

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 3-1: Data-link layer service definition – Type 1 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 3-1: Définition des services de la couche liaison de données – Éléments
de type 1**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 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
Fax: +41 22 919 03 00
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 corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

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 also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC 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: csc@iec.ch.

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.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

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 aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 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.

Service Clients - webstore.iec.ch/csc

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 3-1: Data-link layer service definition – Type 1 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 3-1: Définition des services de la couche liaison de données – Éléments
de type 1**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XF

ICS 25.040.40; 35.100.20; 35.110

ISBN 978-2-8322-1709-2

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

CONTENTS

FOREWORD.....	7
0 INTRODUCTION	9
0.1 General.....	9
0.2 Nomenclature for references within this standard	9
1 Scope.....	10
1.1 General.....	10
1.2 Specifications.....	10
1.3 Conformance.....	10
2 Normative references	11
3 Terms, definitions, symbols, abbreviations and conventions	11
3.1 Reference model terms and definitions.....	11
3.2 Service convention terms and definitions.....	12
3.3 Data-link service terms and definitions	13
3.4 Common symbols and abbreviations	16
3.5 Common conventions	17
4 Overview of the data-link layer service	19
4.1 General.....	19
4.2 Types and classes of data-link layer service.....	21
4.3 Quality-of-service (QoS) attributes common to multiple types of data-link layer service.....	22
5 DL(SAP)-address, queue and buffer management data-link layer service	27
5.1 Facilities of the DL(SAP)-address, queue and buffer management data-link layer service.....	27
5.2 Model of the DL(SAP)-address, queue and buffer management data-link layer service.....	27
5.3 Sequence of primitives at one DLSAP	27
5.4 DL(SAP)-address, queue and buffer management facilities	29
6 Connection-mode data-link layer service	43
6.1 Facilities of the connection-mode data-link layer service	43
6.2 Model of the connection-mode data-link layer service.....	44
6.3 Quality of connection-mode service.....	51
6.4 Sequence of primitives	57
6.5 Connection establishment phase	68
6.6 Connection release phase	75
6.7 Data transfer phase.....	81
7 Connectionless-mode data-link layer service	93
7.1 Facilities of the connectionless-mode data-link layer service	93
7.2 Model of the connectionless-mode data-link layer service	93
7.3 Quality of connectionless-mode service.....	95
7.4 Sequence of primitives	95
7.5 Connectionless-mode functions.....	98
8 Time and scheduling guidance data-link layer service	109
8.1 Facilities and classes of the time and scheduling guidance data-link layer service	109
8.2 Model of the time and scheduling guidance data-link layer service	110
8.3 Quality of scheduling guidance service.....	110
8.4 Sequence of primitives at one DLE.....	110

8.5	Scheduling guidance functions	112
9	DL-management service	123
9.1	Scope and inheritance	123
9.2	Facilities of the DL-management service	123
9.3	Model of the DL-management service	123
9.4	Constraints on sequence of primitives	123
9.5	Set	124
9.6	Get	125
9.7	Action	125
9.8	Event	126
	Bibliography	128

Figure 1	– Relationships of DLSAPs, DLSAP-addresses, DLCEPs, DLCEP-addresses, DLSEP-addresses and group DL-addresses	14
Figure 2	– Example of paths, links, bridges, and the extended link	20
Figure 3	– Types of DL-timeliness In terms of elapsed DL-time and events at the assessing DLCEP	25
Figure 4	– Sequence of primitives for the DL(SAP)-address, queue and buffer management DLS	29
Figure 5	– Supported methods of data management for transmission and delivery	30
Figure 6	– Peer-to-peer and multi-peer DLCs and their DLCEPs	44
Figure 7	– OSI abstract queue model of a peer DLC between a pair of DLS-users	45
Figure 8	– OSI abstract queue model of a multi-peer DLC between a publishing DLS-user and a set of subscribing DLS-users	49
Figure 9	– Summary of DL-connection-mode service primitive time-sequence diagrams for peer DLCs (portion 1)	61
Figure 10	– Summary of DL-connection-mode service primitive time-sequence diagrams for peer DLCs (portion 2)	62
Figure 11	– Summary of DL-connection-mode service primitive time-sequence diagrams for publishers of a multi-peer DLC (portion 1)	63
Figure 12	– Summary of DL-connection-mode service primitive time-sequence diagrams for publishers of a multi-peer DLC (portion 2)	64
Figure 13	– 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 (portion 1)	65
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 (portion 2)	66
Figure 15	– State transition diagram for sequences of DL-connection-mode service primitives at a DLCEP	67
Figure 16	– Peer DLC/DLCEP establishment initiated by a single DLS-user	73
Figure 17	– Multi-peer DLC/DLCEP establishment initiated by the publishing DLS-user	74
Figure 18	– Multi-peer DLC/DLCEP establishment initiated by a subscribing DLS-user	74
Figure 19	– Multi-peer DLC/DLCEP establishment using known DLCEP addresses initiated first by the publishing DLS-user	74
Figure 20	– Multi-peer DLC/DLCEP establishment using known DLCEP addresses initiated first by one or more subscribing DLS-users	74
Figure 21	– Peer DLC/DLCEP establishment initiated simultaneously by both peer DLS-users, resulting in a merged DLC	75

Figure 22 – Multi-peer DLC/DLCEP establishment initiated simultaneously by both publishing and subscribing DLS-users, resulting in a merged DLC	75
Figure 23 – Peer DLS-user invocation.....	78
Figure 24 – Publishing DLS-user invocation.....	78
Figure 25 – Subscribing DLS-user invocation.....	78
Figure 26 – Simultaneous invocation by both DLS-users.....	78
Figure 27 – Peer DLS-provider invocation.....	78
Figure 28 – Publishing DLS-provider invocation	78
Figure 29 – Subscribing DLS-provider invocation	78
Figure 30 – Simultaneous peer DLS-user and DLS-provider invocations	78
Figure 31 – Simultaneous publishing DLS-user and DLS-provider invocations	79
Figure 32 – Simultaneous subscribing DLS-user and DLS-provider invocations	79
Figure 33 – Sequence of primitives in a peer DLS-user rejection of a DLC/DLCEP establishment attempt.....	79
Figure 34 – Sequence of primitives in a publishing DLS-user rejection of a DLC/DLCEP establishment attempt.....	79
Figure 35 – Sequence of primitives in a subscribing DLS-user rejection of a DLC/DLCEP establishment attempt.....	79
Figure 36 – Sequence of primitives in a DLS-provider rejection of a DLC/DLCEP establishment attempt.....	80
Figure 37 – Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: both primitives are destroyed in the queue.....	80
Figure 38 – Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: DL-DISCONNECT indication arrives before DL-CONNECT response is sent.....	80
Figure 39 – 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	80
Figure 40 – 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	81
Figure 41 – 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.....	81
Figure 42 – Sequence of primitives for a CLASSICAL or DISORDERED peer-to-peer queue-to-queue data transfer.....	83
Figure 43 – Sequence of primitives for an ORDERED or UNORDERED peer-to-peer, or an UNORDERED subscriber-to-publisher queue-to-queue data transfer	84
Figure 44 – Sequence of primitives for a publisher-to-subscribers queue-to-queue data transfer	84
Figure 45 – Sequence of primitives for a failed queue-to-queue data transfer	84
Figure 46 – Sequence of primitives for an ORDERED or UNORDERED peer to peer, or an UNORDERED subscriber to publisher, buffer to buffer data transfer	85
Figure 47 – Sequence of primitives for a publisher to subscribers buffer to buffer data transfer	86
Figure 48 – Sequence of primitives for an ORDERED or UNORDERED peer to peer, or an UNORDERED subscriber to publisher, buffer to queue data transfer	86
Figure 49 – Sequence of primitives for a publisher to subscribers buffer to queue data transfer	86

Figure 50 – Sequence of primitives in a peer DLS-user initiated Reset	89
Figure 51 – Sequence of primitives in a publishing DLS-user initiated Reset.....	90
Figure 52 – Sequence of primitives in a subscribing DLS-user initiated Reset.....	90
Figure 53 – Sequence of primitives in a simultaneous peer DLS-users initiated Reset	90
Figure 54 – Sequence of primitives in a simultaneous multi-peer DLS-users initiated Reset.....	90
Figure 55 – Sequence of primitives in a peer DLS-provider initiated Reset.....	90
Figure 56 – Sequence of primitives in a publishing DLS-provider initiated Reset.....	90
Figure 57 – Sequence of primitives in a subscribing DLS-provider initiated Reset	91
Figure 58 – Sequence of primitives in a simultaneous peer DLS-user and DLS-provider initiated Reset.....	91
Figure 59 – Sequence of primitives in a simultaneous publishing DLS-user and DLS- provider initiated Reset	91
Figure 60 – Sequence of primitives in a simultaneous subscribing DLS-user and DLS- provider initiated Reset	91
Figure 61 – Sequence of primitives for Subscriber Query.....	92
Figure 62 – Model for a data-link layer connectionless-mode unitdata transmission or unitdata exchange	94
Figure 63 – Summary of DL-connectionless-mode service primitive time-sequence diagrams.....	97
Figure 64 – State transition diagram for sequences of connectionless-mode primitives at one DLSAP	98
Figure 65 – Sequence of primitives for a successful locally-acknowledged connectionless-mode unitdata transfer.....	101
Figure 66 – Sequence of primitives for a successful remotely-acknowledged connectionless-mode unitdata transfer.....	102
Figure 67 – Sequence of primitives for an unsuccessful connectionless-mode unitdata transfer	102
Figure 68 – Sequence of primitives for connectionless-mode unitdata exchange.....	107
Figure 69 – Sequence of primitives for connectionless-mode listener query	108
Figure 70 – Summary of time and scheduling-guidance service primitive time sequence diagrams.....	112
Figure 71 – Sequence of primitives for DL-time.....	114
Figure 72 – Sequence of primitives for the Compel-Service service	116
Figure 73 – Sequence of primitives for the sequence scheduling services.....	120
Figure 74 – Sequence of primitives for the DLM action service	123
Table 1 – Summary of DL(SAP)-address, queue and buffer management primitives and parameters	28
Table 2 – DL-buffer-and-queue-management create primitive and parameters.....	30
Table 3 – DL-buffer-and-queue-management delete primitive and parameters	33
Table 4 – DL(SAP)-address-management bind primitive and parameters	34
Table 5 – DL(SAP)-role constraints on DLSAPs, DLCEPs and other DLS Primitives	35
Table 6 – DL(SAP)-address-management unbind primitive and parameters	39
Table 7 – DL-buffer-management put primitive and parameters	39
Table 8 – DL-buffer-and-queue-management get primitive and parameters.....	41
Table 9 – Relationships between abstract queue model objects	47

