

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 4-2: Data-link layer protocol specification – Type 2 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-2: Spécification du protocole de la couche liaison de données –
Éléments de type 2**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2019 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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

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

About the IEC

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

About IEC publications

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

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

IEC Customer Service Centre - webstore.iec.ch/csc

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

Electropedia - www.electropedia.org

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

IEC Glossary - std.iec.ch/glossary

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

A propos de l'IEC

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

A propos des publications IEC

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

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

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

Service Clients - webstore.iec.ch/csc

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

Electropedia - www.electropedia.org

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

Glossaire IEC - std.iec.ch/glossary

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 4-2: Data-link layer protocol specification – Type 2 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-2: Spécification du protocole de la couche liaison de données –
Éléments de type 2**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.20; 35.110

ISBN 978-2-8322-9134-4

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

CONTENTS

FOREWORD.....	15
INTRODUCTION.....	17
1 Scope.....	19
1.1 General.....	19
1.2 Specifications	19
1.3 Procedures	19
1.4 Applicability	20
1.5 Conformance	20
2 Normative references	20
3 Terms, definitions, symbols, abbreviated terms and conventions	22
3.1 Reference model terms and definitions	22
3.2 Service convention terms and definitions	24
3.3 Common terms and definitions	24
3.4 Additional Type 2 definitions	25
3.5 Type 2 symbols and abbreviated terms	33
3.6 Conventions for station management objects	34
4 Overview of the data-link protocol.....	35
4.1 General.....	35
4.1.1 DLL architecture	35
4.1.2 Access control machine (ACM) and scheduling support functions	36
4.1.3 Connection-mode, connectionless-mode data transfer and DL service.....	37
4.2 Services provided by the DL	37
4.2.1 Overview	37
4.2.2 QoS.....	37
4.3 Structure and definition of DL-addresses	38
4.3.1 General	38
4.3.2 MAC ID address	38
4.3.3 Generic tag address	39
4.3.4 Fixed tag address	39
4.4 Services assumed from the PhL.....	40
4.4.1 General requirements	40
4.4.2 Data encoding rules.....	41
4.4.3 DLL to PhL interface	41
4.5 Functional classes	43
5 General structure and encoding of PhIDUs and DLPDUs and related elements of procedure	43
5.1 Overview	43
5.2 Media access procedure	43
5.3 DLPDU structure and encoding	47
5.3.1 General	47
5.3.2 DLPDU components	47
5.3.3 Preamble	47
5.3.4 Start and end delimiters.....	47
5.3.5 DLPDU octets and ordering	47
5.3.6 Source MAC ID.....	48
5.3.7 Lpackets field	48
5.3.8 Frame check sequence (FCS).....	48

5.3.9	Null DLPDU	51
5.3.10	Abort DLPDU	51
5.4	Lpacket components	51
5.4.1	General Lpacket structure	51
5.4.2	Size	51
5.4.3	Control	52
5.4.4	Generic tag Lpackets	52
5.4.5	Fixed tag Lpackets	53
5.5	DLPDU procedures	53
5.5.1	General	53
5.5.2	Sending scheduled DLPDUs	54
5.5.3	Sending unscheduled DLPDUs	54
5.5.4	Receiving DLPDUs	54
5.6	Summary of DLL support services and objects	54
6	Specific DLPDU structure, encoding and procedures	56
6.1	Modeling language	56
6.1.1	State machine description	56
6.1.2	Use of DLL- prefix	57
6.1.3	Data types	57
6.2	DLS user services	58
6.2.1	General	58
6.2.2	Connected mode and connectionless mode transfer service	58
6.2.3	Queue maintenance service	60
6.2.4	Tag filter service	60
6.2.5	Link synchronization service	61
6.2.6	Synchronized parameter change service	61
6.2.7	Event reports service	62
6.2.8	Bad FCS service	63
6.2.9	Current moderator service	63
6.2.10	Power up and online services	64
6.2.11	Enable moderator service	64
6.2.12	Listen only service	64
6.3	Generic tag Lpacket	64
6.3.1	General	64
6.3.2	Structure of the generic-tag Lpacket	65
6.3.3	Sending and receiving the generic-tag Lpacket	65
6.4	Moderator Lpacket	65
6.4.1	General	65
6.4.2	Structure of the moderator Lpacket	65
6.4.3	Sending and receiving the moderator Lpacket	66
6.5	Time distribution Lpacket	67
6.5.1	General	67
6.5.2	Structure of the time distribution Lpacket	67
6.5.3	Sending and receiving the time distribution Lpacket	69
6.6	UCMM Lpacket	69
6.6.1	General	69
6.6.2	Structure of the UCMM Lpacket	69
6.6.3	Sending and receiving the UCMM Lpacket	69

6.7	Keeper UCMM Lpacket	69
6.7.1	General	69
6.7.2	Structure of the Keeper UCMM Lpacket	70
6.7.3	Sending and receiving the Keeper UCMM Lpacket	70
6.8	TUI Lpacket	70
6.8.1	General	70
6.8.2	Structure of the TUI Lpacket	70
6.8.3	Sending and receiving the TUI Lpacket	71
6.9	Link parameters Lpacket and tMinus Lpacket	71
6.9.1	General	71
6.9.2	Structure of link parameters and tMinus Lpackets	72
6.9.3	Sending and receiving the tMinus and Link parameters Lpackets	72
6.10	I'm-alive Lpacket	73
6.10.1	General	73
6.10.2	Structure of the I'm-alive Lpacket	73
6.10.3	Sending and receiving I'm Alive	73
6.10.4	I'm alive state processing	73
6.11	Ping Lpackets	74
6.11.1	General	74
6.11.2	Structure of the ping Lpackets	75
6.11.3	Sending and receiving the ping Lpackets	75
6.12	WAMI Lpacket	76
6.12.1	General	76
6.12.2	Structure of the WAMI Lpacket	76
6.12.3	Sending and receiving the WAMI Lpacket	76
6.13	Debug Lpacket	76
6.14	IP Lpacket	77
6.15	Ethernet Lpacket	77
7	Objects for station management	77
7.1	General	77
7.2	ControlNet™ object	79
7.2.1	Overview	79
7.2.2	Class attributes	79
7.2.3	Instance attributes	79
7.2.4	Common services	87
7.2.5	Class specific services	88
7.2.6	Behavior	89
7.2.7	Module status indicator	89
7.3	Keeper object	90
7.3.1	Overview	90
7.3.2	Revision history	90
7.3.3	Class attributes	90
7.3.4	Instance attributes	90
7.3.5	Common services	98
7.3.6	Class specific services	99
7.3.7	Service error codes	105
7.3.8	Behavior	106
7.3.9	Miscellaneous notes	107
7.3.10	Keeper power up sequence	108

7.4	Scheduling object	113
7.4.1	Overview	113
7.4.2	Class attributes	114
7.4.3	Instance attributes	114
7.4.4	Common services	115
7.4.5	Class specific services	117
7.4.6	Typical scheduling session	123
7.5	TCP/IP Interface object	124
7.5.1	Overview	124
7.5.2	Revision history	124
7.5.3	Class attributes	125
7.5.4	Instance attributes	125
7.5.5	Common services	139
7.5.6	Class specific services	141
7.5.7	Behavior	142
7.5.8	Address Conflict Detection (ACD)	144
7.6	Ethernet link object	150
7.6.1	Overview	150
7.6.2	Revision history	150
7.6.3	Class attributes	150
7.6.4	Instance attributes	151
7.6.5	Common services	161
7.6.6	Class specific services	163
7.6.7	Behavior	163
7.7	DeviceNet™ object	164
7.7.1	Overview	164
7.7.2	Revision history	164
7.7.3	Class attributes	165
7.7.4	Instance attributes	165
7.7.5	Common services	172
7.7.6	Class specific services	173
7.8	Connection configuration object (CCO)	173
7.8.1	Overview	173
7.8.2	Revision history	174
7.8.3	Class attributes	174
7.8.4	Instance attributes	176
7.8.5	Connection Configuration Object change control	185
7.8.6	Common services	185
7.8.7	Class specific services	192
7.8.8	Behavior	195
7.9	DLR object	196
7.9.1	Overview	196
7.9.2	Revision history	196
7.9.3	Class attributes	197
7.9.4	Instance attributes	197
7.9.5	Common services	208
7.9.6	Class specific services	212
7.10	QoS object	213
7.10.1	Overview	

7.10.2	Revision History	213
7.10.3	Class attributes	213
7.10.4	Instance Attributes.....	214
7.10.5	Common services	215
7.10.6	Get_Attributes_All response (class level).....	216
7.11	Port object	216
7.11.1	Overview	216
7.11.2	Revision History	216
7.11.3	Class attributes	217
7.11.4	Instance attributes	217
7.11.5	Common services	223
7.12	PRP/HSR Protocol object.....	224
7.12.1	Overview	224
7.12.2	Revision history	224
7.12.3	Class attributes	225
7.12.4	Instance attributes	225
7.12.5	Common Services	232
7.13	PRP/HSR Nodes Table object.....	234
7.13.1	Overview	234
7.13.2	Revision history	234
7.13.3	Class attributes	234
7.13.4	Instance attributes	234
7.13.5	Common services	237
8	Other DLE elements of procedure.....	237
8.1	Network attachment monitor (NAM).....	237
8.1.1	General	237
8.1.2	Default parameters	238
8.1.3	Auto-addressing	239
8.1.4	Valid MAC IDs	239
8.1.5	State machine description	239
8.2	Calculating link parameters.....	245
8.2.1	Link parameters.....	245
8.2.2	Conditions affecting link parameters	246
8.2.3	Moderator change.....	246
8.2.4	NUT timing	246
8.2.5	Slot timing	248
8.2.6	Blanking	248
8.2.7	Example implementation.....	249
9	Detailed specification of DL components	253
9.1	General.....	253
9.2	Access control machine (ACM)	253
9.3	TxLLC.....	273
9.4	RxLLC	277
9.5	Transmit machine (TxM)	281
9.6	Receive machine (RxM)	285
9.7	Serializer	291
9.8	Deserializer	293
9.8.1	Octet construction	293
9.8.2	FCS checking	293

9.8.3	End of DLPDU processing	294
9.9	DLL management.....	294
10	Device Level Ring (DLR) protocol	296
10.1	General.....	296
10.2	Support for Multiple DLR Ring Pairs.....	296
10.3	Supported topologies	297
10.4	Overview of DLR operation	298
10.4.1	Normal operation	298
10.4.2	Link failures	300
10.5	Classes of DLR implementation	301
10.6	DLR behavior.....	302
10.6.1	DLR variables.....	302
10.6.2	Ring supervisor	302
10.6.3	Ring node.....	305
10.6.4	Sign on process.....	306
10.6.5	Neighbor check process	307
10.7	Implementation requirements.....	307
10.7.1	Embedded switch requirements and recommendations	307
10.7.2	DLR implementation requirements	308
10.7.3	IEC 61588 and CP 2/2.1 considerations.....	309
10.7.4	IEEE 802.1D/IEEE 802.1Q STP/RSTP/MSTP considerations	309
10.8	Using non-DLR nodes in the ring network	309
10.8.1	General considerations.....	309
10.8.2	Non-DLR end devices	310
10.8.3	Non-DLR switches	310
10.9	Redundant gateway devices on DLR network.....	313
10.9.1	General	313
10.9.2	Supported topologies.....	313
10.9.3	Redundant gateway capable device.....	314
10.9.4	Redundant gateway device behavior	314
10.10	DLR messages	317
10.10.1	General	317
10.10.2	Common frame header	318
10.10.3	Beacon frame	319
10.10.4	Neighbor_Check request	319
10.10.5	Neighbor_Check_response.....	320
10.10.6	Link_Status/Neighbor_Status.....	320
10.10.7	Locate_Fault.....	321
10.10.8	Announce	321
10.10.9	Sign_On	321
10.10.10	Advertise	322
10.10.11	Flush_Tables.....	322
10.10.12	Learning_Update	323
10.11	State diagrams and state-event-action matrices.....	323
10.11.1	Beacon-based ring node.....	323
10.11.2	Announce-based ring node	330
10.11.3	Ring supervisor	334
10.11.4	Redundant gateway	349

