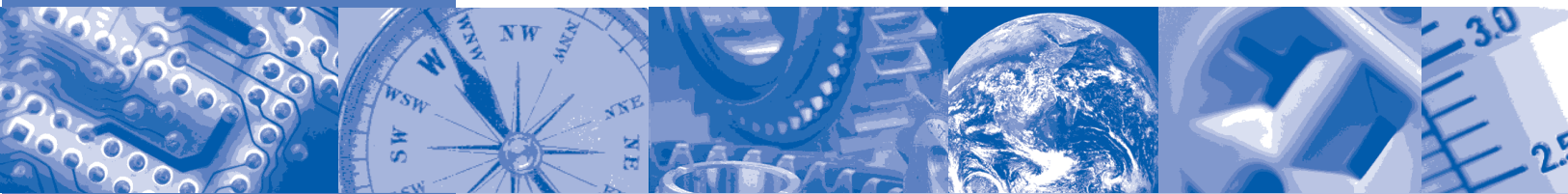


ANSI/ISA–50.02, Part 4–1997

Formerly ANSI/ISA–S50.02, Part 4–1997



Fieldbus Standard for Use in Industrial Control Systems Part 4: Data Link Protocol Specification

Second Printing: January 1999



**ISA–The Instrumentation,
Systems, and
Automation Society**

Approved 6 January 1998



Documents

Document Sharing Hub

ANSI/ISA-50.02, Part 4-1997, Fieldbus Standard for Use in Industrial Control Systems,
Part 4: Data Link Protocol Specification

ISBN: 1-55617-642-2

Copyright © 1997 by the Instrument Society of America. All rights reserved. Not for resale. No part of this publication may be reproduced in any form, including an electronic retrieval system, without the prior written permission of the copyright holders. All requests pertaining to the standard should be submitted to ISA.

ISA
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, North Carolina 27709

Preface

This preface, as well as all footnotes and annexes F-H, is included for informational purposes and is not part of ANSI/ISA-50.02, Part 4-1997.

This Standard has been prepared as part of the service of ISA, the international society for measurement and control, toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms and asks that they be addressed to the Secretary, Standards and Practices Board; ISA; 67 Alexander Drive; P. O. Box 12277; Research Triangle Park, NC 27709; Telephone (919) 549-8411; Fax (919) 549-8288; E-mail: standards@isa.org.

The ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards, recommended practices, and technical reports. The Department is further aware of the benefits to USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, this Department will endeavor to introduce SI-acceptable metric units in all new and revised standards to the greatest extent possible. *The Metric Practice Guide*, which has been published by the Institute of Electrical and Electronics Engineers as ANSI/IEEE Std. 268-1992, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

It is the policy of ISA to encourage and welcome the participation of all concerned individuals and interests in the development of ISA standards. Participation in the ISA standards-making process by an individual in no way constitutes endorsement by the employer of that individual, of ISA, or of any of the standards, recommended practices, and technical reports that ISA develops.

CAUTION – THE USE OF THIS STANDARD MAY INVOLVE HAZARDOUS MATERIALS, OPERATIONS OR EQUIPMENT. THE STANDARD CANNOT ANTICIPATE ALL POSSIBLE APPLICATIONS OR ADDRESS ALL POSSIBLE SAFETY ISSUES ASSOCIATED WITH USE IN HAZARDOUS CONDITIONS. THE USER OF THIS STANDARD MUST EXERCISE SOUND PROFESSIONAL JUDGMENT CONCERNING ITS USE AND APPLICABILITY UNDER THE USER’S PARTICULAR CIRCUMSTANCES. THE USER MUST ALSO CONSIDER THE APPLICABILITY OF ANY GOVERNMENTAL REGULATORY LIMITATIONS AND ESTABLISHED SAFETY AND HEALTH PRACTICES BEFORE IMPLEMENTING THIS STANDARD.

ADDITIONALLY, IMPLEMENTATION OF THE STANDARD MAY REQUIRE USE OF TECHNIQUES, PROCESSES, OR MATERIALS COVERED BY PATENT RIGHTS. ISA TAKES NO POSITION ON THE EXISTENCE OR VALIDITY OF ANY PATENT RIGHTS WHICH MAY BE INVOLVED IN IMPLEMENTING THE STANDARD. ISA WILL NOT BE RESPONSIBLE FOR IDENTIFYING ALL PATENTS THAT MAY REQUIRE A LICENSE BEFORE IMPLEMENTATION OF THE STANDARD OR FOR INVESTIGATING THE VALIDITY OR SCOPE OF ANY PATENTS BROUGHT TO ITS ATTENTION. THE USER SHOULD CAREFULLY INVESTIGATE RELEVANT PATENTS BEFORE USING THE STANDARD FOR THE USER’S INTENDED APPLICATION.

ADDITIONALLY, ISA HAS BEEN INFORMED THAT THIS STANDARD CONTAINS A PATENTED TECHNOLOGY OWNED BY YAMATAKE CORPORATION, TOKYO, JAPAN. YAMATAKE CORPORATION HAS GRANTED NON-EXCLUSIVE, ROYALTY-FREE USE OF THIS PATENT AS LONG AS IT IS USED TO IMPLEMENT THIS PART OF THIS ISA STANDARD. THE DETAILED DOCUMENT IS FILED IN THE ISA OFFICE. ALL OTHER RIGHTS TO THE PATENT ARE RESERVED BY YAMATAKE

CORPORATION. ISA MAKES NO REPRESENTATION AS TO THE REASONABLENESS, VALIDITY, OR APPLICATION OF THE LICENSE OFFERED BY YAMATAKE CORPORATION. FOR FURTHER INFORMATION, PLEASE CONTACT: INTELLECTUAL PROPERTY DEPARTMENT, YAMATAKE CORPORATION, SHIBUYAKU SHIBUYA 2-12-19, TOKYO 150, JAPAN.

The following people served as active contributing members of ISA Subcommittee SP50.3:

NAME	COMPANY
A. Gupta, Chairman	I/O Link, Inc.
T. Phinney, Editor and Principal Contributor	Honeywell, Inc.
H. Dammeyer, Managing Director	Ohio State University
G. Wood, Managing Director	Graeme Wood Consulting
P. Andreassen	FieldNet A/s
W. Borst	Borst Automation
A. Capel	Comgate Engineering Limited
*S. Cavalieri	Universita Degli Studi di Catania
Q. Clark	ISA
R. Crowder	Ship Star Associates, Inc.
L. de Souza	SMAR
B. Dumortier	Schneider Electric
C. Gross	Dow Chemical Company
*David Hewitson	Ronan Engineering Company
T. Ito	Fuji Electric
*V. Jacobson	Ronan Engineering Company
J. Lerare	Cegelec
P. Leviti	Marconi Italiana Automazione SPA
K. Lindner	Endress + Hauser GmbH & Company
*O. Mirabella	Universita Degli Studi di Catania
L. Oreans	Siemens AG
S. Pedersen	ABB Corporate Research
K. Shin	University of Michigan

* One vote per company

The following people served as members of ISA Committee SP50:

NAME	COMPANY
R. Caro, Chairman	Consultant
H. Dammeyer, Managing Director	Ohio State University
G. Wood, Managing Director	Graeme Wood Consulting
C. Akiyama	Yokogawa Electric Corporation
J. Aliperti	Chromatech Consultoria Limited
W. Baker	KML, Inc.
L. Bleckman	HIMA Americas, Inc.
M. Borges	Petrobras Cenpes Supen Diprom
R. Bowden	Fisher-Rosemount Limited
S. Boyer	Iliad Engineering, Inc.
*P. Brett	Honeywell, Inc.
P. Burton	Peter Burton & Associates
R. Canfield	Square D Company
A. Capel	Comgate Engineering, Ltd.

B. Casner
 R. Crowder
 M. Dajani
 E. Delahostria
 U. Doebrich
 J. Dohoney
 B. Dumortier
 *J. Elias
 T. Feindel
 M. Giraudi
 J. Gray
 *P. Griem, Jr.
 B. Gross
 *C. Gross
 A. Gupta
 V. Jacobson
 J. Jamison
 K. Kakizakai
 *R. Ketcham
 T. Kuusisto
 *T. Laine
 H. Landet
 C. Langford
 *R. Lasher
 A. Lesh
 P. Leviti
 K. Lindner
 T. Madden
 A. McCauley, Jr.
 F. McKenna
 R. Meeker
 D. Merriman
 E. Messano
 O. Mirabella
 D. Modell
 A. Mukherji
 *G. Nachev
 L. Natiello
 L. Neitzel
 H. Newman
 *P. Noury
 *E. Pageler
 P. Patel
 *T. Phinney
 F. Russo
 K. Schwarz
 *E. Sederlund
 Y. Shimanuki
 E. Skabowski
 J. Slavinsky
 D. Snizek
 J. Sprague
 R. Squires
 H. Storey
 R. Szanyi
 N. Tobol
 *A. Tresset

Miller Electric Manufacturing Company
 Ship Star Associates, Inc.
 Philip Morris USA
 Allen-Bradley
 Siemens AG
 Jet Propulsion Lab
 Schneider Electric
 Honeywell, Inc.
 R. Stahl, Inc.
 Elsag Bailey
 Krohne America Inc.
 Honeywell, Inc.
 General Motors Corporation
 Dow Chemical Company
 I/O Link, Inc.
 Ronan Engineering Company
 Kenonic Controls Limited
 Fuji Electric Company, Ltd.
 Union Carbide Corporation
 Valmet Automation Inc.
 Cegelec
 Fieldbus International A/s FINT
 Cullen G. Langford Inc.
 Bailey Controls Company
 AMP, Inc.
 Marconi Italiana Automazione
 Endress + Hauser GmbH + Company
 Exxon Company USA
 Chagrin Valley Controls, Inc.
 FMck Associates, Ltd.
 Process Control Solutions, Inc.
 MerTech, Inc.
 Valtek International
 Universita Degli Studi di Catania
 MPI
 SAI
 IsoMatic Laboratories
 Merck & Company Inc.
 Fieldbus Foundation
 Cornell University
 Cegelec
 Fisher²Rosemount Systems Inc.
 V Automation
 Honeywell Inc.
 ENEL-DSR
 Schwarz Communication Consult
 Dow Chemical Company
 Toshiba Corporation
 Tendent Consulting
 Czech Standards Institute
 Lockheed Martin Federal Services
 Saudi Aramco Oil Company
 Measurement Technology Limited
 Shell Oil Company
 Mobil Technology Company
 Ronan Engineering Company
 Cegelec

C. Vaidya
 C. Van Haren
 J. Weiss
 C. Williams
 G. Winchester
 *D. Wroblewski
 *M. Zielinski

Sun Microsystems
 James River Corporation
 Electric Power Research Institute
 Eastman Kodak Company
 National Electrical Manufacturers Association
 Bailey Controls Company
 Fisher•Rosemount Systems Inc.

This published standard was approved for publication by the ISA Standards and Practices Board on August 13, 1997.

NAME	COMPANY
R. Webb, Vice President	Pacific Gas & Electric Company
H. Baumann	H. D. Baumann Associates Limited
D. Bishop	Chevron USA Production Company
P. Brett	Honeywell, Inc.
W. Calder III	Factory Mutual Research Corporation
M. Cohen	Flexonics, Inc.
H. Dammeyer	Ohio State University
R. Dieck	Pratt & Whitney
W. Holland	Southern Company Services, Inc.
H. Hopkins	Utility Products of Arizona
A. Iverson	Ivy Optiks
K. Lindner	Endress + Hauser GmbH & Company
V. Maggioli	Feltronics, Inc.
T. McAviney	Instrumentation & Control Engineering LLC
A. McCauley, Jr.	Chagrin Valley Controls, Inc.
G. McFarland	Honeywell Indus. Auto. & Cont.
E. Montgomery	Fluor Daniel, Inc.
D. Rapley	Rapley Engineering Services Inc.
R. Reimer	Rockwell Automation A-B
J. Rennie	Factory Mutual Research Corporation
W. Weidman	Consultant
J. Weiss	Electric Power Research Institute
J. Whetstone	National Institute of Standards & Technology
M. Widmeyer	Carnegie-Mellon University
H. Wiegler	Canus Corporation
C. Williams	Eastman Kodak Company
G. Wood	Graeme Wood Consulting
M. Zielinski	Fisher•Rosemount Systems Inc.

