce service.				

Le dispositif peut solliciter l'existence d'un service instance de Programme de contrôle de sécurité en interne, tel que spécifié par le fabricant.

Tableau 26 – Services spécifiques de l'objet Programme de contrôle de sécurité

Code de service	Mise en œuvre nécessaire		Nom de service	Description du service
	Classe	Instance		
4B _{hex}			Réservé	
4C _{hex}	n/a	Facultatif	Récupération	Retire le dispositif de l'état Abandon
4E _{hex}	n/a	Facultatif	Perform_Diagnostics	Contribue à l'exécution d'un ensemble de routines de diagnostic par le dispositif le cas échéant
4F _{hex}	n/a	Conditionnel ^a	Configure_Request	Dirige le dispositif vers l'état Configuration le cas échéant. Ce service exige la saisie d'un mot de passe pour son exécution
50 _{hex}	n/a	Conditionnel ^a	Validate_Configuration	Contribue à l'exécution d'un contrôle de validation de la configuration en attente par le dispositif. Ce service exige du logiciel qu'il fournisse le SCID (CRC + Datation)
51 _{hex}	n/a	Conditionnel ^a	Set_Password	Permet de définir un attribut de mot de passe dans l'objet Programme de contrôle
52 _{hex}	n/a	Conditionnel ^a	Configuration_Lock/Unlock	Ce service intervient suite à la présence de l'attribut Verrouillage de configuration dans le Programme de contrôle

Code de service	Mise en œuvre nécessaire		Nom de service	Description du service
	Classe	Instance		
				de sécurité. Ce service exige la saisie d'un mot de passe pour son exécution
53 _{hex}	n/a	Conditionnel ^c	Changement de mode	Doit être mis en œuvre pour les dispositifs logiques qui prennent en charge l'« exécution », indépendamment des connexions E/S. L'exécution des changements de mode dans les dispositifs de sécurité exige un mot de passe
54 _{hex}	n/a	Type 0 requis, Types 1 et 2 Conditionnel ^a	Réinitialisation de sécurité	Réinitialise le dispositif d'une manière similaire à l'objet Identité, à l'exception du fait que l'exécution exige un mot de passe
55 _{hex}	n/a	Conditionnel ^b	Réinitialisation mot de passe	Permet de réinitialiser le mot de passe dans un dispositif. Le contenu des données de ce message est spécifique au fournisseur
56 _{hex}	n/a	Conditionnel ^d	Propose_TUNID	Permet d'initier le paramétrage du TUNID dans un dispositif à l'état out-of-box
57 _{hex}	n/a	Conditionnel ^d	Apply_TUNID	Permet d'effectuer pleinement le paramétrage TUNID
^a Doit être mis en œuvre pour les dispositifs de sécurité FSCP 2/1 qui prennent en charge l'interface SNCT. ^b Conditionné par la prise en charge du service de mot de passe comme partie intégrante de l'interface SNCT. ^c Doit être mis en œuvre pour les dispositifs de sécurité FSCP 2/1 qui prennent en charge l'interface SNCT et qui sont des dispositifs logiques qui prennent en charge l'« exécution », indépendamment des connexions E/S. L'exécution des changements de mode dans les dispositifs de sécurité exige un mot de passe. ^d Doit être mis en œuvre pour les dispositifs de sécurité FSCP 2/1 qui prennent en charge l'interface SNCT, mais vivement recommandé pour les dispositifs configurables par la fonction SafetyOpen de forme 1, mais qui ne prennent pas en charge l'interface SNCT.				

Le dispositif peut solliciter l'existence d'un service instance de Programme de contrôle de sécurité en interne, tel que spécifié par le fabricant. Généralement, ces demandes sont générées suite à un événement tel que l'activation d'un bouton ou la fermeture d'un contact externe.

6.6.7.2 Service de récupération

Ce service permet de faire passer les objets d'application de dispositifs de l'état de panne récupérable à l'état de repos. Cette demande de service peut être émise en interne par les objets d'application.

6.6.7.3 Service Perform_Diagnostics

Le service Perform_Diagnostics permet d'indiquer à l'objet Programme de contrôle de sécurité d'effectuer un essai de diagnostic. Cet essai est de type commun ou dépendant du dispositif. Les essais de diagnostic communs peuvent comprendre les éléments suivants: RAM, EPROM, mémoire non volatile et communications. La structure des essais de diagnostic de

type commun est spécifique à la mise en œuvre. Les détails des diagnostics dépendant du dispositif ne relèvent pas du domaine d'application de la présente partie.

6.6.7.4 Service Configure_Request

Le service Configure_Request permet de faire passer les objets d'application de dispositif de l'état Abandon, Exécution ou Repos à l'état Configuration.

En cas de réception d'une Configure_Request, une erreur "Conflit d'état d'objet (0Chex)" doit être retournée si l'attribut Configuration_Lock est défini. La demande contient trois paramètres, à savoir le paramètre du mot de passe contenu dans le programme de contrôle de sécurité, l'UNID de dispositif cible et l'UNID du point d'origine. Ces valeurs sont adaptées au mot de passe, à TUNID et à CFUNID du dispositif. La présence de l'OUNID est nécessaire pour permettre au dispositif de déterminer si la source est un outil logiciel ou un point d'origine de capacité ACR.

En cas d'acceptation de la commande Configure_Request, le dispositif doit accepter uniquement les messages de configuration destinés aux objets relatifs à la sécurité sur « cette seule connexion ». Les tentatives d'écriture d'autres messages destinés aux objets relatifs à la sécurité doivent être rejetées.

En cas d'échec d'une connexion de configuration pour quelque motif que ce soit, une autre Configure_Request doit être reçue (établissant ainsi une nouvelle source de connexion) avant de pouvoir effectuer des modifications de configuration (même après un cycle d'alimentation).

La structure de Configure_Request est définie dans le Tableau 27.

Tableau 27 – Structure de message Configure_Request

Nom	Type de données	Description
Code de service	USINT	4F _{hex}
Mot de passe	MATRICE de 16 OCTETS	
TUNID	MATRICE de 10 OCTETS	UNID de dispositif cible. Permet de détecter les demandes mal acheminées
OUNID	MATRICE de 10 OCTETS	UNID du dispositif d'origine. Si le point d'origine est un outil logiciel, ce paramètre doit être réglé sur toutes les valeurs FF _{hex} . Si le point d'origine est un dispositif de capacité ACR, ce paramètre doit être l'OUNID du dispositif

Un code d'erreur «Paramètre invalide » (20_{hex}) doit être retourné en cas d'erreur de défaut d'adaptation TUNID ou OUNID lors du traitement d'une demande de configuration du Programme de contrôle de sécurité.

Un code d'erreur «Violation de privilège » (0F_{hex}) doit être retourné en cas de défaut d'adaptation de mot de passe lors du traitement d'une demande de configuration du Programme de contrôle de sécurité.

6.6.7.5 Service Validate_Configuration

Ce service conduit le dispositif à valider une configuration en attente, par la réalisation d'un calcul CRC avec ces données et en le comparant au CRC associé au message. Si le CRC ne produit pas le même résultat, une réponse d'erreur telle que définie ci-dessous est retournée. Dans le cas contraire, une réponse « réussite » est retournée. Voir la Figure 79 et la Figure 810 pour les détails de traitement de ce service. Cette commande ne peut être reçue

que sur la connexion qui a placé le dispositif en mode configuration. La mémoire non volatile du dispositif archive le SCID final dans l'attribut 26 du Programme de contrôle de sécurité.

La structure du message Validate_Configuration est définie dans le Tableau 28.

Tableau 28 – Structure de message Validate_Configuration

Nom	Type de données	Description
Code de service	USINT	50 _{hex}
SCID	STRUCT. de:	ID de configuration de sécurité
SCCRC	4 OCTETS	Proposition de CRC de configuration de sécurité tel que calculé par le logiciel. Cette proposition est validée par le dispositif
SCTS	6 OCTETS	Datation de configuration de sécurité. Cette valeur marque la date et l'heure de la configuration, qui est utilisée comme mécanisme de détection des modifications

La structure de la Réponse de validation aboutie (Success Validate Reply) est définie dans le Tableau 29.

Tableau 29 – Structure de message de réussite Validate_Configuration

Nom	Type de données	Description
Code de service	USINT	50 _{hex}
SCID	STRUCT. de:	ID de configuration de sécurité
SCCRC	4 OCTETS	CRC de configuration de sécurité tel que calculé par le dispositif
SCTS	6 OCTETS	Datation de configuration de sécurité. Cette valeur reflète les données transmises dans la demande Validation

En cas d'échec de la commande Validate_Configuration du Programme de contrôle de sécurité, la réponse d'erreur doit transmettre une "erreur de validation" définie par la classe (D0_{hex}). Le code d'erreur suivant présenté dans le Tableau 30 est défini comme une erreur de validation dans le Programme de contrôle de sécurité, les codes d'erreur supplémentaires sont définis dans le Tableau 31.

Tableau 30 – Code d'erreur Validate_Configuration

Code d'erreur général	Code d'erreur supplémentaire	Nom d'erreur	Description
D0 _{hex}	Voir Tableau 31	Erreur de validation	Cette erreur est une erreur spécifique de classe pour le Programme de contrôle de sécurité; elle est transmise en réponse à une erreur Validate_Configuration.

Tableau 31 – Codes étendus Validate_Configuration

Codes d'erreur supplémentaires	Code de raison	Description
1	Défaut d'adaptation CRC	Il s'agit du code supplémentaire non spécifique par défaut, en l'absence de toute autre information supplémentaire et lorsque le CRC de configuration ne correspond pas à la valeur transmise
2	Paramètre de configuration invalide	Ce code peut être utilisé pour indiquer une erreur de paramètre si un dispositif procède à une validation spécifique des données de configuration. NOTE Ceci peut se produire même en cas de mise en correspondance des CRC.
3	TUNID non défini	Ce code peut être utilisé pour indiquer que le dispositif est en attente de l'état TUNID et que son TUNID nécessite d'être défini

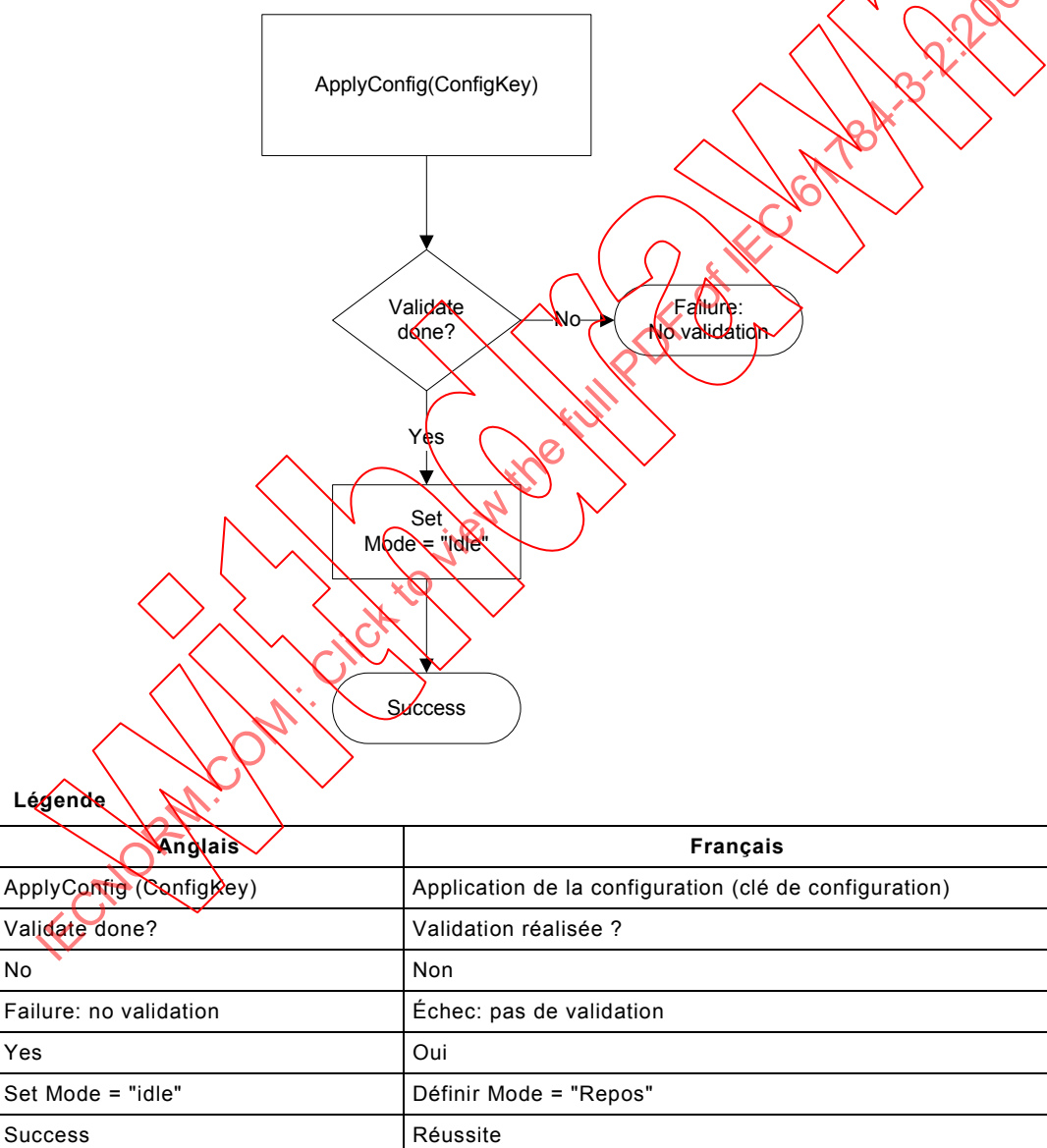
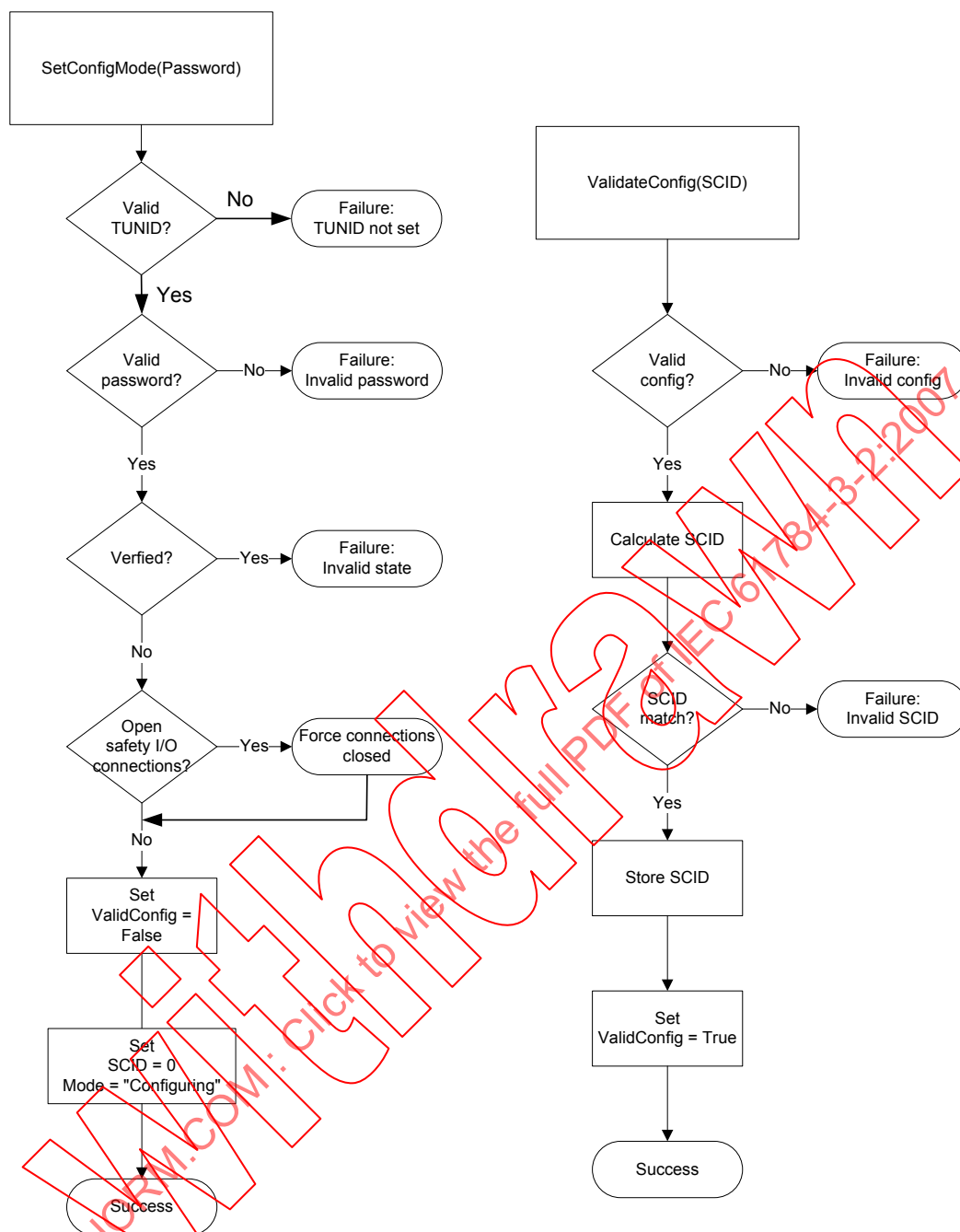


Figure 9 – Application de configuration de dispositif



Légende

Anglais	Français
SetConfigMode (Password)	Définition du mode de configuration (Mot de passe)
Valid TUNID?	TUNID valide ?
No	Non
failure: TUNID not set	Echec: TUNID non défini
yes	Oui
valid password?	Mot de passe valide ?
failure: invalide password	Echec: mot de passe invalide
valid config?	Config valide ?
failure: invalid config	Echec: config invalide
verified?	Vérifié ?

failure: invalid state	Echec: état invalide
calculate SCID	Calculer SCID
openSafety I/O connections?	Connexions E/S safetyopen
force connections closed	Forcer les connexions fermées
SCID match?	SCID adapté ?
failure: invalid SCID	Echec: SCID invalide
set	Définir ...
store SCID	Archiver SCID
success	Réussite

Figure 10 – Organigrammes de traitement de Configuration et Validation

6.6.7.6 Service Set_Password

Le service Set_Password conduit le dispositif à définir un mot de passe dans le Programme de contrôle. Ce service utilise 2 paramètres. Le premier d'entre eux est un paramètre mot de passe qui représente le mot de passe existant, le second étant un paramètre qui représente le nouveau paramètre. Lors du premier paramétrage d'un mot de passe, le paramètre de mot de passe courant est réglé sur la valeur par défaut Out-of-box (zéro).

La structure de Set_Password est définie dans le Tableau 32.

Tableau 32 – Structure de message Set_Password

Nom	Type de données	Description
Code de service	USINT	51 _{hex}
Mot de passe courant	16 OCTETS	
Nouveau mot de passe	16 OCTETS	

Un code d'erreur «Violation de privilège» (0F_{hex}) doit être retourné en cas de défaut d'adaptation du mot de passe courant lors du traitement d'une commande « Définition du mot de passe » du Programme de contrôle de sécurité.

Un mot de passe de caractères réduits par rapport au paramètre de mot de passe doit être aligné à droite en ordre petit-boutiste selon l'octet de poids faible du paramètre de mot de passe de 16 octets. Les octets supérieurs inutilisés du paramètre de mot de passe doivent être remplis de zéros.

6.6.7.7 Service Reset_Password

Ce service permet la réinitialisation des mots de passe dans un dispositif. Le contenu exact et l'usage sont spécifiques au fournisseur afin de protéger l'intégrité des mots de passe. Ce service requiert l'utilisation de 1 paramètre et d'un champ de données spécifique au fournisseur. Le paramètre requis est un champ de taille de données qui permet d'identifier la taille du champ de données spécifique au fournisseur. Le contenu des données du fournisseur est supposé être unique au dispositif et le moyen d'utilisation de ce service doit être prévu par chaque fournisseur.

La structure de Reset_Password est définie dans le Tableau 33.

Tableau 33 – Structure de message Reset_Password

Nom	Type de données	Description
Code de service	USINT	55 _{hex}
Taille des données	USINT	Nombre d'octets de données spécifiques au fournisseur
Données du fournisseur	Taille des données	Données du fournisseur contenant d'autres données nécessaires à la réinitialisation du mot de passe

En cas d'erreur « Taille de données » au niveau de la commande Reset_Password du Programme de contrôle de sécurité, le code d'erreur 15_{hex} (trop grand nombre de données) ou 13_{hex} (trop faible nombre de données) doit être retourné.

Un code d'erreur « Violation de privilège » (0F_{hex}) doit être retourné en cas de défaut d'adaptation des données du fournisseur lors du traitement d'une commande Reset_Password du Programme de contrôle de sécurité.

6.6.7.8 Service Configuration_Lock/Unlock

Le service Configuration_Lock/Unlock conduit le dispositif à modifier l'attribut Configuration_Lock dans le Programme de contrôle de sécurité. Ce service requiert un paramètre de valeur (0 = déverrouillé, 1 = verrouillé) et le paramètre de mot de passe. Le paramètre de mot de passe est comparé au mot de passe archivé et est exécuté uniquement en cas d'adaptation du mot de passe. Une erreur « Défaut d'adaptation de mot de passe » est retournée en cas de défaut d'adaptation du mot de passe.

La structure de Configuration_Lock/Unlock est définie dans le Tableau 34.

Tableau 34 – Structure de message Configuration_Lock/Unlock

Nom	Type de données	Description
Code de service	USINT	52 _{hex}
Valeur	BOOL	0 = non verrouillé, 1 = verrouillé
Mot de passe	16 OCTETS	
TUNID	10 OCTETS	UNID de dispositif cible. Permet de détecter les demandes mal acheminées

En cas de défaut d'adaptation de TUNID de la commande Configuration_Lock/Unlock du Programme de contrôle de sécurité, le code d'erreur (020_{hex}) "Paramètre invalide" doit être retourné.

Un code d'erreur « Violation de privilège » (0F_{hex}) doit être retourné en cas de défaut d'adaptation du mot de passe lors du traitement de la commande Configuration_Lock/Unlock du Programme de contrôle de sécurité.

6.6.7.9 Service de changement de mode

Le service Changement de mode est utilisé par les dispositifs devant passer de l'état Repos à l'état Exécution, indépendamment de l'état de connexion. L'exécution de cette commande exige une autorisation appropriée, d'où la nécessité du mot de passe.

La structure du service Changement de mode est définie dans le Tableau 35.

Tableau 35 – Structure de message Mode_Change

Nom	Type de données	Description
Code de service	USINT	53 _{hex}
Valeur	BOOL	0 = Repos, 1 = Exécution
Mot de passe	16 OCTETS	

Un code d'erreur «Violation de privilège» (0F_{hex}) doit être retourné en cas de défaut d'adaptation du mot de passe de la commande « Changement de mode» du Programme de contrôle de sécurité.

6.6.7.10 Service Safety_Reset

Ce service permet de réinitialiser le dispositif. Le service Safety_Reset se substitue au service de réinitialisation normalisé défini, de telle sorte qu'il peut être qualifié par le mot de passe.

En cas de réception d'une Réinitialisation de sécurité de type 1 ou 2, une erreur "Conflit d'état d'objet (0C_{hex})" doit être retournée si l'attribut Configuration_Lock est défini.

La structure de Safety_Reset est définie dans le Tableau 36.

Tableau 36 – Structure de message Safety_Reset

Nom	Type de données	Description
Code de service	USINT	54 _{hex}
Type de réinitialisation	USINT	Voir Tableau 37
Mot de passe	16 OCTETS	
TUNID	10 OCTETS	UNID de dispositif cible. Permet de détecter les demandes mal acheminées
Attribute_Bit_Map	USINT	Paramètre en mode point qui indique quels attributs doivent être préservés pour la réinitialisation. Ce paramètre est inclus uniquement lorsque le type de réinitialisation est 2. Voir le Tableau 38.

En cas de défaut d'adaptation de TUNID lors de la transmission de la commande Réinitialisation de sécurité du Programme de contrôle de sécurité, le code d'erreur (20_{hex}) «Paramètre invalide» doit être retourné.

Un code d'erreur «Violation de privilège» (0F_{hex}) doit être retourné en cas de défaut d'adaptation du mot de passe lors du traitement d'une commande «Réinitialisation de sécurité» du Programme de contrôle de sécurité.

Tableau 37 – Types de réinitialisation de sécurité du Programme de contrôle de sécurité

Valeur:	Type de réinitialisation
0	Emule le mieux possible la puissance d'itération (des cycles) du dispositif
1	Revient le plus précisément possible à la configuration par défaut, puis émule le mieux possible la puissance d'itération (des cycles)
2	Revient le plus précisément possible à la configuration out-of-box, sauf pour protéger les paramètres indiqués par l'attribut Mode point, puis émule le mieux possible la puissance d'itération (des cycles)

Tableau 38 – Paramètre d'attribut Mode point

Bit	Affectations de l'attribut Mode point
0	Lorsque défini, protège le MACID défini par le logiciel
1	Lorsque défini, protège le débit en bauds défini par le logiciel
2	Lorsque défini, protège le TUNID
3	Lorsque défini, protège le mot de passe
4	Lorsque défini, protège le CFUNID
5	Lorsque défini, protège le OCPUNID
6	Réservé, toujours 0
7	Utilise la mise en correspondance étendue (à définir ultérieurement)

Le Tableau 39 montre de quelle manière un dispositif de sécurité doit traiter les différents types de réinitialisation basés sur la condition Verrouillage/Déverrouillage et sur l'état de connexion de sécurité.

Tableau 39 – Règles de traitement applicables aux types de réinitialisation

Type de réinitialisation	Conditions potentielles à réception d'une Réinitialisation			
	Ouverture / Verrouillage de la connexion de sécurité	Ouverture / Déverrouillage de la connexion de sécurité	Aucune connexion de sécurité / Verrouillé	Aucune connexion de sécurité / Déverrouillé
Type 0	Réponse d'erreur de mode/état	Réponse d'erreur de mode/état	Traitement et réponse	Traitement et réponse
Type 1	Réponse d'erreur de mode/état	Réponse d'erreur de mode/état	Réponse d'erreur de mode/état	Traitement et réponse
Type 2	Réponse d'erreur de mode/état	Réponse d'erreur de mode/état	Réponse d'erreur de mode/état	Traitement et réponse

6.6.7.11 Service Propose_TUNID

Le service Propose_TUNID établit une proposition de paramétrage TUNID dans le dispositif. La valeur n'est pas archivée dans la mémoire non volatile ou dans l'attribut TUNID jusqu'à réception du service d'application.

Le service Propose_TUNID doit être accepté uniquement lorsque le dispositif cible est dans l'état "Waiting_for_TUNID".

Une fois que le TUNID est défini avec succès, il peut passer à l'état Configuration.

Le MACID ou la partie IP du TUNID doit être adapté(e) au MACID de l'objet DeviceNet ou à l'attribut IP de l'objet TCP/IP.

La valeur de l'attribut MACID de l'objet DeviceNet ou de l'attribut IP de l'objet TCP/IP est le résultat de la lecture des commutateurs ou du réglage par outil. Dans chaque cas, la partie MACID/IP de Proposed_TUNID doit correspondre à ce résultat.

La structure du service Propose_TUNID est définie dans le Tableau 40.

Tableau 40 – Service Propose_TUNID

Nom	Type de données	Description
Code de service	USINT	56 _{hex}
TUNID	10 OCTETS	Proposition de valeur UNID de dispositif cible

Si un point d'origine souhaite annuler une proposition/application, la transmission d'une proposition de service avec un TUNID de toutes les valeurs FF_{hex} doit annuler la proposition/application.

Le pseudo-code suivant définit le comportement du service Propose_TUNID:

```

TUNIDProposedHandler()
{
  IF (Device_State=Waiting_for_TUNID)
    THEN IF (Propose_TUNID[TUNID] != all FFhex)
      THEN IF (Propose_TUNID[TUNID_MACID] == MACID attribute/IP
attribute)
        THEN
          TUNIDProposedAttr = Propose_TUNID[TUNID]
          NET LED Flash sequence is started
          Return Success Reply
        ELSE
          Error Code "Invalid Parameter" is returned
      ELSE // cancel propose_apply operation, returns att. to all FFhex
        Stop NET LED Flash Sequence
        TUNIDProposedAttr = Propose_TUNID[TUNID]
        Return Success Reply
    ELSE
      Error Code "Object State Conflict" is returned
}

```

6.6.7.12 Service Apply_TUNID

La structure du service Apply_TUNID est définie dans le Tableau 41 et un organigramme de traitement est défini à la Figure 11.

Tableau 41 – Service Apply_TUNID

Nom	Type de données	Description
Code de service	USINT	57 _{hex}
TUNID	10 OCTETS	Proposition de valeur UNID de dispositif cible

Le service Apply_TUNID doit valider le TUNID par rapport à la valeur proposée. Il interrompt également le clignotement DEL, sauvegarde le TUNID dans la mémoire non volatile, actualise l'attribut TUNID dans le Programme de contrôle, et règle l'attribut proposed_TUNID sur toutes les valeurs FF_{hex}. Le Programme de contrôle doit alors passer à l'état Configuration.

Le pseudo-code suivant définit le comportement du service Apply_TUNID:

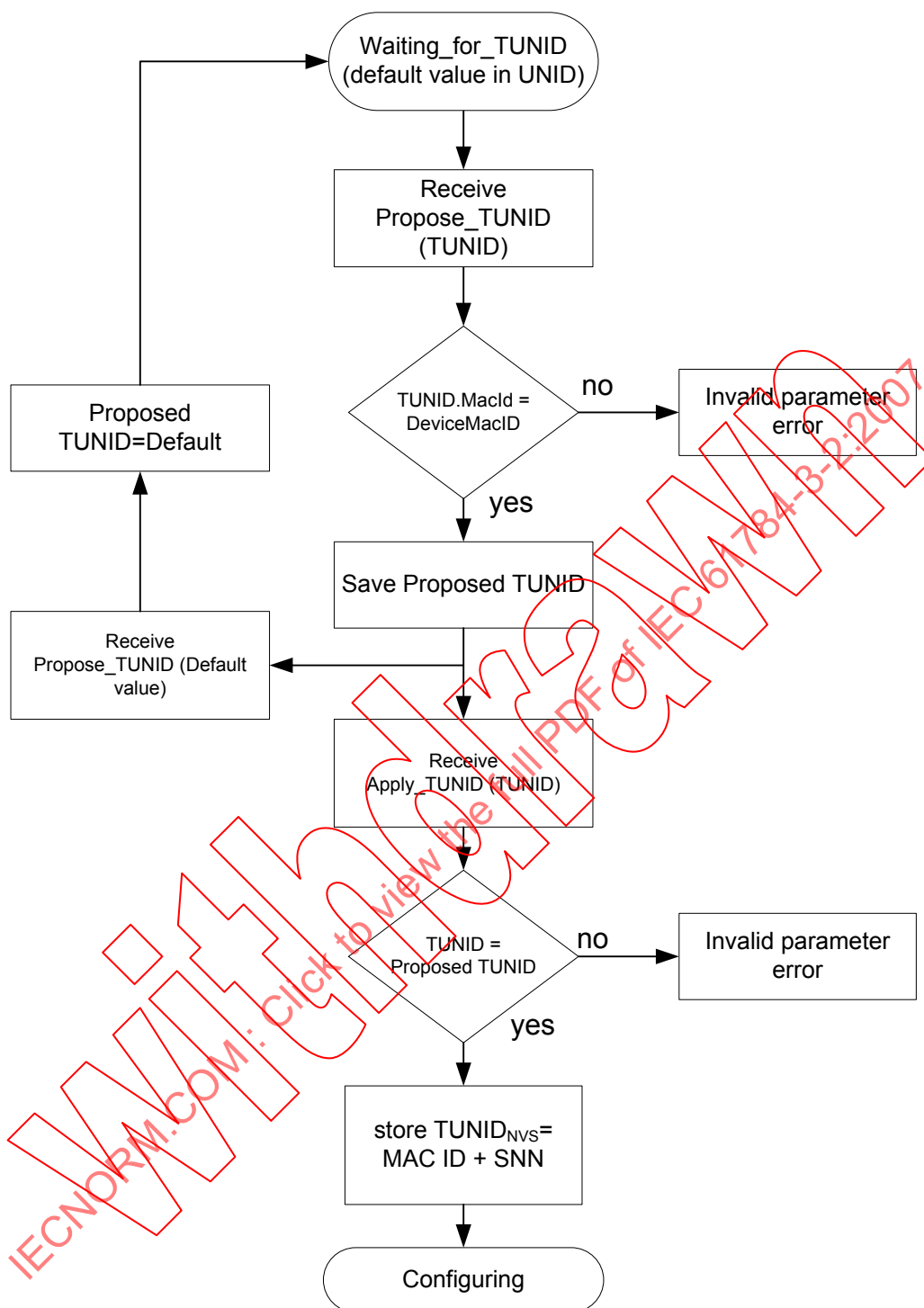
```

Apply_TUNIDHandler()
{
  IF (Device_State=Waiting_for_TUNID)
    THEN IF {(Apply_TUNID[TUNID] != all FFhex)
      AND (TUNIDProposedAttr == Apply_TUNID[TUNID])
    THEN

```

```
        Stop Flash Sequence
        Target_UNID = TUNIDProposedVar[TUNID]
//NV Memory operation
    // Set SNN attribute in DeviceNet or TCP/IP object, NV
Memory operation
        Safety_Network_Number = TUNIDProposedVar[SNN]
        TUNIDProposedVar[TUNID] = TUNID_OUT_OF_BOX_DEFAULT
        Return Success Reply
        Device_State = CONFIGURING
    ELSE
        Error Code "Invalid Parameter" is returned
ELSE
    Error Code "Object State Conflict" is returned
}
```

IECNORM.COM: Click to view the full PDF of IEC 61784-3-2:2007



Légende

Anglais	Français
Waiting_for_TUNID (default value in UNID)	Waiting_for_TUNID (valeur par défaut dans UNID)
receive ...	réception ...
default	par défaut
no	non
yes	oui
invalid parameter error	erreur de paramètre invalide
save	sauvegarde

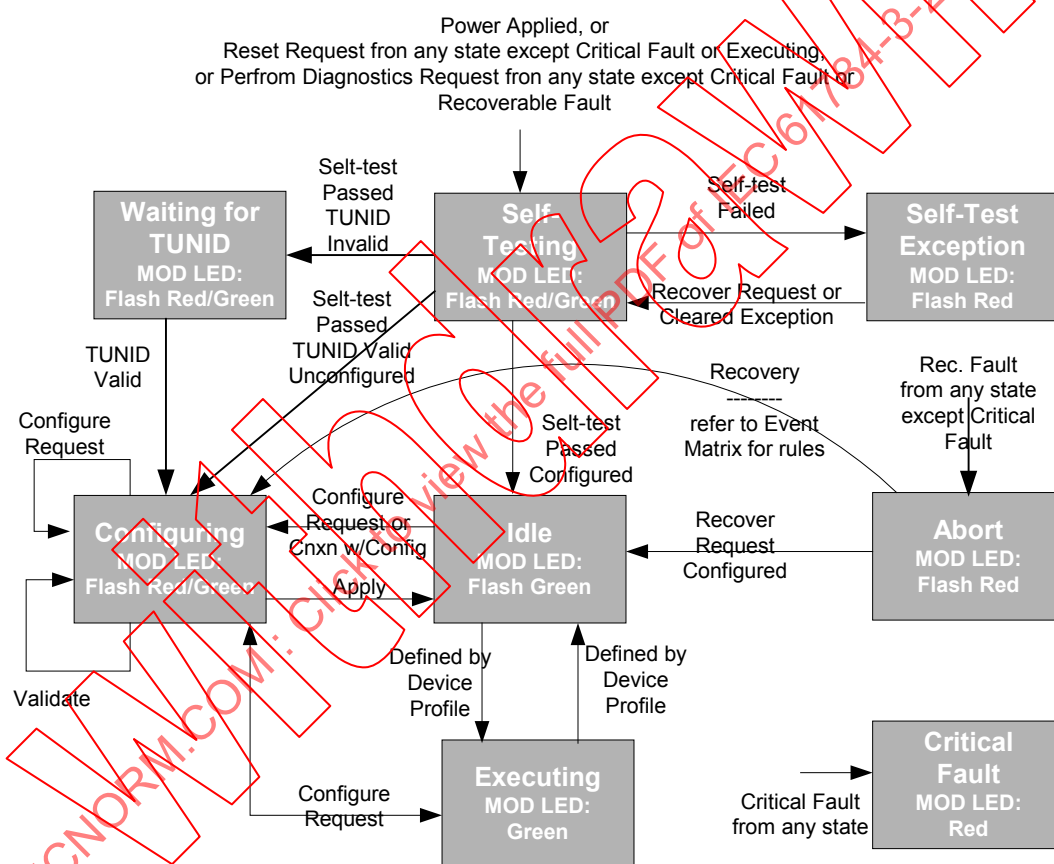
Anglais	Français
(default value)	(valeur par défaut)
store	archive
configuring	configuration

Figure 11 – Traitement UNID pendant l'état "Waiting for TUNID"

6.6.8 Comportement du Programme de contrôle de sécurité

6.6.8.1 États de l'objet Programme de contrôle de sécurité

La Figure 12 montre le diagramme d'états qui définit le comportement de l'objet Programme de contrôle de sécurité. Des informations complémentaires concernant les événements liés audit programme sont données dans le Tableau 42.