Table 10 – Attributes and class requirements of DLCEP data delivery features	53
Table 11 – Summary of DL-connection-mode primitives and parameters (portion 1).....	59
Table 12 – Summary of DL-connection-mode primitives and parameters (portion 2).....	60
Table 13 – DLC / DLCEP establishment primitives and parameters (portion 1).....	69
Table 14 – DLC / DLCEP establishment primitives and parameters (portion 2).....	70
Table 15 – DLC / DLCEP release primitives and parameters	76
Table 16 – Queue data transfer primitive and parameters	81
Table 17 – Buffer sent primitive and parameter	84
Table 18 – Buffer received primitive and parameter	85
Table 19 – DLC/DLCEP reset primitives and parameters (portion 1)	87
Table 20 – DLC/DLCEP reset primitives and parameters (portion 2)	87
Table 21 – Subscriber query primitives and parameters	92
Table 22 – Summary of DL-connectionless-mode primitives and parameters	96
Table 23 – DL-connectionless-mode unitdata transfer primitives and parameters.....	99
Table 24 – DL-connectionless-mode unitdata exchange primitive and parameters	103
Table 25 – Listener query primitives and parameters	108
Table 26 – Summary of DL-scheduling-guidance primitives and parameters	111
Table 27 – DL-time primitive and parameters	113
Table 28 – DL-scheduling-guidance Compel-service primitive and parameters.....	114
Table 29 – DL-scheduling-guidance Schedule Sequence primitives and parameters	117
Table 30 – DL-scheduling-guidance Cancel Schedule primitives and parameters.....	121
Table 31 – DL-scheduling-guidance Subset Sequence primitives and parameters.....	122
Table 32 – Summary of DL-management primitives and parameters	124
Table 33 – DLM-Set primitive and parameters	124
Table 34 – DLM-Get primitive and parameters	125
Table 35 – DLM-Action primitive and parameters	126
Table 36 – DLM-Event primitive and parameters	127

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 3-1: Data-link layer service definition –
Type 1 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 61158-3-1 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2007. This edition constitutes a technical revision.

The main change with respect to the previous edition is listed below:

- Improved terms.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/759/FDIS	65C/769/RVD

Full information on the voting for the approval of this 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.

NOTE 2 Slight variances from the directives have been allowed by the IEC Central Office to provide continuity of subclause numbering with prior editions.

A list of all the parts of the IEC 61158 series, 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 maintenance result 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.

0 INTRODUCTION

0.1 General

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.

Throughout the set of fieldbus standards, the term “service” refers to the abstract capability provided by one layer of the OSI Basic Reference Model to the layer immediately above. Thus, the data-link layer service defined in this standard is a conceptual architectural service, independent of administrative and implementation divisions.

0.2 Nomenclature for references within this standard

Clauses, including annexes, can be referenced in their entirety, including any subordinate subclauses, as “Clause N” or “Annex N”, where N is the number of the clause or letter of the annex.

Subclauses can be referenced in their entirety, including any subordinate subclauses, as “N.M” or “N.M.P” and so forth, depending on the level of the subclause, where N is the number of the subclause or letter of the annex, and M, P and so forth represent the successive levels of subclause up to and including the subclause of interest.

When a clause or subclause contains one or more subordinate subclauses, the text between the clause or subclause heading and its first subordinate subclause can be referenced in its entirety as “N.0” or “N.M.0” or “N.M.P.0” and so forth, where N, M and P are as above. Stated differently, a reference ending with “.0” designates the text and figures between a clause or subclause header and its first subordinate subclause.

NOTE This nomenclature provides a means of referencing text in hanging clauses. Such clauses existed in earlier editions of IEC 61158-3, Type 1 clauses. Those hanging clauses are maintained in this edition to minimize the disruption to existing national and multi-national standards and consortia documents which reference that prior subclause numbering.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 3-1: Data-link layer service definition – Type 1 elements

1 Scope

1.1 General

This part of IEC 61158 provides common elements for basic 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 standard defines in an abstract way the externally visible service provided by the Type 1 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 which they take; and
- c) the interrelationship between these actions and events, and their valid sequences.

The purpose of this standard is to define the services provided to

- the Type 1 fieldbus application layer at the boundary between the application and data-link layers of the fieldbus reference model;
- systems management at the boundary between the data-link layer and systems management of the fieldbus reference model.

1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual data-link layer services 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. A secondary objective is to provide migration paths from previously existing industrial communications protocols.

This specification may be used as the basis for formal DL-Programming-Interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this specification, including

- a) the sizes and octet ordering of various multi-octet service parameters;
- b) the correlation of paired request and confirm, or indication and response, primitives.

1.3 Conformance

This standard does not specify individual implementations or products, nor does it constrain the implementations of data-link entities within industrial automation systems.

There is no conformance of equipment to this data-link layer service definition standard. Instead, conformance is achieved through implementation of the corresponding data-link protocol that fulfills the Type 7 data-link layer services defined in this standard.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. 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.

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 10731:1994, *Information technology – Open Systems Interconnection – Conventions for the definition of OSI services*

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 Reference model terms and definitions

This standard 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	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	[7498-1]
3.1.7	DL-connection-end-point	[7498-1]
3.1.8	DL-connection-end-point-identifier	[7498-1]
3.1.9	DL-connection-mode transmission	[7498-1]
3.1.10	DL-connectionless-mode transmission	[7498-1]
3.1.11	correspondent (N)-entities	[7498-1]
	correspondent DL-entities (N=2)	
	correspondent Ph-entities (N=1)	
3.1.12	DL-duplex-transmission	[7498-1]
3.1.13	(N)-entity	[7498-1]
	DL-entity (N=2)	
	Ph-entity (N=1)	
3.1.14	DL-facility	

3.1.15	flow control	[7498-1]
3.1.16	(N)-layer DL-layer (N=2) Ph-layer (N=1)	[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	(N)-service DL-service (N=2) Ph-service (N=1)	[7498-1]
3.1.32	(N)-service-access-point DL-service-access-point (N=2) Ph-service-access-point (N=1)	[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	systems-management	[7498-1]
3.1.39	DLS-user-data	[7498-1]

3.2 Service convention terms and definitions

This standard 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 **DLS-user-optional-facility**

3.2.16 **indication (primitive);**
 acceptor.deliver (primitive)

3.2.17 **multi-peer**

3.2.18 **request (primitive);**
 requestor.submit (primitive)

3.2.19 **requestor**

3.2.20 **response (primitive);**
 acceptor.submit (primitive)

3.2.21 **submit (primitive)**

3.2.22 **symmetrical service**

3.3 Data-link service terms and definitions

3.3.1

bridge

DL-router

DL-relay entity which performs selective store-and-forward and routing functions

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

and also provides time synchronization among the links to which it is forwarding

3.3.2

DLCEP-address

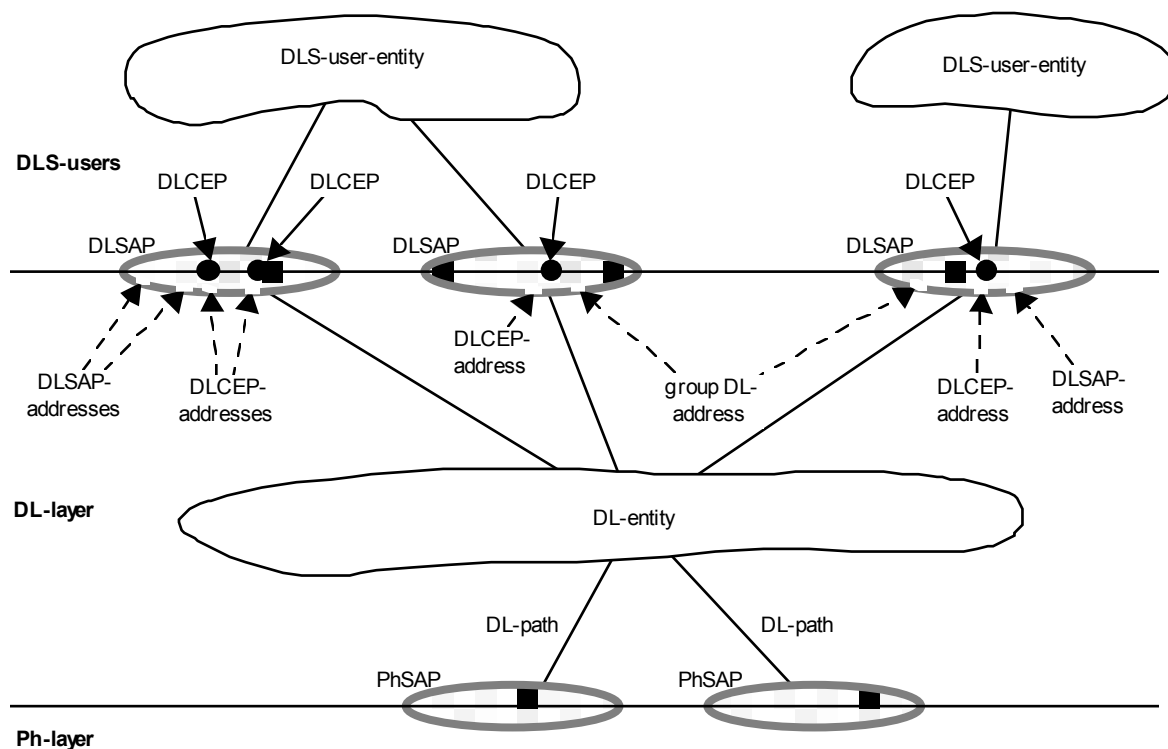
DL-address which designates either

- a) one peer DL-connection-end-point; or

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

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

Note 1 to entry: This is an extension of the use of DL-addresses beyond that specified in ISO/IEC 7498-3 (see Figure 1).



