



BSI Standards Publication

Industrial communication networks — Fieldbus specifications

Part 6-27: Application layer protocol specification — Type 27 elements

National foreword

This British Standard is the UK implementation of EN IEC 61158-6-27:2023. It is identical to IEC 61158-6-27:2023.

The UK participation in its preparation was entrusted to Technical Committee GEL/65, Measurement and control.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 11914 5

ICS 25.040; 25.040.40; 33.040.01; 33.040.40; 35.100.05

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2023.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

Industrial communication networks - Fieldbus Specifications Part
6-27: Application layer protocol specification - Type 27 elements
(IEC 61158-6-27:2023)

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 6-27 : Spécification du protocole de
la couche application - Éléments de type 27
(IEC 61158-6-27:2023)

Industrielle Kommunikationsnetze - Feldbusse - Teil 6-27:
Protokollspezifikation des Application Layer
(Anwendungsschicht) - Typ 27-Elemente
(IEC 61158-6-27:2023)

This European Standard was approved by CENELEC on 2023-04-20. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

The text of document 65C/1205/FDIS, future edition 1 of IEC 61158-6-27, prepared by SC 65C "Industrial networks" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN IEC 61158-6-27:2023.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2024-01-20
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2026-04-20

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

Any feedback and questions on this document should be directed to the users' national committee. A complete listing of these bodies can be found on the CENELEC website.

Endorsement notice

The text of the International Standard IEC 61158-6-27:2023 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standard indicated:

IEC 61158-1:2023 NOTE Approved as EN IEC 61158-1:2023 (not modified)

IEC 61158-2 NOTE Approved as EN 61158-2

IEC 61158-6 (series) NOTE Approved as EN 61158-6 (series)

IEC 61784-1 (series) NOTE Approved as EN IEC 61784-1 (series)¹

IEC 61784-2 (series) NOTE Approved as EN IEC 61784-2 (series)²

¹ To be published. Stage at the time of publication: FprEN IEC 61784-1-X:2023.

² To be published. Stage at the time of publication: FprEN IEC 61784-2-X:2023.

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

NOTE 1 Where an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cencenelec.eu.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61158-5-27	2023	Industrial communication networks - Fieldbus specifications - Part 5-27: Application layer service definition - Type 27 elements	EN IEC 61158-5-27	2023
ISO/IEC 646	-	Information technology; ISO 7-bit coded character set for information interchange	-	-
ISO/IEC 7498-1	-	Information technology - Open Systems Interconnection - Basic reference model: The basic model	-	-
ISO/IEC 9545	-	Information technology - Open Systems Interconnection - Application layer structure	-	-
ISO/IEC 9899	-	Information technology - Programming languages - C	-	-
ISO/IEC 10731	-	Information technology - Open Systems Interconnection - Basic Reference Model - Conventions for the definition of OSI services	-	-
ISO/IEC 19501	2005	Information technology - Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2	-	-
ISO/IEC/IEEE 60559	2020	Information technology - Microprocessor Systems - Floating-Point arithmetic	-	-
ISO/IEC/IEEE 8802-3	-	Telecommunications and exchange between information technology systems – Requirements for local and metropolitan area networks – Part 3: Standard for Ethernet	-	-
IEEE Std 802.1D	-	IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges	-	-

EN IEC 61158-6-27:2023 (E)

IEEE Std 802.1Q	-	IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks	-
IEEE Std 802.1AB	-	IEEE Standard for Local and metropolitan area networks: Station and Media Access Control Connectivity Discovery	-
IEEE Std 802.1AS	-	IEEE standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks	-
IETF RFC 768	1980	User Datagram Protocol	-
IETF RFC 791	1981	Internet Protocol	-
IETF RFC 826	1982	Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware	-
IETF RFC 1112	1989	Host Extensions for IP multicasting	-
IETF RFC 2460	1998	Internet Protocol - Version 6 (IPv6) - Specification	-

CONTENTS

FOREWORD.....	7
INTRODUCTION.....	9
1 Scope.....	10
1.1 General.....	10
1.2 Specifications	10
1.3 Conformance	11
2 Normative references	11
3 Terms, definitions, abbreviated terms, symbols and conventions	12
3.1 Referenced terms and definitions.....	12
3.1.1 Terms and definitions from ISO/IEC 7498-1	12
3.1.2 Terms and definitions from ISO/IEC 9545	12
3.1.3 Terms and definitions from ISO/IEC 8824-1	13
3.1.4 Terms and definitions from ISO/IEC 10731	13
3.1.5 Terms and definitions from ISO/IEC 19501	13
3.2 Additional terms and definitions	13
3.3 Abbreviations and symbols	22
3.4 Conventions.....	24
3.4.1 General conventions.....	24
3.4.2 PDU data type conventions.....	24
3.4.3 State machine conventions	24
4 Abstract syntax.....	26
4.1 General.....	26
4.2 Basic Data types.....	27
4.3 FAL PDU types	28
4.3.1 General	28
4.3.2 Top of APDU types: _APDU.....	31
4.3.3 PDUs for field system management service	31
4.3.4 Detailed definitions of _FID-PDUs	34
4.3.5 PDUs for field device control service	48
4.3.6 PDUs for message service.....	57
4.4 Detailed definitions of _FDCService-PDUs.....	58
4.4.1 Enhanced PDU type	58
4.5 Device profile.....	72
5 Transfer syntax.....	72
5.1 Concepts	72
5.2 Encode rules.....	73
5.2.1 INTEGER and its subtypes	73
5.2.2 REAL type and its subtypes	74
5.2.3 BIT STRING type.....	76
5.2.4 OCTET STRING type and IA5String type	77
5.2.5 NULL type	77
5.2.6 Structure type and Array type	77
6 Structure of FAL protocol state machine	78
7 AP-context state machine (APC SM)	80
7.1 Overview.....	80
7.2 State descriptions	80

7.3	Triggering events	81
7.4	Action descriptions at state transitions	82
8	FAL service protocol machines (FSPM)	84
8.1	Overview	84
8.2	RT Protocol Machine (RT PM)	84
8.2.1	Link layer discovery	84
8.2.2	MAC bridges	84
8.2.3	Virtual bridges	105
8.2.4	IP suite	105
8.2.5	DLL mapping protocol machine (DMPM)	105
8.3	Field System Management Protocol Machine (FSM PM)	110
8.3.1	Overview	110
8.3.2	Discovery and basic configuration	112
8.3.3	Starting up of system	115
8.3.4	Sync methods	144
8.3.5	Plug-and-play entry	153
8.4	Field Device Control Protocol Machine (FDC PM)	154
8.4.1	Protocol overview	154
8.4.2	Cyclic communication mode	156
8.4.3	Event driven communication mode	160
8.4.4	Master Protocol Machine (FDCPM-M)	161
8.4.5	Slave Protocol Machine (FDCPM-S)	169
8.4.6	Error procedure summary	179
8.5	Message Protocol Machine (MSG PM)	181
8.5.1	Protocol overview	181
8.5.2	Requester Protocol Machine (MSGPM-RQ)	183
8.5.3	Responder Protocol Machine (MSGPM-RS)	186
9	Application relationship protocol machine (ARPM)	188
9.1	General	188
9.2	ARPM for FDC ASE	188
9.2.1	Overview	188
9.2.2	ARPM for FDC Master (ARPM-FDCM)	189
9.2.3	ARPM for FDC Slave (ARPM-FDCS)	196
9.3	ARPM for MSG ASE (ARPM-MSG)	203
9.3.1	State descriptions	203
9.3.2	Triggering events	203
9.3.3	Action descriptions at state transitions	204
10	DLL mapping protocol machines (DMPMs)	204
	Annex A (informative) Device profile and FDC command sets	205
	Annex B (normative) Virtual memory space and Device Information	207
B.1	Overview	207
B.2	Communication Data Object	207
B.3	Device Information	208
B.3.1	Device identifier area structure	208
B.3.2	Detail specifications of device IDs	209
	Annex C (informative) Basic message function	217
	Bibliography	218

Figure 1 – Encode of Integer subtypes	73
Figure 2 – Example of transfer of INTEGER value	74
Figure 3 – Encode of Unsigned subtypes	74
Figure 4 – Float32 type encode.....	75
Figure 5 – Float64 type encode.....	75
Figure 6 – Bit field definition example with named bits	76
Figure 7 – Bit field definition example with field size	77
Figure 8 – SEQUENCE type encode	78
Figure 9 – Structure of FAL protocol state machines	80
Figure 10 – State chart diagram of APC SM.....	81
Figure 11 – Master CTC state transition chart.....	86
Figure 12 – Slave CTC state transition chart.....	96
Figure 13 – Structuring of the protocol machines within the DMPM (bridge)	106
Figure 14 – SRC state transition chart	106
Figure 15 – Example of network configuration.....	111
Figure 16 – Network initialization procedures.....	112
Figure 17 – Neighboring node notification sequence	113
Figure 18 – Connected node detection sequence.....	114
Figure 19 – Example of neighboring node information.....	115
Figure 20 – FSMUL state transition chart for the master with SM function	116
Figure 21 – FSMUL state transition chart for the master without SM function	129
Figure 22 – FSMUL state transition chart for a slave	138
Figure 23 – Sequence of SM delay time notification by CDO writing.....	145
Figure 24 – Delay measurement sequence (SM: BM).....	147
Figure 25 – Master delay measurement sequence for the BM (SM: master with band master function (BM))	148
Figure 26 – Delay measurement sequence for the master other than the BM (SM: BM).....	149
Figure 27 – Delay measurement sequence (SM: S#1).....	151
Figure 28 – Master delay measurement sequence for the BM (SM: S#1).....	152
Figure 29 – Plug-and-play entry sequence	154
Figure 30 – Example communication cycle of FDC master AP.....	156
Figure 31 – Example communication cycle of FDC slave AP	157
Figure 32 – Synchronous command communication in sync state	158
Figure 33 – Asynchronous command communication in sync state.....	159
Figure 34 – Asynchronous command communication in async state.....	160
Figure 35 – Event-driven communication	161
Figure 36 – State chart diagram of FDCPM-M.....	162
Figure 37 – State chart diagram of FDCPM-S	170
Figure 38 – PDU transmission flow for user message	182
Figure 39 – PDU transmission flow for one-way message	183
Figure 40 – State chart diagram of MSGPM-RQ	184
Figure 41 – State chart diagram of MSGPM-RS	186
Figure 42 – Example of single transfer process.....	188

