

INTERNATIONAL STANDARD

**Industrial communication networks – Fieldbus specifications –
Part 6-26: Application layer protocol specification – Type 26 elements**



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.

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 It also gives information on projects, replaced and withdrawn publications.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



IEC 61158-6-26

Edition 1.0 2019-06

INTERNATIONAL STANDARD

**Industrial communication networks – Fieldbus specifications –
Part 6-26: Application layer protocol specification – Type 26 elements**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-7016-5

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD.....	10
INTRODUCTION.....	12
1 Scope.....	13
1.1 General.....	13
1.2 Specifications	14
1.3 Conformance	14
2 Normative references	14
3 Terms, definitions, symbols, abbreviations and conventions	15
3.1 Terms and definitions from other ISO/IEC standards.....	15
3.1.1 Terms and definitions from ISO/IEC 7498-1	15
3.1.2 Terms and definitions from ISO/IEC 8822	16
3.1.3 Terms and definitions from ISO/IEC 9545	16
3.1.4 Terms and definitions from ISO/IEC 8824-1	16
3.1.5 Terms and definitions from ISO/IEC 8825-1	17
3.2 Type 26 specific terms and definitions	17
3.3 Abbreviations and symbols	21
3.4 Conventions.....	23
3.4.1 Conventions used in state machines.....	23
3.4.2 Convention for abstract syntax description.....	24
3.4.3 Convention for reserved bits and octets	24
3.4.4 Conventions for bit description in octets	24
4 FAL syntax description	25
4.1 General.....	25
4.2 Overview of Type 26 fieldbus	26
4.2.1 Application field and Common-memory	26
4.2.2 Structure of Type 26 protocol.....	27
4.2.3 Structure of Type 26 FAL.....	28
4.2.4 Data link layer	29
4.3 Operating principle.....	29
4.3.1 Overview	29
4.3.2 Logical ring maintenance	30
4.3.3 Node addition	33
4.3.4 Node in a logical ring	36
4.3.5 Node drop-out	36
4.3.6 Data transmission.....	37
4.3.7 Data transmission frames	46
4.4 FAL PDU abstract syntax	49
4.4.1 Basic abstract syntax.....	49
4.4.2 Transparent-msg- PDU	51
4.4.3 Token-PDU.....	51
4.4.4 Participation-req-PDU	51
4.4.5 Byte-block-read PDUs	51
4.4.6 Byte-block-write PDUs	52
4.4.7 Word-block-read PDUs	52
4.4.8 Word-block-write PDUs.....	52
4.4.9 Network-parameter-read PDUs	52
4.4.10 Network-parameter-write PDUs.....	53



4.4.11	Stop-command PDUs	53
4.4.12	Operation-command PDUs	53
4.4.13	Profile-read PDUs	53
4.4.14	Trigger-PDU	54
4.4.15	Log-data-read PDUs	54
4.4.16	Log-data-clear PDUs	54
4.4.17	Message-return PDUs	54
4.4.18	Vendor-specific-msg PDUs	55
4.4.19	Start-TK-hld-time-mrmt PDUs	55
4.4.20	Terminate-TK-hld-time-mrmt PDUs	55
4.4.21	Start-GP_Comm-sndr-log PDUs	56
4.4.22	Terminate-GP_Comm-sndr-log PDUs	56
4.4.23	Set-remote-node-config-para PDUs	56
4.4.24	Read-rmt-partici-node-mgt-info-para PDUs	56
4.4.25	Read-rmt- node-mgt-info-para PDUs	57
4.4.26	Read-rmt-node-set-info-para PDUs	57
4.4.27	Reset-node PDUs	57
4.4.28	Cyclic-data PDUs	57
4.5	Data type assignments	57
5	Transfer syntax	59
5.1	Encoding rules	59
5.1.1	Basic encoding	59
5.1.2	Fixed length Unsigned encoding	59
5.1.3	Fixed length BitString encoding	59
5.1.4	OctetString encoding	59
5.1.5	SEQUENCE encoding	60
5.2	PDU elements encoding	60
5.2.1	FALARHeader	60
5.2.2	Transparent-msg PDU	63
5.2.3	Token-PDU	64
5.2.4	Participation-req-PDU	65
5.2.5	Byte-block-read PDUs	66
5.2.6	Byte-block-write PDUs	67
5.2.7	Word-block-read PDUs	69
5.2.8	Word-block-write PDUs	71
5.2.9	Network-parameter-read PDUs	73
5.2.10	Network-parameter-write PDUs	76
5.2.11	Stop-command PDUs	79
5.2.12	Operation-command PDUs	81
5.2.13	Profile-read PDUs	83
5.2.14	Trigger-PDU	85
5.2.15	Log-data-read PDUs	86
5.2.16	Log-data-clear PDUs	92
5.2.17	Message-return PDUs	94
5.2.18	Vendor-specific-msg PDUs	96
5.2.19	Start-TK-hld-time-mrmt PDUs	98
5.2.20	Terminate-TK-hld-time-mrmt PDUs	100
5.2.21	Start-GP_Comm-sndr-log PDUs	103
5.2.22	Terminate-GP_Comm-sndr-log PDUs	104

5.2.23	Set-remote-node-config-para PDUs	107
5.2.24	Read-rmt-partici-node-mgt-info-para PDUs	110
5.2.25	Read-rmt- node-mgt-info-para PDUs.....	112
5.2.26	Read-rmt-node-set-info-para PDUs.....	115
5.2.27	Reset-node PDUs	117
5.2.28	Cyclic-data PDUs	118
6	FAL protocol state machines structure	120
6.1	Overview.....	120
6.2	Common variables, parameters, timers, counters, lists and queues	121
6.2.1	V(3CWT), P(3CWT), T(3CWT): Three-lap-time-period-of-the-token-circulation	121
6.2.2	V(ACK): ACK received	121
6.2.3	V(ACK_TN): ACK to this node	121
6.2.4	V(AWT), P(AWT), T(AWT): Waiting-time-period-for-receiving-message-acknowledge.....	122
6.2.5	V(CBN): Current fragment number for fragmented cyclic-data transmission	122
6.2.6	V(CTFG): Cyclic-data fragment transfer	122
6.2.7	V(CTRen), P (CTRen): Cyclic-data receive enable	122
6.2.8	V(CTRQ): Cyclic-data transfer request.....	122
6.2.9	C(MCNT): Cumulative count of message transmission carried over	122
6.2.10	V(MCV): Message transmission carried over.....	122
6.2.11	V(NMTP): No message transmission in previous cycle.....	123
6.2.12	V(MFT), P(MFT), T(MFT): Allowable-minimum-frame-Interval-Time	123
6.2.13	V(MmtCntType): Measurement control type	123
6.2.14	V(MRVRQ): Message receive request.....	123
6.2.15	V(MSRQ): Message transfer request	123
6.2.16	Q(MSRXQ): Message-RX-Queue	123
6.2.17	Q(MTXQ):Message-TX-Queue	124
6.2.18	V(PAT), P(PAT), T(PAT): Participation-request-frame-acceptance-time	124
6.2.19	V(PnMgtIF): Participation-node-management-information List.....	124
6.2.20	V(PWT), T(PWT): Participation-request-frame-transmission-waiting-time.....	124
6.2.21	V(RCT): Allowable-refresh-cycle-time	124
6.2.22	V(RMT), T(RMT): Refresh-cycle-measurement-time.....	124
6.2.23	C(RTX): Retransmission count.....	125
6.2.24	V(SEQ): Sequence number value List.....	125
6.2.25	V(SN): Successor node	125
6.2.26	V(SrtMmt): Measurement started	125
6.2.27	Q(SVRXQ): Server-RX Queue	125
6.2.28	Q(SVTXQ): Server-TX Queue	125
6.2.29	V(TBN), P(TBN): Total fragment number of Cyclic-data	125
6.2.30	V(TDT), P(TDT), T(TDT): Joining-token-detection-time	125
6.2.31	V(THT), P(THT), T(THT): Token-holding-time	126
6.2.32	V(TK): Token holding	126
6.2.33	V(TKH): Token holding node.....	126
6.2.34	V(TN): Node identifier number	126
6.2.35	V(TrWT), T(TrWT): Trigger-frame-transmission-waiting-time	126
6.2.36	V(TSZ), P(TSZ): Total cyclic-data size	126
6.2.37	V(TW), P(TW), T(TW)(): Token-watchdog-time	126

6.2.38	V(VSEQ): Version of sequence number value List	126
6.3	Functions used in state tables	127
7	FAL service protocol machine (FSPM)	129
7.1	Overview	129
7.2	Cyclic-data protocol machine	130
7.2.1	Overview	130
7.2.2	Cyclic-data primitives between FAL user and FSPM	130
7.2.3	State table	131
7.3	Message data protocol machine	132
7.3.1	Overview	132
7.3.2	Message-data primitive between FAL user and FSPM	132
7.3.3	State table	136
7.4	Load measurement protocol machine	144
7.4.1	Overview	144
7.4.2	Load measurement primitives between FAL user and FSPM	144
7.4.3	State table	146
7.5	General purpose communication server protocol machine	149
7.5.1	Overview	149
7.5.2	GP command server primitives between FAL user and FSPM	149
7.5.3	State table	150
7.6	Network management protocol machine	152
7.6.1	Overview	152
7.6.2	Network management primitives	152
7.6.3	State table	153
8	Application relationship protocol machine (ARPM)	155
8.1	Overview	155
8.2	Cyclic-TX/RX control	156
8.2.1	Overview	156
8.2.2	Cyclic-TX/RX control primitives between FSPM and ARPM	156
8.2.3	State table	157
8.3	Message-TX/RX control	157
8.3.1	Overview	157
8.3.2	Message-TX/RX control primitives between FSPM and ARPM	158
8.3.3	State table	158
8.4	Command server TX/RX control	158
8.4.1	Overview	158
8.4.2	Command server TX/RX primitives between FSPM and ARPM	159
8.4.3	State table	159
8.5	AR control	160
8.5.1	Overview	160
8.5.2	AR control primitives between FSPM and ARPM	160
8.5.3	State table	160
9	DLL mapping protocol machine (DMPM)	179
9.1	Overview	179
9.2	Mapping of DMPM service primitives and DLL service primitives	179
9.3	Mapping DMPM service port to DL-SAP	181
9.4	Mapping of Network address to each node	182
	Bibliography	183

Figure 1 – Bit identification in an octet	25
Figure 2 – Bit identification in multiple octets (four-octet case).....	25
Figure 3 – Data sharing with the CM	27
Figure 4 – Protocol stack for Type 26 fieldbus	28
Figure 5 – The structure of ASEs for Type 26 FAL	29
Figure 6 – A token circulation on a logical ring	30
Figure 7 – Logical ring recovery	32
Figure 8 – An example in case of start simultaneously with another node	34
Figure 9 – Start alone case	35
Figure 10 – Node addition: in-ring start-up state	36
Figure 11 – Data sharing with the CM	38
Figure 12 – Configuration of the Common-memory	39
Figure 13 – APDUs of cyclic-data frames containing fragmented data.....	40
Figure 14 – Example of sequential diagram of ACK over UDP channel	43
Figure 15 – Delivery confirmation checked by TCP protocol.....	44
Figure 16 – Train of data frames and a token frame	46
Figure 17 – Frame structure.....	47
Figure 18 – Structure of Trans-msgData	64
Figure 19 – Structure of B_BlK_Rd_rspData with M_RLT = 0	67
Figure 20 – Structure of B_BlK_Rd_rspData in case of M_RLT = 1	67
Figure 21 – Structure of B_BlK_Wt_reqDat.....	69
Figure 22 – Structure of B_BlK_Wt_rspData in case of M_RLT = 1	69
Figure 23 – Structure of W_BlK_Rd_rspData with M_RLT = 0	71
Figure 24 – Structure of W_BlK_Rd_rspData in case of M_RLT = 1	71
Figure 25 – Structure of W_BlK_Wt_reqDat.....	73
Figure 26 – Structure of W_BlK_Wt_rspData in case of M_RLT = 1	73
Figure 27 – Structure of Net-para-Rd-rspData.....	75
Figure 28 – Structure of Net-para-Rd-rspData with M_RLT = 1	76
Figure 29 – Structure of Net-para-Wrt-reqData.....	78
Figure 30 – Structure of Net-para-Wrt-rspData with M_RLT = 1	79
Figure 31 – Structure of Stop-cmdData with M_RLT = 1	81
Figure 32 – Structure of Op-cmdData with M_RLT = 1	82
Figure 33 – Structure of Profile-readData with M_RLT = 0	84
Figure 34 – Structure of Profile-readData with M_RLT = 1	85
Figure 35 – Structure of Log-readData with M_RLT = 0.....	88
Figure 36 – Structure of Log-readData with M_RLT = 1.....	92
Figure 37 – Structure of Log-clearData	93
Figure 38 – Structure of Msg-return-reqData.....	95
Figure 39 – Structure of Msg-return-rspData	95
Figure 40 – Structure of V_msg_reqData	97
Figure 41 – Structure of V_msg_rspData in case of M_RLT = 0	98
Figure 42 – Structure of V_msg_rspData in case of M_RLT = 1	98