Légende

Anglais	Français
Power applied, or Reset Request from any state except Critical Fault or Executing, or Perform Diagnostic Request from any state except Critical Fault or Recoverable Fault	Alimentation appliquée ou demande de réinitialisation de tout état, sauf Panne critique ou Exécution, ou Exécution demande de diagnostic de tout état, sauf Panne critique ou Panne récupérable
self-test passed	Auto-vérification réussie
invalid	invalide
waiting for	attente de
flash red/green	Rouge/vert clignotant
self-testing	auto-vérification
self-test failed	Echec de l'auto-vérification
recover request or cleared exception	Demande de récupération ou exception supprimée

Anglais	Français
flash red	Rouge clignotant
valid	Valide
unconfigured	non configuré
configured	configuré
or	ou
recovery	récupération
refer to event matrix for rules	Se reporter à la matrice d'événements pour les règles
rec. fault from any state except critical fault	Panne récupérable de tout état, sauf panne critique
Recover request configured	Demande de configuration configurée
configuring	configuration
apply	application
idle	repos
abort	abandon
validate	validation
defined by device profile	Défini par le profil de dispositif
executing	Exécution
critical fault from any state	Panne critique de tout état
critical fault	Panne critique
red	Rouge
green	Vert

Figure 12 – Diagramme d'états du Programme de contrôle de sécurité

Tableau 42 – Événements liés au Programme de contrôle de sécurité

Etat	Description
Exécution	Le dispositif est en cours d'exécution (par exemple, est en phase de configuration complète, ne comporte pas d'erreurs et a satisfait aux critères définis par le fournisseur). Cet état, et les transitions vers ou hors de cet état, doit être spécifié davantage dans une spécification de profil de dispositif appropriée
Auto-vérification	L'instance objet existe et a été initialisée; tous les attributs ont des valeurs initiales appropriées (tel qu'indiqué dans le présent document et dans la spécification de profil de dispositif applicable). Les bits d'Etat d'exception ont été réinitialisés. Le dispositif effectue un essai spécifique au dispositif et spécifique au type de dispositif afin de déterminer s'il est qualifié pour exécuter son processus d'application
Exception d'auto-vérification	L'objet a détecté une condition d'exception au cours de l'auto-vérification. Les détails de l'exception sont archivés dans les valeurs d'attribut appropriées de l'objet Programme de contrôle de sécurité
Repos	L'objet et le dispositif ont été initialisés, ont effectué une auto-vérification satisfaisante et ont une configuration valide. De plus, le dispositif n'utilise pas les composants opérationnels de ses fonctions spécifiques. La configuration et le repos sont des états persistants protégés par des cycles d'alimentation
Abandon	L'instance objet est dans l'état Abandon; l'adoption de cet état est possible uniquement par l'occurrence d'événements à génération interne. Les conditions nécessaires à la sortie d'un état de panne récupérable sont définies dans la matrice d'Événements
Attente de TUNID	Le dispositif quitte l'état d'auto-vérification et détermine que le TUNID archivé dans la mémoire non volatile a une valeur par défaut out-of-box. Il reste dans cet état jusqu'à l'achèvement satisfaisant de la séquence de paramétrage TUNID (c'est-à-dire opération de proposition/application dans l'interface SNCT). Le dispositif peut alors être configuré
Configuration	L'objet et le dispositif ont été initialisés, ont effectué une auto-vérification satisfaisante, mais n'ont PAS de configuration valide. La configuration et le repos sont des états prudents protégés par des cycles d'alimentation
Panne critique	L'objet (et le dispositif) sont dans un état de panne à partir duquel aucune récupération ne se produit. Les services d'objet ne peuvent pas être traités. Les conditions

Etat	Description
	nécessaires à la sortie d'un état de panne critique ne relèvent pas du domaine d'application de la présente partie

La valeur de l'attribut Etat du Programme de contrôle de sécurité indique l'état général du dispositif. Elle est actualisée par rapport aux transitions d'état appropriées dans l'objet Programme de contrôle de sécurité. Les valeurs d'attribut 1 à 8 représentent des états valides. Une valeur nulle indique que l'état du Programme de contrôle de sécurité n'est pas connu; les conditions dans lesquelles une valeur nulle peut exister ne relèvent pas du domaine d'application de la présente partie.

6.6.8.2 Matrice d'événements d'état du Programme de contrôle de sécurité

Le Tableau 43 définit les états et événements applicables au Programme de contrôle de sécurité.

Tableau 43 – Matrice d'événements d'état du Programme de contrôle de sécurité

[illegible]

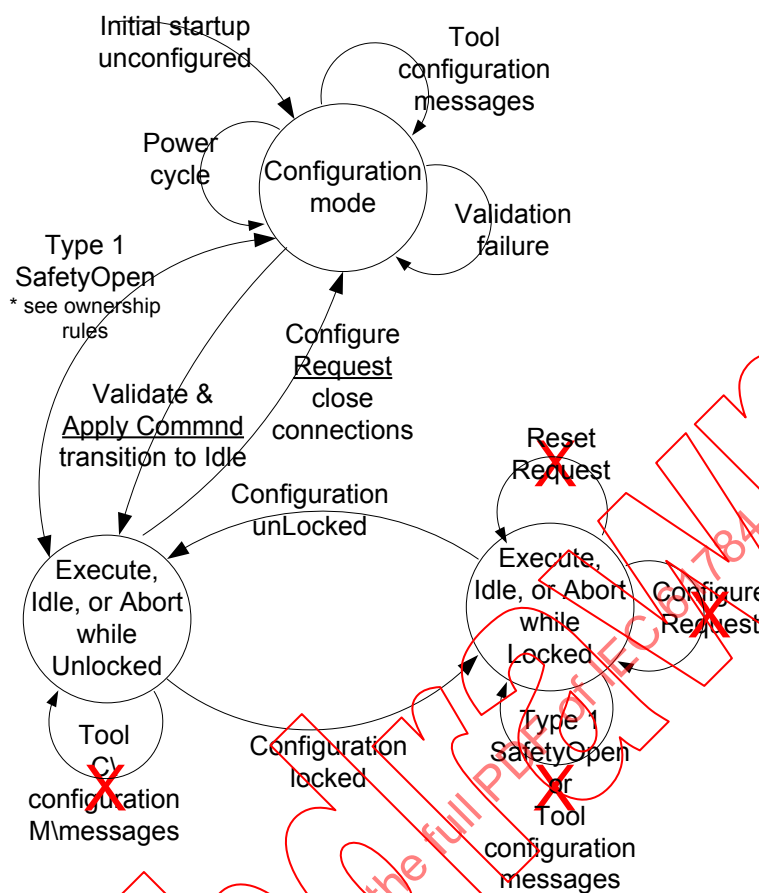
Événement	Etat							
	Attente de TUNID	Configuration (Etat de sécurité)	Repos (Etat de sécurité)	Auto-vérification (Etat de sécurité)	Exception d'auto-vérification (Etat de sécurité)	Exécution	Abandon (Etat de sécurité)	Panne critique (Etat de sécurité)
SafetyOpen de forme 1	Erreur TNS ^c	Application de la transition de configuration vers l'état Exécution si configuration Valide	Transition vers l'état Configuration, Adoption de l'état REPOS/EXECUTION fonction du profil	Erreur OSC ^b	Erreur OSC ^b	Retrait des connexions, transition vers l'état Configuration, adoption de l'état REPOS/EXECUTION fonction du profil	Erreur OSC ^c	Erreur OSC ^b
Demande de réinitialisation	Traitement de type Transition vers l'AUTO-VERIFICATION	Traitement de type Transition vers l'AUTO-VERIFICATION	Traitement de type si non verrouillé Transition vers l'AUTO-VERIFICATION	Traitement de type Transition vers l'AUTO-VERIFICATION	Traitement de type Transition vers l'AUTO-VERIFICATION	Traitement de type Si non verrouillé, Transition vers l'AUTO-VERIFICATION	Traitement de type Si non verrouillé Transition vers l'AUTO-VERIFICATION	Traitement de type Transition vers l'AUTO-VERIFICATION
Demande Abandon interne	Transition vers l'état Abandon	Transition vers l'état Abandon	Transition vers l'état Abandon	Erreur OSC ^b	Erreur OSC ^b	Transition vers l'état Abandon	Erreur AIRS ^a	Erreur OSC ^b
Demande Récupération	Erreur OSC ^b	Erreur OSC ^b	Erreur OSC ^b	Nouvelle AUTO-VERIFICATION	Transition vers l'AUTO-VERIFICATION	Erreur OSC ^b	Si configuré: Transition vers l'état REPOS --- Non configuré: Transition vers l'état Configuration --- Si défaut d'adaptation Commutateur /TUNID Erreur OSC ^b	Erreur OSC ^b
Exécution de la demande Diagnostic	Erreur OSC ^b	Erreur OSC ^b	Transition vers l'AUTO-VERIFICATION	Nouvelle AUTO-VERIFICATION	Transition vers l'AUTO-VERIFICATION	Erreur OSC ^b	Exécution de tous les essais de diagnostic	Erreur OSC ^b
Échec de la connexion de sécurité	Non prise en compte de l'événement	Non prise en compte de l'événement	Non prise en compte de l'événement	Non prise en compte de l'événement ^d	Non prise en compte de l'événement	Profil de dispositif défini	Non prise en compte de l'événement ^d	Non prise en compte de l'événement ^d
Connexion E/S de sécurité établie	Impossible	Impossible	Profil de dispositif défini	Non prise en compte de l'événement ^d	Impossible	Profil de dispositif défini	Impossible	Impossible

Événement	Etat							
	Attente de TUNID	Configuration (Etat de sécurité)	Repos (Etat de sécurité)	Auto-vérification (Etat de sécurité)	Exception d'auto-vérification (Etat de sécurité)	Exécution	Abandon (Etat de sécurité)	Panne critique (Etat de sécurité)
Connexion E/S de sécurité supprimée	Non prise en compte de l'événement ^d	Non prise en compte de l'événement ^d	Non prise en compte de l'événement ^d	Non prise en compte de l'événement ^d	Impossible	Profil de dispositif défini	Non prise en compte de l'événement ^d	Erreur OSC ^b
Changement de mode	Erreur OSC ^b	Erreur OSC ^b	Profil de dispositif défini	Erreur OSC ^b	Erreur OSC ^b	Profil de dispositif défini	Erreur OSC ^b	Erreur OSC ^b
Restauration	Erreur OSC ^b	Restauration de la configuration précédente	Erreur OSC ^b	Erreur OSC ^b	Erreur OSC ^b	Non prise en compte de l'événement ^d	Erreur OSC ^b	Erreur OSC ^b
Verrouillage/ Déverrouillage	Erreur OSC ^b	Erreur OSC ^b	Si valide, exécution et réponse	Erreur OSC ^b	Erreur OSC ^b	Si valide, exécution et réponse	Si valide, exécution et réponse	Erreur OSC ^b
Définitions des DEL applicables aux états								
DEL Etat du module	Rouge /Vert clignotant	Rouge/Vert clignotant	Vert clignotant	Rouge/Vert clignotant	Rouge clignotant	Vert fixe	Rouge clignotant	Rouge fixe
^a Erreur AIRS = Réponse d'erreur «Déjà en Mode/Etat sollicité » (Code 0B_{hex}). ^b Erreur OSC = Réponse d'erreur "Conflit d'état d'objet" (Code 0C_{hex}), dans toute la mesure du possible. ^c Erreur TNS= Réponse d'erreur "TUNID non défini", code étendu 01_{hex} ou code étendu spécifique à l'objet 80D_{hex}. ^d Non prise en compte de l'événement: Défini comme « l'événement n'a pas d'influence sur l'état », aucune réponse nécessaire.								

Le dispositif peut solliciter l'existence d'un service instance de Programme de contrôle de sécurité en interne, tel que spécifié par le fabricant. Généralement, ces demandes sont générées suite à un événement tel qu'un échec de diagnostic.

6.6.8.3 Influence du verrouillage sur le comportement du dispositif

La Figure 13 définit l'influence de la condition « verrouillée » sur la capacité de configuration d'un dispositif; les dispositifs verrouillés rejeteront toutes les tentatives de modification des données de configuration qui affectent le SCID.



Légende

Anglais	Français
initial startup unconfigured	Démarrage initial non configuré
tool configuration messages	Messages de configuration par outil
power cycle	Cycle d'alimentation
configuration mode	Mode configuration
validation failure	Echec de la validation
type 1 SafetyOpen	Safetyopen type 1
see ownership rules	Voir règles de propriété
configure request	Demande de configuration
close connections	Fermer les connexions
validate & apply commnd	Commande Validation & Application
transition to Idle	Transition vers Repos
configuration unLocked	Configuration non verrouillée
reset request	Demande Réinitialisation
execute, idle, or abort while unlocked	Exécution, repos ou abandon lorsque non verrouillé
execute, idle, or abort while locked	Exécution, repos ou abandon lorsque verrouillé
configure request	Demande de Configuration
tool C	C outil
configuration M\messages	M\messages configuration
configuration locked	Configuration verrouillée
or tool configuration messages	Ou messages de configuration par outil

Figure 13 – Relations entre configuration, essais et verrouillage

Le Tableau 44 définit l'influence de la propriété sur la capacité de configuration d'un dispositif. Il montre que la configuration des dispositifs avec propriétaire peut être uniquement de forme 1, ladite configuration étant effectuée par le point d'origine ayant l'UNID du propriétaire. Un dispositif verrouillé ne peut jamais être configuré sans tenir compte de son propriétaire.

Tableau 44 – Contrôle du propriétaire de configuration par rapport à l'état du dispositif

Règles de propriété	Mode de configuration	Mode Exécution/Repos non verrouillé	Mode Exécution/Repos verrouillé
Configuration avec propriétaire (Config. OUNID = UNID du propriétaire)	Forme 1 acceptée par le seul propriétaire	Forme 1 acceptée par le seul propriétaire	Toutes les formes 1 refusées
Configuration sans propriétaire (Config. OUNID = 0)	Forme 1 acceptée par quiconque	Forme 1 acceptée par quiconque	Toutes les formes 1 refusées
Outil SW détenu (Config. OUNID = toutes les valeurs FF _{hex})	Aucune forme 1 acceptée	Aucune forme 1 acceptée	Toutes les formes 1 refusées

6.6.8.4 Mise en correspondance des états de l'objet Programme de contrôle de sécurité et de l'objet Identité

Le Tableau 45 définit la mise en correspondance entre les états du Programme de contrôle de sécurité et les états définis pour l'objet Identité. Cette mise en correspondance des états définit l'interface entre ces trois objets. L'attribut d'état de l'objet Identité (si pris en charge) doit suivre ce tableau.

Tableau 45 – Mise en correspondance des états du Programme de contrôle de sécurité et de l'objet Identité

Programme de contrôle de sécurité	Identité
Auto-vérification	Auto-vérification du dispositif
Exception d'auto-vérification	Principalement non récupérable
Repos	En attente
Configuration	En attente
Attente de UNID	En attente
Exécution	Opérationnel
Abandon	Principalement récupérable
Panne critique	Principalement non récupérable

6.6.8.5 Mise en correspondance des événements des objets Programme de contrôle de sécurité et Identité

Le Tableau 46 décrit la mise en correspondance des événements du Programme de contrôle de sécurité définis et des événements équivalents définis pour l'objet Identité. Cette mise en correspondance des événements définit l'interface entre ces objets. Cette mise en correspondance est fournie à des fins de référence.

**Tableau 46 – Mise en correspondance des événements
de l'objet Programme de contrôle de sécurité**

Mise en correspondance des événements d'état du Programme de contrôle de sécurité avec ->	Événement d'identité équivalent
Alimentation appliquée	Alimentation appliquée
Échec de l'auto-vérification	Essais non satisfaits
Auto-vérification satisfaite	Essais satisfaits
Proposition de TUNID	Aucun événement équivalent
Application de TUNID	Aucun événement équivalent
Condition d'exception supprimée	Panne corrigée
Panne critique	Panne principale non récupérable
Demande de configuration	Désactivé
Validation de configuration	Aucun événement équivalent
Application de la configuration	Activé
SafetyOpen de forme 1	Aucun événement équivalent
Demande Réinitialisation	Réinitialisation
Abandon (Interne)	Panne principale récupérable
Demande Récupération	Panne corrigée
Exécution de la demande Diagnostic	Aucun événement équivalent
Échec de la connexion de sécurité	Désactivé
Connexion de sécurité établie	Activé
Connexion de sécurité supprimée	Aucun événement équivalent
Changement de mode	Aucun événement équivalent
Restauration	Aucun événement équivalent
Verrouillage/Déverrouillage	Aucun événement équivalent

6.6.8.6 Mise en correspondance des événements des objets Identité et Programme de contrôle de sécurité

Le Tableau 47 décrit la mise en correspondance des conditions d'événements définies des objets Identité avec les événements équivalents définis pour l'objet Programme de contrôle de sécurité. Cette mise en correspondance des événements définit les événements équivalents entre ces objets. Cette mise en correspondance est fournie uniquement à des fins de référence; l'objet Programme de contrôle de sécurité est l'objet de contrôle des dispositifs de sécurité et seuls les événements définis par le programme de contrôle doivent être mis en œuvre dans ces mêmes dispositifs.

Tableau 47 – Mise en correspondance des événements d'objet Identité

Mise en correspondance des événements d'identité avec ->	Événements de Programme de contrôle de sécurité
Alimentation appliquée	Alimentation appliquée
Essais non satisfaits	Échec de l'auto-vérification
Essais satisfaits	Auto-vérification satisfaite
Désactivé	Demande de configuration
Activé	Application de la configuration
Panne principale récupérable	Abandon (interne)
Panne principale non récupérable	Panne critique
Panne secondaire	Aucune influence
Panne corrigée	Condition d'exception supprimée
Réinitialisation	Erreur Service non pris en charge

6.7 Objet de validation de sécurité

6.7.1 Généralités

Code de classe: 3Ahex

L'Objet de validation de sécurité est conçu pour être utilisé dans les dispositifs de sécurité sur tous les réseaux FSCP 2/1. Cet objet contient les informations nécessaires à la coordination et au maintien de connexions de sécurité fiables entre les applications relatives à la sécurité client et serveur. Le rôle principal de la fonction Validation de sécurité est d'agir comme gestionnaire de transport de sécurité des multiples connexions CPF 2 de faible niveau qui ensemble constituent une connexion de sécurité complète.

Le dispositif contenant l'Objet de validation de sécurité peut créer plusieurs instances de l'Objet de validation de sécurité pour chaque point de connexion d'application relative à la sécurité (par exemple, ensemble de données d'application) pris en charge par le dispositif.

L'Objet de validation de sécurité prend en charge deux types fondamentaux de transport de sécurité: Point à point et multipoint. Chaque type de transport de sécurité comprend un ensemble défini de connexions CPF 2 de faible niveau qui doit être attribué avec l'Objet de validation de sécurité en cas d'instanciation d'une connexion de sécurité par une fonction SafetyOpen.

6.7.2 Attributs de classe

Les attributs de classe destinés au programme de contrôle de sécurité sont définis dans le Tableau 48.

Tableau 48 – Attributs de classe de l'Objet de validation de sécurité

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
1 - 7	Facultatif	Obtenir			Voir CEI 61158-5-2 et CEI 61158-6-2 pour une description des attributs de classe réservés facultatifs	
8	Requis	Obtenir	Comptage des pannes de connexion de sécurité	UINT	Compteur de diagnostics qui représente le cumul des pannes de connexion de sécurité	

Comptage des pannes de connexion de sécurité: l'attribut Comptage des pannes de connexion de sécurité de l'Objet de validation de sécurité doit représenter le cumul (avec auto-remplacement) des fréquences d'occurrence d'une panne des connexions de sécurité pour quelque motif que ce soit.

6.7.3 Attributs instance

6.7.3.1 Présentation générale des attributs instance

La plupart des valeurs d'attributs sont incorporées au programme ou obtenues à partir du processus d'établissement des connexions dynamiques. La seule valeur d'attribut définissable est le diagnostic de l'âge maximal des données (Max Data Age). Une liste complète des attributs instance est définie dans le Tableau 49.

Tableau 49 – Attributs instance de l'Objet de validation de sécurité

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	NV	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
1	Requis	Obtenir		Etat de l'Objet de validation de sécurité	USINT	Etat de la connexion de sécurité	
2	Requis	Obtenir		Type d'Objet de validation de sécurité	USINT	Type d'Objet de validation de sécurité utilisé dans cette instance	Voir 6.7.3.3
3	Facultatif	Obtenir		Multiplicateur EPI d'intervalle Ping	UINT	Nombre qui définit le Ping_Count_Interval pour une connexion particulière	16 à 1 000
4	Facultatif	Obtenir		Multiplicateur minimal de messages de coordination temporelle	STRUCT. de:		
				Taille de matrice de Multiplicateur minimal de messages de coordination temporelle	USINT	La taille de la matrice est égale au nombre max. de consommateurs pour les producteurs multipoint, et à 1 pour le consommateur point à point et multipoint	
				Multiplicateur minimal de messages de coordination temporelle	MATRICE de: UINT	Nombre minimal d'incréments de 128 μ s susceptible d'être utilisé pour un message de coordination temporelle pour le passage du consommateur au producteur	0 – 7 813
5	Facultatif	Obtenir		Multiplicateur de délai du réseau	STRUCT. de:		
				Taille de matrice de multiplicateurs de délai du réseau	USINT	La taille de la matrice est égale au nombre max. de consommateurs pour les producteurs multipoint, et à 1 pour le consommateur point à point et multipoint	
				Multiplicateur de délai du réseau	MATRICE de: UINT	Nombre maximal d'incréments de 128 μ s qu'un consommateur doit permettre à l'âge des données de sécurité d'atteindre	0 – 45 313

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	NV	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
6	Facultatif	Obtenir		Multiplicateur de temporisation	STRUCT. de:		
				Taille de matrice de multiplicateurs de temporisation	USINT	La taille de la matrice est égale au nombre max. de consommateurs pour les producteurs multipoint, et à 1 pour le consommateur point à point et multipoint	
				Multiplicateur de temporisation	MATRICE de: USINT	Détermine le nombre de messages susceptibles d'être perdus préalablement à toute déclaration d'erreur de connexion	1-4
7	Facultatif	Obtenir		Nombre maximal de consommateurs	USINT	Nombre de consommateurs maximal admis pour la connexion	1 (Point à point) 2-15 (Multipoint)
8	Facultatif ^a	Obtenir		Instance de connexion de données	UINT	ID d'instance de connexion de la connexion de données de sécurité	
9	Facultatif ^a	Obtenir		Instance de connexion de coordination	STRUCT. de:	ID d'instance de connexion de la connexion de coordination de sécurité	1-nombre max. de connexions du dispositif
				Taille de matrice d'instances de connexion de coordination	USINT	La taille de la matrice est égale au nombre max. de consommateurs pour les producteurs multipoint, et à 1 pour le consommateur point à point et multipoint	
				Instance de connexion de coordination	MATRICE de: UINT		
10	Facultatif ^a	Obtenir		Instance de connexion de correction	UINT	ID d'instance de connexion de la connexion de correction de temps	0 (Lorsque non utilisé) 1-nombre max. de connexions du dispositif
11	Facultatif ^a	Obtenir		Rattachement CCO	UINT	Id d'instance d'un CCO qui détient les paramètres d'établissement de connexion pour cette connexion. Utilisé uniquement par les points d'origine de la connexion de sécurité	0 (lorsque non utilisé ou en l'absence de rattachement)
12	Requis	Définir		Age max. des données	UINT	Diagnostic qui détient l'âge des données le plus élevé détecté dans des incréments de 128 µs. Attribut utilisé uniquement pour les consommateurs de sécurité	
13	Requis	Obtenir		Chemin de données	EPATH	Points de liaison avec les données d'application	

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	NV	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
				d'application		associées à cette connexion de sécurité	
14	Facultatif	Obtenir	V	Code d'erreur	UINT	Motif d'erreur dans cette instance	0 en l'absence d'erreur, dans les autres cas, code d'erreur spécifique à la mise en œuvre

^a Ces paramètres peuvent éventuellement être publiés. Cependant, toutes les instances d'Objet de validation doivent avoir des paramètres internes qui représentent ces variables.

6.7.3.2 Etat de l'Objet de validation de sécurité

Cet attribut définit l'état courant de l'instance d'Objet de validation de sécurité. Les transitions propres à cet attribut d'état reflètent l'état courant de l'objet; les définitions sont fournies dans le Tableau 50. Cet attribut peut être utilisé pour déterminer la condition des connexions de sécurité établies.

Tableau 50 – Attributions d'états de l'Objet de validation de sécurité

Valeur	Nom d'état	Description
0	Non attribué	Dans cet état, l'Objet de validation de sécurité n'a pas été attribué à une connexion, ou a été fermé
1	Initialisation	Dans cet état, l'Objet de validation de sécurité se situe dans le processus d'échange des informations de coordination temporelle sur la connexion. La connexion de « sécurité » n'est pas encore pleinement établie
2	Établi	Dans cet état, l'instance Objet de validation de sécurité est pleinement établie et produit/consomme des données de sécurité pour le compte de l'application relative à la sécurité
3	Connexion_Failed	Cet état est actif lorsque toutes les connexions associées à cet Objet de validation ont échoué pour quelque raison que ce soit. Dans le cas des producteurs, toutes les connexions de consommateur multipoint doivent avoir échoué. Le producteur reste dans l'état Établi tant qu'un consommateur est toujours actif.
4-255	Réservé	

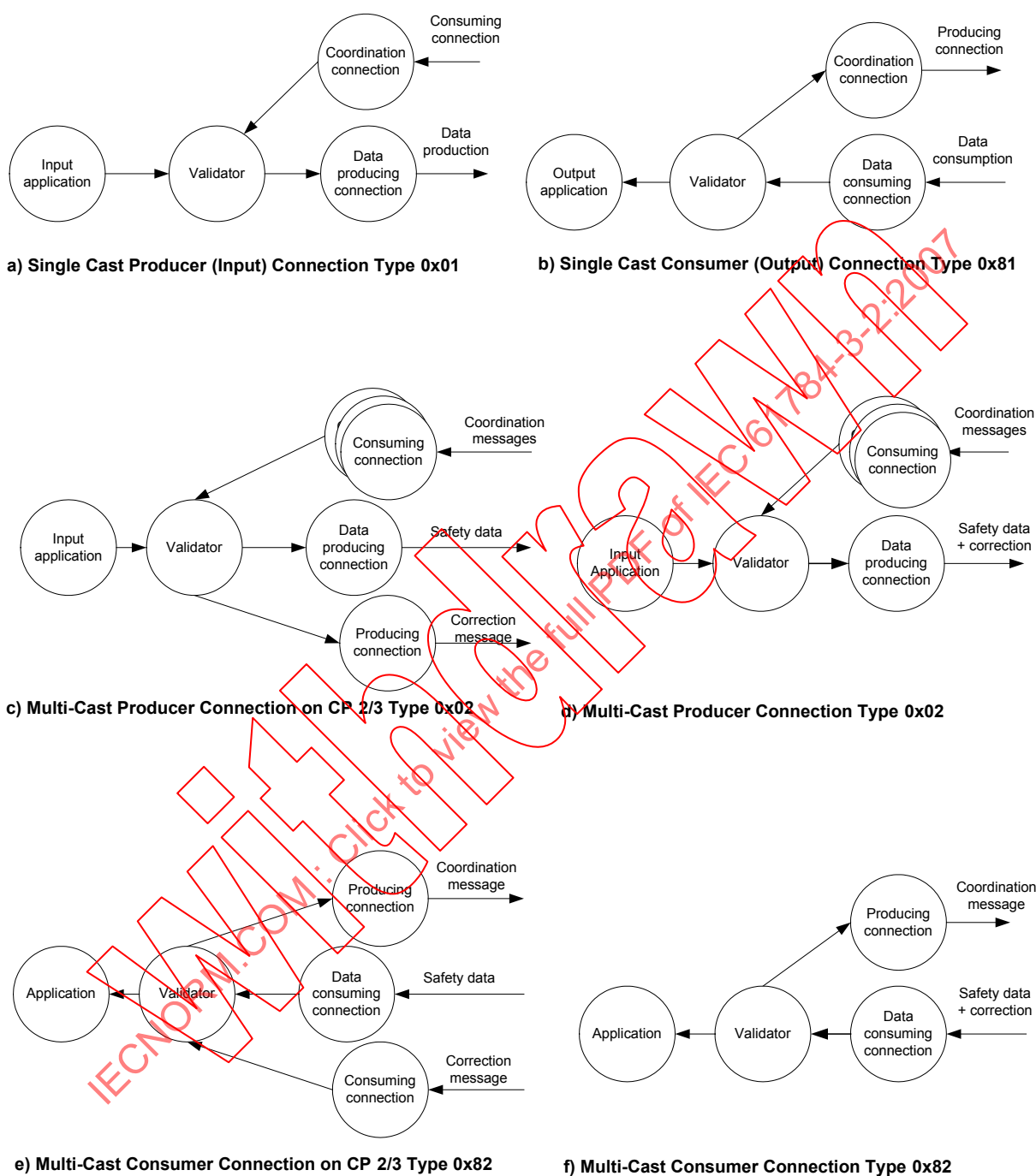
6.7.3.3 Type d'Objet de validation de sécurité

Cet attribut définit le type de transport utilisé par l'objet de validation de sécurité, dont la définition est donnée dans le Tableau 51. Les cibles doivent déduire cette valeur des paramètres de connexion de sécurité de la fonction SafetyOpen.

Tableau 51 – Type d'Objet de validation de sécurité, attributions de champs de bits

Bit 7	Bits 0 – 6
Producteur (client) = 0/ Consommateur (serveur) = 1	Type de connexion de sécurité: 0 = non attribué 1 = point à point 2 = Multipoint 3-127 = Réservé

Les transports propres à chaque type d'Objet de validation de sécurité sont illustrés à la Figure 14. Cet attribut peut être utilisé pour demander à un dispositif de déterminer le type de connexion qui a été établie.



Légende

Anglais	Français
consuming connection	Connexion de consommation
coordination connection	Connexion de coordination
producing connection	Connexion de production
input application	Application d'entrée
validator	Objet de validation
data producing connection	Connexion de production de données

Anglais	Français
data production	Production de données
output application	Application de sortie
data consuming connection	Connexion de consommation de données
data consumption	Consommation de données
a) single cast producer (input) connection Type 0x01	a) Connexion (entrée) de producteur mono-diffusion de type 0x01
b) single cast consumer (output) connection Type 0x81	b) Connexion (sortie) de consommateur mono-diffusion de type 0x81
coordination messages	Messages de coordination
safety data	Données de sécurité
correction message	Message de correction
c) multi-cast producer connection on CP 2/3 Type 0x02	c) Connexion de producteur multidestinataires sur CP 2/3 type 0x02
d) multi-cast producer connection Type 0x02	d) Connexion de producteur multidestinataires de type 0x02
e) multi-cast consumer connection on CP 2/3 Type 0x82	e) Connexion de consommateur multidestinataires sur CP 2/3 type 0x82
f) multi-cast consumer connection Type 0x82	f) Connexion de consommateur multidestinataires de type 0x82

Figure 14 – Types de connexion de sécurité

6.7.3.4 Multiplicateur EPI d'intervalle Ping

Cet attribut est utilisé comme partie intégrante de la fonction Validation de sécurité. Ce paramètre provient des paramètres du Segment de sécurité de la fonction SafetyOpen. Les règles que doivent suivre les producteurs multipoint pour l'évaluation de ce paramètre provenant de consommateurs multiples sont définies dans le Tableau 52.

6.7.3.5 Multiplicateur minimal de messages de coordination temporelle

Cet attribut est utilisé comme partie intégrante de la fonction validation de sécurité. Ce paramètre provient des paramètres du Segment de sécurité de la fonction SafetyOpen. Les règles que doivent suivre les producteurs multipoint pour l'évaluation de ce paramètre provenant de consommateurs multiples sont définies dans le Tableau 52.

Tableau 52 – Règles d'évaluation du paramètre SafetyOpen des producteurs multipoint

Paramètre de sécurité	Règle d'évaluation des producteurs	Réponse d'erreur (le cas échéant)
Multiplicateur EPI d'intervalle Ping	Doit correspondre au P.I.E.M. existant (le consommateur le définit en premier lieu)	Paramètre invalide
Multiplicateur minimal de messages Time_Coord	Accepter tout multiplicateur (peuvent être tous différents)	NA
Multiplicateur de délai du réseau	Accepter tout multiplicateur (peuvent être tous différents)	NA
Multiplicateur de temporisation	Accepter tout multiplicateur (peuvent être tous différents)	NA
Nombre maximal de consommateurs	Doit correspondre au Max_Consumer_Number existant (le consommateur le définit en premier lieu)	Paramètre invalide

6.7.3.6 Multiplicateur de délai du réseau

Cet attribut définit le temps de réaction de connexion associé à cette connexion. Ce paramètre provient des paramètres du Segment de sécurité de la fonction SafetyOpen. Les règles que doivent suivre les producteurs multipoint pour l'évaluation de ce paramètre provenant de consommateurs multiples sont définies dans le Tableau 52.

