

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 6-25: Application layer protocol specification – Type 25 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 6-25: Spécification du protocole de la couche application – Éléments
de type 25**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2019 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 l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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

Tel.: +41 22 919 02 11
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 corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables 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 It also gives information on projects, replaced and withdrawn publications.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 online and once a month by email.

IEC 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: sales@iec.ch.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) 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 IEC

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

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC 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.

IEC Just Published - webstore.iec.ch/justpublished

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

Service Clients - webstore.iec.ch/csc

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

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

67 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 6-25: Application layer protocol specification – Type 25 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 6-25: Spécification du protocole de la couche application – Éléments
de type 25**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-9173-3

**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.....	7
INTRODUCTION.....	9
1 Scope.....	10
1.1 General.....	10
1.2 Specification	11
1.3 Conformance	11
2 Normative references	11
3 Terms, definitions, symbols, abbreviations and conventions	12
3.1 Reference model terms and definitions	12
3.1.1 ISO/IEC 7498-1 terms.....	12
3.1.2 ISO/IEC 8822 terms.....	12
3.1.3 ISO/IEC 9545 terms.....	12
3.1.4 ISO/IEC 8824-1 terms.....	13
3.2 Additional Type 25 terms and definitions.....	13
3.3 Symbols and abbreviations	15
3.4 Conventions.....	16
3.4.1 General conventions.....	16
3.4.2 Conventions for class definitions	16
3.4.3 Conventions for bit description in octets	16
3.4.4 Conventions for state machine descriptions	17
4 FAL syntax description	18
4.1 FAL PDU type S abstract syntax	18
4.1.1 Basic abstract syntax.....	18
4.2 FAL PDU type N abstract syntax	22
4.2.1 Basic abstract syntax.....	22
4.2.2 CyclicData-PDU.....	22
4.2.3 MulticastData-PDU	23
4.2.4 PtoPData-PDU.....	23
4.2.5 Aliveinfo-PDU	23
4.2.6 Aliveinfo6-PDU	23
4.2.7 Inq-PDU	24
4.2.8 Ninq-PDU	24
4.2.9 Reply-PDU	24
4.2.10 RetransEnq-PDU	24
4.2.11 RetransConfirm-PDU	24
4.2.12 RetransNak-PDU	25
4.3 Data type assignments for type S.....	25
4.4 Data type assignments for type N	25
5 FAL transfer syntax	26
5.1 Encoding rules.....	26
5.1.1 Unsigned encoding	26
5.1.2 Octet string encoding.....	27
5.1.3 SEQUENCE encoding.....	27
5.2 FALPDU type S elements encoding.....	27
5.2.1 RCL_header	27
5.2.2 RHE-PDU	28

5.2.3	LCC-PDU	30
5.2.4	LCA-PDU	30
5.2.5	LCN-PDU	31
5.2.6	LNA-PDU	32
5.2.7	SCR-PDU	32
5.2.8	Cyclic_S-PDU	32
5.2.9	Cyclic_header	33
5.2.10	Control-PDU	33
5.2.11	RMTCTL-PDU	33
5.2.12	INFO-PDU	34
5.3	FALPDU type N elements encoding	35
5.3.1	General	35
5.3.2	FALAR-N Header	35
5.3.3	CyclicData-PDU	49
5.3.4	MulticastData-PDU	50
5.3.5	PtoP Data-PDU	50
5.3.6	Aliveinfo-PDU	50
5.3.7	Aliveinfo6-PDU	53
5.3.8	Inq-PDU	54
5.3.9	Ninq-PDU	55
5.3.10	Reply-PDU	55
5.3.11	RetransEnq-PDU	56
5.3.12	RetransConfirm-PDU	56
5.3.13	RetransNak-PDU	57
6	Structure of the FAL protocol state machine	58
7	FAL service protocol machine (FSPM)	58
7.1	Overview	58
7.2	FSPM type S	58
7.2.1	Overview	58
7.2.2	Interface of cyclic communication to FAL users	59
7.2.3	State machine of FSPM	61
7.3	FSPM type N	62
7.3.1	Overview	62
7.3.2	FSPM	63
8	Application relationship protocol machine (ARPM)	65
8.1	ARPM type S	65
8.1.1	Overview	65
8.1.2	Cyclic control	66
8.1.3	Remote control	70
8.1.4	RCL communication control	75
8.1.5	RT communication control	79
8.2	ARPM type N	83
8.2.1	Overview	83
8.2.2	General control	83
8.2.3	Cyclic transmission control	85
8.2.4	Acyclic transmission control	90
8.2.5	RT communication control	105
9	DLL mapping protocol machine (DMPM)	121

9.1	DMPM type S.....	121
9.2	DMPM type N.....	122
9.2.1	General	122
9.2.2	Communication port in transport layer	122
9.2.3	Quality of Service	122
	Bibliography.....	124
	Figure 1 – Bit description in octets	16
	Figure 2 – hd_sa.....	35
	Figure 3 – hd_da.....	36
	Figure 4 – Valid sequence number for reception message	40
	Figure 5 – hd_m_ctl	40
	Figure 6 – Valid reception packet sequence number	44
	Figure 7 – Node-list	55
	Figure 8 – Relationships between protocol machines	58
	Figure 9 – Structure of FSPM type S.....	59
	Figure 10 – Shared memory allocation in type S network	61
	Figure 11 – Structure of FSPM type N.....	63
	Figure 12 – Structure of ARPM type S	66
	Figure 13 – Sequence of cyclic communication.....	67
	Figure 14 – The primitives for cyclic control	67
	Figure 15 – The primitives for Remote control.....	70
	Figure 16 – The primitives for RCL communication control.....	75
	Figure 17 – The primitives for RT communication control	80
	Figure 18 – Structure of ARPM type N	83
	Figure 19 – Primitives of Cyclic transmission control.....	85
	Figure 20 – Primitives of acyclic transmission control	91
	Figure 21 – DSCP format.....	123
	Figure 22 – IEEE 802.1Q tag frame format	123
	Table 1 – State transition descriptions	17
	Table 2 – Descriptions of state machine elements	17
	Table 3 – Conventions used in state machine	17
	Table 4 – Frame Class.....	27
	Table 5 – DA_STaddress – DA_STaddress.....	27
	Table 6 – DA_MACaddress.....	27
	Table 7 – CMD field format	28
	Table 8 – Send Direction	28
	Table 9 – RHE ReceiveStatus.....	29
	Table 10 – Physical Linkdown.....	29
	Table 11 – RHE_pattern 1~4.....	30
	Table 12 – LCC-Kind	30
	Table 13 – RCL Status.....	31
	Table 14 – hd_h_type	31

Table 15 – Usage of Mgn or Lnn	37
Table 16 – Detailed conditions for sequence number check of reception message	39
Table 17 – Valid bits of hd_m_ctl	40
Table 18 – Specified TCD	41
Table 19 – hd_pkind	42
Table 20 – PDU with an effective hd_pseq	42
Table 21 – Detailed conditions for sequence number check of reception packet (Multicast communication with retransmission)	44
Table 22 – Detailed conditions for packet sequence number check	45
Table 23 – Relation between message transmission/reception	46
Table 24 – hd_mode	46
Table 25 – Message priority level	46
Table 26 – Value of α	47
Table 27 – Example of header information for a UDP message fragmentation	47
Table 28 – Example of header information for a TCP message fragmentation	47
Table 29 – inqid_inq_sa value	48
Table 30 – inqid_tr_adr value	48
Table 31 – inqid_inq_seq value	49
Table 32 – Relationship between inqid_id_seq and inqid_tr_adr	49
Table 33 – Type of an alive-message	51
Table 34 – Type of an alive-message protocol	51
Table 35 – Time of each al_mode	52
Table 36 – Status change of tasks	52
Table 37 – Change of tasks content	52
Table 38 – The threshold of transmission factor	60
Table 39 – Example of the traffic control configuration menu	60
Table 40 – Cyclic data state table	61
Table 41 – Acyclic data state table	62
Table 42 – Cyclic data state table	64
Table 43 – Acyclic data state table	64
Table 44 – Cyclic control state table	68
Table 45 – Cyclic control functions	69
Table 46 – Cyclic control variables	70
Table 47 – Remote control state table	71
Table 48 – Remote control functions	74
Table 49 – Remote control variables	75
Table 50 – RCL communication control state table	76
Table 51 – RCL communication control functions	77
Table 52 – RCL communication control variables	79
Table 53 – RT communication control state table	80
Table 54 – RT communication control functions	82
Table 55 – RT communication control variables	82
Table 56 – Cyclic transmission control state table	86

Table 57 – Cyclic transmission control functions	88
Table 58 – Cyclic transmission control variables	90
Table 59 – Acyclic transmission control state table	91
Table 60 – Acyclic transmission control functions	101
Table 61 – Acyclic transmission control variables	104
Table 62 – RT communication control state table	105
Table 63 – RT communication control functions	116
Table 64 – RT communication control variables	120
Table 65 – ARPM to DL mapping	122
Table 66 – Assignment policy of communication ports	122
Table 67 – Default DSCP, IEEE 802.1D and IEEE 802.1Q priority mapping	123

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 6-25: Application layer protocol specification –
Type 25 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.

International Standard IEC 61158-6-25 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

FDIS	Report on voting
65C/948/FDIS	65C/956/RVD

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

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

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

The committee has decided that the contents of this publication will remain unchanged until the stability 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.

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

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 standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-25: Application layer protocol specification – Type 25 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 International Standard 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 25 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 different Types of the fieldbus Application Layer in terms of:

- a) the abstract syntax defining the application layer protocol data units conveyed between communicating application entities,
- b) the transfer syntax defining the application layer protocol data units conveyed between communicating application entities,
- c) the application context state machine defining the application service behavior visible between communicating application entities; and
- d) the application relationship state machines defining the communication behavior visible between communicating application entities.