Figure 43 – Example of dual transfer process	189
Figure 44 – State chart diagram of ARPM-FDCM	190
Figure 45 – Statechart diagram of ARPM-FDCS.....	196
Figure 46 – Statechart diagram of ARPM-MSG	203
Figure B.1 – Memory map of virtual memory space	207
Figure B.2 – Memory map of device ID area	208
Table 1 – State transition descriptions	25
Table 2 – Description of state machine elements	25
Table 3 – Conventions used in state machines	26
Table 4 – Multicast address	29
Table 5 – Length_or_type	29
Table 6 – IPv4 Header	30
Table 7 – IPv6 Header	30
Table 8 – UDP Header	30
Table 9 – Node address	33
Table 10 – Mapping for Protocol State Machines	79
Table 11 – State descriptions of APC SM.....	81
Table 12 – Trigger event descriptions of APC SM	82
Table 13 – Transitions of APC SM	83
Table 14 – Master CTC state transition table	87
Table 15 – Slave CTC state transition table	96
Table 16 – List of CTC macros.....	101
Table 17 – The primitives and parameters for FDC interface issued by FDC	103
Table 18 – The primitives and parameters for FSM interface issued by CTC	104
Table 19 – The list of primitives and parameters (FSM source)	104
Table 20 – The list of primitives and parameters (FSMUL source).....	105
Table 21 – SRC state transition table.....	107
Table 22 – List of SRC macros	108
Table 23 – List of SRC functions	109
Table 24 – Primitives and parameters for SRC-CTC interface	110
Table 25 – Send frame primitive and parameters	110
Table 26 – Receive frame primitives and parameters	110
Table 27 – Primitives and parameters of repeat select service	110
Table 28 – FSMUL state transition table for the master with SM function	117
Table 29 – FSMUL state transition table for the master without SM function	129
Table 30 – FSMUL state transition table for a slave	138
Table 31 – List of FSMUL macros	142
Table 32 – FDC protocol mode	155
Table 33 – State descriptions of FDCPM-M.....	162
Table 34 – Trigger event descriptions of FDCPM-M	163
Table 35 – Transitions of main SM of FDCPM-M.....	164
Table 36 – Transitions of submachine of FDCPM-M.....	166

Table 37 – State descriptions of FDCPM-S	170
Table 38 – Trigger event descriptions of FDCPM-S	171
Table 39 – Transitions of main SM of FDCPM-S	172
Table 40 – Transitions of submachine of FDCPM-S	174
Table 41 – State descriptions of MSGPM-RQ	184
Table 42 – Trigger event descriptions of MSGPM-RQ	185
Table 43 – Transitions of MSGPM-RQ	185
Table 44 – State descriptions of MSGPM-RS	186
Table 45 – Trigger event descriptions of MSGPM-RS	187
Table 46 – Transitions of MSGPM-RS	187
Table 47 – State descriptions of ARPM-FDCM	191
Table 48 – Trigger event descriptions of ARPM-FDCM	193
Table 49 – Transitions of main SM of ARPM-FDCM	193
Table 50 – Transitions of submachine of ARPM-FDCM	195
Table 51 – State descriptions of ARPM-FDCS	197
Table 52 – Trigger event descriptions of ARPM-FDCS	199
Table 53 – Transitions of main SM of ARPM-FDCS	200
Table 54 – Transitions of submachine of ARPM-FDCS	201
Table 55 – State descriptions of ARPM-MSG	203
Table 56 – Trigger event descriptions of ARPM-MSG	203
Table 57 – Transitions of ARPM-MSG	204
Table A.1 – Example of registered device profiles	205
Table A.2 – Example command list of the profile '00'H	206
Table B.1 – Memory map of CDO area	207
Table B.2 – Specifications of device IDs	209
Table C.1 – Example of message command set	217

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 6-27: Application layer protocol specification –
Type 27 elements****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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 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.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

Attention is drawn to the fact that the use of the associated protocol type is restricted by its intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

NOTE Combinations of protocol types are specified in the IEC 61784 1 series and the IEC 61784 2 series.

IEC 61158-6-27 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

The text of this International Standard is based on the following documents:

Draft	Report on voting
65C/1205/FDIS	65C/1234/RVD

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

A list of all parts of the IEC 61158 series, under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

INTRODUCTION

This document is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the "three-layer" fieldbus reference model described in IEC 61158-1.

The application protocol provides the application service by making use of the services available from the data-link or other immediately lower layer. The primary aim of this document is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer application entities (AEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- as a guide for implementers and designers;
- for use in the testing and procurement of equipment;
- as part of an agreement for the admittance of systems into the open systems environment;
- as a refinement to the understanding of time-critical communications within OSI.

This document is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this document together with other documents positioned within the OSI or fieldbus reference models, otherwise incompatible systems can work together in any combination.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent. IEC takes no position concerning the evidence, validity, and scope of this patent right.

The holder of this patent right has assured IEC that s/he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from the patent database available at patents.iec.ch.

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-27: Application layer protocol specification – Type 27 elements

1 Scope

1.1 General

The Fieldbus Application Layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a "window between corresponding application programs."

This part of IEC 61158 provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 27 fieldbus. The term "time-critical" is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This document defines in an abstract way the externally visible behavior provided by the Type 27 fieldbus application layer in terms of

- the abstract syntax defining the application layer protocol data units conveyed between communicating application entities,
- the transfer syntax defining the application layer protocol data units conveyed between communicating application entities,
- the application context state machines defining the application service behavior visibly between communicating application entities, and
- the application relationship state machines defining the communication behavior visibly between communicating application entities.

The purpose of this document is to define the protocol provided to

- define the representation-on-wire of the service primitives defined in IEC 61158-5-27, and
- define the externally visible behavior associated with their transfer.

This document specifies the protocol of the Type 27 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

1.2 Specifications

The principal objective of this document is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-27.

A secondary objective is to provide migration paths from previously existing industrial communications protocols. It is this latter objective which gives rise to the diversity of protocols standardized in IEC 61158-6 series.

1.3 Conformance

This document does not specify individual implementations or products, nor does it constrain the implementations of application layer entities within industrial automation systems.

Conformance is achieved through implementation of this application layer protocol specification.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

NOTE All parts of the IEC 61158 series, as well as the IEC 61784-1 series and the IEC 61784-2 series are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61158-5-27:2023, *Industrial communication networks – Fieldbus specifications – Part 5-27: Application layer service definition – Type 27 elements*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model – Part 1: The Basic Model*

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

ISO/IEC 9899, *Information technology – Programming languages – C*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*

ISO/IEC/IEEE 60559:2020, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

ISO/IEC/IEEE 8802-3, *Telecommunications and exchange between information technology systems – Requirements for local and metropolitan area networks – Part 3: Standard for Ethernet*

IEEE Std 802.1D, *IEEE Standards for local and metropolitan area networks – Media Access Control (MAC) Bridges*

IEEE Std 802.1Q, *IEEE Standards for local and metropolitan area networks – Bridges and Bridged Networks*

IEEE Std 802.1AB, *IEEE Standards for Local and Metropolitan Area Networks: Station and Media Access Control Connectivity Discovery*

IEEE Std 802.1AS, *IEEE Standards for Local and Metropolitan Area Networks: Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*

IETF RFC 768, J. Postel, "*User Datagram Protocol*", August 1980, available at <<https://www.rfc-editor.org/info/rfc768>> [viewed 2022-09-16]

IETF RFC 791, J. Postel, "*Internet Protocol*", September 1981, available at <<https://www.rfc-editor.org/info/rfc791>> [viewed 2022-09-16]

IETF RFC 826, D. Plummer, "*An Ethernet Address Resolution Protocol: or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*", November 1982, available at <<https://www.rfc-editor.org/info/rfc826>> [viewed 2022-09-16]

IETF RFC 1112, S.E. Deering, "*Host Extensions for IP Multicasting*", August 1989, available at <<https://www.rfc-editor.org/info/rfc1112>> [viewed 2022-09-16]

IETF RFC 2460, S. Deering and R. Hinden, "*Internet Protocol, Version 6 (IPv6) Specification*", December 1998, available at <<https://www.rfc-editor.org/info/rfc2460>> [viewed 2022-09-16]

3 Terms, definitions, abbreviated terms, symbols and conventions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- IEC Electropedia: available at <https://www.electropedia.org/>
- ISO Online browsing platform: available at <https://www.iso.org/obp>

3.1 Referenced terms and definitions

3.1.1 Terms and definitions from ISO/IEC 7498-1

For the purposes of this document, the following terms as defined in ISO/IEC 7498-1 apply:

- abstract syntax;
- application-entity;
- application process;
- application protocol data unit;
- application-process-invocation;
- (N)-facility;
- (N)-function;
- peer-(N)-entities;
- presentation context;
- real system;
- transfer syntax.