* One vote per company

Contents

Introduction	19
Keywords	19
General	19
Purpose	19
Intended Users	19
1 Scope	20
1.1 Specifications	20
1.2 Procedures	20
1.3 Applicability	20
1.4 Conformance	21
2 Normative references	21
3 Definitions	22
3.1 Reference model definitions	22
3.2 Service convention definitions	24
3.3 Data Link Service definitions	25
4 Symbols and abbreviations	34
4.1 Data units	34
4.2 Local variables, timers, counters and queues	34
4.3 DLPDU classes	36
4.4 Miscellaneous	37
5 Overview of the DL-protocol	38
5.1 Three-level model of the DLL	38
5.2 Service provided by the DLL	41
5.3 DL-addresses	48
5.4 Service assumed from the Physical Layer	48
5.5 Functions of the DLL	51
5.6 Functional classes	52
5.7 Local parameters, variables, counters, timers and queues	55
6 General structure and encoding of PhIDUs and DLPDUs, and related elements of procedure	69
6.1 PhIDU structure and encoding	69
6.2 Common DLPDU structure, encoding and elements of procedure	73
7 DLPDU-specific structure, encoding and elements of procedure	82
7.1 Establish Connection (EC) DLPDU	84
7.2 Disconnect Connection (DC) DLPDU	86
7.3 Reset Connection (RC) DLPDU	88
7.4 Compel Acknowledgment (CA) DLPDU	90
7.5 Compel Data (CD) DLPDU	97
7.6 Exchange Data (ED) DLPDU	105
7.7 Data (DT) DLPDU	114
7.8 Status Response (SR) DLPDU	122
7.9 Compel Time (CT) DLPDU	125
7.10 Time Distribution (TD) DLPDU	127
7.11 Round-Trip-Delay Query (RQ) DLPDU	128
7.12 Round-Trip-Delay Reply (RR) DLPDU	131
7.13 Probe Node DL-address (PN) DLPDU	133

7.14	Probe Response (PR) DLPDU	135
7.15	Pass Token (PT) DLPDU	137
7.16	Execute Sequence (ES) DLPDU	146
7.17	Return Token (RT) DLPDU	153
7.18	Request Interval (RI) DLPDU	154
7.19	Claim LAS (CL) DLPDU	155
7.20	Transfer LAS (TL) DLPDU	156
7.21	Wakeup (WK) DLPDU	160
7.22	Idle (IDLE) DLPDU	161
7.23	Spare DLPDUs	163
7.24	Reserved (can't use) DLPDUs	164
8	DLPDU-parameter structure and encoding	165
8.1	Structure and encoding of EC-Parameters	165
8.2	Structure and encoding of DC-Parameters	171
8.3	Structure and encoding of RC-Parameters	172
8.4	Structure and encoding of SD-Parameters	174
8.5	Structure and encoding of SR-parameters	181
8.6	Structure and encoding of TD-parameters	182
8.7	Structure and encoding of RQ-parameters	185
8.8	Structure and encoding of RR-parameters	186
8.9	Structure and encoding of PN-parameters	187
8.10	Structure and encoding of DD-parameters	189
9	DL-service elements of procedure	190
9.1	Operation of the DL(SAP)-address, buffer and queue management services ...	190
9.2	Operation of the connection-mode services	194
9.3	Operation of the connectionless-mode services	240
9.4	Operation of the scheduling guidance services	252
10	Other DLE elements of procedure	263
10.1	DLE initialization	263
10.2	LAS behavior and operation	267
10.3	DL-support operation	274
11	Implementation profiles	281
11.1	DL-address, queue and buffer management services	281
11.2	DL-data delivery services	281
11.3	DL-time and time-based scheduling services	283
Annex A — (normative)	Structure and definition of DL-addresses	287
A.1	Form of DL-addresses	287
A.2	Predefined Values and Ranges for DL-Address Components	289
A.3	Predefined DL-Addresses	297
A.4	Representation of DL-Addresses as locally-administered 48-bit MAC-addresses	299
Annex B — (normative)	DL-support sub-protocol	301
B.1	Scope	301
B.2	Overview of LAS operation	301
B.3	DL-support subprotocol definition	302
Annex C — (normative)	DL-bridge elements of procedure and bridge sub-protocol	342
C.1	Global changes	342
C.2	Changes to [IL/1]: Introduction	342
C.3	Changes to [IL/2]: Support of the DL-Service	343
C.4	Changes to [IL/3]: Principles of operation	344

C.5 Changes to [IL/ 4]: The spanning tree algorithm and protocol	349
C.6 Changes to [IL/5]: Encoding of bridge protocol data units	350
C.7 Changes to clause 6: Bridge management	350
C.8 Changes to [IL/7]: Management protocol	351
C.9 Changes to [IL Annex A]: PICS proforma	351
C.10 Delete [IL/Annex C] and [IL/Annex D], both of which relate to source-routing	351
Annex D — (normative) DL-management-information	352
D.1 Scope	352
D.2 DLE configuration parameters	352
D.3 DLE-collected fault-management data	354
Annex E4 — (normative) PICS proforma	357
Introduction	357
E.1 Scope	357
E.2 Normative references	357
E.3 Definitions	358
E.4 Abbreviations	358
E.5 Conformance	358
E.6 Instructions	358
E.7 Identification	360
E.8 Implementation profile	361
E.9 Major low-level capabilities	366
E.10 Major high-level capabilities	382
Annex F — (informative) Formal protocol finite state machines ..	396
F.1 Basic reception and transmission FSMs	396
F.2 FSMs for DLCs	408
F.3 FSMs for scheduling	415
F.4 FSMs for bridges	415
Annex G — (informative) Exemplary FCS implementation	416
Annex H — (informative) DLPDU and DL-addressing short-form summaries	418
H.1 Fields used in short form summaries	418
H.2 DLPDU short-form summary grouped by function	420
H.3 DLPDU short-form summary in alphabetic order of DLPDU names	422
H.4 DLPDU short-form summary in alphabetic order of DLPDU acronyms	423
H.5 DLPDU FC code-point assignment matrix — overview and detail	424
H.6 SD-parameters (status and data-description parameters) of CA, CD, ED and DT DLPDUs	427
H.7 EC parameters of EC DLPDUs	431
H.8 Parameters of DC and RC DLPDUs	432
H.9 Parameters of TD, RQ and RR DLPDUs	433
H.10 Parameters of PN, PT, ES and RI DLPDUs	436
H.11 Addressing summary extracted from Annex A figures and tables	437

Figures

Figure 1 — Relationship of this part of ISA-S50.02 to other Fieldbus and OSI standards.....	19
Figure 2 — Relationships of DLSAPs, DLCEPs, DLSAP-addresses, DLCEP-addresses and group DL-addresses	27
Figure 3 — Linear relationships of sending and receiving DLCEP sequence-number variables	61
Figure 4 — DL-address alternative structures	75
Figure 5 — short DL-address field — alternative implicit structures	76
Figure 6 — node DL-address field — implicit structure.....	76
Figure 7 — State transition diagram for a DLCEP	195
Figure 8 — Projection of the sending and receiving DLCEP sequence-number variables of Figure 3 onto the cyclic sequence-number parameters of CA, CD, DT, ED and RC DLPDUs, with consequent determination of required actions	213
Figure 9 — State transitions of a DLE.....	264
Figure A.1 — Basic Structure of a DL-address	287
Figure A.2 — Basic Structure of a Sub-link Selector	287
Figure A.3 — DL-address Alternative Structures	288
Figure A.4 — Basic Structure of MAC-addresses.....	299
Figure A.5 — Representation of a DL-address as a MAC-address	300
Figure C.1 — Replacement for [IL] Fig 3-2 Bridge ports	345
Figure C.2 — Replacement for [IL] Fig 3-3 Bridge architecture	345
Figure G.1 — Example of FCS generation.....	416
Figure G.2 — Example of FCS syndrome checking on reception.....	417

Figure H.1 — Gross Structure of FC Code Points	424
Figure H.2 [Annex A/1] — Basic Structure of a DL-address	437
Figure H.3 [Annex A/2] — Basic Structure of a Sub-link Selector	437
Figure H.4 [Annex A/3] — DL-Address Alternative Structures.....	437
Figure H.5 [Annex A/4] — Basic Structure of MAC-addresses.....	437
Figure H.6 [Annex A/5] — Representation of a DL-address as a MAC-address	437

Tables

Table 1 — Correlation of DLPDUs with Functional Classes	53
Table 2 — Summary structure of DLPDUs	83
Table 3 — DLPDU restrictions based on dominant token	84
Table 4 — Structure of EC DLPDUs	84
Table 5 — Structure of DC DLPDUs	87
Table 6 — Structure of RC DLPDUs	89
Table 7 — Structure of CA DLPDUs	91
Table 8 — Structure of CD DLPDUs	98
Table 9 — Structure of ED DLPDUs	106
Table 10 — Structure of DT DLPDUs	114
Table 11 — Structure of SR DLPDUs	122
Table 12 — Structure of CT DLPDUs	125
Table 13 — Structure of TD DLPDUs	127
Table 14 — Structure of RQ DLPDUs	129
Table 15 — Structure of RR DLPDUs	131
Table 16 — Structure of PN DLPDUs	134
Table 17 — Structure of PR DLPDUs	135
Table 18 — Structure of PT DLPDUs	137
Table 19 — Structure of ES DLPDUs	146
Table 20 — Structure of RT DLPDUs	153
Table 21 — Structure of RI DLPDUs	154
Table 22 — Structure of CL DLPDUs	155
Table 23 — Structure of TL DLPDUs	157
Table 24 — Structure of WK DLPDUs	160
Table 25 — Structure of Idle DLPDUs	161

Table 26 — Assumed structure of undefined (spare) DLPDUs	163
Table 27 — Assumed structure of Reserved (can't use) DLPDUs	164
Table 28 — Structure of an EC DLPDU's parameters	165
Table 29 — EC-parameters: 1st octet	165
Table 30 — EC-parameters: 2nd octet	166
Table 31 — EC-parameters: 3rd and 4th octets	167
Table 32 — EC-parameters: 5th and 6th octets	167
Table 33 — EC-parameters: 7th octet	168
Table 34 — EC-parameters: 8th octet	168
Table 35 — EC-parameters: 9th and 10th octets	169
Table 36 — EC-parameters: 11th octet	170
Table 37 — EC-parameters: 12th octet	170
Table 38 — EC-parameters: 13th and 14th octets	171
Table 39 — DC-parameters and RC-parameters: 1st octet	171
Table 40 — DC-parameters and RC-parameters: 2nd octet	171
Table 41 — Disconnect reasons	172
Table 42 — Reset reasons	173
Table 43 — RC-parameters: 3rd octet	173
Table 43b — RC-parameters: 4th octet	173
Table 44 — Structure of connectionless-mode CA, CD, DT and ED DLPDUs	175
Table 45 — Short format SD-parameters for connectionless transaction initiators	175
Table 46 — Short format SD-parameters for connectionless responders	176
Table 47 — Reply status for unitdata-acknowledgment and exchange-unitdata-reply DT DLPDUs	177
Table 48 — Structure of connection-oriented CA, CD, DT and ED DLPDUs	179
Table 49 — Short format SD-parameters for DLCEP state	180
Table 50 — Long format SD-parameters for DLCEP state: 1st octet	180
Table 51 — Long format SD-parameters for DLCEP state: 2nd octet	180

Table 52 — Long format SD-parameters for DLCEP state: 3rd octet	180
Table 53 — Reply status for SR DLPDUs	182
Table 54 — Short format SR-parameters	182
Table 55 — Structure of TD-parameters	183
Table 56 — Structure and encoding of the DL-time-quality measures	183
Table 57 — Approximate numeric significance of the bits of seven-octet DL-time	184
Table 58 — Approximate numeric significance of the bits of three-octet short time ...	185
Table 59 — Structure of RQ-parameters	185
Table 60 — Structure of RR-parameters	186
Table 61 — Structure and encoding of the RR-time-quality measures	186
Table 62 — Structure of PN-parameters	187
Table 63 — PN-parameters: 1st octet	188
Table 64 — PN-parameters: 2nd octet	188
Table 65 — PN-parameters: 3rd and 4th octets	188
Table 66 — PN-parameters: 5th octet	189
Table 67 — PN-parameters: 6th octet	189
Table 68 — Structure of DD-parameters	189
Table 69 — Components of returned DL-time	253
Table 70 — Time Synchronization Computation	256
Table 71 — Maximum permitted phase-tracking error in a DLE's sense of DL-time at the minimum requireable Time Distribution period	285
Table A.1 — link node selector addressing	290
Table A.2 — link-local node selector addressing	292
Table A.3 — link-local node designators	294
Table A.4 — node-local selector addressing	296
Table A.5 — predefined flat non-local DL-addresses	297
Table A.6 — predefined flat link-local DL-addresses	298
Table A.7 — predefined node-local DL-addresses	299

Table B.1 — SPDU 1st octet: SPDU class, and protocol version or subclass	303
Table B.2 — Probe-response SPDU	304
Table B.3 — DL-protocol versions supported	304
Table B.4 — PR-SPDU: 3rd and 4th octets	304
Table B.5 — Node-activation SPDU	306
Table B.6 — Node-activation SPDU: 4th octet	306
Table B.7 — LAS-data-base-status SPDU	307
Table B.8 — LAS-data-base-status SPDU: 2nd octet	307
Table B.9 — Live-list - change SPDU	308
Table B.10 — DLE-status structure	309
Table B.11 — Live-list-detail SPDU	309
Table B.12 — DL-conformance-reply SPDU	311
Table B.13 — DL-protocol versions supported	311
Table B.14 — DL-conformance encoding (part 1)	311
Table B.15 — DL-conformance encoding (part 2)	312
Table B.16 — DL-conformance encoding (part 3)	312
Table B.17 — DL-conformance encoding (part 4)	312
Table B.18 — Link-basic-parameters-reply SPDU	313
Table B.19 — Link-master-parameters-reply SPDU	314
Table B.20 — Token-hold-time-request SPDU	315
Table B.21 — Token-hold-time-array SPDU	316
Table B.22 — Sequence element header encoding	318
Table B.23 — short DL-address and duration sequence element	318
Table B.24 — long DL-address and duration sequence element	319
Table B.25 — Wakeup request sequence element	320
Table B.26 — Schedule-request SPDU	321
Table B.27 — Sequence type, schedule type and priority encoding	321

