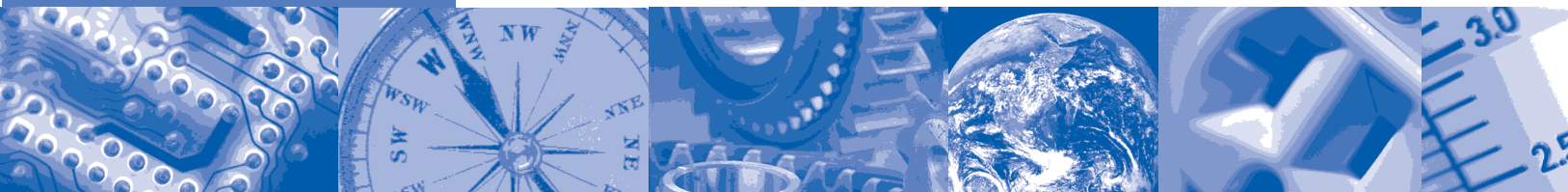


ANSI/ISA-50.02, Part 3-1997

Formerly ANSI/ISA-S50.02, Part 3-1997



Fieldbus Standard for Use in Industrial Control Systems Part 3: Data Link Service Definition

Second Printing: January 1999



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

Approved 6 January 1998



Documents
Document Sharing Hub

ANSI/ISA-50.02, Part 3-1997, Fieldbus Standard for Use in Industrial Control Systems,
Part 3: Data Link Service Definition.

ISBN 1-55617-641-4

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

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

Preface

This preface is included for informational purposes and is not part of ANSI/ISA-50.02, Part 3-1997.

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

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

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

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

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

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

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

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

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

The following people served as members of ISA Committee SP50:

NAME	COMPANY
R. Caro, Chairman	Consultant
H. Dammeyer, Managing Director	Ohio State University
G. Wood, Managing Director	Graeme Wood Consulting
C. Akiyama	Yokogawa Electric Corporation
J. Aliperti	Chromatech Consultoria Limited
W. Baker	KML, Inc.
L. Beckman	HIMA Americas, Inc.
M. Borges	Petrobras Cenpes Supen Diprom
R. Bowden	Fisher-Rosemount Limited
S. Boyer	Iliad Engineering, Inc.
*P. Brett	Honeywell, Inc.
P. Burton	Peter Burton & Associates
R. Canfield	Square D Company
A. Capel	Comgate Engineering, Ltd.
B. Casner	Miller Electric Manufacturing Company
R. Crowder	Ship Star Associates, Inc.
M. Dejani	Phillip Morris USA
E. Delahostria	Rockwell Automation
U. Doebrich	Siemens AG
J. Dohoney	Jet Propulsion Lab

* One vote per company

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

Schneider Electric
 Honeywell, Inc.
 R. Stahl, Inc.
 Elsag Bailey
 Krohne America Inc.
 Honeywell, Inc.
 General Motors Corporation
 Dow Chemical Company
 I/OLink, Inc.
 Ronan Engineering Company
 Kenonic Controls Limited
 Fuji Electric Company, Ltd.
 Union Carbide Corporation
 Valmet Automation Inc.
 Cegelec
 Fieldbus International A/s FINT
 Cullen G. Langford Inc.
 Bailey Controls Company
 AMP, Inc.
 Marconi Italiana Automazione
 Endress + Hauser GmbH + Company
 Exxon Company USA
 Chagrin Valley Controls, Inc.
 FMCK Associates, Ltd.
 Process Control Solutions, Inc.
 MerTech, Inc.
 Valtek International
 Universita Degli Studi di Catania
 MPI
 SAI
 IsoMatic Laboratories
 Merck & Company Inc.
 Fieldbus Foundation
 Cornell University
 Cegelec
 Fisher•Rosemount Systems Inc.
 V Automation
 Honeywell Inc.
 ENEL-DSR
 Schwarz Communication Consult
 Dow Chemical Company
 Toshiba Corporation
 Tendent Consulting
 Czech Standards Institute
 Lockheed Martin Federal Services
 Saudi Aramco Oil Company
 Measurement Technology Limited
 Shell Oil Company
 Mobil Technology Company
 Ronan Engineering Company
 Cegelec
 Sun Microsystems, Inc.
 James River Corporation
 Electric Power Research Institute
 Eastman Kodak Company
 National Electrical Manufacturers Assoc.
 Bailey Controls Company
 Fisher•Rosemount Systems Inc.

* One vote per company

This published standard was approved for publication by the ISA Standards and Practices Board on January 6, 1998.

NAME

R. Webb, Vice President
H. Baumann
D. Bishop
P. Brett
W. Calder III
M. Cohen
H. Dammeyer
R. Dieck
W. Holland
H. Hopkins
A. Iverson
K. Lindner
V. Maggioli
T. McAvinew
A. McCauley, Jr.
G. McFarland
E. Montgomery
D. Rapley
R. Reimer
J. Rennie
W. Weidman
J. Weiss
J. Whetstone
M. Widmeyer
H. Wiegler
C. Williams
G. Wood
M. Zielinski

COMPANY

Pacific Gas & Electric Company
H. D. Baumann Associates Limited
Chevron USA Production Company
Honeywell Inc.
Factory Mutual Research Corporation
Flexonics, Inc.
Ohio State University
Pratt & Whitney
Southern Company Services, Inc.
Utility Products of Arizona
Ivy Optiks
Endress + Hauser GmbH & Company
Feltronics, Inc.
Instrumentation & Control Engineering Services
Chagrin Valley Controls, Inc.
Honeywell Inc.
Fluor Daniel, Inc.
Rapley Engineering Services
Rockwell Automation A-B
Factory Mutual Research Corporation
Consultant
Electric Power Research Institute
National Institute of Standards & Technology
Carnegie-Mellon University
Canus Corporation
Eastman Kodak Company
Graeme Wood Consulting
Fisher•Rosemount Systems Inc.



Documents

Document Sharing Hub

ANSI/ISA-S50.02, Part 3-1997

Contents

Introduction	19
Keywords.....	19
General.....	19
Purpose	19
1 Scope	20
1.1 Specifications	20
1.2 Conformance	20
2 Normative references	20
3 Definitions.....	21
3.1 Reference model definitions	21
3.2 Service convention definitions	23
3.3 Data Link Service definitions	24
4 Symbols and abbreviations	27
5 Conventions	27
5.1 General conventions.....	27
5.2 Parameters	28
5.3 Identifiers	29
6 Overview of the Data Link Service	29
6.1 Overview of DL-subnetwork structuring.....	30
6.2 Overview of DL-naming (addressing)	31
7 Types and classes of Data Link Service	33
8 Quality of Service (QoS) attributes common to multiple types of Data Link Service.....	33
8.1 DLL priority (dynamic QoS attribute)	34
8.2 DLL maximum confirm delay (dynamic QoS attribute)	34
8.3 DLPDU authentication (semi-static QoS attribute)	35
8.4 DL-scheduling-policy (semi-static QoS attribute)	36
8.5 DL-timeliness (dynamic DLCEP QoS attributes)	36
9 Facilities of the DL(SAP)-address, queue and buffer management Data Link Service.....	39

10 Model of the DL(SAP)-address, queue and buffer management	
Data Link Service	39
11 Sequence of primitives at one DLSAP	40
11.1 Constraints on sequence of primitives	40
12 DL(SAP)-address, queue and buffer management facilities.....	41
12.1 Create	42
12.2 Delete	45
12.3 Bind	46
12.4 Unbind	51
12.5 Put	52
12.6 Get.....	53
13 Facilities of the connection-mode Data Link Service.....	57
14 Model of the connection-mode Data Link Service.....	58
14.1 DLCEP-identification	59
14.2 Model of a peer DLC	59
14.3 Model of a multi-peer DLC.....	62
15 Quality of connection-mode service	66
15.1 Determination of QoS for connection-mode service	66
15.2 Definition of QoS parameters	66
16 Sequence of primitives.....	73
16.1 Concepts used to define the connection-mode DL-service	73
16.2 Constraints on sequence of primitives.....	73
17 Connection establishment phase.....	83
17.1 Function.....	83
17.2 Types of primitives and parameters.....	83
17.3 Sequence of primitives	88
18 Connection release phase	90
18.1 Function.....	90
18.2 Types of primitives and parameters.....	91
18.3 Sequence of primitives when releasing an established DLC/DLCEP.....	93
18.4 Sequence of primitives in a DLS-user rejection of a DLC / DLCEP establishment attempt	95
19 Data transfer phase	97
19.1 Queue data transfer.....	97
19.2 Buffer data transfer	100

19.3 Reset	102
19.4 Subscriber query	108
20 Facilities of the connectionless-mode Data Link Service.....	110
21 Model of the connectionless-mode Data Link Service.....	111
21.1 Model of DL-connectionless-mode unitdata transmission	111
21.2 Model of DL-connectionless-mode unitdata exchange.....	111
22 Quality of connectionless-mode service	112
22.1 Determination of QoS for connectionless-mode service.....	113
22.2 Definition of QoS parameters	113
23 Sequence of primitives.....	113
23.1 Constraints on sequence of primitives.....	113
23.2 Relation of primitives at the end-points of connectionless service	114
23.3 Sequence of primitives at one DLSAP	115
24 Connectionless-mode functions	116
24.1 Data transfer	116
24.2 Data exchange	120
24.3 Listener query	127
25 Facilities and classes of the time and scheduling guidance Data Link Service.....	128
26 Model of the time and scheduling guidance Data Link Service.....	130
27 Quality of scheduling guidance service	130
28 Sequence of primitives at one DLE.....	130
28.1 Constraints on sequence of primitives.....	130
29 Scheduling guidance functions.....	132
29.1 DL-time	132
29.2 Compel service	134
29.3 Schedule sequence	136
29.4 Cancel schedule	141
29.5 Subset sequence	142
Annex A — (normative) DL-Management service definition.....	145
A.1 Scope and inheritance	145
A.2 Facilities of the DL-management service	145
A.3 Model of the DL-management service	145