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

- a) define the wire-representation of the service primitives defined in IEC 61158-5-25, and
- b) define the externally visible behavior associated with their transfer.

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

FAL services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented Application Service Elements (ASEs) and a Layer Management Entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can

send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this document to provide access to the FAL to control certain aspects of its operation.

1.2 Specification

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 61158-5-25. 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 subparts of IEC 61158-6.

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.

There is no conformance of equipment to the application layer service definition standard. Instead, 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 IEC 61784-1 and IEC 61784-2 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 61158-3-25:2019, *Industrial communication networks – Fieldbus specifications – Part 3-25: Data-link layer service definition – Type 25 elements*

IEC 61158-5-25:2019, *Industrial communication networks – Fieldbus specifications – Part 5-25: Application layer service definition – Type 25 elements*

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

ISO/IEC 8822, *Information technology – Open Systems Interconnection – Presentation service definition*

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

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

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

IEEE Std 802.1D, *IEEE Standard for Local and metropolitan area networks – Media access Control (MAC) Bridges*; available at <http://www.ieee.org> [viewed 2018-09-17]

IEEE Std 802.1Q, *IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks*; available at <http://www.ieee.org> [viewed 2018-09-17]

IETF RFC 768, *User Datagram Protocol*; available at <http://www.ietf.org> [viewed 2018-09-17]

IETF RFC 791, *Internet Protocol*; available at <http://www.ietf.org> [viewed 2018-09-17]

3 Terms, definitions, symbols, abbreviations and conventions

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

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

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

3.1 Reference model terms and definitions

3.1.1 ISO/IEC 7498-1 terms

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

- a) application entity;
- b) application process;
- c) application protocol data unit;
- d) application service element;
- e) application entity invocation;
- f) application process invocation;
- g) application transaction;
- h) real open system;
- i) transfer syntax.

3.1.2 ISO/IEC 8822 terms

For the purposes of this document, the following terms given in ISO/IEC 8822 apply:

- a) abstract syntax;
- b) presentation context.

3.1.3 ISO/IEC 9545 terms

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

- a) application-association;
- b) application-context;
- c) application context name;
- d) application-entity-invocation;
- e) application-entity-type;
- f) application-process-invocation;
- g) application-process-type;
- h) application-service-element;

- i) application control service element.

3.1.4 ISO/IEC 8824-1 terms

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

- a) object identifier;
- b) type.

3.2 Additional Type 25 terms and definitions

For the purpose of this document, the following definitions also apply.

NOTE Many definitions are common to more than one protocol Type; they are not necessarily used by all protocol Types.

3.2.1

ADP message

message conveyed by an autonomous decentralized system protocol

3.2.2

alive-message

message reporting own node state, periodically transmitted

3.2.3

block

basic unit of data transferred in a cyclic communication, each having a size of 64 octets

3.2.4

category N_f

category of an autonomous decentralized system protocol (full specifications) in type N

3.2.5

category N_l

category of an autonomous decentralized system protocol (light weight specifications) in type N

3.2.6

control communication

acyclic data communication for higher time-critical applications in type S network

3.2.7

cyclic communication

periodic data communication for real-time communication

3.2.8

cyclic transfer memory

memory which is allocated to each node in the data field, which each node transmits periodically for the purpose of sharing of this memory area logically

3.2.9

data field

logical place through which specific data passes, corresponding to real networks

3.2.10

domain

administrative set consisting of multiple data fields

**3.2.11
duplex LAN**

two different network-paths between end nodes

**3.2.12
equipment**

physical hardware connected to the network

EXAMPLE Station, device, and server.

**3.2.13
information communication**

acyclic data communication for lower time-critical application in type S network

**3.2.14
logical node, node**

equipment in data fields, where an autonomous decentralized system's function is installed

**3.2.15
message mode**

identifier that indicates the uses of a message (whether an online message or a test message)

EXAMPLE Online message or a test message.

**3.2.16
multicast group**

group of nodes belonging to a data field to determine whether to receive same data from other node

**3.2.17
node list**

bit array structure of specifying a node to receive a request data

**3.2.18
node mode**

identifier that indicates the usage of a node

EXAMPLE Online node or a test node.

**3.2.19
primary local area network**

local area network (LAN) used in normal state in a duplex LAN system

**3.2.20
priority control**

transmission order control for the packets arrived from networks, according to the virtual local area network (VLAN)

**3.2.21
ring control communication**

control for the type S ring network using ring control (RCL) frames

**3.2.22
remote control**

control for other nodes connected to a network

**3.2.23
secondary local area network**

local area network (LAN) for backup of a primary LAN in a duplex LAN system

3.2.24**transaction code**

information to identify the characteristics of the message within a PDU

3.3 Symbols and abbreviations

ADP	Autonomous Decentralized Protocol
ADS-net	Autonomous Decentralized System – network
AE	Application Entity
AL	Application Layer
AP	Application Process
APDU	Application Protocol Data Unit
APO	Application Process Object
AR	Application Relationship
AREP	Application Relationship Endpoint
ARPM	Application Relationship Protocol Machine
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
ASN.1	Abstract Syntax Notation 1
CRC	Cyclic Redundancy Check
DF	Data Field
Dfn	Data Field Number
DLL	Data-link Layer
Dmn	DoMain Number
DMPM	DLL Mapping Protocol Machine
DSCP	DiffServ Code Point
FAL	Fieldbus Application Layer
FSPM	FAL Service Protocol Machine
GMT	Greenwich Mean Time
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LAN	Local area network
LCA	Loop condition alert (Type S frame type)
LCC	Loop condition check (Type S frame type)
LCN	Loop condition notify (Type S frame type)
LLD	Logical link down
LLU	Logical link up
LNA	Loop notify answer (Type S frame type)
Lnn	Logical Node Number
LSB	Least Significant Bit
MCG	MultiCast Group
Mgn	Multicast Group Number
MSB	Most Significant Bit
MTU	Maximum Transmission Unit

OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PLD	Physical link down
QoS	Quality-of-service
RCL	Ring control
RHE	Rapid hello (Type S frame type)
RMTCTL	Remote control
RT	Real time
SCR	Station condition report (Type S frame type)
TCD	Transaction CoDe
TCP	Transmission Control Protocol
TMID	Transfer Memory Identifier
UDP	User Datagram protocol

3.4 Conventions

3.4.1 General conventions

Conventions given in IEC 61158-3-25 are also applied in this document.

3.4.2 Conventions for class definitions

This description of FAL Type 25 uses a subset of ASN.1. The following structures are used.
Selective type (CHOICE) – Represents a selection from candidate types.

Sequence type (SEQUENCE) – Represents a fixed-order list as in the following example:

```
DLPDU ::= SEQUENCE {
  preamble    Preamble,
  sfd         SFD,
  destaddr    DestAddr,
  srcaddr     SrcAddr,
  vlantag     VLANtag,
  lt          LT,
  dlsdu       FAL-PDU,
  fcs         FCS
}
```

NOTE This example shows that the DLPDU which represents the Ethernet frame is defined as SEQUENCE. The DLPDU consists of Preamble, SFD, DestAddr, SrcAddr, VLANtag, LT, FAL-PDU and FCS.

3.4.3 Conventions for bit description in octets

When identifying each bit in an octet, each bit is identified by a number as shown in Figure 1 and described as Bit n.

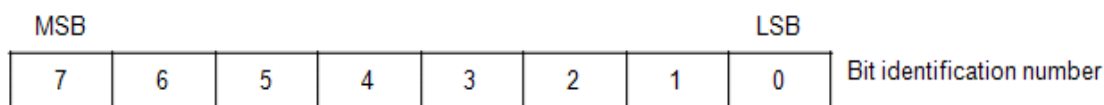


Figure 1 – Bit description in octets

When specifying multiple bits sequentially located, the range symbol (..) is used (e.g.: 7..0, specifies bits 7 through 0, inclusive).

When specifying multiple octets, the LSB of the lowest octet is assigned as 0, and bit identification numbers are assigned in an ascending order.

NOTE For example, when specifying 4 octets, the MSB of the highest octet is Bit 31, the MSB of the second octet is Bit 23, the MSB of the third octet is Bit 15, and the MSB of the lowest octet is Bit 7.

3.4.4 Conventions for state machine descriptions

The state machine description is defined in tabular form as shown in Table 1. The meanings of the elements are shown in Table 2. The conventions used in the state machines are shown in Table 3.

Each row of state table represents a state transition. The first column shows the state transition name or number. The second column shows the current state. The third column shows the events, conditions and actions. The fourth column shows the next state. When an event or condition is fulfilled, the action is performed and the state machine transits to the next state.

Table 1 – State transition descriptions

#	Current state	Event /condition => actions	Next state
---	---------------	-----------------------------------	------------

Table 2 – Descriptions of state machine elements

Description element	Meaning
#	Number of the transition
Current state, Next state	Names of the originating state and the target state of transition
Event	Name or description of the trigger event that fire the transition
/conditions	Boolean expression, which shall be true for the transition to be fired
=>actions	List of assignments and service or function invocations. The action should be atomic. The preceding “=>” is not part of the action
“*/ conditions” may be omitted.	