Figure 43 – Token-holding-time measurement result	102
Figure 44 – Structure of Sndr-logData	106
Figure 45 – Structure of Set-remote-node-config-para-ReqData	108
Figure 46 – Structure of Set-remote-node-config-para-RspData	109
Figure 47 – Structure of Read-rmt-partici-node-mgt-info-ReqData	111
Figure 48 – Structure of Read-rmt-partici-node-mgt-info-RspData	111
Figure 49 – Structure of Rmt-node-mgt-info-paraData	114
Figure 50 – Structure of Set-info-para-read-data	116
Figure 51 – Structure of ACKdata	119
Figure 52 – Relationship between FAL protocol machines	121
Figure 53 – Overall structure of FSPM	130
Figure 54 – State transition diagram of Cyclic-data protocol machine	131
Figure 55 – State transition diagram of Message-data protocol machine	136
Figure 56 – State transition diagram of Load measurement protocol machine	146
Figure 57 – State transition diagram of GP-command-server protocol machine	150
Figure 58 – State transition diagram of Network management protocol machine	153
Figure 59 – Overall structure of ARPM	156
Figure 60 – State transition diagram of Cyclic-TX/RX control	157
Figure 61 – State transition diagram of Message-TX/RX control	158
Figure 62 – State transition diagram of Command server TX/RX protocol machine	159
Figure 63 – Overall state transition diagram of AR control protocol machine	161
Figure 64 – State transition diagram for message-data transmission	173
Figure 65 – State transition diagram for ACK creation and message-data reception	176
Figure 66 – Overall structure of DMPM	179
Figure 67 – DL-SAP mapping	181
Figure 68 – Structure of IP address	182
Table 1 – Conventions used for state machines	23
Table 2 – Conventions used in state machine	23
Table 3 – Available functions to message-data transfer on UDP channel	42
Table 4 – Data transmission frame and the TCD value	47
Table 5 – Upper layer operating condition matrix	61
Table 6 – Transparent-msg-PDU specific values	64
Table 7 – Token-PDU specific values	65
Table 8 – Participation-req -PDU specific values	65
Table 9 – Byte-block-read-req-PDU specific values	66
Table 10 – Byte-block-read-rsp-PDU specific values	66
Table 11 – Byte-block-write-req-PDU specific values	68
Table 12 – Byte-block-write-rsp-PDU specific values	68
Table 13 – Word-block-read-req-PDU specific values	70
Table 14 – Word-block-read-rsp-PDU specific values	70
Table 15 – Word-block-write-req-PDU specific values	72
Table 16 – Word-block-write-rsp-PDU specific values	72

Table 17 – Network-parameter-read-req-PDU specific values	74
Table 18 – Network-parameter-read-rsp-PDU specific values	74
Table 19 – Values of data elements of Net-para-Rd-rspData	76
Table 20 – Network-parameter-write-req-PDU specific values	77
Table 21 – Network-parameter-write-rsp-PDU specific values	77
Table 22 – Values of the data elements of Net-para-Wrt-reqData	78
Table 23 – Stop-command-req-PDU specific values	79
Table 24 – Stop-command-rsp-PDU specific values	80
Table 25 – Operation-command-req-PDU specific values	81
Table 26 – Operation-command-rsp-PDU specific values	82
Table 27 – Profile-read-req-PDU specific values	83
Table 28 – Profile-read-rsp-PDU specific values	83
Table 29 – Trigger-PDU specific values	86
Table 30 – Log-data-read-req-PDU U specific values	87
Table 31 – Log-data-read-rsp-PDU specific values	87
Table 32 – Contents of Log-readData	88
Table 33 – Log-data-clear-req-PDU specific values	92
Table 34 – Log-data-clear-rsp-PDU specific values	93
Table 35 – Message-return-req-PDU specific values	94
Table 36 – Message-return-rsp-PDU specific values	94
Table 37 – Vendor-specific-msg-req-PDU specific values	96
Table 38 – Vendor-specific-msg-rsp-PDU specific values	96
Table 39 – Start-TK-hld-time-mrmt-req-PDU specific values	99
Table 40 – Start-TK-hld-time-mrmt-rsp-PDU specific values	99
Table 41 – Terminate-TK-hld-time-mrmt-req-PDU specific values	100
Table 42 – Terminate-TK-hld-time-mrmt-rsp-PDU specific values	101
Table 43 – Value of the data element of TK-hld-timeData	102
Table 44 – Start-GP_Comm-sndr-log-req-PDU specific values	103
Table 45 – Start-GP_Comm-sndr-log-rsp-PDU specific values	104
Table 46 – Terminate-GP_Comm-sndr-log-req-PDU specific values	104
Table 47 – Terminate-GP_Comm-sndr-log-rsp-PDU specific values	105
Table 48 – Value of the data element of Sndr-logData	106
Table 49 – Set-remote-node-config-para-req-PDU specific values	107
Table 50 – Set-remote-node-config-para-rsp-PDU specific values	107
Table 51 – Value of the data element of Set-remote-node-config-para-ReqData	108
Table 52 – Bit definition of Update flag	109
Table 53 – Value of the data element of Set-remote-node-config-para-RspData	109
Table 54 – Read-rmt-partici-node-mgt-info-para-req-PDU specific values	110
Table 55 – Read-rmt-partici-node-mgt-info-para-rsp-PDU specific values	110
Table 56 – Value of the data element of Read-rmt-partici-node-mgt-info-RspData	112
Table 57 – Read-rmt- node-mgt-info-para-req-PDU specific values	112
Table 58 – Read-rmt- node-mgt-info-para-rsp-PDU specific values	113
Table 59 – Value of the data element of Rmt-node-mgt-info-paraData	114

Table 60 – Bit definition of Node status.....	115
Table 61 – Read-rmt-node-set-info-para-req-PDU specific values	115
Table 62 – Read-rmt-node-set-info-para-rsp-PDU specific values	116
Table 63 – Value of the data element of Set-info-para-read-data	117
Table 64 – Rest-node-req-PDU specific values	117
Table 65 – Rest-node-rsp-PDU specific values	118
Table 66 – Cyclic-data-PDU specific values	118
Table 67 – Value of the element of ACKdata.....	120
Table 68 – Value of R_STSx field	120
Table 69 – Value of R_STSx field	122
Table 70 – Functions used in state tables	127
Table 71 – Cyclic-data primitives between FAL user and FSPM	130
Table 72 – State table of Cyclic-data protocol machine	131
Table 73 – Message-data primitives between FAL user and FSPM	132
Table 74 – State table of Message-data protocol machine	136
Table 75 – Load measurement primitives between FAL user and FSPM.....	145
Table 76 – State table of Load measurement protocol machine.....	146
Table 77 – GP command server primitives between FAL user and FSPM.....	150
Table 78 – State table of General purpose command server protocol machine.....	151
Table 79 – Primitives used in network management protocol machine	152
Table 80 – State table of Network management protocol machine.....	154
Table 81 – Cyclic-TX/RX control primitives between FSPM and ARPM	157
Table 82 – State table of Cyclic-TX/RX control.....	157
Table 83 – Message-TX/RX control primitives between FSPM and ARPM.....	158
Table 84 – State table of Message-TX/RX control	158
Table 85 – Command server TX/RX primitives between FSPM and ARPM	159
Table 86 – State table of Command server TX/RX protocol machine.....	159
Table 87 – AR control primitives between FSPM and ARPM	160
Table 88 – Overall AR control state table	162
Table 89 – State table for message-data transmission	174
Table 90 – State table for ACK creation and message-data reception	177
Table 91 – Mapping of DMPM primitives and DLL service primitives	180
Table 92 – Supposed Transport service primitives	180
Table 93 – Mapping of output and input ports to DL-SAP	181

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
 FIELDBUS SPECIFICATIONS –**
**Part 6-26: Application layer protocol specification –
 Type 26 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 61158-6-26 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement and control.

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.

A bilingual version of this publication may be issued at a later date.

INTRODUCTION

This part of IEC 61158 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 6-26: Application layer protocol specification – Type 26 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 26 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 International Standard defines in an abstract way the externally visible behavior provided by the Type 26 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 61158-5-26, and
- b) define the externally visible behavior associated with their transfer.

This document specifies the protocol of the Type 26 fieldbus Application Layer, in conformance with the OSI Basic Reference Model (see ISO/IEC 7498-1) and the OSI Application Layer Structure (see 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 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 61158-5-26.

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 the 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 61158-1:2019, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-5-26:2019, *Industrial communication networks – Fieldbus specification – Part 5-26: Application layer service definition – Type 26 elements*

IEC 61784-2:2019, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC/IEEE 8802-3*

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

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*

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

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

ISO/IEC 8825-1, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

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

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

IETF RFC 768, *User Datagram Protocol*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 791, *Internet Protocol*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 792, *Internet Control Message Protocol*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 793, *Transmission Control Protocol*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 796, *Address mappings*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 826, *An Ethernet Address Resolution Protocol*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 894, *A Standard for the Transmission of IP Datagrams over Ethernet Networks*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 919, *Broadcasting Internet Datagrams*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 922, *Broadcasting Internet Datagrams in the presence of subnets*, available at <http://www.ietf.org> [viewed 2018-09-20]

IETF RFC 950, *Internet Standard Subnetting Procedure*, available at <http://www.ietf.org> [viewed 2018-09-20]

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 Terms and definitions from other ISO/IEC standards

3.1.1 Terms and definitions from ISO/IEC 7498-1

For the purposes of this document, the following terms and definitions 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 Terms and definitions from ISO/IEC 8822

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

- a) abstract syntax
- b) presentation context

3.1.3 Terms and definitions from ISO/IEC 9545

For the purposes of this document, the following terms and definitions 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 Terms and definitions from ISO/IEC 8824-1

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

- a) object identifier
- b) type
- c) value
- d) simple type
- e) structured type
- f) component type
- g) tag
- h) Boolean type
- i) true
- j) false
- k) integer type
- l) bitstring type
- m) octetstring type
- n) null type

- o) sequence type
- p) sequence of type
- q) choice type
- r) tagged type
- s) any type
- t) module
- u) production

3.1.5 Terms and definitions from ISO/IEC 8825-1

For the purposes of this document, the following terms and definitions given in ISO/IEC 8825-1 apply:

- a) encoding (of a data value)
- b) data value
- c) identifier octets (the singular form is used in this document)
- d) length octet(s) (both singular and plural forms are used in this document)
- e) contents octets

3.2 Type 26 specific terms and definitions

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

3.2.1

application relationship

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation.

Note 1 to entry: This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities.

3.2.2

application process object

component of an application process that is identifiable and accessible through an FAL application relationship

3.2.3

application relationship endpoint

context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

Note 1 to entry: Each application process involved in the application relationship maintains its own application relationship endpoint.

3.2.4

client

<object view> object which uses the services of another (server) object to perform a task

3.2.5

client

<communication view> initiator of a message to which a server react

3.2.6

common memory

virtual memory accessible with logically unique address used for the cyclic-data transmission

Note 1 to entry: It is composed of the memory-area-1 and -2 of the memory size 512 words and 8 192 words respectively and is shared with the Type 26 nodes in a Type 26 fieldbus network.

3.2.7

connection

logical binding between application objects that are within the same or different devices

Note 1 to entry: Connections can be either point-to-point or multipoint.

3.2.8

cyclic-data frame

frame used for the Type 26 cyclic-data transmission

3.2.9

cyclic-data transmission

data transmission with which a Type 26 node sends out data on a pre-assigned area for the node on the common-memory, in broadcasting to all other Type 26 nodes every time the node obtains a token

3.2.10

device

physical hardware connected to the link

Note 1 to entry: A device can contain one or more than one node.

3.2.11

device profile

collection of device dependent information and functionality providing consistency between similar devices of the same device type

3.2.12

error

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

3.2.13

event

instance of a change of conditions

3.2.14

FA-link header

header part of the APDU used for Type 26 data transmission

3.2.15

FA-link protocol

communication protocol for Type 26 fieldbus

Note 1 to entry: It defines the token management, the cyclic data transmission and the message data transmission, with which each Type 26 node can perform data transmission on an equal basis.

3.2.16

final-cyclic-data frame

cyclic-data frame sent out immediately before sending a token-frame out of a Type 26 node

3.2.17

frame

synonym for DLPDU

3.2.18**invocation**

act of using a service or other resource of an application process

Note 1 to entry: Each invocation represents a separate thread of control that can be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation.

3.2.19**in-ring-startup state**

state of a node attempting to participate in Type 26 fieldbus network in the valid-linking state

3.2.20**major-version**

indication of the major-version number of the FA-link protocol

Note 1 to entry: The version number of the FA-link protocol is given as a combination of a major- and a minor-version number. The value of the major-version number is in the range of 1 to 15 (encoded with four bits) and corresponds to the digits before the decimal point (for example, the major-version number is 3 for the FA-link protocol Version 3.01.).

3.2.21**management information**

network-accessible information that supports managing the operation of the fieldbus system, including the application layer

Note 1 to entry: Managing includes functions such as controlling, monitoring, and diagnosing.

3.2.22**message-data frame**

frame used for the Type 26 message-data transmission

3.2.23**message-data transmission**

sporadic data transmission with which a Type 26 node sends out the message data of request, response, information report, and notification upon event happened at the node and so forth, to other Type 26 nodes

3.2.24**minor-version**

indication of the minor-version number of the FA-link protocol

Note 1 to entry: The version number of the FA-link protocol is given as a combination of a major- and a minor-version number. The value of the minor-version number is in the range of 1 to 15 (encoded with four bits) and corresponds to the digits after the decimal point (for example, the minor-version number is 1 for the FA-link protocol Version 3.01.).