A.4 Constraints on sequence of primitives	145
A.5 Set.....	146
A.6 Get	147
A.7 Action	148
A.8 Event	149

Figures

Figure 1 — Relationship of this part of ISA-S50.02 to other Fieldbus and OSI standards.....	19
Figure 2 — Relationships of DLSAPs, DLCEPs, DLSAP-addresses, A DLCEP-addresses, and group DL-addresses	25
Figure 3 — Example of paths, links, bridges, and the extended link.....	31
Figure 4 — Types of DL-timeliness	37
Figure 5 — Sequence of primitives for the DL(SAP)-address, queue and buffer management DL-services	41
Figure 6 — Supported methods of data management for transmission and delivery.....	42
Figure 7 — Peer-to-peer and multi-peer DLCs and their DLCEPs.....	57
Figure 8 — OSI abstract queue model of a peer DLC between a pair of DLS-users.....	59
Figure 9 — OSI abstract queue model of a multi-peer DLC between a publishing DLS-user and a set of subscribing DLS-users	63
Figure 10 — Summary of DL-connection-mode service primitive time-sequence diagrams for Peer DLCs (part 1)	76
Figure 11 — Summary of DL-connection-mode service primitive time-sequence diagrams for Peer DLCs (part 2)	77
Figure 12 — Summary of DL-connection-mode service primitive time-sequence diagrams for publishers of a multi-peer DLC (part 1).....	78
Figure 13 — Summary of DL-connection-mode service primitive time-sequence diagrams for publishers of a multi-peer DLC (part 2).....	79
Figure 14 — Summary of additional DL-connection-mode service primitive time-sequence diagrams for a multi-peer DLC subscriber where the diagrams differ from the corresponding ones for a publisher (part 1) ...	80
Figure 15 — Summary of additional DL-connection-mode service primitive time-sequence diagrams for a multi-peer DLC subscriber where the diagrams differ from the corresponding ones for a publisher (part 2) ...	81
Figure 16 — State transition diagram for sequences of DL-connection-mode service primitives at a DLCEP	82
Figure 17 — Peer DLC/DLCEP establishment initiated by a single DLS-user ..	88
Figure 18 — Multi-peer DLC/DLCEP establishment initiated by the Publishing DLS user	89

Figure 19 — Multi-peer DLC/DLCEP establishment initiated by a Subscribing DLS- user	89
Figure 20 — Multi-peer DLC/DLCEP establishment using known DLCEP addresses initiated first by the Publishing DLS-user	89
Figure 21 — Multi-peer DLC/DLCEP establishment using known DLCEP addresses initiated first by one or more Subscribing DLS-users	89
Figure 22 — Peer DLC/DLCEP establishment initiated simultaneously by both Peer DLS-users, resulting in a merged DLC	90
Figure 23 — Multi-peer DLC/DLCEP establishment initiated simultaneously by both Publishing and Subscribing DLS-users, resulting in a merged DLC	90
Figure 24 — Peer DLS-user invocation.....	93
Figure 25 — Publishing DLS-user invocation	93
Figure 26 — Subscribing DLS-user invocation.....	94
Figure 27 — Simultaneous invocation by both DLS-users.....	94
Figure 28 — Peer DLS-provider invocation	94
Figure 29 — Publishing DLS-provider invocation	94
Figure 30 — Subscribing DLS-provider invocation.....	94
Figure 31 — Simultaneous Peer DLS-user and DLS-provider invocations.....	94
Figure 32 — Simultaneous Publishing DLS-user and DLS-provider invocations.....	94
Figure 33 — Simultaneous Subscribing DLS-user and DLS-provider invocations.....	94
Figure 34 — Sequence of primitives in a Peer DLS-user rejection of a DLC/DLCEP establishment attempt	95
Figure 35 — Sequence of primitives in a Publishing DLS-user rejection of a DLC/DLCEP establishment attempt	95
Figure 36 — Sequence of primitives in a Subscribing DLS-user rejection of a DLC/DLCEP establishment attempt	95
Figure 37 — Sequence of primitives in a DLS-provider rejection of a DLC/DLCEP establishment attempt	95
Figure 38 — Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: both primitives are destroyed in the queue	96

Figure 39 — Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: DL-Disconnect indication arrives before DL-Connect response is sent	96
Figure 40 — Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: Peer DL-Disconnect indication arrives after DL-Connect response is sent	96
Figure 41 — Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: Publisher's DL-Disconnect indication arrives after DL-Connect response is sent	96
Figure 42 — Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: Subscriber's DL-Disconnect request arrives after DL-Connect request has been communicated to the Publisher	97
Figure 43 — Sequence of primitives for a Classical or Disordered peer-to-peer queue-to-queue data transfer	100
Figure 44 — Sequence of primitives for an Ordered or Unordered peer-to-peer, or an Unordered subscriber-to-publisher queue-to-queue data transfer	100
Figure 45 — Sequence of primitives for a publisher-to-subscribers queue-to-queue data transfer	100
Figure 46 — Sequence of primitives for a failed queue-to-queue data transfer	100
Figure 47 — Sequence of primitives for an Ordered or Unordered peer-to-peer, or an Unordered subscriber-to-publisher, buffer to buffer data transfer	102
Figure 48 — Sequence of primitives for a publisher-to-subscribers buffer to buffer data transfer	102
Figure 49 — Sequence of primitives for an Ordered or Unordered peer-to-peer, or an unordered subscriber to publisher, buffer to queue data transfer	102
Figure 50 — Sequence of primitives for a publisher-to-subscribers buffer-to-queue data transfer	102
Figure 51 — Sequence of primitives in a Peer DLS-user initiated Reset	106
Figure 52 — Sequence of primitives in a Publishing DLS-user initiated Reset	107
Figure 53 — Sequence of primitives in a Subscribing DLS-user initiated Reset	107

Figure 54 — Sequence of primitives in a simultaneous Peer DLS-users initiated Reset	107
Figure 55 — Sequence of primitives in a simultaneous Multi-peer DLS-users initiated Reset	107
Figure 56 — Sequence of primitives in a Peer DLS-provider initiated Reset.....	107
Figure 57 — Sequence of primitives in a Publishing DLS-provider initiated Reset	107
Figure 58 — Sequence of primitives in a Subscribing DLS-provider initiated Reset.....	108
Figure 59 — Sequence of primitives in a simultaneous Peer DLS-user and DLS-provider initiated Reset.....	108
Figure 60 — Sequence of primitives in a simultaneous Publishing DLS-user and DLS-provider initiated Reset.....	108
Figure 61 — Sequence of primitives in a simultaneous Subscribing DLS-user and DLS-provider initiated Reset.....	108
Figure 62 — Sequence of primitives for Subscriber Query.....	110
Figure 63 — Model for a data-link connectionless-mode unitdata transmission or unitdata exchange	111
Figure 64 — Summary of DL-connectionless-mode service primitive time-sequence diagrams	115
Figure 65 — State transition diagram for sequences of connectionless-mode primitives at one DLSAP	116
Figure 66 — Sequence of primitives for a successful locally-acknowledged connectionless-mode unitdata transfer	120
Figure 67 — Sequence of primitives for a successful remotely-acknowledged connectionless-mode unitdata transfer	120
Figure 68 — Sequence of primitives for an unsuccessful connectionless-mode unitdata transfer.....	120
Figure 69 — Sequence of primitives for connectionless-mode unitdata exchange	127
Figure 70 — Sequence of primitives for connectionless-mode listener query.....	128
Figure 71 — Summary of time and scheduling-guidance service primitive time sequence diagrams.....	131
Figure 72 — Sequence of primitives for DL-time.....	133

Figure 73 — Sequence of primitives for the Compel Service service	136
Figure 74 — Sequence of primitives for the sequence scheduling services	141
Figure A.1 — Sequence of primitives for the DLM action service	146

Copyright © 2009 by The Instrumentation, Systems, and Automation Society

Tables

Table 1 — Summary of DL(SAP)-address, queue and buffer management primitives and parameters	40
Table 2 — DL-buffer-and-queue-management Create primitive and parameters.....	42
Table 3 — DL-buffer-and-queue-management Delete primitive and parameters.....	45
Table 4 — DL(SAP)-address-management Bind primitive and parameters	47
Table 5 — DL(SAP)-role Constraints on DLSAPs, DLCEPs, and other DLS Primitives	48
Table 6 — DL(SAP)-address-management Unbind primitive and parameters.....	52
Table 7 — DL-buffer-management Put primitive and parameters	52
Table 8 — DL-buffer-and-queue-management Get primitive and parameters.....	54
Table 9 — Relationships between abstract queue model objects	61
Table 10 — Attributes and class requirements of DLCEP data delivery features	68
Table 11 — Summary of DL-connection-mode primitives and parameters (part 1).....	74
Table 12 — Summary of DL-connection-mode primitives and parameters (part 2).....	75
Table 13 — DLC / DLCEP establishment primitives and parameters (part 1).....	84
Table 14 — DLC / DLCEP establishment primitives and parameters (part 2).....	85
Table 15 — DLC / DLCEP release primitives and parameters	91
Table 16 — Queue data transfer primitive and parameters	97
Table 17 — Buffer sent primitive and parameter	101
Table 18 — Buffer received primitive and parameter	101
Table 19 — DLC/DLCEP reset primitives and parameters (part 1)	103
Table 20 — DLC/DLCEP reset primitives and parameters (part 2)	104
Table 21 — Subscriber query primitives and parameters.....	109