Table 3 – Conventions used in state machine

Convention	Meaning
=	Substitution of the right side for the left side
==	A logical condition to indicate an item on the left is equal to an item on the right.
!=	A logical condition to indicate an item on the left is not equal to an item on the right
<	A logical condition to indicate an item on the left is less than the item on the right
>	A logical condition to indicate an item on the left is greater than the item on the right
&&	Logical “AND”
	Logical “OR”
!	Negation operator
+ – * /	Arithmetic operator
;	Breakpoint
++	Increment operator
--	Decrement operator

4 FAL syntax description

4.1 FAL PDU type S abstract syntax

4.1.1 Basic abstract syntax

The definitions of FALPDU are shown below.

```
FAL-PDU ::= CHOICE {
    RCL-PDU,
    RT-PDU
}
```

```
RCL-PDU ::= CHOICE {
    RHE-PDU,
    LCC-PDU,
    LCA-PDU,
    LCN-PDU,
    LNA-PDU,
    SCR-PDU
}
```

```
RT-PDU ::= CHOICE {
    Cyclic_S-PDU,
    CTL-PDU,
    INFO-PDU
}
```

```
CTL-PDU ::= CHOICE {
    Control-PDU,
    RMTCTL-PDU
}
```

4.1.1.1 RCL-PDU

RCL_header to be used in each RCL-PDU is shown as follows.

```
RCL_header ::= SEQUENCE {
    Frame Class          CLASS,
    Dst Address          SEQUENCE {
        Reserved1        Unsigned8,
        DA_STaddress     ST_Address,
        DA_MACaddress    MAC_Address
    },
    Src Address          SEQUENCE {
        Reserved2        Unsigned8,
        SA_STaddress     ST_Address,
        SA_MACaddress    MAC_Address
    },
    RCL Command          RCLCMD,
    Frame Sequence       SequenceNum32,
    Reserved3            OCTET STRING(SIZE 20)
}
```

4.1.1.1.1 RHE-PDU

```

RHE-PDU ::= SEQUENCE {
    RCL_header      RCL_HEADER,
    RHE_header      SEQUENCE {
        RHE_Common  SEQUENCE {
            Send Direction      PortNum,
            RHE Sequence        SequenceNum32
        },
        Neighborhood address    SEQUENCE {
            Reserved1            Unsigned32,
            NB_STaddress         ST_Address,
            NB_MACaddress        MAC_Address
        },
        RHE ReceiveStatus       RHE_RxStatus,
        True Status              SEQUENCE {
            Reserved2            Unsigned32,
            Physical Linkdown    PLD,
            Reserved3            Unsigned32
        },
        Blocking Status         BLK_Status,
        Node Status             Node_Status,
        Reserved4               OCTET STRING(SIZE 20)
    },
    Reserved5             OCTET STRING(SIZE 64),
    RHE_pattern           SEQUENCE {
        RHE_pattern1          Pattern32,
        RHE_pattern2          Pattern32,
        RHE_pattern3          Pattern32,
        RHE_pattern4          Pattern32,
        .
        .
        .
    }
}

```

4.1.1.1.2 LCC-PDU

```

LCC-PDU ::= SEQUENCE {
    RCL_header      RCL_HEADER,
    LCC_header      SEQUENCE {
        LCC Kind      RCLKIND,
        LCC Priority    SEQUENCE {
            LCC_Priority      Priority,
            LCC_STadd         ST_Address,
            LCC_MACadd        MAC_Address
        },
        Reserved1         OCTET STRING(SIZE 52)
    }
}

```

4.1.1.1.3 LCA-PDU

```

LCA-PDU ::= SEQUENCE {
    RCL_header      RCL_HEADER,
    LCA_header      SEQUENCE {
        Reserved1      Unsigned32,
        RCL Status      RCL_Status,
        LCA Priority     SEQUENCE {
            LCA_Priority      Priority,
            LCA_STadd         ST_Address,
            LCA_MACadd        MAC_Address
        },
    },
}

```

```

        Reserved2          OCTET STRING(SIZE 48)
    }
}

```

4.1.1.1.4 LCN-PDU

```

LCN-PDU ::= SEQUENCE {
    RCL_header      RCL_HEADER,
    LCN_header      SEQUENCE {
        Reserved1      Unsigned32,
        RCL_Status     RCL_Status,
        Reserved2      OCTET STRING(SIZE 56)
    }
}

```

4.1.1.1.5 LNA-PDU

```

LNA-PDU ::= SEQUENCE {
    RCL_header      RCL_HEADER,
    LNA_header      SEQUENCE {
        Reserved1      Unsigned32,
        RCL_Status     RCL_Status,
        Reserved2      OCTET STRING(SIZE 56)
    }
}

```

4.1.1.1.6 SCR-PDU

```

SCR-PDU ::= SEQUENCE {
    RCL_header      RCL_HEADER,
    SCR_header      SEQUENCE {
        Entry Number  EntryNum,
        SendStat Time Send_Time,
        Reserved1     OCTET STRING(SIZE 52)
    },
    Event_info      SEQUENCE {
        Event0        EventData,
        Event1        EventData,
        .
        .
        .
        Event16       EventData
    }
}

```

4.1.1.2 RT-PDU

4.1.1.2.1 Cyclic_S-PDU

```

Cyclic_S-PDU ::= SEQUENCE {
    IP_header      IP_HEADER,
    UDP_header     UDP_HEADER,
    Reserved1      Unsigned16,
    Cyclic_header  CYCLIC_HEADER,
    Block_num      SEQUENCE {
        Block number1  BlockNum,
        Block number2  BlockNum,
        .
        .
        .
        Block number21 BlockNum,
        BBlock_num pattern Pattern16
    }
}

```

```

    },
    Cyclic_data      SEQUENCE {
        Block data1      BlockData,
        Block data2      BlockData,
        .
        .
        .
        Block data21     BlockData
    },
    Cyclic_FCS      CFCS
}

Cyclic_header ::= SEQUENCE {
    Cyclic_pattern    CYCLIC_PATTERN,
    Total_Block       TOTAL_BLOCK,
    Src STaddress     ST_Address,
    Reserved1         Unsigned8
}

```

4.1.1.2.2 CTL-PDU

4.1.1.2.2.1 Control-PDU,

```

Control-PDU ::= SEQUENCE {
    IP_header      IP_HEADER,
    UDP_header     UDP_HEADER,
    CTL_data       CTLdata
}

```

4.1.1.2.2.2 RMTCTL-PDU

```

CTL_cyclic-PDU ::= SEQUENCE {
    IP_header      IP_HEADER,
    UDP_header     UDP_HEADER,
    RMTCTL_CMD     SEQUENCE {
        Reserved1      Unsigned16,
        Header Type    HEADER_TYPE,
        Module ID      MODULE_ID,
        Station number  ST_NUM,
        Command code    RMTCTL_CMD,
        Length          LENGTH,
        RMTCTL Status   STATUS,
        RMTCTL Kind     RMTKIND,
        RMTCTL Length   RMT_LENGTH,
        RMTCTL Address  RMT_ADD,
        Send Number     SND_NUM,
        Reserved1       Unsigned16,
        Reserved2       Unsigned32
    },
    RMTCTL data     RMTCTL_DATA
}

```

4.1.1.2.3 INFO-PDU

```

INFO-PDU ::= SEQUENCE {
    IP_header      IP_HEADER,
    UDP_header     UDP_HEADER,
    INFO_data      INFOData
}

```

4.2 FAL PDU type N abstract syntax

4.2.1 Basic abstract syntax

The definitions of FALPDU are shown below.

```
FAL-PDU ::= CHOICE {
    CyclicData-PDU[0] CyclicData-PDU ,
    MulticastData-PDU [1] MulticastData-PDU ,
    PtoP Data-PDU      [2] PtoP Data-PDU ,
    Aliveinfo-PDU      [3] Aliveinfo-PDU ,
    Aliveinfo6-PDU     [4] Aliveinfo6-PDU ,
    Inq-PDU            [5] Inq-PDU ,
    Ninq-PDU           [6] Ninq-PDU ,
    Reply-PDU          [7] Reply-PDU ,
    RetransEnq-PDU     [8] RetransEnq-PDU ,
    RetransConfirm-PDU [9] RetransConfirm-PDU ,
    RetransNak-PDU     [10] RetransNak-PDU
}
```

FALAR-N Header to be used in each PDU is shown as follows.

```
FALAR-N Header ::= SEQUENCE {
    hd_h_type      Hd_h_type ,
    hd_ml          Hd_ml ,
    hd_sa          Hd_sa ,
    hd_da          Hd_da ,
    hd_v_seq       Hd_v_seq ,
    hd_seq         Hd_seq ,
    hd_m_ctl       Hd_m_ctl ,
    hd_inqid       Hd_inqid ,
    hd_tcd         Hd_tcd ,
    hd_ver         Hd_ver ,
    reserved1      OctString SIZE(3) ,
    hd_pkind       Hd_pkind ,
    hd_pseq        Hd_pseq ,
    hd_mode        Hd_mode ,
    hd_pver        Hd_pver ,
    hd_pri         Hd_pri ,
    hd_cbn         Hd_cbn ,
    hd_tbn         Hd_tbn ,
    hd_bsize       Hd_bsize ,
    reserved2      OctString SIZE(4)
}
```

```
Hd_inqid ::= SEQUENCE {
    inqid_inq_sa      Inqid_inq_sa ,
    inqid_tr_adr      Inqid_tr_adr ,
    inqid_id_seq      Inqid_id_seq
}
```

4.2.2 CyclicData-PDU

```
CyclicData-PDU ::= SEQUENCE {
    falArHeader      FALAR-N Header ,
    tmid             Tmid ,
    blockNumber       BlockNumber ,
    blockCount        BlockCount ,
    cyclicData        CyclicData
}
```

4.2.3 MulticastData-PDU

```

MulticastData-PDU ::= SEQUENCE {
    falArHeader      FALAR-N Header ,
    multicastData     MulticastData
}