NOTE 1 DLSAPs and 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. A DLCEP-address also designates a specific point of information flow (its DLCEP) within the DLSAP.

NOTE 3 A single DL-entity can have multiple DLSAP-addresses and group DL-addresses associated with a single DLSAP.

NOTE 4 Figure 1 also shows the relationships of DL-paths and PhSAPs.

Figure 1 – Relationships of DLSAPs, DLSAP-addresses, DLCEPs, DLCEP-addresses, DLSEP-addresses and group DL-addresses

3.3.3

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

3.3.4

DLSAP

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

Note 1 to entry: This definition, derived from ISO/IEC 7498-1, is repeated here to facilitate understanding of the critical distinction between DLSAPs and their DL-addresses.

3.3.5**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.

3.3.6**(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.

3.3.7**DLSEP-address**

DL-address which designates a DL-scheduling-end-point within a DLE

Note 1 to entry: This is an extension of the use of DL-addresses beyond that specified in ISO/IEC 7498-3. (See Figure 1.)

3.3.8**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 can be composed of just a single link.

3.3.9**frame**

denigrated synonym for DLPDU

3.3.10**group DL-address**

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

Note 1 to entry: A single DL-entity can have multiple group DL-addresses associated with a single DLSAP. A single DL-entity also can have a single group DL-address associated with more than one DLSAP.

3.3.11**initiator**

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 1 to entry: Some prior national standards have referred to this role as a "master" role.

3.3.12**multi-peer DLC**

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)

Note 1 to entry: 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. In this last case, the characterizations as publisher and subscriber are misnomers.

Note 2 to entry: 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.

3.3.13

node

single DL-entity as it appears on one local link

3.3.14

peer DLC

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

Note 1 to entry: 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.

3.3.15

receiving DLS-user

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

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

3.3.16

responder

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 1 to entry: Some prior national standards have referred to this role as a "slave" role.

3.3.17

sending DLS-user

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

3.3.18

timeliness

DL-timeliness

attribute of a datum which provides an assessment of the temporal currency of that datum, 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

Note 1 to entry: As a general rule, timeliness is a user attribute which can be affected negatively by the various layers of the data transport system. That is, a datum which was timely when the requesting user presented it to a data communications subsystem for transmission may become untimely due to delays in the communications subsystem.

Note 2 to entry: DL-timeliness is an attribute of a DLS-user datum relating the timing of a DLS-user/DLE interaction which writes or reads that datum to one or more other (earlier) DLS-user/DLE interactions.

Note 3 to entry: These concepts also support migration from previous national standards.

3.3.19

transaction

single DLPDU, or a sequence of two immediately consecutive related DLPDUs, resulting from a single DLS-user request

Note 1 to entry: The DLE sending the first DLPDU of the transaction is known as the initiator; the DLE which sends the second DLPDU of the transaction, if any, is known as the responder.

Note 2 to entry: A DL-entity can be both an initiator and a responder in the same transaction.

3.4 Common symbols and abbreviations

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

3.4.1	DL-	Data-link layer (as a prefix)
3.4.2	DLC	DL-connection
3.4.3	DLCEP	DL-connection-end-point
3.4.4	DLE	DL-entity (the local active instance of the data-link layer)
3.4.5	DLL	DL-layer
3.4.6	DLPCI	DL-protocol-control-information
3.4.7	DLPDU	DL-protocol-data-unit
3.4.8	DLM	DL-management
3.4.9	DLME	DL-management Entity (the local active instance of DL-management)
3.4.10	DLMS	DL-management service
3.4.11	DLS	DL-service
3.4.12	DLSAP	DL-service-access-point
3.4.13	DLSEP	DL-schedule-end-point
3.4.14	DLSDU	DL-service-data-unit
3.4.15	FIFO	First-in first-out (queuing method)
3.4.16	OSI	Open systems interconnection
3.4.17	Ph-	Physical layer (as a prefix)
3.4.18	PhE	Ph-entity (the local active instance of the Physical layer)
3.4.19	PhL	Ph-layer
3.4.20	QoS	Quality-of-service

3.5 Common conventions

3.5.1 Basic conventions

This standard uses the descriptive conventions given 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.

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

This standard uses a tabular format to describe the component parameters of the DLS primitives. The parameters that apply to each group of DLS primitives are set out in tables throughout the remainder of this standard. 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 (see 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 which illustrate these interfaces, dashed lines indicate cause-and-effect or time-sequence relationships, and wavy lines indicate that events are roughly contemporaneous.

3.5.2 Identifiers

Many 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 standard is a purely local issue. Nevertheless, these identifiers are specified explicitly in these primitives to provide a descriptive means

- a) of cancelling (aborting) an outstanding request primitive before receiving its corresponding confirm primitive;
- b) for referring within a request or response primitive to persistent DL-objects, such as buffers and queues, which were created as the result of a previous DLS primitive; and

- c) for referring within an indication or confirm primitive to persistent DL-objects which 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 b); while DL-identifiers are required for c).

4 Overview of the data-link layer service

4.1 General

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 1 A DLS-user may segment arbitrary-length data into limited-length DLSDUs before making DLS 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 2 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 3 The DLS is required to differentiate between the individual systems that are physically or logically connected to a multipoint 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/IEC 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 which becomes empty after being read), where each queue item or buffer can hold a single DLSDU.

4.1.1 Overview of DL-subnetwork structuring

A DL-subnetwork consists of a set of DL-segments (links) interconnected by DL-relay entities (bridges) which provide DL-layer-internal synchronization, 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*.

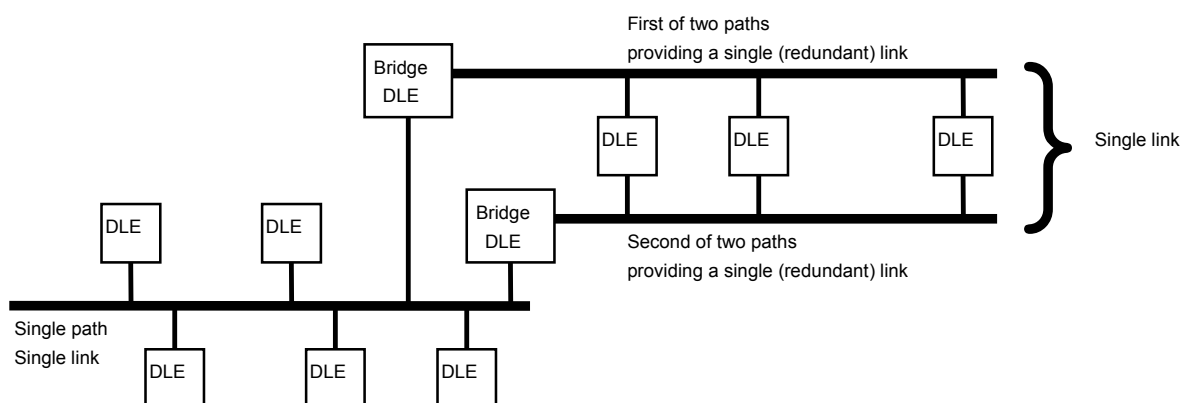


Figure 2 – Example of paths, links, bridges, and the extended link

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 2.

NOTE 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/IEC 7498-1).

NOTE 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.

4.1.2 Overview of DL-naming (addressing)

DL-names, known conventionally as DL-addresses, are identifiers from a defined identifier space — the DL-address-space — which serve to name objects within the scope of the data-link layer. Examples of such objects are data-link layer-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 subspaces 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.

4.1.2.1 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.

- 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.

4.1.2.2 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.

4.1.2.3 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.

4.2 Types and classes of data-link layer service

There are four types of DLS:

- a) a DL(SAP)-address, queue and buffer management service (defined in Clause 5);
- b) a connection-mode data transfer service with four classes of service (defined in Clause 6);
- c) a connectionless-mode data transfer service with three classes of service (defined in Clause 7); and
- d) a time and transaction scheduling service (defined in Clause 8).

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 clause which describes a specific type of DLS.

A DL-management service (DLMS) is defined in Clause 9.

4.3 Quality-of-service (QoS) attributes common to multiple types of data-link layer 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 DLS 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 which 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 DLS. Some implementations of this DLS may provide additional levels of default QoS beyond those specified in this standard.

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 DLS 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 DLS, but dynamic for all DLCEP-establishment requests and responses, where its value serves merely as a default for each appropriate DLS 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.

4.3.1 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 should support three DLL priority levels, each of which should 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

NOTE 1 URGENT and NORMAL are considered *time-critical* priority levels; TIME-AVAILABLE is considered a *non-time-critical* priority level.

NOTE 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.

4.3.2 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 which specifies *UNLIMITED*.

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

The intervals specified are the maximum permissible delays

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

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

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

4.3.3 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 1 This has a slight impact on the residual rate of DLPDU misdelivery; more addressing information reduces the potential for misdelivery.

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

NOTE 2 This continuing background transmission is known as *residual activity*.

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

NOTE 3 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 should include the minimum permitted amount of addressing information;
- 2) SOURCE — each DLPDU should include a source DL-address where possible;
- 3) MAXIMAL — each DL-address should 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.

4.3.4 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 DLS 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 cancelled 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 1 DLC support services such as DL-CONNECT, DL-RESET and DL-DISCONNECT, and intra-DLS-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 2 The choice IMPLICIT is incompatible with a DLCEP which 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.