Table 22 — Summary of DL-connectionless-mode primitives and parameters.....	114
Table 23 — DL-connectionless-mode unitdata transfer primitives and parameters.....	117
Table 24 — DL-connectionless-mode unitdata exchange primitive and parameters.....	122
Table 25 — Listener query primitives and parameters.....	127
Table 26 — Summary of DL-scheduling-guidance primitives and parameters.....	131
Table 27 — DL-time primitive and parameters	132
Table 28 — DL-scheduling-guidance Compel Service primitive and parameters.....	134
Table 29 — DL-scheduling-guidance Schedule Sequence primitives and parameters.....	137
Table 30 — DL-scheduling-guidance Cancel Schedule primitives and parameters.....	141
Table 31 — DL-scheduling-guidance Subset Sequence primitives and parameters.....	142
Table A.1 — Summary of DL-management primitives and parameters.....	146
Table A.2 — DLM-Set primitive and parameters	146
Table A.3 — DLM-Get primitive and parameters.....	147
Table A.4 — DLM-Action primitive and parameters.....	148
Table A.5 — DLM-Event primitive and parameters	149

Introduction

Keywords

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

General

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

Purpose

The purpose of this part of ISA-S50.02 is to define the service provided

- a) to the Fieldbus Application Layer at the boundary between the Application and Data Link Layers of the Fieldbus Reference Model, and
- b) to the OSI Network Layer at the boundary between the Network and Data Link Layers of the OSI Basic Reference Model.

The Data Link Service is provided by the Data Link Protocol making use of the services available from the Physical Layer. This part of ISA-S50.02 also defines the Data Link Service characteristics that the immediately higher-level protocol may exploit. The relationship between the Standards for Fieldbus Data Link Service, Fieldbus Data Link Protocol, Fieldbus Physical Service, and OSI Network or Fieldbus Application Protocol is illustrated in Figure 1.

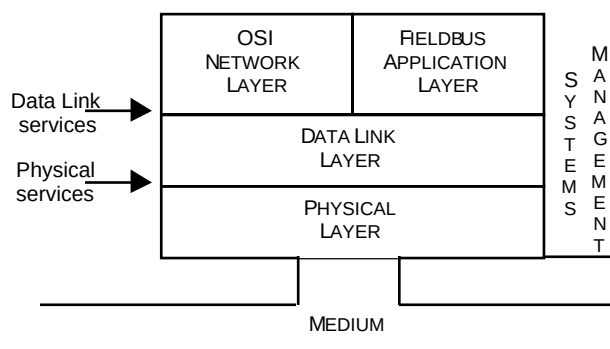


Figure 1 — Relationship of this part of ISA-S50.02 to other Fieldbus and OSI standards

Throughout the set of Fieldbus standards, the term “service” refers to the abstract capability provided by one layer of the OSI or Fieldbus Basic Reference Model to the layer immediately above. Thus, the Data Link Service defined in this document is a conceptual architectural service, independent of administrative divisions.

1 Scope

The Fieldbus Data Link service definition is a Data Link layer Standard designed to support time-critical messaging communications between devices in an automation environment. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant, and possibly human life.

This part of ISA-S50.02 defines in an abstract way the externally visible service provided by the Fieldbus Data Link Layer in terms of:

- a) the primitive actions and events of the service;
- b) the parameters associated with each primitive action and event, and the form that they take; and
- c) the interrelationship between, and the valid sequences of, these actions and events.

The services defined in this part of ISA-S50.02 are a superset of those provided by OSI Data Link Protocols as specified in ISO 8886. The services defined in this part of ISA-S50.02 may be used by any OSI Network Protocol or Fieldbus Application Protocol.

1.1 Specifications

The principal objective of this part of ISA-S50.02 is to specify the characteristics of a conceptual Data Link Service suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of Data Link protocols for time-critical communications.

1.2 Conformance

This part of ISA-S50.02 does not specify individual implementations or products, nor does it constrain the implementations of Data Link entities within industrial automation systems. However, its annexes may recommend exposed programming language interfaces within such systems.

There is no conformance of equipment to this Data Link Service Definition standard. Instead, conformance is achieved through implementation of conforming Data Link protocols that fulfill the Data Link Service defined in this part of ISA-S50.02. Conformance may also be achieved through implementation of exposed programming language interfaces as specified in annexes to this part of ISA-S50.02.

2 Normative references

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

normative documents indicated below. Members of IEC and ISO maintain registers of currently valid Standards.

ISO 7498-1:1984, *Information processing systems — Open systems interconnection — Basic reference model.*

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

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

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

ISO/IEC 10039:1990, *Information technology — Telecommunications and information exchange between systems — Medium access control service definition.*

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

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

ANSI/ISA-S50.02-1997 — *Fieldbus Standard for use in industrial control systems — Part 4: Data link protocol specification.*

3 Definitions

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

3.1 Reference model definitions

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

3.1.1	DL-address	[7498-3]
3.1.2	DL-address-mapping	[7498-1]
3.1.3	called-DL-address	[7498-3]
3.1.4	calling-DL-address	[7498-3]
3.1.5	centralized multi-end-point-connection	[7498-1]
3.1.6	DL-connection	[7498/AD1]
3.1.7	DL-connection-end-point	[7498-1]
3.1.8	DL-connection-end-point-identifier	[7498-3]

3.1.9	DL-connection-mode transmission	[7498/AD1]
3.1.10	DL-connectionless-mode transmission	[7498/AD1]
3.1.11	correspondent (N)-entities correspondent DL-entities correspondent Ph-entities	[7498-1]
3.1.12	DL-duplex-transmission	[7498-1]
3.1.13	DL-entity	[7498-1]
3.1.14	DL-facility	[7498-1]
3.1.15	flow control	[7498-1]
3.1.16	(N)-layer DL-layer Ph-layer	[7498-1]
3.1.17	layer-management	[7498-1]
3.1.18	DL-local-view	[7498-3]
3.1.19	DL-name	[7498-3]
3.1.20	naming-(addressing)-domain	[7498-3]
3.1.21	peer-entities	[7498-1]
3.1.22	primitive name	[7498-3]
3.1.23	DL-protocol	[7498-1]
3.1.24	DL-protocol-connection-identifier	[7498-1]
3.1.25	DL-protocol-data-unit	[7498-1]
3.1.26	DL-relay	[7498-1]
3.1.27	reset	[7498-1]
3.1.28	responding-DL-address	[7498-3]
3.1.29	routing	[7498-1]
3.1.30	segmenting	[7498-1]
3.1.31	DL-service	[7498-1]
3.1.32	DL-service-access-point	[7498-1]

3.1.33	DL-service-access-point-address	[7498-3]
3.1.34	DL-service-connection-identifier	[7498-1]
3.1.35	DL-service-data-unit	[7498-1]
3.1.36	DL-simplex-transmission	[7498-1]
3.1.37	DL-subsystem	[7498-1]
3.1.38	DL-user-data	[7498-1]

3.2 Service convention definitions

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

3.2.1	acceptor
3.2.2	asymmetrical service
3.2.3	confirm (primitive); requestor.deliver (primitive)
3.2.4	deliver (primitive)
3.2.5	DL-confirmed-facility
3.2.6	DL-facility
3.2.7	DL-mandatory-facility
3.2.8	DL-non-confirmed-facility
3.2.9	DL-provider-initiated-facility
3.2.10	DL-provider-optional-facility
3.2.11	DL-service-primitive; primitive
3.2.12	DL-service-provider
3.2.13	DL-service-user
3.2.14	DL-user-optional-facility
3.2.15	indication (primitive); acceptor.deliver (primitive)
3.2.16	multi-peer

- 3.2.17 request (primitive);
requestor.submit (primitive)
- 3.2.18 requestor
- 3.2.19 response (primitive);
acceptor.submit (primitive)
- 3.2.20 submit (primitive)
- 3.2.21 symmetrical service

3.3 Data Link Service definitions

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

3.3.1 DL-segment, link, local link: A single DL-subnetwork in which any of the connected DLEs may communicate directly, without any intervening DL-relaying, whenever all of those DLEs 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 extended link: A DL-subnetwork, consisting of the maximal set of links interconnected by DL-relays, sharing a single DL-name (DL-address) space, in which any of the connected DL-entities may communicate, one with another, either directly or with the assistance of one or more of those intervening DL-relay entities.