```

4.2.4 PtoPData-PDU

```

PtoP Data-PDU ::= SEQUENCE {
    falArHeader      FALAR-N Header ,
    pointToPointData PointToPointData
}

```

4.2.5 Aliveinfo-PDU

```

Aliveinfo-PDU ::= SEQUENCE {
    falArHeader      FALAR-N Header ,
    alive4Header     ALIVE4 Header ,
    al_ExtentionList SEQUENCE of Al_ExtentionList ,
    al_ExtentionInfo Al_ExtenyionInfo
}

```

```

ALIVE4 Header ::= SEQUENCE {
    al_nd_name      Al_nd_name ,
    al_os_name      Al_os_name ,
    al_tm_out       Al_tm_out ,
    al_msgserno     Al_msgserno ,
    al_mode         Al_mode ,
    al_protocol     Al_protocol ,
    al_tg_cmn_cnt   Al_tg_cmn_cnt ,
    al_tg_cflag     Al_tg_cflag ,
    reserved1       OctString SIZE(1) ,
    al_tg_max       Al_tg_max ,
    al_tg_usecmt    Al_tg_usecmt ,
    al_chg_time     Al_chg_time ,
    al_ipv4addr1    Al_ipv4addr1 ,
    al_ipv4addr2    Al_ipv4addr2 ,
    al_ver          AL_ver ,
    reserved2       OctString SIZE(15)
}

```

```

Al_ExtentionList ::= SEQUENCE {
    al_ta_chgalvstat Al_ta_chgalvstat ,
    al_ta_chginfolstat Al_ta_chginfolstat ,
    al_ta_tid        Al_ta_tid ,
    al_ta_data       Al_ta_data
}

```

4.2.6 Aliveinfo6-PDU

```

Aliveinfo6-PDU ::= SEQUENCE {
    falArHeader      FALAR-N Header,
    alive6Header     ALIVE6 Header ,
    al_ExtentionList SEQUENCE of Al_ExtentionList ,
    al_ExtentionInfo Al_ExtentionInfo
}

```

```

ALIVE6 Header ::= SEQUENCE {
    al_nd_name      Al_nd_name ,
    al_os_name      Al_os_name ,
    al_tm_out       Al_tm_out ,
    al_msgserno     Al_msgserno ,

```

al_mode	Al_mode ,
al_protocol	Al_protocol ,
al_tg_cmn_cnt	Al_tg_cmn_cnt ,
al_tg_cflag	Al_tg_cflag ,
reserved1	OctString SIZE(1) ,
al_tg_max	Al_tg_max ,
al_tg_usecnt	Al_tg_usecnt ,
al_chg_time	Al_chg_time ,
reserved2	Al_ipv4addr1 ,
reserved3	Al_ipv4addr2 ,
al_ver	AL_ver ,
reserved4	OctString SIZE(15) ,
al_ipv6addr1	Al_ipv6addr1 ,
al_ipv6addr2	Al_ipv6addr2
}	

4.2.7 Inq-PDU

Inq-PDU ::= SEQUENCE {	
falArHeader	FALAR-N Header ,
inquirydata	InquiryData
}	

4.2.8 Ninq-PDU

Ninq-PDU ::= SEQUENCE {	
falArHeader	FALAR-N Header ,
node-list	Node-List ,
ninquirydata	NinquiryData
}	

4.2.9 Reply-PDU

Reply-PDU ::= SEQUENCE {	
falArHeader	FALAR-N Header
}	

4.2.10 RetransEnq-PDU

RetransEnq-PDU ::= SEQUENCE {	
falArHeader	FALAR-N Header ,
retransRequest	RetransRequest ,
retransRequestNode	RetransRequestNode ,
retransNumberOfRequests	RetransNumberOfRequests ,
reserved1	OctString SIZE(4) ,
retransInfo-List	SEQUENCE of RetransInfoList
}	

RetransInfoList ::= SEQUENCE {	
retransMcg	RetransMcg ,
retransPseqNo	RetransPseqNo
}	

4.2.11 RetransConfirm-PDU

RetransConfirm ::= SEQUENCE {	
falArHeader	FALAR-N Header ,
retransRequest	RetransRequest ,
retransNumberOfRequests	RetransNumberOfRequests ,
retransInfo-List	SEQUENCE of RetransInfoList
}	

4.2.12 RetransNak-PDU

```
RetransNAK-PDU ::= SEQUENCE {
    f alArHeader          FALAR-N Header ,
    retransRequest         RetransRequest ,
    retransNumberOfRequests RetransNumberOfRequests ,
    retransInfo-List      SEQUENCE of RetransInfoList
}
```

4.3 Data type assignments for type S

Data types used in FALPDU type S abstract syntax are shown as follows.

```
BLK_Status ::= Unsigned32
BlockData  ::= OCTET STRING(SIZE=64)
BlockNum   ::= Unsigned16
CLASS      ::= Unsigned16
CLTData    ::= OCTET STRING(SIZE = 18..1472)
CYCLIC_FCS ::= Unsigned32
EntryNum   ::= Unsigned32
ERR_ADD    ::= Unsigned32
HEADER_TYPE ::= Unsigned16
INFOData   ::= OCTET STRING(SIZE = 18..1472)
IP_Header  ::= OCTET STRING(SIZE=20)
MODULE_ID  ::= Unsigned16
Node_Status ::= Unsigned32
Pattern16  ::= Unsigned16
Pattern32  ::= Unsigned32
Pattern8   ::= Unsigned8
PLD        ::= Unsigned32
PortNum    ::= Unsigned32
Priority    ::= Unsigned8
RCL_Status ::= Unsigned32
RCLCMD     ::= Unsigned32
RCLKIND    ::= Unsigned32
RMT_ADD    ::= Unsigned32
RMT_LENGTH ::= Unsigned16
RMTCTL_CMD ::= Unsigned16
RMTCTL_DATA ::= OCTET STRING(SIZE = 0..1440)
RMTKIND    ::= Unsigned16
SendTime   ::= OCTET STRING(SIZE=8)
SequenceNum32 ::= Unsigned32
SND_NUM    ::= Unsigned16
ST_Address ::= Unsigned8
ST_NUM     ::= Unsigned16
STATUS     ::= Unsigned32
TOTAL_BLOCK ::= Unsigned8
UDP_Header ::= OCTET STRING(SIZE=8)
```

4.4 Data type assignments for type N

Data types used in FALPDU type N abstract syntax are shown as follows.

```
Al_ExtentionInfo ::= OctString SIZE(0..664)
Al_chg_time      ::= Unsigned32
Al_ipv4addr1     ::= OctString SIZE(4)
Al_ipv4addr2     ::= OctString SIZE(4)
Al_ipv6addr1     ::= OctString SIZE(16)
Al_ipv6addr2     ::= OctString SIZE(16)
Al_mode          ::= Unsigned8
Al_msgsernot     ::= Unsigned16
Al_nd_name       ::= OctString SIZE(10)
```

```

Al_os_name::= OctString SIZE(10)
Al_protocol::= Unsigned8
Al_ta_chgalvstat::= Unsigned8
Al_ta_chginfostat::= Unsigned8
Al_ta_data::= interger16
Al_ta_tid::= Unsigned16
Al_tg_cflag::= Unsigned8
Al_tg_cmn_cnt::= Unsigned16
Al_tg_max::= Unsigned16
Al_tg_usecnt::= Unsigned16
Al_tm_out::= Unsigned32
AL_ver::= Unsigned8
BlockCount::= Unsigned16
BlockNumber::= Unsigned16
CyclicData::= Octstring SIZE(64..16640)
Data type assignment
Hd_bsize::= unsigned16
Hd_cbn::= unsigned8
Hd_da::= unsigned32
Hd_h_type::= unsigned32
Hd_m_ctl::= unsigned32
Hd_ml::= unsigned32
Hd_mode::= integer16
Hd_pkind::= unsigned8
Hd_pri::= unsigned8
Hd_pseq::= unsigned32
Hd_pver::= unsigned8
Hd_sa::= unsigned32
Hd_seq::= unsigned32
Hd_tbn::= unsigned8
Hd_tcd::= unsigned16
Hd_v_seq::= unsigned32
Hd_ver::= unsigned16
Inqid_id_seq::= unsigned32
Inqid_inq_sa::= unsigned32
Inqid_tr_adr::= unsigned32
InquiryData::= OctString SIZE(0..262144)
MulticastData::= OctString SIZE(0..262144)
Node-List::= OctString SIZE(512)
PointToPointData::= OctString SIZE(0..262144)
RetransMcg::= unsigned32
RetransNumberOfRequests::= unsigned32
RetransPseqNo::= unsigned32
RetransRequest::= unsigned32
RetransRequestNode::= unsigned32
Tmid::= Unsigned32

```

5 FAL transfer syntax

5.1 Encoding rules

5.1.1 Unsigned encoding

The fixed-length, unsigned values Unsigned8, Unsigned16, Unsigned24, and Unsigned32 are encoded as unsigned integers of one octet, two octets, three octets, and four octets in length, respectively. Unsigned16, Unsigned24, and Unsigned32 are encoded as big endians, where the most significant octet is regarded as the first octet, the next octet is regarded as the second octet, and the least significant octet is regarded as the last octet.

5.1.2 Octet string encoding

OctetString which has variable number of octets is encoded octet by octet, in sequential order.

5.1.3 SEQUENCE encoding

SEQUENCE encoding is performed in sequence, starting from the initial element. The identifiers and length used in ASN.1 are not used.

5.2 FALPDU type S elements encoding

5.2.1 RCL_header

5.2.1.1 Frame Class

This field shows the PDU classes described in Table 4.