Table B.28 — Scheduling-completed SPDU	323
Table B.29 — Status and reason codes	323
Table B.30 — Cancel-schedule SPDU	324
Table B.31 — Schedule-canceled SPDU	324
Table B.32 — Link-schedule	325
Table B.33 — Schedule-summary SPDU	326
Table B.34 — Sub-schedule-SPDU reference	327
Table B.35 — Sub-schedule SPDU	328
Table B.36 — Sequence Sub-SPDU	329
Table B.37 — Element-description	329
Table B.38 — Schedule-summary-request SPDU	331
Table B.39 — Sub-schedule-request SPDU	331
Table B.40 — Parameter-list element-header encoding	332
Table B.41 — Begin/end-of-list element	332
Table B.42 — Continuation-of-list element	333
Table B.43 — short DL-address list element	333
Table B.44 — long DL-address element	334
Table B.45 — DLSAP-address-characteristics element	335
Table B.46 — DLCEP-characteristics element	335
Table B.47 — Address-query SPDU	336
Table B.48 — Address-report SPDU	337
Table B.49 — Address-list-query SPDU	338
Table B.50 — DL-address selection criteria	339
Table B.51 — Address-list-reply SPDU	340
Table H.1 — Generic Assignment of FC Code Points	425
Table H.2 — Individual Assignment of FC Code Points	426
Table H.3 — Reply status for SR DLPDUs	430

Table H.4 — Reply status for unitdata-acknowledgment and exchange-unitdata-reply DT DLPDUs	430
Table H.5 — Approximate numeric significance of the bits of seven-octet DL-time	435
Table H.6 — Approximate numeric significance of the bits of N(NT), A...A, and three-octet C(NT)	435
Table H.7 [Annex A/1] — link node selector addressing	438
Table H.8 [Annex A/2] — link-local node selector addressing	438
Table H.9 [Annex A/7] — predefined flat non-local DL-addresses	439
Table H.10 [Annex A/6] — predefined flat link-local DL-addresses	439
Table H.11 [Annex A/3] — link-local node designators	440
Table H.12 [Annex A/4] — node-local selector addressing	440
Table H.13 [Annex A/7]— predefined node-local DL-addresses	440

Introduction

Keywords

fieldbus, time-critical communications, time-scheduled communications, automation networks, control networks

General

This part of ISA-S50.02 is one of a set of Standards produced to facilitate the interconnection of automation system components. It is related to other Standards in the set as defined by the Fieldbus Reference Model, which is based in part on the Reference Model for Open Systems Interconnection. Both Reference Models subdivide the area of standardization for interconnection into a series of layers of specification, each of manageable size.

Purpose

The purpose of this part of ISA-S50.02 is to define the Fieldbus Data Link Protocol. It is most closely related to, and lies within the field of application of, the Fieldbus Data Link Service Definition.

The Data Link Protocol provides the Data Link Service by making use of the services available from the Physical Layer. The relationship between the Standards for Fieldbus Data Link Service, Fieldbus Data Link Protocol, Fieldbus Physical Service, and OSI Network or Fieldbus Application Protocol is illustrated in Figure 1.

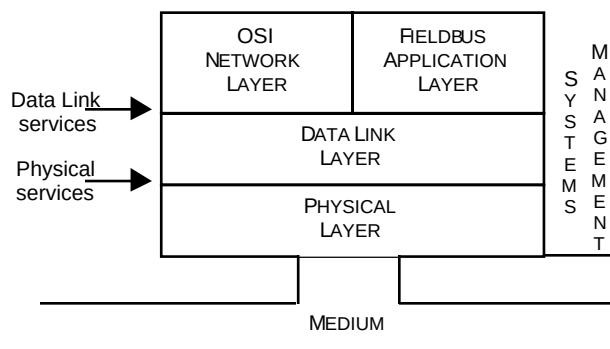


Figure 1 — Relationship of this part of ISA-S50.02 to other Fieldbus and OSI standards

Intended Users

The primary aim of this Standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer Data Link entities 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:

- a) as a guide for implementors and designers;

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

This Standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this Standard, together with other standards positioned within the OSI or Fieldbus Reference Models, otherwise incompatible systems may work together in any combination.

1 Scope

The Fieldbus Data Link protocol specification is a Data Link Layer Standard designed to support time-critical messaging communications to and from devices in an automation environment.

1.1 Specifications

This Standard specifies

- a) procedures for a single protocol for the timely transfer of data and control information from one data-link user entity to a peer user entity, and among the data-link entities forming the distributed data-link service provider; and
- b) the structure of the Fieldbus Data Link (DL) Protocol Data Units used for the transfer of data and control information, and their representation as Physical Interface Data Units.

1.2 Procedures

The procedures are defined in terms of

- a) the interactions between peer DL-entities (DLEs) through the exchange of Fieldbus Data Link Protocol Data Units;
- b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives; and
- c) the interactions between a DLS-provider and a Physical service provider in the same system through the exchange of Physical service primitives.

1.3 Applicability

These procedures are applicable to instances of communication between systems which support time-critical communications services within the Data Link layer of the OSI or Fieldbus Reference Models, and which require the ability to interconnect in an open systems interconnection environment.

Profiles provide a simple multi-attribute means of summarizing an implementation's capabilities, and thus its applicability to various time-critical communications needs. Clause 10 specifies profiles for this Fieldbus protocol.

1.4 Conformance

This Standard also specifies conformance requirements for systems implementing these procedures. This part of ISA-S50.02 does not contain tests to demonstrate compliance with such requirements.

The supplier of a protocol implementation which is claimed to conform to this Standard shall complete a copy of the PICS Proforma (see Annex E), and shall provide the information necessary to identify both the supplier and the implementation.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid Standards.

ISO 7498 : 1984, *Information processing systems — Open systems interconnection — Basic reference model*.

ISO 7498/AD1 : 1987, *Information processing systems — Open systems interconnection — Connectionless data transmission*.

ISO 7498-3 : 1987, *Information processing systems — Open systems interconnection — Naming and addressing*.

ISO/TR 8509 : 1987, *Information processing systems — Open systems interconnection — Service conventions*.

ISO/IEC 10731 : 1992, *Information technology — Open systems interconnection — Conventions for the definition of OSI services*.

ISO/IEC 8886 : 1992, *Information technology — Open systems interconnection — Telecommunications and information exchange between systems — Data link service definition*.

ISO/IEC 9646-1 : 1991, *Information technology — Open systems interconnection — Conformance testing methodology and framework — Part 1: General concepts*.

ISO/IEC 9646-2 : 1991, *Information technology — Open systems interconnection — Conformance testing methodology and framework — Part 2: Abstract test suite specification*.

ISO/IEC TR 18802 : 1993, *Information technology — Telecommunications and information exchange between systems — Overview of Local Area Network Standards*.

ISO/IEC 10038 : 1993, *Information technology — Telecommunications and information exchange between systems — Local area networks — Media access control (MAC) bridges*.

ANSI/ISA S50.02-2 : 1992, *Fieldbus Standard for use in industrial control systems — Part 2 : Physical layer specification and service definition*.

3 Definitions

For the purposes of this part of ISA-S50.02, the following definitions apply.

3.1 Reference model definitions

This part of ISA-S50.02 is based in part on the concepts developed in ISO 7498 parts 1 and 3, and in ISO 7498/AD1, and makes use of the following terms defined therein:

3.1.1	DL-address	[7498-3]
3.1.2	DL-address-mapping	[7498-1]
3.1.3	called-DL-address	[7498-3]
3.1.4	calling-DL-address	[7498-3]
3.1.5	centralized multi-end-point-connection	[7498-1]
3.1.6	DL-connection	[7498/AD1]
3.1.7	DL-connection-end-point	[7498-1]
3.1.8	DL-connection-end-point-identifier	[7498-3]
3.1.9	DL-connection-mode transmission	[7498/AD1]
3.1.10	DL-connectionless-mode transmission	[7498/AD1]
3.1.11	correspondent (N)-entities correspondent DL-entities correspondent Ph-entities	[7498-1]
3.1.12	DL-data-link	[7498-1]
3.1.13	DL-data-source	[7498-1]
3.1.14	demultiplexing	[7498-1]
3.1.15	DL-duplex-transmission	[7498-1]
3.1.16	(N)-entity DL-entity Ph-entity	[7498-1]
3.1.17	flow control	[7498-1]

3.1.18	Ph-interface-control-information	[7498-1]
3.1.19	Ph-interface-data	[7498-1]
3.1.20	Ph-interface-data-unit	[7498-1]
3.1.21	(N)-layer	[7498-1]
	DL-layer (N=2)	
	Ph-layer (N=1)	
3.1.22	layer-management	[7498-1]
3.1.23	multiplexing	[7498-1]
3.1.24	DL-name	[7498-3]
3.1.25	naming-(addressing)-authority	[7498-3]
3.1.26	naming-(addressing)-domain	[7498-3]
3.1.27	naming-(addressing)-subdomain	[7498-3]
3.1.28	peer-entities	[7498-1]
3.1.29	primitive name	[7498-3]
3.1.30	DL-protocol	[7498-1]
3.1.31	DL-protocol-connection-identifier	[7498-1]
3.1.32	DL-protocol-control-information	[7498-1]
3.1.33	DL-protocol-data-unit	[7498-1]
3.1.34	DL-protocol-version-identifier	[7498-1]
3.1.35	reassembling	[7498-1]
3.1.36	recombining	[7498-1]
3.1.37	DL-relay	[7498-1]
3.1.38	reset	[7498-1]
3.1.39	responding-DL-address	[7498-3]
3.1.40	routing	[7498-1]
3.1.41	segmenting	[7498-1]
3.1.42	sequencing	[7498-1]

3.1.43 (N)-service DL-service Ph-service	[7498-1]
3.1.44 (N)-service-access-point DL-service-access-point Ph-service-access-point	[7498-1]
3.1.45 (N)-service-access-point-address DL-service-access-point-address Ph-service-access-point-address	[7498-3]
3.1.46 DL-service-connection-identifier	[7498-1]
3.1.47 DL-service-data-unit	[7498-1]
3.1.48 DL-simplex-transmission	[7498-1]
3.1.49 splitting	[7498-1]
3.1.50 synonymous name	[7498-3]
3.1.51 systems-management	[7498-1]
3.1.52 DL-user-data	[7498-1]

3.2 Service convention definitions

This part of ISA-S50.02 also makes use of the following terms defined in ISO/TR 8509 and ISO/IEC 10731 as they apply to the Data Link Layer:

- 3.2.1** acceptor
- 3.2.2** asymmetrical service
- 3.2.3** confirm (primitive);
requestor.deliver (primitive)
- 3.2.4** deliver (primitive)
- 3.2.5** DL-confirmed-facility
- 3.2.6** DL-facility
- 3.2.7** DL-local-view
- 3.2.8** DL-mandatory-facility
- 3.2.9** DL-non-confirmed-facility
- 3.2.10** DL-provider-initiated-facility
- 3.2.11** DL-provider-optional-facility

- 3.2.12 DL-service-primitive;
primitive
- 3.2.13 DL-service-provider
- 3.2.14 DL-service-user
- 3.2.15 DL-user-optional-facility
- 3.2.16 indication (primitive)
acceptor.deliver (primitive)
- 3.2.17 multi-peer
- 3.2.18 request (primitive);
requestor.submit (primitive)
- 3.2.19 requestor
- 3.2.20 response (primitive);
acceptor.submit (primitive)
- 3.2.21 submit (primitive)
- 3.2.22 symmetrical service

3.3 Data Link Service definitions

For the purpose of this part of ISA-S50.02, the following definitions also apply:

3.3.1 link, local link: A single DL-subnetwork in which any of the connected DLEs may communicate directly, without any intervening DL-relaying, whenever all of those DLEs which are participating in an instance of communication are simultaneously attentive to the DL-subnetwork during the period(s) of attempted communication.

3.3.2 extended link: A DL-subnetwork, consisting of the maximal set of links interconnected by DL-relays, sharing a single DL-name (DL-address) space, in which any of the connected DL-entities may communicate, one with another, either directly or with the assistance of one or more of those intervening DL-relay entities.

NOTE — An extended link may be comprised of just a single link.

3.3.3 node: A single DL-entity as it appears on one local link.

3.3.4 bridge: A DL-relay entity which performs selective store-and-forward, routing, scheduling and DL-time-distribution functions to

- a) connect two or more separate DL-subnetworks (links) to form a unified DL-subnetwork (the extended link);
- b) provide a means by which two end systems can communicate, when at least one of the end systems is periodically inattentive to the interconnecting DL-subnetwork; and
- c) connect the extended link to a separate time-distribution network (for example, a national radio time-distribution network).