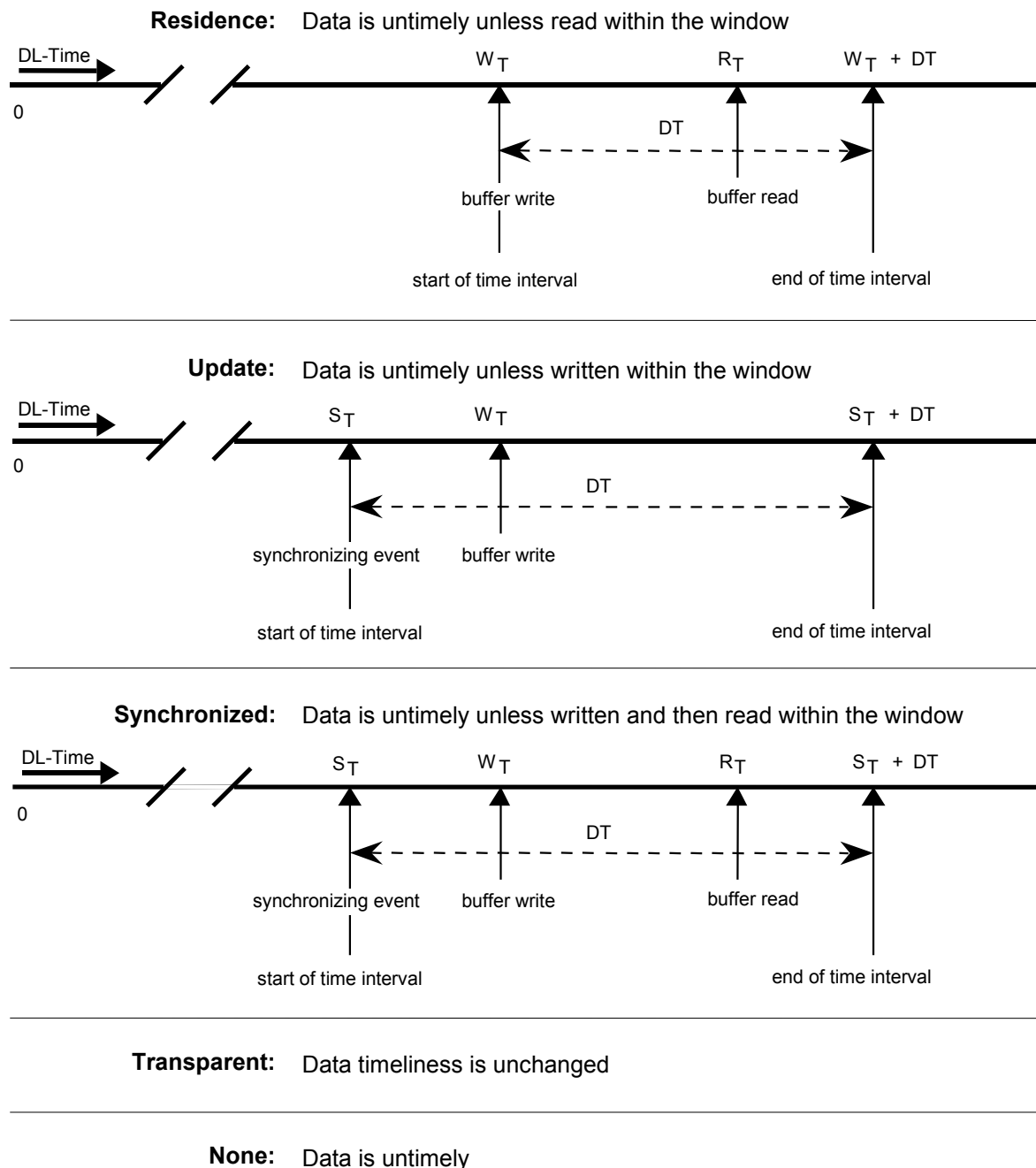
- b) EXPLICIT — any required data or unitdata communications with peer DLS-user(s) from this DLSAP-address, or from this DLCEP, will occur only when the deferral is explicitly cancelled by an involved DLS-user.

NOTE 3 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.

4.3.5 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 which transfer DLS-user data to or from DL-buffers at such DLCEPs.



Legend

- DL-event: DLS-user/DLE interaction, DL-buffer read, DL-buffer write, DL-timeout
 window size interval DT

Figure 3 – Types of DL-timeliness
In terms of elapsed DL-time and events at the assessing DLCEP

Each DLCEP establishment request, and each response, can specify DL-timeliness criteria which 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 3:

- 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);

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

NOTE 1 This type of timeliness was called *asynchronous* in prior national 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);

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

NOTE 2 A type of timeliness closely related to this one was called *punctual* in prior national 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);

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

NOTE 3 This type of timeliness was called *synchronous* in prior national 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 is 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 is FALSE.

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

NOTE 4 DL-time is described in Clause 8.

5 DL(SAP)-address, queue and buffer management data-link layer service

5.1 Facilities of the DL(SAP)-address, queue and buffer management data-link layer service

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

- a) a means for creating and deleting a retentive buffer, or 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.

5.2 Model of the DL(SAP)-address, queue and buffer management data-link layer service

This standard uses the abstract model for a layer service defined in ISO/IEC 10731:1994, Clause 5. 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 DLS 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 of ISO/IEC 10731. Therefore the DL(SAP)-address, queue and buffer management services are provided by request (requestor.submit) primitives with input and output parameters.

5.3 Sequence of primitives at one DLSAP

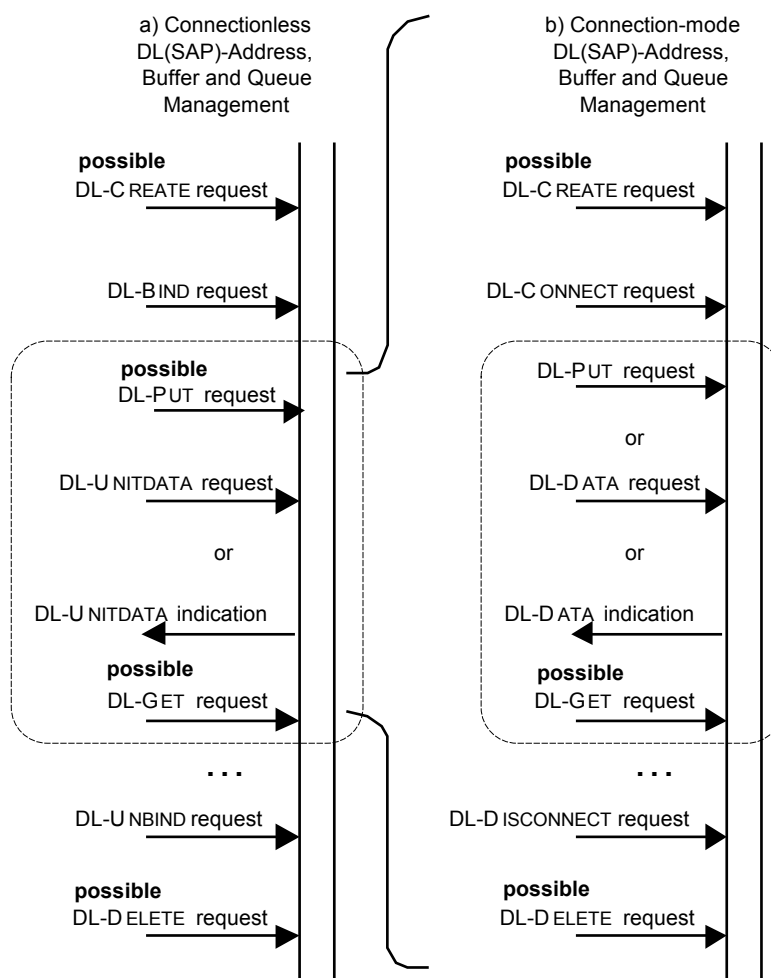
5.3.1 Constraints on sequence of primitives

Subclause 5.3.1 defines the constraints on the sequence in which the primitives defined in 5.4 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 4.

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.		



NOTE 1 Primitives within the outlined areas can be repeated many times between instances of the primitives in the enclosing areas.

NOTE 2 The entire right-hand part of the figure is another alternative to the outlined 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 enclosing area.

NOTE 3 DL-PUT and DL-GET request primitives both can be used earlier and later than shown.

Figure 4 – Sequence of primitives for the DL(SAP)-address, queue and buffer management DLS

5.4 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 5). 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.

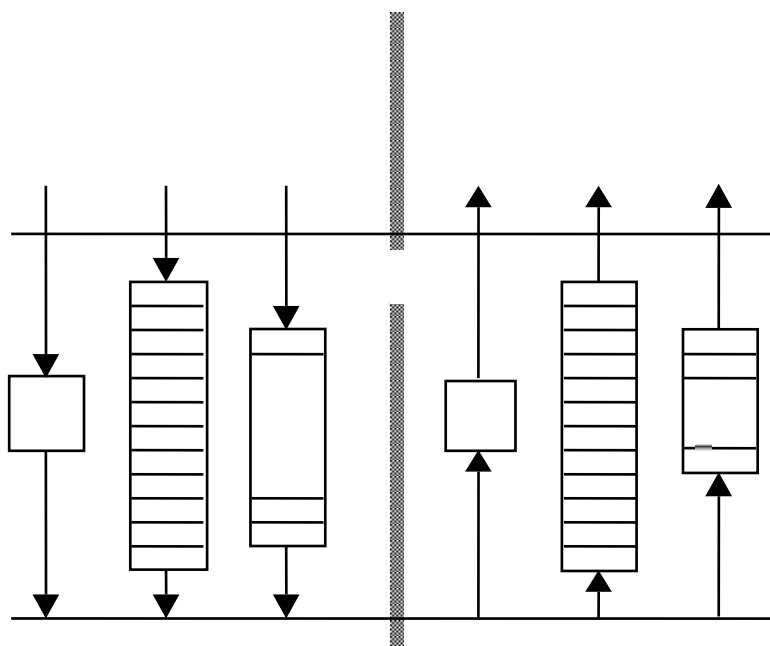


Figure 5 – Supported methods of data management for transmission and delivery

5.4.1 Create

5.4.1.1 Function

The create buffer or queue DLS 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.

5.4.1.2 Types of parameters

Table 2 indicates the primitive and parameters of the create buffer or queue DLS.

Table 2 – DL-buffer-and-queue-management create primitive and parameters

DL-CREATE Parameter name	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

5.4.1.2.1 Buffer-or-queue DLS-user-identifier

This parameter specifies a means of referring to the buffer or queue in output parameters of other local DLS primitives which 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.

5.4.1.2.2 Queuing policy

This 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 which 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.