10.12	Performance analysis	354
10.12.1	General	354
10.12.2	Redundant gateway switchover performance	358
11	PRP and HSR redundancy protocols	359
11.1	General.....	359
11.2	PRP overview	360
11.2.1	General	360
11.2.2	Address Conflict Detection (ACD).....	361
11.3	HSR overview	362
Annex A (normative)	Indicators and switches.....	364
A.1	Purpose	364
A.2	Indicators.....	364
A.2.1	General indicator requirements.....	364
A.2.2	Common indicator requirements	364
A.2.3	Fieldbus specific indicator requirements – option 1	366
A.2.4	Fieldbus specific indicator requirements – option 2	370
A.2.5	Fieldbus specific indicator requirements – option 3.....	374
A.3	Switches	378
A.3.1	Common switch requirements	378
A.3.2	Fieldbus specific switch requirements – option 1	378
A.3.3	Fieldbus specific switch requirements – option 2	378
A.3.4	Fieldbus specific switch requirements – option 3	379
Bibliography		380
Figure 1	– Data-link layer internal architecture.....	36
Figure 2	– Relationships of DLSAPs, DLSAP-addresses, and group DL-addresses.....	38
Figure 3	– Basic structure of a MAC ID address.....	39
Figure 4	– Basic structure of a generic tag address	39
Figure 5	– Basic structure of a fixed tag address	39
Figure 6	– M_symbols and Manchester encoding at 5 MHz	41
Figure 7	– NUT structure	44
Figure 8	– Media access during scheduled time	45
Figure 9	– Media access during unscheduled time	46
Figure 10	– DLPDU format.....	47
Figure 11	– Aborting a DLPDU during transmission	51
Figure 12	– Lpacket format	51
Figure 13	– Generic tag Lpacket format	52
Figure 14	– Fixed tag Lpacket format.....	53
Figure 15	– Goodness parameter of TimeDist_Lpacket	68
Figure 16	– Example I'm alive processing algorithm.....	74
Figure 17	– Keeper CRC algorithm	96
Figure 18	– Keeper object power-up state diagram	109
Figure 19	– Keeper object operating state diagram	110
Figure 20	– Synchronized network change processing	113
Figure 21	– State transition diagram for TCP/IP Interface object.....	143
Figure 22	– State transition diagram for TCP/IP Interface object.....	144

Figure 23 – ACD Behavior	146
Figure 24 – State transition diagram for Ethernet Link object	164
Figure 25 – Connection configuration object edit flowchart.....	196
Figure 26 – NAM state machine	238
Figure 27 – Devices with Multiple DLR Ring Pairs.....	297
Figure 28 – DLR rings connected to switches.....	298
Figure 29 – Normal operation of a DLR network.....	299
Figure 30 – Beacon and Announce frames.....	299
Figure 31 – Link failure	300
Figure 32 – Network reconfiguration after link failure	301
Figure 33 – Neighbor Check process	307
Figure 34 – Unsupported topology – example 1	311
Figure 35 – Unsupported topology – example 2	311
Figure 36 – DLR ring connected to switches through redundant gateways	313
Figure 37 – DLR redundant gateway capable device.....	314
Figure 38 – Advertise frame.....	316
Figure 39 – State transition diagram for Beacon frame based non-supervisor ring node.....	324
Figure 40 – State transition diagram for Announce frame based non-supervisor ring node	330
Figure 41 – State transition diagram for ring supervisor	335
Figure 42 – State transition diagram for redundant gateway.....	350
Figure 43 – PRP network	360
Figure 44 – Directly Attached SANs	361
Figure 45 – Virtual DANs	362
Figure 46 – HSR network.....	362
Figure A.1 – Non redundant network status indicator labeling	370
Figure A.2 – Redundant network status indicator labeling	370
Figure A.3 – Network status indicator state diagram	373
Figure A.4 – Examples of multiple network status indicators	373
Table 1 – Format of attribute tables	34
Table 2 – Data-link layer components	35
Table 3 – MAC ID addresses allocation	39
Table 4 – Fixed tag service definitions	40
Table 5 – Data encoding rules	41
Table 6 – M Data symbols	42
Table 7 – Truth table for ph_status_indication.....	42
Table 8 – FCS length, polynomials and constants	49
Table 9 – DLL support services and objects.....	55
Table 10 – Elementary data types.....	58
Table 11 – DLL events.....	63
Table 12 – Time distribution priority	68
Table 13 – Format of the TUI Lpacket.....	71

Table 14 – ControlNet object class attributes	79
Table 15 – ControlNet object instance attributes	80
Table 16 – TUI status flag bits	84
Table 17 – Mac_ver bits	85
Table 18 – Channel state bits	85
Table 19 – ControlNet object common services	87
Table 20 – ControlNet object class specific services	88
Table 21 – Keeper object revision history	90
Table 22 – Keeper object class attributes	90
Table 23 – Keeper object instance attributes	91
Table 24 – Keeper operating state definitions	94
Table 25 – Port status flag bit definitions	94
Table 26 – TUI status flag bits	95
Table 27 – Keeper attributes	97
Table 28 – Memory requirements (in octets) for the Keeper attributes	98
Table 29 – Keeper object common services	98
Table 30 – Keeper object class specific services	100
Table 31 – Service error codes	101
Table 32 – Wire order format of the TUI Lpacket	105
Table 33 – Service error codes	106
Table 34 – Keeper object operating states	107
Table 35 – Keeper object state event matrix	111
Table 36 – Scheduling object class attributes	114
Table 37 – Scheduling object instance attributes	115
Table 38 – Scheduling object common services	115
Table 39 – Status error descriptions for Create	116
Table 40 – Status error descriptions for Delete and Kick_Timer	117
Table 41 – Scheduling object class specific services	117
Table 42 – Status error descriptions for Read	119
Table 43 – Status error descriptions for Conditional_Write	120
Table 44 – Status error descriptions for Forced_Write	120
Table 45 – Status error descriptions for Change_Start	121
Table 46 – Status error descriptions for Break_Connections	122
Table 47 – Status error descriptions for Change_Complete	122
Table 48 – Status error descriptions for Restart_Connections	123
Table 49 – Revision history	124
Table 50 – TCP/IP Interface object class attributes	125
Table 51 – TCP/IP Interface object instance attributes	126
Table 52 – Status bits	129
Table 53 – Configuration capability bits	130
Table 54 – Configuration control bits	131
Table 55 – Example path	132
Table 56 – Interface configuration components	133

Table 57 – Alloc control values	135
Table 58 – AcdActivity values	136
Table 59 – ArpPdu – ARP Response PDU in binary format	136
Table 60 – Admin Capability member bit definitions	137
Table 61 – Admin Capability member bit definitions	138
Table 62 – TCP/IP Interface object common services	139
Table 63 – Get_Attributes_All reply format.....	140
Table 64 – TCP/IP Interface object class specific services.....	141
Table 65 – Set_Port_Admin_State service request parameters	141
Table 66 – Set_Protocol_Admin_State service request parameters.....	142
Table 67 – Class specific error codes	142
Table 68 – Ethernet link object revision history	150
Table 69 – Ethernet link object class attributes	151
Table 70 – Ethernet link object instance attributes	151
Table 71 – Interface flags bits.....	157
Table 72 – Control bits.....	158
Table 73 – Interface type	159
Table 74 – Interface state	159
Table 75 – Admin state	159
Table 76 – Capability Bits	160
Table 77 – Ethernet Link object common services.....	161
Table 78 – Get_Attributes_All reply format.....	162
Table 79 – Ethernet Link object class specific services	163
Table 80 – DeviceNet object revision history.....	165
Table 81 – DeviceNet object class attributes.....	165
Table 82 – DeviceNet object instance attributes.....	166
Table 83 – Bit rate attribute values	168
Table 84 – BOI attribute values.....	169
Table 85 – Diagnostic counters bit description	171
Table 86 – DeviceNet object common services	172
Table 87 – Reset service parameter	172
Table 88 – Reset service parameter values	172
Table 89 – DeviceNet object class specific services.....	173
Table 90 – Connection configuration object revision history	174
Table 91 – Connection configuration object class attributes	174
Table 92 – Format number values.....	176
Table 93 – Connection configuration object instance attributes	176
Table 94 – Originator connection status values.....	180
Table 95 – Target connection status values	181
Table 96 – Connection flags	181
Table 97 – I/O mapping formats.....	183
Table 98 – Services valid during a change operation	185
Table 99 – Connection configuration object common services.....	186

Table 100 – Get_Attributes_All Response – class level.....	186
Table 101 – Get_Attributes_All response – instance level.....	187
Table 102 – Set_Attributes_All error codes.....	188
Table 103 – Set_Attributes_All request.....	189
Table 104 – Create request parameters.....	190
Table 105 – Create error codes.....	191
Table 106 – Delete error codes.....	191
Table 107 – Restore error codes.....	192
Table 108 – Connection configuration object class specific services.....	192
Table 109 – Change_Start error codes.....	193
Table 110 – Get_Status service parameter.....	193
Table 111 – Get_Status service response.....	194
Table 112 – Get_Status service error codes.....	194
Table 113 – Change_Complete service parameter.....	194
Table 114 – Change_Complete service error codes.....	195
Table 115 – Audit_Changes service parameter.....	195
Table 116 – Audit_Changes service error codes.....	195
Table 117 – Revision history.....	197
Table 118 – DLR object class attributes.....	197
Table 119 – DLR object instance attributes.....	198
Table 120 – Network Status values.....	201
Table 121 – Ring Supervisor Status values.....	201
Table 122 – Capability flags.....	205
Table 123 – Redundant Gateway Status values.....	207
Table 124 – DLR object common services.....	208
Table 125 – Get_Attributes_All Response – Object Revision 1, non supervisor device.....	209
Table 126 – Get_Attributes_All Response – Object Revision 1, supervisor-capable device.....	210
Table 127 – Get_Attributes_All Response – Object Revision 2, non supervisor device.....	210
Table 128 – Get_Attributes_All Response – All other cases.....	211
Table 129 – DLR object class specific services.....	212
Table 130 – QoS object revision history.....	213
Table 131 – QoS object class attributes.....	214
Table 132 – QoS object instance attributes.....	214
Table 133 – Default DCSP values and usages.....	215
Table 134 – QoS object common services.....	216
Table 135 – Port object revision history.....	217
Table 136 – Port object class attributes.....	217
Table 137 – Port object instance attributes.....	218
Table 138 – Port Type and associated Link Object classes and Port Type Name values.....	220
Table 139 – Port Routing Capabilities attribute bit definitions.....	222
Table 140 – Port object common services.....	223
Table 141 – Get_Attributes_All response– class level.....	224