3.1.2 Terms and definitions from ISO/IEC 9545

For the purposes of this document, the following terms as defined in ISO/IEC 9545 apply:

- application-association;
- application-context;
- application-entity-invocation;
- application-entity-type;

- application-service-element.

3.1.3 Terms and definitions from ISO/IEC 8824-1

For the purposes of this document, the following terms as defined in ISO/IEC 8824-1 apply:

- simple type;
- component;
- component type;
- integer type;
- bitstring type;
- octetstring type;
- null type;
- sequence type;
- sequence of type;
- choice type;
- IA5String type;
- encoding.

3.1.4 Terms and definitions from ISO/IEC 10731

For the purposes of this document, the following terms as defined in ISO/IEC 10731 apply:

- OSI-service-primitive; primitive;
- OSI-service-provider; provider;
- OSI-service-user; user.

3.1.5 Terms and definitions from ISO/IEC 19501

For the purposes of this document, the following terms as defined in ISO/IEC 19501 apply:

- event;
- state;
- state machine;
- substate;
- submachine;
- transition.

3.2 Additional terms and definitions

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

3.2.1

alarm

field device status to tell that the device has detected a fatal problem to be solved and cannot continue normal working, through the field device control (FDC) service of the Type 27 fieldbus

Note 1 to entry: Any alarm statuses are latched and need some operations to be cleared.

Note 2 to entry: Alarms are classified into three groups; communication alarms, illegal-command-related ones, and application specific ones. But concrete definitions are dependent on implementation of each field devices.

3.2.2

application process object

network representation of a specific aspect of an application process (AP), which is modelled as a network accessible object contained within an AP or within another APO

Note 1 to entry: Refer IEC 61158-1, 9.3.4.

3.2.3

application process context

AP context

shared knowledge or a common set of rules, governing communication of FAL application entities (AEs) and describing the permissible collective communications behavior between the AEs that are party to a specific set of application relationships (ARs)

Note 1 to entry: Data within AP context can be specified by the user in advance, by the option selected while the user uses a field bus management (FSM) service to read out the facility of peer AP, by the automatic negotiation function that the FSM system handles, and so on. The method that is to be adopted depends on the specification of each implementation.

3.2.4

application process type

AP type

description of a classification of application processes (APs) in terms of a set of capabilities for FAL of the Type 27 fieldbus

Note 1 to entry: AP types are classified into two, C1 master AP and slave AP, by their application roles in the fieldbus network.

3.2.5

async command

type of a command application protocol data unit (APDU) of the FDC service of the Type 27 FAL, which can be issued any time after the previous transaction without consideration of synchronization with the communication cycle

Note 1 to entry: Definitions, which command should be async one or not, are dependent on an application. They can be provided as a registered set of commands and responses or device profiles, see Annex A.

3.2.6

asynchronous communication

state or a way of communication for the FDC service of the Type 27 FAL, in which a command can be issued any time after the previous transaction without consideration of synchronization with the communication cycle

Note 1 to entry: In this state, sync commands cannot be issued, but async commands can.

3.2.7

attribute

information or parameter contained in variable portions of an object

Note 1 to entry: Typically, they provide status information or govern the operation of an object. Attributes also affect the behavior of an object.

3.2.8

C1 master

AP type that has master facilities for the FDC service of the Type 27 FAL, or the device implementing that AP type

Note 1 to entry: Only one C1 master exists in a network of the Type 27 fieldbus.

3.2.9

command

PDU issued by a requester or a master to make a responder or a slave execute some functions

**3.2.10
communication**

process to exchange information in a formal manner between two or more devices, users, APs or entities

**3.2.11
transfer**

process to convey a PDU from a sender to a receiver

**3.2.12
transmission**

process to send out and propagate electrical signals or encoded data

**3.2.13
communication cycle**

period of repetitive activities synchronized with the transmission cycle while the connection establishing for the FDC protocol of the Type 27 FAL

Note 1 to entry: Communication cycle can synchronize with the transmission cycle multiplied by a specified scaling factor.

**3.2.14
connection**

context or logical binding under specific conditions for the FDC protocol between a master object and a slave object for the Type 27 FAL

**3.2.15
cyclic**

repetitive in a regular manner

**3.2.16
cyclic communication**

transmission mode in which request PDUs and response PDUs are exchanged repetitively in the scheduled time slots synchronized with a transmission cycle of the Type 27

Note 1 to entry: In the AL, the communication cycle arises from the transmission cycle in this mode.

**3.2.17
cycle scale counter**

counter to generate a communication cycle by means of scaling a primary cycle or a transmission cycle

**3.2.18
device ID**

part of "Device Information" to identify the device for a specific product type or model of the Type 27 fieldbus

**3.2.19
device information**

formatted and device-embedded information to characterize a device, which mainly consists of data for device model identification and device-profile specific parameters for the Type 27 fieldbus

**3.2.20
device profile**

collection of device-model-common information and functionality providing consistency between different device models among the same kind of device

3.2.21

dual transfer

transfer mode for the FDC protocol of the Type 27 FAL, in which a sender sends a same PDU twice a transaction and a receiver uses them to detect and recover a communication error such as data-corruption or data-loss in cyclic communication mode

3.2.22

event driven communication

transmission mode for the lower layer protocol of the Type 27 fieldbus in which a transaction of command-response-exchanging arises as user's demands

Note 1 to entry: Both the transmission cycle and the communication cycle don't arise in this mode.

3.2.23

error

abnormal condition or malfunction for communication or any other activities

3.2.24

field device control

FDC service

time-critical communication service that handles a fixed length command data to control a field device and the corresponding feedback response data in a severe restriction on delay or jitter for the communication timing for the Type 27 FAL

3.2.25

field device protocol

FDC protocol

time-critical communication protocol that handles a fixed length command data to control a field device and the corresponding feedback response data in a severe restriction on delay or jitter for the communication timing for the Type 27 FAL

3.2.26

master

class or its instance object of FDC application service element (ASE) that plays a role of a command requester for the Type 27 FAL

Note 1 to entry: One or more units exist within the network as the node to send output data (command data) to slaves and receive input data (response data) from slaves in the data link layer. Also, it executes message data sending to/receiving from slaves. It can execute both or either one of input/output data sending/receiving and message data sending/receiving.

Note 2 to entry: Input/output data sending/receiving and message data sending/receiving shall be executed at a constant cycle and in a constant band.

Note 3 to entry: Or, it is an application type that executes field device control in the application layer.

3.2.27

message service

MSG service

communication service that handles the variable length data and does not require a severe restriction on response time

3.2.28

network clock

synchronized and periodically running counter that each node in a same network has, which becomes an oscillation source of the transmission cycle

3.2.29

primary cycle

period of repetitive activities synchronized with the transmission cycle before the connection establishing for the FDC protocol in Type 27 FAL

3.2.30**protocol machine**

state machine that realizes a protocol as the main function of the entity in each layer

3.2.31**requester**

class or its instance object of MSG ASE that plays a role of a command requester or sender for the Type 27 FAL

3.2.32**responder**

class or its instance object of MSG ASE that plays a role of a command responder or receiver for the Type 27 FAL

3.2.33**response**

PDU issued by a responder or a slave to inform a result or some status for the received command to a requester or a master

3.2.34**service**

operation or process that an object performs upon request from another object

3.2.35**single transfer**

normal transfer mode for the FDC protocol of the Type 27 FAL in which a sender sends a PDU once a transaction

3.2.36**slave AP**

AP type that has slave facilities for the FDC service of the Type 27 FAL, or the device implementing that AP type

3.2.37**slave**

class or its instance object of FDC ASE that plays a role of a responder for the Type 27 FAL

Note 1 to entry: A node to receive output data (command data) from a master and send input data (response data) to the master that sent the output data in the data link layer. Also, it executes message data sending to/receiving from the master. It can execute input/output data and message data sending to/receiving from multiple masters. Also, it can receive input data of other slaves.

3.2.38**sync command**

type of command APDU of the FDC service of the Type 27 FAL, which is issued at the synchronized timing with every communication cycle

Note 1 to entry: The definitions, which command is sync one or not, are dependent on an application. They can be provided as a registered set of commands and responses or device profiles, see Annex A.

3.2.39**synchronous communication**

state or a way of communication for the FDC service of the Type 27 FAL, in which a command is issued at the synchronized timing with every communication cycle

Note 1 to entry: In this state, both sync commands and async ones can be issued.

Note 2 to entry: In this state, an out-of-synchronization error of APs shall be detected by measures of the watchdog counter.

3.2.40**transmission cycle**

period of repetitive activities for the lower layers of the Type 27 fieldbus, which of all the slave devices are synchronized with that of a C1 master device by the lower layer protocol

Note 1 to entry: Cycle in which the sync master sends a sync packet. The master having the band master function informs each node in the network. The sync band, the control domain transmission band, the IP transmission band and the guard band need to be within the transmission cycle. The transmission cycle starting point is managed by the sync master and is informed by the sync packet to all nodes. Between the master and slave within the control domain, or in the data link layer between the master having the band master function and the master without it, cyclic transmissions can be repeated at a cycle of n -times (n is an integer equal to or greater than zero) of the transmission cycle and it can vary from slave to slave.