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

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

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

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

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

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

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

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

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



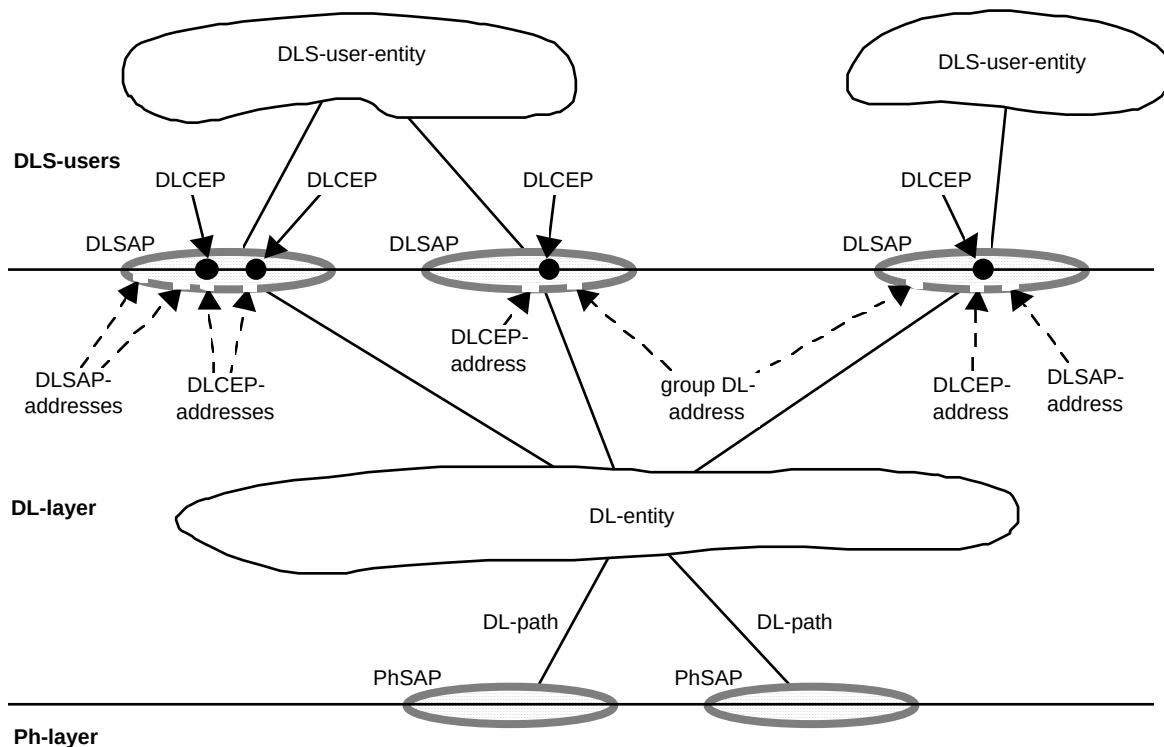
asymmetrical service. It may also be negotiated to provide only DL-simplex service, either from the publisher to the subscribers, or from the subscribers to the publisher.

NOTES

1. In this last case, the characterizations as publisher and subscriber are misnomers.
2. The publishing DLS-user may need to employ control of its publishing rate, because a subscribing DLS-user cannot exert either flow or rate control on its publishing peer entity. Similar considerations apply to subscribing DLS-users with respect to their sending DLSDUs to the publishing DLS-user.

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

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



LEGEND:

1. DLSAPs and PhSAPs are depicted as ovals spanning the boundary between two adjacent layers.
2. DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP. A DLCEP-address also designates a specific point of information flow (its DLCEP) within the DLSAP.
3. A single DL-entity may have multiple DLSAP-addresses and group DL-addresses associated with a single DLSAP.
4. This figure also shows the relationships of DL-paths and PhSAPs.

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

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

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

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

3.3.13 DLCEP-address: A DL-address that designates either:

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

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

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

3.3.14 DLSEP-address: A DL-address that designates a DL-scheduling-end-point within a DLE.

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

3.3.15 transaction: A single DLPDU, or a sequence of two immediately consecutive related DLPDUs, resulting from a single DLS-user request.

3.3.16 initiator: A DLE role in which a DLE sends a DLPDU to a peer responder DLE, which immediately sends a reply DLPDU back to the initiator DLE (and potentially to other DLEs) as part of the same transaction.

NOTE — Some prior standards have referred to this role as a “master” role.

3.3.17 responder: A DLE role in which a DLE sends a DLPDU as an immediate reply to a DLPDU received from a peer initiator DLE, all as part of a single transaction.

NOTE — Some prior standards have referred to this role as a “slave” role.

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

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

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

NOTE — These concepts also support migration from previous standards.

4 Symbols and abbreviations

DL-	Data Link layer (as a prefix)
DLC	DL-connection
DLCEP	DL-connection-end-point
DLE	DL-entity (the local active instance of the Data Link layer)
DLL	DL-layer
DLPDU	DL-protocol-data-unit
DLS	DL-service
DLSAP	DL-service-access-point
DLSDU	DL-service-data-unit
DLSEP	DL-schedule-end-point
FIFO	First-in first-out (queuing method)
OSI	Open systems interconnection
Ph-	Physical layer (as a prefix)
PhE	Ph-entity (the local active instance of the Physical layer)
PhL	Ph-layer
QoS	Quality of service

5 Conventions

5.1 General conventions

This part of ISA-S50.02 uses the descriptive conventions given in ISO TR/8509 and more recently in ISO/IEC 10731.

The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

5.2 Parameters

Service primitives, used to represent service user/service provider interactions (ISO/TR 8509 and ISO/IEC 10731), convey parameters that indicate information available in the user/provider interaction.

This part of ISA-S50.02 uses a tabular format to describe the component parameters of the DL-service primitives. The parameters that apply to each group of DL-service-primitives are set out in tables in clauses 2 through 4, 6 through 8, 13 through 21, 23 through 25, and 27 through 31. Each table consists of up to six columns, containing the name of the service parameter, and a column each for those primitives and parameter-transfer directions used by the DLS:

- the request primitive's input parameters;
- the request primitive's output parameters;
- the indication primitive's output parameters;
- the response primitive's input parameters; and
- the confirm primitive's output parameters.

NOTE — The request, indication, response, and confirm primitives are also known as requestor.submit, acceptor.deliver, acceptor.submit, and requestor.deliver primitives, respectively [ISO/IEC 10731].

One parameter (or part of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive and parameter direction specified in the column:

“M”—parameter is mandatory for the primitive.

“U”—parameter is a User option and may or may not be provided depending on the dynamic usage of the DLS-user. When not provided, a default value for the parameter is assumed.

“C”—parameter is conditional upon other parameters or upon the environment of the DLS-user.

“CU”—parameter is a conditional User option and may or may not be permitted depending upon other parameters or upon the environment of the DLS-user. When permitted, it may or may not be provided depending on the dynamic usage of the DLS-user. When permitted and not provided, a default value for the parameter is assumed.

“ ” (blank)—parameter is never present.

Some entries are further qualified by items in brackets. These may be:

a) a parameter-specific constraint

“(=)” indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.

“(≤)” indicates that the set of parameter values has an implicit order and that the parameter's value is less than or equal to that of the parameter in the service primitive to its immediate left in the table (that is, left by one or two columns).

“(≥)” indicates that the set of parameter values has an implicit order and that the parameter’s value is greater than or equal to that of the parameter in the service primitive to its immediate left in the table (that is, left by one or two columns).

b) an indication that some note applies to the entry

“(n)” indicates that the following note **n** contains additional information pertaining to the parameter and its use.

In any particular interface, not all parameters need be explicitly stated. Some may be implicitly associated with the DLSAP at which the primitive is issued.

In the diagrams that illustrate these interfaces, dashed lines indicate cause-and-effect or time-sequence relationships, and wavy lines indicate that events are roughly contemporaneous.

5.3 Identifiers

Most of the DLS primitives specify one or more identifier parameters that are drawn from either a local DL-identifier space or a local DLS-user-identifier space. The existence and use of such identifiers in an implementation of the services specified in this part of ISA-S50.02 is a purely local issue. Nevertheless, these identifiers are specified explicitly in these primitives to provide a descriptive means:

- a) of relating a confirm primitive with its corresponding request primitive;
- b) of relating a response primitive with its corresponding indication primitive;
- c) of canceling (aborting) an outstanding request primitive before receiving its corresponding confirm primitive;
- d) for referring within a request or response primitive to persistent DL-objects, such as buffers and queues, that were created as the result of a previous DLS primitive; and
- e) for referring within an indication or confirm primitive to persistent DL-objects that were created as the result of a previous DLS primitive.

Adherence to the OSI principle of architectural layering necessitates the presumption of distinct non-intersecting identifier spaces for the DLS-provider and each separate DLS-user, because they may have non-overlapping local views. Consequently, DLS-user identifiers are required for (a) and (e); while DL-identifiers are required for (b), (c), and (d).

6 Overview of the Data Link Service

The DLS provides for the transparent and reliable transfer of data between DLS-users. It makes the way that supporting communications resources are utilized invisible to these DLS-users.

In particular, the DLS provides for the following:

- a) Independence from the underlying Physical Layer. The DLS relieves DLS-users from all direct concerns regarding which configuration is available (for example, direct connection, or indirect through one or more bridges) and which physical facilities are used (for example, which of a set of diverse physical paths).

- b) Transparency of transferred information. The DLS provides for the transparent transfer of DLS-user-data. It does not restrict the content, format, or coding of the information, nor does it ever need to interpret the structure or meaning of that information. It may, however, restrict the amount of information that can be transferred as an indivisible unit.