3.3.5 sending DLS-user: A DL-service user that acts as a source of DL-user-data.

3.3.6 receiving DLS-user: A DL-service user that acts as a recipient of DL-user-data.

NOTE — A DL-service user can be both a sending and receiving DLS-user concurrently.

3.3.7 peer DLC A point-to-point DL-connection offering DL-duplex-transmission between two peer DLS-users where each can be a sending DLS-user, and each as a receiving DLS-user may be able to exert flow control on its sending peer. A peer DLC is negotiated to provide either symmetrical service or asymmetrical service. It may also be negotiated to provide only DL-simplex service.

3.3.8 multi-peer DLC: A centralized multi-end-point DL-connection offering DL-duplex-transmission between a single distinguished DLS-user, known as the **publisher** or **publishing** DLS-user, and a set of peer but undistinguished DLS-users, known collectively as the **subscribers** or **subscribing** DLS-users, where the publishing DLS-user can send to the subscribing DLS-users as a group (but not individually), and the subscribing DLS-users can send to the publishing DLS-user (but not to each other). A multi-peer DLC always provides asymmetrical service. It may also be negotiated to provide only DL-simplex service, either from the publisher to the subscribers, or from the subscribers to the publisher.

NOTES —

1. In this last case, the characterizations as publisher and subscriber are misnomers.
2. The publishing DLS-user should use rate control because the subscribing DLS-users cannot exert flow control on their publishing peer-entity. Similar considerations apply to the subscribing DLS-users.

3.3.9 publisher: The DL-entity of the DLS-user at the centralized end-point of a multi-peer connection. This DLS-user is also known as the **publishing** DLS-user; it can send only to the subscribing DLS-users as a group, but not to any smaller subset of the subscribers.

3.3.10 subscriber: The DL-entity of a non-publishing DLS-user of a multi-peer connection. Such a DLS-user is also known as a **subscribing** DLS-user; it can send only to the publishing DLS-user.

3.3.11 DLSAP: A point at which DL-services are provided by a DL-entity to a higher-layer entity.

NOTE — This definition, derived from ISO 7498-1, is repeated here to facilitate understanding of the critical distinction between DLSAPs and their DL-addresses. (See Figure 2.)

3.3.12 (individual) DLSAP-address: A DL-address that designates only one DLSAP within the extended link. A single DL-entity may have multiple DLSAP-addresses associated with a single DLSAP.

3.3.13 group DL-address: A DL-address that potentially designates more than one DLSAP within the extended link. A single DL-entity may have multiple group DL-addresses associated with a single DLSAP. A single DL-entity also may have a single group DL-address associated with more than one DLSAP.

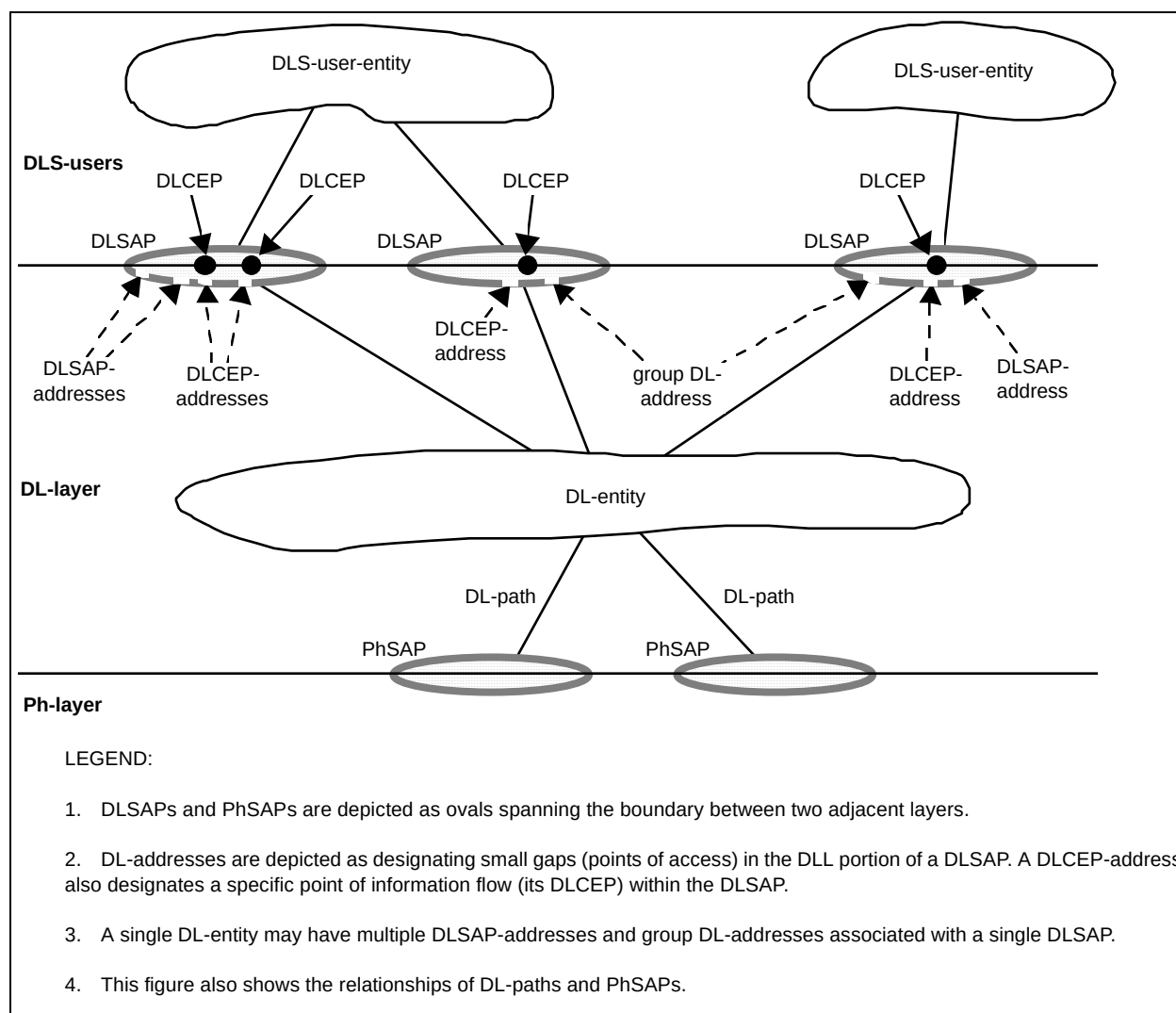


Figure 2 — Relationships of DLSAPs, DLCEPs, DLSAP-addresses, DLCEP-addresses and group DL-addresses

3.3.14 DL(SAP)-address: Either an individual DLSAP-address, designating a single DLSAP of a single DLS-user, or a group DL-address potentially designating multiple DLSAPs of one or more DLS-users.

NOTE — This terminology is chosen because ISO 7498-3 does not permit the use of the term DLSAP-address to designate more than a single DLSAP at a single DLS-user.

3.3.15 DLCEP-address: A DL-address which designates either

- a) one peer DL-connection-end-point; or
- b) one multi-peer publisher DL-connection-end-point, and implicitly the corresponding set of subscriber DL-connection-end-points

where each DL-connection-end-point exists within a distinct DLSAP and is associated with a corresponding distinct DLSAP-address.

NOTE — A DLCEP-address is an extension of the use of DL-addresses beyond that specified in ISO 7498-3.

3.3.16 DLSEP-address: A DL-address which designates a DL-scheduling-end-point within a DLE.

NOTE — This is an extension of the use of DL-addresses beyond that specified in ISO 7498-3.

3.3.17 NODE DL-address: A DL-address which designates the (single) DL-entity associated with a single node (see 3.3.3) on a specific local link.

3.3.18 transaction: Either

- a) a sequence of two related DLPDUs which appear consecutively on the local link, where the first DLPDU required the immediate transmission of the second DLPDU; or
- b) a single DLPDU on the local link which does not require the immediate transmission of a second DLPDU.

The DLE sending the first DLPDU of the transaction is known as the **initiator**; the DLE which sends the second DLPDU of the transaction, if any, is known as the **responder**.

NOTE — A DL-entity can be both an initiator and a responder in the same transaction.

3.3.19 immediate reply: A DLPDU sent by the responder in a transaction as a solicited reply to the immediately-prior DLPDU. The original DLPDU sent by the initiator of the transaction specifies the constraints on the replying DLPDU.

3.3.20 Link Active Scheduler (LAS): A special-purpose elective role of a DLE which schedules the local link and serves as the local source of DL-time for the link. The LAS functions exist within each link master (LM) DLE, which contends for and then activates those LAS functions after detecting the absence on the link of a DLE whose LAS functions are active. The DLE serving as LAS, sometimes referred to as the **LAS DLE**, receives, and responds to, scheduling requests from all DLEs on the link, including itself. It also receives and responds to requests for the current DL-time.

3.3.21 Link Master (LM): A DLE which can also provide the LAS functions for the link, including initializing and scheduling the link.

3.3.22 token: A right to transmit on the local link. This right is assumed by a DLE when it activates its LAS functions. This right may be delegated to individual DLEs, subject to specified constraints on its usage. In all cases, this right ultimately reverts to the DLE which has activated its LAS functions (the LAS DLE) .

Each token is implicitly qualified by the method of its transmission or assumption:

- a) A **scheduler token** is assumed and held by the LAS DLE, and can be sent to another LM DLE on the local link to transfer the activation of LAS functions to that receiving DLE.
- b) A **delegated token** is created by the LAS DLE and sent to a DLE on the local link, and is returned upon completion of its use, or assumed by the LAS DLE at its expiration.
- c) A **reply token** is created by the current delegated token holder (or scheduler token holder if there is no delegated token holder) and sent to a DLE on the local link, requesting an immediate reply; it is returned with that immediate reply, or assumed by the current delegated token holder at the expiration of the reply period.

NOTE — It is possible, though not absolutely required, for a DLE to function simultaneously as an LAS, an initiator, and a responder, delegating a token and receiving that token, requesting an immediate reply and receiving that request, and then replying to itself as requestor and returning the token to itself as LAS. This would require the transmission of at least three DLPDUs — one from each role.

3.3.23 dominant token: The unique right to initiate the next transmission on the local link:

- a) A reply token is the dominant token on the local link during the period after its creation and before its expiration or return, and the reply token holder (the responder) holds the unique right to transmit.
- b) Otherwise, when no reply token exists, a delegated token is the dominant token on the local link during the period after its creation and before its expiration or return, and its token holder (the initiator) holds the unique right to transmit.
- c) At all other times the scheduler token is the dominant token on the local link, and the LAS DLE (which functions as the initiator) holds the unique right to transmit.

3.3.24 node-timer: A frequency-adjustable counter, maintained by the local DLE, which is used to provide the DLS-user with a local multi-partite sense of DL-time (see 5.5.4 and 9.4.1.1) such that one component of that DL-time forms a monotonically-increasing sense of local time available for use within the attached end system. Conceptually, the node-timer counts in nominal units of 2^{-13} ms and has a period of over 100 years. Therefore, any actual counter shall be a binary counter whose least significant bit has a nominal weight of $2^{\pm N}$ ms, with a rollover period greater than the expected maximum interval between resets of the DLE.

NOTE 1 — When no information about that maximum interval is available, an interval of 5 years shall be assumed.

The node-timer is also used within the DL-protocol to provide a shared sense of DL-time which is used both to synchronize DLE scheduling actions, where appropriate, and to synchronize the rate of drift of all of the node-timers on the extended link. This latter is achieved by adjusting the frequency of each DLE's node-timer such that the DLE's monotonic sense of local time maintains an approximately constant phase relationship with that of the DLE serving as the **time-reference DLE**.

NOTE 2 — This adjustment is the reason why the weight of the node-timer unit of counting is only nominally $2^{\pm N}$ ms.

3.3.25 slot-time: A configuration parameter of each local link, measured as an **integral** multiple of the transmission period of one complete octet, with an integral value of between 1 and 4095 octets:

- 1) Slot-time is a fundamental link parameter with multiple uses:

A) Slot-time is used by each Link Master DLE connected to the link in determining the amount of time for which that DLE monitors the link for inactivity before sending a CLAIM LAS (CL) DLPDU. Slot-time is defined such that the nominal link-inactivity monitoring periods of two DLEs which:

- have consecutive NODE DL-addresses; and
- which do not hold any token

differ by exactly one slot-time.

B) Slot-time is used by each Basic, Link Master, or Bridge DLE on the local link to compute the durations of all other periods of link inactivity which the DLE must monitor. All such durations are specified as standardized or configured multiples of one slot-time.

- 2) Slot-time is a configured minimum upper bound on the maximum two-way asynchronism in immediate communications among interacting DLEs on the local link when trying to (re)initialize the link, maximized across all pairs of DLEs on that local link. Given this viewpoint, slot-time is an aggregate measure of the worst-case implementation delays within the intervening media, the PhL, and the PhL/DLL interfaces, all of which limit the rapidity of two-way DLE interaction on the local link.