6.7.3.7 Multiplicateur de temporisation

Cet attribut définit la tolérance de l'objet de validation de sécurité pour les messages perdus préalablement à la défaillance d'une connexion et à l'application d'une mesure de sécurité. Ce paramètre provient des paramètres du Segment de sécurité de la fonction SafetyOpen. Les règles que doivent suivre les producteurs multipoint pour l'évaluation de ce paramètre provenant de consommateurs multiples sont définies dans le Tableau 52.

6.7.3.8 Nombre maximal de consommateurs

Cet attribut définit le nombre maximal de consommateurs autorisés à se connecter à un producteur multipoint (Type 82). Les consommateurs et producteurs multipoint sont tenus de connaître cette valeur pour fonctionner correctement. Ce paramètre provient des paramètres du Segment de sécurité de la fonction SafetyOpen. Les règles que doivent suivre les producteurs multipoint pour l'évaluation de ce paramètre provenant de consommateurs multiples sont définies dans le Tableau 52.

6.7.3.9 Instance de connexion de données

Cet attribut identifie l'instance de l'objet de connexion responsable de la connexion Données de sécurité. Cet attribut est attribué comme partie intégrante du processus d'établissement de la connexion.

6.7.3.10 Instance de connexion de coordination

Cet attribut identifie l'instance de l'objet de connexion responsable de la connexion Coordination temporelle. Cet attribut est attribué comme partie intégrante du processus d'établissement de la connexion.

6.7.3.11 Instance de connexion de correction

Cet attribut identifie l'instance de l'objet de connexion responsable de la connexion Correction de temps si elle est utilisée. Cet attribut est attribué comme partie intégrante du processus d'établissement de la connexion.

6.7.3.12 Rattachement CCO

Cet attribut identifie l'instance Configuration de connexion utilisée pour l'établissement de cette connexion de sécurité. Ce rattachement est nécessaire pour indiquer à la fonction CCO la fermeture de la connexion, de sorte qu'elle puisse tenter de rétablir la connexion. Cet attribut est utilisé uniquement avec les points d'origine de connexion. Cet attribut peut être attribué comme partie intégrante du processus d'établissement de connexion ou incorporé au programme au moment de la conception.

6.7.3.13 Age maximal des données

L'âge des données de paquets reçues est défini comme la différence entre l'horloge de 128 μ s du consommateur et la datation reçue au moment de la réception du paquet. L'attribut Age maximal des données est la valeur maximale de cet âge lorsque la connexion est/a été établie. La valeur peut être utilisée par des outils de diagnostic afin d'accorder le multiplicateur de délai du réseau.

L'attribut Age maximal des données doit toujours refléter la plus grande valeur atteinte par l'âge des données de connexion, sans dépasser la valeur du multiplicateur de délai.

En cas d'anomalie et de fermeture d'une connexion de sécurité, l'attribut Age maximal des données doit être réinitialisé à la valeur 0, lorsque la connexion est rétablie.

6.7.4 Services de classes

Les services de classes pris en charge par l'objet de Validation de sécurité sont définis dans le Tableau 53.

Tableau 53 – Services de classes de l'Objet de validation de sécurité

Code de service	Mise en œuvre nécessaire	Nom	Description du service
0E _{hex}	Requis	Get_Attribute__Single	Utilisé pour la lecture d'une valeur d'attribut de classe de l'Objet de validation de sécurité
4B _{hex}	Facultatif	Réinitialisation de tous les compteurs d'erreurs	Utilisé pour réinitialiser l'attribut de classe 8 dans toutes les instances

6.7.5 Services instance

6.7.5.1 Présentation générale des services instance

Les services instance pris en charge par l'objet de Validation de sécurité sont définis dans le Tableau 54.

Tableau 54 – Services instance de l'Objet de validation de sécurité

Code de service	Mise en œuvre nécessaire	Nom	Description du service
01 _{hex}	Facultatif	Get_Attribute_All	Renvoie les valeurs de tous les attributs instance
0E _{hex}	Requis	Get_Attribute__Single	Utilisé pour la lecture d'une valeur d'attribut de l'Objet de validation de sécurité
10 _{hex}	Requis	Set_Attribute_Single	Utilisé pour réinitialiser l'attribut Age maximal des données
4B _{hex}	Facultatif	Réinitialisation du code d'erreur	Utilisé pour réinitialiser l'attribut instance 14

6.7.5.2 Get_Attributes_All Response

Les données de réponse pour le service Get_Attributes_All sont définies dans le Tableau 55.

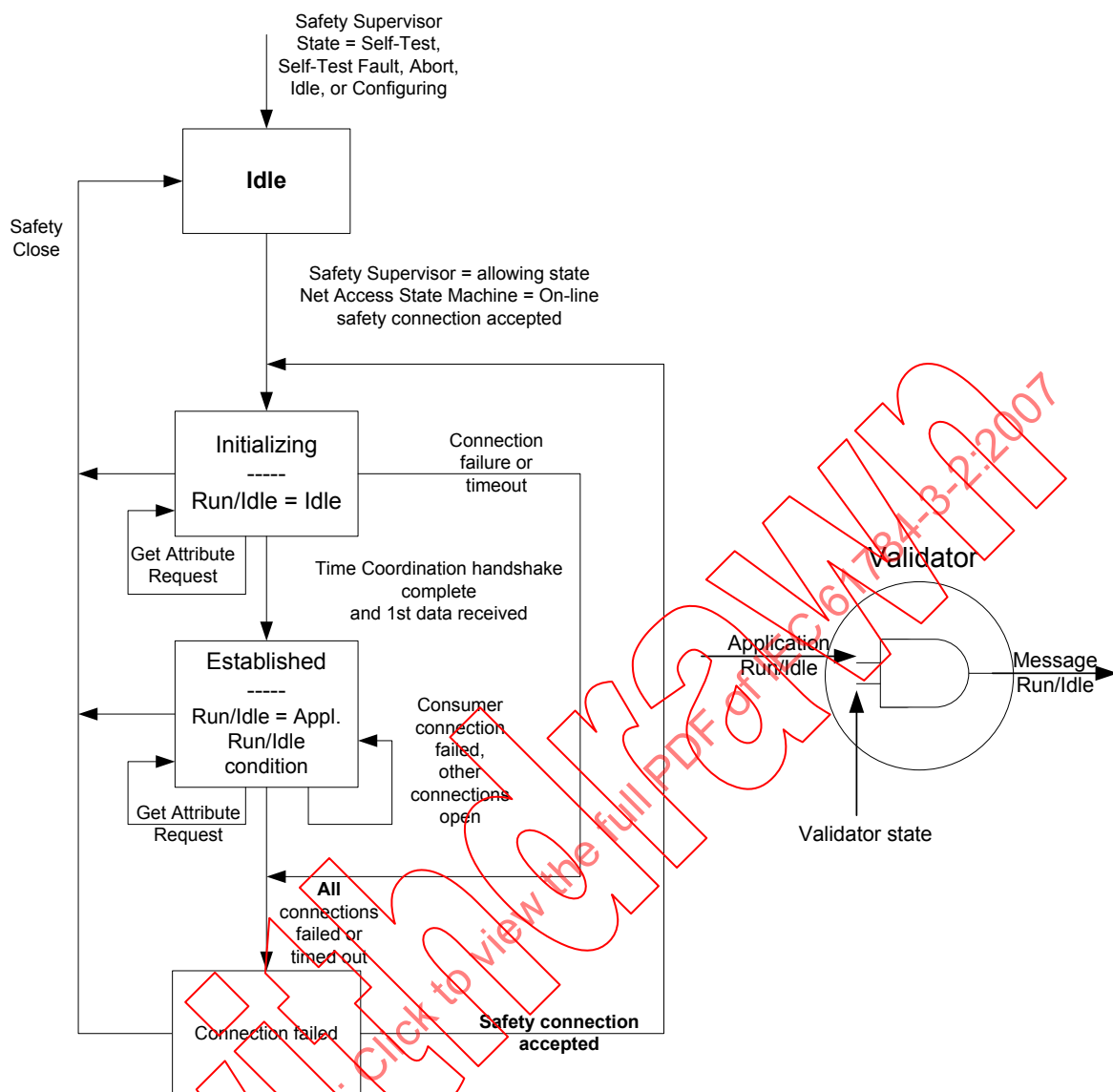
Tableau 55 – Données du service Get_Attributes_All de l'objet de Validation de sécurité

ID d'attribut	Nombre d'octets	Nom	Type de données
1	0	Etat de l'Objet de validation de sécurité	USINT
2	1	Type d'Objet de validation de sécurité	USINT
3	2,3	Multiplicateur EPI d'intervalle Ping	UINT
4	variable	Multiplicateur minimal de messages de coordination temporelle	STRUCT
5	variable	Délai du réseau	STRUCT
6	variable	Multiplicateur de temporisation	STRUCT
7	variable	Nombre maximal de consommateurs	USINT
8	variable	Instance de connexion de données	UINT = 0 si non pris en charge
9	variable	Instance de connexion de coordination	UINT = 0 si non pris en charge
10	variable	Instance de connexion de correction	STRUCT
11	variable	Rattachement CCO	UINT = 0 si non pris en charge
12	variable	Age maximal des données	UINT
13	variable	Chemin d'application	EPATH
14		Code d'erreur	UINT

6.7.6 Comportement d'objet

6.7.6.1 Diagramme de transition d'état

Le comportement de l'Objet de validation de sécurité est illustré à la Figure 15 et au Tableau 56 ci-dessous.



Légende

Anglais	Français
safety supervisor State = Self-Test, Self-Test Fault, Abort, Idle, or Configuring	Etat du programme de contrôle de sécurité = auto-vérification, panne de l'auto-vérification, Abandon, Repos ou Configuration
safety close	Fermeture de la sécurité
idle	Repos
safety supervisor = allowing state Net Access State Machine = On-line safety connection accepted	Programme de contrôle de sécurité = état d'autorisation Diagramme d'états d'accès au réseau = connexion de sécurité en ligne acceptée
initializing	Initialisation
run/idle = idle	Exécution/repos = repos
connection failure or timeout	Echec de connexion ou temporisation
get attribute request	Obtenir demande d'attribut
time coordination handshake complete and 1 st data received	Transfert de coordination temporelle complet et premières données reçues
validator	Objet de Validation
established	Etabli

Anglais	Français
run/idle = appl.	Exécution/repos = appl.
application run/idle	Exécution/repos d'application
message run/idle	Exécution/repos des messages
run/idle condition	Condition exécution/repos
consumer connection failed, other connections open	Echec de connexion du consommateur, autres connexions ouvertes
validator state	Etat de l'objet de validation
all connections failed or timed out	Echec de toutes les connexions ou temporisation
connection failed	Échec de connexion
safety connection accepted	Connexion de sécurité acceptée

Figure 15 – Diagramme de transition d'état de l'Objet de validation de sécurité

6.7.6.2 REPOS (IDLE)

L'état Repos est l'état initial saisi après la mise sous tension/l'auto-vérification. Cet état correspond à l'état normal en l'absence d'ouverture ou d'échec de connexions.

6.7.6.3 Initialisation

Pour les producteurs multipoint, cet état est effectif pour chaque connexion de consommateur traitée par le producteur. Pour les producteurs point à point et l'ensemble des consommateurs, cet état se rapporte à la connexion Objet de validation de sécurité complète.

6.7.6.4 Établi

Pour les producteurs multipoint, cet état est effectif pour chaque connexion de consommateur traitée par le producteur. Pour les producteurs point à point et l'ensemble des consommateurs, cet état se rapporte à la connexion Objet de validation de sécurité complète.

6.7.6.5 Connection_Failed

Cet état est actif lorsque toutes les connexions associées à l'Objet de validation de sécurité ont échoué. Cet état peut perdurer jusqu'à réception d'un SafetyClose ou jusqu'au rétablissement de la connexion à partir d'un point d'origine d'adaptation (c'est-à-dire que OUNID et le chemin d'application correspondent à une ressource de connexion).

6.7.6.6 Matrice d'événements d'état

Le Tableau 56 présente la matrice d'événements d'état qui définit le comportement de l'Objet de validation de sécurité. Le Tableau 57 met en correspondance les états de l'Objet de validation de sécurité et du Programme de contrôle de sécurité.

Tableau 56 – Matrice d'événements d'état de l'Objet de validation de sécurité

Événement	Etat			
	Repos	Initialisation	Exécution	Échec connexion
Alimentation appliquée et auto-vérification effectuée	Etat initial	--	--	
Programme de contrôle de sécurité = état d'autorisation, Diagramme d'états d'accès au réseau = en ligne, Nouvelle connexion de sécurité demandée	Demande de validation et Transition vers l'état Initialisation	Si producteur multipoint, Initialisation de la connexion de sécurité	Si producteur multipoint, initialisation de la connexion de sécurité	Adaptation de OUNID et du chemin d'application, Si adaptation, transition vers l'état Initialisation
Transfert de coordination temporelle achevé	--	Transition vers l'état Exécution	Si producteur multipoint Nouveau consommateur en ligne	--
Échec du consommateur dans un producteur multipoint tandis que d'autres consommateurs sont toujours en ligne	--	DEL de l'état NET rouge clignotant	DEL de l'état NET rouge clignotant	--
Échec de toutes les connexions	--	Échec de la transition vers l'état Connexion	Échec de la transition vers l'état Connexion	DEL de l'état NET rouge clignotant
Adaptation de la demande de connexion du consommateur (c'est-à-dire que OUNID et le chemin d'application correspondent à une connexion de consommateur non aboutie).	--	Transition du consommateur vers l'état Initialisation	Transition du consommateur vers l'état Initialisation	Transition vers l'état Initialisation
Réception de SafetyClose	Réponse satisfaisante, maintien dans l'état Repos	Réponse satisfaisante, Transition vers l'état Repos	Réponse satisfaisante, Transition vers l'état Repos	Réponse satisfaisante, Transition vers l'état Repos

Tableau 57 – Mise en correspondance des états des objets Programme de contrôle de sécurité et Validation de sécurité

Programme de contrôle de sécurité	Objet de validation de sécurité
Auto-vérification	Repos
Exception d'auto-vérification	Repos
Attente de TUNID	Repos
Repos	Repos ^a , Initialisation, Exécution ou Connection_Failed
Configuration	Repos
Exécution	Repos ^a , état Initialisation ou Exécution, ou Connection_Failed
Abandon	Repos
Panne critique	Repos

^a Un Objet de validation de sécurité doit accepter des connexions uniquement lorsque l'état du Programme de contrôle de sécurité peut accepter les connexions dépendant du dispositif (c'est-à-dire état Exécution uniquement, ou état Exécution/Repos). Un Objet de validation de sécurité peut être au repos tandis qu'un autre est toujours actif.

6.8 Objet de configuration de connexion

6.8.1 Généralités

Ce paragraphe définit les extensions de l'Objet de configuration de connexion, de classe F3_{hex} pour FSCP 2/1. Voir la CEI 61158-4-2 pour une définition complète.

Code de classe: F3hex

6.8.2 Extensions d'attributs de classe

Les attributs de classe destinés à être utilisés par les dispositifs FSCP 2/1 sont définis dans le Tableau 58.

Tableau 58 – Extensions d'attributs de classe d'objet de configuration de connexion

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	Nom	Type de données	Description de l'attribut
9	Conditionnel ^a	Définir	Émission de signature	UDINT	Créé et utilisé par le logiciel de configuration pour détecter les modifications apportées aux valeurs d'attributs instance
a Cet attribut est requis avec les seuls dispositifs qui n'utilisent pas l'interface SNCT de sécurité.					

6.8.3 Attributs, ajouts et extensions instance

Le Tableau 59 définit les attributs et modifications instance destinés à être utilisés par les dispositifs FSCP 2/1.

Tableau 59 – Ajouts/extensions d'attributs instance d'Objet de configuration de connexion

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	Mémoire non volatile	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
9	Requis	Définir	Mémoire non volatile	Mise en correspondance de données	STRUCT. de:		
				format_number	UINT	Ce numéro détermine le format de mise en correspondance utilisé	2 – 99 Réserve 100 – 199 Spécifiques au fournisseur. Toutes les autres valeurs sont Réservées
				Data_Map_data_size	UINT	Taille, en octets, des informations de mise en correspondance	
				Data_Map_data	Matrice d'octets	Données de format de mise en correspondance de données basées sur le numéro de format	
10	Facultatif	Définir	Mémoire non volatile	Config #2 Data2	STRUCT. de:		
				config_data_size	UINT	Longueur de config_data en octets	
				config_data	Matrice d'octets	Config #2 data	
11	Facultatif	Définir	Mémoire non volatile	ID de dispositif de remplacement	STRUCT. de:		
				vendor_id	UINT	ID du fournisseur	
				product_type	UINT	Type de dispositif	
				product_code	UINT	Code produit	
				major_rev	USINT	Révision principale	
				minor_rev	USINT	Révision secondaire	
12	Conditionnel ^a	Définir	Mémoire non volatile	Désactivation de connexion	BOOL		0 – connexion activée. 1 – connexion désactivée. En cas de réception d'un service Ouverture, cette valeur doit être réglée sur 0. En cas de réception d'un service Fermeture ou

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	Mémoire non volatile	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
							Arrêt, cette valeur doit être réglée sur 1
13	Conditionnel ^a	Définir	Mémoire non volatile	Paramètres de sécurité	STRUCT. de:		
				réservé	USINT	Réservé	Doit être égal à zéro
				réservé	USINT	Réservé	Doit être égal à zéro
				CRC de configuration	UDINT	CRC-S4 portant sur les données de configuration du dispositif cible dans les attributs 7 et 10	Cette valeur est calculée par le logiciel de configuration sur la base des données des attributs 7 et 10 après application de la configuration
				Signature de configuration	DATE ET HEURE		
				EPI de correction de temps	UDINT		
				Paramètres de connexion de correction de temps	WORD		
				UNID Cible	10 octets		UNID du dispositif cible
				Espace réservé pour l'UNID du point d'origine	10 octets	Tous les zéros	Permet à FW de remplir facilement l'OUNID dans le segment de sécurité
				Multiplicateur EPI d'intervalles Ping	UINT		
				Multiplicateur minimal de messages de coordination temporelle	UINT		
				Multiplicateur Exp. de temps réseau	UINT		
				Multiplicateur de temporisation	USINT		
				Nombre maximal de consommateurs	USINT		
14	Conditionnel ^a	Obtenir	Mémoire non volatile	CRC de paramètre de connexion	UDINT	CRC-S4 portant sur les paramètres de connexion utilisés dans un service SafetyOpen	Cette valeur est calculée par ce dispositif une seule fois, lorsque la configuration est appliquée. Elle doit pouvoir être

ID d'attribut	Mise en œuvre nécessaire	Règle d'accès	Mémoire non volatile	Nom	Type de données	Description de l'attribut	Sémantique de la valeur
							lue par le logiciel de configuration
15	Conditionnel ^a	Définir	Mémoire non volatile	Instance de configuration	UINT	Objet configuré pour ce nœud	= 0, lorsque aucun objet n'est attribué
16	Conditionnel ^a	Définir	Mémoire non volatile	Attribution d'Id	BOOL	Lorsque supprimé, le point d'origine attribue les Id de réseau pour les connexions qu'il a produites	0: attribution de l'Id pour les connexions produites (par défaut) 1: sollicitation de la cible concernant les Id produits
17	Conditionnel ^a	Obtenir		Numéro de série de connexion cible	UINT	Numéro instance de l'Objet de validation de sécurité de la cible. Renvoi du Numéro de série de connexion cible au cours de l'établissement de la connexion. Lecture par les outils de diagnostic	0 lorsque la connexion n'est pas établie

^a Cet attribut est nécessaire pour un dispositif qui prend en charge les connexions de sécurité; cet attribut est facultatif dans les autres cas.

6.8.4 Extensions ou restrictions de sémantique des attributs instance pour des raisons de sécurité

6.8.4.1 Drapeaux de connexion – (Attribut 2)

Ces drapeaux sont définis dans le Tableau 60.

Tableau 60 – Définitions des bits de drapeaux de connexion

Bits	Définition pour les dispositifs de sécurité
Bit 0:	Direction de connexion Toujours 0 = Point d'origine de la connexion
Bits 1-3:	Format de transfert en temps réel O-à-T Toujours 101 = Format de sécurité
Bits 4-6:	Format de transfert en temps réel T-à-O Toujours 101 = Format de sécurité
Bits 7-15:	Non utilisé. toujours 0
Valeur:	005A _{hex} = Point d'origine 005B _{hex} = Cible

6.8.4.2 Indice de données CS – (Attribut 4)

L'indice de données CS n'est pas utilisé, il doit être égal à FFFF FFFF_{hex}.

6.8.4.3 Multiplicateur de temporisation de connexion – (partie de l'attribut 5)

Le producteur et le consommateur utilisent le multiplicateur de temporisation de connexion afin de temporiser les 3 connexions normalisées. Le multiplicateur est utilisé par le consommateur de la connexion pour déterminer s'il convient de temporiser la connexion.

La valeur de temporisation de la connexion est définie comme suit:

RPI de Connexion $\times$ (CTM + 1) $\times$ 4

Intervalle: 0 à 7

où

0 = (0x4);

1 = (0x8);

2 = (0x12);

3 = (0x16);

4 = (0x20);

5 = (0x24);

6 = (0x28);

7 = (0x32).

Valeur: Toujours 1 (x8)

6.8.4.4 Classe de transport et déclencheur – (partie de l'attribut 5)

La valeur de la classe de transport et du déclencheur pour la sécurité est de 40_{hex} pour une connexion d'entrée ou de C0_{hex} pour une connexion de sortie.

6.8.4.5 RPI O-à-T – (partie de l'attribut 5)

Si la direction est 0 (Client / T-à-O), il s'agit du RPI de la coordination temporelle et le RPI T-à-O est le RPI de production des données.

Si la direction est 1 (Serveur / O-à-T), il s'agit du RPI de production des données et le RPI T-à-O est le RPI de coordination temporelle.

Le RPI de coordination temporelle est défini comme suit:

Intervalle: 100 μ s à 100 000 000 μ s

Multiplicateur EPI d'intervalles Ping $\times$ RPI de production de données

RPI de production de données

Intervalle: 100 μ s à 1 000 000 μ s

6.8.4.6 Paramètres de connexion O-à-T – (partie de l'attribut 5)

Les paramètres de connexion O-à-T sont définis dans le Tableau 61.

Tableau 61 – Paramètres de connexion O-à-T

Bits	Paramètre	Valeur
0-8	Taille de connexion (en octets)	Si la direction est 1 (Serveur / O-à-T), il s'agit de la taille du message de sécurité: Si la taille des données est de 1-2 octets: Taille des données + 6 Si la taille des données est de 3-250 octets: (Taille des données x 2) + 8 Si la direction est 0 (Client / T-à-O), il s'agit de la taille du message de coordination temporelle (6 octets) Intervalle: 7-510
9	Fixe/Variable	Toujours 0 (Fixe)
10-11	Priorité	Toujours 01 (Élevée)
12	Réservé	Toujours 0
13-14	Type de connexion	Toujours 10 (Point à point)
15	Propriétaire redondant	Toujours 0 (la sécurité ne prend pas en charge la redondance)

6.8.4.7 RPI T-à-O – (partie de l'attribut 5)

Si la direction est 0 (Client / T-à-O), il s'agit du RPI de la production de données et le RPI O-à-T est le RPI de coordination temporelle.

Si la direction est 1 (Serveur / O-à-T), il s'agit du RPI de coordination temporelle et le RPI O-à-T est le RPI de production de données.

Valeur: voir RPI O-à-T.

6.8.4.8 Paramètres de connexion T-à-O – (partie de l'attribut 5)

Les paramètres de connexion T-à-O sont définis dans le Tableau 62.

Tableau 62 – Paramètres de connexion T-à-O

Bits	Paramètre	Valeur
0-8	Taille de connexion (en octets)	Si la direction est 0 (Client / T-à-O), il s'agit de la taille du message de sécurité. Pour une connexion point à point, voir les paramètres de connexion O-à-T. Pour les connexions multipoint avec une taille de données de 1-2 octets: Taille des données + 12 Pour les connexions multipoint avec une taille de données de 3-248 octets: (Taille des données x 2) + 14 Les calculs de connexions multipoint doivent inclure la section de correction de temps. Ceci est requis pour toutes les connexions multipoint afin de s'assurer du calcul correct du CPCRC dans le service SafetyOpen. La valeur est déterminée par le dispositif cible. Si la direction est 1 (Serveur / O-à-T), il s'agit de la taille du message de coordination temporelle (6 octets). Intervalle: 7-510
9	Fixe/Variable	Toujours 0 (Fixe)
10-11	Priorité	Toujours 01 (Élevée)
12	Réservé	Toujours 0
13-14	Type de connexion	01 (Multipoint) 10 (Point à point) Valeur par défaut: 10 (Point à point)
15	Propriétaire redondant	Toujours 0 (La sécurité ne prend pas en charge la redondance)

6.8.4.9 Chemin de connexion – (Attribut 6)

6.8.4.9.1 Taille du chemin de connexion

Il s'agit du nombre de mots à 16 bits archivés dans le Chemin de connexion. Ce chemin doit être ajusté afin de retirer la taille des éléments de chemin du pont lors du calcul du CPCRC (c'est-à-dire que la taille utilisée pour le calcul doit être la valeur que la cible pourra observer après que le (les) pont(s) aura (auront) retiré ses (leurs) segments et ajusté cette taille de manière appropriée).

6.8.4.9.2 Chemin de connexion

Les chemins de connexion sont associés à la connexion dans l'ordre suivant: (1) Chemin de Pont, (2) Chemin de configuration, (3) Chemin de consommation et (4) Chemin de production.

Le chemin de pont (1) doit être séparé, par le point d'origine, des chemins d'application 2, 3 et 4 au moment de la constitution du service SafetyOpen et lors du calcul du CPCRC.

6.8.4.9.3 Chemin de pont

Chemin valide construit pour le parcours entre le point d'origine et la cible.

6.8.4.9.4 Chemin de configuration

Le chemin est valide si l'attribut 7 ou 8 comporte des données de configuration, ce chemin étant nul dans les autres cas. Le chemin provient dans tous les cas de l'applet cible (c'est-à-dire le fichier EDS).

6.8.4.9.5 Chemin de consommation cible

Le chemin est valide s'il s'agit de la direction 1 (Serveur / O-à-T) ou est nul dans les autres cas. Le chemin provient dans tous les cas de l'applet cible (c'est-à-dire le fichier EDS).

6.8.4.9.6 Chemin de production cible

Le chemin est valide s'il s'agit de la direction 0 (Client / T-à-O) ou est nul dans les autres cas. Le chemin provient dans tous les cas de l'applet cible (c'est-à-dire le fichier EDS).

6.8.4.10 Taille des données de configuration # 1 (partie de l'attribut 7)

Son utilisation est facultative pour la sécurité, la taille des données étant nulle en cas de non-utilisation.

6.8.4.11 Données de configuration # 1 (partie de l'attribut 7)

Son utilisation est facultative pour la sécurité, et doit par ailleurs être concaténée avec le contenu des données de Config#2.

6.8.4.12 Taille des données de configuration # 2 (partie de l'attribut 10)

Nombre d'octets stockés dans les données Config #2 (ci-dessous).

Intervalle: 1 à < Taille de configuration maximale admissible >

6.8.4.13 Données de configuration # 2 (partie de l'attribut 10)

Utilisées pour stocker les données de configuration du dispositif cible.

6.8.4.14 Mise en correspondance des données (Attribut 9)

6.8.4.14.1 Attribution de format de mise en correspondance

Le Tableau 63 définit un ensemble de formats de mise en correspondance actuellement définis dans l'intervalle normalisé. L'intervalle spécifique au fournisseur peut toujours être utilisé pour des formats de mise en correspondance spéciaux.

Tableau 63 – Formats de mise en correspondance des données

Numéro de format	Nom de format	Taille S des données de mise en correspondance de données (en octets)	Données de mise en correspondance de données	Commentaire
0	Format de scrutation ouvert	4 octets	Voir Tableau 64	Format simple qui met en correspondance les données sous forme de granularité de mots dans les tableaux d'images d'entrée et de sortie
1	Format de scrutation à index d'octet	4 octets	Voir Tableau 65	Format simple qui met en correspondance les données sous forme de granularité d'octets dans les tableaux d'images d'entrée et de sortie
2-99	Réservé			
100-199	Spécifique au fournisseur			
200 – 65 535	Réservé			

6.8.4.14.2 Utilisation du format 0 pour les scrutateurs de sécurité

Le format 0 de mise en correspondance de données contient la structure des données de mise en correspondance qui utilise le format décrit dans le type de format de mise en correspondance, défini dans le Tableau 64.

Tableau 64 – Format 0 de mise en correspondance des données

Nom	Type de données	Description	Valeur
Décalage des images de données du point d'origine à la cible	UINT	Décalage, en unités de 16 bits, dans le tableau d'images du point d'origine à la cible	Si la direction est 0 (Client / T-à-O): Toujours 0. Si la direction est 1 (Serveur / O-à-T): Décalage de mots dans le tableau des données de sortie de sécurité
Décalage des images de données de la cible au point d'origine	UINT	Décalage, en unités de 16 bits, dans le tableau d'images de la cible au point d'origine	Si la direction est 0 (Client / T-à-O): Décalage de mots dans le tableau des données d'entrée de sécurité Si la direction est 1 (Serveur / O-à-T): Toujours 0

6.8.4.14.3 Utilisation du format 1 pour les scrutateurs de sécurité

Le format 1 de mise en correspondance de données contient la structure des données de mise en correspondance qui utilise le format décrit dans le type de format de mise en correspondance, défini dans le Tableau 65.

Tableau 65 – Format 1 de mise en correspondance des données

Nom	Type de données	Description	Valeur
Décalage des images de données du point d'origine à la cible	UINT	Décalage, en unités de 8 bits, dans le tableau d'images du point d'origine à la cible	Si la direction est 0 (Client / T-à-O): Toujours 0. Si la direction est 1 (Serveur / O-à-T): Décalage d'octets dans le tableau des données de sortie de sécurité
Décalage des images de données de la cible au point d'origine	UINT	Décalage, en unités de 8 bits, dans le tableau d'images de la cible au point d'origine	Si la direction est 0 (Client / T-à-O): Décalage d'octets dans le tableau des données d'entrée de sécurité. Si la direction est 1 (Serveur / O-à-T): Toujours 0

6.8.4.15 ID de dispositif de remplacement

N'est jamais utilisé.

6.8.4.16 Désactivation de connexion (Attribut 12)

Lorsque l'attribut Activation/Désactivation de connexion dans l'objet CCO est TRUE, la connexion doit rester désactivée dans les cycles d'alimentation. Lorsque l'attribut Désactivation de connexion dans l'objet CCO est FALSE, la connexion doit rester activée dans les cycles d'alimentation.

6.8.5 Paramètres spéciaux relatifs à la sécurité – (Attribut 13)

6.8.5.1 Multiplicateur EPI d'intervalles Ping

Cet attribut est utilisé comme partie intégrante de la fonction Objet de validation de sécurité. Ce paramètre fait partie intégrante des paramètres du Segment de sécurité de la fonction

SafetyOpen. Ping_Interval_EPI_Multiplier est le nombre qui définit le Ping_Count_Interval pour la connexion de sécurité.

Les dispositifs qui mettent en œuvre les extensions de sécurité de l'objet CCO doivent effectuer un contrôle des intervalles portant sur le multiplicateur EPI d'intervalles Ping au cours d'une demande Validate_Configuration.

L'emploi du multiplicateur EPI d'intervalles Ping est défini comme suit:

Intervalle: $[\text{Max\_Consumer\_Number} \times \text{Timeout\_Multiplier.PI} + 15]$ à 1 000.

Valeur par défaut:

Si la direction est 0 (Client / T-à-O): 100

Si la direction est 1 (Serveur / O-à-T): 19

6.8.5.2 Multiplicateur minimal de messages de coordination temporelle

Cet attribut est utilisé comme partie intégrante de la fonction Objet de validation de sécurité. Ce paramètre provient des paramètres du Segment de sécurité de la fonction SafetyOpen. Time_Coord_Msg_Min_Multiplier est le nombre minimal d'incrément de 128 μs susceptible d'être utilisé pour un message de coordination temporelle pour le passage du consommateur au producteur. La valeur par défaut est 0.

Les dispositifs qui mettent en œuvre les extensions de sécurité de l'objet CCO doivent effectuer un contrôle des intervalles portant sur le Time_Coord_Msg_Min_Multiplier au cours d'une demande Validate_Configuration.

L'emploi du multiplicateur minimal de messages de coordination temporelle est défini comme suit:

Intervalle: 0 à 7 813, ce qui équivaut à 0 s à 1 s.

Valeur par défaut: 2 en l'absence de pont; 3 ou plus avec ajout d'un pontage.

6.8.5.3 Multiplicateur de délai du réseau

Cet attribut est utilisé comme partie intégrante de la fonction Objet de validation de sécurité. Ce paramètre fait partie intégrante des paramètres du Segment de sécurité de la fonction SafetyOpen. Network_Time_Expectation_Multiplier est le nombre maximal d'incrément de 128 μs qu'un consommateur doit permettre à l'âge des données de sécurité d'atteindre.

Les dispositifs qui mettent en œuvre les extensions de sécurité de l'objet CCO doivent effectuer un contrôle des intervalles portant sur le Network_Time_Expectation_Multiplier au cours d'une demande Validate_Configuration.

L'emploi du multiplicateur de délai du réseau est défini comme suit:

Intervalle 0 à 45 313, ce qui équivaut à 1 s à 5,8 s.

Valeur $[(\text{Multiplicateur de temporisation} + 2) \times \text{Data\_Production\_RPI}] / 128$

6.8.5.4 Multiplicateur de temporisation

Cet attribut est utilisé comme partie intégrante de la fonction Objet de validation de sécurité. Ce paramètre fait partie intégrante des paramètres du Segment de sécurité de la fonction SafetyOpen.

Les dispositifs qui mettent en œuvre les extensions de sécurité de l'objet CCO doivent effectuer un contrôle des intervalles portant sur le Timeout_Multiplier au cours d'une demande Validate_Configuration.

L'emploi du multiplicateur de temporisation est défini comme suit:

Intervalle 1 à 4

Valeur par défaut 2

6.8.5.5 Nombre maximal de consommateurs

Cet attribut définit le nombre maximal de consommateurs autorisés à se connecter à un producteur multipoint (Type 2). Les consommateurs et producteurs multipoint sont tenus de connaître cette valeur pour fonctionner correctement. Cet attribut est utilisé comme partie intégrante de la fonction Objet de validation de sécurité. Ce paramètre fait partie intégrante des paramètres du Segment de sécurité de la fonction SafetyOpen.

Les dispositifs qui mettent en œuvre les extensions de sécurité de l'objet CCO doivent effectuer un contrôle des intervalles portant sur le Nombre maximal de consommateurs au cours d'une demande Validate_Configuration.