Table 142 – Get_Attributes_All response– instance level	224
Table 143 – Revision history	225
Table 144 – Class attributes	225
Table 145 – Instance attributes	226
Table 146 – Node Type	229
Table 147 – Switching Node	230
Table 148 – HSR Mode	230
Table 149 – RedBox ID	231
Table 150 – PRP/HSR Protocol object common services	232
Table 151 – Get_Attributes_All response	233
Table 152 – Revision history	234
Table 153 – Class attributes	234
Table 154 – Instance attributes	235
Table 155 – Remote Node Type	236
Table 156 – PRP/HSR Nodes Tables object common services	237
Table 157 – Get_Attributes_All response	237
Table 158 – NAM states	238
Table 159 – Default link parameters	239
Table 160 – PhL timing characteristics	246
Table 161 – DLR variables	302
Table 162 – DLR Link speed and duplex requirements	308
Table 163 – Redundant gateway variables	315
Table 164 – MAC addresses for DLR messages	318
Table 165 – IEEE 802.1Q common frame header format	318
Table 166 –DLR message payload fields	318
Table 167 – DLR frame types	319
Table 168 – Format of the Beacon frame	319
Table 169 – Ring State values	319
Table 170 – Format of the Neighbor_Check request	320
Table 171 – Format of the Neighbor_Check response	320
Table 172 – Format of the Link_Status/Neighbor_Status frame	320
Table 173 – Link/Neighbor status values	321
Table 174 – Format of the Locate_Fault frame	321
Table 175 – Format of the Announce frame	321
Table 176 – Format of the Sign_On frame	322
Table 177 – Format of the Advertise frame	322
Table 178 – Gateway state values	322
Table 179 – Format of the Flush_Tables frame	323
Table 180 – Format of the Learning_Update frame	323
Table 181 – Parameter values for Beacon frame based non-supervisor ring node	324
Table 182 – LastBcnRcvPort bit definitions	325
Table 183 – State-event-action matrix for Beacon frame based non-supervisor ring node	325

Table 184 – Parameter values for Announce frame based non-supervisor ring node	331
Table 185 – State-event-action matrix for Announce frame based non-supervisor ring node	332
Table 186 – Parameter values for ring supervisor node	336
Table 187 – LastBcnRcvPort bit definitions	337
Table 188 – State-event-action matrix for ring supervisor node	337
Table 189 – Parameter values for redundant gateway node	350
Table 190 – State-event-action matrix for redundant gateway node	352
Table 191 – Parameters/assumptions for example performance calculations	354
Table 192 – Example ring configuration parameters and performance	358
Table 193 – Variables for performance analysis	358
Table A.1 – Module status indicator	365
Table A.2 – Time Sync status indication	366
Table A.3 – Network status indicators	368
Table A.4 – Network status indicator	372
Table A.5 – Network status indicator	375
Table A.6 – Combined module/network status indicator	376
Table A.7 – I/O status indicator	377
Table A.8 – Bit rate switch encoding	379

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 4-2: Data-link layer protocol specification –
Type 2 elements****FOREWORD**

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

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

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

International Standard IEC 61158-4-2 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This fourth edition cancels and replaces the third edition published in 2014. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- clarifications of ControlNet object in 7.2;
- extensions and clarifications of TCP/IP interface object in 7.5;
- extensions and clarifications of Ethernet Link object in 7.6;
- clarifications of DeviceNet object in 7.7;
- extensions and clarifications of CCO object in 7.8;
- extensions and clarifications of DLR object in 7.9;
- extensions and clarifications of Port object in 7.11;
- addition of PRP/HSR Protocol and PRP/HSR Nodes Table objects in 7.12 and 7.13;
- extensions and clarifications of DLR protocol in Clause 10;
- addition of PRP/HSR protocol mapping in Clause 11;
- update of indicator behaviour in A.2.2 and A.2.4;
- miscellaneous editorial corrections.

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

FDIS	Report on voting
65C/946/FDIS	65C/955/RVD

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

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

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

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

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

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

INTRODUCTION

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

The data-link protocol provides the data-link service by making use of the services available from the physical layer. The primary aim of this document is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer data-link entities (DLEs) 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 implementers 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;
- d) as a refinement to the understanding of time-critical communications within OSI.

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

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents given in several subclauses as indicated in the table below. These patents are held by their respective inventors under license to ODVA, Inc:

US 5,400,331	[ODVA]	Communication network interface with screeners for incoming messages	Subclause 3.4, Clauses 4 to 9
US 5,471,461	[ODVA]	Digital communication network with a moderator station election process	
US 5,491,531	[ODVA]	Media access controller with a shared class message delivery capability	
US 5,493,571	[ODVA]	Apparatus and method for digital communications with improved delimiter detection	
US 5,537,549	[ODVA]	Communication network with time coordinated station activity by time slot and periodic interval number	
US 5,553,095	[ODVA]	Method and apparatus for exchanging different classes of data during different time intervals	Clause 10
US 8,244,838	[ODVA]	Industrial controller employing the network ring topology	

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

ODVA and the holders of these patent rights have assured the IEC that ODVA is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of ODVA and the holders of these patent rights is registered with IEC. Information may be obtained from:

[ODVA] ODVA, Inc.
 2370 East Stadium Boulevard #1000
 Ann Arbor, Michigan 48104
 USA
 Attention: Office of the Executive Director
 e-mail: odva@odva.org

ISO (www.iso.org/patents) and IEC (<http://patents.iec.ch>) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 4-2: Data-link layer protocol specification – Type 2 elements

1 Scope

1.1 General

The data-link layer provides basic time-critical messaging communications between devices in an automation environment.

This protocol provides communication opportunities to all participating data-link entities, sequentially and in a cyclic synchronous manner. Foreground scheduled access is available for time-critical activities together with background unscheduled access for less critical activities.

Deterministic and synchronized transfers can be provided at cyclic intervals up to 1 ms and device separations of 25 km. This performance is adjustable dynamically and on-line by re-configuring the parameters of the local link whilst normal operation continues. By similar means, DL connections and new devices may be added or removed during normal operation.

This protocol provides means to maintain clock synchronization across an extended link with a precision better than 10 µs.

This protocol optimizes each access opportunity by concatenating multiple DLSDUs and associated DLPCI into a single DLPDU, thereby improving data transfer efficiency for data-link entities that actively source multiple streams of data.

The maximum system size is an unlimited number of links of 99 nodes, each with 255 DLSAP-addresses. Each link has a maximum of 2^{24} related peer and publisher DLCEPs.

1.2 Specifications

This document specifies

- a) procedures 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;
- b) the structure of the fieldbus DLPDUs used for the transfer of data and control information by the protocol of this document, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of

- a) the interactions between peer DL-entities (DLEs) through the exchange of fieldbus DLPDUs;
- b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;
- c) the interactions between a DLS-provider and a Ph-service provider in the same system through the exchange of Ph-service primitives.

1.4 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 capabilities of an implementation, and thus its applicability to various time-critical communications needs.

1.5 Conformance

This document also specifies conformance requirements for systems implementing these procedures. This document does not contain tests to demonstrate compliance with such requirements.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

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

IEC 61158-3-2:2014, *Industrial communication networks – Fieldbus specifications – Part 3-2: Data-link layer service definition – Type 2 elements*
IEC 61158-3-2:2014/AMD1:2019

IEC 61158-5-2:2019, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition – Type 2 elements*

IEC 61158-6-2:2019, *Industrial communication networks – Fieldbus specifications – Part 6-2: Application layer protocol specification – Type 2 elements*

IEC 61588:2009, *Precision clock synchronization protocol for networked measurement and control systems*

IEC 61784-3-2, *Industrial communication networks – Profiles – Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2*

IEC 62026-3:2014, *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 3: DeviceNet*

IEC 62439-3:2012¹, *Industrial communication networks – High availability automation networks – Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)*

¹ A newer edition of this standard has been published, but only the cited edition applies.

ISO/IEC 3309², *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures – Frame structure*

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

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*

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

ISO 11898:1993³, *Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication*

IEEE Std 802.1D-2004, *IEEE standard for local and metropolitan area networks – Media Access Control (MAC) bridges*, available at <http://www.ieee.org> [viewed 2018-09-03]

IEEE Std 802.1Q-2005⁴, *IEEE standard for local and metropolitan area networks – Virtual bridged local area networks*, available at <http://www.ieee.org> [viewed 2018-09-03]

IEEE Std 802.3-2015, *IEEE Standard for Ethernet*, available at <http://www.ieee.org> [viewed 2018-09-03]

IETF RFC 951, *Bootstrap Protocol (BOOTP)*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 1643, *Definitions of Managed Objects for the Ethernet-like Interface Types*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 2131, *Dynamic Host Configuration Protocol*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 2132, *DHCP Options and BOOTP Vendor Extensions*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 2863, *The Interfaces Group MIB*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*, available at <http://www.ietf.org> [viewed 2018-09-03]

² This standard has been withdrawn.

³ A newer edition of this standard has been published, but only the cited edition applies.

⁴ A newer edition of this standard has been published, but only the cited edition applies.

IETF RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*, available at <http://www.ietf.org> [viewed 2018-09-03]

IETF RFC 5227:2008, *IPv4 Address Conflict Detection*, available at <http://www.ietf.org> [viewed 2018-09-03]

3 Terms, definitions, symbols, abbreviated terms and conventions

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

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

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

3.1 Reference model terms and definitions

This document is based in part on the concepts developed in ISO/IEC 7498-1 and ISO/IEC 7498-3, and makes use of the following terms defined therein.

3.1.1	called-DL-address	[ISO/IEC 7498-3]
3.1.2	calling-DL-address	[ISO/IEC 7498-3]
3.1.3	centralized multi-end-point-connection	[ISO/IEC 7498-1]
3.1.4	correspondent (N)-entities correspondent DL-entities (N=2) correspondent Ph-entities (N=1)	[ISO/IEC 7498-1]
3.1.5	demultiplexing	[ISO/IEC 7498-1]
3.1.6	DL-address	[ISO/IEC 7498-3]
3.1.7	DL-address-mapping	[ISO/IEC 7498-1]
3.1.8	DL-connection	[ISO/IEC 7498-1]
3.1.9	DL-connection-end-point	[ISO/IEC 7498-1]
3.1.10	DL-connection-end-point-identifier	[ISO/IEC 7498-1]
3.1.11	DL-connection-mode transmission	[ISO/IEC 7498-1]
3.1.12	DL-connectionless-mode transmission	[ISO/IEC 7498-1]
3.1.13	DL-data-sink	[ISO/IEC 7498-1]
3.1.14	DL-data-source	[ISO/IEC 7498-1]
3.1.15	DL-duplex-transmission	[ISO/IEC 7498-1]
3.1.16	DL-facility	[ISO/IEC 7498-1]
3.1.17	DL-local-view	[ISO/IEC 7498-3]
3.1.18	DL-name	[ISO/IEC 7498-3]
3.1.19	DL-protocol	[ISO/IEC 7498-1]
3.1.20	DL-protocol-connection-identifier	[ISO/IEC 7498-1]
3.1.21	DL-protocol-control-information	[ISO/IEC 7498-1]
3.1.22	DL-protocol-data-unit	[ISO/IEC 7498-1]
3.1.23	DL-protocol-version-identifier	[ISO/IEC 7498-1]
3.1.24	DL-relay	[ISO/IEC 7498-1]