3.2.41**transmission mode**

state or a way of transmission for the lower layer protocol of the Type 27 fieldbus; cyclic mode, event driven mode

3.2.42**virtual memory space**

large data block of APOs for the Type 27 FAL that can be read and written with pseudo-memory-addresses to provide consistency between different device models

Note 1 to entry: The virtual memory space includes the device information and other vender specific area, see Annex B.

3.2.43**warning**

field device status to tell that the device has detected a slight or passing problem but still working normally through the field device control (FDC) service of the Type 27 fieldbus

Note 1 to entry: Any warning statuses are latched and need to be operated to clear them.

Note 2 to entry: Warnings are classified into three groups, communication warnings, illegal-command-related ones, and application specific ones. But concrete definitions are dependent on an implementation of each field devices.

3.2.44**node**

device that comply with the protocols described in this specification

3.2.45**node type**

master or slave

3.2.46**multi control domain system**

configuration where multiple control domains exist in the network

3.2.47**connected node**

collective designation of master and slaves that are connected to the same network

3.2.48**primary node****initiator**

node to start cyclic transmissions and message transmissions

Note 1 to entry: Only the master serves as the primary node.

3.2.49**secondary node
responder**

node to execute cyclic transmissions and message transmissions according to the command from the primary node

3.2.50**output data
command data**

data to be sent from a master to a slave

3.2.51**input data
response data**

data to be sent from a slave to a master

3.2.52**input/output data**

input data and output data

3.2.53**control data**

collective designation of output data and input data

3.2.54**message data**

data transmitted separately from the control data, which are exchanged between a master and a slave and sent when requested by the application

3.2.55**cyclic transmission**

cyclic exchanging of control data between a master and a slave

3.2.56**message transmission**

exchanging of message data to transmit information separately from the control data, which is, unlike cyclic transmission, conducted only when required by the application

3.2.57**IP transmission**

communication using IP packets

3.2.58**control domain**

logical group which consists of a master and slaves to exchange control data and/or message data

Note 1 to entry: Multiple control domains can exist in the network, and in this case, they are indicated as control domain #*n* (*n* is an integer).

Note 2 to entry: A slave shall belong to one or more control domains.

3.2.59**sync band**

band to transmit sync packet

3.2.60**cyclic transmission band**

band to execute cyclic transmissions

3.2.61

output data transmission band

band to transmit output data in cyclic transmissions

3.2.62

input data transmission band

band to transmit input data in cyclic transmissions

3.2.63

message transmission band

band to execute message transmissions

3.2.64

control domain transmission band

total band of the cyclic transmission band and the message transmission band within a control domain

3.2.65

IP transmission band

band to execute IP packet transmissions

3.2.66

guard band

band to prohibit sending/receiving by the devices in the network

3.2.67

destination address

address to specify the control data destination, having 16-bit information

3.2.68

source address

address to specify the control data source, having 16-bit information

3.2.69

node address

address to be specified as destination address or source address

3.2.70

logical node number

number from 1 to 127 to be used as serial number in ascending order, which enables the master to specify the slave, because there are 127 connectable slaves to the node address having 16-bit space in the data link layer

3.2.71

transmission delay time

time required for the frame sent by the sync master or the master in the data link layer to reach each node

Note 1 to entry: A total time of the signal propagation time in the transmission cables and the repeat time at the relay nodes.

3.2.72

response monitoring time

time to monitor the reception of responses

Note 1 to entry: This is measured from the starting of sending of data sent from the primary node (including the transmission time of the data sent from the primary node).

3.2.73**consecutive transmissions**

continuous transmissions of output data

3.2.74**retransmission request**

executing of input data retransmission request

3.2.75**event driven communication**

communication mode where the master and the slave exchange data at an arbitrary timing, without executing clock synchronization in the devices by the sync packet sent from the sync master

3.2.76**transmission delay measurement**

individual measurements by the sync master of transmission delay time for the nodes other than its own node

3.2.77**upstream**

direction of data forwarding from the primary node to the secondary node

3.2.78**downstream**

direction of data forwarding from the secondary node to the primary node

3.2.79**upstream port**

port that becomes upstream among the communication ports to be implemented

Note 1 to entry: While multiple control domains exist in the network (two or more masters exist), in cases where the upstream port differs from control domain to control domain, it is referred to as the control domain # n (n is an integer) upstream port.

3.2.80**downstream port**

port that becomes downstream among the communication ports to be implemented

Note 1 to entry: While multiple control domains exist in the network (two or more masters exist), in cases where the downstream port differs from control domain to control domain, it is referred to as the control domain # n (n is an integer) downstream port.

3.2.81**transfer method to upstream**

upstream transfer method among the transfer methods in the node where two or more communication ports are implemented

3.2.82**transfer method to downstream**

downstream transfer method among the transfer methods in the node where two or more communication ports are implemented

3.2.83**store and forward**

transfer method to store the received packets in the node first and forward them to the ports other than the receiving ports according to the result of stored data analysis and the band conditions upon receiving in the band

3.2.84**cut-through**

transfer method to forward the received packets immediately to the ports other than the receiving port

3.2.85**repeat**

transfer method to transfer the received packets in the same way as cut-through in cases where sending is not in progress in its own node, or to retain the received packets during sending from its own node and forward to the ports other than the receiving port after finishing sending from its own node

3.2.86**IP packet**

all packets other than those with the MAC frame length/type field of RT-PDU ('F68E'H) in this document

3.2.87**sync packet**

packet to synchronize transmission cycles of all the devices in the network, which is sent from the sync master

3.3 Abbreviations and symbols

For the purposes of this document, the following abbreviations and symbols apply.

AE	Application Entity
AL	Application Layer
A-, AL-	Application Layer (as a prefix)
ALME	Application Layer Management Entity
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Invocation
APO	Application Process Object
APC	Application Process Context (as prefix of a protocol for Type 27 fieldbus)
APC SM	Application Process Context State Machine (for Type 27 fieldbus)
AR	Application Relationship
AR ASE	Application Relationship Application Service Element
AREP	Application Relationship End Point
ARPM	Application Relationship Protocol Machine (for Type 27 fieldbus)
ARPM-FDCM	ARPM for Field Device Control service Master (for Type 27 fieldbus)
ARPM-FDCS	ARPM for Field Device Control service Slave (for Type 27 fieldbus)
ARPM-MSG	ARPM for Message service (for Type 27 fieldbus)
ASCII	American Standard Code for Information Interchange
ASDU	Application Service Data Unit
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
BM	Band Master
CDO	Communication Data Object
CMD	Command PDU for FDC service (for Type 27 fieldbus)

Cnf	confirm primitive
CTC	Communication Transfer Control Underlying Layer
DA	Destination Node Address
DLPDU	Data Link-Protocol Data Unit
DLSDU	Data Link Service Data Unit
DMPM	Data Link Mapping Protocol Machine
DMA	Destination MAC Address
E2PROM	Electrically erasable programmable read only memory
FAL	Fieldbus Application Layer
FCS	Frame check sequence
FDC-	Field Device Control (as prefix of a service or a protocol for Type 27 fieldbus)
FDC ASE	Field Device Control Application Service Element (for Type 27 fieldbus)
FDCPM	Field Device Control Protocol Machine (for Type 27 fieldbus)
FDCPM-M	Field Device Control Protocol Machine for Master (for Type 27 fieldbus)
FDCPM-S	Field Device Control Protocol Machine for Slave (for Type 27 fieldbus)
FIFO	First In First Out
FSPM	FAL service protocol machine
FSM-	Fieldbus System Management (as prefix of a service for Type 27 fieldbus)
FSM ASE	Fieldbus System Management Application Service Element for Type 27 fieldbus
FSMUL	Field System Management Underlying Layer
HMI	Human-machine Interface
HS	Hand Shake
I/O	Input/output
ID	Identifier
Ind	indication primitive
LME	Layer Management Entity
Lsb	least significant bit
LT	Length or Type
MAC	Media Access Control
Msb	most significant bit
MSG	Message (as prefix of a service or a protocol for Type 27 fieldbus)
MSG ASE	Message Application Service Element for Type 27 fieldbus
MSGPM	Message Protocol Machine for Type 27 fieldbus
MSGPM-RQ	MSGPM for Requester for Type 27 fieldbus
MSGPM-RS	MSGPM for Responder for Type 27 fieldbus
OSI	Open Systems Interconnection
PM	Protocol machine
PDU	Protocol Data Unit
PhL	Ph-layer
QoS	Quality of Service
RAM	Random access memory
Req	request primitive

Rsp	response primitive
RSP	Response PDU for FDC service (for Type 27 fieldbus)
SA	Source Node Address
SAP	Service Access Point
SDN	Send Data with no Acknowledge
SDU	Service Data Unit
SFD	Start Frame Delimiter
SM	Sync Master
SMA	Source MAC Address
SMIB	System Management Information Base
SRC	Send Receive Control Underlying Layer
UML	Unified Modelling Language

3.4 Conventions

3.4.1 General conventions

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5-27. The protocol specification for each of the ASEs is defined in this document.

The class definitions define the attributes of the classes supported by each ASE. In this document, it is assumed that every attribute are accessible only from the owner's instance object itself directly or through services of the class.

This document uses the descriptive conventions given in ISO/IEC 10731.

3.4.2 PDU data type conventions

The data types of FAL PDUs are defined with the notations of ASN.1.

Character strings that begin with "_" (low line) are used as data type symbols of FAL PDUs or their fields. A same string as a PDU type without "_" means a PDU instance of the corresponding data type.