L'emploi du nombre maximal de consommateurs est défini comme suit:

Intervalle: 1 à 15

Valeur par défaut: 15

Valeur: est issue de l'applet cible (c'est-à-dire le fichier EDS).

6.8.5.6 UNID de connexion cible

Cet attribut constitue l'UNID du dispositif de sécurité cible pour cette connexion. Ce paramètre fait partie intégrante des paramètres du segment de sécurité de la fonction SafetyOpen. Il est vérifié par le dispositif cible afin de confirmer que la demande n'a pas fait l'objet d'un mauvais acheminement. L'essai de la connexion par l'utilisateur constitue la seule méthode permettant de confirmer que cette valeur est correcte.

Le format UNID cible avec le CP 2/3 est défini comme suit:

OCTET 0-5: Numéro de réseau de sécurité

OCTET 6: CP 2/3 MACID

OCTET 7-9: Toujours 0 pour CP 2/3

6.8.5.7 CRC de configuration de sécurité (SCCRC)

Le SCCRC est utilisé comme signature de la configuration du dispositif et peut servir à confirmer l'intégrité de la configuration dans la durée. Le logiciel de configuration doit calculer le SCCRC sur les données de configuration dans leur ordre exact d'apparition dans le message SafetyOpen. Le même calcul est effectué par les dispositifs cibles et comparé avec cette valeur.

Le SCCRC doit être calculé sur les données de configuration par le dispositif, à chaque validation de la configuration de ce dernier (c'est-à-dire Demande Validate_Configuration au Programme de contrôle de sécurité ou connexion de sécurité de type 1).

Le SCCRC doit être archivé dans la mémoire non volatile avec les autres attributs NV à chaque application d'une configuration (c'est-à-dire application de la demande au Programme de contrôle de sécurité).

Le manuel de sécurité doit indiquer à l'utilisateur de soumettre à essai les configurations de connexion de sécurité après leur application à un point d'origine, afin de confirmer que la connexion cible fonctionne tel que prévu.

L'utilisation du SCCRC du dispositif cible est présentée dans le Tableau 66.

Tableau 66 – Valeurs SCCRC du dispositif cible

Type de connexion de sécurité	Valeur SCCRC
Type 1	SCCRC identifiant les données de configuration dans les données Config #2
Type 2a	SCCRC identifiant la configuration prévue du dispositif cible
Type 2b	0

6.8.5.8 Signature de Configuration de sécurité (Datation)

Les valeurs SCTS du dispositif cible sont présentées dans le Tableau 67.

Tableau 67 – Valeurs SCTS du dispositif cible

Type de connexion de sécurité	Valeur SCTS
Type 1	SCTS identifiant les données de configuration dans les données Config #1
Type 2a	SCTS identifiant la configuration prévue du dispositif cible
Type 2b	0

6.8.5.9 EPI de correction de temps

Ce paramètre définit l'EPI du message de correction de temps utilisé dans une connexion multipoint.

La valeur d'une connexion multipoint est la suivante:

Multiplicateur EPI d'intervalles Ping x RPI de production de données;

Dans les autres cas, cette valeur est nulle.

6.8.5.10 Paramètres de connexion de correction de temps

Définit les paramètres de connexion utilisés dans le message de correction de temps utilisé dans une connexion multipoint.

La valeur est 0000_{hex} pour une connexion non multipoint ou 2406_{hex} pour une connexion multipoint, tel que défini dans le Tableau 68.

Tableau 68 – Paramètres de connexion de correction de temps pour une connexion multipoint

Bits	Paramètre	Valeur
0-8	Taille de connexion (en octets)	Taille du message de correction de temps de 6 octets
9	Fixe/Variable	Toujours 0 (Fixe)
10-11	Priorité	Toujours 01 (Élevée)
12	Réservé	Toujours 0
13-14	Type de connexion	01 (Multipoint)
15	Propriétaire redondant	Toujours 0 (la sécurité ne prend pas en charge la redondance)

6.8.5.11 CRC des paramètres de connexion (CPCRC) – (Attribut 14)

L'attribut CRC des paramètres de connexion doit être calculé par ce dispositif d'origine à chaque validation de la configuration du point d'origine (c'est-à-dire Demande Valide Configuration au programme de contrôle de sécurité).

Le CPCRC doit être archivé dans la mémoire non volatile avec les autres attributs NV à chaque application d'une configuration (c'est-à-dire application de la demande au Programme de contrôle de sécurité).

Il convient d'exiger de l'utilisateur qu'il soumette à essai cette connexion, après qu'elle soit appliquée, afin de confirmer la configuration de connexion. A partir de là, le CPCRC est utilisé comme signature de cette configuration de connexion et peut servir à confirmer l'intégrité de la configuration dans la durée. Le calcul proprement dit est effectué avec les paramètres de connexion de la fonction SafetyOpen dans l'ordre exact de leur apparition dans le message. Le même calcul est effectué par les dispositifs cibles et comparé avec cette valeur. l'ordre exact doit donc être respecté.

6.8.5.12 Instance de configuration – (Attribut 15)

Cet attribut assure la liaison avec l'instance de l'objet attribué à cette connexion. L'attribut peut être utilisé lorsque la fonction SafetyOpen ne permet pas de transmettre les données de configuration de la cible. Une valeur de 0 (valeur par défaut) pour cet attribut indique l'absence d'attribution d'objet.

6.8.5.13 Attribution d'Id

Cet attribut booléen permet d'accorder à l'utilisateur une certaine souplesse concernant la méthode d'attribution d'identifiants par les points d'origine à l'établissement d'une connexion. Ceci est particulièrement important pour le FSCP 2/1 sur le CP 2/3. Le comportement par défaut du point d'origine consiste à fournir des identifiants pour les connexions dans lesquelles il est le producteur. Si l'utilisateur définit cet attribut, le point d'origine demande en revanche à la cible de fournir un identifiant pour ces connexions générées. Un utilisateur peut souhaiter effectuer cette opération lorsque le point d'origine comporte un grand nombre de connexions, mais ne comporte pas un nombre suffisant d'éléments pour procéder à l'application. Cet attribut s'applique uniquement aux productions non multipoint (ces dernières doivent toujours être attribuées par le producteur). Cette fonction étant considérée comme une fonction avancée, il convient de la présenter à un utilisateur avec la plus grande attention. On doit recommander à l'utilisateur qu'il convient d'utiliser cette fonction uniquement dans des scénarios d'application limités où une répétabilité de connexion peut être garantie.

6.8.6 Services spécifiques à l'objet

L'Objet Configuration de connexion fournit les services spécifiques à l'objet suivants, définis dans le Tableau 69.

Tableau 69 – Services spécifiques à l'objet Configuration de connexion

Service	Mise en œuvre nécessaire		Nom de service	Description du service
	Classe	Instance		
4B _{hex}	Conditionnel ^a	N/A	Kick-Timer (Stimulateur de temporisateur)	Stimule l'édition du temporisateur de chien de garde
4C _{hex}	Facultatif	Conditionnel ^b	Ouverture	Ouvre les connexions
4D _{hex}	Facultatif	Conditionnel ^b	Fermeture	Ferme les connexions
4E _{hex}	Facultatif	Conditionnel ^b	Arrêt	Interrompt les connexions
4F _{hex}	Conditionnel ^a	N/A	Modification	Gère l'édition de sessions

Service	Mise en œuvre nécessaire		Nom de service	Description du service
	Classe	Instance		
			démarrage	
50 _{hex}	Requis	N/A	Obtenir état	Obtenir état pour des connexions multiples
51 _{hex}	Conditionnel ^a	N/A	Modification achevée	Achève l'édition de sessions
52 _{hex}	Conditionnel ^a	N/A	Vérification des modifications	Vérifie les modifications en attente
^a Les dispositifs de sécurité ne doivent pas mettre en œuvre ces services, dans la mesure où l'outil de configuration logicielle coordonne la configuration par l'intermédiaire du Programme de contrôle de sécurité. ^b Les dispositifs de sécurité FRS357 doivent prendre en charge les services Ouverture, Fermeture et Arrêt. Ces services doivent être acceptés uniquement en cas de déverrouillage du dispositif. Si le dispositif est verrouillé, il doit répondre par une erreur « Conflit d'état du dispositif ».				

6.8.7 Extensions des services communs pour la sécurité

6.8.7.1 Get_Attribute_All (Obtenir Tous Attributs) (Code de service 01_{hex})

Les attributs ajoutés à la fin de la réponse Get_Attribute_All sont définis dans le Tableau 70. Tous les autres paramètres de ce service sont tels que définis dans l'objet CCO normalisé dans la CEI 61158-5-2 et la CEI 61158-6-2.

Tableau 70 – Données du service de réponse Get_Attributes_All (attributs ajoutés)

Nom	Type de données	Description
Paramètres de sécurité	STRUCT. de:	
	USINT	Remplissage, réglé sur zéro
	USINT	Remplissage, réglé sur zéro
	UDINT	CRC de configuration (SCCRC)
	DATE ET HEURE	Datation de configuration (SCTS)
	UDINT	EPI de correction de temps
	WORD	Paramètres de connexion de correction de temps
	10 octets	UNID Cible
	10 octets	Remplissage, réglé sur zéro
	UINT	Multiplicateur EPI d'intervalles Ping
	UINT	Multiplicateur minimal de messages de coordination temporelle
	UINT	Multiplicateur Exp. de temps réseau
	USINT	Multiplicateur de temporisation
	USINT	Nombre maximal de consommateurs
CRC de paramètre de connexion	UDINT	CRC-32 portant sur les paramètres de connexion utilisés dans un service SafetyOpen
Instance de configuration	UINT	
Drapeau d'attribution d'ID	BOOL	
Numéro de série de connexion cible	UINT	Instance d'Objet de validation de sécurité renvoyée par la cible lors de l'établissement de la connexion

6.8.7.2 Set_Attribute_All (Code de service 02_{hex})

Les attributs ajoutés à la fin de la réponse Set_Attribute_All doivent être étendus tel que défini dans le Tableau 71. Tous les autres paramètres de ce service sont tels que définis dans l'objet CCO normalisé dans la CEI 61158-5-2 et la CEI 61158-6-2.

Tableau 71 – Données du service de Demande Set_Attributes_All (attributs ajoutés)

Nom	Type de données	Description
Paramètres de sécurité	STRUCT. de:	
	USINT	Remplissage, réglé sur zéro
	USINT	Remplissage, réglé sur zéro
	UDINT	CRC de configuration (SCCRC) (valeur réglée = 0, le dispositif effectue ce calcul)
	DATE ET HEURE	Datation de configuration (SCTS) (le cas échéant, sinon = 0)
	UDINT	EPI de correction de temps
	WORD	Paramètres de connexion de correction de temps
	10octets	UNID Cible
	10octets	Remplissage, réglé sur zéro
	UINT	Multiplicateur EPI d'intervalles Ping
	UINT	Multiplicateur minimal de messages de coordination temporelle
	UINT	Multiplicateur Exp. de temps réseau
	USINT	Multiplicateur de temporisation
	USINT	Nombre maximal de consommateurs
Instance de configuration	UINT	
Drapeau d'attribution d'ID	BOOL	

6.8.7.3 Restauration (Code de service 15_{hex})

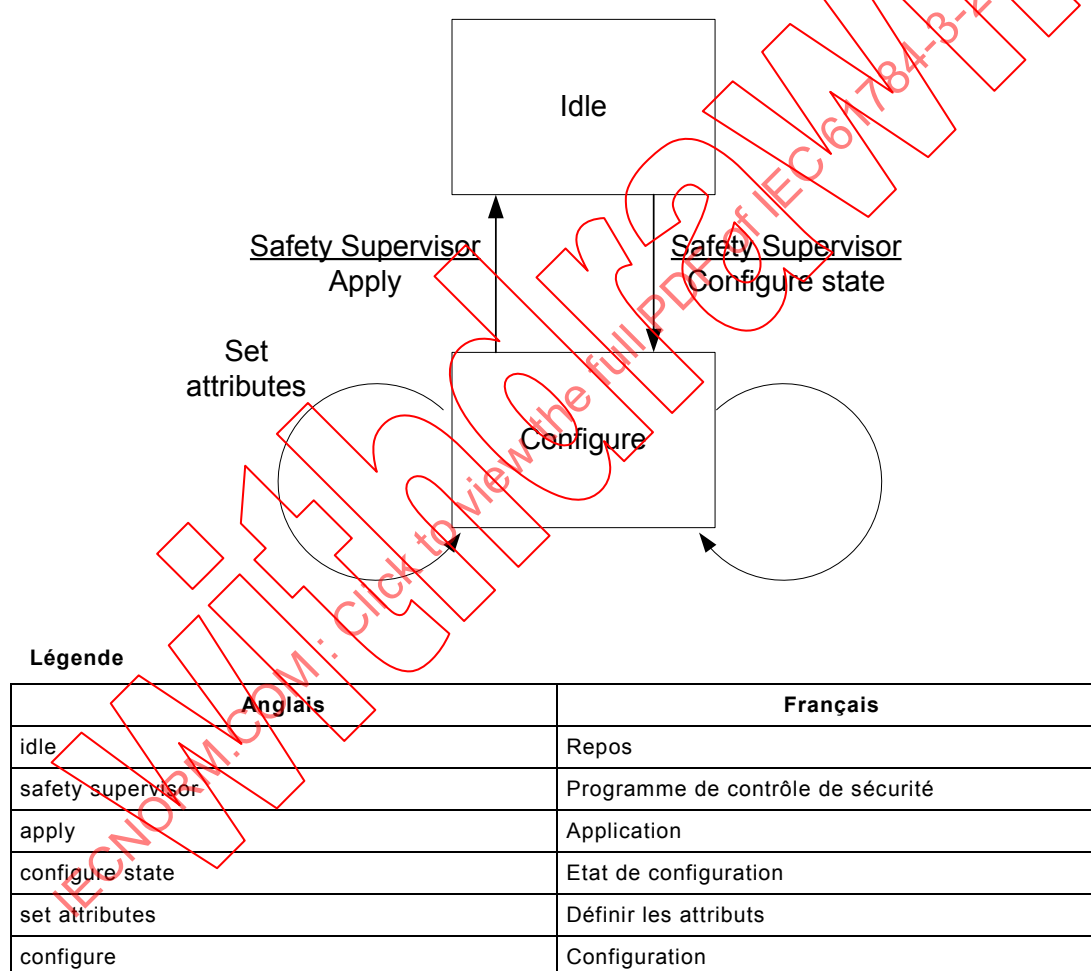
Les dispositifs de sécurité ne doivent pas mettre en œuvre le service Restauration au niveau de la classe ou de l'instance, dans la mesure où l'outil de configuration coordonne la configuration par l'intermédiaire du Programme de contrôle de sécurité.

6.8.8 Comportement d'objet

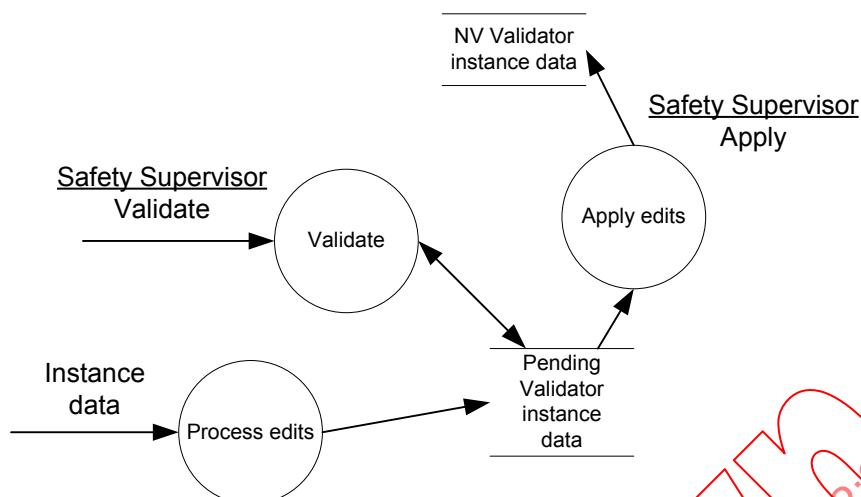
Le comportement de l'objet Configuration de connexion pour les dispositifs de sécurité doit être contrôlé par le Programme de contrôle de sécurité et respecter le comportement défini dans le Tableau 72, la Figure 16 et la Figure 17.

Tableau 72 – Mise en correspondance des états des objets Programme de contrôle de sécurité et CCO

Programme de contrôle de sécurité	Configuration de connexion
Auto-vérification	Éditions non admises
Exception d'auto-vérification	Éditions non admises
Repos	Éditions non admises
Configuration	Éditions admises
Exécution	Éditions non admises
Abandon	Éditions non admises
Panne critique	Éditions non admises

**Figure 16 – Diagramme d'états de l'objet Configuration de connexion**

Le flux de données applicable à l'objet Configuration de connexion dans l'état de configuration est défini à la Figure 17.



Légende

Anglais	Français
NV validator instance data	Données d'instance de l'objet de validation NV
safety supervisor	Programme de contrôle de sécurité
apply	Application
validate	Validation
apply edits	Application des éditions
instance data	Données d'instance
process edits	Traitement des éditions
pending validator instance data	Données d'instance de l'objet de validation en attente

Figure 17 – Flux de données de l'objet Configuration de connexion

7 Protocole de couche de communication de sécurité

7.1 Format PDU de sécurité

7.1.1 Codage PDU de sécurité

7.1.1.1 Règles générales de codage

Le protocole de sécurité définit cinq formats de codage de section utilisés pour tous les messages de sécurité. Les cinq types de formats de codage de section sont les suivants:

- section de données à 1 ou 2 octets (voir 7.1.1.4),
- section de données de 3 à 250 octets (voir 7.1.1.5),
- section de datation (voir 7.1.1.6),
- section de coordination temporelle (voir 7.1.1.7),
- section de correction de temps (voir 7.1.1.7).

Ces codages de section sont utilisés dans les six types de formats de connexion. Ces six types de formats de connexion sont les suivants:

- point à point à 1 ou 2 octets (voir 7.1.1.11),
- multipoint à 1 ou 2 octets format 1 (voir 7.1.1.12),
- multipoint à 1 ou 2 octets format 2 (voir 7.1.1.13),
- multipoint de 3 à 248 octets format 1 (voir 7.1.1.14),

- multipoint de 3 à 248 octets format 1 (voir 7.1.1.15),
- multipoint de 3 à 248 octets format 2 (voir 7.1.1.16).

Trois types de messages permettent de transmettre les six types de formats de connexion. Ces types de messages sont les suivants:

- Message de données,
- Message de coordination temporelle,
- Message de correction de temps.

Le Tableau 73 définit les sections utilisées dans les divers types de messages pour les six types de format de connexion différents:

Tableau 73 – Sections de connexion et formats PDU

Format de connexion de sécurité			Types de messages					
			Message de données			Message de coordination temporelle	Message de correction de temps	
Taille des données	Type de connexion de sécurité	Type de saut de réseau	Section de données à 1 ou 2 octets	Section de données de 3 à 250 octets	Section de datation	Section de correction de temps	Section de coordination temporelle	Section de correction de temps
1 ou 2 octets	Point à point	Tous	X		X		X	
	Multipoint	CP 2/3	X		X		X	X
		CP 2/2	X		X	X	X	
3 à 248 octets	Point à point	Tous		X	X		X	
	Multipoint	CP 2/3		X	X		X	X
		CP 2/2		X	X	X	X	

7.1.1.2 Octet Mode

L'octet Mode est décrit en dehors des définitions de sections, étant donné qu'il se trouve à la fois dans les sections de données et dans la section de datation. La disposition de l'octet Mode est définie à la Figure 18. Les bits 0 à 4 de l'octet Mode font partie intégrante de la section de datation, tandis que les bits 5 à 7 font partie intégrante des sections de données. Les variables utilisées dans l'octet Mode, les descriptions succinctes et les références visant à compléter les descriptions sont données dans le Tableau 74.

Octet Mode						
7	6	5	4	3	2	1 0
Run_Idle	TBD_2_Bit	TBD_Bit	N_Run_Idle	TBD_2_Bit_Copy	N_TBD_Bit	Ping_Count
N_Run_Idle ^a	N_TBD_2_Bit ^a	N_TBD_Bit ^a				

Légende

	= Couvert par le CRC de données réelles
	= Couvert par le CRC de données complémentaires
	= Couvert par le CRC de la section de datation

^a Les valeurs ne sont pas transmises dans le paquet de données, elles sont déduites par le consommateur.

Figure 18 – Format de l'octet Mode

Tableau 74 – Variables de l'octet Mode

Nom	Description	Référence
Run_Idle	Run_Idle doit être utilisée pour indiquer la convivialité des données tel que déterminé par l'application relative à la sécurité du producteur	7.6.1.1
N_Run_Idle	Complément de Run_Idle	7.6.1.2
TBD_2_Bit	Réservé pour une utilisation future	7.6.1.3
TBD_2_Bit Copy	Copie du TBD_2_Bit	7.6.1.4
TBD_Bit	Réservé pour une utilisation future. Le TBD_Bit doit être réglé sur 0 par l'Objet de validation de sécurité	7.6.1.6
N_TBD_Bit	Complément du TBD_Bit	7.6.1.7
Ping_Count	Le comptage de demandes Ping permet de déterminer le moment auquel un consommateur doit envoyer un message de coordination temporelle	7.6.1.5

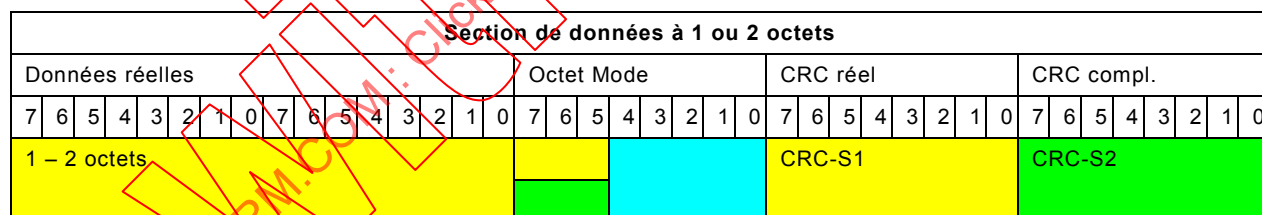
7.1.1.3 Traitement CRC de l'octet Mode

Le traitement de l'octet Mode dans les CRC doit être effectué comme suit:

- Logique de l'octet Mode de type CRC préalablement à l'inclusion dans le CRC;
- CRC des données réelles Octet Mode AND $E0_{hex}$;
- CRC des données complémentaires (Octet Mode XOR FF_{hex}) AND $E0_{hex}$;
- CRC de section de datation Octet Mode AND $1F_{hex}$.

7.1.1.4 Section de données à 1 ou 2 octets

La section de données à 1 ou 2 octets doit comporter le ou les octets des données réelles, l'octet Mode, le CRC réel et le CRC complémentaire. La disposition de la section de données à 1 ou 2 octets est définie à la Figure 19.



Légende

- = Couvert par le CRC de données réelles
- = Couvert par le CRC de données complémentaires
- = Couvert par le CRC de la section de datation

Figure 19 – Section de données à 1 ou 2 octets

Le calcul du CRC des données réelles à 1 ou 2 octets doit inclure:

- L'identifiant du producteur (PID) (par CRC-S1);
- L' (Octet Mode) AND ($E0_{hex}$) (par CRC-S1);
- Le ou les octets des données réelles (par CRC-S1).

Le calcul du CRC des données complémentaires à 1 ou 2 octets doit inclure:

- L'identifiant du producteur (PID) (par CRC-S1);
- L' (Octet Mode XOR FF_{hex}) AND ($E0_{hex}$) (par CRC-S2);

- Les octets des données réelles XOR FF_{hex} (par CRC-S2).

NOTE Le PID utilise le CRC-S1 pour le calcul du CRC réel et complémentaire.

7.1.1.5 Section de données de 3 à 250 octets

La section de données de 3 à 250 octets doit comporter les octets des données réelles, l'octet Mode, le CRC réel, les octets des données complémentaires et le CRC complémentaire. La disposition de la section de données à 3 ou 250 octets est définie à la Figure 20.

Section de données de 3 à 250 octets				
Données réelles	Octet Mode		CRC réel	Données complémentaires
3 – 250 octets			CRC-S3	3-250 octets
				CRC-S3

Légende

- = Couvert par le CRC de données réelles
- = Couvert par le CRC de données complémentaires
- = Couvert par le CRC de la section de datation

Figure 20 – Format de section de données de 3 à 250 octets

Le calcul du CRC des données réelles de 3 à 250 octets doit inclure:

- L'identifiant du producteur (PID) (par CRC-S3);
- L' (Octet Mode) AND (E0_{hex}) (par CRC-S3);
- Le ou les octets des données réelles (par CRC-S3).

Le calcul du CRC des données complémentaires de 3 à 250 octets doit inclure:

- L'identifiant du producteur (PID) (par CRC-S3);
- L' (Octet Mode XOR FF_{hex}) AND (E0_{hex}) (par CRC-S3);
- Les octets des données réelles XOR FF_{hex} (par CRC-S3).

7.1.1.6 Section de datation

La section de Datation se compose de l'octet Mode, de Time_Stamp et du CRC de datation. La disposition de la section de Datation est définie à la Figure 21.

Les descriptions des variables de la section de Datation sont définies dans le Tableau 75.

Section de Datation																											
7	6	5	4	3	2	1	0					7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Octet Mode												Datation										CRC-S1					

Légende

- = Couvert par le CRC de données réelles
- = Couvert par le CRC de données complémentaires
- = Couvert par le RC de la section de Datation

Figure 21 – Format de section de datation

Tableau 75 – Variables de datation

Nom	Description	Référence
Data_Time_Stamp	Point à point -- Data_Time_Stamp représente la datation du prélèvement des données pour production, par rapport à l'horloge du consommateur. Multipoint -- Data_Time_Stamp représente la datation du prélèvement des données pour production, par rapport à l'horloge du producteur.	7.6.2

Le calcul du CRC de datation doit inclure:

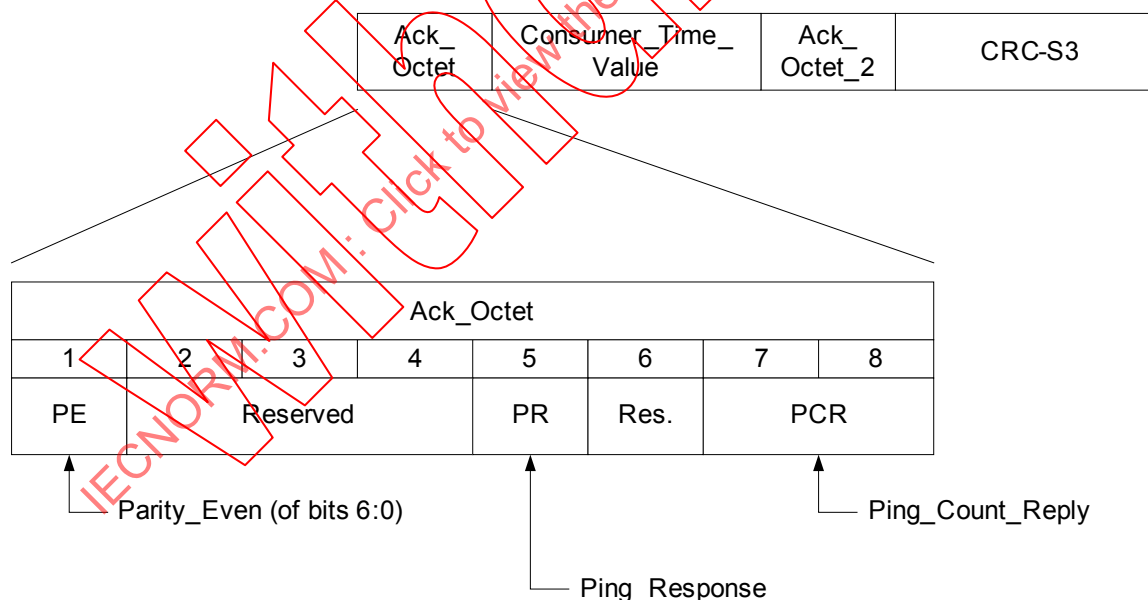
- L'octet Identifiant du producteur (PID) (par CRC-S1);
- L' (Octet Mode) AND (1F_{hex}) (par CRC-S1);
- Le ou les octets de datation (par CRC-S1).

7.1.1.7 Section de coordination temporelle

La section de coordination temporelle doit être transmise au producteur par le consommateur en réponse à un changement dans le comptage de demandes Ping.

La section de coordination temporelle est utilisée comme Message de coordination temporelle pour toutes les demandes de coordination temporelle point à point et multipoint.

Le codage applicable à la section de coordination temporelle est défini à la Figure 22. Les variables utilisées dans la section de coordination temporelle, les descriptions succinctes et les références visant à compléter les descriptions sont données dans le Tableau 76.



Légende

Anglais	Français
reserved	Réservé
of bits	des bits

Figure 22 – Codage du message de coordination temporelle

Tableau 76 – Variables du message de coordination temporelle

Nom	Description	Référence
Consumer_Time_Value	Consumer_Time_Value dans le message de coordination temporelle doit marquer (le plus précisément possible) l'heure à laquelle le consommateur transmet un message de réponse aux demandes Ping	7.6.3.2
Ack_Octet	Se compose de:	
Parity_Even	Le bit Parity_Even dans le message de coordination temporelle doit constituer la parité paire des bits 0 à 6 du message	
Ping_Response_Bit	Le Ping_Response_Bit dans le message de coordination temporelle doit indiquer la transmission d'une réponse aux demandes Ping	7.6.3.1
Ping_Count_Reply	Le Ping_Count_Reply dans le message de coordination temporelle doit indiquer la demande de Ping à laquelle répond le consommateur	7.6.3.3
Réservé	Réservé, réglé sur zéro	
CRC-S3	CRC couvrant les données contenues dans le message de coordination temporelle	

La règle suivante doit être utilisée pour le calcul du bit de parité dans le message de coordination temporelle:

Bit 7 de parité paire de forme Ack_Octet sur le Ack_Octet.

La règle suivante doit être utilisée pour le paramétrage des bits dans le Ack_Octet_2 du message de Coordination temporelle:

$Ack\_Octet\_2 = ((Ack\_Octet \text{ XOR } FF_{hex}) \text{ AND } 55_{hex}) \text{ OR } (Ack\_Octet \text{ AND } AA_{hex})$.

7.1.1.8 Calcul du CRC de coordination temporelle

Le calcul du CRC de coordination temporelle doit inclure:

- L'identifiant de consommateur (PID) (par CRC-S3) ;
- L'octet Ack (par CRC-S3);
- La Consumer_Time_Value (par CRC-S3).

Ack_Octet_2 dans le message de Coordination temporelle ne doit pas être inclus dans le CRC de sécurité.

7.1.1.9 Section de correction de temps

Le message de correction de temps doit être transmis par un producteur multipoint à chaque consommateur multipoint, pour chaque intervalle Ping. Le message de correction de temps transmet les informations spécifiques au consommateur, à ce même consommateur à un débit inférieur à celui des données produites. Le codage du message de correction de temps est illustré à la Figure 23. Les variables utilisées dans la section de correction de temps sont définies dans le Tableau 77.

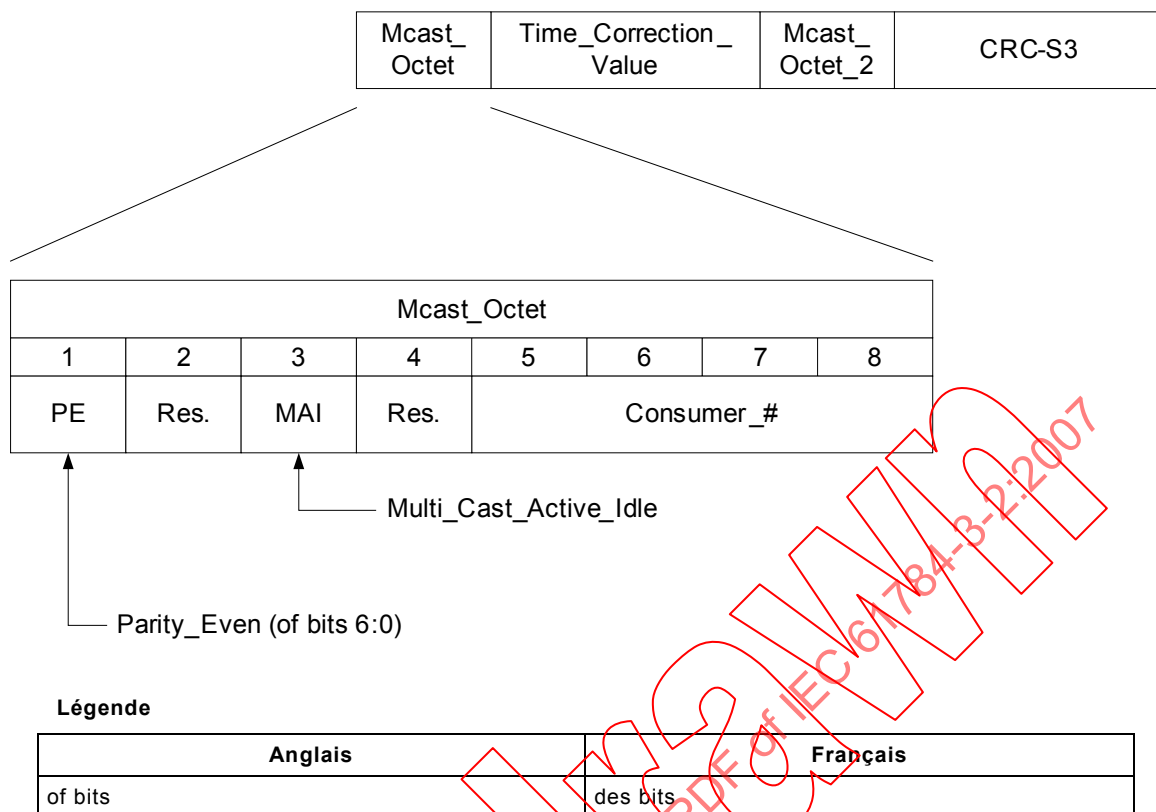


Figure 23 – Codage du message de correction de temps

Tableau 77 – Variables du message de correction de temps

Nom	Description	Référence
Consumer_Time_Correction_Value	Consumer_Time_Correction_Value dans le message de correction de temps doit être utilisée pour corriger la datation des données et évaluer l'heure par rapport à l'horloge des consommateurs multipoint	7.6.4.2
MCast_Octet	Se compose de:	
Parity_Even	Le bit Parity_Even dans le message de correction de temps doit constituer la parité paire des bits 0 à 6 du message	
Multi_Cast_Active_Idle	Multi_Cast_Active_Idle dans le message de correction de temps doit indiquer "Repos" si la transmission précède la réponse de ce consommateur avec un message de coordination temporelle valide	7.6.4.3
Consumer_#	Le Consumer_# dans le message de correction de temps doit indiquer le numéro de consommateur auquel s'adresse ce message	7.6.4.1
Réservé	Réservé, réglé sur zéro	7.6.4.4
CRC-S3	CRC couvrant les données contenues dans le message de coordination temporelle	

La règle suivante doit être utilisée pour le calcul du bit de parité dans le message de correction de temps:

Bit 7 de la parité paire de forme MCast_Octet dans le MCast_Octet.