NOTE — A DLS-user may segment arbitrary-length data into limited-length DLSDUs before making DL-service requests, and may reassemble received DLSDUs into those larger data units.

- c) Reliable transfer of data. The DLS relieves the DLS-user from concerns regarding insertion or corruption of data, or, if requested, loss, duplication, or misordering of data, which may occur. In some cases of unrecovered errors in the Data Link Layer, duplication or loss of DLSDUs may occur. In cases where protection against misordering of data is not employed, misordering can occur.

NOTE — Detection of duplicate, lost or misordered DLSDUs may be performed by DLS-users.

- d) Quality of Service (QoS) selection. The DLS provides DLS-users with a means to request and to agree upon a quality of service for the transfer of data. QoS is specified by means of QoS parameters representing aspects such as mode of operation, transit delay, accuracy, and reliability.
- e) Addressing. The DLS allows the DLS-user to identify itself and to specify the DLSAPs between which a DLC is to be established. DL-addresses have only regional significance within a specific DL-segment. Extended DL-addresses have only regional significance within a specific DL-subsystem over a set of bridged DL-segments. Therefore, it is not appropriate to define a global addressing structure.

NOTE — The DLS is required to differentiate between the individual systems that are physically or logically connected to a multi-point data link and to differentiate between connections. For commonality with other service definitions, this mechanism is referred to as addressing and the objects used to differentiate between systems are referred to as addresses. In a formal sense, this is an extension of the use of addresses beyond that specified in ISO 7498-3.

- f) Scheduling. The DLS allows the set of DLS-users to provide some guidance on the internal scheduling of the distributed DLS-provider. This guidance supports the time-critical aspects of the DLS, by permitting the DLS-user some degree of management over when opportunities for communication will be granted to various DLEs for various DLSAP-addresses and DLCEPs.
- g) Common time sense. The DLS can provide the DLS-user with a sense of time that is common to all DLS-users on the extended link.
- h) Queues and buffers. The DLS can provide the sending or receiving DLS-user with either a FIFO queue or a retentive buffer or a non-retentive buffer (that is, one that becomes empty after being read), where each queue item or buffer can hold a single DLSDU.

6.1 Overview of DL-subnetwork structuring

A DL-subnetwork consists of a set of DL-segments (links) interconnected by DL-relay entities (bridges) that provide DL-layer-internal coordination and routing services to the set of interconnected DLEs.

A DL-segment consists of a set of DLEs, all of which are connected directly (that is, without intervening DL-relay entities) to a single shared logical communications channel, known as a **link**.

A link (logical communications channel) consists of one or more physically-independent logically-parallel cooperatively-scheduled real communications channels, which are known as **paths**.

A single shared PhL-provider enables communications among the DLEs on a given path. A link is made up of conceptually parallel paths. An example is shown in Figure 3.

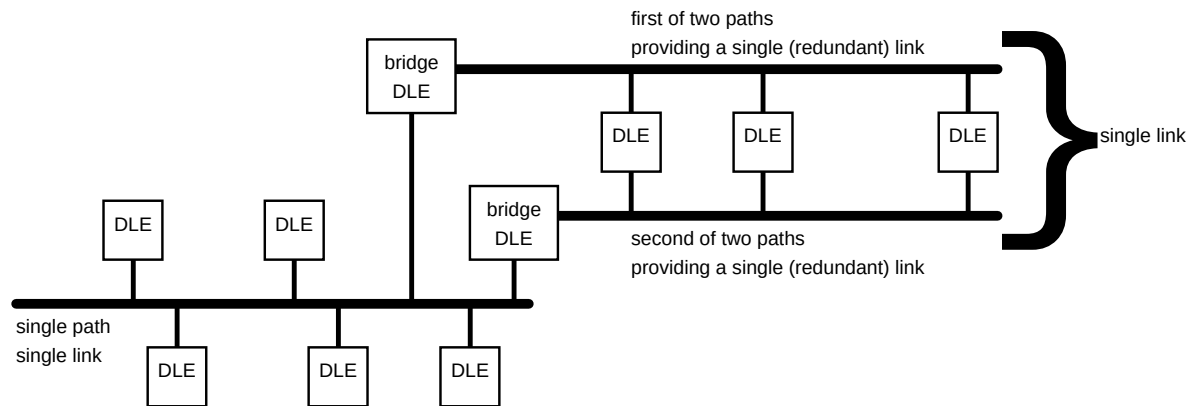


Figure 3 — Example of paths, links, bridges, and the extended link

NOTES

1. A link consisting of more than one path is an instance of DL-redundancy. This is distinct from Ph-redundancy, which is necessarily hidden from the DLL and all DLEs due to the principles of layering (ISO 7498-1).
2. In a logical sense, DLEs are connected to links and bridges interconnect links. Yet in a physical sense, DLEs are connected to paths and bridges interconnect paths. DL-communication-services are independent of the specific path employed, and the DLS-user has no cognizance of any path multiplicity.

6.2 Overview of DL-naming (addressing)

DL-names, known conventionally as DL-addresses, are identifiers from a defined identifier space — the DL-address-space — that serve to name objects within the scope of the data link layer. Examples of such objects are data-link-service-access-points (DLSAPs), data-link-connection-end-points (DLCEPs), and data-link-entities (DLEs).

The DL-address-space from which DL-addresses are drawn may be partitioned into sub-spaces of DL-addresses:

- a) to cluster addresses of the same generic function, such as
 - 1) DLSAP-addresses naming specific DLSAPs;
 - 2) DL-addresses naming groups of DLSAPs;
 - 3) DL-addresses designating one or more DLCEPs; and
 - 4) DL-addresses designating DLEs;
- b) to cluster addresses for administrative purposes, such as addresses that are
 - 1) known to be local to, and allocable by, a single DLE;
 - 2) known to be local to a single link (DL-segment) but not to have a known specific DLE-locality; or
 - 3) known to have no implicit DL-locality;

- c) to cluster addresses for routing purposes, such as addresses that are known to be local to a single DL-segment or a single DLE.

6.2.1 Functional partitioning of the DL-address-space

The space of DL-addresses may be partitioned functionally as follows:

- a) DLE-specific DLSAP-addresses;
- b) other DLSAP-addresses;

NOTE — These addresses can designate at most one DLSAP within a single DLE within the entire set of DL-interconnected DLEs.

- c) group DL-addresses designating a group of DLSAPs;

NOTE — These addresses are sometimes (incorrectly) referred to as group-DLSAP-addresses (see 3.3.11).

- d) DLE-specific DLCEP-addresses;
- e) other DLCEP-addresses;
- f) specific aspects of a specific DLE; or
- g) specific aspects of a group of DLEs.

6.2.2 Administrative partitioning of the DL-address-space

The space of DL-addresses may be partitioned administratively as follows:

- a) DLE-specific DL-addresses, which are known to refer to objects within the scope of a specific DLE, and which are allocable by that DLE;
- b) DL-segment (link) specific but DLE-independent DL-addresses, which are known to refer to objects within the scope of a specific DL-segment, and which are allocable locally by the DL-address-space administrator for that DL-segment; or
- c) DL-segment-independent DL-addresses, which are known to refer to objects within the DL-connected set of DL-segments, and which are allocable by the DL-address-space administrator for the connected set of DL-segments.

NOTE — A DL-address-space administrator can always allocate a set of addresses to a subordinate administrator for its sub-administration. For example, the DL-administrator for the entire set of DL-connected DL-segments can allocate a contiguous block of unassigned DL-addresses to the DL-administrator for a specific DL-segment, or to the local administrator within a single DLE.

6.2.3 Routing-related partitioning of the DL-address-space

The space of DL-addresses may be partitioned to assist DL-routing activities as follows:

- a) DLE-specific DL-addresses;
- b) DL-segment (link) specific but DLE-independent DL-addresses; or
- c) DL-segment-independent DL-addresses.

7 Types and classes of Data Link Service

There are four types of DLS:

- a) a DL(SAP)-address, queue and buffer management service (defined in clauses 9 through 12);
- b) a connection-mode data transfer service with four classes of service (defined in clauses 13 through 19);
- c) a connectionless-mode data transfer service with three classes of service (defined in clauses 20 through 24); and
- d) a time and transaction scheduling service (defined in clauses 25 through 29).

All four types of DLS are always provided; the DLS-user may choose those most appropriate for use. Within the DLS types, the DLS-user is limited to those classes of service supported by the selected DL-protocol implementation.

NOTE — Classes of service are defined in detail in the clauses that describe a specific type of DLS.

8 Quality of Service (QoS) attributes common to multiple types of Data Link Service

A DLS-user may select, directly or indirectly, many aspects of the various data link services. The term “Quality of Service” (QoS) refers to those aspects that are under the direct control of the DLS-provider. QoS can only be properly determined when DLS-user behavior does not constrain or impede the performance of the DLS.

Static QoS attributes are selected once for an entire type of DLS. **Dynamic** QoS attributes are selected independently at each DL-service invocation. **Semi-static** QoS attributes are static attributes for one type of DLS and serve as defaults for corresponding dynamic attributes in another type of DLS.

Most QoS attributes have default values that can be set by DL-management and then overridden on a per-DLSAP-address basis by the DLS-user.

NOTE — The existence of multiple levels of default QoS attribute values and of means of setting those default values can simplify use of the DL-services. Some implementations of this DL-service may provide additional levels of default QoS beyond those specified in this part of ISA-S50.02.