Slot-time shall be computed as the sum of:

- a) the worst-case two way propagation delays through the intervening media and intervening PhEs, such as repeaters, between any two PhEs associated with their respective DLEs on the local link, measured between the point of connection to the local link of each of those PhEs, including worst-case internal logic and analog circuit delays in any intermediate repeaters;
- b) the maximum-inactivity-to-claim-LAS-delay as defined in 3.3.25.1; and
- c) a safety factor, which is used to account for:
 - 1) the difference in two relevant internal delays consisting of
 - i) the delay between
 - A) the presentation of the first non-silent PhPDU to the DLE's associated PhE at that PhE's point of attachment to
 - B) the indication of start of activity from that PhE to that DLE; and
 - ii) the delay between
 - A) the presentation of the last non-silent PhPDU to the DLE's associated PhE at that PhE's point of attachment to
 - B) the indication of end of activity from that PhE to that DLE;
 - 2) the rate differences in the internal timer clocks among the DLEs on the local link;
 - 3) the limited resolution of the measurements and potential measurement errors; and
 - 4) any extra delay needed to make the sum of (a),(b) and (c) round to an integral multiple of the transmission period of one octet.

NOTES —

- 1. These delays are defined in a manner that permits their measurement.
- 2. Finer resolution than one octet is not possible without knowledge of the specific associated PhL, because Ph-timing comes from the PhL and the formal PhL-DLL interface provides only octet timing see 5.4).
- 3. The definitions of 3.3.25 through 3.3.28 are not recursive, because 3.3.25 is based on a different measurement than 3.3.26 through 3.3.28. The definitions of 3.3.26 through 3.3.28 are in units of one slot-time to reduce the complexities of implementations of this protocol. As a result, a single 8-bit hardware timer, prescaled by the slot-time, enabled on bus inactivity, with usage-specific stopping and retriggering conditions, can provide the required functionality for all of the slot-time-based timers of this protocol.

3.3.25.1 maximum-inactivity-to-claim-LAS-delay A configuration parameter of each local link, its minimum value is the worst-case internal delays of any Link Master DLE and associated PhEs connected to that link. This internal delay is the sum of the following two components:

- a) the internal delay between
 - 1) the moment of presentation of the last non-silent PhPDU of a transmission to the DLE's associated PhE at that PhE's point of connection to the local link; and
 - 2) the resulting start of the DLE's internal timer which monitors link inactivity; and
- b) the internal delay between
 - 1) the expiration of that timer; and

- 2) the resulting moment of presentation of the first non-silent PhPDU of a transmission of the required CLAIM LAS (CL) DLPDU by the DLE's associated PhE at that PhE's point of connection to the local link.

The unit of measurement of this aggregate delay is one-eighth of the transmission period of one octet (that is, one nominal "bit-period"). The range of values of this parameter is 1 to 4095 nominal "bit-periods."

3.3.26 maximum-response-delay: A measure, in units of one slot-time, greater than the worst-case period of **local** PhE-inactivity which can be observed by a DLE

- a) which has just received a delegated token, before initiating transmission of a DLPDU; and
- b) which has just received a reply token, before initiating transmission of the required immediate reply DLPDU.

A DLE's minimum required value for maximum-response-delay shall be determined by measuring at the responding DLE the delay between concluding the reception of a requesting DLPDU and initiating the transmission of the immediately-subsequent responding DLPDU.

Maximum-response-delay is a configuration parameter of each local link, and shall have a value of 1 to 11. When multiplied by the duration of one slot-time, the product shall be at least as large as the largest of the maximum-response-delay values required by each of the current or anticipated DLEs on the local link.

3.3.27 immediate-response-recovery-delay: A measure, similar to maximum-response-delay, of the worst-case period of **peer** PhE-inactivity which can be observed

- a) by a DLE holding a delegated token while awaiting an immediate response from another correctly-functioning DLE on the local link; and
- b) by any other DLE on the local link, after receiving the initiating transaction of a two-DLPDU transaction, before detecting the link activity caused by transmission of that immediate response DLPDU; and
- c) by the LAS DLE while waiting for the initial link activity after
 - a reply token (via a PROBE NODE-ADDRESS (PN) DLPDU;
 - a delegated token (via a PASS TOKEN (PT) or EXECUTE SEQUENCE (ES) DLPDU); and
 - the scheduler token (via a TRANSFER LAS (TL) DLPDU)

to another correctly-functioning DLE on the local link.

More formally, immediate-response-recovery-delay provides a bound for the worst-case delay between receipt of the Ph-DATA confirm primitive for a Ph-DATA request primitive whose PhIDU specified END-OF-DATA-AND-ACTIVITY (see 5.4.3), and subsequent receipt of the next START-OF-ACTIVITY PhIDU (see 5.4.3), that can be observed by the sending DLE on the local link during

- 1) an initiator-responder interaction, in which a transaction-initiating DLE sends a DLPDU requiring an immediate reply to a responding DLE, and that responding DLE replies by sending a second DLPDU; and
- 2) a token-passing interaction, in which the LAS DLE sends a delegated token DLPDU to the next intended token-holding DLE, or the scheduler token to a successor (as LAS) DLE, and that addressed DLE responds by sending a second DLPDU.

The value of immediate-response-recovery-delay, in units of one slot-time, shall be (maximum-response-delay + 1).

3.3.28 token-recovery-delay: A measure, similar to maximum-response-delay, greater than the worst-case period of **peer** PhE-inactivity which can be observed by the LAS DLE while another correctly-functioning DLE is using a token.

More formally, token-recovery-delay provides a bound greater than the worst-case delay between receipt of an END-OF-ACTIVITY OR END-OF-DATA-AND-ACTIVITY PhIDU (see 5.4.4) and subsequent receipt of the next START-OF-ACTIVITY PhIDU, that can be observed by any DLE on the local link during normal link operation. This delay can be exceeded only when a DLE fails while holding a delegated or scheduler token.

The value of token-recovery-delay, in units of one slot-time, may be any value between (maximum-response-delay + 3) and 14.

3.3.29 link-id: The two-octet primary identifier for the local link, within the extended link, whose values are constrained as specified in Annex A/A.2.1.

3.3.30 node-id: The one-octet primary identifier for the DLE on the local link, whose values are constrained as specified in Annex A/A.2.2. A value of zero is also permitted, which inhibits all transmission by this DLE.

3.3.31 delocalization: The DLE-internal process of converting synonymous DL-addresses to a canonical form for transmission or during reception (see 6.2.2).

3.3.32 fractional-duty-cycle (FDC): A DLE which is not continuously attentive to received signaling, usually found in an end-system with an austere electrical power budget. Such a DLE generally requires the assistance of a bridge (DL-relay) DLE to communicate with other DLEs on the same link, to provide for cases where the two DLEs are not concurrently active.

Three classes of FDC DLE have been identified:

class A — FDC DLEs which are totally inattentive to their local links during their “sleep” periods, and control the timing of their temporary “re-awakening” to link communications solely on the basis of internal DL-time or some other measurement;

class B — FDC DLEs which monitor the link during their periods of “sleep” for a particular wakeup DLPDU (see 7.21) addressed specifically to the DLE’s NODE DL-address, “re-awakening” upon receipt of such a DLPDU; and

class C — FDC DLEs which monitor the link during their periods of “sleep” for any DLPDUs addressed specifically to the DLE’s NODE DL-address, “re-awakening” upon receipt of such a DLPDU.

NOTE — DLEs which “sleep” between DLPDUs, but that respond to all DLPDUs addressed to any of their DL-addresses, are not FDC DLEs.

3.3.33 DL-address user-request queue: A multi-priority FIFO-within-priority queue, associated with a specific DL-address of a DLE, of in-process DLS-user-requests. This queue is partitioned into three disjoint sections:

- 1) those requests which have already been transmitted but remain available for retransmission, confirmation, or both — that is, which have not reached their maximum confirm delay and which
 - are still within the DLCEP’s transmit window; or
 - are awaiting an acknowledging response DLPDU at the DLSAP;

- 2) those requests which are ready for transmission but have not been completely transmitted; and
- 3) those requests which are either awaiting a DL-COMPEL-SERVICE request to release them for transmission, or are outside the DLCEP's transmit window, or both.

NOTES —

1. Section 1 of the queue contains DLS-user requests which may have been successfully communicated to another DLS-user. Retransmission of all or part of the DLS-user-data associated with these requests may be required, and can be attempted before the requests are purged from this partition (by reset or peer acknowledgment on a peer DLC; by reset or user-request timeout or the need for sequence-number reuse on a multi-peer DLC).
2. Section 2 of the queue contains DLS-user requests which are ready for transmission, but could not have been completely communicated to another DLS-user.
3. Section 3 of the queue can be non-empty only when the DL-scheduling-policy for the DL-address is EXPLICIT, or when the number of queued DLSDUs exceeds the DLCEP's transmit window, or both.
4. Members of a given priority are advanced from one section to another until they are removed from the queue. (In practice, the section partitions may be moved, rather than the members.) The FIFO ordering within each priority is strictly maintained.

One such queue is associated with each DLSAP-address, each peer or publisher DLCEP-address, and each subscriber's DLCEP (which can be considered to be associated with an implicit DLCEP-address) of the DLE, and with the DLE's NODE DL-address (see 3.3.17 and 6.2.2.3).

3.3.34 DLE unscheduled-service queue: A multi-priority FIFO-within-priority queue of

- a) **references** to DL-address user-request queues (see 3.3.33);

NOTE – See 5.7.1.17(a) for a more elaborate definition.

- b) **references** to locally-scheduled active sequences (see Annex B) resulting from DL-SCHEDULE-SEQUENCE requests (see 9.4.3.1); and
- c) DT DLPDUs containing DLS-user-data which are delayed responses to received CD and ED DLPDUs, queued in support of the DL-UNITDATA-EXCHANGE service.

NOTE — Since this is a multi-priority FIFO-within-priority queue, members are removed in priority order, and within a single priority, in FIFO order.

3.3.35 timeliness, DL-timeliness: Timeliness is an attribute of a datum which provides an assessment of the temporal currency of that datum. This attribute is of particular importance in sampled-data systems, which may need to make decisions based on the timeliness, or lack of timeliness, of current data samples.

As a general rule, timeliness is a user attribute which can be affected negatively by the various layers of the data transport system. That is, a datum which was timely when the requesting user presented it to a data communications subsystem for transmission may become untimely due to delays in the communications subsystem.

DL-timeliness is an attribute of a DLS-user datum relating the timing of a DLS-user/DLE interaction which writes or reads that datum to one or more other DLS-user/DLE interactions.

NOTE — These concepts also support migration from previous standards.

4 Symbols and abbreviations

4.1 Data units

BPDU	Bridge-protocol-data-unit, used in the ISO/IEC 10038 inter-bridge protocol
DLPDU	DL-protocol-data-unit
DLSDU	DL-service-data-unit
pDLSDU	partial DL-service-data-unit — one segment of a multi-segment DLSDU
PhIDU	Ph-interface-data-unit
SPDU	Support-protocol-data-unit, used to support the full DL-protocol

4.2 Local variables, timers, counters and queues

V(ST)	slot-time	See 5.7.1.1
V(PhLO)	per-DLPDU-PhL-overhead	See 5.7.1.2
V(MRD)	maximum-response-delay	See 5.7.1.3
V(IRRd)	immediate-response-recovery-delay	See 5.7.1.4
V(MRC)	maximum-retry-count	See 5.7.1.5
V(NRC)	network-retry-count	See 5.7.1.6
V(NDL)	network-DLPDU-lifetime	See 5.7.1.7
V(TN)	this-node	See 5.7.1.8
V(TL)	this-link	See 5.7.1.9
V(MEP)	DL-MAC-address-embedding-prefix	See 5.7.1.10
C(RD)	remaining-duration down-counter	See 5.7.1.11
V(MID)	minimum-inter-DLPDU-delay	See 5.7.1.12
T(IRRd)	immediate-response-recovery-delay monitor	See 5.7.1.13
V(RA)	reply-address	See 5.7.1.14
V(OTA)	outstanding-transaction-array	See 5.7.1.15
V(LTI)	last-transaction-index	See 5.7.1.16
Q(US)	unscheduled-service queue	See 5.7.1.17
V(RID)	random-identifier	See 5.7.1.18
C(NT)	node-time up-counter	See 5.7.1.19
V(LSTO)	local-link-scheduling-time-offset	See 5.7.1.20
V(DLTO)	DL-time-offset	See 5.7.1.21
V(TQ)	time-quality	See 5.7.1.22