3.1.25	DL-service-connection-identifier	[ISO/IEC 7498-1]
3.1.26	DL-service-data-unit	[ISO/IEC 7498-1]
3.1.27	DL-simplex-transmission	[ISO/IEC 7498-1]
3.1.28	DL-subsystem	[ISO/IEC 7498-1]
3.1.29	DL-user-data	[ISO/IEC 7498-1]
3.1.30	flow control	[ISO/IEC 7498-1]
3.1.31	layer-management	[ISO/IEC 7498-1]
3.1.32	multiplexing	[ISO/IEC 7498-3]
3.1.33	naming-(addressing)-authority	[ISO/IEC 7498-3]
3.1.34	naming-(addressing)-domain	[ISO/IEC 7498-3]
3.1.35	naming-(addressing)-subdomain	[ISO/IEC 7498-3]
3.1.36	(N)-entity DL-entity Ph-entity	[ISO/IEC 7498-1]
3.1.37	(N)-interface-data-unit DL-service-data-unit (N=2) Ph-interface-data-unit (N=1)	[ISO/IEC 7498-1]
3.1.38	(N)-layer DL-layer (N=2) Ph-layer (N=1)	[ISO/IEC 7498-1]
3.1.39	(N)-service DL-service (N=2) Ph-service (N=1)	[ISO/IEC 7498-1]
3.1.40	(N)-service-access-point DL-service-access-point (N=2) Ph-service-access-point (N=1)	[ISO/IEC 7498-1]
3.1.41	(N)-service-access-point-address DL-service-access-point-address (N=2) Ph-service-access-point-address (N=1)	[ISO/IEC 7498-1]
3.1.42	peer-entities	[ISO/IEC 7498-1]
3.1.43	Ph-interface-control-information	[ISO/IEC 7498-1]
3.1.44	Ph-interface-data	[ISO/IEC 7498-1]
3.1.45	primitive name	[ISO/IEC 7498-3]
3.1.46	reassembling	[ISO/IEC 7498-1]
3.1.47	recombining	[ISO/IEC 7498-1]
3.1.48	reset	[ISO/IEC 7498-1]
3.1.49	responding-DL-address	[ISO/IEC 7498-3]
3.1.50	routing	[ISO/IEC 7498-1]
3.1.51	segmenting	[ISO/IEC 7498-1]
3.1.52	sequencing	[ISO/IEC 7498-1]
3.1.53	splitting	[ISO/IEC 7498-1]
3.1.54	synonymous name	[ISO/IEC 7498-3]
3.1.55	systems-management	[ISO/IEC 7498-1]

3.2 Service convention terms and definitions

This document also makes use of the following terms defined in 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 Common terms and definitions

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

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

3.3.1

**DL-segment
link
local link**

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

3.3.2

DLSAP

distinctive point at which DL-services are provided by a single DL-entity to a single higher-layer entity

Note 1 to entry: Definition is provided in a generic manner “(N)-SAP” in ISO/IEC 7498-1.

3.3.3

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, each of a single DLS-user

Note 1 to entry: This terminology is chosen because ISO/IEC 7498-3 does not permit the use of the term DLSAP-address to designate more than a single DLSAP at a single DLS-user.

3.3.4

(individual) DLSAP-address

DL-address that designates only one DLSAP within the extended link

Note 1 to entry: A single DL-entity may have multiple DLSAP-addresses associated with a single DLSAP.

3.3.5

extended link

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 1 to entry: An extended link may be composed of just a single link.

3.3.6

frame

denigrated synonym for DLPDU

3.3.7

group DL-address

DL-address that potentially designates more than one DLSAP within the extended link

Note 1 to entry: 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.

3.3.8

node

single DL-entity as it appears on one local link

3.3.9

receiving DLS-user

DL-service user that acts as a recipient of DL-user-data

Note 1 to entry: A DL-service user can be concurrently both a sending and receiving DLS-user.

3.3.10

sending DLS-user

DL-service user that acts as a source of DL-user-data

3.4 Additional Type 2 definitions

3.4.1

actual packet interval

API

measure of how frequently a specific connection produces its Lpacket data

3.4.2

allocate

take a resource from a common area and assign that resource for the exclusive use of a specific entity

3.4.3

application

function or data structure for which data is consumed or produced

3.4.4

application objects

multiple object classes that manage and provide the run time exchange of messages across the network and within the device

3.4.5

attribute

description of an externally visible characteristic or feature of an object

Note 1 to entry: The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behavior of an object. Attributes are divided into class attributes and instance attributes.

3.4.6

behavior

indication of how the object responds to particular events

Note 1 to entry: Its description includes the relationship between attribute values and services.

3.4.7

bit

unit of information consisting of a 1 or a 0

Note 1 to entry: This is the smallest data unit that can be transmitted.

3.4.8

blanking or blanking time

length of time required after transmitting before the node is allowed to receive

3.4.9

client

- 1) object which uses the services of another (server) object to perform a task
- 2) initiator of a message to which a server reacts

3.4.10

communication objects

components that manage and provide run time exchange of messages across the network such as the ControlNet object, the Unconnected Message Manager (UCMM)

3.4.11

connection

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

3.4.12

connection ID

CID

identifier assigned to a transmission, associated with a particular connection between producers and consumers, that identifies a specific piece of application information

Note 1 to entry: Within the data link this is a generic tag.

3.4.13

cyclic redundancy check

CRC

residual value computed from an array of data and used as a representative signature for the array

3.4.14**cyclic**

term used to describe events which repeat in a regular and repetitive manner

3.4.15**deafness**

situation where the node cannot hear the moderator DLPDU but can hear other link traffic

3.4.16**device**

physical hardware connection to the link

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

3.4.17**DLPDU**

data-link protocol data unit

Note 1 to entry: A DLPDU consists of a source MAC ID, zero or more Lpackets and an FCS as transmitted or received by an associated PhE.

3.4.18**end delimiter**

unique set of M_symbols that identifies the end of a DLPDU

3.4.19**error**

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

3.4.20**extended link**

links connected by DL-routers (sometimes called a local area network)

3.4.21**FCS error**

error that occurs when the computed frame check sequence value after reception of all the octets in a DLPDU does not match the expected residual

3.4.22**fixed tag**

two octet identifier (tag) which identifies a specific service to be performed by either

- a) that receiving node on the local link which has a specified MAC ID, or
- b) all receiving nodes on the local link

Note 1 to entry: Identification of the target node(s) is included in the two octet tag.

3.4.23**frame**

denigrated synonym for DLPDU

3.4.24**Frame Check Sequence****FCS**

redundant data derived from a block of data within a DLPDU (frame), using a hash function, and stored or transmitted together with the block of data, in order to detect data corruption

3.4.25

generic tag

three octet identifier (tag) which identifies a specific piece of application information

3.4.26

guardband

time slot allocated for the transmission of the moderator DLPDU

3.4.27

implicit token

mechanism that governs the right to transmit

Note 1 to entry: No actual token message is transmitted on the medium. Each node keeps track of the MAC ID of the node that it believes currently holds the right to transmit. The right to transmit is passed from node to node by keeping a record of the node that last transmitted. A slot time is used to allow a missing node to be skipped in the rotation.

3.4.28

implicit token register

register that contains the MAC ID of the node that holds the right to transmit

3.4.29

instance

actual physical occurrence of an object within a class, identifying one of many objects within the same object class

EXAMPLE California is an instance of the object class state.

Note 1 to entry: The terms object, instance, and object instance are used to refer to a specific instance.

3.4.30

instance attribute

attribute whose value is unique to an object instance and whose definition is shared by all instances of an object

3.4.31

instantiated

object that has been created in a device

3.4.32

Keeper

object responsible for distributing link configuration data to all nodes on the link

3.4.33

link

collection of nodes with unique MAC IDs, attached to Ph-segments connected by Ph-repeaters

3.4.34

little endian

model of memory organization which stores the least significant octet at the lowest address, and of transmission organization which transfers the least significant octet first on the medium

3.4.35

Lpacket

well-defined sub-portion of a DLPDU consisting of

- a) size and control information,
- b) a fixed tag or a generic tag, and
- c) DLS-user data or, when the tag has DL-significance, DL-data

3.4.36**M_symbol(s)**

symbols that represent the data bits and related information encoded and transmitted by the PhL

3.4.37**maximum scheduled node**

node with highest MAC ID that can use scheduled time on a link

3.4.38**maximum unscheduled node**

node with highest MAC ID that can use unscheduled time on a link

3.4.39**Message Router**

object within a node that distributes messaging requests to the appropriate application objects

3.4.40**moderator**

node with the lowest MAC ID that is responsible for transmitting the moderator DLPDU

3.4.41**moderator DLPDU**

DLPDU transmitted by the node with the lowest MAC ID for the purpose of synchronizing the nodes and distributing the link configuration parameters

3.4.42**multipoint connection**

connection from one node to many

Note 1 to entry: Multipoint connections allow messages from a single producer (publisher) to be received by many consumer (subscriber) nodes.

3.4.43**network**

series of nodes connected by some type of communication medium

Note 1 to entry: The connection paths between any pair of nodes can include repeaters, routers and gateways.

3.4.44**network access monitor****NAM**

component within a node that manages communication with a temporary node connected to the nodes local NAP

3.4.45**network access port****NAP**

PhL variant that allows a temporary node to be connected to the link by connection to the NAP of a permanent node

3.4.46**network address or node address**

node's address on the link

Note 1 to entry: This is also called MAC ID.

3.4.47

network status indicators

indicators on a node indicating the status of the Physical and Data Link Layers

3.4.48

network update time

NUT

repetitive time interval in which data can be sent on the link

3.4.49

node

logical connection to a local link, requiring a single MAC ID

Note 1 to entry: A single physical device may appear as many nodes on the same local link. For the purposes of this protocol, each node is considered to be a separate DLE.

3.4.50

non-concurrence

situation where a transmission is received from an unexpected MAC ID, which appears to violate the time based access protocol

Note 1 to entry: This may occur when a connection is made between two working links that are not synchronized with each other but who have the same configuration information.

3.4.51

non-data symbol

PhL symbol which violates the requirements of Manchester biphase L encoding

3.4.52

object

abstract representation of a particular component within a device

Note 1 to entry: An object can be

- 1) an abstract representation of a computer's capabilities. Objects can be composed of any or all of the following components:
 - a) data (information which changes with time);
 - b) configuration (parameters for behavior);
 - c) methods (things that can be done using data and configuration);
- 2) a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behavior.

3.4.53

object specific service

service defined by a particular object class to perform a required function which is not performed by a common service

Note 1 to entry: An object specific service is unique to the object class which defines it.

3.4.54

originator

client responsible for establishing a connection path to the target

3.4.55

permanent node

node whose connection to the network does not utilize the network access port (NAP) PhL variant

Note 1 to entry: This node may optionally support a NAP PhL variant to allow temporary nodes to connect to the network.

3.4.56**produce**

act of sending data to be received by a consumer

3.4.57**producer**

node that is responsible for sending data

3.4.58**redundant media**

system using more than one medium to help prevent communication failures

3.4.59**requested packet interval****RPI**

measure of how frequently the originating application requires the transmission of Lpackets or data packets from the target application

3.4.60**rogue**

node that has received a moderator DLPDU that disagrees with the link configuration currently used by this node

3.4.61**scheduled**

data transfers that occur in a deterministic and repeatable manner on predefined NUTs

3.4.62**server**

object which provides services to another (client) object

3.4.63**service**

operation or function than an object and/or object class performs upon request from another object and/or object class

Note 1 to entry: A set of common services is defined and provisions for the definition of object-specific services are provided. Object-specific services are those which are defined by a particular object class to perform a required function which is not performed by a common service.

3.4.64**slot time**

maximum time required for detecting an expected transmission

Note 1 to entry: Each node waits a slot time for each missing node during the implied token pass.