Four QoS attributes of the DL-data transfer services apply conceptually to both connection-mode and connectionless operation. The DLS-user may specify values for these attributes when binding a DLSAP-address to the DLS-user’s DLSAP; any unspecified attributes will assume the default values set by DL-management:

- a) Two of these four attributes are considered dynamic; their DLSAP-address-related values serve merely as defaults for each appropriate DL-service-invocation and can be overridden on an instance by instance basis.

- b) The third and fourth attributes are semi-static; they are static for connectionless DL-services, but dynamic for all DLCEP-establishment requests and responses, where its value serves merely as a default for each appropriate DL-service-invocation and can be overridden on an instance by instance basis.

A fifth attribute applies only to connection-mode operation and is dynamic for each DLCEP.

8.1 DLL priority (dynamic QoS attribute)

All DLCEP establishment requests and responses, all connectionless data transfer requests, and many DL-scheduling requests, specify an associated DLL priority used in scheduling DLL data transfer services. This DLL priority also determines the maximum amount of DLS-user-data that can be conveyed in a single DLPDU. This maximum is determined by the DL-protocol specification.

The DL-protocol shall support three DLL priority levels, each of which shall be capable of conveying a specified amount of DLS-user data per appropriate DLPDU. The three DLL priorities with their corresponding ranges of conveyable DLS-user-data (per DLPDU) are, from highest priority to lowest priority:

- a) urgent — capability of conveying up to 64 DLS-user octets per DLPDU;
- b) normal — capability of conveying up to 128 DLS-user octets per DLPDU; and
- c) time-available — capability of conveying up to 256 DLS-user octets per DLPDU.

NOTES

1. URGENT and NORMAL are considered **time-critical** priority levels; TIME-AVAILABLE is considered a **non-time-critical** priority level.
2. DLCEP establishment may negotiate URGENT to NORMAL or TIME-AVAILABLE, or NORMAL to TIME-AVAILABLE.

The default QoS value can be set by DL-management; when not so set its value is TIME-AVAILABLE.

8.2 DLL maximum confirm delay (dynamic QoS attribute)

Each DLCEP establishment request, and each response, specifies upper bounds on the maximum time duration permitted for the completion of each related instance of a sequence of connection-oriented DLS primitives:

- a) the common maximum time for completion of
 - a related sequence of DL-CONNECT primitives;
 - a related sequence of DL-RESET primitives;
 - a related sequence of DL-SUBSCRIBER-QUERY primitives; and
- b) the maximum time for completion of a related sequence of DL-DATA primitives.

Each connectionless service request specifies an upper bound on the maximum time duration permitted for the completion of each related instance of a sequence of connectionless DLS primitives:

- c) the maximum time for completion of a related sequence of locally-confirmed DL-UNITDATA primitives; and

- d) the common maximum time for completion of
- a related sequence of remotely-confirmed DL-UNITDATA primitives;
 - a related sequence of DL-LISTENER-QUERY primitives; or
 - an instance of the DL-UNITDATA-EXCHANGE service.

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

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

The intervals specified are the maximum permissible delays:

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

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

The default QoS values can be set by DL-management; when not so set the value for each of these QoS parameters is **UNLIMITED**.

8.3 DLPDU authentication (semi-static QoS attribute)

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

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

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

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

NOTE — This continuing background transmission is known as **residual activity**.

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

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

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

- 1) **ORDINARY** — each DLPDU shall include the minimum permitted amount of addressing information;
- 2) **SOURCE** — each DLPDU shall include a source DL-address where possible;



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

The default QoS value can be set by DL-management; when not so set its value is ORDINARY. DLCEP establishment may negotiate ORDINARY to source to maximal.

8.4 DL-scheduling-policy (semi-static QoS attribute)

This attribute is static for connectionless services, but is dynamic for connection-mode services.

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

NOTE — DLC support services such as DL-CONNECT, DL-RESET, and DL-DISCONNECT, and intra-DL-provider services such as DL-SUBSCRIBER-QUERY and DL-LISTENER-QUERY, are not affected by this attribute.

The two choices are:

- a) **IMPLICIT** — any required communications with peer DLS-user(s) from this DLSAP-address, or from this DLCEP, will occur as soon as possible;

NOTE — The choice IMPLICIT is incompatible with a DLCEP that is bound as sender to a buffer, because writing to a buffer does not trigger transmission. Thus the only usable choice for a sending buffer is EXPLICIT.

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

NOTE — Possible use of previously scheduled communications opportunities makes it possible for this deferral and subsequent release to result in earlier communications with the peer DLS-users than that provided by the IMPLICIT alternative.

The default QoS value cannot be set by DL-management; its value is always IMPLICIT.

8.5 DL-timeliness (dynamic DLCEP QoS attributes)

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

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

Types of DL-Timeliness

in terms of elapsed DL-time and events at the assessing DLCEP

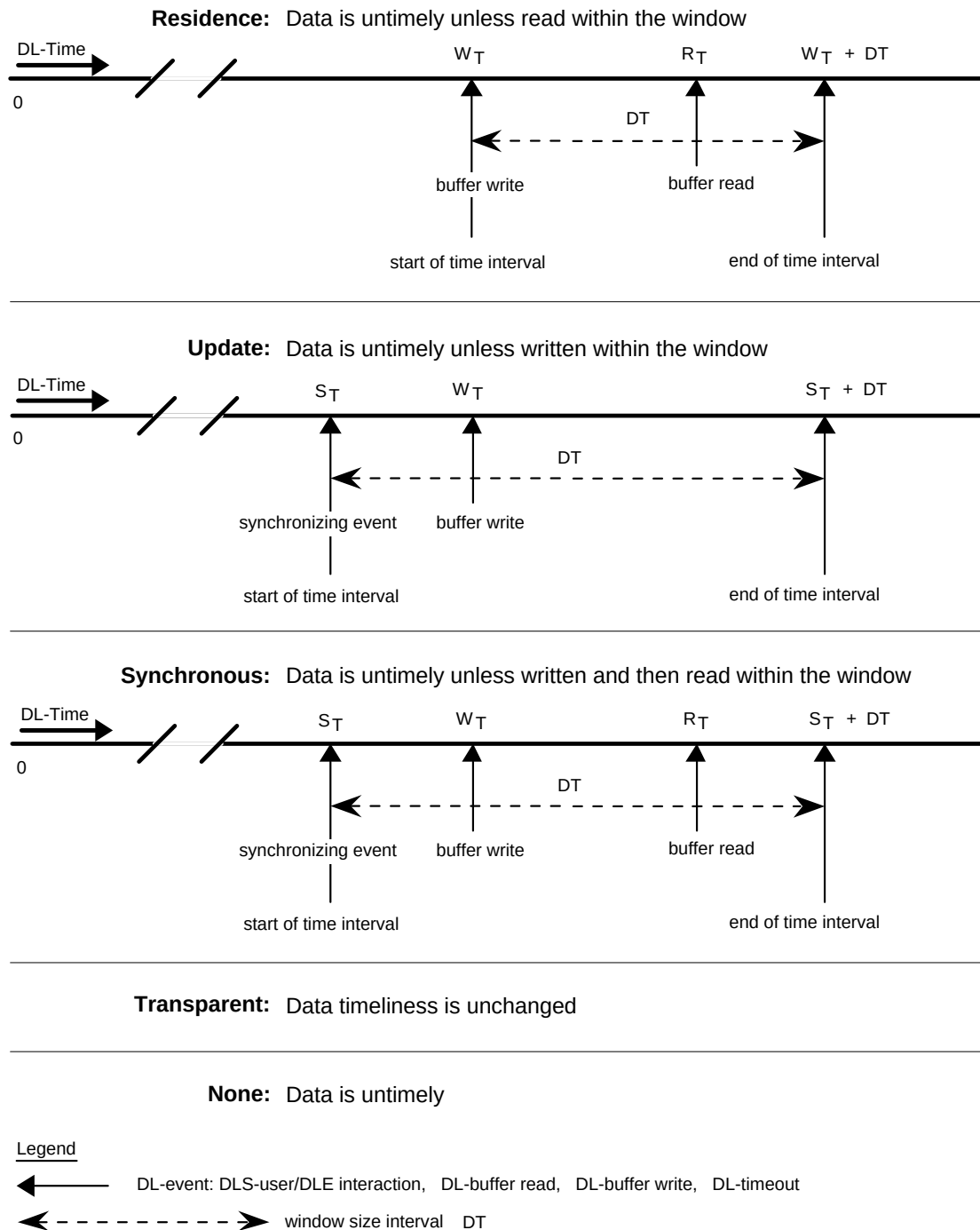


Figure 4 — Types of DL-timeliness

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

- 2) the moment when the buffer is read (by a DL-GET request primitive or by transmission from the buffer at a DLCEP);

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

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

b) **UPDATE** timeliness is an assessment based upon the time interval between

- 1) the moment of occurrence of a multi-DLE synchronizing event (a DL-BUFFER-RECEIVED indication or DL-BUFFER-SENT indication); and
- 2) the moment when the buffer is written (by a DL-PUT request primitive or by reception into the buffer at a DLCEP);

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

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

c) **SYNCHRONIZED** timeliness is an assessment based upon the time intervals and timing relationships between

- 1) the moment of occurrence of a multi-DLE synchronizing event (a DL-BUFFER-RECEIVED indication or DL-BUFFER-SENT indication);
- 2) the moment when the buffer is written (by a DL-PUT request primitive or by reception into the buffer at a DLCEP); and
- 3) the moment when the buffer is read (by a DL-GET request primitive or by transmission from the buffer at a DLCEP);

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

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

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

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

The DL-time when the original buffer is written by a DL-PUT request primitive also can be conveyed to DLS-users that read a copy of the buffer. This DL-time is not available when the buffer timeliness is FALSE.