V(MD)	measured-delay	See 5.7.1.23
V(LN)	LAS-node	See 5.7.1.24
V(TSC)	time-synchronization-class	See 5.7.1.25
T(TDP)	time-distribution-period monitor	See 5.7.1.26
V(TSL)	time-source-link	See 5.7.1.27
P_U(SDUL)	DLSDU-length request parameter	See 5.7.2.1
P_U(SDU)	DLSDU request parameter	See 5.7.2.2
P_U(MCD)	maximum-confirm-delay parameter	See 5.7.2.3
T_U(MCD)	maximum-confirm-delay monitor	See 5.7.2.4
Q_A(UR)	user-request queue	See 5.7.3.1
V_C(ST)	DLCEP state	See 5.7.4.1
V_C(NP)	negotiated DLCEP parameters	See 5.7.4.2
V_C(N)	next sequence number to assign to a DLSDU	See 5.7.4.3
V_C(R)	maximum non-transmittable DLSDU sequence number	See 5.7.4.4
V_C(A)	maximum acknowledged DLSDU sequence number	See 5.7.4.5
V_C(M)	minimum untransmitted DLSDU sequence number	See 5.7.4.6
V_C(MS)	minimum untransmitted segment number	See 5.7.4.7
V_{C,K}(SS)	to-be-sent segments of a DLSDU	See 5.7.4.8
T_{C,K}(SS)	sent-segments monitor for a DLSDU	See 5.7.4.9
V_C(L)	last-reported DLSDU sequence number	See 5.7.4.10
V_C(H)	highest-detected DLSDU sequence number	See 5.7.4.11
V_C(HS)	highest-detected segment number of the highest-detected DLSDU sequence number	See 5.7.4.12
V_{C,K}(MRS)	missing received segments of a DLSDU	See 5.7.4.13
V_{C,K}(RRS)	retransmission-request required segments of a DLSDU	See 5.7.4.14
T_{C,K}(RRS)	retransmission request monitor for a DLSDU	See 5.7.4.15
T_C(RAS)	residual activity stimulus	See 5.7.4.16
T_C(RAM)	residual activity monitor	See 5.7.4.17
V_C(TNA)	DL-time of last network access	See 5.7.4.18
V_B(TW)	DL-time of last buffer write	See 5.7.4.19
V_B(TP)	DL-time of production	See 5.7.4.20
V_B(TS)	Timeliness-status of buffer write	See 5.7.4.21
V(DTA)	delegation-address	See 5.7.5.1

V(LL)	local-link live-list	See 5.7.5.2
V(TCL)	token-circulation list	See 5.7.5.3
V(ENRL)	expected-non-response list	See 5.7.5.4
V(MST)	maximum-scheduled-traffic	See 5.7.5.5
V(MSO)	maximum-scheduling-overhead	See 5.7.5.6
V(DMDT)	default-minimum-token-delegation-time	See 5.7.5.7
V(DTHT)	default-token-holding-time	See 5.7.5.8
V(LTHT)	link-maintenance-token-holding-time	See 5.7.5.9
V(MTHA)	maximum-token-holding-time-array	See 5.7.5.10
V(TTRT)	target-token-rotation-time	See 5.7.5.11
V(ATRT)	actual-token-rotation-time	See 5.7.5.12
V(RTHA)	remaining-token-holding-time-array	See 5.7.5.13
V(NTHN)	next-token-holding-node	See 5.7.5.14
V(FUN)	first unpolled node id	See 5.7.5.15
V(NUN)	number of consecutive unpolled node ids	See 5.7.5.16
P(TRD)	token-recovery-delay	See 5.7.5.17
V(TDP)	time-distribution-period	See 5.7.5.18
V(MICD)	maximum-inactivity-to-claim-LAS-delay	See 5.7.5.19
V(LDDP)	LAS-data-base-distribution-period	See 5.7.5.20
V(ML)	maximum-link	See 5.7.6.1

4.3 DLPDU classes

CA	Compel Acknowledgment	See 7.4
CD	Compel Data	See 7.5
CL	Claim LAS	See 7.19
CT	Compel Time	See 7.9
DC	Disconnect Connection	See 7.2
DT	Data	See 7.7
EC	Establish Connection	See 7.1
ED	Exchange Data	See 7.6
ES	Execute Sequence	See 7.16
IDLE	Idle	See 7.22
PN	Probe Node-address	See 7.13
PR	Probe Return	See 7.14
PT	Pass Token	See 7.15

RC	Reset Connection	See 7.3
RI	Request Interval	See 7.18
RQ	Round-trip-delay Query	See 7.11
RR	Round-trip-delay Reply	See 7.12
SR	Status Response	See 7.8
TD	Time Distribution	See 7.10
TL	Transfer LAS	See 7.20
RT	Return Token	See 7.17
WK	Wakeup	See 7.21

4.4 Miscellaneous

DL-	Data Link layer (as a prefix)
DLC	DL-connection
DLCEP	DL-connection-end-point
DLE	DL-entity (the local active instance of the Data Link layer)
DLL	DL-layer
DLM-	DL-management (as a prefix)
DLMS	DL-management-service
DLP-	DL-protocol (as a prefix)
DLPCI	Data Link-protocol-control-information
DLPDU	DL-protocol-data-unit
DLS	DL-service
DLSAP	DL-service-access-point
DLSDU	DL-service-data-unit
DLSEP	DL-schedule-end-point
FC	Frame Control (first octet of each DLPDU)
FCS	Frame Check Sequence
FDC	Fractional Duty Cycle (type of DLE)
FIFO	First-in first-out (queuing method)
LAS	Link Active Scheduler
LLC	Logical Link Control
LM	Link Master
MAC	Medium Access Control
OSI	Open Systems Interconnection
Ph-	Physical layer (as a prefix)

PhE	Ph-entity (the local active instance of the Physical layer)
PhICI	Physical-interface-control-information
PhID	Physical-interface-data
PhL	Ph-layer
PhS	Ph-service
PhSAP	Ph-service-access-point
QoS	Quality of Service
SLAE	Systems-load-application-entity
SMAE	Systems-management-application-entity

5 Overview of the DL-protocol

5.1 Three-level model of the DLL

The DLL is modeled as

- a) a low-level of path-access-and-scheduling functions, which supports DL-service functions.
- b) an intermediate-level of bridge-operation functions, which in turn supports DL-service functions.
- c) a higher-level of connection-mode and connectionless data transfer, bridge-coordination, and DL-service functions.

Interoperating with all three levels are DL-management functions, including bridge and redundant-path management functions where relevant.

NOTES —

1. The term “sublayer” is not appropriate for describing these levels, since ISO 7498 requires that when multiple sublayers are defined, all but one of them must be optional.
2. This three-level partitioning closely resembles the partitioning into lower-level “MAC”, intermediate-level “bridge-operation”, and higher-level “LLC” functions found in the ISO/IEC LAN standards [ISO/IEC TR 18802], with two significant differences:
 - A) This protocol’s low-level functionality is contained entirely within the Data Link layer as specified by ISO 7498. In contrast, the ‘MAC’ functionality of the ISO/IEC LAN protocols spans the lower part of the OSI Data Link layer and the upper part of the OSI Physical layer.
 - B) This protocol employs a single level of DL-addressing within the Data Link layer. In contrast, the ISO/IEC LAN protocols employ two levels of DL-addressing, one within the “MAC” and “bridge-operation” functionality and a second within the “LLC” functionality.

5.1.1 Path access and scheduling level

- a) The path-access-and-scheduling level forms each DLPDU from DL-protocol control information and DLS-user-data; computes and appends an appropriate frame check sequence (FCS); and passes the whole as a sequence of PhIDUs (see 5.4) to the PhE for transmission to peer PhEs for reporting to peer DLEs.

In some cases it also appends the low-order three octets either

- of the current value of the local node-timer, C(NT); or
- of a computed value minus the current value of the local node-timer, C(NT),

during DLPDU formation, immediately preceding the appended FCS.

b) The path-access-and-scheduling level receives a sequence of PhIDUs from the PhE, concatenates those PhIDUs into a received DLPDU, computes a frame check sequence over the entire sequence of received data, and checks for the proper residual value. The first octet of the received sequence is examined to determine the type of the DLPDU, and an attempt is made to parse that DLPDU into its DL-protocol control information and DLS-user-data components. If the FCS residual was correct and the parse was successful, then the appropriate low-level actions are performed, possibly including reporting the parsed DLPDU to a higher level.

In some cases the value of the low-order three octets of the local node-timer, C(NT), at the time of receipt, is appended to this parsed DLPDU.

c) The path-access-and-scheduling level provides the basic functions of both responder and initiator. As a responder, it provides the sequencing functions necessary

- 1) for receiving a DLPDU, possibly conveying a reply token; and
- 2) in that latter case (of the received reply token), for sending a DLPDU as an immediate reply to the just-received DLPDU.

As an initiator, it provides the sequencing functions necessary for

- 3) receiving a delegated token;
- 4) sending one or more DLPDUs, including those requiring an immediate reply;
- 5) receiving such an immediate reply, or inferring its absence; and
- 6) returning a delegated token.

d) The path-access-and-scheduling level provides the low-level scheduling functionality required for scheduling DLPDU transmissions on a specific path, including any interactions with the local link's link active scheduler (LAS) to coordinate the schedule with other DLEs or to request needed path transmission capacity.

The actions of (a) and (b) are augmented within a bridge (relay DLE) to permit the retransmission of a received sequence of data octets, including the received FCS, with possible constrained alterations of the first octet and compensating alterations (see 6.1.1.3) of the received FCS, prior to retransmission.

The actions of (c) and (d) are based in part on two request-management and scheduling queues:

- i) a DL-address-specific request queue (see 3.3.33), associated with the sending address, which is used to manage DLS-user requests originated from that DL-address; and
- ii) the common DLE unscheduled-service queue (see 5.7.1.17), which is used to manage the servicing of unscheduled requests upon receipt of the "circulated" token.

Some of the more complex scheduling functionality of this level requires, and uses, the services of the upper level (see 5.1.3) by acting as a DLE-internal quasi-DLS-user.

5.1.2 Bridge-operation level

The bridge-operation level provides the intermediate-level functionality of

- a) logically inter-connecting multiple local links into a single extended-link by physically interconnecting multiple paths;
 - b) serving as a (possibly-distributed) “time-space-time” switch:
 - 1) providing DLPDU store-and-forward functions to permit communication between DLEs on the extended link which could not otherwise communicate; and
- NOTE — This includes the coordination with fractional-duty-cycle (FDC) DLEs (see 3.3.32) necessary to permit alternating periods of “FDC-DLE-awake” and “FDC-DLE-asleep” operation.
- 2) providing surrogate functions to permit delayed-reply interactions between DLEs on the extended link as an extrapolation of “immediate-reply” transactions on a local link;
 - c) providing a shared sense of DL-time throughout the extended link; and
 - d) coordinating local-link scheduling among two or more local links to provide any necessary multi-link coordination of scheduling within the extended link.

Much of the bridge-operation level of functionality is active only in “bridge” DLEs (see 5.6).

5.1.3 Connection-mode and connectionless data transfer, bridge-coordination, and DL-service level

The connection-mode and connectionless data transfer, bridge-coordination, and DL-service level provides the higher-level functionality of

- a) managing all DLE interactions with the DLS-user, converting all DLS-user request and response primitives into the necessary sequence of DLE operations, and generating DLS-user indication and confirm primitives where appropriate;
- b) managing the sequencing of each active DLCEP, including
 - 1) determination of the type and sequence of DLPDUs to be transmitted from the DLCEP;
 - 2) QoS negotiation;
 - 3) determination of the DLPCI to be included in each DLPDU; and
 - 4) segmentation and reassembly of large DLSDUs;
- c) managing the sequencing of unitdata transactions which require a response from a peer DLE, including
 - 1) determination of the DLPCI to be included in each DLPDU; and
 - 2) correlation of a non-immediate reply DLPDU with the requesting transaction;
- d) processing all inter-DLE state-information-distribution DLPDUs, such as TIME DISTRIBUTION (TD) (see 7.10 and 9.4.1) and bridge configuration (see Annex C) DLPDUs and LAS-backup SPDUs (see Annex B); and
- e) managing all query-DLPDU/reply-DLPDU interactions with other DLEs, other than those which occur as an “immediate reply” and which are made on a reactive basis by the lower-level functions of 5.1.1.

These query-DLPDU/reply-DLPDU interactions also include computation of round-trip-delays (see 7.11, 7.12, and 9.4.1), support of remotely-initiated DL-SUBSCRIBER-QUERY (see 9.2.3) and DL-LISTENER-QUERY (see 9.3.2) requests, and inter-bridge exchanges of bridge state information (see Annex C).

5.2 Service provided by the DLL

The DLL provides connectionless data transfer services for limited-size DLSDUs, connection-mode data transfer services for limited-size DLSDUs, an internally-synchronized time service, scheduling services to control the time allocation of the underlying shared PhL service, and a DL(SAP)-address, queue and buffer management service.

NOTE — Table 10 of Part 3 of ISA-S50.02 shows many of the relationships among DLC QoS attributes.

Some relevant QoS attributes are as follows:

5.2.1 QoS — DLCEP class

Each DLCEP establishment request specifies the class of the DLCEP. The three choices for DLCEP-class are

- a) **PEER** — the DLS-user can exchange DLSDUs with one other peer DLS-user;
- b) **PUBLISHER** — the DLS-user can send DLSDUs to a set of zero or more associated subscribing DLS-users, and receive DLSDUs from any of those subscribing DLS-users;
- c) **SUBSCRIBER** — the DLS-user can receive, and request, DLSDUs from the associated publishing DLS-user, and can send DLSDUs to that publishing DLS-user.