3.2.25**network load measurement function**

provision for monitoring and measurement of the communication load over a Type 26 fieldbus with a Type 26 node

3.2.26**network-startup state**

state of a node attempting to participate in Type 26 fieldbus network not yet in the valid-linking state

3.2.27**participation-request-frame**

frame used for a Type 26 node notifying of its participation-request in a Type 26 fieldbus network

3.2.28

protocol type

identification of communication protocol

3.2.29

publisher

role of an AR endpoint that transmits APDUs onto the fieldbus for consumption by one or more subscribers

3.2.30

resource

processing or information capability of a subsystem

3.2.31

subscriber

role of an AREP in which it receives APDUs produced by a publisher

3.2.32

sequence number

consecutive number used to determine whether a receiving message-data from other nodes has been previously received and is redundant

Note 1 to entry: The value is set to 0 (zero) at the time of the network-startup and is increased by 1 (one) every time on a message-data transmission completed.

3.2.33

server

<communication view> role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request

3.2.34

server

<object view> object which provides services to another (client) object

3.2.35

token-frame

frame used for hand-over the token between Type 26 nodes

3.2.36

transaction code

code used to discriminate and identify each data transmission frame, and which represents the transaction of a receiving frame

3.2.37

trigger-frame

frame used in the network-startup state for a Type 26 node notifying of its entrance into a new participation-request over a Type 26 fieldbus network

3.2.38

valid-linking state

state in which a token is consecutively circulated among nodes over Type 26 fieldbus network

3.2.39

version of sequence number

version number of the sequence number in a receiving message-data from other nodes used to determine whether the receiving message-data is valid and is not overdue

Note 1 to entry: The value is set at the time when a Type 26 node completes to participate in a Type 26 fieldbus network, and is maintained until the node drops out of the Type 26 fieldbus network.

3.2.40**virtual-address-space**

virtual memory space with pseudo logical address, used for the message-transmission

3.3 Abbreviations and symbols

3CWT	Waiting time period for 3 laps of token circulation
AE	Application Entity
AL	Application Layer
ALME	Application layer management entity
ALP	Application layer protocol
AP	Application Process
APDU	Application Protocol Data Unit
APO	Application Process Object
AR	Application Relationship
AREP	Application Relationship End Point
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
AWT	Waiting time period for receiving message-acknowledge
ASN.1	Abstract Syntax Notation One
BCD	Binary Coded Decimal
BCT	Broadcast transmission
BER	Basic Encoding Rule
BSIZE	Current-fragment-block-size
CBN	Current-fragment-block-number
C_AD1	Data-head-address on common-memory-area-1
C_AD2	Data-head-address on common-memory-area-2
CM	Common Memory
cnf	confirmation primitive
C_SZ1	Data-size on common-memory-area-1
C_SZ2	Data-size on common-memory-area-2
DA	Destination-address
DCS	Distributed Control system
DL	Data-link
DLC	Data-link Connection
DLCEP	Data-link Connection End Point
DLL	Data-link layer
DLPDU	Data-link Protocol Data Unit
DLSAP	Data-link Service Access Point
DLSDU	Data-link Service Data Unit
DNA	Destination-node-number
DNS	Domain name service
FA	Factory Automation
FAL	Fieldbus Application Layer
FIFO	First In First Out
HMI	Human-machine interface
H_TYPE	Header-type
ID	Identifier

IEC	International Electrotechnical Commission
ind	indication primitive
IP	Internet protocol
ISO	International Organization for Standardization
LME	Layer management entity
LSB	least significant bit
MAC	Media Access Control
M_ADD	Message-offset-address in the virtual-address-space
MAJ_VER	FA-link protocol version: major-version
M_CTL	Message-control
MFT	Allowable-minimum-frame-interval-time
MIN_VER	FA-link protocol version: minor-version
MSB	most significant bit
MSN	Manufacturer's model name of a Type 26 node
M_SZ	Message-data-size in the virtual-address-space
M_RLT	Message result
NC	Numerical Controller
NDN	Node name of Type 26 node
OSI	Open Systems Interconnection
PAT	Participation-request-frame-acceptance-time
PC	Personal Computer
PDU	Protocol Data Unit
PhL	Ph-layer
PLC	Programmable Logic Controller
PPT	Peer-to-peer transmission
P_TYPE	Protocol-type
PWT	Participation-request-frame-transmission-waiting-time
RCT	Allowable-refresh-cycle-time
req	request primitive
RMT	Refresh-cycle-measurement-timer
rsp	response primitive
RC	Robot Controller
RT	Real Time
SA	Source-address
SAP	Service Access Point
SDU	Service Data Unit
SEQ	Sequence-number
SNA	Source-node-number
TBN	Total-fragment-block-number
TCD	Transaction code
TCP	Transmission Control Protocol
TDT	Joining-token-detection-time
TBN	Total-fragment-block-number
TFL	Total frame length in octet including both of header and data
THT	Token-holding-time
TrWT	Waiting time period before sending out a trigger-frame

TW	Token-watchdog
UDP	User Datagram protocol
ULS	Upper layer status
VDN	Vender code of Type 26 node
V_SEQ	Version of sequence number

3.4 Conventions

3.4.1 Conventions used in state machines

Protocol sequences are described by means of state machines.

The state machine description and the meaning of the elements of a state machine description are defined in tabular form as shown in Table 1.

Each row of the 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 transitions to the next state.

Table 1 – Conventions used for state machines

#	Current state	Event / condition => action	Next state
Name or number of this state transition	The current state to which this state transition applies	Events or conditions that trigger this state transaction. => The actions that are taken when the above events or conditions are met. The actions are always indented below events or conditions	The next state after the actions in this transition is taken

The conventions used in the state machines are shown in Table 2.

Table 2 – Conventions used in state machine

Notation	Description
:=	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. xxx is a parameter name. Example: Identifier:= reason means value of a 'reason' parameter is assigned to a parameter called 'Identifier'. "xxx" Indicates fixed value Example: Identifier:= "abc" means value "abc" is assigned to a parameter named 'Identifier'.
=	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 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
<>	A logical condition to indicate an item on the left is not equal to an item on the right
&&	Logical "AND"
	Logical "OR"
!	Negation operator

Notation	Description
+, -, X, /	Arithmetic operator
;	Breakpoint
- - "comment" or /* "comment" */	Comments out conditions and actions. The "- - "comment" " type notation is used in one-line comments, and the "/* "comment text" */ " type notation is used in one or multiple-line comments.

Further the following conventions apply.

- The construct described below as an example allows the execution of a sequence of actions in a loop within one transition. The loop is executed for all values from start_value to end_value.

Example:

```
for (Identifier:= start_value to end_value)
    actions
endfor
```

- The construct described below as an example allows the execution of alternative actions depending on some condition (which might be the value of some identifier or the outcome of a previous action) within one transition.

Example:

```
If (condition)
then
    actions
else
    actions
endif
```

- Notations as defined in C language according to ISO/IEC 9899 is used to describe conditions and actions.

3.4.2 Convention for abstract syntax description

This document uses a subset of ASN.1 according to ISO/IEC 8824-1 for the description.

The description is provided by means of a combination of SEQUENCE type, CHOICE type, basic types and subtype elements defined in ASN.1.

3.4.3 Convention for reserved bits and octets

The term "reserved" is used to describe bits in octets or whole octets for indicating that certain values within the range of a parameter are reserved for future extensions. All the bits or the octets shall be set to zero at the sending side and is not validated intentionally at the receiving side except it is explicitly stated that a state machine shall validate the bits or octets reserved.

3.4.4 Conventions for bit description in octets

Each bit in an octet is identified by a number and is described as Bit n. Figure 1 shows the bit identification in an octet.

When specifying bit identification in multiple octets, the LSB of the lowest octet is identified as Bit 0, and the bit identification numbers are assigned in an ascending order. Figure 2 shows the bit identification in four octets as an example.

Bits are grouped as group of bits. Each bit or group of bits shall be addressed by its Bit Identification. When specifying multiple bits sequentially located, the range separator “..” is used. A group of bits of Bit 0 through Bit 4 denotes Bit 0..4.

Alias names are used for each bit or group of bits. The grouping of individual bits is in ascending order without gaps. The values for a group of bits are provided as binary, decimal or hexadecimal values, and the value is only valid for the group bits.

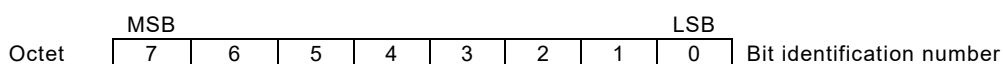


Figure 1 – Bit identification in an octet

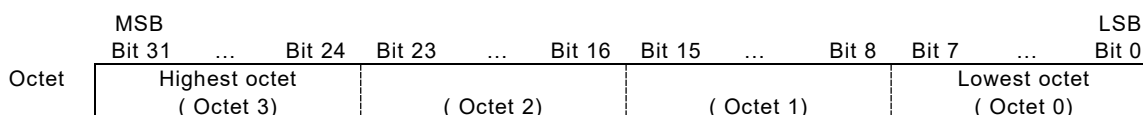


Figure 2 – Bit identification in multiple octets (four-octet case)

4 FAL syntax description

4.1 General

This document specifies the Application layer protocol of Type 26 fieldbus that is essential for the ISO/IEC/IEEE 8802-3 based FA Link network (FL-net¹) which will be incorporated as one of the communication networks for the Real-time Ethernet (RTE) defined in IEC 61784-2.

The Type 26 fieldbus meets the industrial automation market objective of providing predictable time-deterministic and reliable time-critical data transfer and means, which allow co-existence with non-time-critical data transfer over the ISO/IEC/IEEE 8802-3 series communications medium, for support of cooperation and synchronization between automation processes on field devices in a real-time application system.

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.

Plant production may be compromised due to errors, which could be introduced to the control system if the network does not provide a time-critical response. Therefore the following characteristics are required for a time-critical control network:

- A deterministic response time between the control device nodes.
- Ability to share process data seamlessly across the control system.

The Type 26 fieldbus is applicable to such an industrial automation environment, in which time-critical communications are primarily required.

¹ FL-net is the trade name of JEMA/FL-net: The Japan Electrical Manufacturers' Association / the Factory Automation Link network. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trade name. Use of the trade name requires permission of the trade name holder.

4.2 Overview of Type 26 fieldbus

4.2.1 Application field and Common-memory

The Type 26 fieldbus is applicable to a controller level network for efficiently executing the control and state data exchange between various kinds of field devices such as Programmable Logic Controllers (PLC), Robot Controllers (RC), Numerical Controllers (NC), Factory Automation (FA) controllers, Distributed Control systems (DCS), Personal Computers (PC) for the control purpose and the human-machine interfaces and so forth, in industrial automation control systems in which a time-criticality for their data exchanges is required.

The communications between these field devices also require simplicity in actual application programming as well as the time-critical transfer of the control and state data required among these field devices. To meet such requirement the Type 26 fieldbus provides so-called Common-memory (CM) which provides a function to share the control and state data seamlessly across an industrial control network system.

The CM is a virtual common memory accessible with logically unique address across the Type 26 nodes in a Type 26 fieldbus network, and the APs running over each Type 26 node can handle the CM as a global common memory shared among the Type 26 nodes over the Type 26 fieldbus network.

Cyclic-data transmission is used for refreshing and equalizing the contents of the CM among the Type 26 nodes by means of a cyclic broadcasting performed at a constant rate after a Type 26 node writes a data piece on a pre-allocated memory area of the CM for the Type 26 node.

The data distribution/refreshing function to all Type 26 nodes provided by the CM by means of the cyclic-data transmission enables each node to share the same data with each other, and to make good use of the data; i.e. a memory space allocated to each node is a container for application data in general use and provides flexibility to apply in a variety of industrial application processes.

In operational view of the CM, the memory area in the CM allocated to a certain node is used as the transmission area of the node by a cyclic-data transmission as a sender. On the other hand as receiving nodes of the cyclic-data transmission, the memory area of the sender node is used as a receiving area for the sender nodes.

Figure 3 shows the data sharing with the CM.

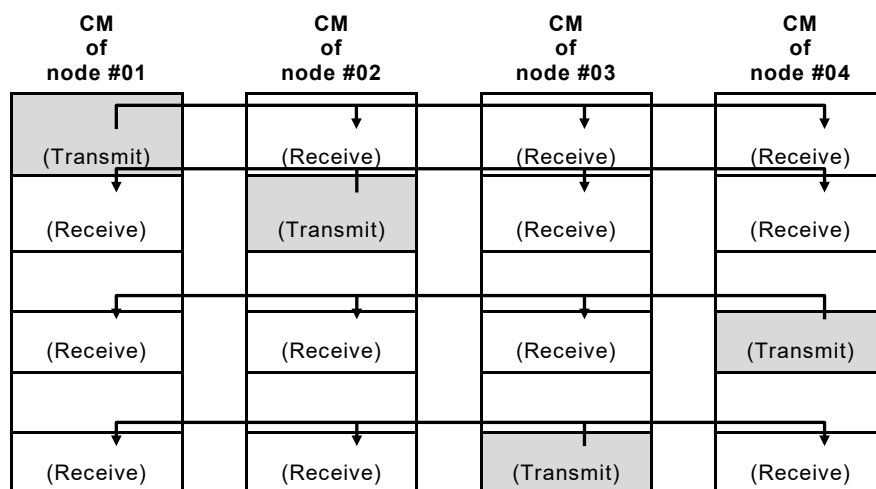


Figure 3 – Data sharing with the CM

4.2.2 Structure of Type 26 protocol

The application layer exploits the services of the data-link layer immediately below, in terms of the “three-layer” Fieldbus Reference Model which is based in part on the OSI Basic Reference Model. The OSI reference model provides a layered approach to communications standards, whereby the layers can be developed and modified independently. IEC 61158 specifies functionality from top to bottom of a full OSI stack and, potentially, some functions for the users of the stack.