NOTE 1 Buffer and queue bindings result from DL-BIND request, DL-CONNECT request and DL-CONNECT response primitives.

NOTE 2 An explicit queue needs to 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.

NOTE 3 A retentive buffer (BUFFER-R) needs to 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.

NOTE 4 A non-retentive buffer (BUFFER-NR) needs to 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.

NOTE 5 Buffers can be overwritten at any time, with complete loss of the prior contents except to those users which 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.

5.4.1.2.2.1 Maximum queue depth

This 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 should include the values one (1), two (2), three (3), and four (4).

NOTE Implementations of this DLS 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.

5.4.1.2.3 Maximum DLSDU size

This 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.

5.4.1.2.4 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is 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 standard.

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-NR 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.

5.4.1.2.5 Buffer-or-queue DL-identifier

This 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 which 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.

5.4.1.3 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 4.

5.4.2 Delete

5.4.2.1 Function

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

NOTE 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.

NOTE 2 This facility may also be provided by local DL-management actions, which are beyond the scope of this standard.

5.4.2.2 Types of parameters

Table 3 indicates the primitive and parameters of the delete buffer or queue DLS.

Table 3 – DL-buffer-and-queue-management delete primitive and parameters

DL-DELETE Parameter name	Request	
	input	output
Buffer-or-queue DL-identifier	M	
Status		M

5.4.2.2.1 Buffer-or-queue DL-identifier

This 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.

5.4.2.2.2 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is 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 standard.

5.4.2.3 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 4.

5.4.3 Bind

5.4.3.1 Function

The bind DL(SAP)-address DLS 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.

5.4.3.2 Types of parameters

Table 4 indicates the primitive and parameters of the bind DL(SAP)-address DLS.

Table 4 – DL(SAP)-address-management bind primitive and parameters

Parameter name	DL-BIND	Request	
		input	output
DL(SAP)-address DLS-user-identifier	M		
DL(SAP)-address	M		
DL(SAP)-role	M		
Indicate-null-UNITDATA-EXCHANGE-transactions	C		
Remote-DLSAP-address	C		
Receiving-buffer-or-queue-bindings			
URGENT-buffer-or-queue DL-identifier	U		
NORMAL-buffer-or-queue DL-identifier	U		
TIME-AVAILABLE-buffer-or-queue DL-identifier	U		
Sending-buffer-or-queue-bindings			
URGENT-buffer-or-queue DL-identifier	CU		
NORMAL-buffer-or-queue DL-identifier	CU		
TIME-AVAILABLE-buffer-or-queue DL-identifier	CU		
Default QoS as sender			
DLL priority	CU		
DLL maximum confirm delay			
on DL-CONNECT, DL-RESET, DL-SUBSCRIBER-QUERY	CU		
on DL-DATA	CU		
on locally-confirmed DL-UNITDATA	CU		
on remotely-confirmed DL-UNITDATA, DL-LISTENERQUERY, DL-UNITDATA-EXCHANGE	CU		
DLPDU authentication	CU		
DL-scheduling-policy	CU		
Status			M
DL(SAP)-address DL-identifier			C

5.4.3.2.1 DL(SAP)-address DLS-user-identifier

This parameter specifies a means of referring to the DL(SAP)-address in output parameters of other local DLS primitives which convey the name of the DL(SAP)-address from the local DLE to the local DLS-user.

The naming-domain of the DL(SAP)-address DLS-user-identifier is the DLS-user's local-view.

5.4.3.2.2 DL(SAP)-address

This parameter specifies an individual local DLSAP-address or group DL-address to be associated with the invoking (necessarily local) DLSAP.

5.4.3.2.3 DL(SAP)-role

This parameter constrains, as specified in Table 5, the DL-connectionless DL-UNITDATA and DL-UNITDATA-EXCHANGE service primitives that can be issued with this DL(SAP)-address at the local DLSAP (to which this DL(SAP)-address is being bound). It also constrains whether the DL(SAP)-address may have an associated DLCEP and the permitted types of bindings (implicit queue, explicit queue or explicit buffer) to the DL(SAP)-address. Permitted values for this parameter are specified in Table 5.

Table 5 – DL(SAP)-role constraints on DLSAPs, DLCEPs and other DLS Primitives

DL(SAP)-role	Sending DLSAP-Address bindings	Receiving DL(SAP)-Address bindings	DLCEP permitted	DL-Unitdata		Unitdata-Exchange
				Request permitted	Indication possible	Indication possible
basic	IQ, EQ	IQ, EQ	yes	yes	yes	no
group	—	IQ, EQ	no	no	yes	No
initiator	EB	EB, EQ	no	no	no	yes
constrained responder	EB	EB, EQ	no	no	no	yes
unconstrained responder	EB	EB, EQ	no	no	no	yes
Key						
— : not applicable		IQ : implicit queue on transmit, immediate (as soon as possible) delivery on receipt				
EQ : explicit queue		EB: explicit retentive or non-retentive buffer				

The default value for this parameter is BASIC.

NOTE The roles of INITIATOR, CONSTRAINED RESPONDER, and UNCONSTRAINED RESPONDER support migration from previous national standards.

5.4.3.2.3.1 Indicate-null-unitdata-exchange-transactions

This Boolean parameter is present when the DL(SAP)-role parameter has the value INITIATOR, CONSTRAINED RESPONDER, or UNCONSTRAINED RESPONDER. When present, the indicate-null-UNITDATA-EXCHANGE-transactions parameter specifies whether an instance of a UNITDATA-EXCHANGE transaction which occurs at this DLSAP-address generates a DL-UNITDATA-EXCHANGE indication even when no DLS-user data transfer (in either direction) occurred.

NOTE 1 Such an indication would attest to the DLS-user that communication with the DLE of the addressed remote peer DLS-user was still possible. In other DL-protocols in which all DLS-users are on the same unbridged local link, this attestation is sometimes provided by a "live list".

NOTE 2 UNITDATA-EXCHANGE and the use of the indicate-null-UNITDATA-EXCHANGE-transactions parameter are covered in Clause 7.

5.4.3.2.3.2 Remote-DLSAP-address

This parameter is present when the DL(SAP)-role parameter has the value CONSTRAINED RESPONDER. When present, this parameter specifies an individual DLSAP-address, specifying that the DL-UNITDATA-EXCHANGE service may be initiated only from the specified DLSAP-address.

5.4.3.2.4 Receiving buffer-or-queue bindings

When present, each buffer-or-queue DL-identifier parameter specifies the local DL-identifier returned by a successful prior DL-CREATE request primitive (or DL-management action) that created a buffer or queue, which has not yet been deleted.

When the DL(SAP)-role parameter has the value BASIC or GROUP, then

- a) explicit bindings to a buffer are not permitted;
- b) explicit bindings to a queue are permitted; and
- c) if no binding at a given DLL-priority exists, then the DLSAP-address is implicitly bound at that priority to the default OSI delivery service, which is immediate (as soon as possible) delivery.

When bound as in b), the maximum-DLSDU-size for each specified receive queue should accommodate the maximum amount of DLS-user data permitted within a single DLPDU of the priority corresponding to the binding, as specified in 4.3.1.

When the DL(SAP)-role parameter has the value INITIATOR, CONSTRAINED RESPONDER, or UNCONSTRAINED RESPONDER, then

- 1) explicit bindings to a buffer are permitted;
- 2) explicit bindings to a queue are permitted; and
- 3) if no binding at a given DLL-priority exists, then it is not possible to receive a DLSDU of that priority at that DLSAP-address.

NOTE If a queue is bound to the receiving (DLS-provider to DLS-user) direction of data transfer at a DL(SAP)-address at a specified priority, as in b) or e), then the DLS-user has specified the maximum number of queued received DLSUDs, and can choose when to process those DLSUDs.

5.4.3.2.4.1 Urgent buffer-or-queue DL-identifier

When permitted, specifying a buffer-or-queue DL-identifier results in the identified buffer or queue being bound to the DLSAP for use in reception at URGENT priority.

5.4.3.2.4.2 Normal buffer-or-queue DL-identifier

When permitted, specifying a buffer-or-queue DL-identifier results in the identified buffer or queue being bound to the DLSAP for use in reception at NORMAL priority.

5.4.3.2.4.3 Time-available buffer-or-queue DL-identifier

When permitted, specifying a buffer-or-queue DL-identifier results in the identified buffer or queue being bound to the DLSAP for use in reception at TIME-AVAILABLE priority.

5.4.3.2.5 Sending buffer-or-queue bindings

When present, each buffer-or-queue DL-identifier parameter specifies the local DL-identifier returned by a successful prior DL-CREATE request primitive (or DL-management action) that created a buffer or queue that has not yet been deleted.

When the DL(SAP)-role parameter has the value BASIC, then

- a) explicit bindings to a buffer are not permitted;
- b) explicit bindings to a queue are permitted; and
- c) if no binding at a given DLL-priority exists, then the DLSAP-address is implicitly bound at that priority to a default OSI queue.

NOTE If a queue is bound to the sending (DLS-user to DLS-provider) direction of data transfer at a DLSAP-address at a specified priority, as in b) or c), then transmission of the queued DLSUDs in FIFO order will be attempted, as appropriate, when the queue is non-empty.

When the DL(SAP)-role parameter has the value INITIATOR, CONSTRAINED RESPONDER, or UNCONSTRAINED RESPONDER, then

- 1) explicit bindings to a buffer are permitted;

- 2) explicit or implicit bindings to a queue are not permitted; and
- 3) if no binding at a given DLL-priority exists, then it is not possible to source a DLSDU at that priority from that DLSAP-address.

When the DL(SAP)-role parameter has the value GROUP, then no bindings are permitted or implied; it is not possible to attribute a group DL-address as the source of a DLSDU.

5.4.3.2.5.1 Urgent buffer-or-queue DL-identifier

When permitted, specifying a buffer-or-queue DL-identifier results in the identified buffer or queue being bound to the DLSAP for use in transmission at URGENT priority.

5.4.3.2.5.2 Normal buffer-or-queue DL-identifier

When permitted, specifying a buffer-or-queue DL-identifier results in the identified buffer or queue being bound to the DLSAP for use in transmission at NORMAL priority.