5.2.2 QoS — DLCEP data delivery features

Both members of a peer DLC, or the publishing DLS-user of a multi-peer DLC, specify the data delivery features of the DLC's DLCEP(s). The five choices for DLCEP data delivery features, and their effects, are

- a) **CLASSICAL** — the DLS-user can send data which will be delivered without loss, duplication or misordering. All relevant DLS-users will be notified of any loss of synchronization on the DLC.
- b) **DISORDERED** — the DLS-user can send data which will be delivered immediately upon receipt, without duplication but potentially in a different order than that of the sending DLS-user. All relevant DLS-users will be notified of any unrecoverable loss of DLS-user-data or loss of synchronization on the DLC.
- c) **ORDERED** — the DLS-user can send data which will be delivered immediately upon receipt, without duplication or misordering, but with potential loss of some DLS-user-data. Loss of DLS-user-data will not be reported, and recovery of DLS-user-data lost prior to the last-reported DLS-user-data will not be attempted.
- d) **UNORDERED** — the DLS-user can send data which will be delivered immediately upon receipt. Loss, duplication and misordering of DLS-user-data will not be detected or reported. No attempt will be made by the DLS-provider to recover from such events.
- e) **NONE** — the DLS-user cannot send data in this direction of data transfer.

On a peer DLC, the QoS value for the sending DLCEP data delivery features may be chosen independently for each direction of data transfer. On a multi-peer DLC, the QoS value for the DLCEP data delivery features for the subscribers-to-publisher direction of data transfer is restricted to UNORDERED and NONE. The default QoS value for the DLCEP data delivery features in the publisher-to-subscribers direction is UNORDERED.

5.2.3 QoS — DLL priority

All DLCEP establishment requests and responses, all connectionless data transfer requests, and many DL-scheduling requests, specify an associated DLL priority used in scheduling DLL data transfer services. This DLL priority also determines the maximum amount of DLS-user-data that can be conveyed in a single DLPDU. The three DLL priorities with their corresponding ranges of conveyable DLS-user-data (per DLPDU) are, from highest priority to lowest priority:

- a) **URGENT** — ≤ 64 DLS-user octets per DLPDU;
- b) **NORMAL** — ≤ 128 DLS-user octets per DLPDU;
- c) **TIME-AVAILABLE** — ≤ 256 DLS-user octets per DLPDU.

NOTES —

- 1. URGENT and NORMAL are considered **time-critical** priority levels; TIME-AVAILABLE is considered a **non-time-critical** priority level.
- 2. DLC establishment and DLC release DLPDUs which are sent at TIME-AVAILABLE priority are restricted to convey no more DLS-user-data than would be permitted at NORMAL priority — 128 octets.

5.2.4 QoS — DLPDU authentication

Each DLCEP establishment negotiation, and each connectionless data transfer, uses this attribute to determine

- a) a lower bound on the amount of DL-addressing information used in the DLPDUs that provide the associated DLL data transfer services;

NOTE 1 — This has a slight impact on the residual rate of DLPDU misdelivery; more addressing information reduces the potential for misdelivery.

- b) whether the current state of a sending peer or publisher DLCEP should be sent at low-frequency to the DLC's peer or subscriber DLCEP(s) even when there are no unconfirmed DLS-user requests outstanding at the sending DLCEP; and

- c) whether all related scheduling actions should be executed locally.

NOTE 2 — These last two aspects are of particular importance in safety systems.

The three levels specifiable, with their amounts of DL-addressing information, are

- 1) **ORDINARY** — each DLPDU shall include the minimum amount of addressing information necessary;
- 2) **SOURCE** — each DLPDU shall include a source DL-address where possible; and
- 3) **MAXIMAL** — each DL-address shall include the maximal amount of addressing information possible. Also, all related scheduling actions should be executed locally; and each sending peer or publisher DLCEP of the DLC should maintain a low-frequency report of state information when there is no DLS-user activity.

5.2.5 QoS — DLL maximum confirm delay

Each DLCEP establishment request, and each response, specifies upper bounds on the maximum time duration permitted for the completion

- a) of a DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY primitives, and, separately;
- b) of a DL-DATA primitive.

Each connectionless service request specifies an upper bound on the maximum time duration permitted for the completion

- c) of a locally-confirmed DL-UNITDATA primitive, and, separately;
- d) of a remotely-confirmed DL-UNITDATA primitive, a DL-LISTENER-QUERY primitive, or an instance of the DL-UNITDATA-EXCHANGE service.

Each parameter either has the value **UNLIMITED** or specifies an interval, in units of 1 ms, from 1 ms to 60 s, inclusive. The value **UNLIMITED** provides compatibility with prior OSI protocols, and provides a means for DL-CONNECT requests to remain in a “listening” or “half-open” state. The completion status of “timeout” cannot occur on a DLS-user request which specifies **UNLIMITED**.

The parameters for the DL-DATA and locally-confirmed DL-UNITDATA primitives specify intervals less than or equal to that for the DL-CONNECT, DL-RESET, DL-SUBSCRIBER-QUERY, remotely-confirmed DL-UNITDATA, and DL-LISTENER-QUERY primitives.

The intervals specified are the maximum permissible delays

- 1) between the issuing of the specified request primitives and the issuing of the corresponding confirm primitives; and
- 2) between the initiation and completion of a single instance of the specified publishing or unitdata-exchange service.

NOTE — For DLEs that do not support a time resolution of 1 ms, the requested time interval may be rounded up to the next-greatest multiple of that resolution that the DLE does support, or to approximately 60 s if the DLE has no sense of time.

Failure to complete a DL-CONNECT or DL-RESET request within the specified interval shall result in a DLS-provider initiated release of the DLCEP, and possibly of the DLC.

5.2.6 QoS — DL-scheduling-policy

For each DLSAP-address, and each DLCEP, the DLS-user can override the normal (implicitly-scheduled) DLL policy of providing the requested DL-service as soon as possible, and instead can defer any inter-DLS-user communication required by a DL-DATA or DL-UNITDATA request DLS-primitive until that deferral is released by an involved DLS-user. Each such release, by execution of a DL-COMPEL-SERVICE request, specifying the DLSAP-address or DLCEP, permits the completion of just a single deferred request DLS-primitive. Only DL-services that provide DLS-user intercommunication are affected by this attribute.

The two choices are:

- a) **IMPLICIT** — any required communications with peer DLS-user(s) from this DLSAP-address, or from this DLCEP, will occur as soon as possible; and
- b) **EXPLICIT** — any required data or unitdata communications with peer DLS-user(s) from this DLSAP-address, or from this DLCEP, will occur only when the deferral is explicitly released by an involved DLS-user.

NOTE — Scheduling of DLPDU transmission to support the DL-CONNECT, DL-RESET and DL-DISCONNECT services, and to support responder-deferred replies in the DL-UNITDATA-EXCHANGE and remotely-confirmed DL-UNITDATA services, is always **IMPLICIT**. Scheduling of DLPDU transmission to initiate the DL-UNITDATA-EXCHANGE service is always **EXPLICIT**.

5.2.7 QoS — maximum DLSDU sizes

Each DLC/DLCEP establishment request, and each response, specifies an upper bound on the size (in octets) of DLSDUs which will be offered for transmission, and an upper bound on the size of DLSDUs which are acceptable for reception.

For peer DLCs, the negotiated maximum DLSDU size shall be determined independently for each direction of data transfer as the smallest of that offered by the sender, that permitted by the sender's local DL-management, that permitted by the receiver's local DL-management, and that permitted by the receiver.

The sender's offered size may be any value between zero and 16 times the maximum number of DLS-user octets per DLPDU, as specified in 5.2.3. The receiving DLE and all DL-management agents shall choose their maximum permitted sizes from the following list of sizes: 0, 64, 128, 256 $\times N$ where $1 \leq N \leq 16$.

NOTES —

1. The maximum size DLSDU supported by this protocol is 16 times the maximum number of octets of DLS-user-data conveyable in a single DLPDU, as determined by the DLC's DLL Priority (see 5.2.3).
2. The set of maximum permitted DLSDU sizes is limited to the above small list of values to promote interoperation. The sender's maximum specified size is permitted to take any value within the range permitted by the DLC's DLL priority to facilitate optimization of the shared communications capacity of the DLL.
3. A value of zero (0) corresponds to the choice of simplex service, as specified by the DLS-user by the choice NONE as described in 5.2.2(e).

For multi-peer DLCs, the negotiated maximum DLSDU size shall be the smaller of that offered by the publisher and that permitted by the publisher's local DL-management. For subscribers of multi-peer DLCs, the DLC shall be refused by the DLS-provider (the subscriber's DLE) if the maximum DLSDU size established by the publisher is larger than the smaller of that permitted by the subscriber and that permitted by the subscriber's local DL-management.

The publisher's offered size in the publisher-to-subscribers direction may be any value between zero and 16 times the maximum number of DLS-user octets per DLPDU, as specified in 5.2.3. The publisher's offered size in the subscribers-to-publisher direction may be any value between zero and the maximum number of DLS-user octets per DLPDU, as specified in 5.2.3.

The subscribers and all DL-management agents shall choose their maximum permitted sizes from the following list of sizes: 0, 64, 128, 256 $\times N$ where $1 \leq N \leq 16$.

NOTES —

4. The maximum size DLSDU supported by this protocol in the publisher-to-subscribers direction is 16 times the maximum number of octets of DLS-user-data conveyable in a single DLPDU, as determined by the DLC's DLL Priority (see 5.2.3).
5. The maximum size DLSDU supported by this protocol in the subscribers-to-publisher direction is the maximum number of octets of DLS-user-data conveyable in a single DLPDU, as determined by the DLC's DLL Priority (see 5.2.3).
6. The set of maximum permitted DLSDU sizes is limited to the above small list of values to promote interoperation. The publisher's maximum specified DLSDU size is permitted to take any value within the range permitted by the DLC's DLL priority to facilitate optimization of the shared communications capacity of the DLL.
7. A value of zero (0) corresponds to the choice of simplex service, as specified by the DLS-user by the choice NONE as described in 5.2.2(e).

The default value for both the sender's and receiver's maximum DLSDU size is the maximum number of DLS-user octets which can be carried by a single DLPDU of the specified DLL Priority. The DLS-provider shall always support this DLSDU size.

5.2.8 QoS — DLCEP and DLSAP-address buffer-and-queue bindings

Each DLCEP establishment request, and each response, can bind one or two local retentive buffers or specified-depth FIFO queues, created by DL-CREATE buffer and queue management primitives (or by DL-management), to the DLCEP:

NOTE — When these bindings are made for a DLS-user of a peer DLC, or a publishing DLS-user of a multi-peer DLC, they determine the maximum transmit window (that is, number of transmitted but unacknowledged DLSDUs) for that direction of DLC data transfer. Since the size of the transmit window can also be limited by DL-management, or by an implementation, the queue depth only imposes an upper limit on the window size.

- a) One queue or retentive buffer can be bound to a DLCEP to convey DLSDUs from the DLS-user to the DLS-provider.
- b) One queue or retentive buffer can be bound to a DLCEP to convey DLSDUs from the DLS-provider to the DLS-user.
- c) It is also possible to bind a queue or retentive buffer to be written at one DLCEP and to source data at another DLCEP. Such an intermediate buffer or queue can serve to cross scheduling boundaries or redistribute received DLS-user-data to a second set of DLS-users.

Such a binding is made by specifying, for the appropriate parameter, a buffer-or-queue DL-identifier which resulted from a prior DL-CREATE request (or by DL-management), and which has not yet been deleted.

When the sending DLCEP data delivery features specify UNORDERED or ORDERED, both the sender and receiver(s) may specify a queuing policy of BUFFER-R or QUEUE. When the DLCEP's sending data delivery features specify DISORDERED or CLASSICAL, both the sender and receiver(s) may specify a queuing policy of QUEUE; a queuing policy of BUFFER-R is not permitted. A queuing policy of BUFFER-NR is never permitted.

Each DLSAP-address bind request can bind up to six local retentive buffers or non-retentive buffers or specified-depth FIFO queues, created by DL-CREATE buffer and queue management primitives (or by DL-management), to the DLSAP-address:

- d) One buffer or queue can be bound to the sending direction of a DLSAP-address at each priority to convey DLSDUs from the DLS-user to the DLS-provider. Buffers can be bound only to DLSAP-addresses whose DLSAP-role is INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER. Queues can be bound only to DLSAP-addresses whose DLSAP-role is BASIC.
- e) One buffer or queue can be bound to the receiving direction of a DLSAP-address at each priority to convey DLSDUs from the DLS-provider to the DLS-user. Buffers can be bound only to DLSAP-addresses whose DLSAP-role is INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER. Queues can be bound to all DLSAP-addresses.

5.2.8.1 Binding to a buffer

When a sending buffer is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

- a) a DL-PUT request primitive overwrites the buffer with a DLSDU, or may set the buffer empty.

NOTE — After creation, the buffer is empty.