3.4.65**start delimiter**

unique set of M_symbols that identifies the beginning of a DLPDU

3.4.66**supernode**

DLE with a MAC ID of zero

Note 1 to entry: This DLE DL-address is reserved for special DLL functions

3.4.67

table unique identifier

TUI

unique reference code used by ControlNet and Keeper objects to reference a consistent set of link configuration attribute

3.4.68

tag

shorthand name for a specific piece of application information, two or three octets in length

3.4.69

target

end-node to which a connection is established

3.4.70

temporary node

same as transient node

3.4.71

tMinus

number of NUTs before a new set of link configuration parameters are to be used

3.4.72

tone

instant of time which marks the boundary between two NUTs

3.4.73

tool

executable software program which interacts with the user to perform some function

EXAMPLE Link scheduling software (LSS).

3.4.74

transaction id

field within a UCMM header that matches a response with the associated request, which a server echoes in the response message

3.4.75

transient node

node that is only intended to be connected to the network on a temporary basis using the NAP PhL medium connected to the NAP of a permanent node

3.4.76

Unconnected Message Manager

UCMM

component within a node that transmits and receives unconnected explicit messages and sends them directly to the Message Router object

3.4.77

unconnected service

messaging service which does not rely on the set up of a connection between devices before allowing information exchanges

3.4.78

unscheduled

data transfers that use the remaining allocated time in the NUT after the scheduled transfers have been completed

3.4.79**vendor ID**

identification of each product manufacturer/vendor by a unique number

Note 1 to entry: Vendor IDs are assigned by ODVA, Inc.

3.5 Type 2 symbols and abbreviated terms

ACD	Address Conflict Detection	
ACM	Access control machine	
BOOTP	Bootstrap Protocol	[IETF RFC 951]
CA	Clock accuracy	
CAN	Controller area network	[ISO 11898:1993]
COP	Connection originator password	
DANP	Doubly Attached Node operating PRP	
DHCP	Dynamic Host Configuration Protocol	[IETF RFC 2131]
DLR	Device level ring	
DSCP	DiffServ code point	
FCS	Frame Check Sequence	
HSR	High-availability Seamless Redundancy	[IEC 62439-3:2012]
IP	Internet Protocol	[IETF RFC 791]
LAN	Local Area Network	
LED	Light emitting diode.	
LRE	Link Redundancy Entity (associated with PRP)	
LSS	Link scheduling software	
MAC ID	MAC address of a node	
MSTP	Multiple spanning tree protocol	[IEEE 802.1Q]
NAM	Network attachment monitor	
NAP	Network access port	
ND	Non-data symbol	
PRP	Parallel Redundancy Protocol	[IEC 62439-3:2012]
NUT	Network update time	
PT	Programming terminal (a temporary network connection)	
PTP	Precision Time Protocol	[IEC 61588:2009]
QoS	Quality of service	
Rcv	Receive	
RedBox	PRP Redundancy Box (for attaching non-PRP nodes)	
RPI	Requested Lpacket interval	
RSTP	Rapid spanning tree protocol	[IEEE 802.1D]
Rx	Receive	
RxLLC	Receive logical link control	
RxM	Receive machine	
SAN	Singly Attached Node (in a PRP topology)	
SEM	State event matrix	
SMAX	MAC ID of the maximum scheduled node	
STD	State transition diagram, used to describe object behavior	
STP	Spanning tree protocol	[IEEE 802.1D]
TCP	Terminal Control Protocol	[IETF RFC 793]
TFTP	Trivial File Transfer Protocol	[IETF RFC 1350]

TUI	Table unique identifier
Tx	Transmit
TxLLC	Transmit logical link control
TxM	Transmit machine
UCCM	Unconnected Message Manager
UMAX	MAC ID of maximum unscheduled node
USR	Unscheduled start register
VLAN	Virtual LAN

3.6 Conventions for station management objects

Objects for station management are specified in Clause 7, using attributes, services and behaviour.

Object attributes are specified in tables formatted according to Table 1.

Table 1 – Format of attribute tables

Attri-bute ID	Need in implementation	Access rule	NV	Name	Data type	Description of attribute	Semantics of values
See a)	See b)	See c)	See d)	See e)	See 6.1.3.	See f)	See g)

a) “Attribute ID” is the identification value assigned to an attribute.

b) “Need in implementation” specifies whether or not the attribute is necessary in the implementation. An attribute may be optional, required or conditional.

A conditional attribute is required if certain object behaviours and/or attributes are implemented as defined by the class.

If an Attribute is optional, then a default value shall be defined in the semantics. An optional attribute's default value shall provide the same behaviour as if the attribute was not implemented. If the default value of an optional attribute does not match the behaviour of the object when the object does not implement the attribute, the object definition shall declare the behaviour.

In all cases, the term “default” indicates a “factory default” as shipped from the vendor.

c) “Access rule” specifies how a requestor can access an attribute. The definitions for access rules are:

- Settable (Set) – The attribute shall be accessed by at least one of the set services. For enumerated attributes, if the device supports only one of the values and the object definition does not require more than one value to be supported, then the attribute may be implemented as Get only.

Settable attributes, unless otherwise specified by the object definition, shall be also accessed by get services.

- Gettable (Get) – The attribute shall be accessed by at least one of the get services.

d) “NV” indicates whether an attribute value is maintained through power cycles. This column is used in object definitions where non-volatile storage of attribute values is required. An entry of “NV” indicates value shall be saved, “V” means not saved.

e) Name refers to the attribute.

f) Description of Attribute provides general information about the attribute.

g) Semantics of Values specifies the meaning of the value of the attribute.

4 Overview of the data-link protocol

4.1 General

4.1.1 DLL architecture

The Type 2 DLL is modeled as an integrated Access Control Machine (ACM), with scheduling support functions, designed for reliable and efficient support of higher-level connection-mode and connectionless data transfer services. Interoperating with these higher level functions are DLL management functions.

PhL framing and delimiters are managed by DLL functions for serializing and deserializing M_symbol requests and indications.

Within the Data Link Layer, the Access Control Machine (ACM) has the primary responsibility for detecting and recovering from network disruptions. The major objectives of the ACM are

- to make sure that the local node detects and fully utilizes its assigned access slots in the schedule;
- to make sure that the local node does not interfere with the transmissions of other nodes, especially the moderator node;
- to start the next NUT (see 4.1.2) on schedule, whether the moderator DLPDU is heard or not;
- if the local node is the moderator, to transmit each moderator DLPDU strictly on schedule.

The Data Link Layer is comprised of the components listed in Table 2:

Table 2 – Data-link layer components

Components	Description
Access Control Machine (ACM)	receives and transmits control DLPDUs and header information, and determines the timing and duration of transmissions.
Transmit LLC (TxLLC)	buffers DLSUs received from Station Management and the DLS-user, and determines which should be transmitted next.
Receive LLC (RxLLC)	performs the task of quarantining received link units until they are validated by a good FCS.
Transmit Machine (TxM)	receives requests to send DLPDU headers and trailers, and Lpackets from the ACM, and breaks them down into octet symbol requests that are sent to the Serializer.
Receive Machine (RxM)	assembles received Lpackets from octet symbols received from the Deserializer, and submits them to the RxLLC.
Serializer	receives octet symbols, encodes and serializes them, and sends them as M_symbols to the PhL. It is also responsible for generating the FCS.
Deserializer	receives M_symbols from the PhL, converts M_symbols into octets and sends them to the receive machine. It is also responsible for checking the FCS.
DLL Management Interface	holds the Station Management variables that belong to the DLL, and helps manage synchronized changes of the link parameters.

The internal arrangement of these components, and their interfaces, are shown in Figure 1. The arrowheads illustrate the primary direction of flow of data and control.

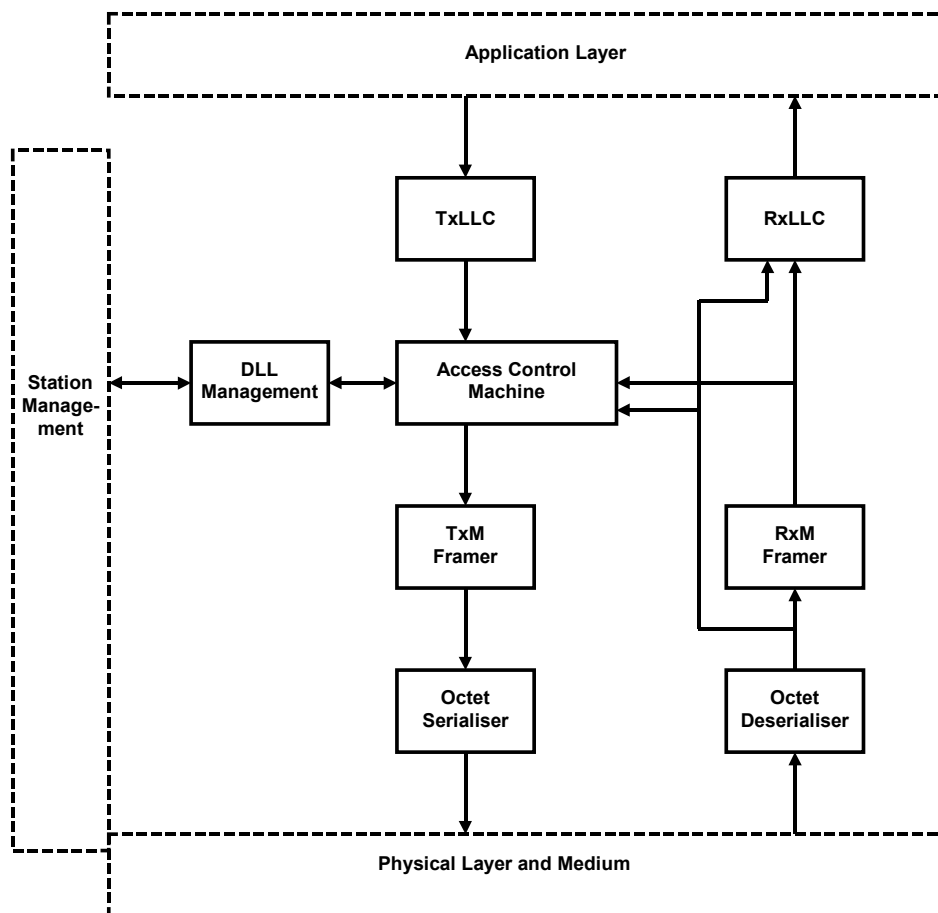


Figure 1 – Data-link layer internal architecture

4.1.2 Access control machine (ACM) and scheduling support functions

The ACM functions schedule all communication from one DLE to another. The timing of this communication is regulated at two levels,

- to provide an accurate time based distribution of scheduled opportunities for designated communications in accord with, and with a pre-established schedule for, the local link and any extended network of links,
- to provide a democratic distribution of unscheduled opportunities for arbitrary communications, generally in a cyclic but asynchronous manner.

Accurate schedule timing as provided by a) is very important to support many control and data collection tasks in the applications domain of this protocol. The schedule is based on a fixed, repetitive time cycle called network update time (NUT). The NUT is maintained in close synchronism among all nodes participating in the DLL and used to provide two types of access opportunity:

- One opportunity for each active station to transfer one scheduled DLPDU containing multiple scheduled DLSDUs.

Only DLSDUs with scheduled as their QoS parameter may be included in a scheduled DLPDU.

NOTE DLSDUs with scheduled as their QoS parameter are usually connection mode transfers with generic as their tag parameter.