Table 4 – Frame Class

Value	Description
0x0001	Class 1 frame (RHE-PDU)
0x0002	Class 2 frame (LCC-PDU, LCA-PDU, LCN-PDU, LNA-PDU, SCR-PDU)

5.2.1.2 Dst Address

5.2.1.2.1 Reserved1

This field is reserved for future use. This value is 0x00.

5.2.1.2.2 DA_STaddress

This field indicates the destination station address described in Table 5.

Table 5 – DA_STaddress – DA_STaddress

Value	Description
0xFF	RHE-PDU, LCC-PDU, LCN-PDU, SCR-PDU
SA_STaddress in the received LCC-PDU	LCA-PDU
SA_STaddress in the received LCN-PDU	LNA-PDU

5.2.1.2.3 DA_MACaddress

This field indicates the destination MAC address described in Table 6.

Table 6 – DA_MACaddress

Value	Description
FF-FF-FF-FF-FF-FF	RHE-PDU, LCC-PDU, LCN-PDU, SCR-PDU
SA_MACaddress in the received LCC-PDU	LCA-PDU
SA_MACaddress in the received LCN-PDU	LNA-PDU

5.2.1.3 Src Address

5.2.1.3.1 Reserved2

This field is reserved for future use. This value is 0x00.

5.2.1.3.2 SA_STaddress

This field indicates the station address of the own node.

5.2.1.3.3 SA_MACaddress

This field indicates the MAC address of the own node.

5.2.1.4 RCL Command

Table 7 shows RCL Command (CMD) field format. This field is used to identify the RCL frame. Each bit in this field has the following meanings:

Table 7 – CMD field format

Bit	Value						Description
	RHE	LCC	LCA	LCN	LNA	SCR	
[3:0]	1						reserved field1
[7:4]	0	0	1	2	3	0	unique number 1
[11:8]	0	1	1	1	1	3	unique number 2
[15:12]	0						reserved field2
[23:16]	1	2	2	2	2	2	indicates the frame class
[31:24]	0						reserved field3

5.2.1.5 Frame Sequence

The frame sequence number is used to detect loss and duplication of received frames. The range of this value is from 0x00000000 to 0xFFFFFFFF with step size 1.

5.2.1.6 Reserved3

This field is reserved for future use. The value is 0.

5.2.2 RHE-PDU

5.2.2.1 RHE_Common

5.2.2.1.1 Send Direction

This field contains the transmission port of the RHE-PDU as described in Table 8.

Table 8 – Send Direction

Value	Description
0x01	Port A
0x02	Port B

5.2.2.1.2 RHE Sequence

This field contains the identification number of the RHE-PDU. The range of this value is from 0x0000 to 0xFFFF. The same value is used in port A and port B.

5.2.2.2 Neighborhood address

5.2.2.2.1 Reserved1

This field is reserved for future use. The value is 0x00.

5.2.2.2.2 NB_STaddress

This field indicates the station address of the neighborhood node.

5.2.2.2.3 NB_MACaddress

This field indicates the MAC address of the neighborhood node.

5.2.2.3 RHE RecieveStatus

This field indicates the status about receiving RHE-PDU as described in Table 9.

Table 9 – RHE ReceiveStatus

Value	Description
1	Receiving RHE-PDU
0	Not receiving RHE-PDU

5.2.2.4 True Status

5.2.2.4.1 Reserved2

This field is reserved for future use. The value is 0x00000000.

5.2.2.4.2 Physical Linkdown

This field indicates the link status at the port as described in Table 10.

Table 10 – Physical Linkdown

Value	Description
1	Physical Link up
0	Physical Link down

5.2.2.4.3 Reserved3

This field is reserved for future use. The value is 0x00000000.

5.2.2.5 Blocking Status

This field indicates the status code about the port.

5.2.2.6 Node Status

This field indicates the status code about the node.

5.2.2.7 Reserved4

This field is reserved for future use. This value is 0.

5.2.2.8 Reserved5

This field is reserved for future use. This value is 0.

5.2.2.9 RHE_pattern1~4

This field is used to check the pattern of RHE-frames as described in Table 11. The patterns are repeated until the maximum length of ISO/IEC 8802-3 standard's frame.

Table 11 – RHE_pattern 1~4

Pattern	Octets	Value
RHE_pattern1	4	0xFFFFFFFF
RHE_pattern2	4	0xAAAAAAAA
RHE_pattern3	4	0x55555555
RHE_pattern4	4	0x00000000

5.2.3 LCC-PDU

5.2.3.1 LCC Kind

This field indicates the reason of sending LCC-PDU as described in Table 12.

Table 12 – LCC-Kind

Value	Description
0x00000101	Logical link up port A of the node
0x00000201	Competitive of receiving other LCC-PDU

5.2.3.2 LCC Priority

5.2.3.2.1 LCC_Priority

This field indicates the priority of LCC-PDU. This value is 0x0C.

5.2.3.2.2 LCC_STadd

This field indicates the station address of the own node.

5.2.3.2.3 LCC_MACadd

This field indicates the MAC address of the own node.

5.2.3.3 Reserved1

This field is reserved for future use. This value is 0.

5.2.4 LCA-PDU

5.2.4.1 Reserved1

This field is reserved for future use. This value is 0x01.

5.2.4.2 RCL Status

This field indicates the status of the node sending LCA-PDU as described in Table 13.

Table 13 – RCL Status

Field name	Bit	Value	Description
-	31 .. 24	0x00	Reserved
Node status	23	0 or 1	1: The node is in "Edge-A" state. When the bit 23 and 22 are "1", the node is in "Isolated" state. 0: The node is in other node status (Edge-B or Intermediate).
	22	0 or 1	1: The node is in "Edge-B" state. When the bit 23 and 22 are "1", the node is in "Isolated" state. 0: The node is in other node status (Edge-A or Intermediate).
-	21	0	Reserved
Blocking status	20	0 or 1	1: Port A or B is logical link down port.
-	19 .. 8	0x000	Reserved
Port status	7	0 or 1	1: Port A is in "Port_Linkup" state.
	6	0 or 1	1: Port B is in "Port_Linkup" state.
	5	0 or 1	1: Port A is physical link up.
	4	0 or 1	1: Port B is physical link up.
-	3 .. 0	0x0	Reserved

5.2.4.3 LCA Priority

5.2.4.3.1 LCA_Priority

This field indicates the priority in the received LCC-PDU.

5.2.4.3.2 LCA_STadd

This field indicates the station address in the received LCC-PDU.

5.2.4.3.3 LCA_MACadd

This field indicates the MAC address in the received LCC-PDU.

5.2.4.4 Reserved2

This field is reserved for future use. This value is 0.

5.2.5 LCN-PDU

5.2.5.1 Reserved1

This field is reserved for future use. This value is 0x01.

5.2.5.2 RCL Status

This field indicates the status of the node sending LCN-PDU. It has the same format as "5.2.4.2 RCL Status" described in Table 13.

5.2.5.3 Reserved2

This field is reserved for future use. This value is 0.

5.2.6 LNA-PDU

5.2.6.1 Reserved1

This field is padding area. This value is 0x00.

5.2.6.2 RCL Status

This field indicates the status of the node sending LNA-PDU. It has the same format as “5.2.4.2 RCL Status” described in Table 13.

5.2.6.3 Reserved2

This field is reserved for future use. This value is 0.

5.2.7 SCR-PDU

5.2.7.1 Entry Number

This field indicates the number to contain the SCR_Event entry.

5.2.7.2 SendStat Time

This field contains the sending time of seconds.

5.2.7.3 Reserved1

This field is reserved for future use. This value is 0.

5.2.7.4 Event0~16

This field contains the event information and specifies each manufacture.

5.2.8 Cyclic_S-PDU

5.2.8.1 IP_header

The encoding of the fields shall be according to RFC 791.

5.2.8.2 UDP_header

The encoding of the fields shall be according to RFC 768.

5.2.8.3 Reserved1

This field is reserved for future use. This value is 0x0000.

5.2.8.4 Block_number1~21

This field indicates the identification number of the cyclic data block and specifies each manufacture.

5.2.8.5 Block_num pattern

This field is used to check the Cyclic-PDU and specifies each manufacture.

5.2.8.6 Block data1~21

This field contains the cyclic data.

5.2.8.7 Cyclic_FCS

This field is an error checking code from Cyclic-PDU.

The generating polynomial is $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1$.

5.2.9 Cyclic_header

5.2.9.1 Cyclic_pattern

This field is used to check the Cyclic_header and specifies each manufacture.

5.2.9.2 Total_Block

This field indicates the number of blocks in the Cyclic-PDU.

5.2.9.3 Src STaddress

This field indicates the MAC address of the own node.

5.2.9.4 Reserved1

This field is reserved for future use. This value is 0x00.

5.2.10 Control-PDU

5.2.10.1 IP_header

Refer to 5.2.8.1.

5.2.10.2 UDP_header

Refer to 5.2.8.2.

5.2.10.3 CTL_data

This field contains the data of the CTL-PDU.

5.2.11 RMTCTL-PDU

5.2.11.1 IP_header

Refer to 5.2.8.1.

5.2.11.2 UDP_header

Refer to 5.2.8.2.

5.2.11.3 RMTCTL_CMD

5.2.11.3.1 Reserved1

This field is reserved for future use. This value is 0x0000.

5.2.11.3.2 Header Type

This field is used to recognize the remote control frame. This value is “RC” in ASCII code (“RC” = 0x5243).

5.2.11.3.3 Module ID

This field indicates the module type.

5.2.11.3.4 Station number

This field indicates the station address of the destination node.

5.2.11.3.5 Command code

This field specifies the command type.

5.2.11.3.6 Length

This field indicates the length of RMTCTL-PDU.

5.2.11.3.7 RMTCTL Status

This field indicates the status code about the remote control operations.