La règle suivante doit être utilisée pour le paramétrage des bits dans le Mcast_Octet_2 du message de correction de temps:

$\text{MCast\_Octet\_2} = ((\text{MCast\_Octet} \text{ XOR } \text{FF}_{\text{hex}}) \text{ AND } 55_{\text{hex}}) \text{ OR } (\text{MCast\_Octet} \text{ AND } \text{AA}_{\text{hex}}).$

7.1.1.10 Calcul du CRC de correction de temps

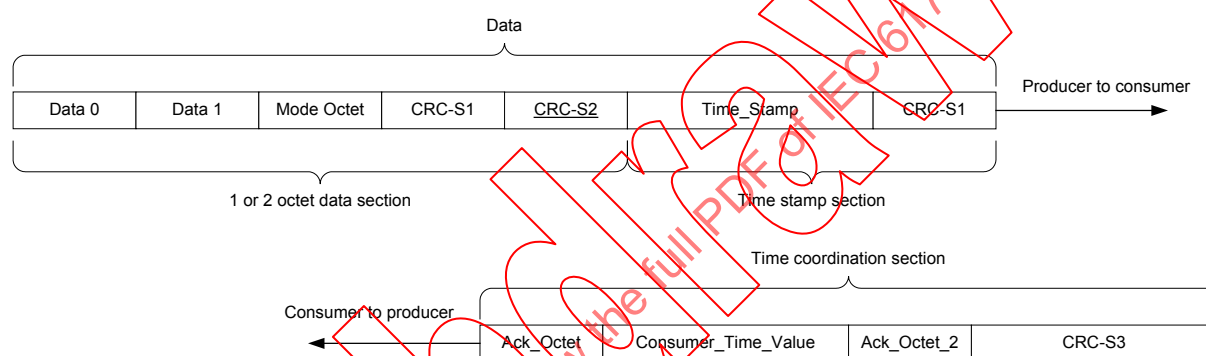
Le calcul du CRC de correction de temps doit inclure:

- L'identifiant du producteur (PID) (par CRC-S3);
- Mcast_Octet (par CRC-S3);
- Time_Correction_Value (par CRC-S3).

MCast_Octet_2 du message de correction de temps ne doit pas être inclus dans le CRC de sécurité.

7.1.1.11 Format de connexion de sécurité point à point à 1 ou 2 octets

La Figure 24 définit le codage du format de connexion de sécurité point à point à 1 ou 2 octets.



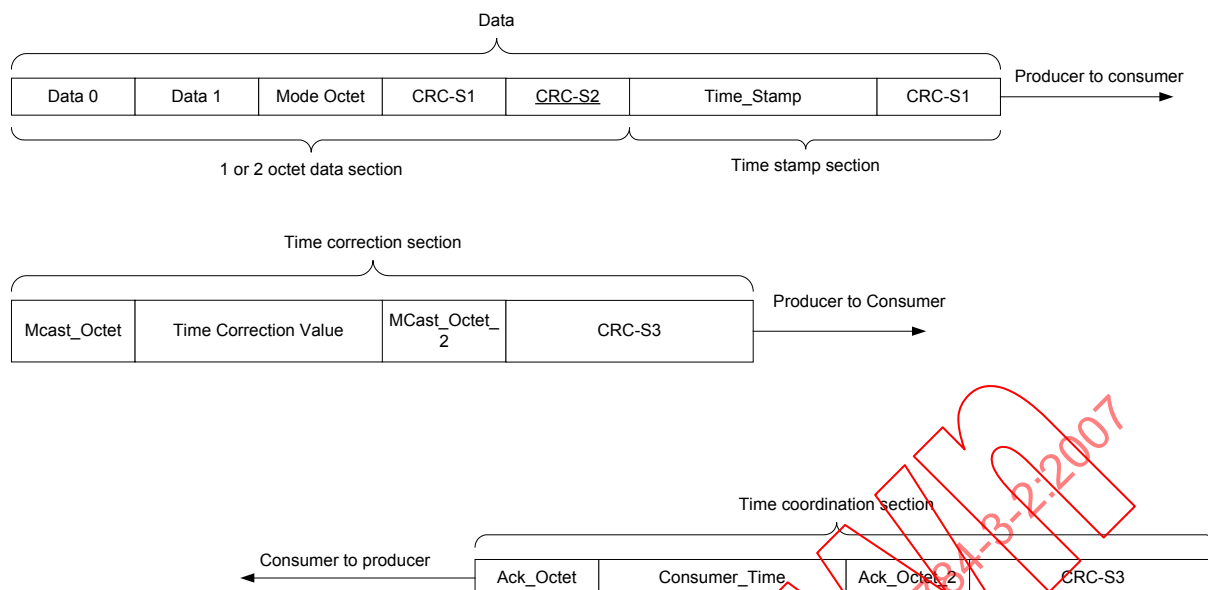
Légende

Anglais	Français
data	Données
mode octet	Octet Mode
producer to consumer	Producteur vers consommateur
1 or 2 octet data section	Section de données à 1 ou 2 octets
time stamp section	Section de datation
time coordination section	Section de coordination temporelle
consumer to producer	Consommateur vers producteur

Figure 24 – Codage PDU point à point à 1 ou 2 octets

7.1.1.12 Format de connexion de format 1 multipoint à 1 ou 2 octets

La Figure 25 définit le codage du format de connexion de Format 1 à 1 ou 2 octets.



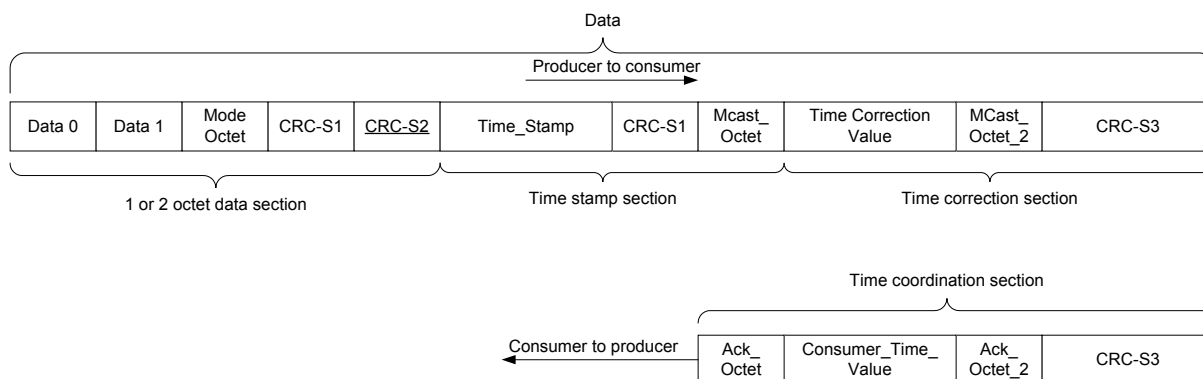
Légende

Anglais	Français
data	Données
mode octet	Octet Mode
producer to consumer	Producteur vers consommateur
1 or 2 octet data section	Section de données à 1 ou 2 octets
time stamp section	Section de datation
time correction section	Section de correction de temps
time correction value	Valeur de correction de temps
producer to consumer	Producteur vers consommateur
time coordination section	Section de coordination temporelle
consumer to producer	Consommateur vers producteur

Figure 25 – Codage PDU multipoint à 1 ou 2 octets

7.1.1.13 Format de connexion de sécurité de format 2 multipoint à 1 ou 2 octets

La Figure 26 définit le codage du format de connexion de sécurité de format 2 multipoint à 1 ou 2 octets.



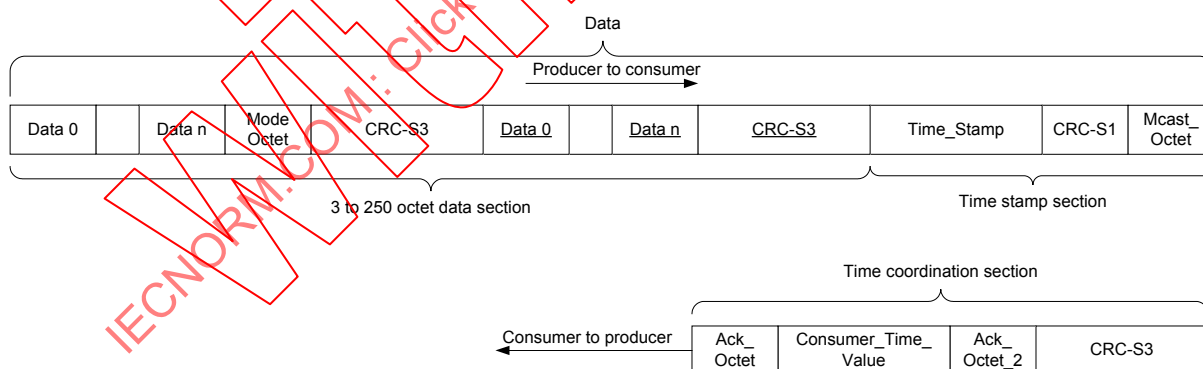
Légende

Anglais	Français
data	Données
producer to consumer	Producteur vers consommateur
mode octet	Octet Mode
time correction value	Valeur de correction de temps
1 or 2 octet data section	Section de données à 1 ou 2 octets
time stamp section	Section de datation
time correction section	Section de correction de temps
time coordination section	Section de coordination temporelle
consumer to producer	Consommateur vers producteur

Figure 26 – Format de connexion de sécurité de format 2 multipoint à 1 ou 2 octets

7.1.1.14 Format de connexion de sécurité point à point de 3 à 250 octets

La Figure 27 définit le codage du format de connexion de sécurité point à point de 3 à 250 octets.



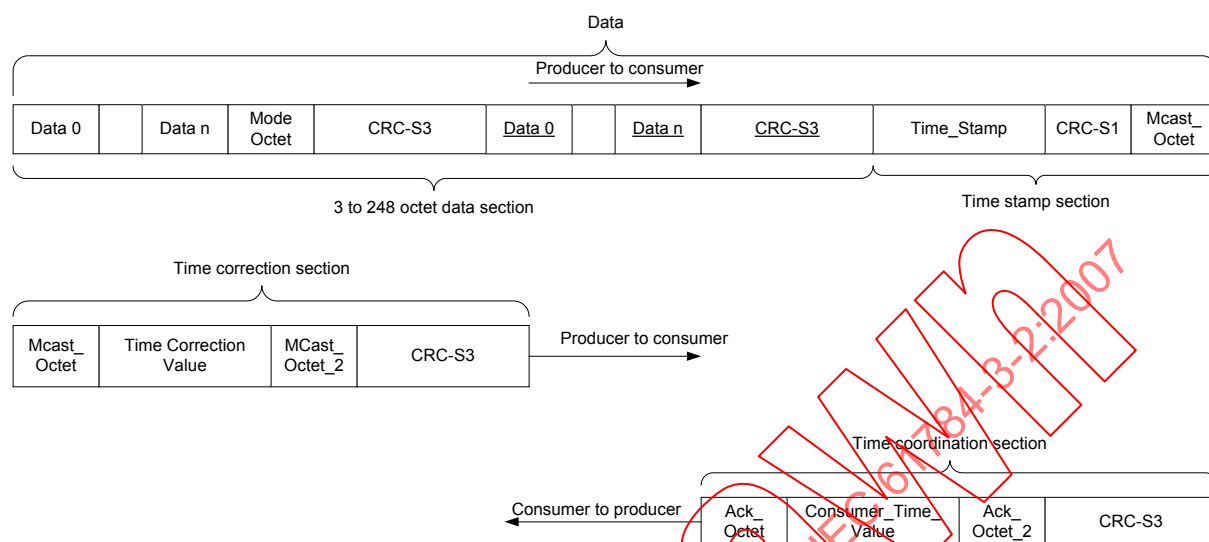
Légende

Anglais	Français
data	Données
producer to consumer	Producteur vers consommateur
mode octet	Octet Mode
3 to 250 octet data section	Section de données de 3 à 250 octets
time stamp section	Section de datation
time coordination section	Section de coordination temporelle
consumer to producer	Consommateur vers producteur

Figure 27 – Codage PDU point à point de 3 à 250 octets

7.1.1.15 Format PDU de sécurité de format 1 multipoint de 3 à 248 octets

La Figure 28 définit le codage du format de connexion de sécurité multipoint de 3 à 248 octets.



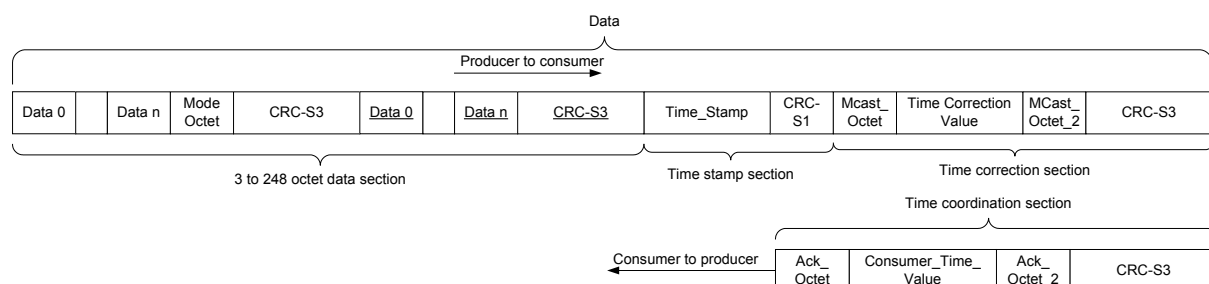
Légende

Anglais	Français
data	Données
producer to consumer	Producteur vers consommateur
mode octet	Octet Mode
3 to 248 octet data section	Section de données de 3 à 248 octets
time stamp section	Section de datation
time correction section	Section de correction de temps
producer to consumer	Producteur vers consommateur
time coordination section	Section de coordination temporelle
consumer to producer	Consommateur vers producteur

Figure 28 – Codage PDU multipoint de 3 à 248 octets

7.1.1.16 Format PDU de sécurité de format 2 multipoint de 3 à 248 octets

La Figure 29 définit le codage du format de connexion de sécurité multipoint de 3 à 248 octets.



Légende

Anglais	Français
data	Données

Anglais	Français
producer to consumer	Producteur vers consommateur
mode octet	Octet Mode
3 to 248 octet data section	Section de données de 3 à 248 octets
time stamp section	Section de datation
time correction value	Valeur de correction de temps
time correction section	Section de correction de temps
time coordination section	Section de coordination temporelle
consumer to producer	Consommateur vers producteur

Figure 29 – Format de connexion de sécurité multipoint de 3 à 248 octets

7.1.2 CRC de sécurité

7.1.2.1 Présentation générale du CRC de sécurité

Le FSCP 2/1 utilise trois CRC de sécurité. Un CRC de 8 bits (CRC-S1 ou CRC-S2) est utilisé pour les connexions de sections de données contenant jusqu'à deux octets de données de sécurité, ainsi que les sections de datation. Un CRC de 16 bits (CRC-S3) est utilisé pour la section de données de 3 à 250 octets et pour les messages de correction de temps et de coordination temporelle. Un quatrième CRC relatif à la sécurité, un CRC de 32 bits (CRC-S4), est utilisé pour l'établissement d'une connexion, la configuration et l'extension de sécurité de la définition du fichier EDS.

7.1.2.2 Polynômes de CRC

Le Tableau 78 définit les polynômes utilisés dans le FSCP 2/1.

Tableau 78 – Polynômes de CRC utilisés

Nom	Polynôme générateur	Valeur Hex de polynôme
CRC-S1	$g(x) = x^8 + x^5 + x^4 + x^2 + x^1 + 1$	137 _{hex}
CRC-S2	$g(x) = x^8 + x^5 + x^4 + x^3 + x^1 + 1$	13b _{hex}
CRC-S3	$g(x) = x^{16} + x^{11} + x^3 + x^2 + x^1 + 1$	1080F _{hex}
CRC-S4	$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^4 + x^2 + x^1 + 1$	Bit inversé à EDB88320 _{hex}

7.1.2.3 Spécifications d'utilisation des CRC

L'utilisation du CRC de messagerie E/S est définie en 7.1, avec application des CRC-S1, CRC-S2 et CRC-S3, de nombreux autres CRC étant toutefois définis dans le FSCP 2/1 pour l'établissement d'une connexion, la configuration et l'extension de sécurité de la définition du fichier EDS. Le Tableau 79 définit le type d'algorithme utilisé et la valeur de départ pour chaque utilisation.

Tableau 79 – Sections de connexion et formats de messages

Nom du CRC	Algorithme utilisé	Valeur de départ	Description
SCCRC	CRC-S4 (voir A.1)	FFFFFFFF _{hex}	CRC de configuration de sécurité
CPCRC	CRC-S4 (voir A.1)	FFFFFFFF _{hex}	CRC de paramètre de connexion
CRC de mot de passe	CRC-S4 (voir A.1)	FDFDFDFD _{hex}	Utilisé pour le cryptage des mots de passe avant transmission
CRC du fichier EDS	CRC-S4 (voir A.1)	FDFDFDFD _{hex}	Utilisé pour assurer l'intégrité des fichiers EDS utilisés par les dispositifs de sécurité

7.2 Comportement du protocole de communication

7.2.1 Séquence des contrôles de sécurité

La séquence suivante des contrôles doit être utilisée pour vérifier les messages.

- Contrôle des comptages de demandes Ping;
- Évaluation du CRC de section de datation;
- Évaluation des datations;
- Évaluation du CRC des données réelles;
- Évaluation du CRC des données complémentaires;
- Évaluation d'une contre-vérification.

7.2.2 Terminaison de connexion

Lorsqu'un producteur détecte une erreur qui exige une terminaison de connexion (c'est-à-dire une réponse aux demandes Ping hors de l'intervalle Ping du producteur), il doit terminer la connexion et notifier l'application de l'action.

Tous les consommateurs doivent surveiller la transmission périodique des données et passer à un état de sécurité en cas d'arrêt des transmissions périodiques. Si le consommateur détecte une erreur, il doit passer à un état de sécurité et terminer la connexion de consommation associée à cette erreur.

Lorsque la couche de sécurité détecte une erreur qui exige la terminaison d'une connexion, la séquence suivante doit être suivie pour effectuer ladite terminaison.

- La couche de sécurité détecte une erreur qui exige une terminaison de connexion.
- La couche de sécurité notifie le programme d'application par le paramétrage de l'état de connexion, afin d'indiquer une panne de communication de sécurité.
- L'application relative à la sécurité doit transférer les données et les E/S (par exemple, réglage des sorties sur l'état de sécurité) associées à la connexion vers un état de sécurité.
- La couche de sécurité doit notifier le système de communications sous-jacent d'une erreur et solliciter la terminaison de la connexion par le paramétrage de son état de manière à indiquer une erreur de communication de sécurité.
- La couche de sécurité doit maintenir son état de sécurité jusqu'à la suppression de la panne et jusqu'au lancement d'une séquence de redémarrage de connexion.

7.2.3 Erreur de contre vérification

Lorsque des données de sécurité ayant fait l'objet d'une contre-vérification se révèlent différentes, les données sont traitées comme des données erronées et le consommateur doit terminer la connexion (voir 7.2.2).

7.3 Opération de datation

La datation permet de calculer l'actualité de réception des données dans le consommateur. La Figure 30 illustre la séquence des opérations de datation pour une paire producteur/consommateur point à point simple.

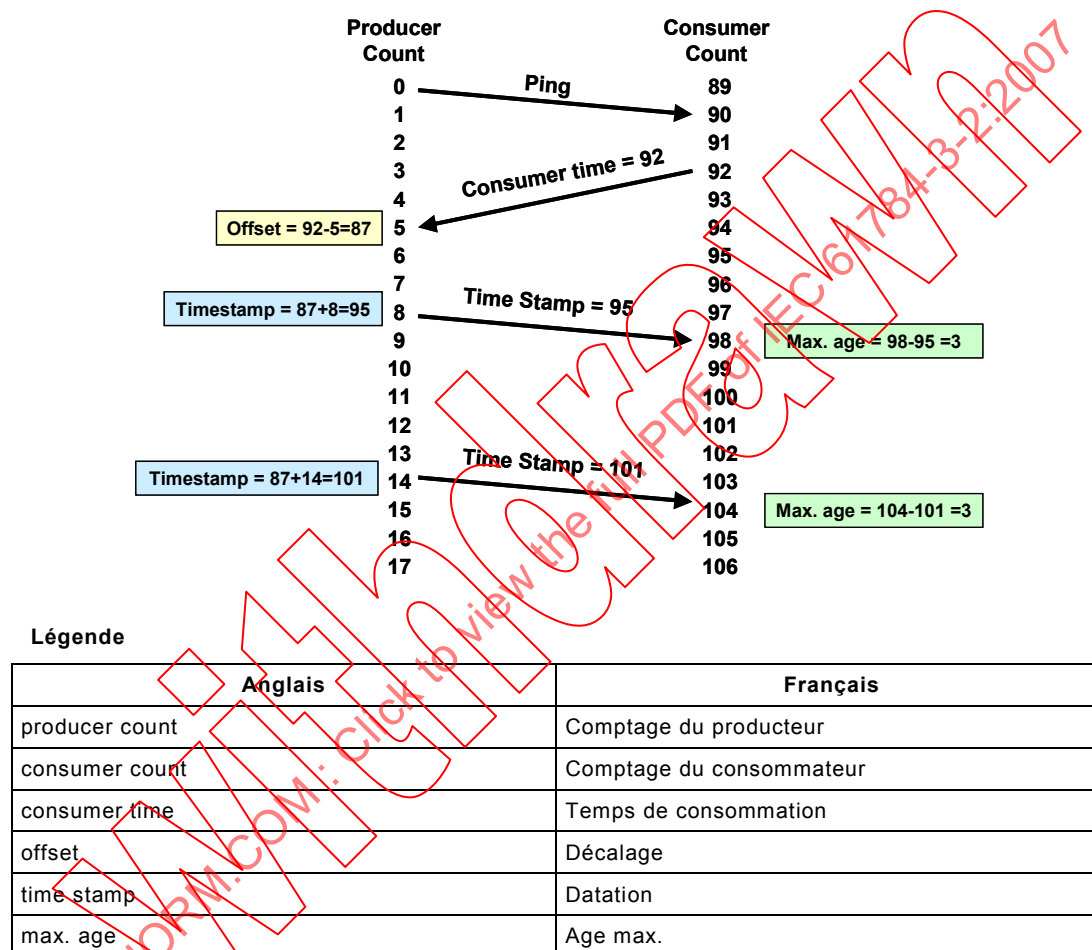


Figure 30 – Séquence de datation

A l'établissement de la connexion, le producteur doit solliciter une réponse à la demande Ping initiale.

Lorsque le consommateur constate une modification dans le comptage des demandes Ping, il doit préparer un message de réponse de coordination temporelle contenant la valeur de consommation de son comptage d'horloge.

Le producteur doit utiliser le comptage d'horloge dans la réponse de coordination temporelle afin de calculer un décalage qui sera appliqué aux productions futures.

Pour chaque production ordonnancée, le producteur doit utiliser la valeur de son propre comptage d'horloge, ainsi que le décalage pour calculer la datation, et transmettre ces informations au consommateur avec les données.

Le consommateur doit utiliser la valeur qu'il a reçue du producteur pour calculer l'âge des données, et ce, par soustraction de la datation du producteur de son comptage d'horloge courant. Si le delta calculé est inférieur au delta maximal spécifié par l'utilisateur, les données sont correctes et sont utilisées par l'application interne au dispositif. Si le delta est supérieur à la valeur spécifiée par l'utilisateur, les données sont considérées comme trop anciennes pour être utilisées, ce qui entraîne la terminaison de la connexion.

Le producteur doit solliciter une autre réponse à la demande Ping du consommateur pour chaque intervalle Ping. Ceci occasionne un nouveau calcul du décalage du producteur et empêche le vieillissement lent des données provoqué par des erreurs de temps.

On ajoute un facteur de correction de temps afin de tenir compte de l'obliquité et de la dérive maximales des horloges dans un système asynchrone. Ce message permet d'ajuster le décalage du producteur.

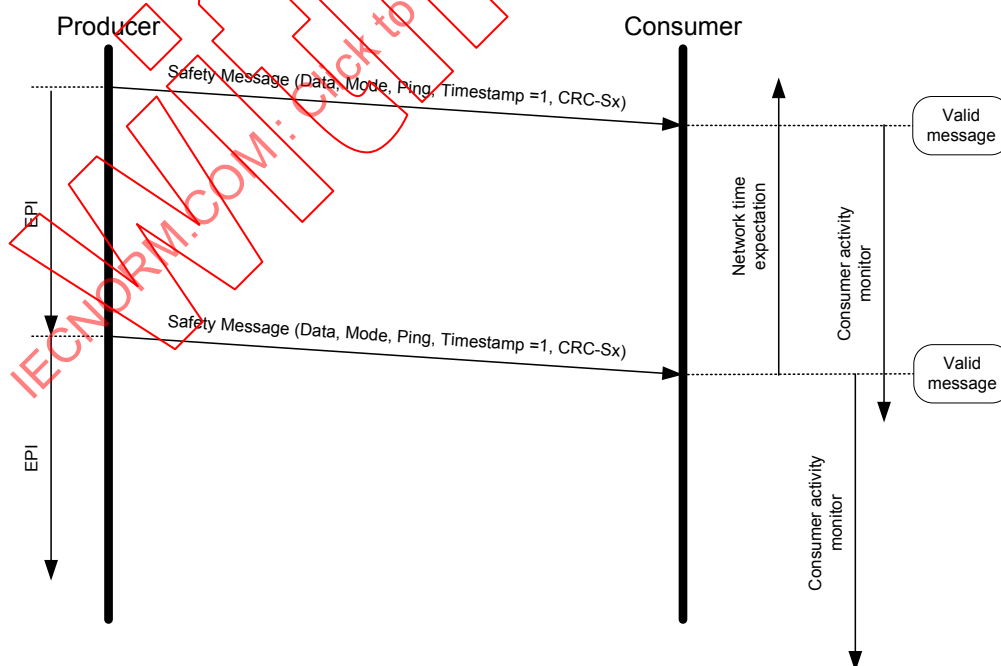
7.4 Diagrammes séquentiels de protocoles

7.4.1 Généralités

Les diagrammes séquentiels de protocoles illustrent la relation entre producteurs et consommateurs par rapport à la datation et aux événements. L'écoulement du temps s'effectue du haut vers le bas de chaque figure. Les messages normaux reçus par le producteur ou le consommateur sont présentés dans les zones de texte normal. Les messages anormaux sont présentés dans les zones de texte en gras. Les messages attendus mais non reçus sont présentés sous forme de texte libre.

7.4.2 Transmission de sécurité normale

Une communication normale entre un producteur et un consommateur de sécurité exige une séquence de messages périodiques. Lorsque le consommateur reçoit un message valide, la transaction s'effectue tel que cela est défini à la Figure 31.



Légende

Anglais	Français
producer	Producteur

Anglais	Français
consumer	Consommateur
safety message	Message de sécurité
valid message	Message valide
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur

Figure 31 – Diagramme d'une séquence de sécurité producteur/consommateur normale

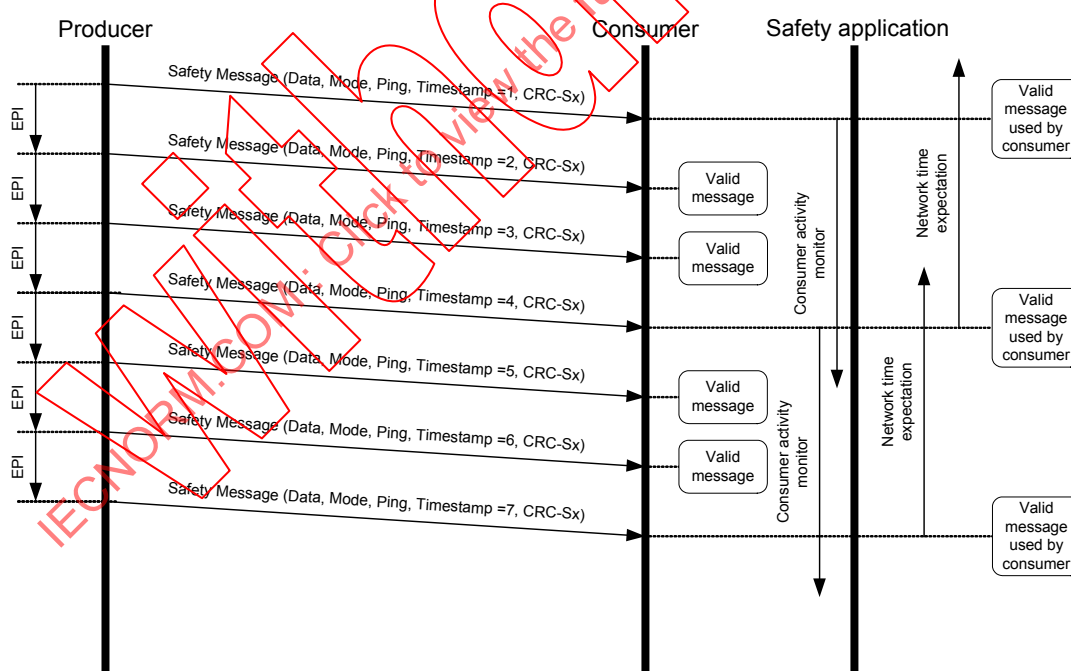
Le producteur de sécurité génère des données à une fréquence fixe appelée Intervalle de paquets attendu (EPI).

Les consommateurs déclenchés par la liaison doivent maintenir un temporisateur d'activité réinitialisé après réception d'un message attendu valide.

Si un message valide n'a pas été reçu avant expiration du temporisateur d'activité, le consommateur déclenché par la liaison doit terminer la connexion (voir 7.2.2) et passer à l'état de sécurité défini.

Les consommateurs doivent vérifier l'âge des données reçues par un contrôle des datations reçues.

Si l'âge des données se révèle supérieur au délai du réseau, le consommateur doit terminer la connexion (voir 7.2.2) et passer à l'état de sécurité défini.



Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety application	Application de sécurité
safety message	Message de sécurité
valid message used by consumer	Message valide utilisé par le consommateur

Anglais	Français
valid message	Message valide
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur

Figure 32 – Diagramme d'une séquence de sécurité producteur/consommateur normale (production répétée)

Les communications de sécurité doivent être configurables avec la production répétée. Dans ce cas, le consommateur n'utilise pas chaque paquet qui lui est transmis.

Lorsqu'une communication de sécurité est transmise avec une production répétée, les consommateurs déclenchés par l'application doivent utiliser le dernier paquet reçu pour achever la transaction, tel qu'illustré à la Figure 32.

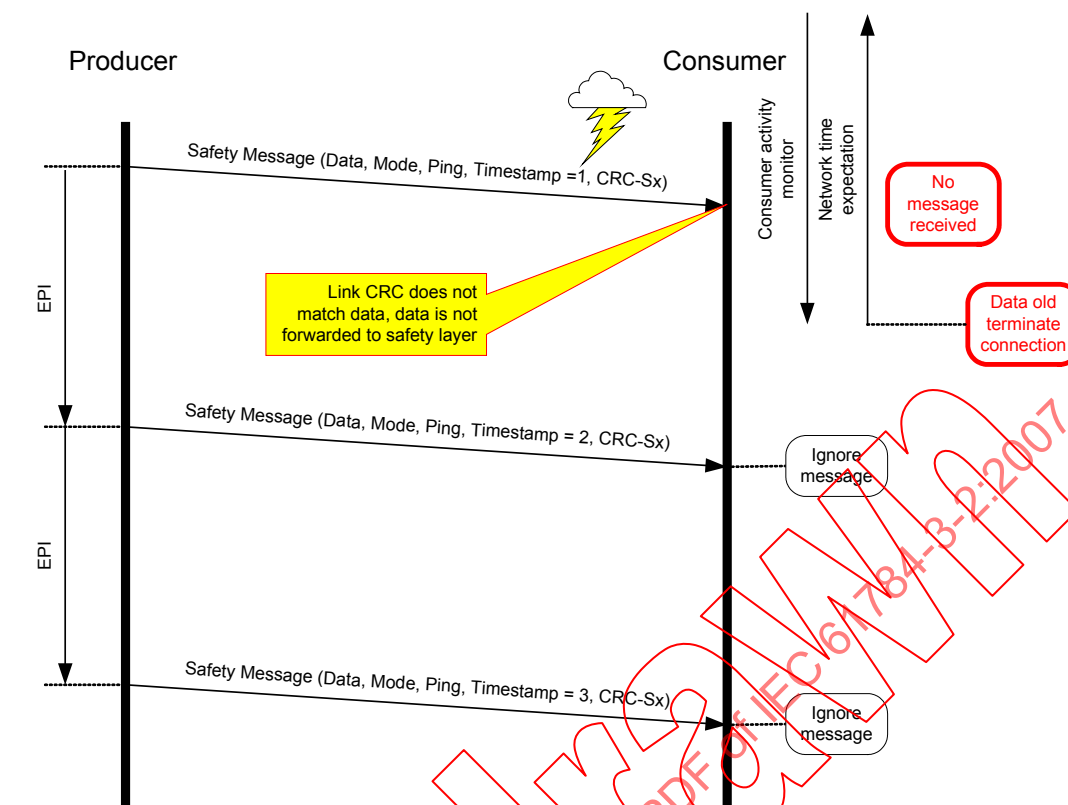
Le consommateur doit constater l'augmentation de datation pour chaque période successive; à défaut, la connexion est terminée (voir 7.2.2).

L'envoi d'un message avec une production répétée offre l'avantage supplémentaire d'utiliser les données les plus récentes.

7.4.3 Transmission de messages perdus, corrompus et retardés

Il est possible, en raison de perturbations extérieures, que le message transmis du producteur au consommateur soit corrompu, tel que défini à la Figure 33. Si un message corrompu est détecté au niveau de la couche normalisée, il doit être éliminé et non présenté à l'application. Ceci entraîne une expiration du contrôleur d'activité du consommateur. L'âge des données dépasse également le délai du réseau et le consommateur termine la connexion (voir 7.2.2) et passe à l'état de sécurité défini.

Si les CRC-Sx et les données transmises sont vérifiés et si l'on constate qu'ils comportent une erreur, le consommateur doit alors conclure que la liaison sous-jacente n'est pas fiable, terminer la connexion et passer à l'état de sécurité défini (voir 7.2.2).