- One opportunity for at least one active station to send one background (unscheduled) DLPDU containing multiple DLSDUs of any QoS type. If time is available within a NUT, one background MAC opportunity is offered to each other active station in the order of their MAC ID address. The station MAC ID for the first background opportunity is the

NUT is incremented (+1 modulo the set of background addresses) for each successive NUT to democratically share the available time among active stations.

Support procedures are included for automatic transfer to backup scheduling services and to manage the use of redundant Ph media.

4.1.3 Connection-mode, connectionless-mode data transfer and DL service

The connection mode, connectionless mode and DL services provide the necessary functionality for

- managing DL provider interactions with the DL user by converting request and response primitives into necessary DLE operations and generating indications and confirm primitives where appropriate,
- determination of the DL address and control information to be added to each DLSDU,
- local confirmation of status for each outgoing DLSDU,
- DLPDU formation by concatenation of multiple DL user DLSDUs for efficient transfer as single PhPDUs, subject to DLPDU size limitations, QoS parameters and the type of access opportunity.

4.2 Services provided by the DL

4.2.1 Overview

The DLL provides connectionless data transfer services and 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, a DL(SAP)-address and queue management service, and a packing/unpacking service that combines multiple DLSDUs into a single data transfer DLPDU for efficient use of each access opportunity.

4.2.2 QoS

4.2.2.1 Definition

The QoS parameter specifies priority options for DLSDUs and access opportunities. The available values are as follows:

4.2.2.2 DLL scheduled priority

The scheduled priority QoS provides accurate time based cyclic and acyclic sending of DLSDUs. The execution timing for this fieldbus scheduled service can be accurate and repeatable to better than 1 ms. Scheduled priority is normally used with connected mode services.

NOTE The master Keeper is responsible for regular publication of a unique table identifier (TUI) which ensures all nodes have a reference for the current set of link operating parameters. This TUI publication uses a fixed tag and is sent with QoS priority set to scheduled.

4.2.2.3 DLL high priority (unscheduled)

The high priority QoS provides acyclic sending of DLSDUs with a bounded upper time for the sending delay. Data on this priority is sent only when all scheduled data has been sent and a non-scheduled sending opportunity is available.

4.2.2.4 DLL low priority (unscheduled)

The low priority QoS provides for sending of DLSDUs only on an as-available basis. Data on this priority is sent only when all other data priorities have been sent and a non-scheduled sending opportunity is available.

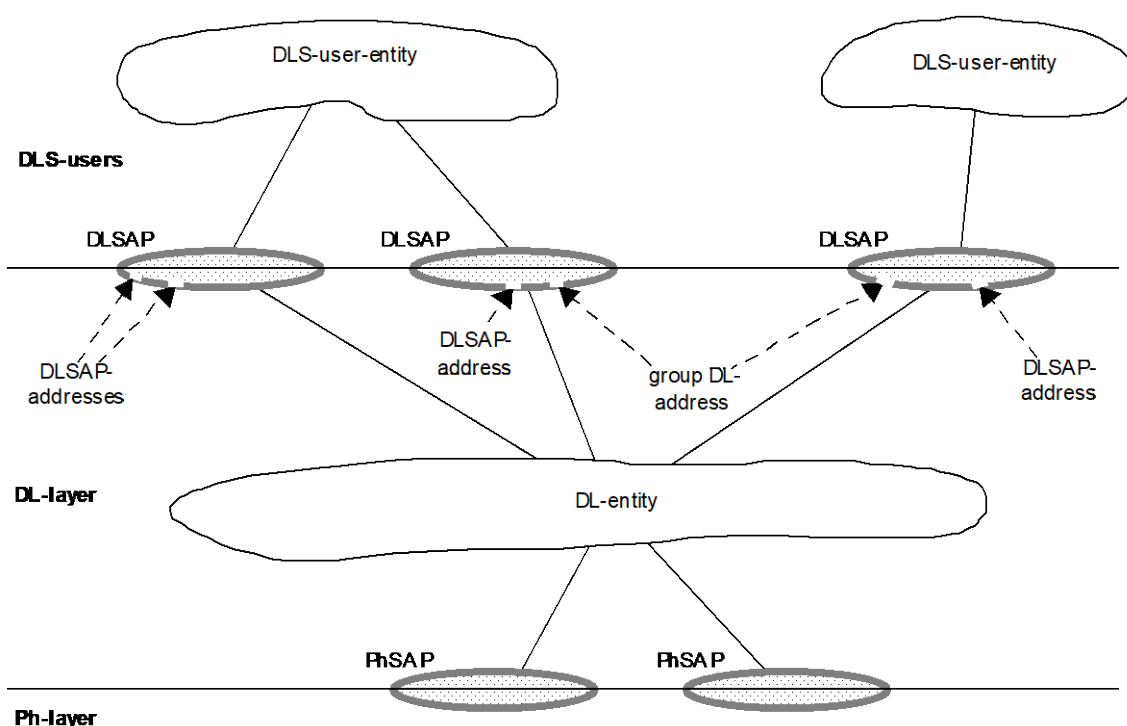
NOTE High and Low priority are used only in a local sense to set the order of servicing among locally submitted, unscheduled DLS-user-data.

4.3 Structure and definition of DL-addresses

4.3.1 General

DL-addresses are used as MAC ID addresses (Link node designators), Fixed tag addresses (DLSAP-addresses and group DL-addresses), and Generic tag addresses (DLCEP-addresses).

NOTE The definition of DLSAP, derived from ISO/IEC 7498-1, is explained here to facilitate understanding of the critical distinction between DLSAPs and their DL-addresses (see Figure 2).



NOTE 1 DLSAPs and physical layer service access points (PhSAPs) are depicted as ovals spanning the boundary between two adjacent layers.

NOTE 2 DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP.

Figure 2 – Relationships of DLSAPs, DLSAP-addresses, and group DL-addresses

A single DLE may have multiple DLSAP-addresses and group DL-addresses associated with a single DLSAP.

Subclause 4.3 defines the form and structure of DL-addresses and includes specific definitions for some standard DL-addresses.

All addresses are formed of octets. Each octet shall be transferred to the Ph layer low-bit first and multiple octets shall be transferred low-order octet first (little endian format).

Three types of DL addresses are used (see 4.3.2, 4.3.3 and 4.3.4).

4.3.2 MAC ID address

MAC ID addresses identify physical nodes on the local link as shown in Figure 3.

Permitted values are within the range 0 to 99, values from 100 to 254 are reserved.

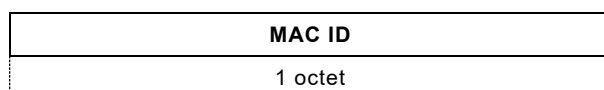
**Figure 3 – Basic structure of a MAC ID address**

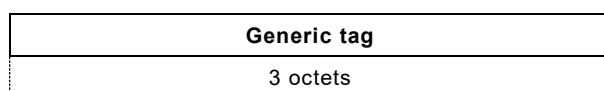
Table 3 specifies the use of MAC ID addresses.

Table 3 – MAC ID addresses allocation

MAC ID	Meaning and purpose
0x0	temporary address used for maintenance
SMAX	current maximum address value for Scheduled access opportunities. SMAX ≤ UMAX
UMAX	current maximum address value for Unscheduled access opportunities SMAX. ≤ UMAX ≤ 0x63
0x64 to 0xFE	reserved
0xFF	broadcast address to all MAC ID nodes on this link

4.3.3 Generic tag address

Generic tag addresses types identify all data transfers for connected mode services. The available capacity in the DLL is 3 octets, see Figure 4.

**Figure 4 – Basic structure of a generic tag address**

Generic tags are used to send connected DLSDUs and may be associated with any of the available QoS priorities (Scheduled, High, Low).

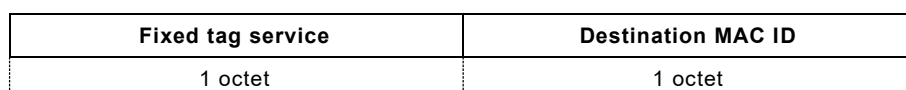
NOTE Negotiation and management of generic tags and connected mode services are the responsibility of the DL user.

4.3.4 Fixed tag address

Fixed tag addresses types identify DLSAPs within a designated station MAC ID. They are used with all data transfers for connectionless mode services.

NOTE Fixed tags are used to send connectionless DLSDUs and are usually associated with unscheduled QoS priorities (High, Low). Typical uses of fixed tags are for station management and to establish connections.

Each fixed tag address shall consist of two octets as shown in Figure 5.

**Figure 5 – Basic structure of a fixed tag address**

The first octet shall specify the service access point in the destination node as specified in Table 4. The second octet shall be the address of the destination node. The destination address shall be either a MAC ID or 0xFF, which indicates the broadcast address to all MAC IDs on the local link.

Table 4 – Fixed tag service definitions

Fixed tag service	Meaning
0x00	moderator
0x01 - 0x08	vendor specific
0x09	ping request
0x0A – 0x14	vendor specific
0x15	tMinus
0x16 - 0x28	vendor specific
0x29	ping reply
0x2A - 0x3F	vendor specific
0x40 - 0x6F	reserved
0x70 – 0x7F	vendor-specific
0x80	I'm alive
0x81	link parameters
0x82	reserved
0x83	UCMM
0x84	TUI
0x85	IP (reserved for Internet Protocol)
0x86	WAMI
0x87	reserved
0x88	Keeper UCMM
0x89	Ethernet (used for address resolution protocol)
0x8A - 0x8B	reserved
0x8C	Time distribution
0x8D – 0x8F	reserved for time distribution
0x90	debug
0x91 - 0xCF	reserved
0xD0 - 0xEF	group addresses
0xF0 - 0xFF	vendor specific

Fixed tags in the range 0xD0 to 0xEF shall be reserved to permit group screening of connection identifiers.

4.4 Services assumed from the PhL

4.4.1 General requirements

Subclauses 4.4.2 to 4.4.3 include requirements for interface to a Type 2 PhL that are not required for other Types:

- Transmit Machine;
- Receive Machine;
- Serializer;
- Deserializer.

This is necessary because the Ph-Interface Data Units (PhIDUs) that Type 2 passes across the DLL–PhL interface are individual serial data and non-data symbols, designated on the DLE side as M_Symbols (see Table 5), whereas PhIDUs for other Types are not conceptualized in this way. As a consequence, a Type 2 Data Link Layer requires a Type 2 PhL.

4.4.2 Data encoding rules

The M_Symbols transferred at the DLL to PhL interface are of four types designated 0, 1, +, -. They will be encoded inside the PhL into appropriate Ph_Symbols as shown in Table 5. The M_0 and M_1 will be encoded into Ph_Symbols that represent Manchester biphas L data encoding rules. The M_ND symbols are non-data symbols to allow for the creation of unique data patterns used for start and end delimiters. The example of signal voltage waveform as it might be applied to an electrically-conductive medium is shown in an idealized form in Figure 6 to provide an example of the data encoding rules shown in Table 5.