5.4.3.2.5.3 Time-available buffer-or-queue DL-identifier

When permitted, specifying a buffer-or-queue DL-identifier results in the identified buffer or queue being bound to the DLSAP for use in transmission at TIME-AVAILABLE priority.

5.4.3.2.6 Default QoS as sender

The DLS-user may specify default values for some of the QoS parameters that apply to connection-mode and connectionless data transmission, as described in 4.3. These default values will be used whenever data or unitdata transmission services are initiated at this DLSAP-address, unless explicitly overridden during an actual service invocation.

When the DL(SAP)-role parameter has the value INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER, some of these QoS attributes as sender are irrelevant and should be absent. When the DL(SAP)-role parameter has the value GROUP, all of these QoS attributes as sender are irrelevant and should be absent.

5.4.3.2.6.1 DLL priority

This QoS attribute is not relevant when the DL(SAP)-role parameter has the value INITIATOR, CONSTRAINED RESPONDER, UNCONSTRAINED RESPONDER, or GROUP, and thus should be absent.

5.4.3.2.6.2 DLL maximum confirm delay

This QoS attribute is not relevant when the DL(SAP)-role parameter has the value CONSTRAINED RESPONDER, UNCONSTRAINED RESPONDER, or GROUP, and thus should be absent.

5.4.3.2.6.3 DLPDU authentication

This QoS attribute is not relevant when the DL(SAP)-role parameter has the value GROUP, and thus should be absent.

5.4.3.2.6.4 DL-scheduling-policy

This QoS attribute is not relevant when the DL(SAP)-role parameter has the value INITIATOR, CONSTRAINED RESPONDER, UNCONSTRAINED RESPONDER, or GROUP, and thus should be absent.

5.4.3.2.7 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

- a) “success”;
- b) “failure — insufficient resources”;
- c) “failure — DL(SAP)-address invalid or unavailable”;
- d) “failure — DL(SAP)-role not supported”;
- e) “failure — remote DL(SAP)-address invalid”;
- f) “failure — invalid buffer or queue binding”;
- g) “failure — parameter inconsistent with DL(SAP)-role”;
- h) “failure — parameter violates management constraint”;
- i) “failure — number of requests violates management constraint”; or
- j) “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 standard.

5.4.3.2.8 DL(SAP)-address DL-identifier

The DL(SAP)-address DL-identifier parameter is present when the status parameter indicates that the DL-BIND request primitive was successful. The DL(SAP)-address DL-identifier parameter gives the local DLS-user a means of referring to the DL(SAP)-address in input parameters of other local DLS primitives which convey the name of the DL(SAP)-address from the local DLS-user to the local DLE.

The naming-domain of the DL(SAP)-address DL-identifier is the DL-local-view.

5.4.3.3 Sequence of primitives

The sequence of primitives in

- a) binding a DL(SAP)-address to the invoking DLSAP, and optionally binding one or more buffers or queues to the DL(SAP)-address; and
- b) later unbinding the DL(SAP)-address and buffers or queues from the DLSAP

is defined in the time sequence diagram of Figure 4.

5.4.4 Unbind

5.4.4.1 Function

The unbind DL(SAP)-address DLS primitive is used to unbind a DL(SAP)-address from the invoking DLSAP. Any buffers or queues that were explicitly bound to the DL(SAP)-address are also unbound from that DL(SAP)-address at the same time.

NOTE 1 This primitive can only be used to unbind a DL(SAP)-address that was bound to the DLSAP by a prior DL-BIND request primitive. It cannot be used to unbind a DL(SAP)-address that was bound to the DLSAP by prior local DL-management actions.

NOTE 2 This facility may also be provided by local DL-management actions, which are beyond the scope of this standard.

5.4.4.2 Types of parameters

Table 6 indicates the primitive and parameters of the unbind DL(SAP)-address DLS.

Table 6 – DL(SAP)-address-management unbind primitive and parameters

DL-UNBIND Parameter name	Request
	input
DL(SAP)-address DL-identifier	M

5.4.4.2.1 DL(SAP)-address DL-identifier

This parameter specifies the local DL-identifier returned by a successful prior DL-BIND request primitive. The DLS-provider should unbind the local DL-identifier and its associated DL(SAP)-address, and any associated buffers or queues, from the invoking DLSAP, after first disconnecting all DLCEPs, unbinding all buffer and queues which were bound to those DLCEPs, and terminating all outstanding DL-UNITDATA requests associated with that DLSAP-address.

5.4.4.3 Sequence of primitives

The sequence of primitives in

- a) binding a DL(SAP)-address, and possibly buffers or queues, with the invoking DLSAP; and
- b) later unbinding the DL(SAP)-address and those buffers or queues from the DLSAP

is defined in the time sequence diagram of Figure 4.

5.4.5 Put**5.4.5.1 Function**

The put buffer DLS primitive is used to copy a DLSDU to a buffer. In some cases it may also be used to set the buffer empty.

5.4.5.2 Types of parameters

Table 7 indicates the primitive and parameters of the put buffer DLS.

Table 7 – DL-buffer-management put primitive and parameters

DL-PUT Parameter name	Request	
	input	output
Buffer DL-identifier	M	
DLS-user-data	U	
DLS-user-data-timeliness	U	
Status		M

5.4.5.2.1 Buffer DL-identifier

This parameter specifies the local DL-identifier returned by a successful prior DL-CREATE request primitive which created a buffer (or by DL-management).

5.4.5.2.2 DLS-user-data

When present, this parameter specifies one or more octets of DLS-user-data, up to the maximum DLSDU size specified in the associated DL-CREATE request primitive, possibly further constrained by DLC negotiation. The DLE may also note the current DL-time for later reporting as the time-of-production.

When DLS-user-data is not present, then

- a) If the buffer is bound to a DLCEP-address, then the DL-Put request fails and a status of “failure — invalid DLSDU size” is returned to the requesting DLS-user.
- b) Otherwise, when a) does not apply, then the DL-Put request primitive resets the buffer to its initial empty state.

5.4.5.2.3 DLS-user-data-timeliness

This parameter specifies whether the associated DLS-user-data meets the requesting DLS-user's timeliness criteria, or not. Its value is either TRUE (one or more criteria exist, and all were met) or FALSE (either no criteria exist, or one or more of the criteria were not met).

If the data in this buffer is then sent to other DLS-users through a DLC, then this timeliness attribute will be logically ANDed with the assessment(s) of any DLCEP-evaluated timeliness criteria, and the result associated with the receiving buffer, if any.

NOTE Buffer timeliness is presented to the DLS-user(s) by the DL-GET primitive.

The default value for this parameter is FALSE.

5.4.5.2.4 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

- a) “success”;
- b) “failure — invalid DLSDU size”; or
- c) “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 standard.

5.4.5.3 Sequence of primitives

The sequence of primitives in using a buffer is defined in the time sequence diagram of Figure 4.

5.4.6 Get

5.4.6.1 Function

The get buffer or queue DLS primitive is used to read a DLSDU from a buffer, or attempt to remove a DLSDU from a FIFO queue.

When the DL-GET request primitive specifies a buffer (see 5.4.5.2.1, 6.3.2.9.1a), and 7.5.2.2.3) and the buffer is empty, then the DL-GET request primitive returns an appropriate failure status. Otherwise, the DL-GET request primitive copies the DLSDU from the buffer and returns it.

When the DL-GET request primitive specifies an explicit queue and the queue is empty, then the DL-GET request primitive returns an appropriate failure status. Otherwise, the DL-GET request primitive dequeues the DLSDU from the FIFO queue and returns it, together with associated DL(SAP)-addresses or an associated DLCEP DL-identifier, if appropriate.

5.4.6.2 Types of parameters

Table 8 indicates the primitive and parameters of the get buffer or queue DLS.

Table 8 – DL-buffer-and-queue-management get primitive and parameters

Parameter name	DL-GET	Request	
		input	output
Buffer-or-queue DL-identifier		M	
Status			M
Reported-service-identification-class			C
Reported-service-identification			
Receiving DLCEP DLS-user-identifier			C
Called DL(SAP)-address DLS-user-identifier			C
Calling DLSAP-address			C
DLL-priority			C
DLS-user-data			C
DLS-user-data-timeliness			
Local DLE timeliness			C
Sender and remote DLE timeliness			C
Time-of-production			C

5.4.6.2.1 Buffer-or-queue DL-identifier

This parameter specifies the local DL-identifier returned by a successful prior DL-CREATE request primitive (or by DL-management).

5.4.6.2.2 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. Attempting to copy a DLSDU from a buffer that is empty, or to remove a DLSDU from a FIFO queue that is empty, will result in failure. The value conveyed in this parameter is as follows:

- a) “success”;
- b) “possible failure — empty buffer”;
- c) “failure — empty queue”; or
- d) “failure — reason unspecified”.

NOTE 1 An empty buffer can be considered to contain a null DLSDU when used with the unitdata exchange service, and so may be treated as “success” by the DLS-user.

NOTE 2 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 standard.

5.4.6.2.3 Reported-service-identification-class

This parameter is present when the status parameter indicates that the DL-GET request was successful. When present, this parameter specifies the structure of the associated reported-service-identification parameter. The values of this parameter are:

- a) none — implying that a buffer is being retrieved and that source information is not present;
- b) DLCEP — the receiving DLCEP DL-identifier is present; or
- c) DL(SAP)-address — the receiving DL(SAP)-address DL-identifier, the sending DLSAP-address, and the priority of the DLSDU are present.

5.4.6.2.4 Reported-service-identification

This compound parameter is present when the status parameter indicates that the DL-GET request was successful. When present, this parameter identifies the service endpoint(s) and priority of the reported DLSDU.

5.4.6.2.4.1 Receiving-DLCEP DLS-user-identifier

This parameter is present when the reported-service-identification-class parameter specifies DLCEP. When present, this parameter identifies the DLCEP at which the associated DLSDU was received.

5.4.6.2.4.2 Called-DL(SAP)-address DLS-user-identifier

This parameter is present when the reported-service-identification-class parameter specifies DL(SAP)-ADDRESS. When present, this parameter identifies the destination DL(SAP)-address at which the associated DLSDU was received.