Example CMD-PDU means an instance of _CMD-PDU and a following statement is omitted.

```
CMD-PDU _CMD-PDU::= { value }
```

3.4.3 State machine conventions

The protocol sequences are described by means of State Machines.

In state chart diagrams, states are represented as boxes, and state transitions are shown as arrows. Their conventions are given in ISO/IEC 19501 as Universal Modeling Language (UML).

Names of states and transitions of the state diagram correspond to the names in the textual listing of the state transitions.

The textual listing of the state transitions is structured as Table 1.

The first row contains the name of the transition by an index number.

The second row defines the source state or the current state before the transition.

The third row contains an event and actions. The event is followed by optional arguments in parentheses, "(" and ")", and guard conditions in brackets, "[" and "]", as the first line. And the actions with starting character "/" follow the event line.

The last row contains the target state or the next state after the transition.

If the event occurs and the conditions are fulfilled the transition fires, i.e. the actions are executed and the next state is entered.

Table 1 – State transition descriptions

T#	Source State	Event (arguments) [conditions] / Action	Target state

A meaning of each element in Table 1 is shown in Table 2, based on UML "transition" definition (ISO/IEC 19501).

Table 2 – Description of state machine elements

Description element	Meaning
Source state Target state	Names of the originating state and the target state of transition.
T#	Name or number of the transition.
Event	Name or description of the trigger event that fire the transition.
(arguments)	A parameter value, an expression, or a sequence of those divided by "," for the event. The preceding "(" and the succeeding ")" are not part of the argument list.
[conditions]	Boolean expression, which is true for the transition to be fired. The preceding "[" and the succeeding "]" are not part of the condition.
/ Action	List of assignments and service or function invocations. The action should be atomic. The preceding "/" is not part of the action.
NOTE "(arguments)" and "[condition]" can be omitted if not necessary.	

The conventions used in the state machines are shown in Table 3.

Table 3 – Conventions used in state machines

Convention	Meaning
<code>/*-- description --*/</code>	Describes and explains conditions and/or procedure by using normal sentence between <code>/*--</code> and <code>--*/</code> instead of using the pseudo code notation. Example: <code>/*-- The PM examines an occurrence of any alarms for the corresponding remote device.</code> <code>-- If an alarm is detected, the PM notifies it to the FSM ASE.</code> <code>-- If not, the PM transfers the MSGService-PDU to the DLL.</code> <code>--*/</code>
<code>=</code>	Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event
<code>Axx</code>	Parameter name if 'a' is a letter Example: <code>Identifier = reason;</code> means value of a 'reason' parameter is assigned to a parameter called 'Identifier'
<code>"xxx"</code>	Fixed visible string Example: <code>Identifier = "abc";</code> means value "abc" is assigned to a parameter named 'Identifier'
<code>Nnn</code>	If all elements are digits, the item represents a numerical constant shown in decimal representation.
<code>0xnn</code>	If all elements nn are digits, the item represents a numerical constant shown in hexadecimal representation.
<code>==</code>	Logical condition to indicate an item on the left is equal to an item on the right
<code><</code>	Logical condition to indicate an item on the left is less than the item on the right
<code>></code>	Logical condition to indicate an item on the left is greater than the item on the right
<code>!=</code>	Logical condition to indicate an item on the left is not equal to an item on the right
<code>&&</code>	Logical "AND"
<code> </code>	Logical "OR"
<code>!</code>	Logical "NOT"
<code>+</code> <code>-</code> <code>*</code> <code>/</code>	Arithmetic operators
<code>;</code>	Separator of expressions

Further notations as defined in C language (ISO/IEC 9899) can be used to describe conditions and actions.

4 Abstract syntax

4.1 General

Packet formats and frame formats comply with the ISO/IEC/IEEE 8802-3.

4.2 Basic Data types

The following data types can be used in the Type 27 FAL.

- Basic types as simple types of ASN.1:
 - INTEGER;
 - REAL;
 - BIT STRING;
 - OCTET STRING;
 - NULL.
- Specific basic types as subtypes from ASN.1:
 - Integer8 ::= INTEGER (–128..127);
 - Integer16 ::= INTEGER (–32 768 .. 32 767);
 - Integer32 ::= INTEGER (–2 147 483 648 .. 2 147 483 647);
 - Integer64 ::= INTEGER ((–1)×'8000 0000 0000 0000'H..'7FFF FFFF FFFF FFFF'H);
 - Unsigned8 ::= INTEGER (0..'FF'H);
 - Unsigned16 ::= INTEGER (0..'FFFF'H);
 - Unsigned32 ::= INTEGER (0..'FFFF FFFF'H);
 - Unsigned64 ::= INTEGER (0..'FFFF FFFF FFFF FFFF'H);
 - Float32 ::= 0 | negative infinity | positive infinity
 | REAL (WITH COMPONENTS(mantissa ((–1)× c_{32}), base (2), exponent (q_{32})))
 | REAL (WITH COMPONENTS(mantissa (c_{32}), base (2), exponent (q_{32})))

where,

$$c_{32} ::= \text{REAL} \left(1.. \left(\sum_{i=0}^{23} \frac{1}{2^i} \right) \right) = \text{REAL} (1..(2-2^{-23})), \text{ and}$$

$$q_{32} ::= \text{INTEGER} (-127..127).$$

NOTE 1 This type has a length of four octets. The format for float32 is that defined by ISO/IEC/IEEE 60559 as single precision, see 5.2.2.

- Float64 ::= 0 | negative infinity | positive infinity
 | REAL (WITH COMPONENTS(mantissa ((–1)× c_{64}), base (2), exponent (q_{64})))
 | REAL (WITH COMPONENTS(mantissa (c_{64}), base (2), exponent (q_{64})))

where,

$$c_{64} ::= \text{REAL} \left(1.. \left(\sum_{i=0}^{52} \frac{1}{2^i} \right) \right) = \text{REAL} (1..(2-2^{-52})), \text{ and}$$

$$q_{64} ::= \text{INTEGER} (-1\ 023..1\ 023).$$

NOTE 2 This type has a length of eight octets. The format for float64 is that defined by ISO/IEC/IEEE 60559 as double precision. See 5.2.2.

- IA5String
- Structure types with component types of ASN.1:
 Structure type is defined by a combination of SEQUENCE type, CHOICE type of ASN.1, basic types and subtype elements defined in ASN.1. The FAL PDU described in 4.3 and 4.4 that follows this subclause correspond this Structure type.
 - SEQUENCE
 - CHOICE

- Array types with a component type of ASN.1:

Array types are a list of any data type defined with SEQUENCE OF type of ASN.1. Typical array types are shown with SEQUENCE OF type, below:

- SEQUENCE OF;
- BitArray ::= SEQUENCE OF (BIT STRING SIZE (8));
- Array8 ::= SEQUENCE OF Integer8;
- Array16 ::= SEQUENCE OF Integer16;
- Array32 ::= SEQUENCE OF Integer32;
- Array64 ::= SEQUENCE OF Integer64.

4.3 FAL PDU types

4.3.1 General

4.3.1.1 Packet format

The overview of the MAC frame used for this protocol is as described below. VLAN tags specified by IEEE Std 802.1Q can be used as an option to designate the priority level of the packets (see 4.3.1.2.4, 8.2). Normally, a format without IP headers is used (see 4.3.1.2.5). In cases where connected devices exchange data with a device connected to an external IP network, the UDP/IP protocol shall be used as option. Note that a part of UDP/IP header fields are not used for this protocol (see 4.3.1.2.60).

```

DLPDU ::= SEQUENCE {
    preamble                OCTET STRING SIZE (7),
    start_frame_delimiter    OCTET STRING SIZE (1),
    destination_MAC_address OCTET STRING SIZE (6),
    source_MAC_address       OCTET STRING SIZE (6),
    data                     _DLSDU,
    frame_check_sequence     OCTET STRING SIZE (4)
}

_DLSDU ::= SEQUENCE {
    ( VLAN_tag             OCTET STRING SIZE (4) ) | Empty,
    length_or_type         OCTET STRING SIZE (2),
    data                   _IP4-UDP-PDU | _IP6-UDP-PDU | _RT-PDU
    padding                OCTET STRING SIZE (0..46)
}

_IP4-UDP-PDU ::= SEQUENCE {
    IPv4_header            OCTET STRING SIZE (20),
    UDP_header             OCTET STRING SIZE (8),
    payload                OCTET STRING SIZE (1..1472)
}

_IP6-UDP-PDU ::= SEQUENCE {
    IPv6_header            OCTET STRING SIZE (40),
    UDP_header             OCTET STRING SIZE (8),
    payload                OCTET STRING SIZE (1..1452)
}

_RT-PDU ::= {
    payload                OCTET STRING SIZE (1..1500)
}

```

4.3.1.2 Common component

4.3.1.2.1 Preamble field and start_frame_delimiter field

This field shall comply with ISO/IEC/IEEE 8802-3.

4.3.1.2.2 Destination_MAC_address field

This field shall comply with ISO/IEC/IEEE 8802-3. Note that the usages of multicast address in Table 4 are as specified below.

Table 4 – Multicast address

Multicast address	Name	Usage
01:20:B5:00:04:00	Common multicast address	Simultaneous transmissions in the same network
01:20:B5:00:04:01	Multicast address to neighbouring node	This is used for sending to a node whose MAC address is not known. The received node shall not transfer to other ports.
01:20:B5:00:04:21 to 01:20:B5:00:04:2F	For control domain multicast address	For simultaneous transmissions in a control domain

4.3.1.2.3 Source_MAC_address field

This field shall comply with ISO/IEC/IEEE 8802-3.