Figure 4 shows the protocol stack for a Type 26 fieldbus, the OSI layers and the equivalent layers in the IEC 61158 basic fieldbus reference model.

Functions of the intermediate OSI layers, layers 3 through 6, are consolidated into the IEC 61158 data-link layer for a Type 26 fieldbus as the lower layer of the FA Link protocol. The data-link layer is that of a general-purpose Ethernet with UDP (see RFC 768), TCP (see RFC 793), IP (see RFC 791) and Ethernet layer (see RFC 894, ISO/IEC/IEEE 8802-3) as the main components. Type 26 fieldbus utilizes that of the services provided with a general-purpose Ethernet protocol suite.

The application layer for Type 26 fieldbus is the main component of the FA Link protocol stack, and is specified as the upper layer of the FA Link protocol. The application layer is mapped directly on the data-link layer as the lower layer of the FA Link protocol.

Two kinds of data transmissions are defined; i.e. Cyclic-data transmission and Message-data transmission. The cyclic-data and the message-data transmissions are the data transmission of UDP connectionless based. Data transmission based on connection-oriented TCP is only for the General-purpose-command-server transmission.

This document defines the protocol specification, and the service elements and services are defined in IEC 61158-5-26.

OSI layer		IEC 61158 layer	
7	Application		Application
6	Presentation	null	IEC 61158-5-26 IEC 61158-6-26
5	Session	null	
4	Transport		Data-link RFC 768, (RFC 793)
3	Network		RFC 791 RFC 919, RFC 922 (RFC 792, RFC 950)
2	Data-link		RFC 894/ ISO/IEC/IEEE 8802-3 MAC (RFC 826)
1	Physical		Physical ISO/IEC/IEEE 8802-3 Phy

Figure 4 – Protocol stack for Type 26 fieldbus

4.2.3 Structure of Type 26 FAL

The FAL is defined as a set of object-oriented ASEs. Each ASE specification is composed of three parts: the class definitions, the services and the protocol specification. The first two are contained and defined in IEC 61158-5-26.

The FAL ASEs and their architectural relationships for Type 26 FAL are shown in Figure 5.

Each ASE provides the services as follows:

- The cyclic-data ASE provides the cyclic data communication service with Type 26 specific common-memory defined in 6.4 of IEC 61158-5-26, and the cyclic-data transmission is performed at a constant rate for updating the common-memory;
- Message data ASE provides the message-data communication service between nodes and the message-data transmission is performed on demand by the ALS-user in order to intercommunicate between the APs running on remote nodes;
- The load measurement ASE provides services to conduct the start and the termination of the communication load measurement on the target node, and provides the measurement result containing the token holding time related and the general purpose communication load related statistical information on that node available for analyzing the total communication load condition over the Type 26 fieldbus network;
- The General purpose command server ASE provides the general purpose command server communication service, and provides the facilities for setting and reading the values of various network visible APOs of other nodes, and for requesting the start and the stop of the communication load measurement to other nodes;
- The Network management ASE provides services to access Type 26 APO attributes; modify the APO instances; configure the nodes; inform the status report on the unexpected events, the errors and the status changes of the nodes; manage network visible APOs accessed through the FAL.

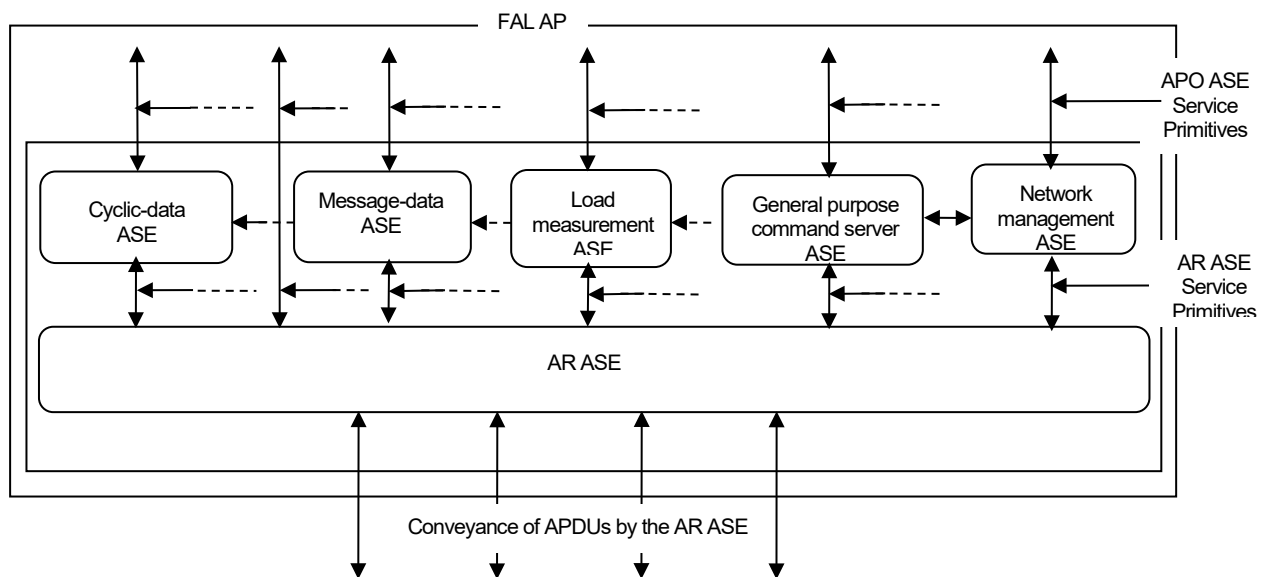


Figure 5 – The structure of ASEs for Type 26 FAL

4.2.4 Data link layer

Two types of addressing exist for DL-addressing in terms of the Fieldbus three layer reference model, one is over UDP and other is over TCP protocol channels which are represented as the T- profile in terms of the OSI reference model.

The cyclic-data and the message-data transmissions are the data transmission of UDP connectionless based. The message-data transmission based on connection-oriented TCP is only for the General-purpose-command-server transmission.

The cyclic-data and the message-data transmissions utilize the local broadcast medium/mechanism over the UDP and the TCP protocols, and are restricted generally to the local link. Requirement for the Data link layer as the FAL lower layer shall specify the broadcast communication function as the mandatory fundamental function.

With respect to the Ethernet MAC address, the default set to the Ethernet MAC interface by the device vendor is used as the MAC address, and for the IP address, class C of IP address defined in RFC 796 is assigned and used. The default value of the IP address is specified as 192.168.250.N with the value of 255.255.255.0 as the subnet mask, and the value of N shall be identical to the Node number of 1 to 254.

4.3 Operating principle

4.3.1 Overview

A token controls the right to send the data out of a Type 26 node to a Type 26 fieldbus network. The node which holds (possesses) the token can primarily perform the data transmission over the Type 26 fieldbus network.

The token is passed by Type 26 nodes, and as the token is passed from node to node a logical ring is formed. Logical ring maintenance functions within the Type 26 nodes are provided for ring initialization, lost token recovery, multiple tokens, new node addition, node drop-out, entering and exiting ring, monitoring and detection fault and so forth.

Operation in normal, steady state on the node consists of a data transmission phase and a token transmission phase. The two kinds of data transmissions for Type 26 nodes are the cyclic-data transmission and the message-data transmission.

The cyclic-data transmission is primarily carried out by the node every time the node obtains the token; on the other hand the message-data transmission is dependent on request and the communication load condition at that time. After the node has completed transmitting any data, the node passes the token to the successor by sending out a token frame.

4.3.2 Logical ring maintenance

4.3.2.1 Token passing

A token is passed by the nodes, and as the token is passed from node to node a logical ring is formed. Figure 6 shows a token circulation on a logical ring.

The token transfer is performed by sending a token frame out of a node (predecessor) to its successor after the predecessor has completed sending any data frames. The token frame contains a destination node number (DNA) and a source node number (SNA). The DNA indicates the node number of successor, and the SNA indicates the node number of predecessor.

Type 26 fieldbus is in general a broadcast type, and every Type 26 node receives the token frame. The node has the right to send out the data by recognizing and accepting the token frame of which the DNA number is identical with the node number.

The token passing is always sequential in a logical sense. During normal, steady-state operation, a data transmission phase and a token transmission phase are performed. After the the node has completed transmitting any data, the node passes the token to the successor by sending out a token frame.

The token or the right to send out data passes from node to node in a logically circular fashion. The token or the right to send out data is circulated from node to node in ascending numerical order of the node number.

The node which holds the token and has the largest node number in a Type 26 fieldbus network shall pass the token to the node which has the smallest node number. Figure 6 shows a Type 26 fieldbus network which is consisting of Node #1 to Node #N. Each node number of the Node #1 to the Node #N is 1 to N respectively in ascending numerical order. In that configuration, the Node #N which holds the token and has the largest node number shall pass the token to the Node #1 which has the smallest node number in order to maintain the logical ring of the Type 26 fieldbus network.

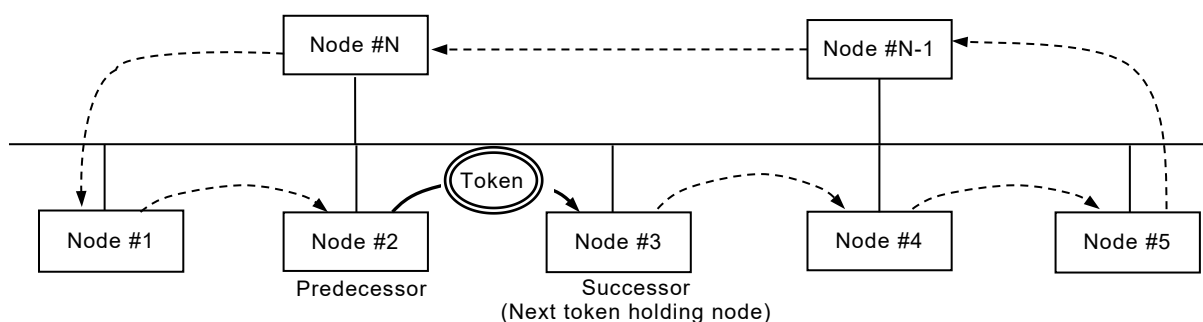


Figure 6 – A token circulation on a logical ring

4.3.2.2 Right to transmit

A node can hold the right to transmit data over the Type 26 fieldbus network after obtaining the token until the time that the node shall hand over the token to the next node or the successor.

Operation in normal, steady state on the node consists of a data transmission phase and a token transmission phase. In the data transmission phase, two kinds of data transmissions exist; i.e. the cyclic-data transmission and the message-data transmission. The cyclic-data transmission is primarily carried out by the node every time the node obtains the token; on the other hand the message-data transmission is dependent on request and the communication load condition at that time.

During the above process, the data transmission is performed as follows:

- a) The token-holding-time (THT) controls and limits the time period within which every node is able to be holding the token. The time period of holding the token by the nodes shall not exceed the time period of the THT;
- b) While holding the token, the node can send out one or more cyclic-data frames every time and further can send out at most one message-data frame;
- c) After the node has completed transmitting any data within the time period of the THT, the node shall pass the token to the successor by sending out a token frame;
- d) The token frame is never sent out upon failure that the time period of holding the token exceeds the time period of the THT.

NOTE Nodes which do not hold the token are exceptionally able to send out Type 26 specific control frames. A new node can send out these frames during the entering state in order to join in an operating Type 26 fieldbus network or the logical ring in operation. In addition, a node can send out a new token frame in order to reform and recover the logical ring, after it has been disrupted by one of the following events: token lost, node dropped out of the logical ring, failure happening within nodes, and so forth.

4.3.2.3 Logical ring recovery

When a node recognizes and determines the logical ring is disrupted whether or not it holds the token, sends out a token frame in order to reform and recover the logical ring.

Each node is monitoring the state of a token circulation on the logical ring by means of two observational timers that are the refresh-cycle-measurement-timer (RMT) and the token-watchdog-timer (TW).

The RMT is provided with each node and is an observational timer to measure the refresh-cycle-time which is the elapsed time period from the time after obtaining the token by a node until the time the node obtains the token once again. When the RMT overruns the limit of allowable-refresh-cycle-time (RCT) set to each node and common to all nodes, the RMT is considered to be time out.

The TW-timer is provided with each node and is used as an observational timer to measure the elapsed time period from the time a node obtains the token until the time the node sends out a token frame over the fieldbus network.

Each node has the maximum observational time limit to the TW-timer. Each node has the knowledge on the TW values of other nodes, for instance by means of checking the APDU header received from other nodes, and the node can utilize these TW values of other nodes to detect some kind of failure by means of checking the elapsed time period during which a node holds the token.

When the TW-sum-time in the case of using the maximum observational time limit to the TW-timer of each node overruns the limit of the sum total of the maximum observational time limit to the TW-timer of each node which resides sequentially on the logical ring from a token

holding node to this node in ascending numerical order of the node number, the TW-timer of some node is considered to be time out.

A token frame shall be sent out of a node upon time out both of the RMT and the TW-timer at a time. Both timers out at the same time results from the logical ring disrupted, so that the token frame shall be sent out to reform and recover the logical ring.