5.4.6.2.4.3 Calling-DLSAP-address

This parameter is present when the reported-service-identification-class parameter specifies DL(SAP)-ADDRESS. When present, this parameter identifies the source DLSAP-address from which the associated DLSDU was sent.

NOTE If the DLS-user has issued a DL-BIND request for the calling-DLSAP-address, then this parameter also can take the form of a DLSAP-address DLS-user-identifier.

5.4.6.2.4.4 DLL-priority

This parameter is present when the reported-service-identification-class parameter specifies DL(SAP)-ADDRESS. When present, this parameter identifies the sender's DLL priority of the received DLSDU.

5.4.6.2.5 DLS-user-data

This parameter is present when the status parameter indicates that the DL-GET request was successful. This parameter returns a single DLSDU consisting of one or more octets of DLS-user-data, up to the maximum DLSDU size specified in the associated DL-CREATE request primitive (or by DL-management).

5.4.6.2.6 DLS-user-data-timeliness

This parameter is present when the status parameter indicates that the DL-GET request was successful and the buffer-or-queue DL-identifier parameter specifies a buffer. When present, this parameter specifies up to three aspects of DLS-user-data timeliness.

5.4.6.2.6.1 Local-DLE-timeliness

This parameter specifies whether the associated DLS-user-data met the receiving DLCEP's timeliness criteria, or not. Its value is either TRUE (timeliness criteria exist, and all were met) or FALSE (either no timeliness criteria exist, or one or more of the criteria were not met). When the buffer is not bound as a sink to a DLCEP, but is instead written by a local DL-PUT request, then the value of this parameter is always TRUE. When the buffer is bound as a sink to a DLSAP-address whose DL(SAP)-role is INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER, then the value of this parameter is always FALSE.

5.4.6.2.6.2 Sender-and-remote-DLE-timeliness

This parameter specifies whether the associated DLS-user-data met both the sending DLS-user's timeliness criteria, and any sending and intermediary receiving DLCEPs' timeliness criteria, or not. Its value is either TRUE (so specified by the sending DLS-user and timeliness

criteria exist for each of the transporting DLCEPs, and all were met) or FALSE (either so specified by the sending DLS-user, or timeliness was not selected for one or more of the transporting DLCEP(s), or one or more of the timeliness criteria were not met).

NOTE DL-timeliness is partitioned into separate receiver timeliness and sender timeliness:

- a) to distinguish between those aspects of timeliness whose criteria are specified by the receiving DLS-user and those which are not;
- b) to provide assistance in DLS-user diagnosis of the source of non-timeliness — either the DLS-user-data source and remote portions of the distributed DLS-provider, or the local DLE and the receiving DLS-user; and
- c) to facilitate migration of existing national standards.

5.4.6.2.6.3 Time-of-production

This parameter is present only when

- a) the sending DLCEP specified TRUE for its time-of-production QoS parameter (see 6.3.2.10.1.4); and
- b) the value of the sender-and-remote-DLE-timeliness parameter (see 5.4.6.2.6.2) returned by this DL-GET primitive is TRUE.

This parameter specifies the DL-time at which a DLS-user transferred the associated DLS-user-data to the (distributed) DLS-provider.

NOTE 1 For a buffer bound as a source at a DLCEP, this is the DL-time at which the sending DLS-user issued the DL-PUT request which placed the associated DLS-user-data in the sending DL-buffer.

NOTE 2 If a DL-relay entity (a bridge) acts as a distributor by connecting a receiving DLCEP to a second explicitly-scheduled sending DLCEP through a shared intermediate buffer, then at the final receiver the time-of-production is still the time at which the sending DLS-user (at the first DLCEP) issued the DL-PUT request which placed the associated DLS-user-data in the original sending DL-buffer.

5.4.6.3 Sequence of primitives

The sequence of primitives in using a buffer or queue is defined in the time sequence diagram of Figure 4.

6 Connection-mode data-link layer service

6.1 Facilities of the connection-mode data-link layer service

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

- a) A means to establish (see Figure 6) either
 - 1) a peer DLC between two DLS-users for exchanging DLSDUs between the two DLS-users, or

NOTE 1 It is also possible for a peer DLC to be negotiated to provide just DL-simplex transmission from one of the DLS-users to the other.
 - 2) a multi-peer DLC between a single publishing DLS-user and a set of subscribing DLS-users for sending DLSDUs
 - i) from the publishing DLS-user to the set of subscribing DLS-users, and
 - ii) optionally, from any of the subscribing DLS-users to the publishing DLS-user.

NOTE 2 It is also possible for a multi-peer DLC to be negotiated to provide just DL-simplex transmission from the publishing DLS-user to the set of subscribing DLS-users, or from the set of subscribing DLS-users to the publishing DLS-user.

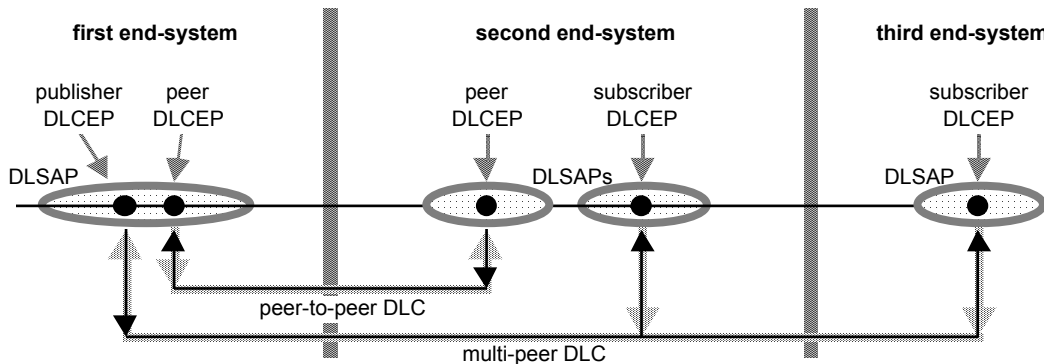


Figure 6 – Peer-to-peer and multi-peer DLCs and their DLCEPs

- b) A means of establishing an agreement for a certain quality-of-service (QoS) associated with a DLC, between the initiating DLS-user, the responding DLS-user(s), and the distributed DLS-provider.
- c) A means of transferring DLSDUs of limited length on a DLC. The transfer of DLSDUs is transparent, in that the boundaries of DLSDUs and the contents of DLSDUs are preserved unchanged by the DLS, and there are no constraints on the DLSDU content (other than limited length) imposed by the DLS.

NOTE 3 The length of a DLSDU is limited because of internal mechanisms employed by the DL-protocol (see ISO/IEC 7498-1).

- d) A means of conveying timeliness information about those DLSDUs and their conveyance by the (distributed) DLS-provider in certain modes of DLC operation.
- e) A means by which the receiving DLS-user at a peer DLCEP may flow control the rate at which the sending DLS-user may send DLSDUs, if supported by the DLC's QoS.

NOTE 4 A subscribing DLS-user of a multi-peer DLC may not flow control the rate at which the publishing DLS-user sends DLSDUs because this flow control would affect all the DLC's other DLS-users. Instead, the publishing DLS-user should employ some form of self-administered control of its publishing rate to minimize the probability of DLSDU loss due to congestion at receiving DLS-users. A sending DLS-user at a peer DLCEP whose QoS does not support flow control should adopt a similar policy of rate control.

- f) A means by which a DLCEP, and possibly a DLC, can be returned to a defined state and the activities of the DLS-users resynchronized by use of a Reset DL-facility.

NOTE 5 Reset of a multi-peer DLCEP by a subscribing DLS-user or its local DLE does not cause the DLC to be reset at any of its other DLCEPs.

- g) A means by which the publishing DLS-user of a multi-peer DLC can query whether there are any subscribing DLS-users of the DLC.
- h) A means for the unconditional, and therefore possibly destructive, release of a DLCEP and possibly a DLC, by one of the DLC's DLS-users or by the DLS-provider.

NOTE 6 Release of a multi-peer DLCEP by a subscribing DLS-user or its local DLE does not cause the release of the DLC at any of its other DLCEPs.

There are four classes of connection-mode DLS: classes A (most comprehensive) to D (minimal). They differ in only a single QoS attribute — the available set of data delivery features (see 6.3.2.2). Classes A and B are capable of emulating OSI connection-mode and OSI connectionless services. Classes C and D are capable of emulating OSI connectionless services. All offer features that provide a basis for data distribution and distributed database cache-coherency protocols.

NOTE 7 True connectionless services also are available, independent of these four classes. See Clause 7.

6.2 Model of the connection-mode data-link layer service

Subclause 6.2 uses the abstract model for a layer service defined in ISO/IEC 10731:1994, Clause 5. The model defines interactions between the DLS-user and the DLS-provider that take place at the DLC's DLSAPs. Information is passed between the DLS-user and the DLS-provider by DLS primitives that convey parameters.

6.2.1 DLCEP-identification

A DLS-user may need to distinguish among several DLCEPs at the same DLSAP; thus a local DLCEP-identification mechanism is provided. All primitives issued at such a DLSAP within the context of a DLC use this mechanism to identify the local DLCEP. Other uses exist, such as the timeliness QoS parameters of 6.3.2.10 and the scheduling primitives of 8.5.2 and, 8.5.3. Thus this DLCEP-identification is explicitly included within the primitives that reference a local DLCEP. The naming-domain of this DLCEP-identification is the DL-local-view (see also 3.5.2).

6.2.2 Model of a peer DLC

Between the two end-points of a peer DLC there may exist a QoS-dependent flow control function that relates the behavior of the DLS-user receiving data to the ability of the other DLS-user to send data. As a means of specifying this flow control feature and its relationship with other capabilities provided by the connection-mode DLS, the OSI abstract queue model of a peer DLC, which is described in the following paragraphs, is used.

NOTE 1 The abstract queues referred to in this model are used solely for describing the interactions at the DLCEPs between the DLS-users, as mediated by the DLS-provider. They have no intended relationship to the actual queues managed by the primitives of Clause 5.