NOTE — DL-time is described in clauses 25 through 29.

9 Facilities of the DL(SAP)-address, queue and buffer management Data Link Service

The DLS provides the following facilities to the DLS-user:

- a) A means for creating and deleting a retentive buffer, a non-retentive buffer, or a FIFO queue of specified depth, for use
 - 1) in communicating DLS-user-data between a DLS-user and the DLS-provider;
 - 2) in redistributing received DLS-user data without continuing DLS-user intervention; and
 - 3) in facilitating DLS-user supervision of the timing of DLSDU-conveyance to peer DLS-users;
- b) A means for associating an individual DLSAP-address or group DL-address, referred to collectively as a DL(SAP)-address, with, and disassociating a DL(SAP)-address from, the DLSAP at which the request is made.

Default values for some Quality of Service (QoS) attributes for connection-mode and connectionless data transfer services using the specified DL(SAP)-address can be specified when the association is made.

Additionally, the DLS-user may specify that previously created buffers or FIFO queues be used for each potential direction and priority of connectionless data transfer at the specified DL(SAP)-address.

- c) A means by which DLSDUs of limited size are written to or read from a buffer or read from a FIFO queue.

10 Model of the DL(SAP)-address, queue and buffer management Data Link Service

This part of ISA-S50.02 uses the abstract model for a layer service defined in clause 4 of ISO/TR 8509 and ISO/IEC 10731. The model defines interactions between the DLS-user and the DLS-provider that take place at a DLSAP. Information is passed between the DLS-user and the DLS-provider by DL-service-primitives that convey parameters.

The DL(SAP)-address, queue and buffer management primitives are used to provide a local service between a DLS-user and the local DLE. Remote DLEs and remote DLS-users are not directly involved, so there is no need for the other primitives (indication, response, confirm) of ISO/TR 8509. Therefore the DL(SAP)-address, queue and buffer management services are provided by request (requestor.submit) primitives with input and output parameters.

11 Sequence of primitives at one DLSAP

11.1 Constraints on sequence of primitives

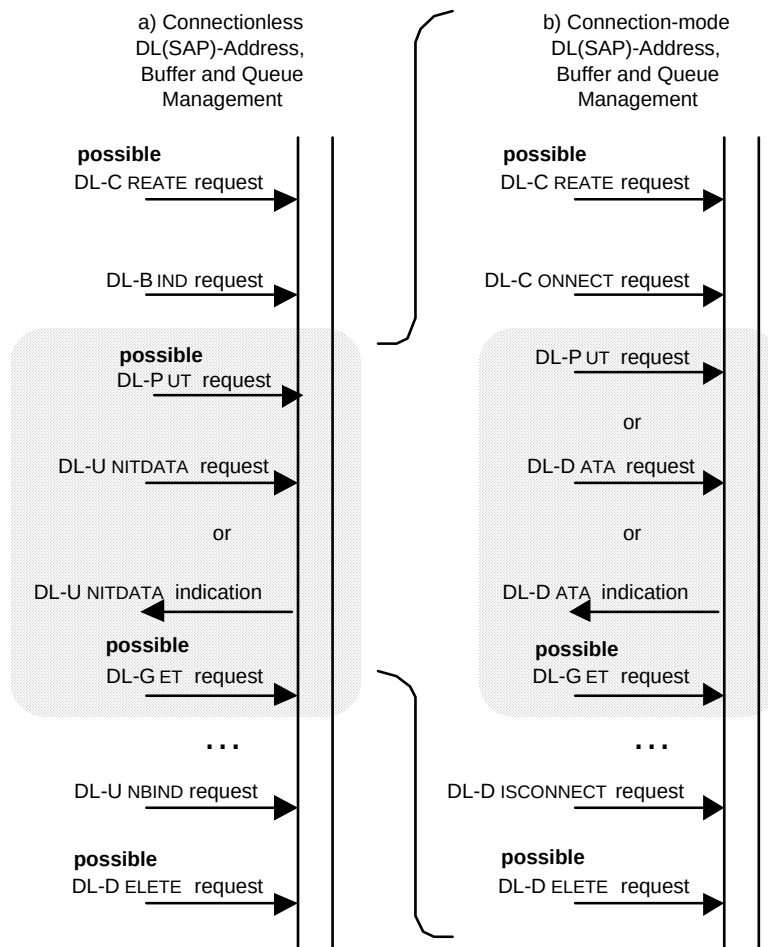
Table 1 — Summary of DL(SAP)-address, queue and buffer management primitives and parameters

Service	Primitive	Parameter
Buffer or Queue Creation	DL-CREATE request	(<i>in</i> Buffer-or-queue DLS-user-identifier, Queuing policy, Maximum DLSDU size, <i>out</i> Status, Buffer-or-queue DL-identifier)
Buffer or Queue Deletion	DL-DELETE request	(<i>in</i> Buffer-or-queue DL-identifier, <i>out</i> Status)
DL(SAP)-address Activation	DL-BIND request	(<i>in</i> DL(SAP)-address DLS-user-identifier, DL(SAP)-address, DL(SAP)-role, Receiving-buffer-or-queue-bindings, Sending-buffer-or-queue-bindings, Default-QoS-as-sender, <i>out</i> Status, DL(SAP)-address DL-identifier)
DL(SAP)-address Deactivation	DL-UNBIND request	(<i>in</i> DL(SAP)-address DL-identifier)
Update Buffer	DL-PUT request	(<i>in</i> Buffer DL-identifier, DLS-user-data, DLS-user-data-timeliness, <i>out</i> Status)
Copy Buffer or Dequeue	DL-GET request	(<i>in</i> Buffer-or-queue DL-identifier, <i>out</i> Status, Reported-service-identification-class, Reported-service-identification, DLS-user-data, DLS-user-data-timeliness)

NOTE — DL-identifiers in parameters are local and assigned by the DLS-provider and used by the DLS-user to designate a specific DL(SAP)-address, DLCEP, schedule, buffer-or-queue to the DLS-provider at the DLS interface. DLS-user-identifiers in parameters are local and assigned by the DLS-user and used by the DLS-provider to designate a specific DL(SAP)-address, DLCEP, schedule, buffer-or-queue to the DLS-user at the DLS interface.

This subclause defines the constraints on the sequence in which the primitives defined in clause 12 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur.

The DL(SAP)-address, queue and buffer management primitives and their parameters are summarized in Table 1. The major relationships among the primitives at a single DLE are shown in Figure 5.



LEGEND —

- 1.Primitives within the gray-background areas can be repeated many times between instances of the primitives in the clear-background areas.
- 2.The entire right-hand part of the figure is another alternative to the gray-background area of the left-hand part of the figure, and also can be repeated many times between instances of the primitives in the left-hand clear-background areas.
- 3.DL-PUT and DL-GET request primitives both can be used earlier and later than shown.

Figure 5 — Sequence of primitives for the DL(SAP)-address, queue and buffer management DL-services

12 DL(SAP)-address, queue and buffer management facilities

DL(SAP)-address management facilities bind a DL(SAP)-address to, and unbind a previously bound DL(SAP)-address from, the DLSAP at which the primitive is invoked. Such a binding is required while communicating using the specified DL(SAP)-address.

Queue and buffer management facilities permit, but do not require, a DLS-user to use retentive or non-retentive buffers or specified depth FIFO queues when employing the DLS-provider's data communications facilities. (See Figure 6.) Since these buffers and queues are managed by the

DLS-provider, they support DLS-user interactions and data transfer and scheduling paradigms not available with DLS-user-based queuing.

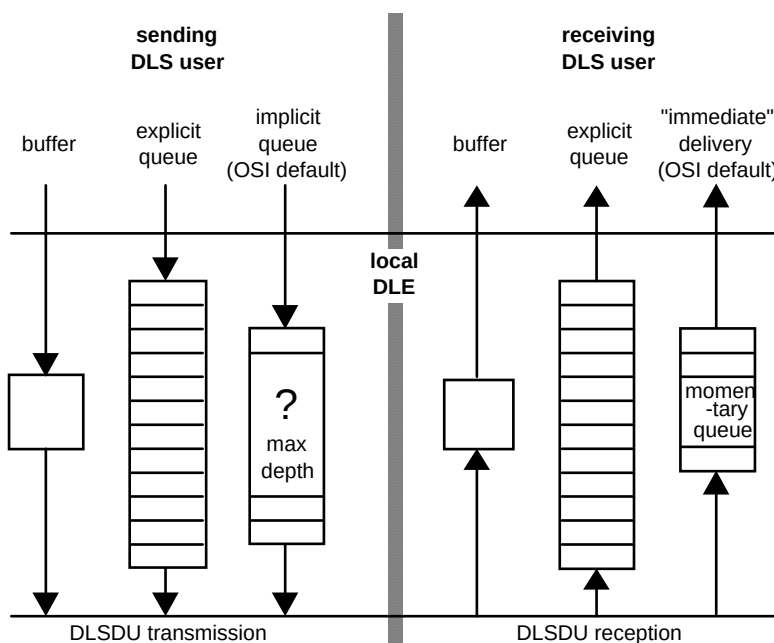


Figure 6 — Supported methods of data management for transmission and delivery

12.1 Create

12.1.1 Function

The create buffer or queue DL-service-primitive can be used to create a retentive buffer or non-retentive buffer or limited-depth FIFO queue for later constrained association with a DLSAP — either through a DL(SAP)-address or through a DLCEP. The resulting buffer or FIFO queue initially will be empty.