Table 5 – Data encoding rules

Data bits (common name)	M_Symbol representation	Ph_Symbol encoding	Manchester encoded
data “zero”	M_0 or {0}	{L,H}	0
data “one”	M_1 or {1}	{H,L}	1
“non_data+”	M_ND+ or {+}	{H,H}	no data
“non_data-”	M_ND- or {-}	{L,L}	no data

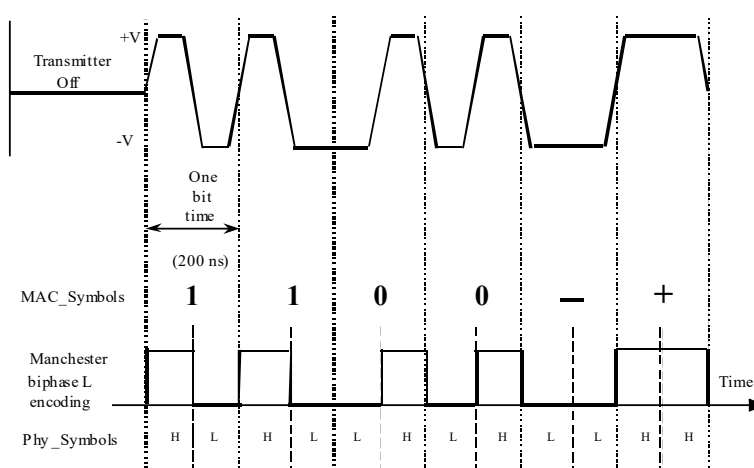


Figure 6 – M_symbols and Manchester encoding at 5 MHz

4.4.3 DLL to PhL interface

4.4.3.1 General

The DLL to PhL interface need not be exposed in the implementation of any PhL variant. This interface may be internal to the node. If conformance to the DLL to PhL interface is claimed, it shall conform to the requirements of 4.4.3.

4.4.3.2 Ph_lock_indication

ph_lock_indication shall provide an indication of either data lock or Ph_Symbol synchronization by the PhL. Valid states for ph_lock_indication shall be true and false. ph_lock_indication shall be true whenever valid Ph_Symbols are present at the PhL interface and the PhL to DLL interface timing of M_Symbols conforms to the PhL requirements. It shall be false between frames (when no Ph_Symbols are present on the medium) or whenever data synchronization is lost or the timing fails to conform to the PhL requirements. ph_lock_indication shall be true prior to the beginning of the start delimiter.

4.4.3.3 Ph_frame_indication

ph_frame_indication shall provide an indication of a valid data frame from the PhL. Valid states for ph_frame_indication shall be true and false. ph_frame_indication shall be true upon ph_lock_indication = true and reception of the first valid start delimiter. ph_frame_indication shall be false at reception of next M_ND symbol (following the start delimiter) or ph_lock_indication = false.

NOTE This signal provides octet synchronization to the DLL.

4.4.3.4 Ph_carrier_indication

ph_carrier_indication shall represent the presence of a signal carrier on the medium. The ph_carrier_indication shall be true if internal PhL identification of signal carrier has been true during any of the last 4 M_symbol times and it shall be false otherwise.

4.4.3.5 Ph_data_indication

ph_data_indication shall represent the M_Symbols that are decoded from the PhL internal representations such as those shown in Table 5. Valid M_symbols shall be M_0 {0}, M_1 {1}, M_ND+ {+}, and M_ND- {-} as shown in Table 6.

Table 6 – M Data symbols

Data bits (common name)	M_Symbol representation
data “zero”	M_0 or {0}
data “one”	M_1 or {1}
“non_data+”	M_ND+ or {+}
“non_data-”	M_ND- or {-}

The ph_data_indication shall represent the M_Symbols as decoded within the PhL whenever ph_lock_indication is true.

4.4.3.6 Ph_status_indication

ph_status_indication shall represent the delimiter status of the frame which was received from the PhL as shown in Table 7. Valid symbols shall be Normal, Abort, and Invalid. ph_status_indication shall indicate Normal after reception of a frame (ph_frame_indication = true) composed of a start delimiter, valid data (no M_ND symbols) and an end delimiter. ph_status_indication shall indicate Abort after reception of a frame (ph_frame_indication = true) composed of a start delimiter, valid data, and a second start delimiter. ph_status_indication shall indicate Invalid after reception of a frame (ph_frame_indication = true) composed of a start delimiter and the detection of any M_ND symbol which was not part of a start or end delimiter.

Table 7 – Truth table for ph_status_indication

Ph_status_indication	Ph_frame_indication	Start delimiters in a single frame	End delimiter detection	Any non-delimiter Manchester violations
Normal	true	1	true	false
Abort	true	2	don't care	false
Invalid	true	1	don't care	true

4.4.3.7 Ph_data_request

ph_data_request shall present the M_Symbols to be transmitted. Valid symbols shall be M_0, M_1, M_ND+ or M_ND– as shown in Table 6. ph_data_request shall indicate M_0 when no data is to be transmitted (and ph_frame_request = false).

4.4.3.8 Ph_frame_request

ph_frame_request shall be true when ph_data_request presents M_Symbols to be encoded into the appropriate Ph_Symbols by the PhL, and shall be false when no valid M_Symbols are to be transferred to the PhL.

4.5 Functional classes

Implementations of this protocol fall into two broad classes:

- connection originators (sometimes called Scanners);
- connection responders or connection targets (sometimes called Adaptors).

Devices with connection origination capability will include functions such as Scheduling object, support for external Configuration Tools and usually a Keeper object.

Connection responders are typically simpler devices which do not need the full set of capabilities.

This specification does not mandate particular groupings of functions for classes of implementation devices.

NOTE Grouping of services and functions into implementation classes can be done by separate specifications for profiles.

5 General structure and encoding of PhIDUs and DLPDUs and related elements of procedure

5.1 Overview

The DLL and its procedures are those necessary to provide the services offered to DLS user by using the services available from the PhL. The main services of this protocol are

- a) provision of timely access opportunities to support scheduled data-transfer DL-services on one link and on an extended multi-link network;
- b) the packing of multiple DLS-user transfers into a single PhL transfer unit (DLPDU) to efficiently use each transfer opportunity;
- c) the efficient distribution of DLS-user data from a publishing DLS-user to a set of subscribing DLS-users;
- d) the provision of a synchronized sense of internal time among the DLEs and available to the DLS-users of the extended link;
- e) the standardized availability of local DL-address, buffer and queue management capabilities.

5.2 Media access procedure

The primary task of the Data Link Layer is to determine, in co-operation with other Data Link Layers on the same link, the granting of permission to transmit on the medium. At its upper interface, the DLL provides services to receive and deliver service data units for the DLS user and Station Management.

The DLL protocol is based on a fixed, repetitive time cycle called a network update time (NUT). The NUT is maintained in close synchronism among all nodes on the link. A node is not permitted access to transmit on the medium if its NUT does not agree with the NUT currently being used on a link wide basis. Different links may have different NUT duration.

Each node contains its own timer synchronized to the NUT for the local link. Media access is determined by local sub-division of the NUT into access slots. Access to the medium is in sequential order based on the MAC ID of the node. Specific behaviors have been incorporated into the access protocol allowing a node which temporarily assumes a MAC ID of zero to perform link maintenance. The MAC ID numbers of all nodes on a link are unique. Any DLL detecting the presence of a duplicate MAC ID shall immediately stop transmitting.

An implicit token passing mechanism is used to grant access to the medium. Each node monitors the source MAC ID of each DLPDU received. At the end of a DLPDU, each DLL sets an “implicit token register” to the received source MAC ID + 1. If the implicit token register is equal to the local MAC ID, then the node shall transmit one DLPDU containing one or more Lpackets with data or a null DLPDU. In all other cases, the node watches for either a new DLPDU from the node identified by the “implicit token register” or a time-out value if the identified node fails to transmit. In each case, the “implicit token” is automatically advanced to the next MAC ID. All nodes have the same value in their “implicit token register” preventing collisions on the medium.

The time-out period (called the “slot time”) is based on the amount of time required for

- the current node to hear the end of the transmission from the previous node;
- the current node to begin transmitting;
- the next node to hear the beginning of the transmission from the current node.

The slot time is adjusted to compensate for the total length of the medium since the propagation delay of the medium effects the first and last item on the previous list.

NOTE The calculation of slot time is specified in 8.2.

Each NUT is divided into three major parts: scheduled, unscheduled, and guardband as shown in Figure 7. This sequence is repeated in every NUT. The implicit token passing mechanism is used to grant access to the medium during both the unscheduled and scheduled intervals.

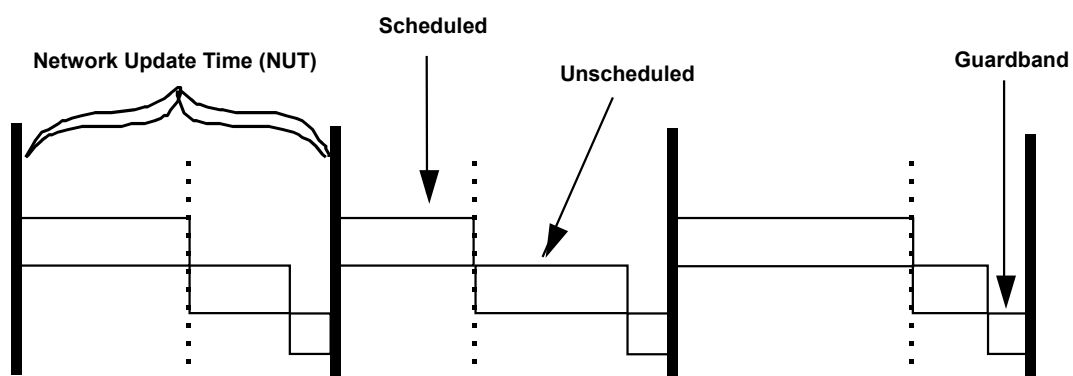


Figure 7 – NUT structure

During the scheduled part of the NUT, each node, starting with node 0 and ending with node SMAX, gets a chance to transmit time-critical (scheduled) data. SMAX is the MAC ID of the highest numbered node that has access to the medium during the scheduled part of the NUT. Every node between 0 and SMAX has only one opportunity to send one DLPDU of scheduled data in each NUT. The opportunity to access the medium during the scheduled time is the

same for each node in every NUT. This allows data that is transmitted during the scheduled portion of the NUT to be sent in a predictable and deterministic manner. Figure 8 shows how the permission to transmit is granted during the scheduled time. The DLS-user regulates the amount of data that each node may transmit during this scheduled token pass.