This abstract queue model of a peer DLC is discussed only to aid in the understanding of the end-to-end DLS features perceived by DLS-users. It is not intended to serve as a substitute for a precise, formal description of the DLS, nor as a complete specification of all allowable sequences of DLS-primitives. (Allowable primitive sequences are specified in 6.4. See also the note below.) In addition, this model does not attempt to describe all the functions or operations of DLEs that are used to provide the DLS. No attempt to specify or constrain DLE implementation is implied.

NOTE 2 The internal mechanisms that support the operation of the DLS are not visible to the DLS-user. Along with the interactions between service primitives described by this model (for example, the issue of a DL-RESET request at a DLSAP may prevent the receipt of a DL-DATA indication, corresponding to a previously issued DL-DATA request, by the peer DLS-user) there may also be

- a) constraints applied locally on the ability to invoke DLS-primitives; and
- b) service procedures defining particular sequencing constraints on some DLS-primitives.

6.2.2.1 OSI abstract queue model concepts

The OSI abstract queue model represents the operation of a peer DLC in the abstract by a pair of abstract queues linking the two DLCEPs. There is one abstract queue for each direction of information flow (see Figure 7).

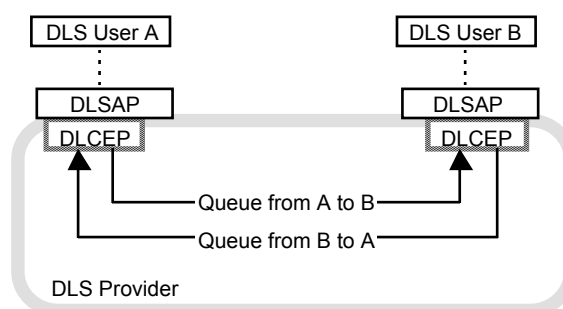


Figure 7 – OSI abstract queue model of a peer DLC between a pair of DLS-users

Each OSI abstract queue represents a flow control function in one direction of transfer. The ability of a DLS-user to add objects to an abstract queue will be determined by the behavior of the other DLS abstract queue. Objects are entered or removed from the abstract queue as a result of interactions at the two DLSAPs. Each peer DLC is considered to have one pair of abstract queues.

The following objects may be placed in an abstract queue by a DLS-user:

- a) a connect object, representing a DL-Connect primitive and its parameters;
- b) a data object, representing a DL-Data primitive and its parameters;
- c) a reset object, representing a DL-Reset primitive and its parameters; and
- d) a disconnect object, representing a DL-Disconnect primitive and its parameters.

The following objects may be placed in an abstract queue by the DLS-provider:

- 1) a reset object, representing a DL-Reset primitive and its parameters;
- 2) a synchronization mark object (6.2.2.4); and
- 3) a disconnect object, representing a DL-Disconnect primitive and its parameters.

The abstract queues are defined to have the following general properties:

- i) An abstract queue is empty before a connect object has been entered, and can be returned to this state, with loss of its contents, by the DLS-provider.
- ii) Objects are entered into an abstract queue by the sending DLS-user, subject to control by the DLS-provider. Objects may also be entered by the DLS-provider.
- iii) Objects are removed from the abstract queue under the control of the receiving DLS-user.
- iv) Objects are normally removed in the same order that they were entered (however, see 6.2.2.3).
- v) An abstract queue has a limited capacity, but this capacity is not necessarily either fixed or determinable.
- vi) Depending on the peer DLC's QoS, the abstract queue may occasionally duplicate or lose data objects.

6.2.2.2 Peer DLC / DLCEP establishment

When the DLS-provider receives a DL-CONNECT request primitive at one of the DLSAPs, it associates a pair of abstract queues and an abstract peer DLC between the two DLSAPs, and enters either a connect object or a disconnect object into the appropriate abstract queue. From the standpoint of the DLS-users of the peer DLC, the abstract queues remain associated with the peer DLC until a disconnect object representing a DL-DISCONNECT primitive is either entered into or removed from the abstract queue.

DLS-user A, which initiates a peer DLC establishment by entering a connect object representing a DL-CONNECT request primitive into the abstract queue from DLS-user A to DLS-user B, is not allowed to enter any other object, other than a disconnect object, into the abstract queue until after the connect object representing the DL-CONNECT confirm primitive has been removed from the DLS-user B to DLS-user A abstract queue.

In the abstract queue from DLS-user B to DLS-user A, objects other than a disconnect object can be entered only after DLS-user B has entered a connect object representing a DL-CONNECT response primitive, and has received a subsequent DL-CONNECTION-ESTABLISHED indication primitive. A disconnect object can be entered at any time.

While a peer DLC exists, the properties exhibited by the abstract queues represent the agreements concerning QoS reached among the DLS-users and the DLS-provider during this DLC establishment procedure.

6.2.2.3 Data transfer

Flow control on the peer DLC is represented in this queue model by the management of the queue capacity, allowing objects to be added to the queues. The addition of an object may prevent the addition of a further object.

Once objects are in the queue, the DLS-provider may manipulate pairs of adjacent objects, resulting in deletion. An object may be deleted if, and only if, the object that follows it is defined to be destructive with respect to the object. If necessary, the last object on the queue will be deleted to allow a destructive object to be entered — they may therefore always be added to the queue. Disconnect objects are defined to be destructive with respect to all other objects. Reset objects are defined to be destructive with respect to all other objects except connect and disconnect objects.

The relationships between objects that may be manipulated in the above fashion are summarized in Table 9.

Table 9 – Relationships between abstract queue model objects

The following (column) object is defined with respect to the preceding (row) object	Connect	Data	Reset	Synchronization mark	Disconnect
Connect	N/A	—	—	N/A	DES
Data	N/A	—	DES	N/A	DES
Reset	N/A	—	DES	—	DES
Synchronization Mark	N/A	—	DES	N/A	DES
Disconnect	N/A	N/A	N/A	N/A	DES
where N/A : X will not precede Y in a valid state of a queue — : not to be destructive nor to be able to advance ahead DES : to be destructive to the preceding object.					

Whether the DLS-provider performs actions resulting in deletion or not will depend upon the behavior of the peer DLC users and the agreed QoS for the peer DLC. With few exceptions, if a DLS-user does not remove objects from an abstract queue, the DLS-provider should, after some unspecified time, perform all the permitted deletions.

6.2.2.4 Peer DLC / DLCEP reset

To model the reset service accurately, a synchronization mark object is required. The synchronization mark object exhibits the following characteristics:

- It cannot be removed from an abstract queue by a DLS-user.
- An abstract queue appears empty to a DLS-user when a synchronization mark object is the next object in the queue.
- A synchronization mark can be destroyed by a disconnect object (see Table 9).
- When a reset object is immediately preceded by a synchronization mark object, both the reset object and the synchronization mark object are deleted from the abstract queue.

The initiation of a reset procedure is represented in the two abstract queues as follows:

- Initiation of a reset procedure by the DLS-provider is represented by the introduction into each abstract queue of a reset object followed by a synchronization mark object.
- A reset procedure initiated by the DLS-user is represented by the addition, by the DLS-provider, of objects into both abstract queues:
 - from the reset requestor to the peer DLS-user of a reset object; and
 - from the peer DLS-user to the reset requestor of a reset object followed by a synchronization mark object.

NOTE 1 The DLS-provider is always able to add a synchronization mark object following a reset object.

Unless destroyed by a disconnect object, a synchronization mark object remains in the abstract queue until the next object following it in the abstract queue is a reset object. Then the DLS-provider deletes both the synchronization mark object and the following reset object.

NOTE 2 Restrictions on the issuance of certain other types of DL-primitives are associated with the initiation of a reset procedure. These restrictions result in constraints on the entry of certain object types into the abstract queue until the reset procedure is completed (see 6.7.3.3).

6.2.2.5 Peer DLC / DLCEP release

The insertion into an abstract queue of a disconnect object, which may occur at any time, represents the initiation of a peer DLC release procedure. The release procedure may be destructive with respect to other objects in the two queues. The release procedure eventually results in the emptying of the queues and the disassociation of the queues from the peer DLC.

The insertion of a disconnect object may also represent the rejection of a peer DLC establishment attempt or the failure to complete peer DLC establishment. In such cases, if a connect object representing a DL-CONNECT request primitive is deleted by a disconnect object, then the disconnect object is also deleted. The disconnect object is not deleted when it deletes any other object, including the case where it deletes a connect object representing a DL-CONNECT response.

6.2.3 Model of a multi-peer DLC

Between the publishing and subscribing end-points of a multi-peer DLC, during the process of DLC establishment, there exists a relationship between the behavior of the responding DLS-user(s) and that of the initiating DLS-user. After DLC establishment, there exists a relationship between the publishing DLS-user and the subscribing DLS-user(s). As a means of specifying these relationships, the OSI abstract queue model of a multi-peer DLC, which is described in the following paragraphs, is used.

NOTE 1 The abstract queues referred to in this model are used solely for describing the interactions at the DLCEPs between the DLC users and the DLC provider. They have no relationship to actual queues managed by the primitives of Clause 5.

This abstract queue model of a multi-peer DLC is discussed only to aid in the understanding of the end-to-end DLS features perceived by DLS-users. It is not intended to serve as a substitute for a precise, formal description of the DLS, nor as a complete specification of all allowable sequences of DLS primitives. (Allowable primitive sequences are specified in 6.4. See also the note below.) In addition, this model does not attempt to describe all the functions or operations of DLEs that are used to provide the DLS. No attempt to specify or constrain DLE implementation is implied.

NOTE 2 The internal mechanisms that support the operation of the DLS are not visible to the DLS-user. In addition to the interactions between service primitives described by this model (for example, the issue of a DL-DISCONNECT request at a receiving DLSAP will prevent the receipt of a DL-DATA indication, corresponding to a previously issued DL-DATA request, by the sending DLS-user) there may also be:

- a) constraints applied locally on the ability to invoke DLS-primitives; and
- b) service procedures defining particular sequencing constraints on some DLS-primitives.

6.2.3.1 OSI abstract queue model concepts

The OSI abstract queue model represents the operation of a multi-peer DLC in the abstract by a set of abstract queues linking the publishing DLCEP with the subscribing DLCEPs — one queue per receiving DLCEP (see Figure 8).