4.3.1.2.4 VLAN_tag field

This field shall comply with ISO/IEC/IEEE 8802-3. Implementation is optional.

4.3.1.2.5 Length_or_type field

Without an IP header, this field shall be RT-PDU. With an IP header, this field shall comply with ISO/IEC/IEEE 8802-3. Values mainly used are as shown in Table 5.

Table 5 – Length_or_type

Value	Meaning
'F68E'H	RT-PDU
'0800'H	IP4-UDP-PDU, IP4-UDP-PDU
other	In accordance with ISO/IEC/IEEE 8802-3

4.3.1.2.6 IPv4_header field, IPv6_header field and UDP_header field

This field shall comply with IETF RFC 791, IETF RFC 2460 and IETF RFC 768. Note that a part of fields are not used (see Table 6, Table 7 and Table 8).

Table 6 – IPv4 Header

Alignment [octet]	Descriptions	Remarks
0	Version 4 (IPv4) and header length: 5 quadlets (20 octets)	'45'H, fixed
1	Service type	'00'H, fixed
2 to 3	Total length	
4 to 5	Identifier	Not used
6 to 7	Flag and fragment offset	0, fixed
8	Time To Live (TTL)	'01'H, fixed
9	High-order protocol type	'11'H(UDP), fixed
10 to 11	IP header checksum	Without checking
12 to 15	Source IP address	
16 to 19	Destination IP address	

Table 7 – IPv6 Header

Alignment [octet]	Descriptions	Remarks
0 to 3	Version 6 (IPv6)	6, fixed
	Traffic class	Not used (0, fixed)
	Flow label	Not used (0, fixed)
4 to 5	Payload length	
6	Subsequent header	'11'H(UDP), fixed
7	Maximum hop number	
8 to 23	Source IP address	
24 to 39	Destination IP address	

Table 8 – UDP Header

Alignment [octet]	Descriptions	Remarks
0 to 1	Source UDP port number	Not used
2 to 3	Destination UDP port number	'F68E'H, fixed
4 to 5	UDP data length	Not used
6 to 7	UDP checksum	Not used (0, fixed)

4.3.1.2.7 Payload field

_APDU is stored in this field (see 4.3.2).

4.3.1.2.8 Frame_check_SEQUENCE field

This field shall comply with ISO/IEC/IEEE 8802-3.

4.3.2 Top of APDU types: **_APDU**

_APDU consists of 14 types of FIDAP-PDU. And **_FDCServicePDU** shall consist of a pair of PDUs; **_CMD-PDU** and **_RSP-PDU**, used for FDC commands and its responses respectively. This is used in **_OUTPUT-FIDAP-PDU** (See 4.3.4.2), **_INPUT-FIDAP-PDU** (See 4.3.4.3), and **_EVD-FIDAP-PDU** (See 4.3.4.11). Furthermore, **_MSGServicePDU** shall consist of a pair of PDUs; **_MSGREQ-PDU** and **_MSGRSP-PDU**, used for MSG requests and its responses respectively. This is used in **_MSG-FIDAP-PDU** (See 4.3.4.9).

```

_APDU ::= _FIDAP-PDU -- FrameID Application Service PDU
_FIDAP-PDU := _SYNC-FIDAP-PDU
               | _OUTPUT-FIDAP-PDU
               | _INPUT-FIDAP-PDU
               | _TOKEN-FIDAP-PDU
               | _REQ_SEND-FIDAP-PDU
               | _MEAS_DLY-CMD-FIDAP-PDU | _MEAS_DLY-RSP-FIDAP-PDU
               | _READ_CDO-CMD-FIDAP-PDU | _READ_CDO-RSP-FIDAP-PDU
               | _WRITE_CDO-CMD-FIDAP-PDU | _WRITE_CDO-RSP-FIDAP-PDU
               | _MSG-FIDAP-PDU
               | _ASync_IND-FIDAP-PDU
               | _EVD-FIDAP-PDU
_FDCServicePDU ::= _CMD-PDU -- FDC command PDU type
                   | _RSP-PDU -- FDC response PDU type
_MSGServicePDU ::= _MSGREQ-PDU -- MSG request PDU type
                   | _MSGRSP-PDU -- MSG response PDU type

```

4.3.3 PDUs for field system management service

4.3.3.1 Overview

FID-PDU shall be edited with a SDU in the FSM service primitive and be sent via AR. And a FSM ASE receives the PDU from the peer ASE via AR.

An FSM ASE user distinguishes the types by using the command code field as the key code.

```

_FIDAP-PDU ::= SEQUENCE {
               fpduLength  _FPDULength,
               fcmd        _FCMDCode,
               fHS,        _FHandShake,
               fpduBody    OCTET STRING SIZE (1..1496)
            }

```

- **fpduLength** field

This field shows the octet number of fpuBody. The Data Type is **_FPDULength**. It shall have three sub fields, data length, divided packet number and resend flag.

```

_FPDULength ::= SEQUENCE {
    Data length          BIT STRING SIZE (12),
    Divided packet number BIT STRING SIZE (3),
    Resend flag,         BIT STRING SIZE (1)
}

```

Data length

Octet number of MAC data, not including the data length itself (2 octets)

Divided packet number

Store the packet number when packets are divided into multiple packets due to size restriction of the MAC data. (Without dividing: 0)

Resend flag

A flag indicating resending of packets. Set to 1 upon resending.

- fcmd field

A command code field to indicate the code for identifying the contents of fpduBody. This field shall be encoded as _FCMDCode for Unsigned8. Already defined common commands are indicated as _PrimaryFCMD.

```

_FCMDCode      ::= Unsigned8
_PrimaryFCMD   ::= _FCMDCode (
    SYNC ('01'H),      1)
    OUTPUT ('02'H),    1)
    INPUT ('03'H),     1)
    TOKEN ('04'H),     1) 2)
    REQ_SEND ('05'H),  1) 2)
    MEAS_DLY ('06'H),  1)
    READ_CDO ('07'H),
    WRITE_CDO ('08'H),
    MSG ('09'H),
    ASYNC_IND ('0A'H),
    EVD ('0B'H)
)

```

- 1) They cannot be used for _IP4-UDP-PDU and _IP6-UDP-PDU. They cannot be forwarded to an external IP network.
- 2) While the send right moves from the source to the destination for a token, the right to send a sending request remains at the source side and one sending request is made to the destination.

- fHS field

This field shall indicate handshake/notification. It shall have five meaningful sub fields, data length, divided packet number and resend flag.

```

_FHandShake ::= SEQUENCE {
    Request / Response flag      BIT STRING SIZE (1),
    Completed flag               BIT STRING SIZE (1),
    reserve1                     BIT STRING SIZE (1),
    Network reset start flag     BIT STRING SIZE (1),
    Network reset start confirmation flag BIT STRING SIZE (1),
    Network reset start completed flag BIT STRING SIZE (1),
    reserve2                     BIT STRING SIZE (1)
}

```

Request / Response flag

Flag to identify request or response

- 0: Request;
- 1: Response.

Completed flag

Flag to identify acknowledgment or negative acknowledgment

- 0: Acknowledgment;
- 1: Negative acknowledgment.

Network reset start flag

Flag to notify network reset start

- 0: Not notify;
- 1: Notify

Network reset start confirmation flag

Flag to notify network reset start confirmation

- 0: Not notify;
- 1: Notify.

Network reset start completed flag

Flag to notify network reset start completed

- 0: Not notify;
- 1: Notify.

- fpduBody field

This is a data field. The data format will be explained by command code in the following subclauses. In this subclause, node address common to all commands, which is included in the data, are specified. The configuration of destination address is the same as that of source address, as shown in Table 9.

Table 9 – Node address

Alignment [octet]	Name	Data type	Descriptions	Remarks
0 to 1	Node address	Unsigned16	ID number of the node '0000'H to 'FFEF'H: for unicast 'FFF0'H to 'FFFE'H: for multicast 'FFFF'H: for broadcast	An exclusive number within the network shall be allocated.

4.3.4 Detailed definitions of _FID-PDUs

4.3.4.1 SYNC command

The SYNC command shall be used for sync information notification from the sync master.

```

_SYNC-FIDAP-PDU      ::= _FIDAP-PDU ( WITH COMPONENTS
                        { ..., fcmd (sync), fHS,
                          fpduBody (fcmdBody _SYNC-FIDAPBody)
                        }
                      )

_SYNC-FIDAPBody      ::= SEQUENCE {
                        pDstAddress    Unsigned16,
                        pSrcAddress    Unsigned16,
                        pCycleCounter  Unsigned16,
                        reserve        Unsigned32,
                        pTimeStamp     Unsigned64,
                        pUserData      OCTET STRING SIZE (0..24)
                      }

```

- fcmd field
This field shall indicate SYNC command.
- fHS field
This is a data field to indicate handshake/notification with the following value range: '00'H: Normally (When not in use);
 - '10'H (NRS): Network reset start;
 - '40'H (NRS_COMP): Network reset start completed.
- pSrcAddress field
This field shall indicate the source node address. Set the sync master address (see Table 9).
- pCycleCounter field
This field shall be used to increment a counter by one every transmission cycle. This shall be used to handle multiple transmission cycles.
- pTimeStamp (TS_fsync) field
This field stores a time stamp. (Unit: 1 ns)
- pUserData field
This field stores user defined data. This is an option area. When sending a packet where sync command is stored, the common multicast address shall be used as the destination MAC address (see Table 4).