Figure 7 shows the sequential process for the logical ring recovery in the case that the node #3 dropped out from the logical ring of five nodes. The sequential process is as follows:

- a) The node #1 passes the token to the node #2,
- b) The node #2 passes the token to the node #3,
- c) Since the node #3 has not responded, a token frame shall be sent out from the node #4 to the node #5 after the TW-timer and the RMT out of the node #4,
- d) The node #5 passes the token to the node #1,
- e) The node #1 passes the token to the node #2,
- f) Since the node #3 has dropped out of the logical ring, the node #2 shall pass the token to the node #4,
- g) The token is passed to the node #5 from the node #4. The token circulates on new logical ring of node #1, #2, #4 and #5.

NOTE When the nodes receive a token frame from the node which is identified not as a member node in the logical ring and then is identified as a new node, the TW value for the node is set to the maximum time value of 255 ms.

When the nodes receive a token frame from the node which is once a member node of the logical ring, the TW value for the node is set to the same value which is used and set to as the TW value of the node at that time.

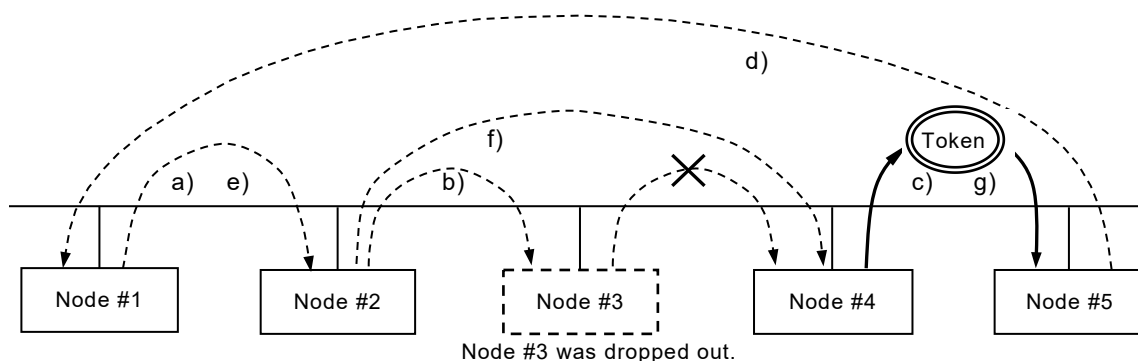


Figure 7 – Logical ring recovery

4.3.2.4 Multiple tokens

There are two types of conditions for a node to handle the problem of multiple tokens:

- a) When a node is holding the token and receives a token frame of which the DNA number is not identical to the node number and is smaller than that of the node, the node shall abandon the right to send out the data; shall give priority to the node with the DNA number and shall enter into the waiting for the token state;
- b) On the other hand, when a node with the token receives a token frame with the DNA number which addresses to other node and is not smaller than that of the node, the node shall discard the token frame received and shall maintain the state in which the node is holding the token and is holding the right to send out the data.

4.3.3 Node addition

4.3.3.1 Overview

A new node in the entering state into a logical ring shall monitor the activity on the Type 26 fieldbus network until the joining-token-detection-time timer (TDT-timer) is timed out. The default observational time period is 3 000 ms and the value is set to the TDT-timer.

There are two kinds of conditions to start the process for node addition; one is under the condition in the network start-up and the other is under the condition in the in-ring start-up state:

- a) During the time period when the node does not detect any token frames, the node recognizes and determines that the logical ring over the Type 26 fieldbus network is not yet formed and is in the network start-up state;
- b) On the other hand, when the node detects any token frames, the node determines that the logical ring over the network has already formed and is in operation, and then the node shall enter into the in-ring start-up state.

4.3.3.2 Node addition in the network start-up state

This process and the procedures are identical to form a new logical ring or to start up a new Type 26 fieldbus network.

There are two types of conditions for a new node to enter into this process; i.e. another node is simultaneously entering and the new node alone enters into this process under the condition in the network start-up state.

Figure 8 shows an example of the sequential timing diagram in the case that a node enters into this process simultaneously with another node under the condition in the network start-up state. Figure 9 shows the sequential timing diagram in the case that a node is alone and enters into this process under the condition in the network start-up state.

The process and the procedures are as follows:

- a) A new node shall start the Trigger-frame-transmission-waiting-time timer (TrWT-timer) after the TDT-timer out without any token frames received. In case of no reception of any trigger frames from another node until the TrWT-timer out, a trigger frame shall be sent out of the new node as shown in Figure 9;
- b) On the contrary, the new node has received any trigger frames from another node during the time period of the TrWT. In this case the trigger frame shall not be sent out of the new node, and the new node shall start both of the Participation-request-frame-transmission-waiting-time timer (PWT-timer) and the participation-request-frame-acceptance-time timer (PAT-timer) upon reception of a trigger frame from another node as shown in Figure 8;
- c) After the TrWT-timer out, the new node shall start both of the Participation-request-frame-transmission-waiting-time timer (PWT-timer) and the participation-request-frame-acceptance-time timer (PAT-timer) in case of no reception of any trigger frames during the time period of the TrWT as shown in Figure 9;
- d) Each node has different values of the time period of the TrWT and the PWT, and the values are set as follows:
 - $\text{TrWT} = R \times 4 \text{ (ms)}$;
R is the remainder of the node number divided by 8;
 - $\text{PWT} = (\text{the node number}) \times 4 \text{ (ms)}$.

The new node shall send out a participation-request frame upon the PWT-timer out;

- e) The new node shall wait for reception of any participation-request frames from another node until the PAT-timer out. The default observational time period is 1 200 ms and the value is set to the PAT-timer;

- f) During the PAT time period upon reception of any participation-request frames, each node shall check and validate the contents of receiving participation-request frames; i.e. shall check the overlap of each common-memory-area claiming for its own use by each node and the duplication of the node number of other nodes; and further shall enrol the sender node related information with the validated common-memory area information and so on into the participating-node-management information table. The participating-node-management-information is updated upon reception of any participation-request frames;
- g) Upon the PWT-timer out, the new node shall send out the participation-request frame;
- h) Upon the PAT-timer out, according to the contents of the participating-node-management-information table new logical ring shall be formed; and then the node with the minimum node number shall send out a token frame first as shown in Figure 8;
- i) On the contrary, in the case of no reception from other nodes until the PAT-timer out, the TDT-timer shall be started again and the above mentioned process and procedures shall be followed and repeated until any trigger frames, any participation-request frames or any token frames from other nodes is received as shown in Figure 9. After repeating three cycle times or sending out the participation-request frame three times, upon the PAT-timer out, the node shall recognize and determine that only the node exists and attempts alone to form a new logical ring; and then shall notify the FAL-user of the node status with a flag of waiting for any receptions from other nodes;
- j) The node which detects and identifies the overlap of the node number with other nodes by means of validating the participation-request frame received from another node shall not enter and participate in the new logical ring; and shall not send out any data and shall discard any receiving frames. Error information with the duplicated node number shall be notified to the FAL-user;
- k) The node which detects and identifies the overlap of the common-memory area with other nodes by means of validating the participation-request frame received from another node shall enter and participate in the new logical ring; shall set the common-memory area unavailable: i.e. set to zero (0) the head-address and the size of the common memory area for own use of the node; and shall not send out any cyclic-data. Error information with the overlap of the common-memory area shall be notified to the FAL-user; the validity of the contents of the common-memory shall be flagged as invalid and unavailable.

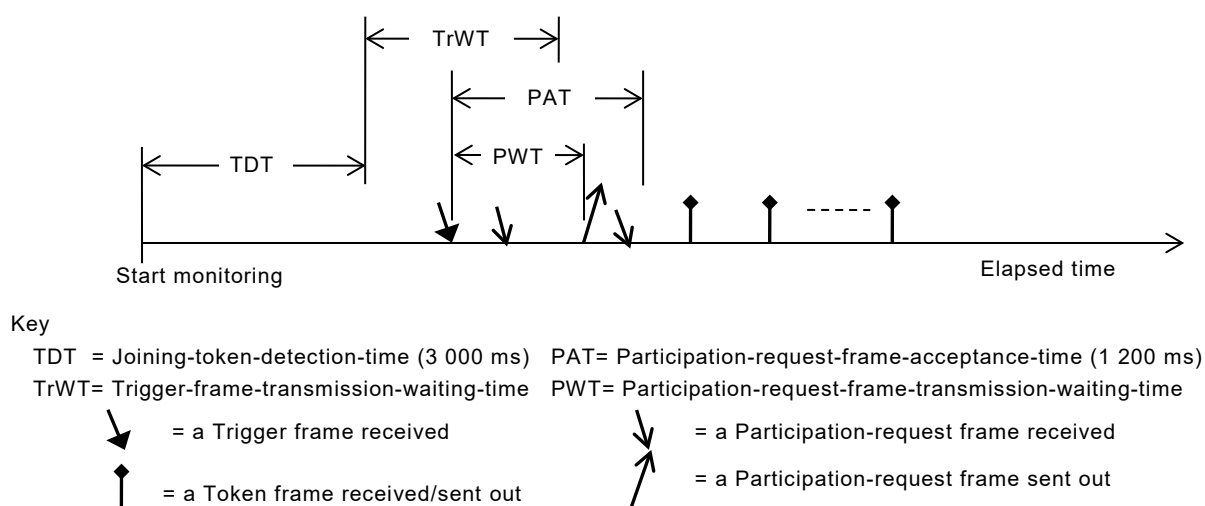


Figure 8 – An example in case of start simultaneously with another node

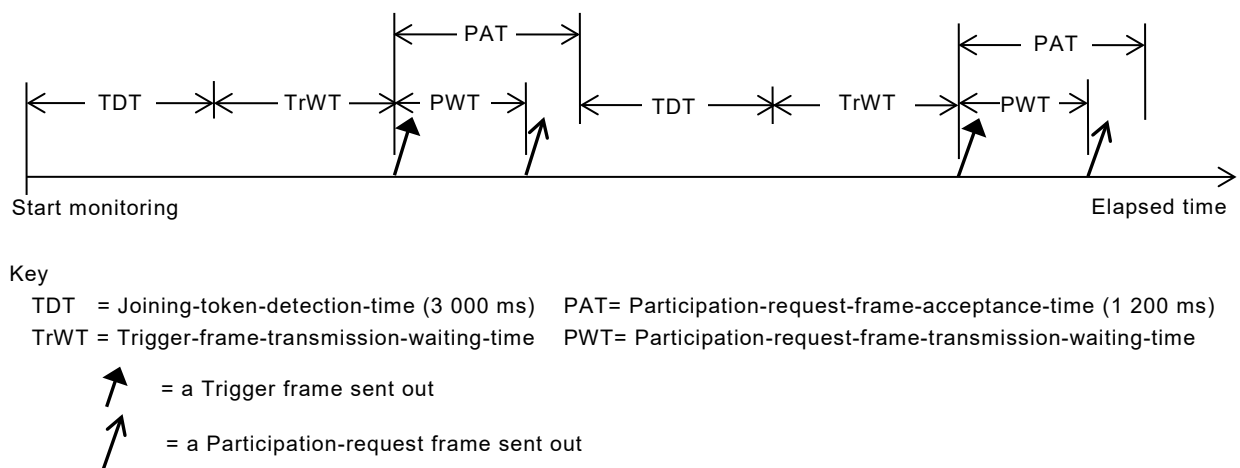


Figure 9 – Start alone case

4.3.3.3 New node addition in the in-ring star-up state

A new node in the entering state into a logical ring shall monitor the activity on the Type 26 fieldbus network until the TDT-timer out. During the TDT time period when the node detects any token frames, the node determines that the logical ring over the network has already formed and is in operation, and then the node enters into the in-ring start-up state.

Figure 10 shows an example of the sequential timing diagram for a new node of node #5 in the in-ring start-up state into 4-node logical ring in operation.

The process and the procedures are as follows:

- The node shall keep the monitoring until the token circulates on the logical ring three times after receiving the token frame with the minimum node number in the logical ring;
- During the three-lap time period of the token circulation on the logical ring, the node shall check and detect the overlap of the common-memory-area claiming for its own use with the area used by other nodes and the duplication of the node number to other nodes by means of the validation of any receiving frames containing these information; and after the validation of any receiving frames, the node shall enrol the sender node related information with the validated common-memory area information and so on into the participating-node-management information table. The participating-node-management-information is updated during this time period;
- After the three-lap time period of the token circulation (3CWT time period) with no duplicated node number, the node shall start the PWT-timer to send out a participation-request frame to enter into the logical ring;
- Upon the PWT-timer out, the node shall send out a participation-request frame even though the node is not holding the token;
- The node which detects and identifies the duplication of the node number to other nodes shall not send out any participation-request frames; and shall not enter into the logical ring. The node shall notify the FAL-user of the error information with the duplicated node number and the node-status with a flag of node number duplication;
- The node which detects and identifies the overlap of the common-memory area with other nodes shall enter and participate in the new logical ring; shall set the common-memory area unavailable: i.e. set to zero (0) the head-address and the size of the common memory area for own use of the node; and shall not send out any cyclic-data. Error information with the overlap of the common-memory area shall be notified to the FAL-user; and the validity of the contents of the common-memory shall be flagged as invalid and unavailable;
- After sending out a participation-request frame, in case of no reception of any token frames addressed to this node until the 3CWT-timer out, the node shall attempt and follow again the above mentioned process and the procedures to enter into the logical ring;

- h) After sending out a participation-request frame, in case of no reception of any token frames until the 3CWT-timer out, the node shall attempt and follow again the above mentioned process and the procedures to enter into the logical ring;
- i) When a node on a logical ring receives any participation-request frames, the node shall update and enrol the receiving node related information into the participating-node-management-information after validation without the overlapping of the common-memory address space and further the duplication of the node number. Detected error information shall be notified to the FAL-user.