5.2.11.3.8 RMTCTL Kind

This field indicates the module type for remote memory reading.

5.2.11.3.9 RMTCTL Length

This field indicates the data size of remote memory writing or reading.

5.2.11.3.10 RMTCTL Address

This field indicate the address of remote memory writing or reading.

5.2.11.3.11 Send Number

This field is use to manage of the remote control by the FAL-user.

5.2.11.3.12 Reserved2

This field is reserved for future use. This value is 0x0000.

5.2.11.3.13 Reserved3

This field is reserved for future use. This value is 0x00000000.

5.2.11.3.14 RMTCTL data

This field contains the data of the remote control command.

5.2.12 INFO-PDU

5.2.12.1 IP_header

Refer to 5.2.8.1.

5.2.12.2 UDP_header

Refer to 5.2.8.2.

5.2.12.3 INFO_data

This field contains the data of the INFO-PDU.

5.3 FALPDU type N elements encoding

5.3.1 General

FALPDU type N headers excluding user data are specified using the TCP/IP network octet ordering convention, which is known as big endian.

5.3.2 FALAR-N Header

5.3.2.1 hd_h_type

This field shall indicate the header type specification described in

Table 14. Specify "NUXM" or "NUV6" in ASCII. PDUs transmitted in IPv4 Data field specify "NUXM". PDUs transmitted in IPv6 Data field specify "NUV6".

Table 14 – hd_h_type

Value	PDU type
"NUXM"	PDUs transmitted in IPv4 Data field
"NUV6"	PDUs transmitted in IPv6 Data field

5.3.2.2 hd_ml

This field shall indicate the length of the total message including FALAR-N header length (= 64 octets) and message length. ($hd_ml = 64 + \text{message length}$).

5.3.2.3 hd_sa

This field shall contain the address of the source node (Logical Node Number) described in Figure 2.

0	7	8	15	16	31 bit
Dmn		Dfn		Lnn	

Dmn: DoMain Number (=0)

Dfn: Data Field Number

Lnn: Logical Node Number

Figure 2 – hd_sa

Dmn:

A domain is defined as the concept above the data field. A domain consists of multiple data fields. Usually, one data field is defined as a local location and connected to other domains via wide area communication networks. Within a system, a number that uniquely identifies a domain is called a Domain Number (Dmn). Unique Dmn in the range of 1 to 64 is assigned to each of domains in the system. Either a domain number 0 (Dmn 0) or non-specified domain number indicate a local domain (default).

Dfn:

Data Field (DF) is a place where ADP messages consisting of information of specific characteristics can flow. Communication using ADP is available between nodes belonging to one data field. This means that any equipment in a system shall belong to one or more data fields. One data field is configured per network address or sub-network address of an IP address. Within a decentralized system, a number that uniquely identifies a data field is called a Data Field Number (Dfn). Unique Dfn is assigned to each of data fields within the system, in the range of 1 to 255. Dfn 0 is reserved for communication within the current node. An IPv4 network shall be assigned with IPv4 Dfn. An IPv6 network shall be assigned with IPv6 Dfn.

NOTE 1 No data field can be set up across multiple LAN segments. Every LAN segment is assigned with Dfn.

NOTE 2 One LAN segment can contain more than one data field.

Lnn:

A logical node is equipment in a data field. Within a data field, a number that uniquely identifies a logical node is called a Logical Node Number (Lnn). Unique Lnn in the range of 1 to 4095 is assigned to each of devices. Lnn 0 is reserved for communication within the current node.

5.3.2.4 hd_da

This field shall contain the destination multicast group address (Multicast Group Number) or the destination node address (Logical Node Number) described in Figure 3.

(Multicast communication)

0	7	8	15	16	31 bit
Dmn		Dfn		Mgn	

(Peer to Peer communication)

0	7	8	15	16	31 bit
Dmn		Dfn		Lnn	

Dmn: DoMain Number (=0)
Dfn: Data Field Number
Mgn: Multicast Group Number
Lnn: Logical Node Number

Figure 3 – hd_da

Dmn,

Refer to 5.3.2.3.

Dfn,

Refer to 5.3.2.3.

Mgn,

A multicast group (MCG) is a group of nodes belonging to a data field to determine whether to receive data multicast/ broadcast from a node. A multicast group shall be specified to perform multicast transmission. Any node shall belong to a multicast group to be used for multicast reception. One data field can contain more than one multicast group. Each node can be added to more than one multicast group as far as the groups are within a data field to which the node belongs. Multicast Group Number (Mgn) is a number that uniquely identifies a

multicast group within a data field. Unique Mgn in the range of 1 to 255 is assigned to each of multicast groups in the data field. Mgn 0 is reserved for the system to send/receive alive-messages.

A Mgn of MulticastData-PDUs shall be different from a Mgn of RetransX PDUs.

Here, RetransX PDU indicates any one of RetransEnq-PDU, RetransCinfirm-PDU or RetransNak-PDU.

A Mgn of Multicast communication with retransmission is also different from a Mgn of both Multicast communication (with no retransmission) and RetransX-PDU.

Lnn,

Refer to 5.3.2.3.

Table 15 shows either Mgn or Lnn used by each PDU.

Table 15 – Usage of Mgn or Lnn

Field(Mgn , Lnn)		Description(PDU)
Mgn	Lnn	
Applicable	N/A	CyclicData-PDU
Applicable	N/A	MulticastData-PDU
N/A	Applicable	PtoPData-PDU
Applicable (=0)	N/A	Aliveinfo-PDU
Applicable (=0)	N/A	Aliveinfo6-PDU
Applicable	Applicable	Inq-PDU
Applicable	N/A	Ninq-PDU
N/A	Applicable	Reply-PDU
Applicable	N/A	RetransEnq-PDU
Applicable	N/A	RetransConfirm-PDU
Applicable	N/A	RetransNak-PDU

5.3.2.5 hd_v_seq

This field shall indicate the version identification of the sequence number of the transmission message (hd_seq). This field normally specifies the time when the sequence number of the transmission message (hd_seq) is initialized. The objective of this field enables a receiving node to detect that a sending node has newly initialized the sequence number. New setting value of hd_v_seq shall be different from old value of it.

5.3.2.6 hd_seq

5.3.2.6.1 hd_seq

This field shall indicate the sequence number of the transmission message. The range of values is from 0x00000001 to 0x7FFFFFFF, and used cyclically.

5.3.2.6.2 Transmission sequence number and version number management

5.3.2.6.2.1 Management at a transmitting node

Assignment:

The version numbers (V_SEQ) shall be assigned to each node. The version numbers are assigned with time stamps. Non-zero value of the time stamp shall be different from the preceding V_SEQ value.

a) Multicast communication

Each node has transmission sequence numbers (S_SEQ). The transmission sequence numbers (S_SEQ) shall be managed by each Dfn (Data field number), by each Mgn (Multicast group number) and by each message priority level.

b) Peer to Peer communication

Each node has transmission sequence numbers (S_SEQ). The transmission sequence numbers (S_SEQ) shall be managed by each Dfn, by the connection for each node and by each message priority level.

Initialization:

a) Multicast communication

On opening a multicast communication port, V_SEQ corresponding to the port shall be set to the opening time and S_SEQ corresponding to the port shall be set to 0.

b) Peer to Peer communication

On establishing the TCP connection, V_SEQ corresponding to the connection shall be set to the establishing time and S_SEQ corresponding to the connection shall be set to 0.

Transmission:

On transmitting of a message, V_SEQ and updated S_SEQ are set to hd_v_seq and hd_seq in the FALAR_N Header, respectively. S_SEQ will be updated as follows:

- If S_SEQ is not 0x7FFFFFFF, S_SEQ is incremented by one.
- If S_SEQ is 0x7FFFFFFF, S_SEQ is reset to 1 without increment.

5.3.2.6.2.2 Management at a receiving node

Assignment:

a) Multicast communication

Each node has the reception sequence numbers (R_SEQ) lastly received and a reception version number (R_V_SEQ) lastly received. The R_SEQ shall be managed by each Dfn (Data Field Number), by each Mgn (Multicast Group Number), by each source node, and by each message priority level.

b) Peer to Peer communication

Each node has the reception sequence numbers (R_SEQ) lastly received and reception version number (R_V_SEQ) lastly received. The R_SEQ shall be managed by each Dfn (Data field number), by the connection for each node and by each message priority level.

Initialization:

a) Multicast communication

On opening a multicast communication port, R_V_SEQ and R_SEQ shall be initialized to 0.

b) Peer to Peer communication

On establishing the TCP connection, R_V_SEQ corresponding to the connection and R_SEQ corresponding to the connection shall be initialized to 0.

Reception:

On reception of a message, R_V_SEQ and R_SEQ are retrieved by hd_sa, by hd_seq, by hd_pri and by hd_v_seq in the FALAR_N Header.

NOTE Any message with hd_v_seq = 0, hd_seq = 1, hd_cbn = 1 and hd_tbn = 1 is passed to an application without checking the sequence number.

5.3.2.6.2.3 Sequence number check for reception message

- 1) If R_V_SEQ is equal to 0,

the message is unconditionally received and hd_v_seq and hd_seq in the FALAR_N Header update R_V_SEQ and R_SEQ, respectively.
- 2) If R_V_SEQ is not 0 and hd_v_seq is equal to R_V_SEQ,

R_SEQ and hd_seq are compared for detecting a duplicated message or lost messages. Table 16 and Figure 4 show conditions of sequence number check for reception message.
- 3) Communication
 - a) Multicast communication

If R_V_SEQ is not 0 and hd_v_seq is not equal to R_V_SEQ, hd_v_seq and hd_seq values update the R_V_SEQ and R_SEQ values, respectively. On detecting a duplicated message, a receiving node discards the received message.
 - b) Peer to Peer communication

If R_V_SEQ is not 0 and hd_v_seq is not equal to R_V_SEQ, a receiving node disconnects the TCP connection and discards the received message.