NOTE — This facility may also be provided by local DL-management actions, which are beyond the scope of this standard.

12.1.2 Types of parameters

Table 2 indicates the primitive and parameters needed for the create buffer or queue DL-service.

Table 2 — DL-buffer-and-queue-management Create primitive and parameters

DL-Create	Request	
	input	output
Buffer-or-queue DLS-user-identifier	M	
Queuing policy	M	
Maximum queue depth	C	
Maximum DLSDU size	M	
Status		M
Buffer-or-queue DL-identifier		C

12.1.2.1 Buffer-or-queue DLS-user-identifier

The buffer-or-queue DLS-user-identifier parameter specifies a means of referring to the buffer or queue in output parameters of other local DLS primitives that convey the name of the buffer or queue from the local DLE to the local DLS-user.

The naming-domain of the buffer-or-queue DLS-user-identifier is the DLS-user's local-view.

12.1.2.2 Queuing policy

The Queuing Policy parameter specifies whether to create either:

- a) **BUFFER-R** — a retentive buffer, whose contents are not affected by being read, which can be overwritten (as a single atomic action) by either the DLS-provider or DLS-user, and which can be used only with the connectionless unitdata-exchange and connection-oriented data services; or
- b) **BUFFER-NR** — a non-retentive buffer, which is set empty after being read, which can be overwritten (as a single atomic action) by either the DLS-provider or DLS-user, and which can be used only with the connectionless unitdata-exchange service; or
- c) **QUEUE** — a FIFO queue of maximum depth K that contains between zero and K DLSDUs, which will reject attempts to remove DLSDUs when empty and to append DLSDUs when full, and which can be used with the connectionless unitdata-transfer and connection-oriented data services, and for DLSDU-receipt with the connectionless unitdata-exchange service.

The values of the Queuing Policy are BUFFER-R, BUFFER-NR, and QUEUE.

NOTES

1. Buffer and queue bindings result from DL-BIND request, DL-CONNECT request, and DL-CONNECT response primitives.
2. An explicit queue must have one output binding and at least one input binding to be useful. Multiple input bindings provide a means of coalescing inputs from many sources into a single queue. Multiple output bindings are not permitted, because a queue-not-empty condition typically causes transmission to be attempted at the DLCEP, or from the DLSAP-address, to which the queue is bound.
3. A retentive buffer (BUFFER-R) must have one input binding and at least one output binding to be useful. Multiple input bindings are not permitted, because they would permit any data source to overwrite the buffer at any time, with no indication to the buffer users of which source was involved. Multiple output bindings are useful, to share the contents of the buffer among multiple users or to support differing-rate redistribution of cached data within bridges.
4. A non-retentive buffer (BUFFER-NR) must have one input binding and one output binding to be useful. Multiple input bindings are not useful for the reason stated in 2. Multiple output bindings are not useful; their existence would lead to non-intuitive semantics for the buffer.
5. Buffers can be overwritten at any time, with complete loss of the prior contents except to those users that had begun to read the prior contents of the buffer (via DL-GET request primitives or sending DLCEPs) before the overwriting begins. Therefore buffers are well suited for caching of distributed data, where it is desired to retain the most recent value of received information, and are poorly suited for general messaging.

12.1.2.2.1 Maximum queue depth

The Maximum Queue Depth parameter is present when the Queuing Policy parameter has the value QUEUE. When present, this parameter specifies K , the maximum number of items in the associated queue. The supported values for this parameter shall include the values one (1), two (2), three (3), and four (4).

NOTE — Implementations of this DL-service may extend the range of this parameter to include:

- a) values greater than four (4); and
- b) the value zero (0), creating a null queue.

12.1.2.3 Maximum DLSDU size

The Maximum DLSDU Size parameter specifies an upper bound on the size (in octets) of DLSDUs that can be put into the buffer or queue. The maximum size permitted for such DLSDUs may be constrained by a companion DL-protocol specification and by DL-management

NOTE — This parameter does not preclude implementations from using a fixed, small set of record sizes when allocating buffers or entries in a queue.

12.1.2.4 Status

The status parameter allows the DLS-user to determine whether the requested DL-service was provided successfully or failed for the reason specified. The value conveyed in this parameter will be as follows:

- a) "success"
- b) "failure — insufficient resources"
- c) "failure — parameter violates management constraint"
- d) "failure — number of requests violates management constraint" or
- e) "failure — reason unspecified"

NOTE — Addition to, or refinement of, this list of values to convey more specific diagnostic and management information is permitted in a DL-protocol standard that provides services as specified in this part of ISA-S50.02.

If the status is "success," then the created buffer or queue is subject to the following constraints:

- 1) If a BUFFER-NR or BUFFER-R was created, then it can be written explicitly either
 - by one priority of a receiving DLSAP-address whose DL(SAP)-role is INITIATOR, CONSTRAINED RESPONDER, or UNCONSTRAINED RESPONDER;
 - by a receiving DLCEP; or
 - by a DLS-user through a DL-PUT request primitive, if no other binding exists to write the buffer.

It is not permitted to bind such a buffer so that it is written from two distinct sources.

- 2) If a BUFFER-R or QUEUE was created, then it can be read explicitly either
 - by one priority of a sending DLSAP-address whose DL(SAP)-role is BASIC;
 - by a sending DLCEP; or
 - by a DLS-user through a DL-GET request primitive, if no other binding exists to read the buffer.

It is not permitted to bind such a buffer or queue so that it is read by two distinct sinks.

12.1.2.5 Buffer-or-queue DL-identifier

The buffer-or-queue DL-identifier parameter is present when the Status parameter indicates that the DL-CREATE request primitive was successful. The buffer-or-queue DL-identifier parameter gives the local DLS-user a means of referring to the buffer or queue in input parameters of other local DLS primitives that convey the name of the buffer or queue from the local DLS-user to the local DLE.

The naming-domain of the buffer-or-queue DL-identifier is the DL-local-view.

12.1.3 Sequence of primitives

The sequence of primitives in creating, later using, and eventually deleting a buffer or queue is defined in the time sequence diagram of Figure 5.

12.2 Delete

12.2.1 Function

The delete buffer or queue DL-service-primitive can be used to delete a buffer or queue created by an earlier create buffer or queue DL-service-primitive.

NOTES

1. This primitive can only be used to delete a buffer or queue that was created by a prior DL-CREATE request primitive; it cannot be used to delete a buffer or queue that was created by prior local DL-management actions.
2. This facility may also be provided by local DL-management actions, which are beyond the scope of this standard.

12.2.2 Types of parameters

Table 3 indicates the primitive and parameters needed for the delete buffer or queue DL-service.

Table 3 — DL-buffer-and-queue-management Delete primitive and parameters

DL-Delete	Request	
	input	output
Parameter Name		
Buffer-or-queue DL-identifier	M	
Status		M

12.2.2.1 Buffer-or-queue DL-identifier

The buffer-or-queue DL-identifier parameter specifies the local DL-identifier returned by a successful prior DL-CREATE request primitive whose buffer or queue has not yet been deleted. The DLS-provider will release the local DL-identifier and associated DLS-provider resources.

The DLS-user may not delete a buffer or queue that is still associated with a DLSAP.

NOTE — Such associations can occur only as a result of a DL-BIND request, or DL-CONNECT request or response, or of DL-management action.

12.2.2.2 Status

The status parameter allows the DLS-user to determine whether the requested DL-service was provided successfully or failed for the reason specified. The value conveyed in this parameter will be as follows:

- a) "success"
- b) "failure — resource in use"
- c) "failure — management-controlled resource" or
- d) "failure — reason unspecified"

NOTE — Addition to, or refinement of, this list of values to convey more specific diagnostic and management information is permitted in a DL-protocol standard that provides services as specified in this part of ISA-S50.02.

12.2.3 Sequence of primitives

The sequence of primitives in creating and later deleting a buffer or queue is defined in the time sequence diagram of Figure 5.

12.3 Bind

12.3.1 Function

The bind DL(SAP)-address DL-service-primitive is used:

- a) to associate a DL(SAP)-address with the DLSAP at which the primitive is invoked;
- b) to establish the DL(SAP)'s role, if any, when participating in the DL-Unitdata and DL-Unitdata-Exchange connectionless services at that DL(SAP)-address;
- c) to associate up to six previously created retentive buffers or non-retentive buffers or limited-depth FIFO queues with the various priorities and directions of potential data transfer at the specified DL(SAP)-address; and
- d) to specify default values for some Quality of Service (QoS) attributes for connection-mode and connectionless data transfer services using the specified DL(SAP)-address.

NOTE — This facility may also be provided by local DL-management actions, which are beyond the scope of this standard.

12.3.2 Types of parameters

Table 4 indicates the primitive and parameters needed for the bind DL(SAP)-address DL-service.