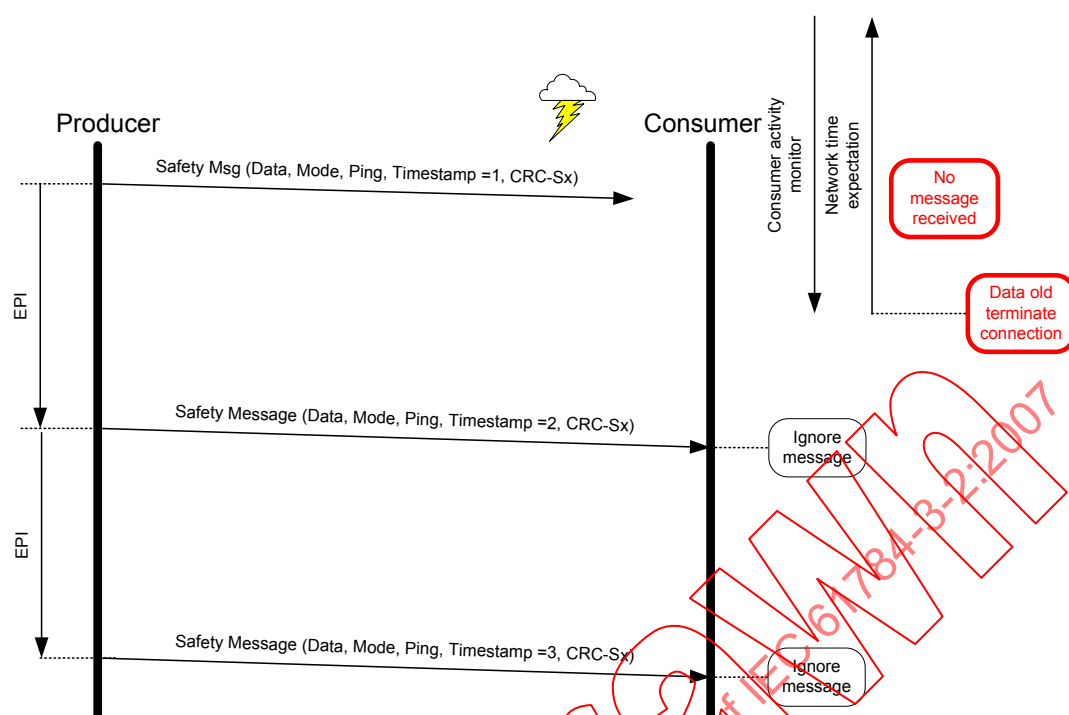


Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety message	Message de sécurité
no message received	Aucun message reçu
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur
Link CRC does not match data, data is not forwarded to safety layer	Le CRC de liaison ne correspond pas aux données, les données ne sont pas transmises à la couche de sécurité
data old terminate connection	Les anciennes données terminent la connexion
ignore message	Ignorer message

Figure 33 – Diagramme séquentiel d'un message corrompu d'un producteur à un consommateur

Il est possible, en raison de perturbations extérieures, que le message transmis du producteur au consommateur soit perdu, tel que défini à la Figure 34. Ceci entraîne une expiration du contrôleur d'activité du consommateur. L'âge des données dépasse également le délai du réseau et le consommateur termine la connexion (voir 7.2.2) et passe à l'état de sécurité défini.



Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety message	Message de sécurité
no message received	Aucun message reçu
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur
data old terminate connection	Les anciennes données terminent la connexion
ignore message	Ignorer message

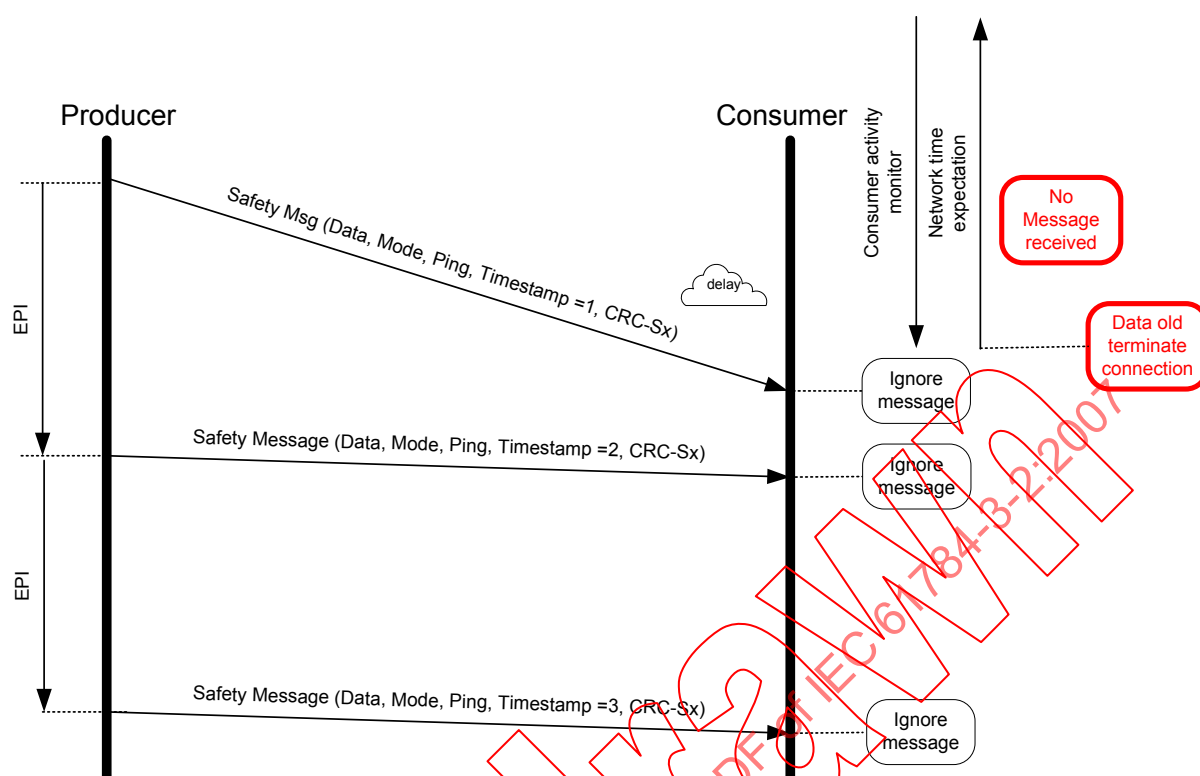
Figure 34 – Diagramme séquentiel d'un message producteur perdu vers consommateur

La transmission des messages peut être retardée. Si un message est retardé de sorte

- qu'il est reçu au delà du délai du contrôleur d'activité du consommateur (application déclenchée par la liaison), ou
- que l'âge des données dépasse le délai du réseau,

alors tous les messages ultérieurs sont ignorés, et la connexion doit être terminée.

Dans les deux cas, le consommateur termine la connexion (voir 7.2.2) et passe à l'état de sécurité défini. Ce comportement est défini à la Figure 35.



Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety message	Message de sécurité
no message received	Aucun message reçu
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur
data old terminate connection	Les anciennes données terminent la connexion
ignore message	Ignorer message
delay	Retard

Figure 35 – Diagramme séquentiel d'un message retardé

7.4.4 Transmission de messages perdus, corrompus ou retardés avec production répétée

Il est possible, en raison de perturbations extérieures, que le message transmis du producteur au consommateur soit corrompu, tel que défini à la Figure 36. La production peut être répétée afin de minimiser les déclenchements de nuisance et accroître la disponibilité du système entier.



Légende

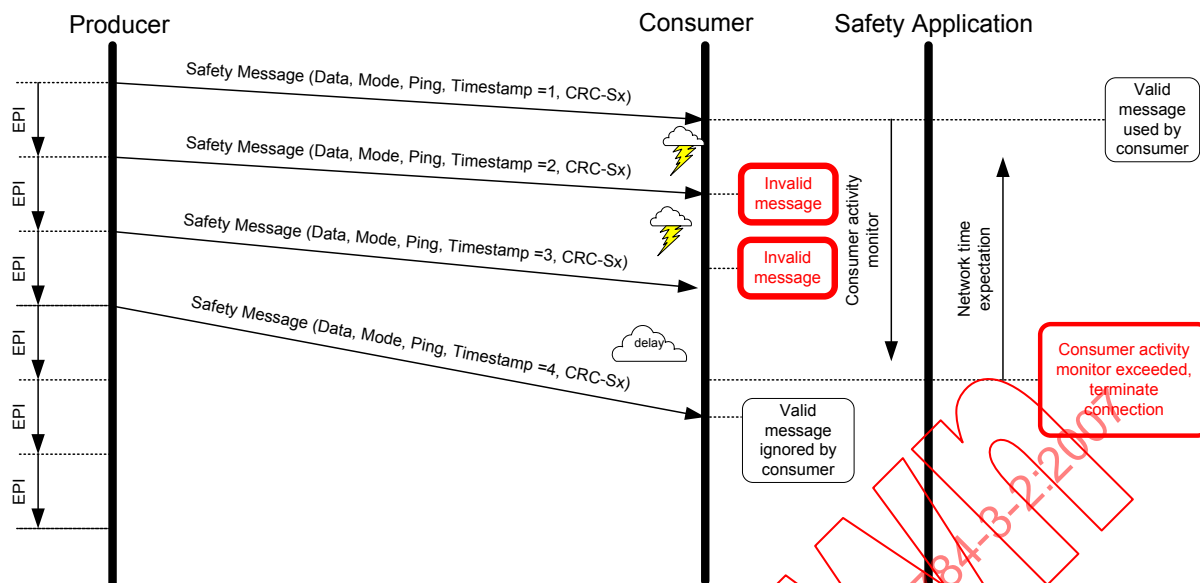
Anglais	Français
producer	Producteur
consumer	Consommateur
safety application	Application de sécurité
safety message	Message de sécurité
valid message used by consumer	Message valide reçu par le consommateur
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur
Link CRC does not match data, data is not forwarded to safety layer	Le CRC de liaison ne correspond pas aux données, les données ne sont pas transmises à la couche de sécurité
link valid message	Message de liaison valide

Figure 36 – Diagramme séquentiel d'un message producteur corrompu vers consommateur avec production répétée

Si des erreurs sont détectées dans les couches normalisées, les messages peuvent être éliminés, tant qu'un message attendu valide est reçu dans le contrôleur d'activité du consommateur et que l'âge des données ne dépasse pas le délai du réseau.

Les messages répétés valides doivent être écrasés. L'application utilise le message valide le plus récent.

Il peut être possible, en raison de perturbations extérieures, qu'une série de messages soit retardée, entraînant la perte de la connexion. Cette séquence est définie à la Figure 37. La connexion est terminée si aucun message valide n'est reçu avant expiration du temporisateur périodique du consommateur (voir 7.2.2).

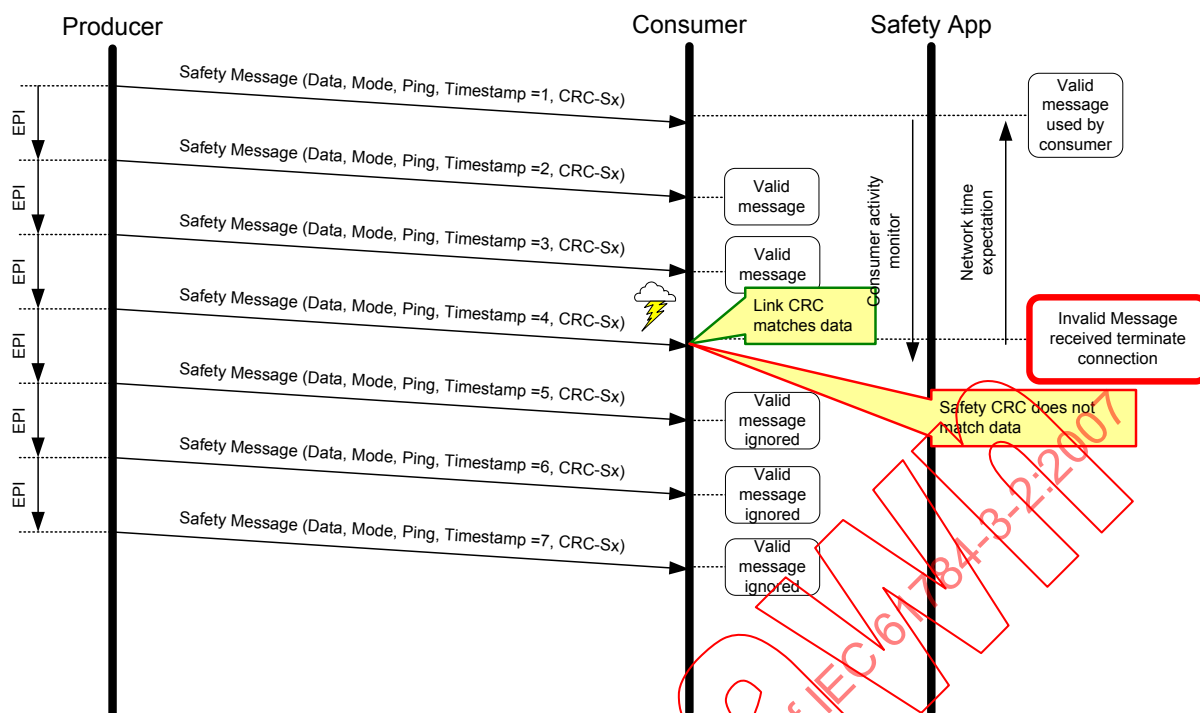


Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety application	Application de sécurité
safety message	Message de sécurité
valid message used by consumer	Message valide reçu par le consommateur
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur
invalid message	Message invalide
delay	Retard
consumer activity monitor exceeded, terminate connection	Délai du contrôleur d'activité du consommateur dépassé, terminaison de la connexion
valid message ignored by consumer	Message valide ignoré par le consommateur

Figure 37 – Diagramme séquentiel d'une terminaison de connexion occasionnée par des retards

Un message de liaison valide qui ne satisfait pas au contrôle CRC de sécurité conduira le consommateur à supposer que la liaison sous-jacente n'est pas fiable et donc à procéder à la terminaison immédiate de la connexion. Cette séquence est définie à la Figure 38.



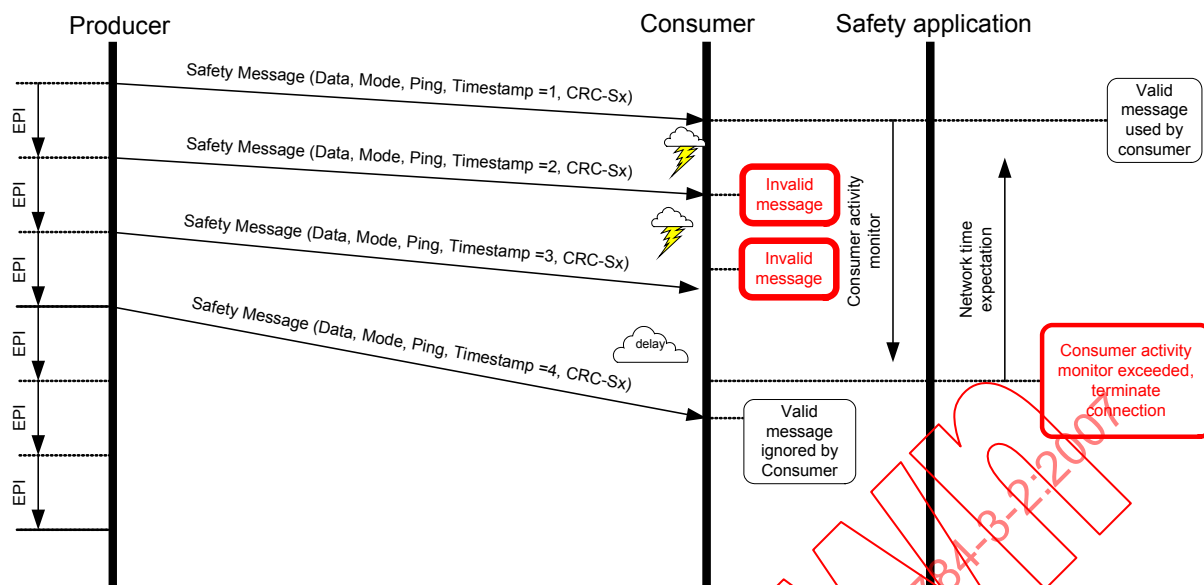
Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety application	Application de sécurité
safety message	Message de sécurité
valid message used by consumer	Message valide reçu par le consommateur
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur
valid message	Message valide
link CRC matches data	Le CRC de liaison correspond aux données
invalid message received terminate connection	Message invalide reçu, terminaison de la connexion
valid message ignored	Message valide ignoré
safety CRC does not match data	Le CRC de sécurité ne correspond pas aux données

Figure 38 – Diagramme séquentiel d'un contrôle CRC de sécurité non satisfait

7.4.5 Demande Ping point à point

Une séquence de transaction de messages de demande et de réponse Ping permet de vérifier que les messages ne sont pas retardés par des ponts. La description complète du producteur est donnée en 7.5.5 et celle du consommateur est donnée en 7.5.6. Cette séquence de messages pour une demande Ping point à point est définie à la Figure 39.



Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety application	Application de sécurité
safety message	Message de sécurité
valid message used by consumer	Message valide reçu par le consommateur
network time expectation	Délai du réseau
consumer activity monitor	Contrôleur d'activité du consommateur
invalid message	Message invalide
delay	Retard
consumer activity monitor exceeded, terminate connection	Délai du contrôleur d'activité du consommateur dépassé, terminaison de la connexion
valid message ignored by consumer	Message valide ignoré par le consommateur

Figure 39 – Diagramme séquentiel d'une demande Ping point à point – réponse normale

A chaque intervalle Ping de producteur point à point, le comptage de demandes Ping doit être incrémenté dans l'en-tête de message de sécurité disponible suivant.

L'intervalle Ping de producteur point à point doit être un multiple de l'EPI; les demandes Ping peuvent être formulées à une fréquence de l'EPI jusqu'à 0,5 s.

Lorsqu'un consommateur point à point constate une modification dans le comptage de demandes Ping, il génère un message de coordination temporelle.

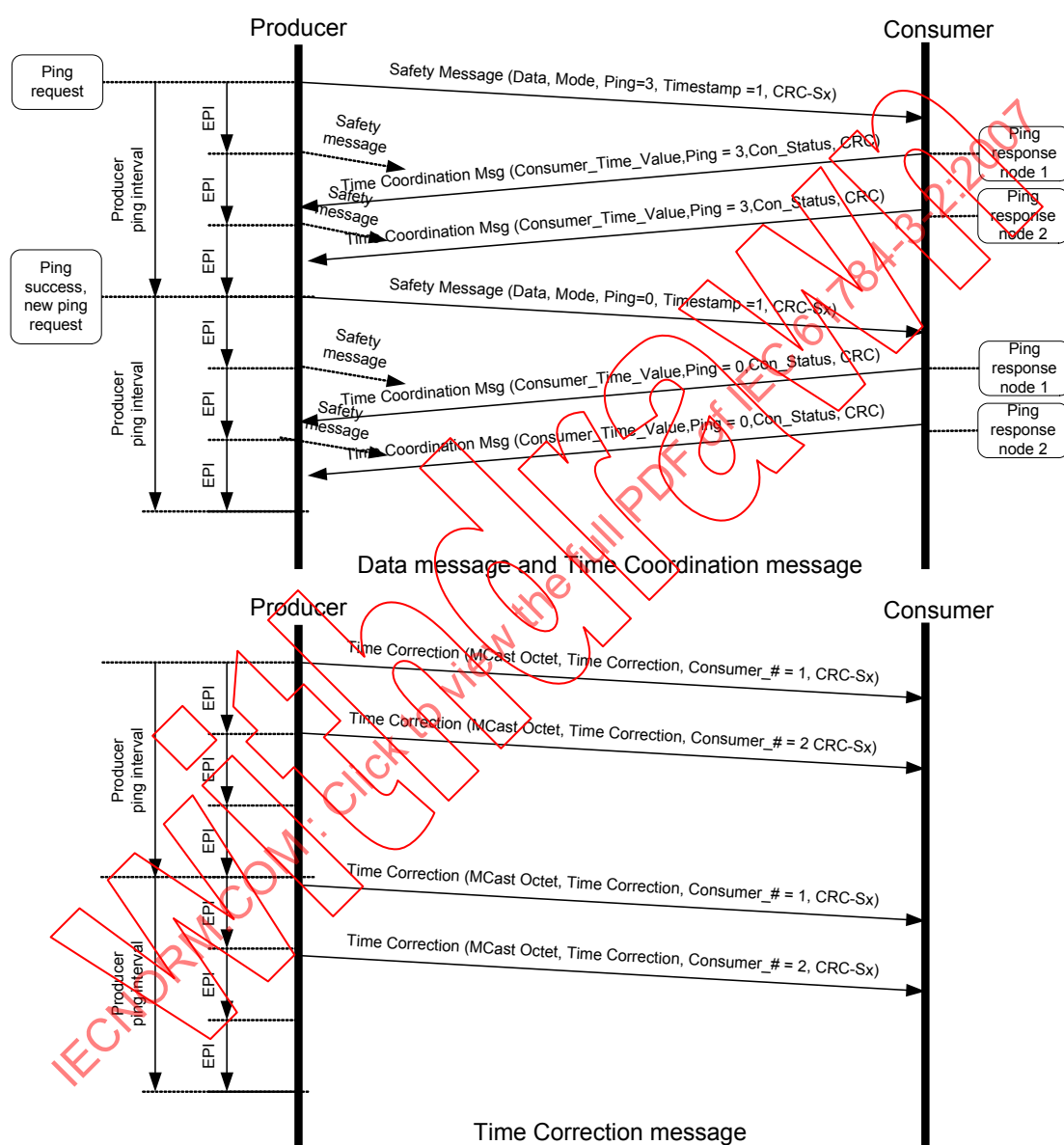
Le message de coordination temporelle (voir 7.4.2) doit être reçu par le producteur préalablement à l'expiration de l'intervalle Ping de producteur courant. Si le message de coordination temporelle n'est pas reçu dans l'intervalle Ping de producteur, le producteur doit terminer la connexion (voir 7.2.2).

7.4.6 Demande Ping multipoint avec sécurité CP 2/3

Une demande et une réponse à demande Ping permettent de vérifier que les messages ne sont pas retardés par des ponts.

Pour les connexions multipoint de sécurité CP 2/3, un message distinct doit être utilisé pour communiquer les informations de correction de temps du producteur à chaque consommateur multipoint.

La description complète du producteur est donnée en 7.5.5 et celle du consommateur est donnée en 7.5.6. Cette séquence de messages pour une demande Ping multipoint est définie à la Figure 40. Le message de correction de temps est défini sur un diagramme séparé pour des raisons de clarté.



Anglais	Français
ping success, new ping request	Demande ping satisfaisante, nouvelle demande Ping
time coordination msg	Message de coordination temporelle
data message and time coordination message	Message de données et message de coordination temporelle
time correction	Correction de temps
time correction message	Message de correction de temps

Figure 40 – Diagramme séquentiel d'une demande Ping multipoint satisfaisante, sécurité CP 2/3

Pendant chaque intervalle Ping de producteur multipoint, une demande Ping est transmise par les producteurs multipoint aux consommateurs multipoint sollicitant une réponse.

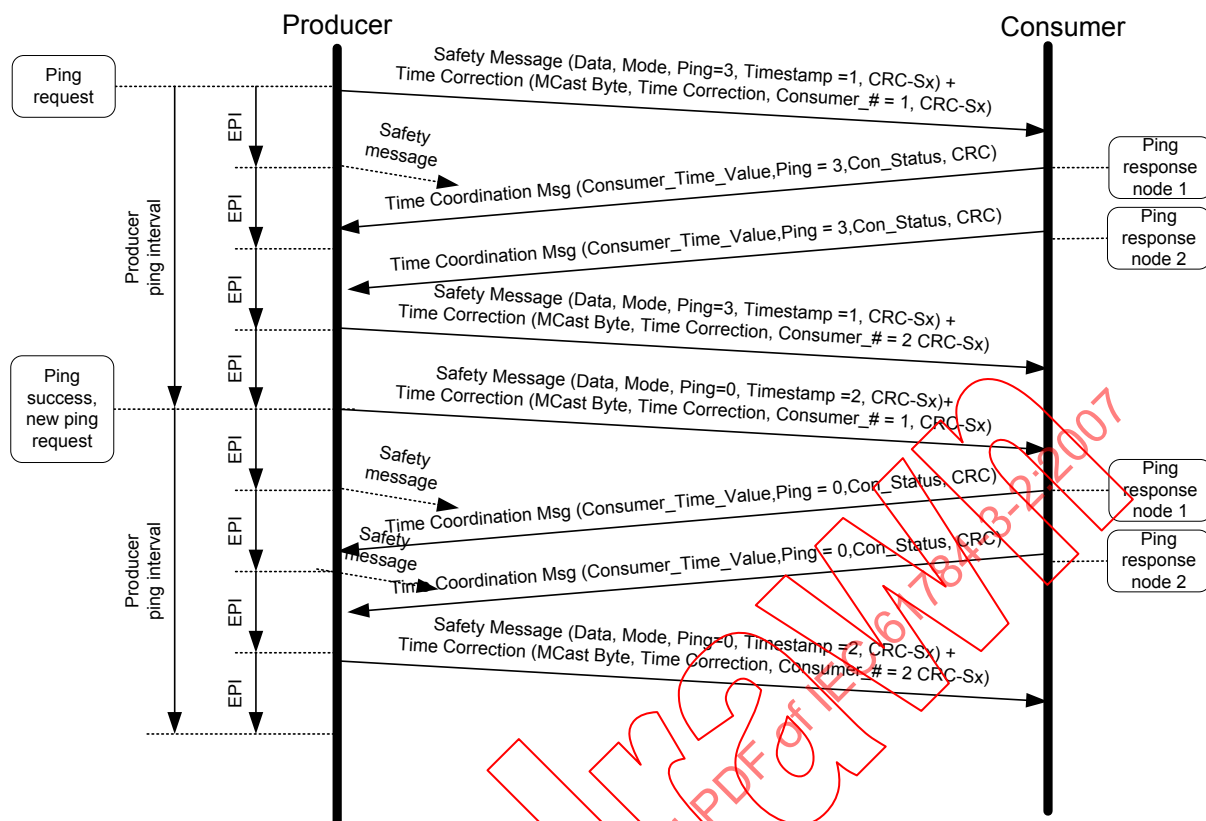
Tous les consommateurs multipoint doivent répondre à une modification de la valeur de demande Ping, et les consommateurs autres que le consommateur 1 doivent attendre Consumer_Number -1 x l'EPI pour répondre.

Tous les consommateurs multipoint doivent répondre dans le délai de l'intervalle Ping de producteur unique.

Les messages de correction de temps doivent être transmis aux consommateurs multipoint à une fréquence qui permet d'assurer que chaque consommateur obtient un message de correction de temps dans un intervalle Ping de producteur unique.

7.4.7 Demande Ping multipoint sur les réseaux de sécurité CP 2/2

Une demande et une réponse à demande Ping sont utilisées pour vérifier que les messages ne sont pas retardés par des ponts. Pour les réseaux de sécurité CP 2/2, le message de coordination temporelle doit être concaténé au message de données de sécurité et transmis comme message unique. La description complète du producteur est donnée en 7.5.5 et celle du consommateur est donnée en 7.5.6. Cette séquence de messages pour une demande Ping multipoint est définie à la Figure 41.



Légende

Anglais	Français
producer	Producteur
consumer	Consommateur
safety message	Message de sécurité
ping request	Demande Ping
producer ping interval	Intervalle Ping du producteur
ping response	Réponse à demande Ping
node	Nœud
ping success, new ping request	Demande ping satisfaisante, nouvelle demande Ping
time coordination msg	Message de coordination temporelle
data message and time coordination message	Message de données et message de coordination temporelle
time correction	Correction de temps
time correction message	Message de correction de temps

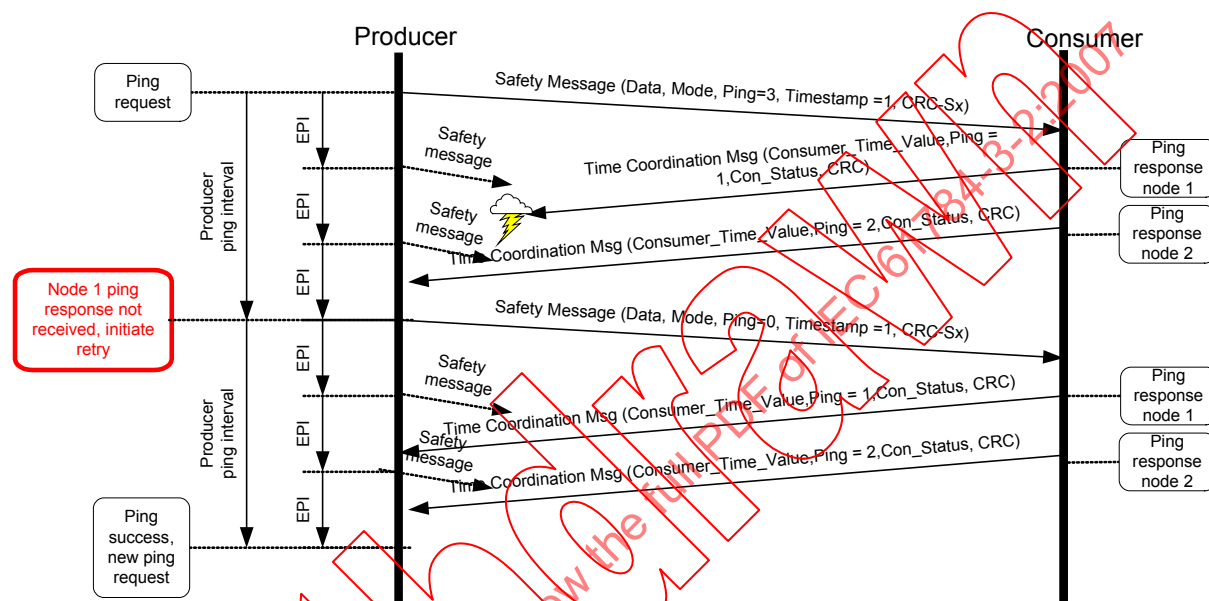
Figure 41 – Diagramme séquentiel d'une demande Ping multipoint satisfaisante, sécurité CP 2/2

Pendant chaque intervalle Ping de producteur, une demande Ping est transmise par les producteurs multipoint aux consommateurs multipoint sollicitant une réponse. Tous les consommateurs répondent à une modification de la valeur de demande Ping, et les consommateurs autres que le consommateur 1 attendent Consumer_Number - 1 x l'EPI pour répondre. Tous les consommateurs multipoint doivent répondre dans le délai de l'intervalle Ping de producteur unique.

Les messages de correction de temps sont transmis au consommateur à une fréquence qui permet d'assurer que chaque consommateur obtient un message de correction de temps dans un intervalle Ping de producteur unique.

7.4.8 Demande Ping multipoint – Retransmission aboutie

Une demande et une réponse à demande Ping sont utilisées pour vérifier que les messages ne sont pas retardés par des ponts. La description complète du producteur est donnée en 7.5.5 et celle du consommateur est donnée en 7.5.6. Cette séquence de messages pour une demande Ping multipoint avec retransmissions est définie à la Figure 42. La connexion de correction de temps n'est pas définie pour des raisons de clarté.



Légende	
Anglais	Français
producer	Producteur
consumer	Consommateur
safety message	Message de sécurité
ping request	Demande Ping
producer ping interval	Intervalle Ping du producteur
ping response	Réponse à demande Png
node	Nœud
ping success, new ping request	Demande ping satisfaisante, nouvelle demande Ping
time coordination msg	Message de coordination temporelle
node 1 ping response not received, initiate retry	Réponse à demande Ping de nœud 1 non reçue, lancer retransmission

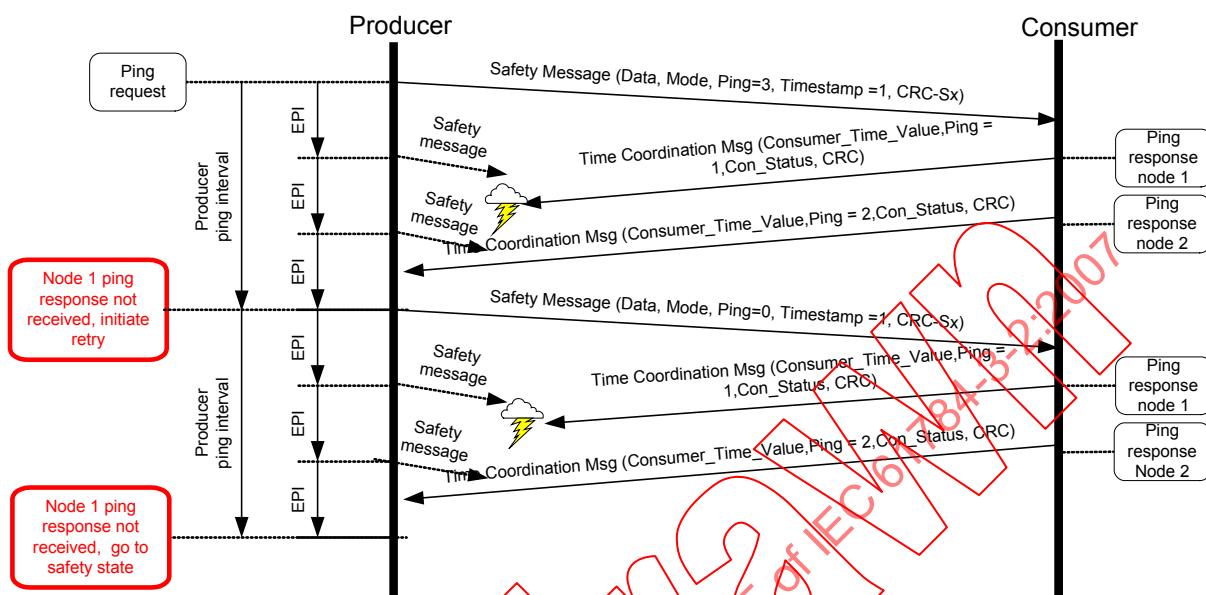
Figure 42 – Diagramme séquentiel d'une retransmission de demande Ping multipoint

La perte ou le retard d'un message de coordination temporelle multipoint après le nombre de Timeout_Multiplier +1 des temporisations de réponse à demande Ping de producteur doit entraîner la génération d'une erreur et la fermeture de la connexion.

7.4.9 Demande Ping multipoint – Retransmission avec temporisation

Une demande et une réponse à demande Ping de producteur sont utilisées pour vérifier que les messages ne sont pas retardés par des ponts. La description complète du producteur est

Node 1 ping response not received, go to safety state



Anglais	Français
producer	Producteur
consumer	Consommateur
safety message	Message de sécurité
ping request	Demande Ping
producer ping interval	Intervalle Ping du producteur
ping response	Réponse à demande Ping
node	Nœud
time coordination msg	Message de coordination temporelle
node 1 ping response not received, initiate retry	Réponse à demande Ping de nœud 1 non reçue, lancer retransmission
node 1 ping response not received, go to safety state	Réponse à demande Ping de nœud 1 non reçue, passer à l'état de sécurité

Figure 43 – Diagramme séquentiel d'une temporisation de demande Ping multipoint

Si la réponse à demande Ping demeure retardée, le producteur indique une anomalie de son état à destination du consommateur spécifique et le consommateur passe à l'état de sécurité défini. Si la demande Ping a été retardée en raison d'une panne du consommateur, ce dernier sera déjà passé à un état de sécurité.