Table 16 – Detailed conditions for sequence number check of reception message

No.	Conditions	Result
1	$R_SEQ \neq 0x7fffffff \ \&\& \ R_SEQ + 1 == SEQ$	Normal message reception
2	$R_SEQ == 0x7fffffff \ \&\& \ SEQ == 1$	Normal message reception
3	$R_SEQ > N1 \ \&\& \ R_SEQ - N1 < SEQ \leq R_SEQ$	Duplicated message reception
4	$R_SEQ \leq N1 \ \&\& \ (0 < SEQ \leq R_SEQ \ \ 0x7fffffff - (N1 - R_SEQ) < SEQ \leq 0x7fffffff)$	Duplicated message reception
5	Other conditions	Missing message
Key: SEQ: Value of hd_seq in the FALAR_N Header N1: Transmission sequence number assumed to be duplicated. This value is implemented matter.		

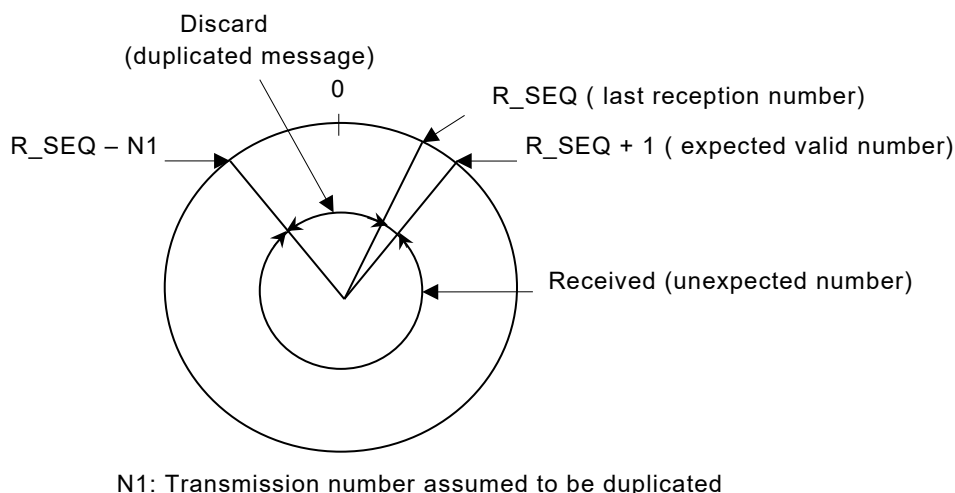


Figure 4 – Valid sequence number for reception message

5.3.2.7 hd_m_ctl

This field shall specify information on message transmission control described in Figure 5.

0	1	2	3	4	5	6	31 bit
MLT	ONE	INQ	RPL	NIQ	MCR	Rsv	

Key:

MLT: Multicast communication (= 0x80000000)

ONE: Peer to Peer (One to One) communication (= 0x40000000)

INQ: Inquiry communication (= 0x20000000)

RPL: Reply response communication (= 0x10000000)

NIQ: N-inquiry communication (= 0x08000000)

MCR: Multicast communication with re-transmission control (= 0x04000000)

Rsv: Reserved

Figure 5 – hd_m_ctl

Table 17 shows the valid bits of hd_m_ctl field in each PDU.

Table 17 – Valid bits of hd_m_ctl

PDU	Valid bits of hd_m_ctl
CyclicData-PDU	0x80000000
MulticastData-PDU	0x80000000 (Multicast communication type)
	0x84000000 (Multicast communication with re-transmission type)
PtoPData-PDU	0x40000000
Aliveinfo-PDU	0x80000000
Aliveinfo6-PDU	0x80000000
Inq-PDU	0xA0000000 (Multicast communication type)
	0x60000000 (Peer to Peer communication type)

PDU	Valid bits of hd_m_ctl
Ninq-PDU	0x88000000
Reply-PDU	0x50000000
RetransEnq-PDU	0x84000000
RetransConfirm-PDU	0x84000000
RetransNak-PDU	0x84000000

5.3.2.8 hd_inqid

This field shall specify inquiry/response identifier, which consists of inqid_inq_sa, inqid_tr_adr and inqid_id_seq. These three fields are described in 5.3.2.20, 5.3.2.21 and 5.3.2.22 respectively. This hd_inqid field is valid in Inq-PDU, Ninq-PDU and Reply-PDU.

5.3.2.9 hd_tcd

This field shall specify transaction code. All transactions are assigned transaction codes (TCDs) to identify them. A TCD represents transaction data. Each application program transmits data with a specified TCD into a data field. On the other hand, each application program receives data with a specifying TCD from a data field. Unique TCDs shall be defined for each data field. TCDs are classified into those for user transactions and those for system transactions, as shown below.

- TCDs for user transactions: 1 to 59 999 (user-definable)
- TCDs for system transactions: 60 000 to 65 534 (reserved for the system)

A system transaction is created when the system detects an event (such as an error or internal status change) and notifies the user program. Table 18 shows specified TCD.

Table 18 – Specified TCD

Specified TCD	Description
60 003	Alive report
60 061	Retransmission control
60 027	File transfer
60 056	Cyclic memory transfer
60 058	Cyclic memory transfer (Fast type)

5.3.2.10 hd_ver

This field shall specify a program version identifier. A current value is 0.

5.3.2.11 hd_pkind

This field shall specify a transaction identifier as shown in Table 19. This field is valid for a one to one communication (PtoPData-PDU) using a duplex LAN.

Table 19 – hd_pkind

Value	Description
1	ACK Request (ACK.req); Request the destination node (Dfn, Lnn) to inform a reception packet sequence number (R-PSEQ)
2	ACK Response (ACK.inform); Inform the reception packet sequence number (R-PSEQ) to ACK Request node.
0	Others (NoACK)

5.3.2.12 hd_pseq

5.3.2.12.1 hd_pseq

This field shall specify a packet sequence number. The range of values is from 0x1 to 0xFFFFFFFF, and used cyclically. This field is effective for "Multicast communication with retransmission" and for "One to one communication using a duplex LAN" as shown in Table 20.

Table 20 – PDU with an effective hd_pseq

PDU	Effectiveness	Description	Value of hd_pseq
MulticastData-PDU	Effective	Multicast communication with retransmission	0x1 – 0x FFFFFFFF
PtoPData-PDU	Effective	One to one communication (PtoPData-PDU) using a duplex LAN	0x1 – 0x FFFFFFFF
Other PDU	Ineffective	Others (including one to one communication (PtoPData-PDU) using a single LAN)	0 (recommend)

5.3.2.12.2 Management at a transmitting node

Assignment:

Each node has management information (S_PSEQ) of a transmitting packet sequence number.

a) Multicast communication with retransmission

A S_PSEQ shall be assigned by Dfn (Data field number) and by Mgn (Multicast group number).

b) One to one communication using a duplex LAN

A S_PSEQ shall be assigned by Dfn (Data field number) and by the TCP connection for each node.

Initialization:

a) Multicast communication with retransmission

On opening a multicast communication port, the S_PSEQ corresponding to the port shall be set to 0.

b) One to one communication using a duplex LAN

On establishing the TCP connection, the S_PSEQ corresponding to the connection shall be set to 0.

Transmission:

On transmission of a packet, the updated S_PSEQ is set to hd_pseq in the FALAR_N Header.

S_PSEQ will be updated as follows.

- If S_PSEQ is not 0xFFFFFFFF, S_PSEQ is incremented by one.
- If S_PSEQ is 0xFFFFFFFF, S_PSEQ is reset to 1 without increment.

NOTE In "Multicast communication with retransmission ", these three PDU (RetransEnq-PDU, RetransConfirm-PDU, RetransNak-PDU) don't use the hd_pseq field.

5.3.2.12.2.1 Management at a receiving node**Assignment:**

Each node has management information (R_PSEQ) of a received packet sequence number.

a) Multicast communication with retransmission

A R_PSEQ shall be managed by Dfn (Data field number), by Mgn (Multicast group number) and by source node.

b) One to one communication using a duplex LAN

A R_PSEQ shall be assigned by Dfn (Data field number) and by the TCP connection for each node.

Initialization:

a) Multicast communication with retransmission

- On opening a multicast communication port, the S_PSEQ corresponding to the port shall be set to 0.
- On the failure of retransmission request, the R_PSEQ corresponding to the port shall be set to 0.

b) One to one communication using a duplex LAN

On establishing the TCP connection, the R_PSEQ corresponding to the connection shall be set to 0.

Reception:

On reception of a packet, the R_PSEQ is retrieved by hd_pseq in the FALAR_N Header.

5.3.2.12.2.2 Packet sequence number check

a) Multicast communication with retransmission

On reception of a packet, comparison R_PSEQ with hd_pseq in the FALAR_N Header is executed.

R_PSEQ will be updated as follows.

- 1) If R_PSEQ is equal to 0,
the packet is unconditionally received and hd_pseq in the FALAR_N Header set to R_PSEQ.
- 2) If R_PSEQ is not equal to 0,
R_PSEQ and hd_pseq are compared for detecting a duplicated packet or lost packets.
Table 21 and Figure 6 show conditions for packet sequence number check.