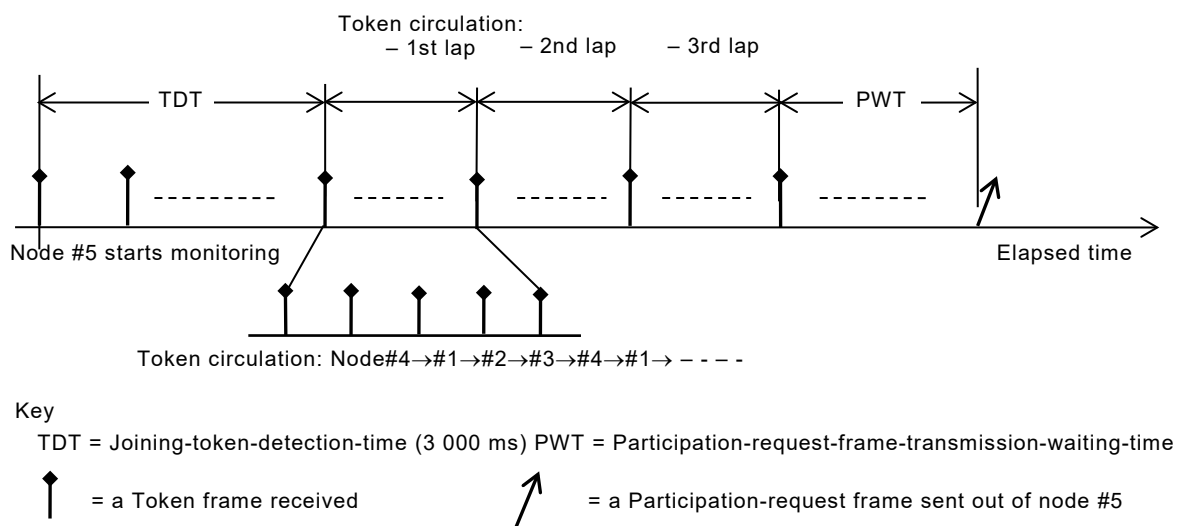


Figure 10 – Node addition: in-ring start-up state

4.3.4 Node in a logical ring

When a node on a logical ring in normal, steady state receives any participation-request frames and/or any final cyclic-data frames which contain the fragmented cyclic-data and are sent out last by the nodes, the node shall update and enrol the receiving node related information into the participating-node-management-information even though the sender node number is not identified after validation without the overlapping of the common-memory address space and further the duplication of the node number. Detected error information shall be notified to the FAL-user.

4.3.5 Node drop-out

Node drop-out or node exiting out of the logical ring is as follows:

- a) Each node on a logical ring in normal, steady state shall check the sender node number of any token frames every time on the reception; and shall detect whether no reception of the token frame from a certain node continues for three consecutive token circulation periods; and shall recognize and determine the corresponding certain node is removed and dropped out of the logical ring;
- b) When the node determines the corresponding certain node is dropped out of the logical ring, the node shall update the corresponding certain node related information within the participating-node-management information to change the status as the corresponding certain node removed.
- c) When the node recognizes and determines all other nodes are removed during the token circulation periods and only the node itself exists, the node shall move into the initial state and take the procedures specified in 4.3.3.2 and 4.3.3.3.
- d) Each node on a logical ring in normal, steady state shall monitor and check any token frames addressed to the node. In the case that no reception of the token frame addressed

to the node from other nodes continues for three consecutive token circulation periods, the node shall recognize and determine the node is dropped out of the logical ring. The node shall move into the initial state and take the procedures specified in 4.3.3.2 and 4.3.3.3.

4.3.6 Data transmission

4.3.6.1 Overview

Operation in normal, steady state on the node consists of a data transmission phase and a token transmission phase.

In the data transmission phase, two kinds of data transmissions are performed:

- Cyclic-data transmission;
- Message-data transmission.

4.3.6.2 Cyclic-data transmission

4.3.6.2.1 Overview

The Type 26 fieldbus is applicable to a controller level network for efficiently executing the control and state data exchange between various kinds of field devices in industrial automation control systems.

The communications between field devices requires simplicity in actual application programming as well as the time-critical transfer of the control and state data among field devices. To meet such a requirement, the CM provides a function to share the control and state data seamlessly across an industrial control network system.

The CM is a virtual common memory accessible with logically unique address across the Type 26 nodes, and the APs running over each node can handle the CM as a global common memory shared among the nodes over the Type 26 fieldbus network.

Cyclic-data transmission is used for refreshing and equalizing the contents of the CM among the nodes by means of a cyclic broadcasting performed at a constant rate after a node writes a data piece on a pre-allocated memory area of the CM for each node.

The data distribution/refreshing function to all nodes in temporal and spatial coherency provided with the CM by means of the cyclic-data transmission performed at a constant rate enables each node to share the same data with each other and to make good use of the data; i.e. a memory space allocated to each node is a container for application data in general use and provides flexibility to apply in a variety of industrial application processes.

Figure 11 shows the data sharing with the CM.

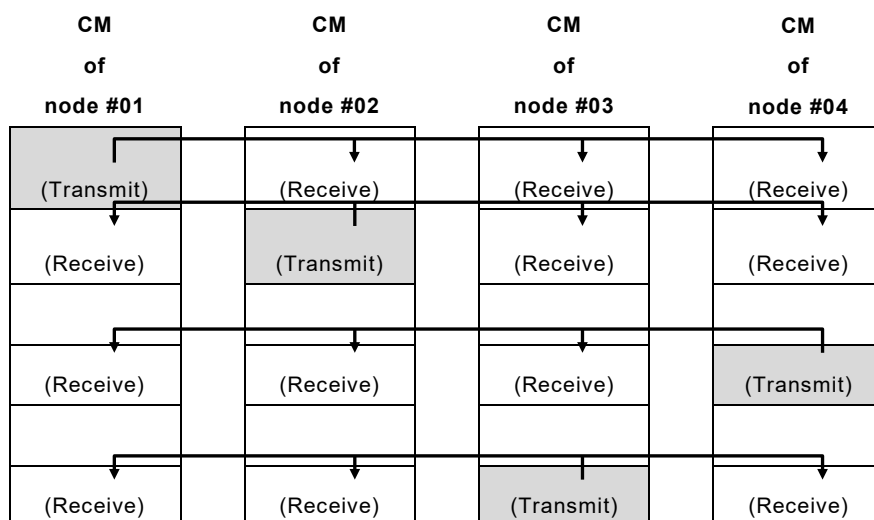


Figure 11 – Data sharing with the CM

4.3.6.2.2 Configuration of Common-memory

Figure 12 shows the configuration of the CM which is composed of two memory areas; i.e. the memory-area-1 and -2 with the memory size of 512 words and 8 192 words respectively. The total memory size of the CM is up to 8 704 words.

The memory area allocated to each node as the output/write area on one or both of the memory-area-1 and -2 is dependent on the implementation. Any memory size in word boundary is allocable from 0 (zero) to the maximum space size of the memory-area-1 and/or -2 to each node, and the memory space allocation is designated with the memory size and the head-address. The memory space allocated to each node shall not overlap with each other. And the allocated memory space is identical to the cyclic-data transmission area by the node.

In case of total zero memory space allocated to a node, it means no cyclic-data transmission to be carried out by the node. It is applicable to a mixture system of the nodes with and without the CM.

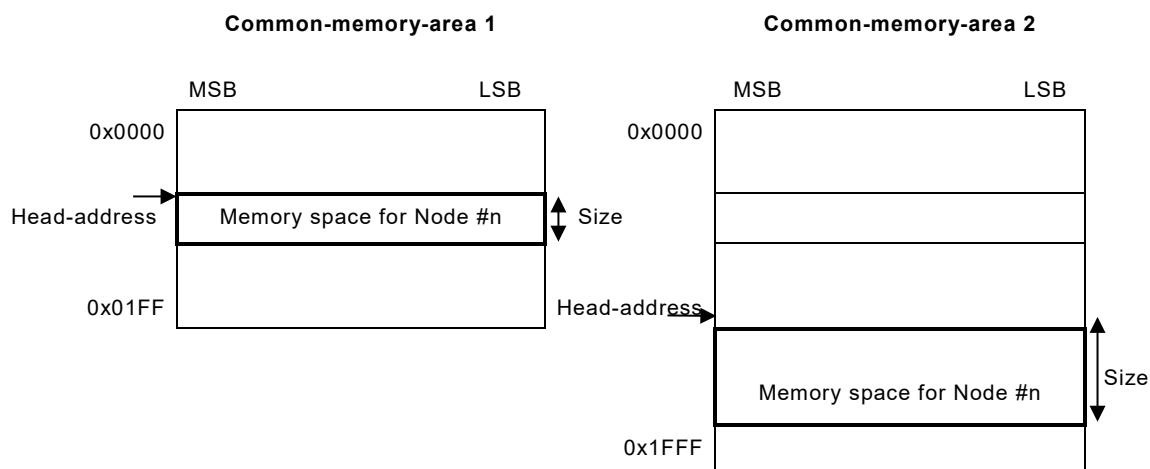


Figure 12 – Configuration of the Common-memory

4.3.6.2.3 Cyclic-data transfer

Cyclic-data is transferred by means of the cyclic-data transmission.

The memory area on the CM allocated to a certain node is used as the transmission area of the node by the cyclic-data transmission as a sender. On the other hand as receiving nodes of the cyclic-data transmission, the memory area of the sender node is used as a receiving area for the data from the sender node.

Cyclic-data transmission shall be performed each time the node obtains the token, and all the data on the CM area allocated to the node shall be sent out of the node.

Dependent on the total amount of data to be transferred, one or more cyclic-data frames are sent out each time the node obtains the token. In the case that the total amount of data is less than or equal to 1 024 octets, data resided in the memory space allocated for the node on the CM-area-1 and -2 shall be packed into a single cyclic-data frame and shall be sent out at a time. On the other hand in the case that the total amount of data is over 1 024 octets, the data shall be fragmented into multiple packets and shall be sent out by means of consecutive multiple cyclic-data frames while holding the token.

The total amount of data over 1 024 octets which resides in the memory space riding both on the CM-area-1 and -2 shall be fragmented into two or more packets for multiple cyclic-data frames to be sent out. The first cyclic-data frame shall contain all the data resided in the memory space on the CM-area-1 and all or part of the data resided in the memory space on the CM-area-2 but the total volume of data contained in the cyclic-data frame shall not exceed 1 024 octets. The following cyclic-data frames shall be sent out with the data by 1 024 octets until all the data is packed into multiple cyclic-data frames. The last cyclic-data frame, so-called "final-cyclic-data frame" shall contain the data of 1 024 octets or less size.

Figure 13 shows an example of the APDUs for multiple cyclic-data frames containing fragmented data. In this case since the total amount of data is over 1 024 octets, the data are fragmented and packed into three cyclic-data frames.

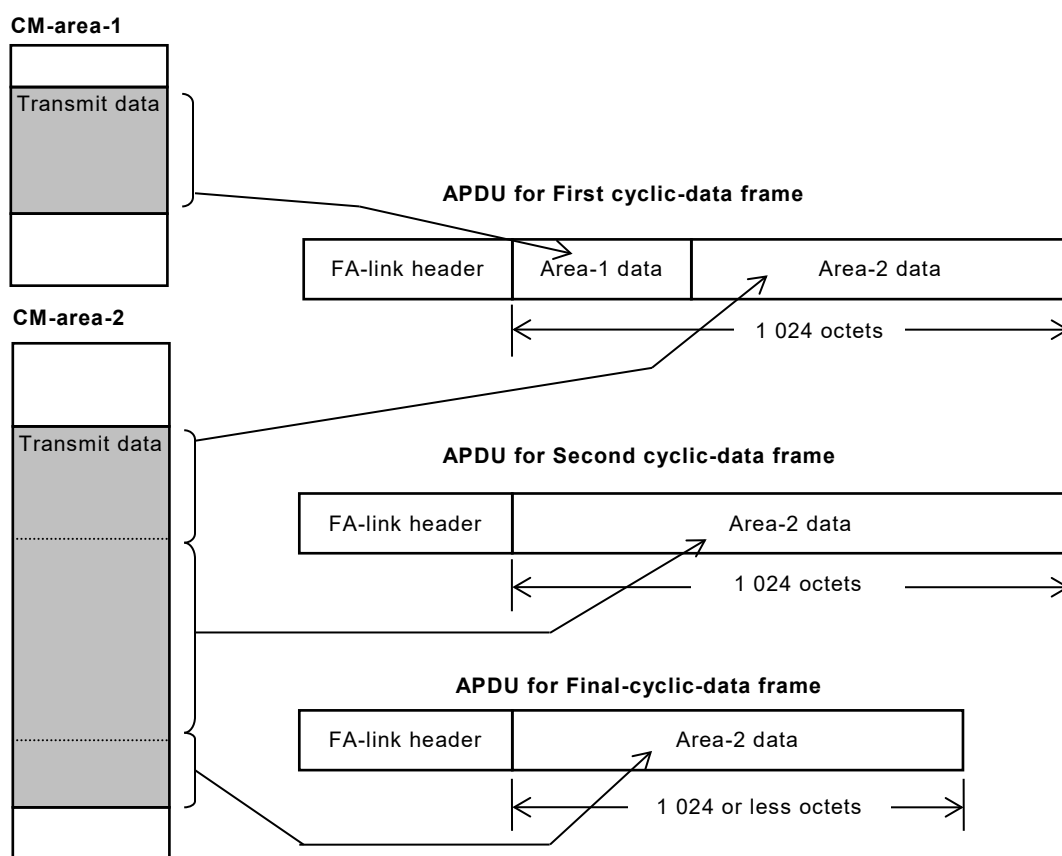


Figure 13 – APDUs of cyclic-data frames containing fragmented data

Each cyclic-data frame shall contain the information in the FA-link header to disassemble/assemble the cyclic-data APDU as follows:

- TBN: total-fragment-block-number;
- CBN: current-fragment-block-number;
- BSIZE: current-fragment-block-size;
- C_AD1: data-head-address on common-memory-area-1;
- C_AD2: data-head-address on common-memory-area-2;
- C_SZ1: data-size on common-memory-area-1;
- C_SZ2: data-size on common-memory-area-2.