- b) a DL-COMPEL request primitive, specifying either
 - 1) a DLCEP; or
 - 2) unitdata-exchange at a DLSAP-address

causes the transmission, at the first opportunity, of the then most recent contents of the buffer; the primitive does not itself specify a DLSDU.

c) a DL-BUFFER-SENT indication primitive notifies the DLS-user of the specific DLCEP on which the retentive buffer was transmitted, and to which the buffer is bound, that the buffer was just transmitted.

d) a DL-UNITDATA-EXCHANGE indication primitive notifies the single DLS-user attached to the specific DLSAP-address from which the buffer was transmitted (and to which the buffer is bound) that the buffer was just transmitted on that DLSAP-address, and emptied if it is non-retentive (BUFFER-NR).

When a receiving buffer is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

e) a DL-BUFFER-RECEIVED or DL-UNITDATA-EXCHANGE indication primitive notifies the DLS-user of the overwriting of the buffer by a newly-received DLSDU; the primitive does not itself specify a DLSDU.

f) a DL-GET request primitive copies the DLSDU from the buffer, and empties the buffer if it is non-retentive (BUFFER-NR).

Multiple concurrent output bindings to a retentive buffer are permitted as an implementation and conformance (see 11.1) option, but are not required by this protocol.

5.2.8.2 Binding to a FIFO queue

When a sending FIFO queue of maximum depth K is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

a) a DL-PUT request primitive is not permitted.

b) a DL-DATA or DL-UNITDATA request primitive attempts to append a DLSDU to the queue, but fails if the queue already contains K DLSDUs. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

When a receiving FIFO queue of maximum depth K is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

c) a DL-GET request primitive attempts to remove a DLSDU from the queue, but fails if the queue is empty.

d) a DL-DATA or DL-UNITDATA or DL-UNITDATA-EXCHANGE indication primitive notifies the receiving DLS-user of the result of appending a newly-received DLSDU to the receive queue; the primitive does not itself specify a DLSDU.

Multiple concurrent input bindings to a FIFO queue are permitted as an implementation option, but are not required by this protocol.

5.2.8.3 Default bindings

When these binding options are not specified, or for DLS-primitives for which explicit binding is not applicable, the conventional implicitly-queued sending and direct receiving interfaces between DLS-user and DLS-provider are employed. In this case each DL-DATA and DL-UNITDATA primitive conveys a DLSDU, and each DL-CONNECT, DL-DISCONNECT and DL-RESET primitive may convey a DLSDU:

a) DL-PUT and DL-GET request primitives are not permitted.

- b) A DL-DATA or DL-UNITDATA or DL-DISCONNECT request primitive, or a DL-CONNECT or DL-RESET request or response primitive, issued by the sending DLS-user attempts to append a DLSDU to the implicit queue, but fails if the queue is full. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.
- c) A DL-DATA or DL-UNITDATA or DL-DISCONNECT indication primitive, or a DL-CONNECT or DL-RESET indication or confirm primitive, notifies a receiving DLS-user of a newly-received DLSDU, and conveys that DLSDU to the DLS-user. No apparent queuing is provided within the DLL.

5.2.9 DL-timeliness

This attribute applies only to DL-buffers, to DLCEPs at which DL-buffers are bound, and to those DLS-primitives which transfer DLS-user-data to or from DL-buffers at such DLCEPs.

Each DLCEP establishment request, and each response, can specify DL-timeliness criteria which are to apply to information sent from, or received into, buffers at that DLCEP. Four types of DL-timeliness can be supported: RESIDENCE timeliness, UPDATE timeliness, SYNCHRONIZED timeliness, and TRANSPARENT timeliness. All four types of timeliness, and the case where there is no timeliness, are shown in Figure 4 of Part 3 of ISA-S50.02:

- a) **RESIDENCE** timeliness is an assessment based upon the length of time that a DLS-user datum has been resident in a buffer, which is the time interval between
 - 1) the moment when the buffer is written (by a DL-PUT request primitive, or by reception into the buffer at a DLCEP); and
 - 2) the moment when the buffer is read (by a DL-GET request primitive, or by transmission from the buffer at a DLCEP);

$$\text{DL-timeliness} \equiv 0 \leq (R_T - W_T) < \Delta T \quad (\text{Eq. 1})$$

NOTE — This type of timeliness was called **Asynchronous** in prior standards.

- b) **UPDATE** timeliness is an assessment based upon the time interval between
 - 1) the moment of occurrence of a multi-DLE synchronizing event (a DL-BUFFER-RECEIVED indication or DL-BUFFER-SENT indication); and
 - 2) the moment when the buffer is written (by a DL-PUT request primitive, or by reception into the buffer at a DLCEP);

$$\text{DL-timeliness} \equiv 0 \leq (W_T - S_T) < \Delta T \quad (\text{Eq. 2})$$

NOTE — A type of timeliness closely related to this one was called **Punctual** in prior standards.

- c) **SYNCHRONIZED** timeliness is an assessment based upon the time intervals and timing relationships between
 - 1) the moment of occurrence of a multi-DLE synchronizing event (a DL-BUFFER-RECEIVED indication or DL-BUFFER-SENT indication);
 - 2) the moment when the buffer is written (by a DL-PUT request primitive, or by reception into the buffer at a DLCEP); and
 - 3) the moment when the buffer is read (by a DL-GET request primitive, or by transmission from the buffer at a DLCEP);

$$\text{DL-timeliness} \equiv 0 \leq (W_T - S_T) \leq (R_T - S_T) < \Delta T \quad (\text{Eq. 3})$$

NOTE — This type of timeliness was called **Synchronous** in prior standards.

d) **TRANSPARENT** timeliness occurs when timeliness is selected on a DLCEP but none of the above assessments are performed. In such a case the DLC preserves any prior buffer timeliness, but does not itself invalidate that timeliness. When no prior buffer timeliness exists, the default timeliness value shall be **TRUE**.

e) **No** timeliness occurs when timeliness is not selected on a DLCEP. In such a case the DL-timeliness attribute of DLS-user-data always shall be **FALSE**.

Where a buffer read or write occurs over a significant time interval, and not just as a momentary event, then the overall timeliness of the read or write operation shall be computed as the timeliness at the beginning of the read or write operation, logically ANDed with the timeliness at the end of the read or write operation, all using the same timeliness criteria.

5.2.10 Remote-DLE-confirmed

Each unitdata transfer request can specify whether confirmation of receipt of the associated DLSDU by the (implicitly addressed) remote DLE is required. Its permissible values are **TRUE** and **FALSE**.

NOTE — Selection of the value **TRUE** inevitably uses more link capacity than does selection of the value **FALSE**.

5.3 DL-addresses

DL-addresses are used as DLSAP-addresses and group DL-addresses, as DLCEP-addresses, and as DLSEP-addresses. DL-addresses conform to the structure and coding limitations specified in Annex A.

NOTE — DL-addresses are also addressed to some extent in 6.2.

5.4 Service assumed from the Physical Layer

This sub-clause defines the assumed Physical Service (PhS) primitives and their constraints on use by the DLE.

NOTE — Proper layering requires that an (N+1)-layer entity not be concerned with, and that an (N)-service interface not overly constrain, the means by which an (N)-layer provides its (N)-services. Thus the Ph-service interface does not require DLEs to be aware of internal details of the PhE (for example, preamble, postamble and frame delimiter signal patterns, number of bits per baud), and should not prevent the PhE from using appropriate evolving technologies.

5.4.1 Assumed primitives of the PhS

The granularity of transmission in the Fieldbus protocol is one octet. This is the granularity of PhS-user data and timing information exchanged at the PhL – DLL interface.

5.4.1.1 PhS characteristics reporting service

The PhS is assumed to provide the following service primitive to report essential PhS characteristics used in DLL transmission, reception, and scheduling activities:

Ph-CHARACTERISTICS indication (minimum-data-rate, framing-overhead)

where

minimum-data-rate — specifies the effective minimum rate of data conveyance in bits/second, including any timing tolerances, of any PhE on the local link.

NOTE 1 — A PhE with a nominal data rate of 1 Mbit/s $\pm$ 0,01% would specify a minimum data rate of 0,9999 Mbit/s.

framing-overhead — specifies the maximum number of bit periods, where

period = $\frac{1}{\text{data rate}}$, used in any transmission for PhPDUs which do not directly convey data (for example, PhPDUs conveying preamble, frame delimiters, postamble, inter-frame “silence”, and so on).

NOTE 2 — If the framing overhead is F and two DL message lengths are L_1 and L_2 , then the time to send two immediately consecutive messages of lengths L_1 and L_2 will be at least as great as the time required to send one message of length $L_1 + F + L_2$.

If this framing-overhead is more than the DLE's configured per-DLPDU-PhL-overhead, V(PhLO), then the DLE shall report this discrepancy to DL-management and shall not issue Ph-DATA requests while the discrepancy exists.

NOTE 3 — This restriction prohibits DLE transmission while this discrepancy exists. The DLE's local station management may remedy this discrepancy by reconfiguring V(PhLO) to a greater value.

5.4.1.2 PhS transmission and reception services

The PhS is assumed to provide the following service primitives for transmission and reception:

Ph-DATA request (class, data);

Ph-DATA indication (class, data); and

Ph-DATA confirm (status)

where

class — specifies the Ph-interface-control-information (PhICI) component of the Ph-interface-data-unit (PhIDU). For a Ph-DATA request, its possible values are:

START-OF-ACTIVITY — transmission of the PhPDUs which precede Ph-user data should commence;

DATA — the single-octet value of the associated data parameter should be transmitted as part of a continuous correctly-formed transmission; and

END-OF-DATA-AND-ACTIVITY — the PhPDUs which terminate Ph-user data should be transmitted after the last preceding octet of Ph-user data, culminating in the cessation of active transmission;

For a Ph-DATA indication, its possible values are:

START-OF-ACTIVITY — reception of an apparent transmission from one or more PhEs has commenced;

DATA — the associated data parameter was received as part of a continuous correctly-formed reception;

END-OF-DATA — the ongoing continuous correctly-formed reception of Ph-user data has concluded with correct reception of PhPDUs implying END-OF-DATA;

END-OF-ACTIVITY — the ongoing reception (of an apparent transmission from one or more PhEs) has concluded, with no further evidence of PhE transmission; and

END-OF-DATA-AND-ACTIVITY — simultaneous occurrence of END-OF-DATA and END-OF-ACTIVITY;

data — specifies the Ph-interface-data (PhID) component of the PhIDU. It consists of one octet of Ph-user-data to be transmitted (Ph-DATA request) or which was received successfully (Ph-DATA indication).

status — specifies either success or the locally-detected reason for inferring failure.

The Ph-DATA confirm primitive provides the critical physical timing feedback necessary to inhibit the DLE from starting a second transmission before the first is complete. The final Ph-DATA confirm of a transmission shall not be issued until the PhE has completed the current transmission.

5.4.2 Notification of PhS characteristics

The PhE has the responsibility for notifying the DLE of those characteristics of the PhS which are relevant to DLE operation. This notification is accomplished by the PhE by issuing a single Ph-CHARACTERISTICS indication primitive at each of the PhE's PhSAPs at PhE startup.

5.4.3 Transmission of Ph-user-data

The PhE determines the timing of all transmissions. When a DLE has a DLPDU to transmit, and the DL-protocol gives that DLE the right to transmit, then the DLE shall send the DLPDU, including a concatenated FCS, by making a well-formed sequence of Ph-DATA requests, consisting of a single request specifying START-OF-ACTIVITY; followed by three to 300 consecutive requests, inclusive, specifying DATA; and concluded by a single request specifying END-OF-DATA-AND-ACTIVITY.

The PhE signals its completion of each Ph-DATA request, and its readiness to accept a new Ph-DATA request, with a Ph-DATA confirm primitive; the status parameter of the Ph-DATA confirm primitive conveys the success or failure of the associated Ph-DATA request. A second Ph-DATA request should not be issued until after the Ph-DATA confirm corresponding to the first request has been received from the PhE.

5.4.4 Reception of Ph-user-data

The PhE reports a received transmission with a well-formed sequence of Ph-DATA indications, which shall consist of either

- a) a single indication specifying START-OF-ACTIVITY; followed by consecutive indications specifying DATA; followed by a single indication specifying END-OF-DATA; and concluded by a single indication specifying END-OF-ACTIVITY; or
- b) a single indication specifying START-OF-ACTIVITY; followed by consecutive indications specifying DATA; followed by a single indication specifying END-OF-DATA-AND-ACTIVITY; or
- c) a single indication specifying START-OF-ACTIVITY; optionally followed by one or more consecutive indications specifying DATA; and concluded by a single indication specifying END-OF-ACTIVITY.

NOTE — This last sequence is indicative of an incomplete or incorrect reception. Detection of an error in the sequence of received PhPDUs, or in the PhE's reception process, disables further Ph-DATA indications with a class parameter specifying DATA, END-OF-DATA, or END-OF-DATA-AND-ACTIVITY until after both the end of the current period of activity and the start of a subsequent period of activity have been reported by Ph-DATA indications specifying END-OF-ACTIVITY and START-OF-ACTIVITY, respectively.

In the first two cases, the DLE concatenates the received data and then attempts to parse it into a DLPDU followed by a concatenated FCS. In the last case the DLE discards all reported data and reports the event to DL-management.