4.3.4.2 OUTPUT command

The OUTPUT command is used for command data from a master to slaves. There are two types of output data formats. The basic format shall be used for one node destination. The extended format shall be made by joining multiple basic formats to multiple node destinations. For user data, output data for the number of devices in the source node shall be stored.

```

_OUTPUT-FIDAP-PDU ::= _FIDAP-PDU ( WITH COMPONENTS
    { ..., fcmd (output), fHS,
      fpduBody (fcmdBody _OUTPUT-FIDAPBody)
    }
  )

_OUTPUT-FIDAPBody ::= SEQUENCE {                                     -- Basic format
    pDstAddress  Unsigned16,
    reserve      Unsigned2,
    pLength      Unsigned14,
    pUserData    _FDCServicePDU +
  }
| _OUTPUT-FIDAPBody                                             -- Extended format

```

- fcmd field
- This field shall indicate OUTPUT command. fHS field
This field shall indicate handshake/notification with the following value range:
 - '00'H: Normally (When not in use);
 - '10'H (NRS): Network reset start;
 - '20'H (NRS_CONF): Network reset start confirmation.
- pDstAddress field
This field shall indicate the destination node address. The address of slaves to receive output data shall be set (see Table 9).
- pLength field
This field shall indicate the user data octet number.
- pUserData field
This field stores output data. _FDCServicePDU shall be stored as the output data by device for all the devices (see 4.3.5).

4.3.4.3 INPUT command

The INPUT command is used for command data from slaves to a master. For user data, input data for the number of devices in the source node shall be stored.

```

_INPUT-FIDAP-PDU ::= _FIDAP-PDU ( WITH COMPONENTS
    { ..., fcmd (input), fHS,
      fpduBody (fcmdBody _INPUT-FIDAPBody)
    }
  )

_INPUT-FIDAPBody ::= SEQUENCE {
    pSrcAddress  Unsigned16,
    reserve      Unsigned2,
    pLength      Unsigned14,
    pUserData    _FDCServicePDU +
  }

```

- fcmd field
This field shall indicate INPUT command.

- fHS field
This field shall indicate handshake/notification with the following value range:
 - '00'H: Normally (When not in use);
 - '10'H (NRS): Network reset start.
- pSrcAddress field
This field shall indicate the source node address. Address of the slaves which sent input data shall be set (see Table 9).
- pLength field
This field shall indicate the user data octet number.
- pUserData field
This field stores input data. _FDCServicePDU shall be stored as the input data by device for all the devices (see 4.3.5).

4.3.4.4 TOKEN command

The TOKEN command is used to notify receiving/giving of the send right between masters.

```

_TOKEN-FIDAP-PDU      ::= _FIDAP-PDU ( WITH COMPONENTS
                        { ..., fcmd (token), fHS,
                          fpduBody (fcmdBody _TOKEN-FIDAPBody)
                        }
                        )

_TOKEN-FIDAPBody      ::= SEQUENCE {
                        pDstAddress      Unsigned16,
                        pSrcAddress      Unsigned16,
                        reserve          Unsigned8,
                        pTokenType       Unsigned8
                        }
  
```

- fcmd field
This field shall indicate TOKEN command.
- fHS field
This field shall indicate handshake/notification with the following value range:
 - '00'H: Normally (not used).
- pDstAddress field
This field shall indicate the destination node address to which the send right is given. The destination master address to which the send right is given shall be set (see Table 9).
- pSrcAddress field
This field shall indicate the source node address that gives the send right. The source master address which gives the send right shall be set (see Table 9).
- reserve
This is a reserve area. Set to 0.
- pTokenType field
This field shall indicate token types with the following value range:
 - 1 (MST_MEAS_DLY): Master delay measurement;
 - 2 (CONFIG): Configuration.

4.3.4.5 REQ_SEND command

The REQ_SEND command shall be used when making a request to send a packet. There are three types of send request formats by request type. They shall be respectively used as the send command for input data retransmission request, delay measurement request, sync notification and network reset start notification.

```

_REQ_SEND-FIDAP-PDU ::= _FIDAP-PDU ( WITH COMPONENTS
    { ..., fcmd (req_send), fHS,
      fpduBody (fcmdBody _REQ_SEND-FIDAPBody)
    }
  )
_REQ_SEND-FIDAPBody ::= SEQUENCE {
    reserve1          Unsigned16,
    pPollingType      Unsigned8,
    pNumAddr          Unsigned8,
    ( pReqDstAddress  Unsigned16 )
  }
| SEQUENCE {
    pDstAddress      Unsigned16,
    pPollingType     Unsigned8,
    pCalcType        Unsigned8,
    pSyncAddr        Unsigned16,
    pSyncMACAddr     Unsigned48
  }
| SEQUENCE {
    pDstAddress      Unsigned16,
    pPollingType     Unsigned8,
    reserve2         Unsigned8
  }
}

```

-- Input data retransmission

-- Delay measurement request

-- Sync notification and network reset start notification

- fcmd field
This field shall indicate SEND command.
- fHS field
This field shall indicate handshake/notification with the following value:
 - '00'H: Normally (not used).
- reserve1
This is a reserve area. Set to 0.
- pPollingType field
This field shall indicate request types with the following value range:
 - 1 (INPUT): Input data retransmission;
 - 2 (MEAS_DLY): Delay measurement request;
 - 3 (SYNC): Sync notification;
 - 4 (NET_NOTI): Network reset start notification.
- pNumAddr field
This field shall indicate the number of addresses and this is valid when the request type is 1. The number of stored addresses shall be set.

- **pReqDstAddress field**
This field shall indicate the node address of the input data transmission request destination and this is valid when the request type is 1. The address of the slave which requests the input data retransmission shall be set (see Table 9).
- **pDstAddress field**
This field shall indicate the destination node address, and this is valid when the request type is 2, 3, or 4. The address of the slave to receive sending request shall be set (see Table 9).
- **pCalcType field**
This field shall indicate the calculation type of the delay measurement, and this is valid when the request type is 2. It takes the following values range:
 - 0 (INIT): Initialization processing
 - 1 (PAP): Plug-and-play joining processing
- **pSyncAddr field**
This field shall indicate the sync master address as the reply destination, and this is valid when the request type is 2. The sync master address shall be set (see Table 9).
- **pSyncMACAddr field**
This field shall indicate the sync master MAC address as the reply destination, and this is valid when the request type is 2. The sync master MAC address shall be set.
- **reserve2**
This is a reserve area. Set to 0.

4.3.4.6 MEAS_DLY command and response

The MEAS_DLY shall be used for command and response of delay measurements of transmission paths (see 8.3.4.1).

```

_MEAS_DLY-CMD-      ::= _FIDAP-PDU ( WITH COMPONENTS
_FIDAP-PDU
    { ..., fcmd (measure_dly), fHS,
      fpduBody (fcmdBody  _MEAS_DLY-CMD-FIDAPBody)
    }
  )

_MEAS_DLY-CMD-      ::= SEQUENCE {
_FIDAPBody
    pDstAddress      Unsigned16,
    pSrcAddress      Unsigned16,
    reserve          Unsigned16,
    pTimeStamp3      Usingned32
  }

_MEAS_DLY-RSP-      ::= _FIDAP-PDU ( WITH COMPONENTS
_FIDAP-PDU
    { ..., fcmd (measure_dly), fHS,
      fpduBody (fcmdBody  _MEAS_DLY-RSP-FIDAPBody)
    }
  )
    
```

```

_MEAS_DLY-RSP-      ::= SEQUENCE {
_FIDAPBody
    pDstAddress      Unsigned16,
    pSrcAddress      Unsigned16,
    reserve          Unsigned16,
    pTimeStampT3     Usingned32,
    pTimeStampT4     Usingned32
}

```

- **fcmd field**
This field shall indicate MEAS_DLY.
- **fHS field**
This field shall indicate handshake/notification with the following value range to identify command or response:
 - '00'H (REQ): Request;
 - '01'H (ACK): Acknowledgment;
 - '03'H (NACK): Negative acknowledgment.
- **pDstAddress field**
This field shall indicate the destination node address. The sync master address or the broadcast address shall be set (see Table 9).
- **pSrcAddress field**
This field shall indicate the source node address. The source node address shall be set (see Table 9).
- **reserve**
This is a reserve area. Set to 0.
- **pTimeStampT3 field**
This field shall indicate the time stamp T3. The time stamp upon sending of the delay measurement request shall be set. (Unit: 1 ns)
- **pTimeStampT4 field**
This field shall indicate the time stamp T4. The time stamp upon receiving of the delay measurement request shall be set. (Unit: 1 ns)

4.3.4.7 READ_CDO command and response

The READ_CDO shall be used for a DLO ID read command/response.

```

_READ_CDO-CMD-      ::= _FIDAP-PDU( WITH COMPONENTS
_FIDAP-PDU
    { ..., fcmd (read_cdo), fHS,
      fpduBody (fcmdBody  _READ_CDO-CMD-FIDAPBody)
    }
)

_READ_CDO-CMD-      ::= SEQUENCE {
_FIDAPBody
    pDstAddress      Unsigned16,
    pSrcAddress      Unsigned16,
    reserve          Unsigned8,
    pNumCDO_ID       Usingned8,
    ( pCDO_ID        Usingned16 ) +
}