Each node receiving a cyclic-data frame shall check the value of TBN in the FA-link header and determine whether the cyclic-data frame is fragmented or not. In the case of the cyclic-data frame fragmented, the receiving node shall store each received block into the receive buffer one by one upon reception of the fragmented cyclic-data frames; and after receiving the final-cyclic-data frame without error, all data received from other nodes shall be written into the corresponding memory address space on the common-memory-area-1 and -2.

In the case that each node fails to receive a series of the cyclic-data frames from other nodes without error, the node shall discard all the received data stored in the receive buffer without writing into the corresponding memory address space on the common-memory-area-1 and -2.

4.3.6.3 Message-data transmission

4.3.6.3.1 Overview

Message-data transmission is used by the message-data transfer service which is based on aperiodic data transfer and is sporadically occurred upon FAL-user request to transfer one or more messages to other nodes. Message-data transfer is performed by means of message-data transmission on the communication channel provided by the DL service of UDP (see RFC 768) or TCP (see RFC 793) as the lower layer of Type 26 FAL.

NOTE Message-data transmission in general is performed by means of connectionless UDP communication channels. The message-data transmission based on connection-oriented TCP channels is only for the General-purpose-command-server-communication which is utilized for intercommunication between the nodes and the general-purpose Human-Machine-Interface tool.

4.3.6.3.2 Message-data transfer

Message-data is transferred by means of the message-data transmission.

Message -data transmission shall be performed as follows:

- a) When a node obtains the token and the message-data transfer is requested by the FAL-user, the node shall send out at a maximum one message-data frame before sending out the cyclic-data frames;
- b) The maximum size of the message-data contained in one message-data frame shall be restricted to 1 024 octets excluding the FA-link header;
- c) Two types of the message-data transmission are applicable: i.e. one-to-one transfer type and one-to-N broadcasting type. The destination node number (DNA) contained in the message-data frame of one-to-one type message-data transmission shall be specified from 1 to 254 as the receiver node number, on the other hand the DNA shall be set to 255 for one-to-N broadcasting type message-data transmission; and the node receiving the message-data frame shall check the received DNA and determines whether the message-data frame is one-to-one type or one-to-N type message-data transmission.

The one-to-one type message-data transmission is based on confirmed type data transmission; on the other hand the one-to-N type message-data transmission is of unconfirmed type.

The one-to-one type message-data transmission provides the functions for automatic retransmission and sequence number management. The automatic retransmission is performed, for instance upon no acknowledgement from the receiver node after sending out a message-data frame addressed to a receiver node. The sequence number management is utilized to check and determine whether a receiving message-data frame has been previously received and is redundant.

Table 3 summarizes the functions available to each type of message-data transmission.

Delivery confirmation and automatic retransmission is performed explicitly by means of the message-data transmission on connectionless UDP channels. On the other hand, delivery confirmation and automatic retransmission by means of the message-data transmission on connection-oriented TCP channels is performed implicitly. The TCP protocol provides these functions.

Table 3 – Available functions to message-data transfer on UDP channel

Message-data transmission	Delivery confirmation	Automatic retransmission	Sequence number management
One-to-one type	Available	Available	Available
One-to-N type	–	–	Available

4.3.6.3.3 Traffic control of message-data transmission

The traffic control function is performed primarily for the message-data transfer on connectionless UDP channels.

When a node obtains the token, the node shall send out at a maximum one message-data frame before sending out the cyclic-data frames and it shall be performed on the basis of the condition of communication load at that time. The condition of the communication load for the node possible to send out one message-data frame is as follows:

- The value of the RMT shall be less than or equal to 90 % value of the RCT, or
- The value of the RMT is greater than 90 % value of the RCT and is less than the value of the RCT and no message-data frame was sent out last time when the node obtained the token.

When the value of the RMT is equal to or greater than the value of the RCT, no message-data frame shall be sent out of the node.

4.3.6.3.4 Delivery confirmation

Delivery confirmation for the message-data transfer shall be performed explicitly by means of a series of the message-data transmission on connectionless UDP channels. On the other hand, delivery confirmation by means of the message-data transmission on connection-oriented TCP communication channels is performed implicitly and the TCP protocol provides the function.

- Delivery confirmation is performed by Acknowledge response from the node receiving a confirm type message-data frame. Acknowledge response is the confirmation responded to the sending node by the receiving node upon reception of a confirm type message-data frame without error. Each node shall provide the function for Acknowledge response.
- The acknowledge response or ACK shall be conveyed by means of cyclic-data transmission; i.e. a cyclic-data frame to be sent out of the receiving node shall contain the ACK to the sending node.
- The cyclic-data frame is sent out of the receiving node when the receiving node obtains the token, so that the cyclic-data frame can convey one or more ACKs which corresponds to the confirm type message-data frames received until the receiving node obtains the token and/or the receiving node sends out the cyclic-data frame with these ACKs.

NOTE The ACKs are conveyed by the final-cyclic-data frame in which at a maximum eight ACKs can be contained.

Figure 14 shows an example of a sequential diagram of exchanging ACKs for message-data transmissions on connectionless UDP channels among node #1, node #2 and node #3.

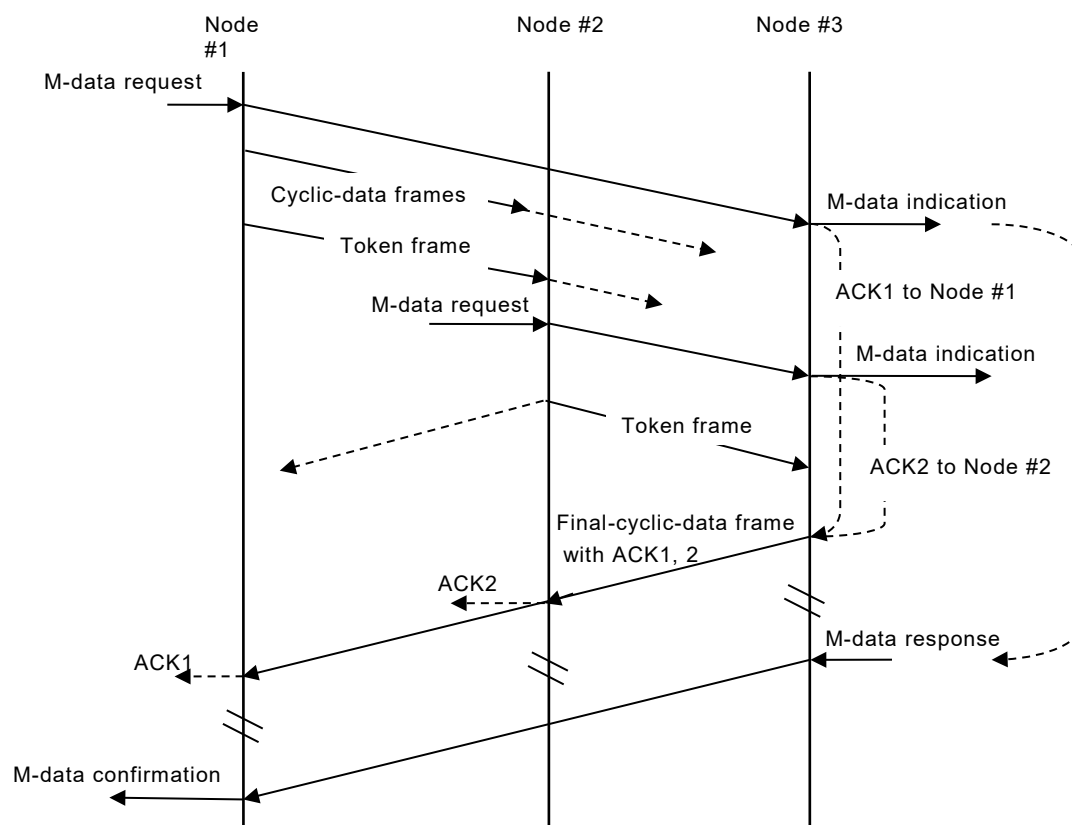


Figure 14 – Example of sequential diagram of ACK over UDP channel

Figure 15 shows the delivery confirmation or ACK sequence for a message-data transmission implicitly checked by TCP protocol on a connection-oriented TCP channel between node #1 and node #2.

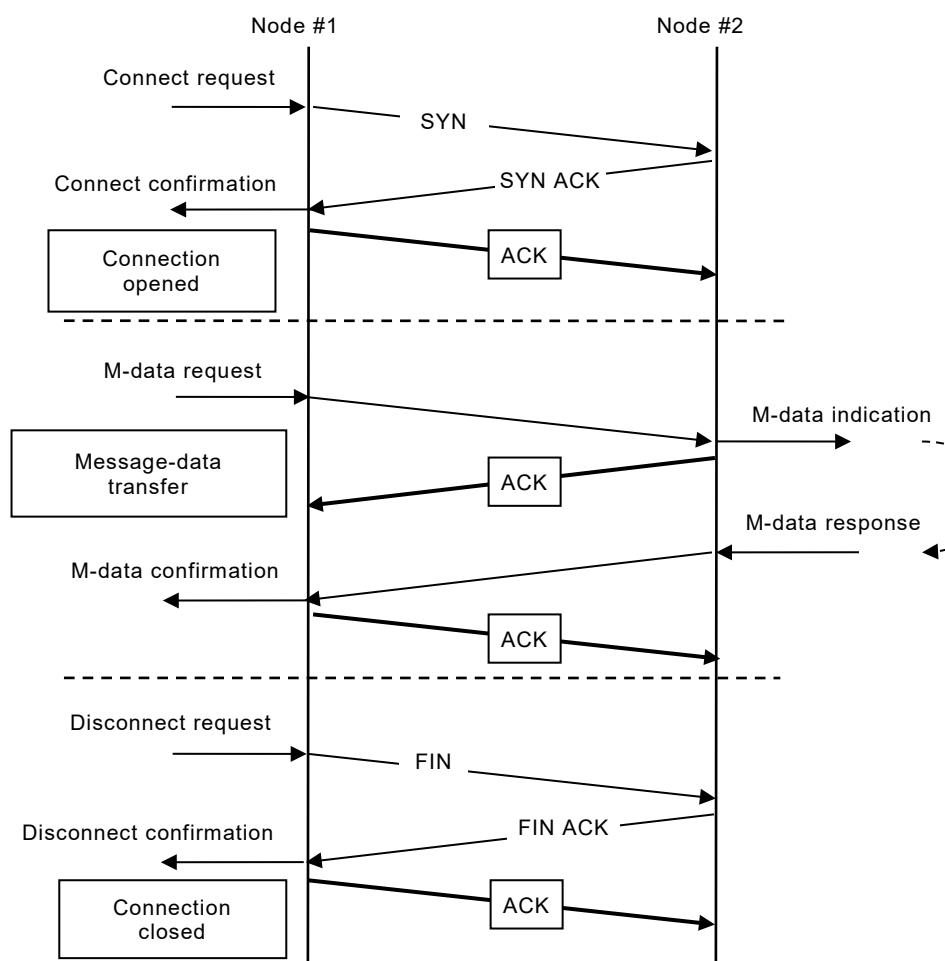


Figure 15 – Delivery confirmation checked by TCP protocol

4.3.6.3.5 Automatic retransmission

The automatic retransmission shall be performed by the nodes upon no ACK from the destination node of a message-data frame sent out and upon receiving ACK with error from the destination node, after sending out a message-data frame to the destination node.

Each node shall be provided with the automatic retransmission function as follows:

- Upon reception of an ACK with error, the node shall perform the retransmission of the message-data transmission;
- Upon the waiting-time-period-for-receiving-message-acknowledge timer (AWT-timer) out under no reception of any ACK from the destination node during the three-lap time period of the token circulation (3CWT time period) to the node after sending out a message-data frame, the node shall perform the retransmission of the message-data transmission;
- Retransmission by the node shall be carried out at a maximum three times;
- Upon no reception of any ACK from the destination node after retransmissions three times, the node shall cancel the message-data transmission and shall notify of the error result with error information to the FAL-user.

4.3.6.3.6 Sequence Number Management

The sequence number management is utilized to check and determine whether a receiving message-data frame has been previously received and is redundant.