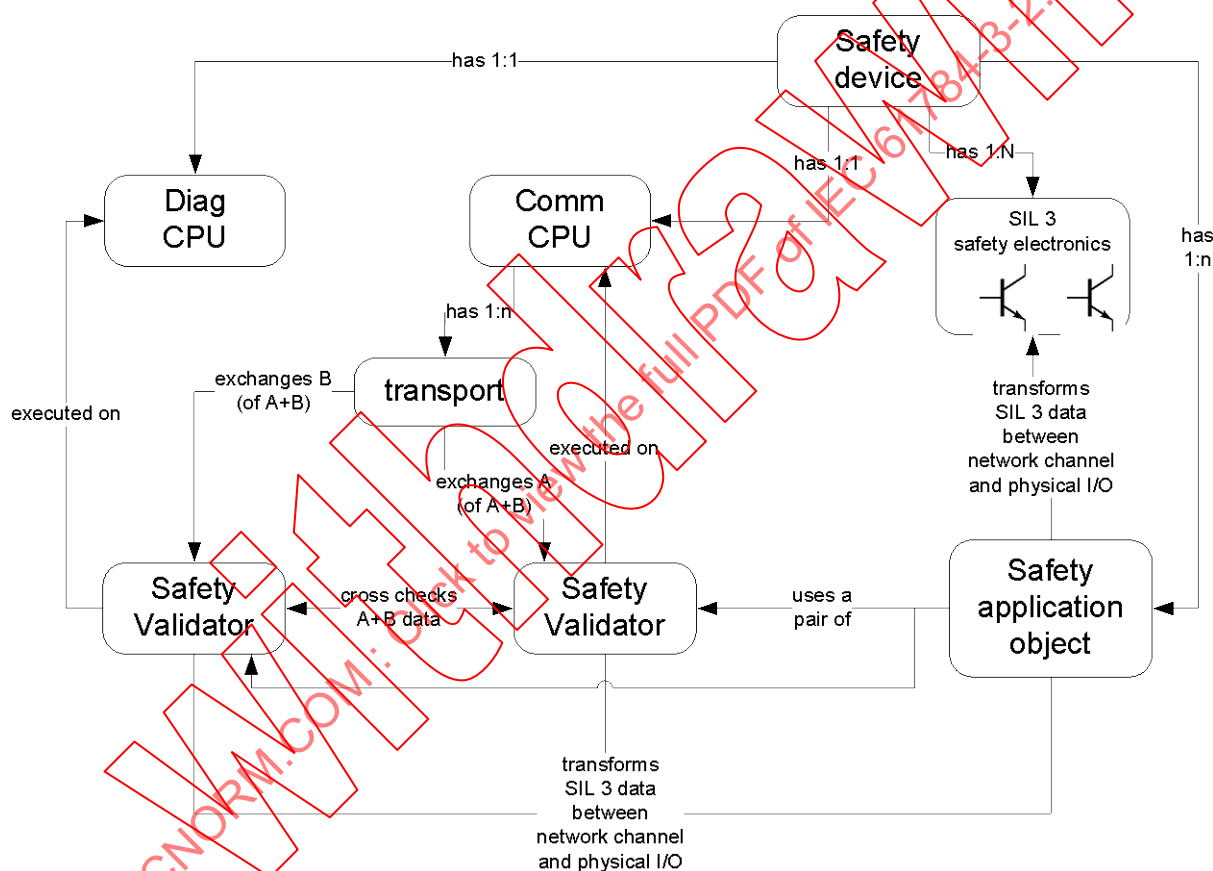
7.5.1 Généralités

La couche de sécurité est une couche de service située entre la couche de transport CPF 2 normalisée et la couche d'application relative à la sécurité, tel que défini à la Figure 44. Il n'est pas nécessaire que cette couche de sécurité soit directement adressable dans le cadre CPF 2; il s'agit en revanche d'un comportement associé à une connexion entre les applications qui échangent des données dans une application SIL 3. Ce comportement est mis en œuvre en tant qu'objet de validation de sécurité tel que décrit en 6.7.

NOTE Plusieurs méthodes peuvent permettre de réaliser ce comportement souhaité. Les objets indiqués dans ce paragraphe sont à considérer comme modèle de référence. Des produits et des applications individuels peuvent choisir de mettre en œuvre ce comportement, s'ils le considèrent nécessaire. Cependant, ce comportement est tenu d'être exprimé ici de manière à constituer la base de développement des produits.

7.5.2 Vue de haut niveau d'un dispositif de sécurité

La Figure 44 illustre un modèle de référence d'un dispositif de sécurité contenant un ou plusieurs objets d'application relative à la sécurité. Chaque objet d'application relative à la sécurité fonctionne avec une paire d'objets de validation de sécurité afin d'échanger les données SIL 3 entre la couche de transport CPF 2 et les circuits électroniques E/S de sécurité. Ces circuits électroniques sont conçus de manière à fournir un niveau de détection d'erreur et de localisation des pannes adapté aux applications SIL 3. Chacun des deux objets de validation de sécurité est associé à un canal de communication de réseau tel qu'un CP 2/3 ou CP 2/2. Ce modèle suppose une topologie de sécurité 1oo2 dans laquelle deux CPU indépendants participent au protocole de sécurité.



Légende

Anglais	Français
safety device	Dispositif de sécurité
diag CPU	CPU diag
comm CPU	CPU comm
SIL 3 safety electronics	Circuits électroniques de sécurité SIL 3
has ...	a ...
executed on	exécuté sur
exchanges ...	échange ...
(of A+B)	(de A+B)
transforms SIL 3 data between network channel and	transforme les données SIL 3 entre le canal du

Anglais	Français
physical I/O	réseau et les E/S physiques
safety validator	Objet de validation de sécurité
cross checks A+B data	contre-vérification des données A+B
uses a pair of	utilise une paire de
safety application object	Objet d'application de sécurité

Figure 44 – Diagramme de relations des entités de modèle de référence d'un dispositif de sécurité

7.5.3 Objet de validation de sécurité

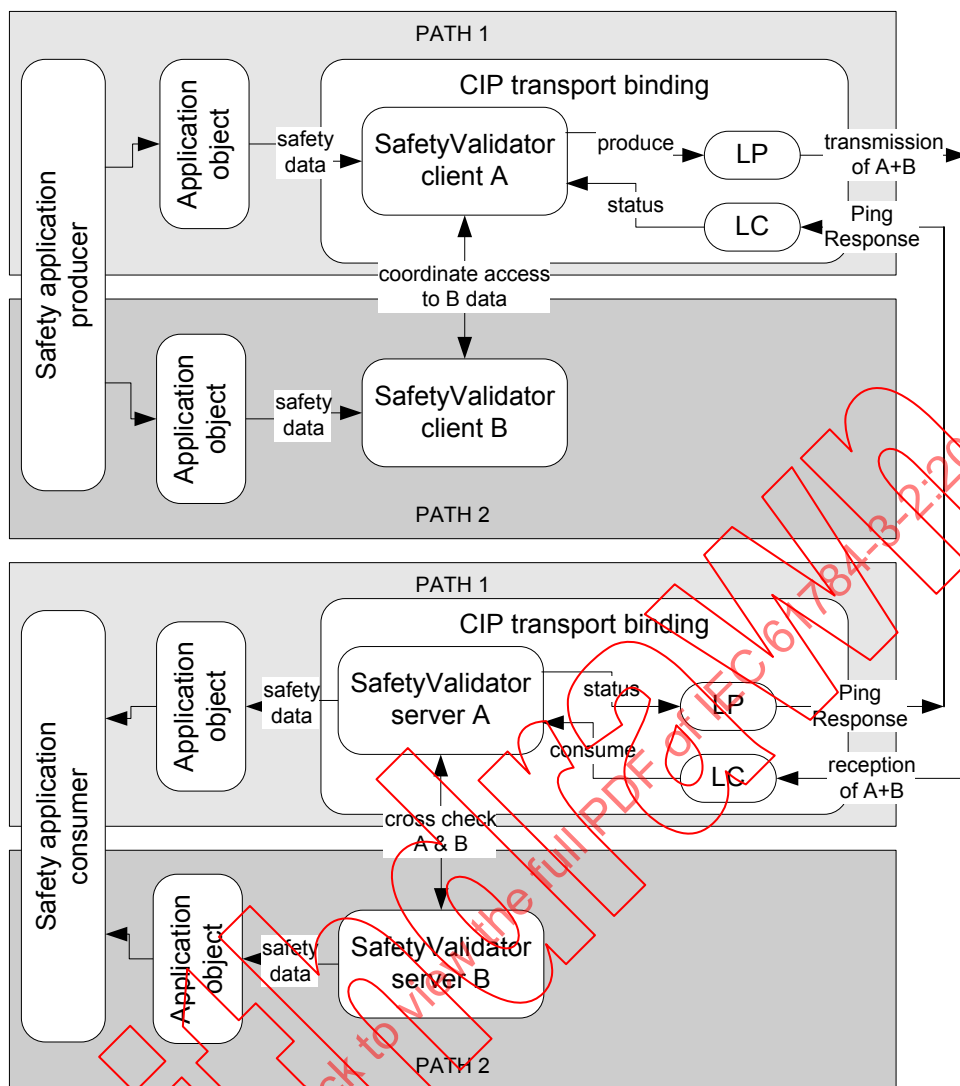
Le paragraphe 6.7 définit un Objet de validation de sécurité duquel les fonctions SafetyValidatorServer et SafetyValidatorClient sont mises en œuvre.

Les principes généraux et les attributs communs de ces deux fonctions au sein de l'objet sont définis en 7.5.4.

7.5.4 Relations entre SafetyValidatorServer et SafetyValidatorClient

La Figure 45 illustre une vue de haut niveau de deux dispositifs qui échangent des données via un SafetyValidatorClient et un SafetyValidatorServer. Une application de production de sécurité utilise un SafetyValidatorClient pour transmettre des données de sécurité à une application de consommation de sécurité qui utilise un SafetyValidatorServer.

NOTE Seul un des chemins redondants assure l'interface avec la couche de transport CPF 2. Les données A+B (réelles et complémentaires) sont encapsulées dans une trame de transport unique et un seul processeur est impliqué dans la réception et la transmission des trames de données.



Légende

Anglais	Français
PATH	CHEMIN
safety application producer	Producteur d'application de sécurité
application object	Objet d'application
CIP transport binding	Rattachement de transport CIP
safety data	données de sécurité
produce	produit
transmission of	transmission de
status	état
ping response	réponse à demande Ping
coordinate access to B data	coordonne l'accès aux données B
safety application consumer	Consommateur d'application de sécurité
consume	consomme
reception of	réception de
cross check	contre-vérification

Figure 45 – Deux dispositifs échangeant des données de sécurité via un SafetyValidatorClient et un SafetyValidatorServer

7.5.5 Définition de la fonction SafetyValidatorClient

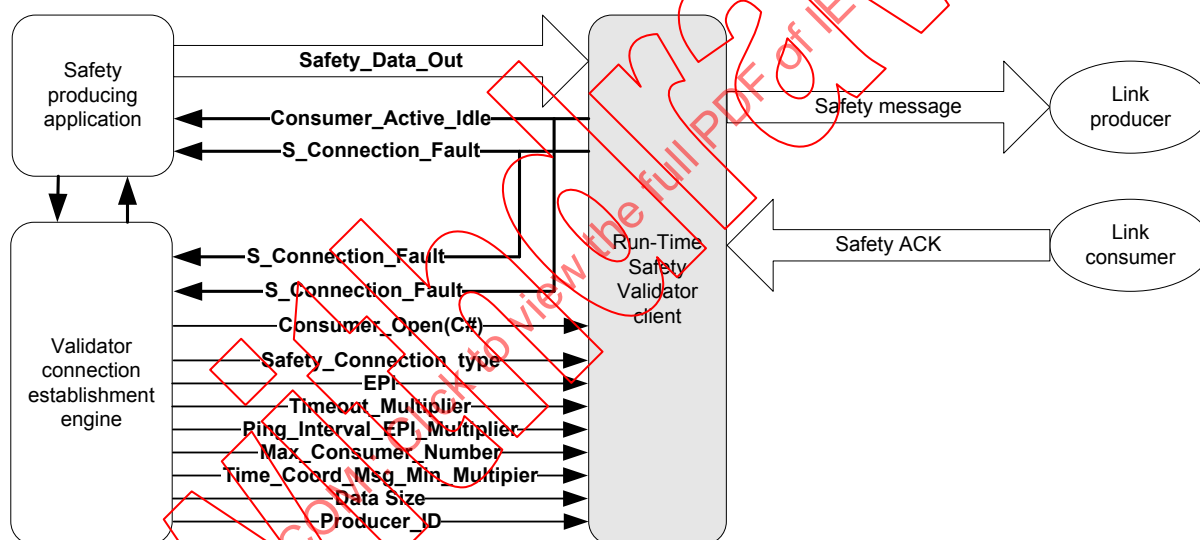
7.5.5.1 Généralités

La classe SafetyValidatorClient constitue la représentation de la couche de sécurité qui augmente une connexion normalisée sur le CP 2/3 ou CP 2/2 afin de transmettre les données de sécurité SIL3. Elle coordonne la production des données d'application avec une instance homologue de la classe SafetyValidatorClient.

7.5.5.2 Production de sécurité

Le présent paragraphe décrit la production des données et les interfaces entre l'application, le SafetyValidatorClient et la couche de liaison sous-jacente. Le comportement d'exécution de SafetyValidatorClient et ses interfaces constituent l'objet du présent paragraphe. L'application proprement dite doit présenter une intégrité élevée, ce qui peut être accompli de nombreuses manières et constitue une décision de conception finale du fournisseur.

La Figure 46 illustre graphiquement le flux de données entre le SafetyValidatorClient et une application relative à la sécurité de production. Safety_Data_Out est produit à une fréquence périodique, désignée comme l'Intervalle de paquets attendu (EPI). Le SafetyValidatorClient doit échantillonner, saisir et dater les données à transmettre à chaque période EPI.



Légende

Anglais	Français
safety producing application	Application de production de sécurité
safety message	Message de sécurité
link producer	Producteur de liaison
link consumer	Consommateur de liaison
safety Ack	Acq. De sécurité
run-time safety validator client	Exécution de Safety validator client
validator connection establishment engine	Unité d'établissement de connexion d'objet de validation

Figure 46 – Flux de données de production de sécurité

7.5.5.3 Interface d'application de production

L'application de production fournit le Safety_Data_Out au SafetyValidatorClient. Le SafetyValidatorClient doit élaborer les Mode_Octet, Actual_Data et Complement_Data, ainsi que la section de datation.

Le SafetyValidatorClient doit renvoyer l'état de connexion de sécurité à l'application de production pour chaque consommateur.

7.5.5.4 Logique de production des données de sécurité

Le présent paragraphe décrit la logique que tous les producteurs de données de sécurité doivent suivre. La mise en œuvre réelle de la logique peut varier, mais des résultats équivalents doivent être obtenus. La logique définie dans la présente partie suppose une application de production asynchrone qui met les données de sécurité à disposition du SafetyValidatorClient.

Les définitions relatives aux variables utilisées dans la logique de production de données de sécurité sont données en 7.6.

7.5.5.5 Exemple de logique d'amorçage (démarrage) à froid pour la production des données de sécurité

La logique de production dédiée à l'initiation d'un amorçage à froid d'une connexion est la suivante:

```

////////////////////////////////////
// Cold start after connection establishment processing
////////////////////////////////////
// This Logic should be executed at the transition of the
// producing connection from closed to open.
// For Point-to-point connections this logic may
// be performed any time the connection is Opened or Re-Opened.
////////////////////////////////////

Ping_Interval_EPI_Count = 0,
RR_Con_Num_Index_Pntr = Max_Consumer_Number,

// Initialize the ping count in the safety message to 0
Mode_Octet.Ping_Count = 0,

// Time_Drift_Per_Ping_Interval, the minimum value is 1
Time_Drift_Per_Ping_Interval =
Roundup(EPI * Ping_Interval_EPI_Multiplier / 320000,

FOR (Consumer_Num = 1 to Max_Consumer_Number),
{
    // Producer Dynamic Variables
    Consumer_Active_Idle[Consumer_Num-1] = Idle,
    S_Connection_Fault[Consumer_Num-1] = OK,
    Producer_Rcvd_Time_Value[Consumer_Num-1] = 0x0000,
    Consumer_Time_Correction_Value[Consumer_Num-1] = 0x0000,
    Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1] =
0x0000,
    Consumer_Time_Value[Consumer_Num-1] = 0x0000

    // Producer Derived Variables

    // Time_Drift_Constant, the minimum value is 1.
    (Time_Drift_Constant is not
    // saved but is used in the calculation of the
    // Connection_Correction_Constant below.)
    Time_Drift_Constant =
        Roundup((Timeout_Multiplier[Consumer_Num-1] +1) * EPI *
        Ping_Interval_EPI_Multiplier / 320000),

    // Connection_Correction_Constant
    Connection_Correction_Constant[Consumer_Num-1] =

```

```
Time_Drift_Constant + 1 - Time_Coord_Msg_Min_Multiplier [Consumer_Num-1],

// Time_Coord_Response_EPI_Limit
Time_Coord_Response_EPI_Limit[Consumer_Num-1]= Roundup((5000000 +
    (Time_Coord_Msg_Min_Multiplier[Consumer_Num-1]×128) +
    (EPI × ( Consumer_Num - 1))) / EPI),

// Time_Coord_Response_EPI_Limit has a maximum value of 1000
IF (Time_Coord_Response_EPI_Limit[Consumer_Num-1] > 1000),
THEN
{
    Time_Coord_Response_EPI_Limit[Consumer_Num-1] = 1000,
}
ENDIF
}
ENDFOR
```

7.5.5.6 Exemple de logique de réamorçage d'un consommateur multipoint pour la production de données de sécurité

Cette logique permet l'arrêt et le réamorçage de chaque consommateur multipoint, tandis que la production multipoint destinée à d'autres consommateurs se poursuit.

Le présent paragraphe définit la logique nécessaire pour réamorcer un consommateur individuel, mais il ne représente pas une mise en œuvre efficace de la méthode de détection du réamorçage de la connexion individuelle.

```
////////////////////////////////////
// Re-initialization of Multipoint production to individual consumers
////////////////////////////////////
// This Logic should be executed after a multipoint consumer performs
// a successful open to a multipoint producer while the production
// is in process to re-initialize production to that consumer number
////////////////////////////////////
FOR (Consumer_Num = 1 to Max_Consumer_Number),
{
    IF (Consumer_Open(Consumer_Num-1)transitions from Closed to Open)
    THEN
    {
        // Producer Dynamic Variables
        Consumer_Active_Idle[Consumer_Num-1] = Idle,
        S_Connection_Fault[Consumer_Num-1] = OK,
        Producer_Rcvd_Time_Value[Consumer_Num-1] = 0x0000,
        Consumer_Time_Correction_Value[Consumer_Num-1] = 0x0000,
        Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1]= 0x0000,

        // Producer Derived Variables
        // Time_Drift_Constant, the minimum value is 1
        //(Time_Drift_Constant is not
        // saved but is used in the calculation of the
        // Connection_Correction_Constant below.)
        Time_Drift_Constant =
        Roundup((Timeout_Multiplier[Consumer_Num-1] +1) × EPI ×
            Ping_Interval_EPI_Multiplier / 320000),
        // Connection_Correction_Constant
        Connection_Correction_Constant[Consumer_Num-1] =
        Time_Drift_Constant + 1 - Time_Coord_Msg_Min_Multiplier
        [Consumer_Num-1],
```

```

// Time_Coord_Response_EPI_Limit
Time_Coord_Response_EPI_Limit[Consumer_Num-1]= Roundup((5000000 +
    (Time_Coord_Msg_Min_Multiplier[Consumer_Num-1]×128) +
    (EPI × ( Consumer_Num - 2))) / EPI),

// Time_Coord_Response_EPI_Limit has a maximum value of 1000
IF (Time_Coord_Response_EPI_Limit[Consumer_Num-1] > 1000),
THEN
{
    Time_Coord_Response_EPI_Limit[Consumer_Num-1] = 1000,
}
ENDIF
}
ENDIF
}

```

7.5.5.7 Exemple de production de données combinée

La logique requise commune du SafetyValidatorClient pour toutes les connexions de sécurité est présentée ci-dessous.

```

////////////////////////////////////
// start of EPI Safety Data production processing
////////////////////////////////////
// This logic assumes that the application has passed the Safety
// Data and Application_Run_Idle to the Safety Validator.
//
////////////////////////////////////
// Capture the time to be used for the time stamp
Producer_Safe_Data_TS = Producer_Clk Count,

// Check if it is time to increment the Ping Count
Ping_Interval_EPI_Count = Ping_Interval_EPI_Count + 1,

IF (Ping_Interval_EPI_Count >= Ping_Interval_EPI_Multiplier),
THEN
{
    Increment Mode_Octet.Ping_Count,
    Ping_Interval_EPI_Count = 0,
}
ENDIF

// Set the remainder of the Mode_Octet bits
Mode_Octet.Run_Idle = Application_Run_Idle,
Mode_Octet.TBD_Bit = 0,
Mode_Octet.TBD_2_Bit = 0,

// Execute the safety producer, connection type specific functions
IF (Connection_Type == multipoint),
THEN
{
    multi_cast_producer_function(),
}
ELSE
{
    single_cast_producer_function(),
}
ENDIF

// set Mode_Octet redundant bits (2:4) to the correct values
Mode_Octet = (Mode_Octet AND 0xe3)OR((Mode_Octet>>3) AND 0x1c XOR
0x14),

```

```
Calculate Data Section CRC,  
Calculate Time_Stamp Section CRC,  
Trigger the sending of the Data Message,  
  
// If on FSCP 2 on CP 2/3 and Multipoint, send the Time Correction  
// Message if it is time.  
IF ((Connection_Type == CP 2/3) AND  
(Send_Time_Correction_Message == 1)),  
THEN  
{  
    Trigger the sending of the Time Correction Message,  
    Send_Time_Correction_Message = 0,  
}  
ENDIF  
  
/////////////////////////////////////  
// end of safety data production processing  
/////////////////////////////////////  
  
/////////////////////////////////////  
// Start of point-to-point production function  
/////////////////////////////////////  
single_cast_producer_function()  
// Check if a Time Coordination message has been received in  
// the allotted time.  
IF (Ping_Interval_EPI_Count == 8),  
THEN  
{  
    Increment Ping_Int_Since_Last_Time_Coord_Msg_Count[0],  
    IF (Ping_Int_Since_Last_Time_Coord_Msg_Count[0]  
>= (Timeout_Multiplier[0] + 2)),  
        THEN  
        {  
            S_Connection_Fault[0]=Faulted,  
        }  
    ENDIF  
}  
ENDIF
```

```
/////////////////////////////////////  
// Hold Data_Time_Stamp = 0 until time coordination msg received  
IF(Consumer_Active_Idle[0] == Idle),  
THEN  
{  
    Mode_Octet.Run_Idle = Idle,  
    Time_Stamp_Section.Data_Time_Stamp = 0x0000,  
}  
ELSE  
{  
    Time_Stamp_Section.Data_Time_Stamp = Producer_Safe_Data_TS +  
    Consumer_Time_Correction_Value[0],  
}  
ENDIF
```

```
END single_cast_producer_function()  
/////////////////////////////////////  
// end of point-to-point processing  
/////////////////////////////////////  
  
/////////////////////////////////////  
// Start of multipoint production function  
/////////////////////////////////////>
```

```

multi_cast_producer_function()
// Check if its time to start sending Time Correction messages
// Time Correction messages are sent starting at the 8th EPI
production
// within the Ping interval
IF (Ping_Interval_EPI_Count == 8),
THEN
{
RR_Con_Num_Index_Pntr = 0,
}
ENDIF

// Check if a Time Coordination message has been received in
// the allotted time.
IF (RR_Con_Num_Index_Pntr < Max_Consumer_Number),
THEN
{
Increment
Ping_Int_Since_Last_Time_Coord_Msg_Count[RR_Con_Num_Index_Pntr],
IF (Ping_Int_Since_Last_Time_Coord_Msg_Count[RR_Con_Num_Index_Pntr]
    >= (Timeout_Multiplier[RR_Con_Num_Index_Pntr] + 2)),
THEN
{
Set S_Connection_Fault[RR_Con_Num_Index_Pntr]=Faulted,
}
ENDIF
ENDIF

// Check if its time to send an active non-error Time Correction
// message
IF ((RR_Con_Num_Index_Pntr < Max_Consumer_Number) AND
    (Consumer_Active_Idle[RR_Con_Num_Index_Pntr] == Active)), AND
    (Consumer_Open[RR_Con_Num_Index_Pntr] == Open) AND
    (S_Connection_Fault[RR_Con_Num_Index_Pntr] == OK)
THEN
{
Send_Time_Correction_Message = 1,
Mcast_Octet.Consumer_# = RR_Con_Num_Index_Pntr + 1,
Mcast_Octet.Multi_Cast_Active_Idle =
Consumer_Active_Idle[RR_Con_Num_Index_Pntr],
Time_Correction_Section.Time_Correction_Value =
Consumer_Time_Correction_Value[RR_Con_Num_Index_Pntr],
Set Mcast_Octet.Parity_Even to the correct value,
MCast_Octet_2 = ((MCast_Octet XOR 0xFF) AND 0x55) OR
(MCast_Octet AND 0xAA)
Calculate CRC of time correction message,
}
ELSE
{
//load null TIME_CORRECTION Section,
Mcast_Octet.Consumer_# = 0,
Mcast_Octet.Multi_Cast_Active_Idle = Idle,
Time_Correction_Section.Time_Correction_Value = 0,
Set Mcast_Octet.Parity_Even to the correct value,
MCast_Octet_2 = ((MCast_Octet XOR 0xFF) AND 0x55) OR
(MCast_Octet AND 0xAA)
Calculate CRC of time correction section,
}
ENDIF

// If Time Correction messages are being sent, increment to the
// next consumer
IF (RR_Con_Num_Index_Pntr < Max_Consumer_Number),

```

```

THEN
{
RR_Con_Num_Index_Pntr = RR_Con_Num_Index_Pntr + 1,
}
ENDIF

// Set the time stamp value
Time_Stamp_Section.Data_Time_Stamp = Producer_Safe_Data_TS,
END multi_cast_producer_function()
////////////////////////////////////
// end of multipoint processing
////////////////////////////////////

```

7.5.5.8 Exemple de logique de réception des messages de coordination temporelle

Le SafetyValidatorClient doit accomplir les étapes suivantes pour chaque message de coordination temporelle reçu. Pour les connexions de sécurité point à point et multipoint, le message de coordination temporelle reçoit une Consumer_Time_Value.

Les producteurs doivent permettre la réception des réponses de coordination temporelle au cours du Ping_Interval ou du Ping_Interval +1.

```

////////////////////////////////////
//Time Coordination message reception processing
////////////////////////////////////
// Consumer_Num equals 1 for point-to-point
// Consumer_Num equals the consumer the message
// was received from for multipoint
////////////////////////////////////
// This Logic should only be executed if Ack_Octet.Ping_Response
// of the message is equal to 1 AND Consumer_Open[Consumer_Num-1]
// is equal to Open. If Ack_Octet.Ping_Response is equal to 0 OR
// Consumer_Open[Consumer_Num-1] is equal to Closed, this
// message should be ignored.
////////////////////////////////////
// reject Time Coordination packets if CONSUMER_TIMESTAMP has
// previously been captured AND this is not the first time coordination
// message from this Consumer
IF ( (Consumer_Time_Value[Consumer_Num-1] ==
Time_Coordination_Section.Consumer_Time_Value) AND
(Consumer_Active_Idle[Consumer_Num-1] == Active) )
THEN
{
// the only time this would be the case is if that Time
// Coordination was already received
Return
}
ELSE
{
// Capture the received CONSUMER time
Consumer_Time_Value[Consumer_Num-1] =
Time_Coordination_Section.Consumer_Time_Value;
}
ENDIF

// Capture the received time
Producer_Rcvd_Time_Value[Consumer_Num-1] = Producer_Clk_Count,
}
ENDIF

// Perform the integrity checks

```



```

Check the CRC of the Time Coordination Section,
IF ((Ack_Octet_parity_incorrect) OR (Ack_Octet_2 !=
    (((Ack_Octet XOR 0xFF) AND 0x55) OR (Ack_Octet AND 0xAA))))
THEN
{
Ack_Octet_Error = true,
}
ELSE
{
Ack_Octet_Error = false,
}
ENDIF
IF ((CRC_Error) OR (Ack_Octet_Error = true) OR
// Ensure that the Time Coordination message returned
// with the same ping interval or the next ping interval.
// The next ping interval is allowed for the case that a
// multipoint consumer connects to an existing producer.
// If the new consumer receives its first message near the end
// of the ping interval, the time coordination may arrive back
// at the producer during the next ping interval.
    ((Ack_Octet.Ping_Count_Reply != Ping_Count) AND
    (Ack_Octet.Ping_Count_Reply != Ping_Count-1)) OR
// Ensure that the Time Coordination message returned
// within the approximately 5 second limit
((Consumer_Active_Idle[Consumer_Num-1] == Active) AND
    (Ping_Interval_EPI_Count >
        Time_Coord_Response_EPI_Limit[Consumer_Num-1])),
THEN
{
S_Connection_Fault[Consumer_Num-1] = Faulted,
}
ENDIF

// Determine the worst case Consumer_Time_Correction_Value based on
the
// received Time_Coordination section
Worst_Case_Consumer_Time_Correction_Value[Consumer_Num-1] =
    Time_Coordination_Section.Consumer_Time_Value
    - Producer_Rcvd_Time_Value[Consumer_Num-1]
- Connection_Correction_Constant[Consumer_Num-1],

// Determine the worst case time drift since the last
// Time_Coordination information was received. This is equal to the
// number of Ping Intervals since the last Time_Coordination message
// times the Time_Drift_per_Ping_Interval plus 1 for the asynchronous
// clocks. This value will not be used until the second
// Time_Coordination message is received.

Time_Drift_Since_Last_Time_Coord =
    ({Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1]+ 1}
    × Time_Drift_Per_Ping_Interval) + 1,

// If this Time_Coordination Transport delay is greater than the
// last Time_Coordination Transport delay by more than the
appropriate
// number of Time_Drift_Per_Ping_Intervals, then, use the previous
// Correction value minus the appropriate number of
// Time_Drift_Per_Ping_Intervals. This action is not taken until the
// second Time_Coordination message is received.
// The math for this check should be 16 bit or greater
IF ((Consumer_Active_Idle[Consumer_Num-1] = Active) AND
    ((Consumer_Time_Correction_Value[Consumer_Num-1]-
        Worst_Case_Consumer_Time_Correction_Value[Consumer_Num-1] -

```

```

    Time_Drift_Since_Last_Time_Coord) AND 0x8000 == 0)),
THEN
{
Consumer_Time_Correction_Value[Consumer_Num-1] =
Consumer_Time_Correction_Value[Consumer_Num-1]
Time_Drift_Since_Last_Time_Coord,
}
ELSE
{
Consumer_Time_Correction_Value[Consumer_Num-1] =
Worst_Case_Consumer_Time_Correction_Value[Consumer_Num-1],
}
ENDIF

// Set the flag indicating a Time_Coordination message has been
received
Consumer_Active_Idle[Consumer_Num-1] = Active,

// Reset the Time_Coordination message timer
Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1] = 0,
////////////////////////////////////
// end producer time coordination information reception processing
////////////////////////////////////

```

7.5.6 Définition de la fonction SafetyValidatorServer

7.5.6.1 Généralités

La fonction SafetyValidatorServer constitue la couche de sécurité qui augmente une connexion normalisée sur le CP 2/3 ou le CP 2/2 afin de transmettre les données SIL3. Elle coordonne la réception des données de sécurité avec une instance homologue de la classe du SafetyValidatorClient et fournit ces données à l'application relative à la sécurité de consommation.

7.5.6.2 Consommation de sécurité

Le présent paragraphe décrit la consommation des données et les interfaces entre l'application, le SafetyValidatorServer et la couche de liaison sous-jacente.

Il existe deux versions de contrôle des consommateurs de données de sécurité. Il s'agit des:

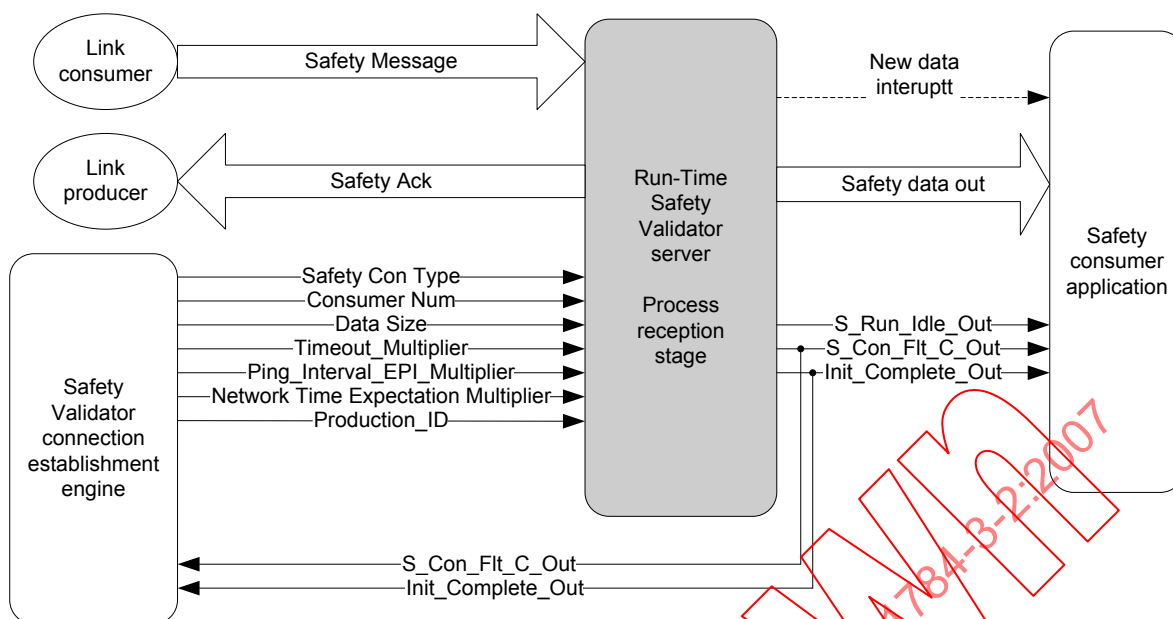
- SafetyValidatorServer – Déclenché par la liaison .
Ceci convient aux applications de consommation synchronisées sur la réception des données, ou aux applications de consommation qui présentent un retard continu par rapport à la réception des données.
- SafetyValidatorServer – Déclenché par l'application.
Ceci convient aux applications de consommation à caractère périodique et non synchronisées sur la réception des données.

Dans les deux versions, le SafetyValidatorServer doit exécuter les aspects suivants du contrôle des données de sécurité:

- coordination temporelle avec le producteur;
- vérification de l'intégrité de la section de datation;
- vérification de l'intégrité des données;
- vérification du délai du réseau.

7.5.6.3 SafetyValidatorServer - déclenché par la liaison

La Figure 47 illustre les flux de données applicables au SafetyValidatorServer dans le mode déclenché par la liaison.