**Table 21 – Detailed conditions for sequence number check of reception packet
(Multicast communication with retransmission)**

No.	Conditions	Result
1	$R_PSEQ \neq 0xffffffff \ \&\& \ R_PSEQ + 1 == PSEQ$	Normal packet reception ($R_PSEQ = PSEQ$)
2	$R_PSEQ == 0xffffffff \ \&\& \ PSEQ == 1$	Normal packet reception ($R_PSEQ = PSEQ$)
3	$R_PSEQ > N1 \ \&\& \ R_PSEQ - N1 < PSEQ \leq R_PSEQ$	Duplicated packet reception
4	$R_PSEQ \leq N1 \ \&\& \ (0 < PSEQ \leq R_PSEQ \ \ 0xffffffff - (N1 - R_PSEQ) < PSEQ \leq 0xffffffff)$	Duplicated packet reception
5	Other conditions	Lost packets

PSEQ: Value of hd_pseq in the FALAR_N Header.
N1: Transmission packet sequence number assumed to be duplicated. This value is implemented matter.

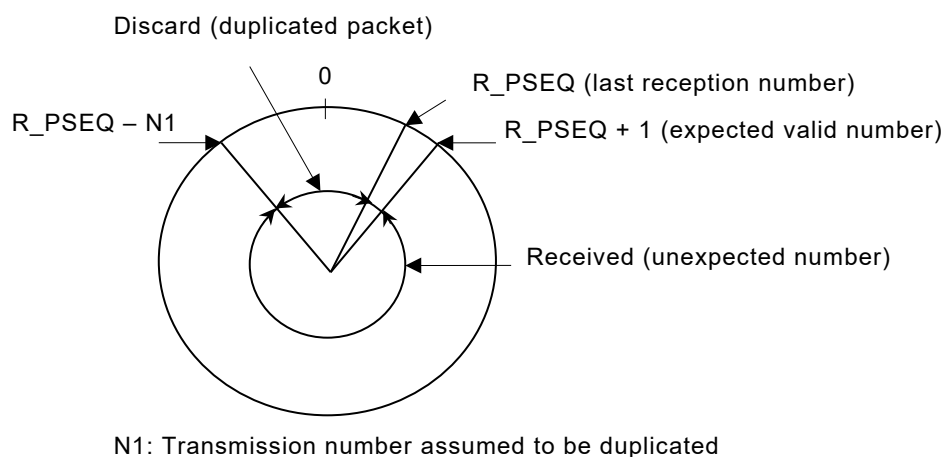


Figure 6 – Valid reception packet sequence number

- b) One to one communication using a duplex LAN
- 1) If R_PSEQ is equal to 0 and hd_pseq in the FALAR_N Header is equal to 1, set the value of hd_pseq to R_PSEQ (first normal packet received).
 - 2) If R_PSEQ is equal to 0 and hd_pseq in the FALAR_N Header is not equal to 1, the receiving node discards the received packet and disconnects the TCP connection.
 - 3) If R_PSEQ is not equal to 0, R_PSEQ and hd_pseq are compared for detecting a duplicated packet or lost packets. Table 22 shows conditions for packet sequence number check.

Table 22 – Detailed conditions for packet sequence number check

No.	Conditions	Result
1	$R_PSEQ \neq 0xffffffff \ \&\& \ R_PSEQ + 1 == PSEQ$	Normal packet reception ($R_PSEQ = PSEQ$)
2	$R_PSEQ == 0xffffffff \ \&\& \ PSEQ == 1$	Normal packet reception ($R_PSEQ = PSEQ$)
3	Other conditions	Duplicated packet reception or lost packets
One to one communication using a duplex LAN		

On detecting a duplicated packet or lost packets, the receiving node discards the received packet.

5.3.2.13 hd_mode

5.3.2.13.1 Overview

This protocol allows adding a node mode to a message to indicate whether the node is online or under testing. Since a node can control transmitting/receiving messages selectively based on the discriminating mode (online / testing), the node can eliminate any interrupting test message when it is on the way of testing.

When a node is performing a simulation test, it can receive either online message or test message.

5.3.2.13.2 Message modes

A message mode is assigned by every message. There are two types of message modes:

- 1) Online mode,
- 2) Test mode.

5.3.2.13.3 Node modes

A node mode is assigned by every Dfn (Data field number). There are two types of node modes:

- 1) Online mode,
- 2) Test mode.

A node with online mode shall set online message mode to hd_mode field.

A node with test mode shall set test message mode to hd_mode field.

5.3.2.13.4 Transmitting/Receiving message control

Table 23 shows the relations between message modes and node modes. Message control using these relations is executed.

Table 23 – Relation between message transmission/reception

Message mode	Node mode			
	Online		Test	
	Transmission	Reception	Transmission	Reception
Online	Effective	Effective	Not effective	Effective
Test	Not effective	Discard	Effective	Effective
Effective: transmitting / receiving a message; Not effective: no transmitting a message				

5.3.2.13.5 hd_mode

This field shall specify a message mode as shown in Table 24.

Table 24 – hd_mode

Value	Description
0	Online mode
1	Test mode

5.3.2.14 hd_pver

This field shall specify a protocol version. A current value is 1.

5.3.2.15 hd_pri

This field shall specify a message priority level as shown in Table 25.

Table 25 – Message priority level

Priority level ^a	Description
0	Out of priority control (No priority control is implemented)
1	Message with the highest priority level 1
2	Message with the priority level 2
3	Message with the priority level 3
4	Message with the priority level 4
5	Message with the priority level 5
6	Message with the priority level 6
7	Message with the lowest priority level 7
^a Higher numbers means lower priority.	

5.3.2.16 hd_cbn

5.3.2.16.1 Overview for fragmenting and assembling messages

The specifications limit the maximum user message size to 16 KB for the TCP/UDP based communication, while the maximum transfer size (MTU) for LANs is limited to 1 500 octets under the Ethernet standard.

For the TCP based communication, the IPv4 and TCP headers use 20 octets respectively and the FALAR-N header uses 64 octets, so available free data storage size can be calculated as (MTU – 104) octets.

On the other hand, the IPv6 header uses 40 octets and the FALAR-N header uses 64 octets, so available free data storage size can be calculated as (MTU – 124) octets.

For the UDP based communication, available free data storage size can be calculated as (MTU – 92) octets or as (MTU-112) octets, respectively.

NOTE The maximum free user's area in the TCP segment can be less than 1 460 octets or less than 1 440 octets through a negotiation on the TCP segment size when establishing a TCP connection.

If a message size is more than α octets, the message shall be divided into multiple PDUs with added FALAR-N headers respectively. Table 26 shows the value of α . The PDUs are sent to the data field in order. The receiver node receives the PDUs and assembles them into one original message.

Table 26 – Value of α

Communication type	Value of α
IPv4 + TCP	MTU – 104
IPv4 + UDP	MTU – 92
IPv6 + TCP	MTU – 124
IPv6 + UDP	MTU – 112

5.3.2.16.2 Header information for fragmentation and assembly

Table 27 shows one example of the header information for a UDP message fragmentation.

Table 28 shows one example of the header information for a TCP message fragmentation. hd_cbn, hd_tbn and hd_bsize refer to 5.3.2.16, 5.3.2.18 and 5.3.2.19 respectively.

Table 27 – Example of header information for a UDP message fragmentation

Example of fragmentation	Fragmented into four packets				No fragmentation
	1st packet	2nd packet	3rd packet	4th packet	Single packet
hd_seq	100	100	100	100	101
hd_cbn / hd_tbn	1 / 4	2 / 4	3 / 4	4 / 4	1 / 1
hd_bsize	1 472	1 472	1 472	340	1 024
hd_ml	4 564	4 564	4 564	4 564	1 024

Table 28 – Example of header information for a TCP message fragmentation

Example of fragmentation	Fragmented into four packets				No fragmentation
	1st packet	2nd packet	3rd packet	4th packet	Single packet
hd_seq	100	100	100	100	101
hd_cbn / hd_tbn	1 / 4	2 / 4	3 / 4	4 / 4	1 / 1
hd_bsize	1 460	1 460	1 460	376	1 024
hd_ml	4 564	4 564	4 564	4 564	1 024

5.3.2.17 hd_cbn

This field shall specify a current fragmented PDU block number (more than 1).

5.3.2.18 hd_tbn

This field shall specify a total block number of fragmented PDUs (more than 1).

5.3.2.19 hd_bsize

This field shall specify a PDU block size (including FALAR-N header size).

5.3.2.20 inqid_inq_sa

This field shall specify a source address. This hd_inqid.inq_sa value is specified in Table 29.

Table 29 – inqid_inq_sa value

Elements of hd_inqid field	Description		
	Inq-PDU, Ninq-PDU	Reply-PDU	Other PDU
Inqid_inq_sa	Source address(= hd_sa)	Same value that is set to inqid_inq_sa of the received Inq-PDU or Ninq-PDU	0

5.3.2.21 inqid_tr_adr

This field shall indicate the inquiry-transaction sequence number of the inquiry / N-inquiry message. The range of values is from 0x00000001 to 0xFFFFFFFF, and used cyclically. When an N-inquiry message is transmitted, this value shall be incremented by one.

In the case of retransmitting the N-inquiry message, the same value shall be used.

This inqid_tr_adr value is managed separately for each inqid_id_seq. Refer to 5.3.2.22.

This inqid_tr_adr value is specified in Table 30.

Table 30 – inqid_tr_adr value

Elements of hd_inqid field	Description			
	Inq-PDU	Ninq-PDU	Reply-PDU	Other PDU
Inqid_tr_adr	any	1 .. 0xFFFFFFFF	Same value that is set to inqid_tr_adr of the received Inq-PDU or Ninq-PDU	0

5.3.2.22 inqid_id_seq

This field shall specify the inquiry-identification sequence number. Each inquiry / N-inquiry message requires a unique inqid_inq_seq value in the source node. This inqid_id_seq value is specified in Table 31.