The sequence number management by each node is as follows:

- a) The sequence number management by each node shall be performed by means of both of the version of sequence number and the sequence number;
- b) The value of the version of sequence number shall be set at the time when a node enters into the logical ring as a new node and shall be maintained until the node drops out of the logical ring, and is used for other nodes to determine whether a receiving message-data from the node is valid and not overdue;
- c) The value of the version of sequence number shall be a unique number and is supposed to be generated by each node, for instance from current time data, random number data or random data and so forth. However the method to generate the unique number is implementation matter and is intentionally excluded in this document;
- d) The value of the sequence number is used for other nodes to determine whether a receiving message-data from the node has been previously received and is redundant, and shall be set to zero (0) at the time of the network start-up, and shall be increased by one (1) every time on a message-data transmission completed; i.e. the sequence number shall be increased by one upon completion of a message-data transmission with ACK reception without error or completion with retransmission over due to error and then cancelling the message-data transmission; the value of the sequence number shall be a roll-over binary counter value of 32-bit length in the range of 0x1 to 0xFFFF FFFF;
- e) The version of sequence number and the sequence number shall be set in the FA-link header and shall be conveyed by means of the cyclic-data and the message-data transmission to other nodes;
- f) The values of the version of sequence number and the sequence number which are conveyed by a message-data transmission from other nodes shall be put into the corresponding entry area to each node in the message-sequence-number-management-information on one node to node basis upon reception of a message-data frame from a node;
- g) When the value of the version of sequence number in the FA-link header of received message-data frame differs from the value of the corresponding node in the message-sequence-number-management-information, the value in the message-sequence-number-management-information shall be updated to the value in the FA-link header; and the received message-data frame shall be discarded; and ACK with error status of the version of sequence number different shall be sent back to the initiator node or the source node of the message-data frame;
- h) The contents of the message-sequence-number-management-information shall be cleared in advance of each node entering into the logical ring;
- i) When the value of the version of the sequence number of the corresponding node in the message-sequence-number-management-information is zero (0) as its initial value, the receiving message-data frame shall be valid without check; and the value of the version of the sequence number in the FA-link header of a received message-data frame shall be put into the corresponding entry area to that node in the message-sequence-number-management-information;
- j) When the value of the sequence number in the FA-link header of the receiving message-data frame is equal to the value of the sequence number of the corresponding node in the message-sequence-number-management-information, the node receiving the message-data frame shall recognize that the receiving message-data frame is being retransmitted by the corresponding node; and the received message-data is already processed and then the receiving message-data frame shall be discarded; and the ACK without error status shall be sent back to the corresponding node;
- k) When the value of the sequence number in the FA-link header of the receiving message-data frame differs from the value of the sequence number of the corresponding node in the message-sequence-number-management-information, the node receiving the message-data frame shall recognize the receiving message-data as a new one and shall put the value of received sequence number into the corresponding entry area to that node in the message-sequence-number-management-information.

4.3.7 Data transmission frames

4.3.7.1 Data transfer frame

Operation in normal, steady state on the node consists of a data transmission phase and a token transmission phase after a node obtains the token.

In the data transmission phase there exists several patterns of the cyclic-data transmission and the message-data transmission to be carried out by a node as follows:

- a) When a node has no cyclic-data to send out due to no allocation of the memory area specific for this node on the common-memory and has a request to send a message-data from the FAL-user, upon obtaining the token the node first sends out a message-data frame and in succession sends out a cyclic-data frame with no cyclic-data. After sending out the cyclic-data frame without the cyclic-data, then the node sends out a token frame to pass the token to the next node on a logical ring;
- b) When a node has cyclic-data of which data volume is less than or equal to 1 024 octets and has a request to send a message-data from the FAL-user, upon obtaining the token the node first sends out a message-data frame and in succession sends out one cyclic-data frame with the cyclic-data, then the node sends out a token frame to pass the token to the next node on a logical ring;
- c) When a node has cyclic-data of which data volume is greater than 1 024 octets and has a request to send a message-data from the FAL-user, upon obtaining the token the node first sends out a message-data frame and in succession sends out a series of the cyclic-data frames with fragmented cyclic-data. After sending out the final-cyclic-data frame, the node sends out a token frame to pass the token to the next node on a logical ring.

Figure 16 shows the train of data frames and a token frame sending out of a node.

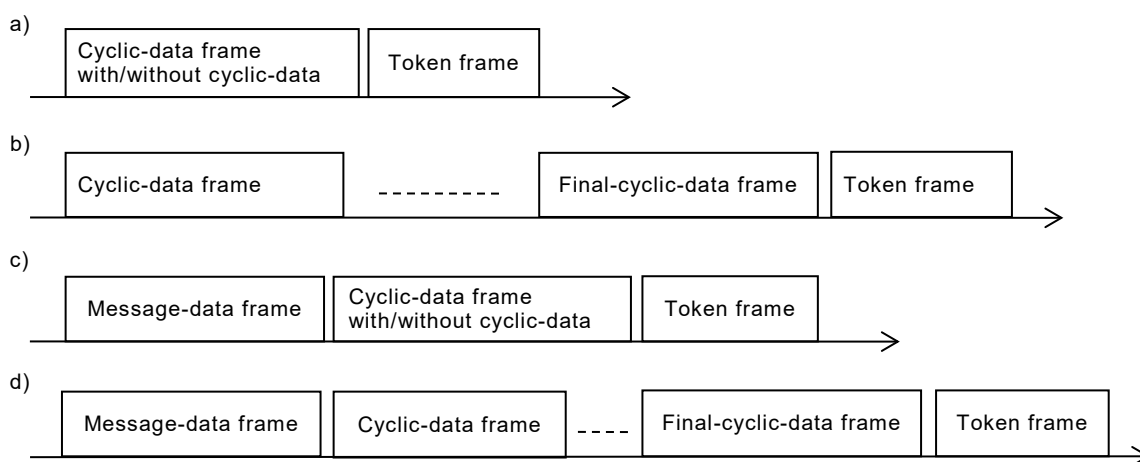
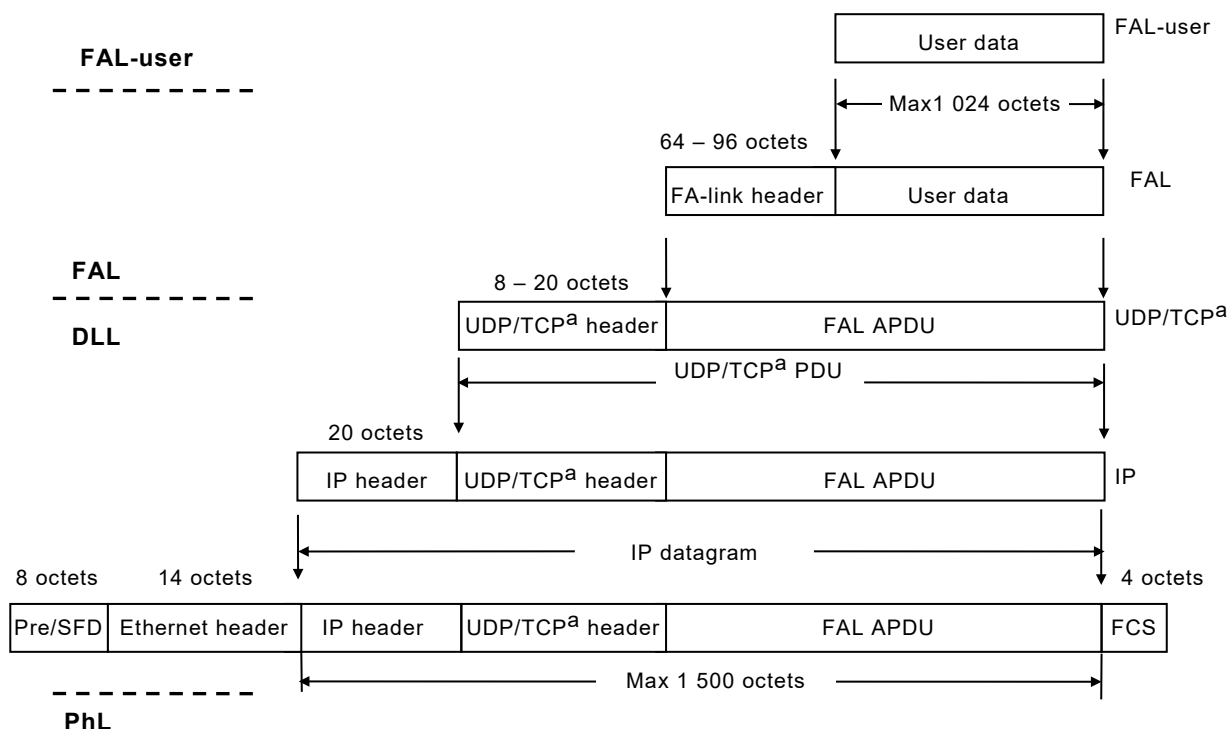


Figure 16 – Train of data frames and a token frame

4.3.7.2 Frame structure

Figure 17 shows the frame structure sent out on Ethernet or ISO/IEC/IEEE 8802-3 medium.



^a Message-data transmission is performed primarily by means of connectionless UDP communication channels. The message-data transmission based on connection-oriented TCP communication channels is only for the General-purpose-command-server-communication which is utilized for intercommunication between the nodes and the general-purpose Human-Machine-Interface tool.

Figure 17 – Frame structure

4.3.7.3 Summary of data transmission frames

Table 4 summarises the data transmission frames including the channel control frames for Type 26 fieldbus.

Transaction code (TCD) is used to discriminate and identify each data transmission frame, and represents the transaction of a receiving frame. Several sets of the values and TCDs are uniquely defined and exclusively reserved for future use; however some of the values reserved are available for Type 26 application users uniquely without overlapping to define for their own application.

Table 4 – Data transmission frame and the TCD value

Data transmission frame	Value of TCD
Transparent-message	10 000 to 59 999 ('2710'H..'EA5F'H')
Token	65 000 ('FDE8'H')
Cyclic-data	65 001 ('FDE9'H')
Participation-request (Participation-req)	65 002 ('FDEA'H')
Byte-block-read request (Byte-block-read-req)	65 003 ('FDEB'H')
Byte-block-read response (Byte-block-read-rsp)	65 203 ('FEB3'H')
Byte-block-write request (Byte-block-write-req)	65 004 ('FDEC'H')
Byte-block-write response (Byte-block-write-rsp)	65 204 ('FEB4'H')
Word-block-read request (Word-block-read-req)	65 005 ('FDED'H')

Data transmission frame	Value of TCD
Word-block-read response (Word-block-read-rsp)	65 205 ('FEB5'H)
Word-block-write request (Word-block-write-req)	65 006 ('FDEE'H)
Word-block-write response (Word-block-write-rsp)	65 206 ('FEB6'H)
Network-parameter-read request (Network-parameter-read-req)	65 007 ('FDEF'H)
Network-parameter-read response (Network-parameter-read-rsp)	65 207 ('FEB7'H)
Network-parameter-write request (Network-parameter-write-req)	65 008 ('FDF0'H)
Network-parameter-write response (Network-parameter-write-rsp)	65 208 ('FEB8'H)
Stop-command request (Stop-command-req)	65 009 ('FDF1'H)
Stop-command response (Stop-command-rsp)	65 209 ('FEB9'H)
Operation-command request (Operation-command-req)	65 010 ('FDF2'H)
Operation-command response (Operation-command-rsp)	65 210 ('FEBA'H)
Profile-read request (Profile-read-req)	65 011 ('FDF3'H)
Profile-read response (Profile-read-rsp)	65 211 ('FEBB'H)
Trigger	65 012 ('FDF4'H)
Log-data-read request (Log-data-read-req)	65 013 ('FDF5'H)
Log-data-read response (Log-data-read-rsp)	65 213 ('FEBD'H)
Log-data-clear request (Log-data-clear-req)	65 014 ('FDF6'H)
Log-data-clear response (Log-data-clear-rsp)	65 214 ('FEBE'H)
Message-return request (Message-return-req)	65 015 ('FDF7'H)
Message-return response (Message-return-rsp)	65 215 ('FEBF'H)
Vendor-specific-message request (Vendor-specific-msg-req)	65 016 ('FDF8'H)
Vendor-specific-message response (Vendor-specific-msg-rsp)	65 216 ('FEC0'H)
Start-TK-holding-time-measurement request (Start-TK-hld-time-mrmt-req)	65 020 ('FDFC'H)
Start-TK-holding-time-measurement response (Start-TK-hld-time-mrmt-rsp)	65 220 ('FEC4'H)
Terminate-TK-holding-time-measurement request (Terminate-TK-hld-time-mrmt-req)	65 021 ('FDFD'H)
Terminate-TK-holding-time-measurement response (Terminate-TK-hld-time-mrmt-rsp)	65 221 ('FEC5'H)
Start-GP_Comm-sender-log request (Start-GP_Comm-sndr-log-req)	65 022 ('FDFE'H)
Start-GP_Comm-sender-log response (Start-GP_Comm-sndr-log-rsp)	65 222 ('FEC6'H)
Terminate-GP_Comm-sender-log request (Terminate-GP_Comm-sndr-log-req)	65 023 ('FDFE'H)
Terminate-GP_Comm-sender-log response (Terminate-GP_Comm-sndr-log-rsp)	65 223 ('FEC7'H)
Set-remote-node-configuration-parameter request (Set-remote-node-config-para- req)	65 024 ('FE00'H)
Set-remote-node-configuration-parameter response (Set-remote-node-config-para- rsp)	65 224 ('FEC8'H)
Read-remote-participating-node-management-information-parameter request (Read-rmt-partici-node-mgt-info-para-req)	65 025 ('FE01'H)
Read-remote-participating-node-management-information-parameter response (Read-rmt-partici-node-mgt-info-para-rsp)	65 225 ('FEC9'H)
Read-remote-node-management-information-parameter request (Read-rmt-node-mgt-info-para-req)	65 026 ('FE02'H)