Légende

Anglais	Français
link consumer	Consommateur de liaison
safety message	Message de sécurité
new data interruptt	Interruption des nouvelles données
link producer	Producteur de liaison
safety Ack	Acq. de sécurité
Run-time safetyValidatorServer	Safetyvalidatorserver d'exécution
process reception stage	Étape de traitement de réception
safety consumer application	Application de consommation de sécurité
safety validator connection establishment engine	Unité d'établissement de connexion de l'objet de validation de sécurité

Figure 47 – Contrôle des données de sécurité de consommation

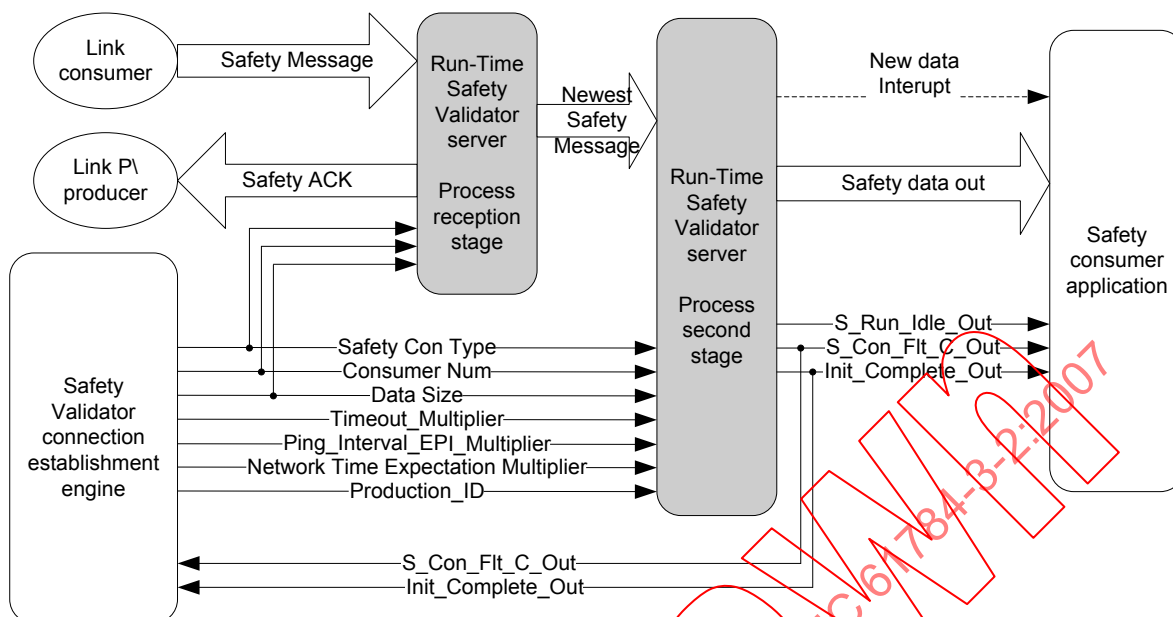
Le SafetyValidatorServer déclenché par la liaison doit exécuter tous les aspects du protocole de sécurité de chaque message reçu.

Pour les connexions multipoint, le SafetyValidatorServer déclenché par la liaison doit exécuter également tous les aspects du protocole de sécurité de chaque section Time_Correction reçue.

Le SafetyValidatorServer doit s'assurer de la contre-vérification des données de sécurité présentées à l'application de consommation et du respect du délai du réseau.

Le SafetyValidatorServer déclenché par la liaison doit régler le nouveau drapeau de données à chaque fois que les données sont prêtes à être utilisées.

7.5.6.4 SafetyValidatorServer – déclenché par l'application



Légende

Anglais	Français
link consumer	Consommateur de liaison
safety message	Message de sécurité
new data interruptt	Interruption des nouvelles données
safety Ack	Acc. de sécurité
Run-time safetyValidatorServer	SafetyValidatorServer d'exécution
safety consumer application	Application de consommation de sécurité
safety validator connection establishment engine	Unité d'établissement de connexion de l'objet de validation de sécurité
newest safety message	Nouveau message de sécurité
link P producer	Liaison P-producteur
process reception stage	Etape de réception de traitement
safety data out	Sortie des données de sécurité
process second stage	Deuxième étape de traitement

Figure 48 – SafetyValidatorServer – déclenché par l'application

Le SafetyValidatorServer déclenché par l'application ne traite pas entièrement chaque message.

L'étape de réception de traitement du SafetyValidatorServer déclenché par l'application doit vérifier le comptage de demandes Ping pour tous les messages de données entrants.

Pour la consommation multipoint des SafetyValidatorServers déclenchés par l'application, la toute dernière section Time_Correction reçue doit être transmise à l'application de consommation.

L'application de consommation relative à la sécurité doit solliciter de nouvelles données auprès du SafetyValidatorServer afin que ce dernier effectue le traitement d'un message.

Dans le cas des applications périodiques non synchronisées sur la réception des données, ceci réduit la charge de traitement, dans la mesure où la section de données fait uniquement l'objet d'une contre-vérification (dans la deuxième étape de traitement de la Figure 48) pour les messages de données de sécurité utilisés. Ceci est particulièrement utile pour le cas où les EPI de réseau présentent une fréquence plus rapide que la fréquence périodique de l'application de consommation.

L'application de consommation doit s'assurer qu'elle obtient de nouvelles données au moins une fois toutes les 2 s. afin de s'assurer que la datation n'est pas remplacée.

7.5.6.5 Logique de réception des données de sécurité

Le processus de réception des données de sécurité comprend:

- un contrôle Ping_Count;
- un contrôle de la section de datation;
- un contrôle de l'intégrité des données;
- un contrôle de la datation; et
- la déduction de Run_Idle et de l'état de panne.

Les définitions des variables utilisées dans la logique de réception des données de sécurité sont données en 7.6.8.

Le Tableau 80 définit les étapes de traitement des données de sécurité exécutées par le SafetyValidatorServer déclenché par la liaison.

Tableau 80 – Réception des données – Déclenchée par la liaison

	Point à point	Multipoint
Contrôle de Ping_Response	Non	Non
Contrôle de Ping_Count	Oui	Oui
Contrôle du CRC de section de datation	Oui	Oui
Contrôle de Run_Idle	Oui	Oui
Contrôle de TBD_Bit	Oui	Oui
Contrôle des CRC de données	Oui	Oui
Contrôle des données réelles/complémentaires	Oui	Oui
Déduction de Data_Time_Stamp	Oui	Oui

Le Tableau 81 définit les étapes de traitement des données de sécurité qui doivent être exécutées par le SafetyValidatorServer déclenché par la liaison pour la réception de chaque section de correction de temps.

Tableau 81 – Réception Time_Correction – Déclenchée par la liaison

Multipoint	Multipoint
Contrôle MCast_Octet et MCast_Octet_2	Oui, si adaptation de consumer_#
Contrôle du CRC de Time_Correction	Oui, si adaptation de consumer_#
Obtention de Time_Correction_Value	Oui, si adaptation de consumer_#
Contrôle de Multi_Cast_Active_Idle	Oui, si adaptation de consumer_#

Le Tableau 82 définit les étapes de traitement des données de sécurité exécutées à la réception de chaque message par le SafetyValidatorServer déclenché par l'application.

Tableau 82 – Réception des données – Déclenchée par l'application

	Point à point	Multipoint
Contrôle de Ping_Response	Non	Non
Contrôle de Ping_Count	Oui	Oui
Contrôle du CRC de section de datation	Non	Non
Contrôle de Run_Idle	Non	Non
Contrôle de TBD_Bit	Non	Non
Contrôle des CRC de données	Non	Non
Contrôle des données réelles/complémentaires	Non	Non
Déduction de Data_Time_Stamp	Non	Non

Le Tableau 83 définit les étapes de traitement des données de sécurité qui doivent être exécutées à réception de chaque section Time_Correction par les SafetyValidatorServers déclenchés par l'application.

Tableau 83 – Réception de Time_Correction – Déclenchée par l'application

	Multipoint
Contrôle de MCast_Octet et MCast_Octet_2	Transmission du message si adaptation de consumer_#
Contrôle du CRC de Time_Correction	Non
Obtention de Time_Correction_Value	Non
Contrôle de Multi_Cast_Active_Idle	Non

Le Tableau 84 définit les étapes de traitement des données de sécurité exécutées par le SafetyValidatorServer déclenché par l'application, en cas de sollicitation des données de sécurité.

Tableau 84 – Application de consommation – Contrôle des données de sécurité

	Point à point	Multipoint
Contrôle de Ping_Response	NA	NA
Contrôle de Ping_Count	NA	NA
Contrôle du CRC de section de datation	Oui	Oui
Contrôle de MCast_Octet et MCast_Octet_2	NA	Oui (à partir de la dernière correction)
Contrôle du CRC de Time_Correction	NA	Oui (à partir de la dernière correction)
Obtention de Time_Correction_Value	NA	Oui (à partir de la dernière correction)
Contrôle de Multi_Cast_Active_Idle	NA	Oui (à partir de la dernière correction)
Contrôle de Run_Idle	Oui	Oui
Contrôle de TBD_Bit	Oui	Oui
Contrôle des CRC de données	Oui	Oui
Contrôle des données réelles/complémentaires	Oui	Oui
Déduction de Data_Time_Stamp	Oui	Oui

7.5.6.6 Contrôle des comptages de demandes Ping

Le SafetyValidatorServer doit contrôler le Ping_Count à chaque message. Le contrôle de Ping_Count vérifie le comptage des demandes Ping pour déterminer s'il a varié.

Si un SafetyValidatorServer détecte une variation du comptage des demandes Ping, il doit produire un message de coordination temporelle dans cet intervalle Ping ou dans un délai de 5 s, selon la plus petite durée.

Il convient de ne pas traiter les erreurs de contrôle de Ping_Count comme des défaillances dangereuses. Il n'est pas nécessaire de vérifier le CRC de la section de datation pour effectuer le contrôle de Ping_Count.

7.5.6.7 Intervalle maximal de vérification des données et du délai du réseau

L'intégrité des données doit faire l'objet d'un contrôle au moins une fois toutes les 2 s.

La datation et le délai du réseau (NTE) doivent faire l'objet d'un contrôle au moins une fois toutes les 2 s.

7.5.6.8 Exemple d'initialisation d'amorçage à froid

```

////////////////////////////////////
// Cold start after connection establishment processing
////////////////////////////////////
// This Logic should be executed at the transition of the
// consuming connection from closed to open.
////////////////////////////////////

S_Con_Flt_C_Out = OK,
Init_Complete_Out = 0,
S_Run_Idle_Out = Idle,

Last_Ping_Count = 3,
Time_Coordination_Count_Down = 0x0000,

Corrected_Data_Time_Stamp = 0x0000,
Last_Data_Time_Stamp = 0x0000,
Last_Rcvd_Multi_Cast_Active_Idle = Idle,
Last_Rcvd_Time_Correction_Value = 0x0000,
Time_Correction_Ping_Interval_Count = 0x0000,
Time_Correction_Received_Flag = 0,
Data_Age = 0x0000,
Max_Data_Age = 0x0000,
////////////////////////////////////
// end, Cold start after connection establishment processing
////////////////////////////////////

```

7.5.6.9 SafetyValidatorServer – logique déclenchée par la liaison

Les paragraphes 7.5.6.10 et 7.5.6.11 décrivent la logique de réception des données de sécurité du consommateur pour le SafetyValidatorServer – option déclenchée par la liaison. L'expression « Déclenché par la liaison » signifie que la logique dans son ensemble est exécutée à chaque réception.

Le paragraphe 7.5.6.10 décrit la logique combinée associée à la réception des paquets de données de sécurité pour les types de connexion de sécurité point à point et multipoint.

Le paragraphe 7.5.6.11 décrit la logique associée à la réception des messages de correction de temps pour le type de connexion de sécurité multipoint.

Cette logique doit être exécutée uniquement en cas d'ouverture de la connexion sans aucune erreur.

7.5.6.10 Exemple de logique de réception combinée – logique déclenchée par la liaison

La logique suivante est la logique de réception de SafetyValidatorServer requise pour la réception des paquets de données de sécurité du type de connexion de sécurité point à point ou multipoint, pour le SafetyValidatorServer – Logique déclenchée par la liaison.

Les connexions multipoint doivent transmettre un message de coordination temporelle (Consumer_Num – 1) après variation du comptage de demandes Ping.

Les consommateurs multipoint qui se connectent à un producteur existant doivent transmettre le message Time_Coordination immédiatement à la première réception valide.

```

////////////////////////////////////
// Safety Data Consumption - link triggered logic
////////////////////////////////////

// Check if its time to send a Time Coordination Message
// Multipoint connections send a Time Coordination Message
// (Consumer_Num - 1) receptions after the ping count change.
IF (received Mode_Octet.Ping_Count != Last_Ping_Count),
THEN
{
    Time_Coordination_Count_Down = Consumer_Num,
    Increment Time_Correction_Ping_Interval_Count,
}
ENDIF

// For multipoint-consumers connecting to an existing producer,
send
// the Time_Coordination message immediately upon the 1st reception.
IF (It is the first data reception),
THEN
{
    Produce Time_Coordination_Message,
    Time_Coordination_Count_Down = 0,
}
ENDIF
// Check if a ping count change has been detected and the Time
// Coordination message has not been sent yet.
IF (Time_Coordination_Count_Down > 0),
// Send the Time_Coordination message if the decremented
// Time_Coordination_Count_Down transitions to 0
THEN
{
    decrement Time_Coordination_Count_Down,
    IF (Time_Coordination_Count_Down == 0),
    THEN
    {
        Produce Time_Coordination_Message,
    }
    ENDIF
}
ENDIF
// Save ping count for next reception check
Last_Ping_Count = Mode_Octet.Ping_Count,

// check for data integrity faults
Temp_Fault_Flag = OK,
IF ((Time_Stamp_Section_CRC != OK) OR (Actual_Data_Section_CRC != OK)
OR (Complement_Data_Section_CRC != OK)
OR (Mode_Octet.Run_Idle != not Mode_Octet.N_Run_Idle)
OR (Mode_Octet.TBD_2_Bit !=

```



```

Mode_Octet.TBD_2_Bit_Copy)
OR (Mode_Octet.TBD_Bit != not Mode_Octet.N_TBD_Bit)
OR (Actual_Data != not Complement_Data) [for > 2 octets only]
),
THEN
{
Temp_Fault_Flag = Faulted,
}
ENDIF

// execute the safety connection type specific functions
IF (Connection_Type == Multipoint),
THEN
{
multi_point_consumer_function(),
}
ELSE
{
single_point_consumer_function(),
}
ENDIF

// Set up Time_Stamp_Delta, to be used to check for out of sequence
// messages Time_Stamp_Delta is not calculated until the EPI
// reception after Init_Complete_Out is set to 1.
IF (Init_Complete_Out == 0),
THEN
{
Time_Stamp_Delta = 1,
}
ELSE
{
Time_Stamp_Delta = Time_Stamp_Section.Data_Time_Stamp -
Last_Data_Time_Stamp,
}
ENDIF

// Save Time stamp for next reception sequence check
Last_Data_Time_Stamp = Time_Stamp_Section.Data_Time_Stamp,

// Calculate the worst case age
IF (Init_Complete_Out_Temp == 1),
THEN
{
Data_Age = Consumer_Clk_Count - Corrected_Data_Time_Stamp,
}
ENDIF

// Calculate the maximum worst case age
IF (Data_Age > Max_Data_Age)
THEN
{
Max_Data_Age = Data_Age,
}
ENDIF

// Check for Age, Integrity, and out of sequence faults
IF (( unsigned_int_16( Data_Age > Network_Time_Expectation_Multiplier)
AND (Init_Complete_Out_Temp == 1))
OR (Temp_Fault_Flag == Faulted)
OR (Time_Stamp_Delta < 0)
OR (Time_Stamp_Delta > Network_Time_Expectation_Multiplier))
THEN

```

```

{
    S_Con_Flt_C_Out_Temp = Faulted,
}
ENDIF

////////////////////////////////////
// The following information is passed on to the consuming application
//   with block integrity: Safety
Data, Init_Complete_Out, S_Run_Idle_Out
//   and S_Con_Flt_C_Out.
// The time between the checking of the age of the data and passing
// the data off to the consuming application shall be protected under
// the time_stamp check or the consuming application time checks.
////////////////////////////////////
Make Safety Data Available to Consuming Application
Init_Complete_Out = Init_Complete_Out_Temp,
S_Run_Idle_Out    = S_Run_Idle_Out_Temp,
S_Con_Flt_C_Out   = S_Con_Flt_C_Out_Temp,
////////////////////////////////////
// end Safety Data Consumption - link triggered logic
////////////////////////////////////

////////////////////////////////////
// Start of point-to-point consumer function
////////////////////////////////////
single_cast_consumer_function()
// For point-to-point multipoint, no time stamp correction is required
Corrected_Data_Time_Stamp = Time_Stamp_Section.Data_Time_Stamp,

// Determine the Init_Complete_Out state to be passed on to the
// consuming application.
// For point-to-point or bi-dir, a time_stamp other than 0 or a Run
// indication, indicate that the Time_Coordination_Message has been
// received by the Client and the corrected time stamp is being used.
IF ((Corrected_Data_Time_Stamp != 0) OR (Mode_Octet.Run_Idle == Run)),
THEN
{
    Init_Complete_Out_Temp = 1,
}
ENDIF

// Determine the Run_Idle indication to be passed on to the consuming
// application.
IF ((Mode_Octet.Run_Idle == Idle) OR (Init_Complete_Out_Temp == 0))
THEN
{
    S_Run_Idle_Out_Temp = Idle,
}
ELSE
{
    S_Run_Idle_Out_Temp = Run,
}
ENDIF

END single_cast_consumer_function()
////////////////////////////////////
// end of point-to-point processing
////////////////////////////////////

////////////////////////////////////
// Start of multipoint consumer function
////////////////////////////////////

```

```

multi_cast_consumer_function()
// If multipoint, check that a time correction message has been
// received within the last Timeout_Multiplier + 1 Ping Intervals
IF (Time_Correction_Ping_Interval_Count > (Timeout_Multiplier + 1))
THEN
{
    Temp_Fault_Flag = Faulted,
}
ENDIF

// If multipoint, add correction value to time stamp
Corrected_Data_Time_Stamp = Time_Stamp_Section.Data_Time_Stamp +
    Last_Rcvd_Time_Correction_Value
// Determine the Init_Complete_Out state to be passed on to the
// consuming application.
// For multipoint, the reception of a Time_Coorection Message,
indicate
// that data may be used.
IF (Time_Correction_Received_Flag == 1)
THEN
{
    Init_Complete_Out_Temp = 1,
}
ENDIF
// Determine the Run_Idle indication to be passed on to the consuming
// application.
IF ((Mode_Octet.Run_Idle == Idle) OR
    (Init_Complete_Out_Temp == 1) OR
    (Time_Correction_Received_Flag == 1) OR //used only on
multipoint
    (Last_Rcvd_Multi_Point_Active_Idle==Idle))//used only on
multipoint
THEN
{
    S_Run_Idle_Out_Temp = Idle,
}
ELSE
{
    S_Run_Idle_Out_Temp = Run,
}
ENDIF
end multi_point_consumer_function()
////////////////////////////////////
// end of multipoint processing
////////////////////////////////////

```

7.5.6.11 Exemple de réception d'un message de correction de temps – logique déclenchée par la liaison

La logique suivante est la logique SafetyValidatorServer requise associée à la réception du message de correction de temps pour le type de connexion de sécurité multipoint, pour le consommateur – déclenché par la liaison.

```

////////////////////////////////////
// Consumer processing - Time Correction Message reception
////////////////////////////////////
// Is the Time Correction message for this Consumer ?
IF (MCast_Octet.Consumer_Num = Consumer_Num)
    // Perform integrity checks.
THEN
{
    IF (Time Coordination message CRC incorrect)
    OR (MCast_Octet parity incorrect)

```

```

OR {MCast_Octet_2 != (((MCast_Octet XOR 0xFF) AND 0x55)
    OR (MCast_Octet AND 0xAA))}
    OR { (MCast_Octet.Multi_Cast_Active_Idle == Idle)
AND (Last_Rcvd_Multi_Cast_Active_Idle == Active)},
    // only one transition from connection idle to active will
    // be allowed, to prevent an erroneous idle indication,
    // a transistion from active to idle will be latched into
    // a fault state
    THEN
    {
    S_Con_Flt_C_Out = Faulted,
    }
ENDIF

```

```

    // Save the Time_Correction_Value.
    Last_Rcvd_Time_Correction_Value =
Time_Correction_Section.Time_Correction_Value,

    // Save the Multi_Cast_Active_Idle bit.
    Last_Rcvd_Multi_Cast_Active_Idle =
MCast_Octet.Multi_Cast_Active_Idle,

    // Set the Time_Correction_Received_Flag.
    Time_Correction_Received_Flag = 1,

    // Reset the Time Correction timeout counter
    Time_Correction_Ping_Interval_Count = 0,
ENDIF
////////////////////////////////////
// end - Time Correction Message reception
////////////////////////////////////

```

7.5.6.12 SafetyValidatorServer - logique déclenchée par l'application

Pour le SafetyValidatorServer – déclenché par l'application, la même logique décrite ci-dessus doit être appliquée par la combinaison du SafetyValidatorServer, mais uniquement lorsque sollicité par l'application de consommation. Les mises en œuvre déclenchées par l'application traitent uniquement les paquets de sécurité lorsque l'application le nécessite. Cette stratégie de mise en œuvre est utile pour les applications dont la fréquence d'exécution est habituellement inférieure à une fréquence d'EPI typique. Les fonctions peuvent être réparties tel que défini dans le Tableau 81 et le Tableau 84.

Les étapes de traitement de protocoles définies dans le Tableau 84 ne sont à exécuter qu'avec les messages utilisés par l'application de consommation.

7.5.6.13 Exemple de logique de production des messages de coordination temporelle

Le SafetyValidatorServer doit exécuter les étapes suivantes à chaque envoi d'un message de coordination temporelle. Pour les connexions de sécurité point à point et multipoint, les données de coordination temporelle sont transmises dans le message de coordination temporelle. Cette logique est décrite également dans la single_cast_function() de la section de logique de production combinée.

```

////////////////////////////////////
//Time Coordination message production processing
////////////////////////////////////
// Set the Ping_Count_Reply to the value captured at the ping request
Ack_Octet.Ping_Count_Reply = Mode_Octet.Ping_Count,

// Set the Ping_Response bit to 1
Ack_Octet.Ping_Response = 1,

// Set the Consumer_Time_Value to the producers clock value

```

```

Time_Coordination_section.Consumer_Time_Value = Producer_Clk_Count,

// Add data integrity measures
Set Ack_Octet.Reserved(6:4) = 0,
Set Ack_Octet.Reserved(2) = 0,
Set Ack_Octet.Parity_Even bit to the correct value,
Ack_Octet_2 !=(((Ack_Octet XOR 0xFF)AND 0x55)OR
               (Ack_Octet AND 0xAA)),
Calculate CRC on Time Coordination message,
////////////////////////////////////
// end time coordination message send logic
////////////////////////////////////

```

7.6 Spécifications des messages de sécurité et des données de protocole

7.6.1 Octet de mode

7.6.1.1 Mode_Octet.Run_Idle

La valeur nulle du bit Run_Idle doit indiquer l'état Repos (Idle), et la valeur 1 du bit Run_Idle doit indiquer l'état Exécution (Run).

Pour les connexions point à point, le bit Run_Idle doit être réglé sur l'état Repos si l'application de production est « au repos » ou si le producteur attend les informations concernant les messages de coordination temporelle à recevoir au début de la production des données.

Si ces informations ont été reçues, le bit Run_Idle doit être réglé sur l'état Run_idle indiqué par l'application de production.

Pour les connexions multipoint, le bit Run_Idle doit être réglé sur l'état Run_Idle indiqué par l'application de production.

7.6.1.2 Mode_Octet.N_Run_Idle

Le bit complémentaire N_Run_Idle doit toujours compléter le bit Run_Idle.

7.6.1.3 Mode_Octet.TBD_2_Bit

Le TBD_2_bit Réserve Mode_Octet doit être réglé sur 0 par le producteur.

7.6.1.4 Mode_Octet.TBD_2_Copy

Le bit TBD_2_Copy doit toujours être égal au bit TBD_2.

7.6.1.5 Mode_Octet.Ping_Count

Le consommateur utilise le Ping_Count pour déterminer le moment où il convient d'envoyer un message de coordination temporelle.

7.6.1.6 Mode_Octet.TBD_Bit

Le TBD_bit Réserve Mode_Octet doit être réglé sur 0 par le producteur. Les bits réservés doivent être ignorés par le consommateur, sauf pour le contrôle du CRC et le contrôle réel/complémentaire de la paire TBD_Bit/N_TBD_Bit.

7.6.1.7 Mode_Octet.N_TBD_Bit

Le bit complémentaire N_TBD_Bit doit toujours compléter le TBD_Bit.

7.6.2 Section de datation

Time_Stamp.Section.Data_Time_Stamp: la Data_Time_Stamp correspond à l'heure de prélèvement des données pour cette production.

Pour les connexions point à point, la Data_Time_Stamp est définie par rapport au temporisateur placé sur le consommateur. Pour les connexions multipoint, la Data_Time_Stamp est définie par rapport au temporisateur placé sur le producteur.

La Data_Time_Stamp est toujours exprimée en incréments de 128 μ s.

7.6.3 Message de coordination temporelle

7.6.3.1 Bit Ack_Octet.Ping_Response

Le bit Ping_Response doit indiquer qu'une réponse à demande Ping est transmise par le consommateur des données de sécurité sur ce nœud, à la production des données.

Si le bit Ping_Response est égal à 1, il doit indiquer que les Consumer_Time_Value et Ping_Count_Reply sont valides pour cette production des données.

Si le bit Ping_Response bit est égal à 0, il doit indiquer que les Consumer_Time_Value et Ping_Count_Reply ne sont pas valides pour cette production des données, et qu'il convient de les ignorer.

7.6.3.2 Ack_Octet.Consumer_Time_Value

La Consumer_Time_Value doit être la valeur d'horloge du consommateur des données de sécurité au moment de la transmission du message.

La Consumer_Time_Value permet de déduire la Data_Time_Stamp pour la production de données dans les autres directions.

7.6.3.3 Ack_Octet.Ping_Count_Reply

La Ping_Count_Reply est utilisée par chaque consommateur pour indiquer la demande Ping à laquelle il est répondu.

La Ping_Count_Reply est égale à la valeur du Ping_Count auquel il est répondu.

7.6.3.4 Bits réservés de coordination temporelle Ack_Octet

Le consommateur doit régler les bits réservés de coordination temporelle sur la valeur 0.

Le producteur doit ignorer les bits réservés sauf pour le contrôle du CRC et de Ack_Octet_2.

7.6.4 Message de correction de temps

7.6.4.1 Mcast_Octet.Consumer_#

Pour les données à production multipoint, les bits Multi_Cast_Active_Idle et Consumer_Time_Correction_Value doivent être dirigés vers le consommateur indiqué par le champ Consumer_# du message.

Le Consumer_# est dirigé vers un consommateur spécifique (1 à 15).

Les bits Multi_Cast_Active_Idle et Consumer_Time_Correction_Value doivent être appliqués par le consommateur indiqué par le Consumer_#.

Un Consumer_# égal à 0 doit être utilisé pour indiquer l'absence de transfert des bits Multi_Cast_Active_Idle et Consumer_Time_Correction_Value si nécessaire.

7.6.4.2 Time_Correction_Section.Consumer_Time_Correction_Value

Le consommateur multipoint utilise Consumer_Time_Correction_Value pour corriger Producer_Time_Stamp, la datation résultante étant ainsi définie par rapport à l'horloge du consommateur.

7.6.4.3 Mcast_Octet.Multi_Cast_Active_Idle

La valeur 0 du bit Multi_Cast_Active_Idle doit indiquer l'état Repos, et la valeur 1 du bit Multi_Cast_Active_Idle doit indiquer l'état Actif.

Une fois le bit Multi_Cast_Active_Idle réglé sur "Actif" après la première production de données, ce bit ne doit plus être réglé sur repos jusqu'à la réinitialisation de la connexion de sécurité.

Si le consommateur constate la transition du bit Multipoint_Active_Idle de l'état Actif à l'état Repos, il doit demeurer dans l'état Repos jusqu'à la réinitialisation de la connexion de la connexion de sécurité.

Pour les connexions multipoint, le bit Run_Idle de Mode_Octet doit être réglé sur l'état Run_Idle indiqué par l'application de production.

Le consommateur est tenu d'observer les deux bits Run_Idle et Multi_Cast_Active_Idle.

7.6.4.4 Bits réservés de correction de temps Mcast_octet

Le producteur doit régler les bits réservés de correction de temps sur 0.

Le consommateur doit ignorer les bits réservés sauf pour le contrôle du CRC et de MC_Octet_2.

7.6.5 Production des données de sécurité

7.6.5.1 Etat de la connexion de production

L'état de la connexion du producteur doit être déterminé par la combinaison des drapeaux S_Connection_Fault et Consumer_Active_Idle (voir Tableau 85 pour une définition).

Tableau 85 – Détermination de l'état de la connexion du producteur

S_Connection_Fault	Consumer_Active_Idle	État de la connexion du producteur	Commentaire
Panne	Négliger	État = Défectueux (panne)	Réamorçage requis
OK	Repos	État = Repos	La connexion de sécurité n'a pas été activée
OK	Actif	État = Actif	La connexion de sécurité est active

7.6.5.2 Consumer_Open

Pour les producteurs de données de sécurité, la ressource Consumer_Open utilisée entre l'unité d'établissement de connexion de l'objet de validation et l'Objet de validation d'exécution doit permettre d'indiquer si un consommateur particulier des données produites a ou non une connexion ouverte active.

Si le producteur est multipoint, une ressource Consumer_Open doit exister pour chaque consommateur multipoint, de 1 à Max_Consumer_Number (index de 0 à Max_Consumer_Number – 1).

Les valeurs Consumer_Open sont «Ouvert» et «Fermé».

L'indication Consumer_Open doit agir comme une activation de l'Objet de validation d'exécution, sur une base connexion par connexion. Après l'établissement satisfaisant de la connexion, les drapeaux Consumer_Open doivent passer de l'état Fermé à l'état Ouvert.

L'unité d'établissement de connexion de l'objet de validation doit fournir une matrice de ressources d'état de la connexion du consommateur pour les producteurs multipoint qui doivent être attribués lorsqu'un consommateur établit une connexion.

Le nombre de consommateurs attribué à un Objet de validation d'exécution doit être le nombre transmis au consommateur dans la réponse d'établissement de connexion.

Le nombre de connexions du consommateur peut être traité comme un groupe d'attribution pouvant être réutilisé et réattribué si nécessaire.

L'unité d'établissement de connexion de l'objet de validation doit s'assurer que la ressource Consumer_Open pour tout nombre de consommateurs attribué est réglée sur l'état Fermé pendant une durée minimale de $(\text{Timeout\_Multiplier} + 2) \times \text{Intervalle Ping}$ préalablement à toute réattribution de ce nombre de consommateurs à une autre connexion. Cette opération est destinée à s'assurer de la temporisation des connexions établies précédemment, et ce, préalablement à toute attribution de ce nombre à une nouvelle connexion.

En d'autres termes, si l'unité d'établissement de connexion de l'objet de validation constate le réglage de S_Connection_Fault, elle doit régler la ressource Consumer_Active sur Fermé pendant une durée minimale de $(\text{Timeout\_Multiplier} + 2) \times \text{Intervalle Ping}$, ou éventuellement jusqu'à ce que le même consommateur tente de rétablir la même connexion.

7.6.5.3 Application_Run_Idle

Si l'application de production contrôle de manière active les données, l'indication Run_Idle doit mentionner l'état Exécution. Si l'application de production ne contrôle pas de manière active les données, et s'il convient que ces données adoptent un état de sécurité, l'indication Run_Idle doit mentionner l'état Repos.

7.6.5.4 Consumer_Active_Idle

Consumer_Active_Idle est un drapeau d'exécution qui indique si des informations valides concernant le message de coordination temporelle ont été reçues pour ce consommateur.

Si un producteur de données de sécurité génère des données et si des informations valides concernant la coordination temporelle ont été reçues sans aucune erreur, l'état du drapeau Consumer_Active_Idle doit être égal à Actif, ou à Repos dans le cas contraire.

Pour les producteurs multipoint, un drapeau Consumer_Active_Idle doit exister pour chaque consommateur multipoint, numéroté de 1 à Max_Consumer_Number.

Les valeurs Consumer_Active_Idle sont «Repos» et «Actif».

A l'établissement d'une connexion, tous les drapeaux Consumer_Active_Idle doivent être initialisés sur l'état Repos.

7.6.5.5 S_Connection_Fault

Drapeau d'exécution qui indique si la connexion de sécurité avec le client est OK ou Défectueuse.

L'indication S_Connection_Fault d'une panne doit signaler que le producteur a décelé un problème au niveau de la connexion de sécurité avec ce consommateur ou que le consommateur a signalé en retour une erreur au producteur.

Si le producteur de données de sécurité a reçu une information invalide concernant la coordination temporelle, ou une temporisation de ladite information, l'état du drapeau S_Connection_Fault doit être égal à « Panne », ledit état étant OK dans les autres cas.

Pour les producteurs multipoint, il doit y avoir un drapeau S_Connection_Fault pour chaque consommateur multipoint, numéroté de 1 à Max_Consumer_Number.

Les valeurs de S_Connection_Fault sont OK et Défectueux.

A l'établissement d'une connexion, toutes les variables S_Connection_Fault doivent être initialisées sur OK.

7.6.5.6 Variables statiques d'entrée du producteur

Il s'agit des paramètres de temps de configuration qui définissent l'opération du producteur de protocoles de sécurité.

7.6.5.7 EPI

Le terme EPI désigne l'intervalle de paquets attendu. La fréquence de production des données de sécurité ne doit pas dépasser l'EPI. L'EPI est exprimé comme une valeur entière en microsecondes. La plage légale des valeurs est comprise entre 100 μ s et 1 000 000 μ s.

Pour toutes les connexions de sécurité, le producteur et tous les consommateurs doivent avoir le même EPI.

7.6.5.8 Timeout_Multiplier [par consommateur]

Le Timeout_Multiplier indique le nombre de nouvelles tentatives de production de données à utiliser dans les équations de détection d'une défaillance de connexion.

Le Timeout_Multiplier doit être utilisé pour déterminer le nombre d'intervalles Ping (Timeout_Multiplier + 1) qu'un producteur est tenu d'attendre avant de rejeter un consommateur qui a omis d'envoyer une réponse de coordination temporelle suite à une demande Ping.

La plage légale des valeurs Timeout_Multiplier doit correspondre aux valeurs 1, 2, 3 ou 4.

Pour tout producteur, chaque consommateur peut avoir un Timeout_Multiplier différent.

7.6.5.9 Connection_Type

Connection_Type spécifie si la connexion est point à point ou multipoint.

Pour tout producteur, chaque connexion de producteur et la (les) connexion(s) de consommateur correspondante(s) doivent avoir le même Type de connexion.