```

```

_READ_CDO-RSP- ::= _FIDAP-PDU ( WITH COMPONENTS
FIDAP-PDU
    { ..., fcmd (read_cdo), fHS,
      fpduBody (fcmdBody _READ_CDO-RSP-FIDAPBody)
    }
  )

_READ_CDO-RSP- ::= SEQUENCE {
FIDAPBody
    pDstAddress    Unsigned16,
    pSrcAddress    Unsigned16,
    reserve        Unsigned8,
    pNumCDO_ID    Usingned8,
    ( SEQUENCE {
        pCDO_ID    Usingned16,
        pResult    Usingned16,
        pCDO_Data  OCTET STRING SIZE (1..48),
        padding    OCTET STRING SIZE (0..3)
      } ) +
  }

```

- fcmd field
This field shall indicate READ_CDO.
- fHS field
This field shall indicate handshake/notification with the following value range to identify command or response:
 - '00'H (REQ): Request;
 - '01'H (ACK): Acknowledgment;
 - '03'H (NACK): Negative acknowledgmente.
- pDstAddress field
This field shall indicate the destination node address. For a command, the read destination node address shall be set. For a response, the source node address shall be set (see Table 9).
- pSrcAddress field
This field shall indicate the source node address. The source node address shall be set (see Table 9).
- reserve
This is a reserve area. Set to 0.
- pNumCDO_ID field
This field shall indicate the read target CDO_ID number. The stored CDO_ID number shall be set.
- pCDO_ID field
This field shall indicate the read target CDO_ID. CDO_ID shall be set.
- pResult field
This field shall indicate the result of CDO ID read with the following value range:
 - 0: Normally completed;
 - 1: CDO_ID error (corresponding COD_ID does not exist);
 - 2: Cannot read (reading of CDO_ID with a write only attribute);

– 3: Reading of unimplemented CDO_ID.

- pCDO_Data field

This field shall indicate the CDO ID value. The value that has been read according to the CDO ID data size shall be set.

- padding field

This shall be inserted to become a 4-octet border according to the CDO ID data size. Set to 0.

4.3.4.8 WRITE_CDO command and response

The WRITE_CDO shall be used for a DLO ID read request/response.

```

_WRITE_CDO-CMD-      ::= _FIDAP-PDU ( WITH COMPONENTS
FIDAP-PDU
    { ..., fcmd (write_cdo), fHS,
      fpduBody (fcmdBody  _WRITE_CDO-CMD-FIDAPBody)
    }
  )

_WRITE_CDO-CMD-      ::= SEQUENCE {
FIDAPBody
    pDstAddress      Unsigned16,
    pSrcAddress      Unsigned16,
    reserve1         Unsigned8,
    pNumCDO_ID       Usingned8,
    ( SEQUENCE {
        pCDO_ID      Usingned16,
        reserve2     Usingned16,
        pCDO_Data    OCTET STRING SIZE (1..48),
        padding      OCTET STRING SIZE (0..3)
    })
  }

_WRITE_CDO-RSP-      ::= _FIDAP-PDU ( WITH COMPONENTS
FIDAP-PDU
    { ..., fcmd (write_cdo), fHS,
      fpduBody (fcmdBody  _WRITE_CDO-RSP-FIDAPBody)
    }
  )

_WRITE_CDO-RSP-      ::= SEQUENCE {
FIDAPBody
    pDstAddress      Unsigned16,
    pSrcAddress      Unsigned16,
    reserve1         Unsigned8,
    pNumCDO_ID       Usingned8,
    ( SEQUENCE {
        pCDO_ID      Usingned16,
        pResult      Usingned16 )
    }) +
  )

```

- fcmd field

This field shall indicate WRITE_CDO.

- fHS field
This field shall indicate handshake/notification with the following value range to identify command or response:
 - '00'H (REQ): Request;
 - '01'H (ACK): Acknowledgment;
 - '03'H (NACK): Negative acknowledgment.
- pDstAddress field
This field shall indicate the destination node address. For a command, the write destination node address shall be set. For a response, the source node address shall be set (see Table 9).
- pSrcAddress field
This field shall indicate the source node address. The source node address shall be set (see Table 9).
- reserve1
This is a reserve area. Set to 0.
- pNumCDO_ID field
This field shall indicate the write target CDO_ID number. The stored CDO_ID number shall be set.
- pCDO_ID field
This field shall indicate the write target CDO_ID. CDO_ID shall be set.
- reserve2
This is a reserve area. Set to 0.
- pCDO_Data field
This field shall indicate the CDO_ID value. The value to write shall be set according to the CDO_ID data size.
- padding field
This shall be inserted to become a 4-octet border according to the CDO_ID data size. Set to 0.
- pResult field
This field shall indicate the result of CDO ID write with the following value range:
 - 0: Normally completed;
 - 1: CDO_ID error (corresponding COD_ID does not exist);
 - 2: Cannot write (writing of CDO_ID with a read only attribute);
 - 3: Writing of unimplemented CDO_ID;
 - 4: Writing value error.

4.3.4.9 MSG command

The MSG command shall be used to notify receiving and giving of message between nodes.

```

_MSG-FIDAP-PDU      ::= _FIDAP-PDU ( WITH COMPONENTS
                        { ..., fcmd (msg), fHS,
                          fpduBody (fcmdBody _MSG-FIDAPBody)
                        }
                      )
_MSG-FIDAPBody      ::= SEQUENCE {
                        pDstAddress  Unsigned16,
                        pSrcAddress  Unsigned16,
                        pMsgControl  _MSG_CONTROL,
                        pUserData    _MSGServicePDU
                      }

```

- **fcmd field**
This field shall indicate MSG command.
- **fHS field**
This field shall indicate handshake/notification with the following value:
– '00'H: Normally (not used).
- **pDstAddress field**
This field shall indicate the destination node address. The destination node address shall be set (see Table 9).
- **pSrcAddress field**
This field shall indicate the source node address. The source node address shall be set (see Table 9).
- **pMsgControl field**
This field shall indicate the message control information. There are two types of message control information formats: I-format and S-format.

```

_MSG_CONTROL        ::= SEQUENCE {                                     -- I-format
                        pNR      BIT STRING SIZE (7),
                        pPF      BIT STRING SIZE (1),
                        pNS      BIT STRING SIZE (1),
                        pType    BIT STRING SIZE (1)
                      }
| SEQUENCE {                                     -- S-format
                        pNR      BIT STRING SIZE (7),
                        reserve1 BIT STRING SIZE (1),
                        reserve2 BIT STRING SIZE (4),
                        pStype    BIT STRING SIZE (2),
                        reserve3 BIT STRING SIZE (1),
                        pType    BIT STRING SIZE (1)
                      }

```

pNR

A field to indicate the serial number of receiving data.

pPF

A flag to identify polling/final fragment.

- 0: Polling;
- 1: Final fragment.

pNS

A field to indicate the serial number of sending data.

pType

A flag to identify the format.

- 0: I-format;
- 1: S-format.

reserve1

A reserve area. Set to 1.

reserve2

A reserve area. Set to 0.

pSType

A field to indicate the S-format frame type.

reserve3

A reserve area. Set to 0.

- pUserData field

This field shall store the message. It shall store a part of divided _MSGServicePDU or _MSGServicePDU (see 4.3.6).

4.3.4.10 ASYNC_IND command

The ASYNC_IND command shall be used for an event notification.

```

_ASYNC_IND-FIDAP-PDU ::= _FIDAP-PDU ( WITH COMPONENTS
    { ..., fcmd (async_ind), fHS,
      fpduBody (fcmdBody _ASYNC_IND-FIDAPBody)
    }
  )
_ASYNC_IND-FIDAPBody ::= SEQUENCE {
    pDstAddress  Unsigned16,
    pSrcAddress  Unsigned16,
    pEventType   Unsigned8,
    pParam       _ASYNC_IND_PARAM,
    pUserData    _ASYNC_IND_DATA | empty
  }

```

- fcmd field

This field shall indicate ASYNC_IND command.

- **fHS field**
This field shall indicate handshake/notification with the following value:
 - '00'H: Normally (not used).
- **pDstAddress field**
This field shall indicate the destination node address. The destination master address shall be set. Note that this shall be set to broadcast for the event type 1 (see Table 9).
- **pSrcAddress field**
This field shall indicate the source node address. The source master address shall be set (see Table 9).
- **pEventType field**
This field shall indicate the event type with the following value range:
 - 1 (NEI_NOTI): Neighbouring node notification;
 - 2 (SUP_NOTI): Start-up notification;
 - 3 (SUP_RESP): Start-up notification response);
 - 4 (NEI_LIST): Neighbouring node list notification;
 - 5 (MEAS_DLY): Delay measurement;
 - 6 (MEAS_DLY_COMP): Delay measurement completed;
 - 7 (CHG_ADDR): Node address change;
 - 8 (NRS_COMP): Network reset start completed.
- **pParam field**
This field shall indicate the parameter, and this is valid for event type 3. For other event types, this shall be set to 0.

```

_ASYNC_IND_PARAM ::= SEQUENCE {
    pNodeType      BIT STRING SIZE (4),
    pPossibilitySM  BIT STRING SIZE (1),
    pMeasureDly     BIT STRING SIZE (1),
    reserve         BIT STRING SIZE (2)
}

```

pNodeType

A field to indicate the operation type. .

- 1: Master;
- 2: Slave.

pPossibilitySM

A flag to identify SM handling capability.

- 0: With SM capability;
- 1: Without SM capability.

pMeasureDly

This is a flag to identify the master delay measurement completion and this shall be set only for a master. For a slave, this shall be set to 0.

- 0: Master delay measurement unimplemented;
- 1: Master delay measurement completed.

reserve

This is a reserve area. Set to 0,