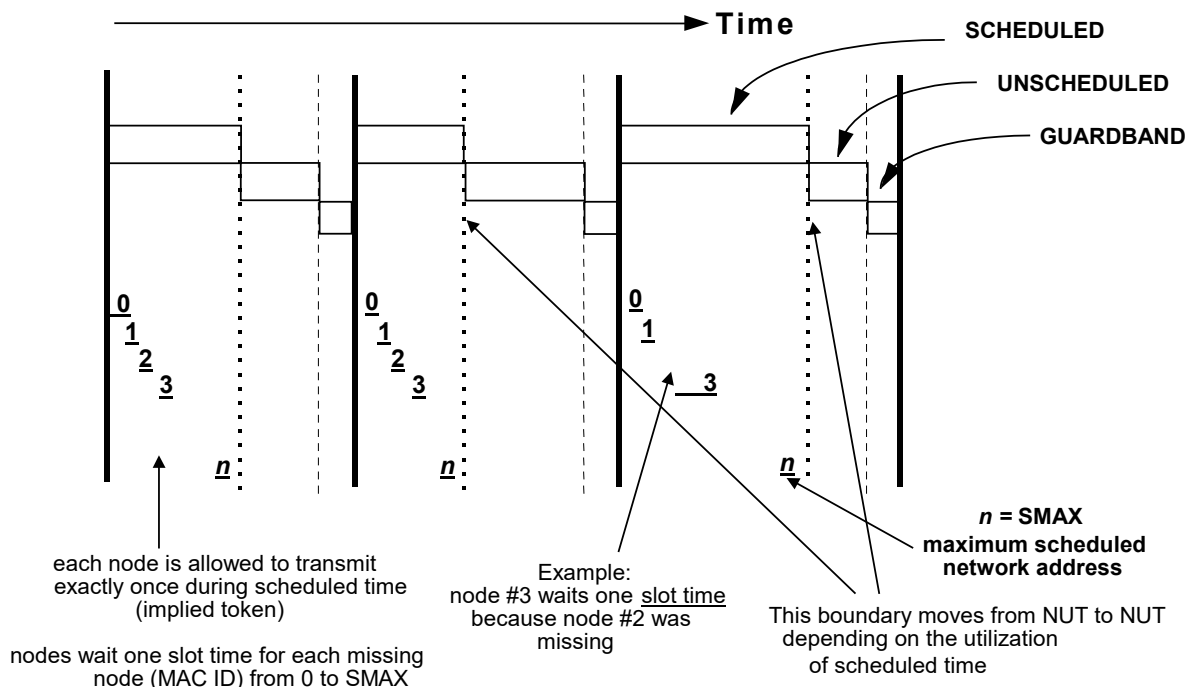


Figure 8 – Media access during scheduled time

During the unscheduled part of the NUT, each node from 0 to UMAX shares the opportunity to transmit one DLPDU of non-time critical data in a round robin fashion, until the allocated NUT duration is exhausted. UMAX is the MAC ID of the highest numbered node that has access to the medium during the unscheduled part of the NUT. The round robin method of access opportunity enables every node between 0 and UMAX to have zero, one or many opportunities to send unscheduled data depending on how much of the NUT remains after the completion of the scheduled traffic. Variations in scheduled traffic means the opportunity to access the medium during the unscheduled time may be different for each node in every NUT. Figure 9 shows how the permission to transmit is granted during the unscheduled time. The MAC ID of the node that goes first in the unscheduled part of the NUT is incremented by 1 for each NUT. The unscheduled token begins at the MAC ID specified in the unscheduled start register (USR) of the previous moderator DLPDU. The USR increments by one modulo (UMAX+1) each NUT. If the USR reaches UMAX before the guardband, it returns to zero and the token pass continues.

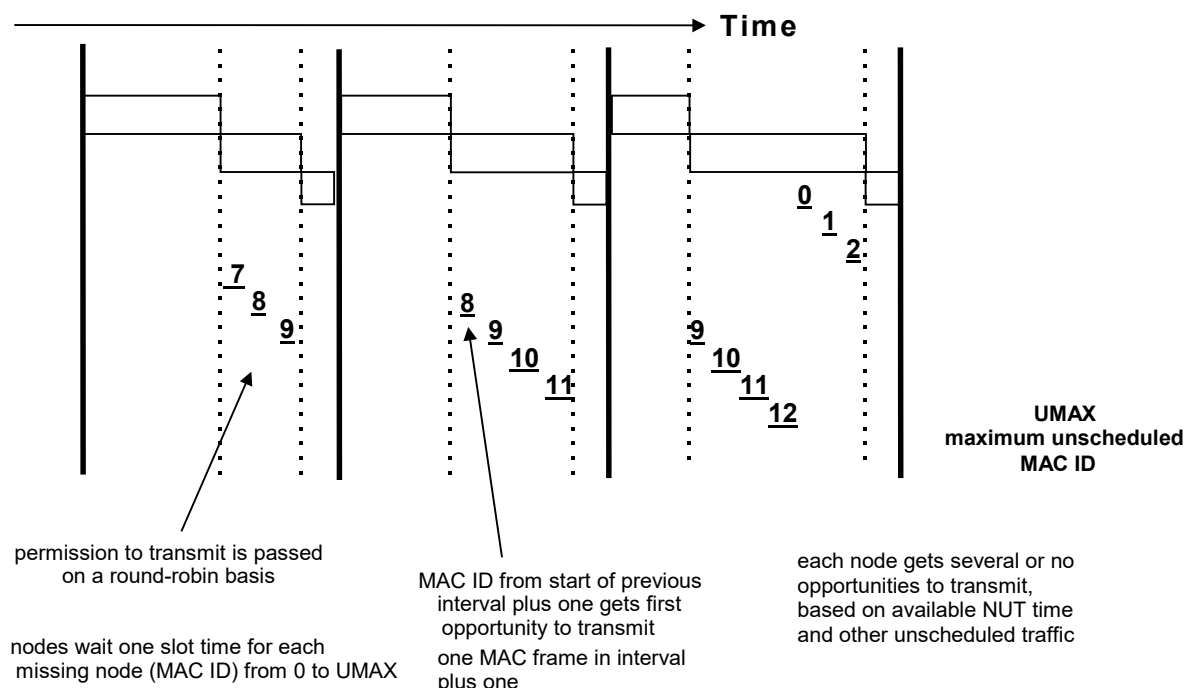


Figure 9 – Media access during unscheduled time

When the guardband is reached, all nodes stop transmitting. A node is not allowed to start a transmission unless it can be completed before the beginning of the guardband. During the guardband, the node with the lowest MAC ID (called the “moderator”) transmits a maintenance message (called the “moderator DLPDU”) that accomplishes two things:

- it keeps the NUT timers of all nodes synchronized;
- it publishes critical link parameters enabling all members of the local DLL group to share a common version of important DLL values such as NUT, slot time, SMAX, UMAX, etc.

The moderator transmits the moderator DLPDU, which re-synchronizes all nodes and restarts the NUT. Following the receipt of a valid moderator DLPDU, each node compares its internal values with those transmitted in the moderator DLPDU. A node using link parameters that disagree with the moderator disables itself. If the moderator DLPDU is not heard for 2 consecutive NUTs, the node with the lowest MAC ID assumes the moderator role, and begins transmitting the moderator DLPDU in the guardband of the third NUT. A moderator node that notices another node on-line and transmitting with a MAC ID lower than its own immediately cancels its moderator role.

Typical situations that may cause disruption of the DLL access protocol include

- induced noise on the link;
- poor cable or termination quality;
- physically connecting two links together while the network is operating.

One common consequence of such disruption is that nodes may be caused to disagree as to which node should be transmitting; this is called a network “non-concurrence”. Another potential problem occurs when the nodes do not agree to the same link configuration parameters. A node that disagrees with the link parameters as transmitted by the moderator is called a “rogue” and immediately stops transmitting. The DLL access protocol is designed to recover a rogue node and bring it back on-line.

5.3 DLPDU structure and encoding

5.3.1 General

DLPDUs are the DLPDUs passed to the Ph layer for sending on the Ph medium. The DLPDUs are formed by the transmit logical link control (TxLLC) protocol in two stages:

- DLSDUs are formed into Link-units (Lpackets);
- Multiple Lpackets are packed into one DLPDU subject to constraints set by the QoS required, the type of access opportunity available and limits on DLPDU overall length.

5.3.2 DLPDU components

The general DLPDU structure is shown in Figure 10 and managed by the DLL support units TxFramer, RxFramer, Octet Serializer and Octet Deserializer as shown in Figure 1.

The components of the DLPDU shall be transmitted in the following order: preamble, start delimiter, source MAC ID, zero or more Lpackets, FCS and end delimiter.

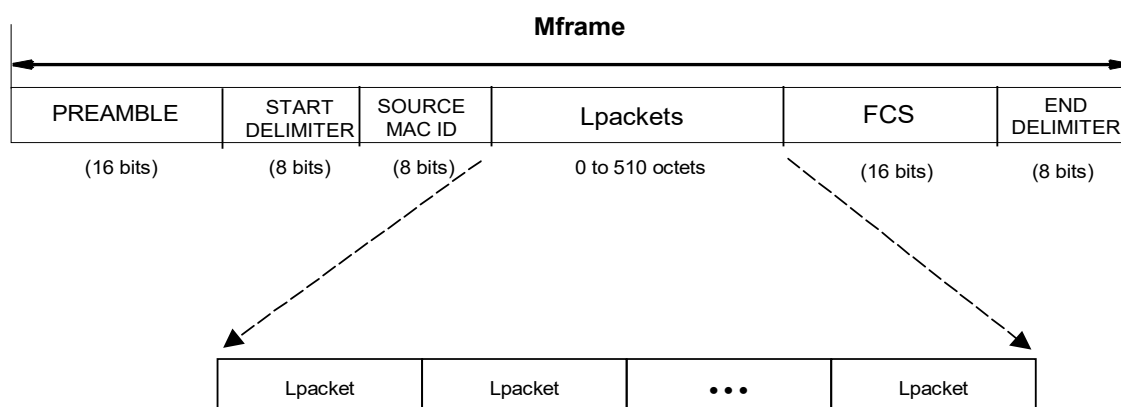


Figure 10 – DLPDU format

5.3.3 Preamble

The preamble shall be composed of 16 consecutive {1} M_Symbols.

5.3.4 Start and end delimiters

The start delimiter shall consist of these M_Symbols, {+, 0, –, +, –, 1, 0, 1}, transmitted from left to right.

The end delimiter shall consist of these M_Symbols, {1, 0, 0, 1, +, –, +, –}, transmitted from left to right.

The use of non-data M_Symbols shall be reserved for the start and end delimiters.

5.3.5 DLPDU octets and ordering

The source MAC ID, Lpackets, and FCS shall be composed of octets. An octet shall be composed of exactly eight M_Symbols each of which shall be either {0} or {1}. An octet shall be transferred to the PhL low bit first. Multiple octet fields shall be transmitted low order octet first (little endian format).

5.3.6 Source MAC ID

The source MAC ID address shall be a single octet as specified in 4.3.2. A node may also temporarily assume a MAC ID of zero to perform link maintenance.

5.3.7 Lpackets field

Zero or more Lpackets shall be concatenated into a single DLPDU subject to the constraint that the maximum number of octets which compose all Lpackets in a DLPDU shall be 510.

An additional constraint is that DLPDUs sent during a scheduled access opportunity shall only contain Lpackets with a QoS parameter of scheduled.

5.3.8 Frame check sequence (FCS)

5.3.8.1 Frame check sequence requirements

The FCS shall be a modified CCITT checksum using the 16 bit polynomial: $G(X) = X^{16} + X^{12} + X^5 + 1$. A two octet FCS shall be included in each DLPDU.

The formal concepts for its generation and checking on reception are described in 5.3.8.2. The generation and checking of the FCS are further specified in 9.7 and 9.8.

5.3.8.2 Frame check sequence formal concepts

5.3.8.2.1 Frame check sequence field

Within 5.3.8.2, any reference to bit K of an octet is a reference to the bit whose weight in a one-octet unsigned integer is 2^K .

NOTE This is sometimes referred to as "little endian" bit numbering.

For this document, as in other international standards (for example, ISO/IEC 3309, ISO/IEC 8802 and ISO/IEC 9314-2), DLPDU-level error detection is provided by calculating and appending a multi-bit frame check sequence (FCS) to the other DLPDU fields during transmission to form a "systematic code word" (see for example [20]⁵) of length n consisting of k DLPDU message bits followed by $n - k$ (equal to 16) redundant bits, and by calculating during reception that the message and concatenated FCS form a legal (n, k) code word. The mechanism for this checking is as follows, where the generic form of the generator polynomial for this FCS construction is specified in equation (4) and the polynomial for the receiver's expected residue is specified in equation (9). The specific polynomials for this document are specified in Table 8.

⁵ Numbers in square brackets refer to the Bibliography.