INTERNATIONAL STANDARD

IEC 61158-3

Third edition
2003-05

Digital data communications for measurement and control – Fieldbus for use in industrial control systems –

Part 3: Data link service definition

Reference number
IEC 61158-3:2003(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**
- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (http://www.iec.ch/searchpub/cur_fut.htm) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (http://www.iec.ch/online_news/justpub/jp_entry.htm) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 61158-3

Third edition
2003-05

Digital data communications for measurement and control – Fieldbus for use in industrial control systems –

Part 3: Data link service definition

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XM

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

DIGITAL DATA COMMUNICATIONS FOR MEASUREMENT AND CONTROL – FIELDBUS FOR USE IN INDUSTRIAL CONTROL SYSTEMS –

Part 3: Data Link Service definition

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the 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, the IEC publishes International Standards. 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. The 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 the 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 National Committees.
3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
6) Attention is drawn to the possibility that some of the elements of this part of this International Standard may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61158-3 has been prepared by subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

This third edition cancels and replaces the second edition published in 2000. This third edition constitutes a technical revision.

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

FDIS	Report on voting
65C/290/FDIS	65C/298/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 edition includes the following significant changes from the prior edition:

a) addition of multicast send and receive data and clock synchronization to the Type 3 service definition;
b) a substantial rewrite of 3.8 and Clause 17, part of the Type 6 service definition.

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

The committee has decided that the contents of this publication will remain unchanged until 2007. At this date, the publication will be

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

IEC 61158 consists of the following parts, under the general title *Digital data communications for measurement and control — Fieldbus for use in industrial control systems*:

Part 1: *Overview and guidance for the IEC 61158 series*

Part 2: *Physical Layer specification and service definition*

Part 3: *Data Link Service definition*

Part 4: *Data Link Protocol specification*

Part 5: *Application Layer Service definition*

Part 6: *Application Layer protocol specification*

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, which is based in part on the Basic Reference Model for Open Systems Interconnection. Both Reference Models subdivide the area of standardization for interconnection into a series of layers of specification, each of manageable size.

The Data Link Service is provided by the Data Link Protocol making use of the services available from the Physical Layer. This part of the IEC 61158 series defines the Data Link Service characteristics that the immediately higher-level protocol may exploit. The relationship between the International Standards for Fieldbus Data Link Service, Fieldbus Data Link Protocol, Fieldbus Application Protocol and Systems Management is illustrated in Figure 1.

NOTE Systems Management, as used in this standard, is a local mechanism for managing the layer protocols

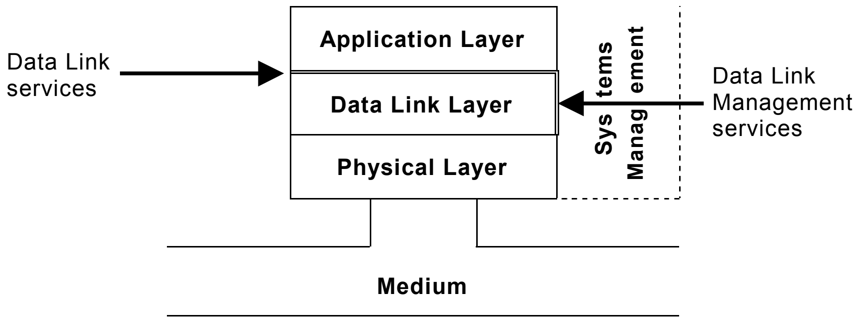


Figure 1 – Relationship of IEC 61158-3 to other fieldbus layers and to users of the Fieldbus Data Link Service

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 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.

DIGITAL DATA COMMUNICATIONS FOR MEASUREMENT AND CONTROL – FIELDBUS FOR USE IN INDUSTRIAL CONTROL SYSTEMS –

Part 3: Data Link Service definition

1 Scope and object

1.1 Overview

This part of IEC 61158 provides 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 part of IEC 61158 defines in an abstract way the externally visible service provided by the Fieldbus Data Link Layer in terms of

a) the primitive actions and events of the service;

b) the parameters associated with each primitive action and event, and the form which they take; and

c) the interrelationship between these actions and events, and their valid sequences.

The purpose of this part of IEC 61158 is to define the services provided to

1) the various types of Fieldbus Application Layer at the boundary between the Application and Data Link Layers of the Fieldbus Reference Model, and

2) Systems Management at the boundary between the Data Link Layer and Systems Management of the Fieldbus Reference Model.

Seven distinct types of services are defined in this part of IEC 61158; each has a corresponding protocol in IEC 61158-4. The seven distinct types of DL-service are:

Type 1 — A DL-service which provides a superset of those services expected of OSI Data Link Protocols as specified in ISO/IEC 8886.

Type 2 — A DL-service which provides both a connected and a connectionless subset of those services specified in ISO/IEC 8886.

Type 3 — A DL-service which provides a connectionless subset of those services specified in ISO/IEC 8886.

Type 4 — A DL-service which provides a connectionless subset of those services specified in ISO/IEC 8886.

NOTE 1 This part of IEC 61158 does not define a Type 5 Data Link service. Other parts of IEC 61158 define a Type 5 Application Layer service and protocol. The designation Type 5 is reserved in this part of IEC 61158 to maintain numbering consistency with the other parts of IEC 61158.

Type 6 — A DL-service which provides both a connected and a connectionless subset of those services provided by OSI Data Link Protocols as specified in ISO/IEC 8886.

Type 7 — A DL-service which provides both a connected and a connectionless subset of those services provided by OSI Data Link Protocols as specified in ISO/IEC 8886.

Type 8 — A DL-service which provides a connection-oriented subset of those services specified in ISO/IEC 8886.

NOTE 2 Many of these Types of service are suitable for use with multiple higher-layer protocols. In addition to the potential ability of these types of Data Link service to support unrelated Types of Fieldbus Application Layer protocol, some of these Types of Data Link service also may be able to support:

a) the OSI Network Layer at the boundary between the Network and Data Link Layers of the OSI Basic Reference Model

b) the IETF (IP) Network Layer

c) the Smart Transducer Interface for Sensors and Actuators as defined in IEEE 1451.2.

Where the scope of addressing is adequate, some of these Types of Data Link service also may be able to support

d) an OSI Transport Layer Protocol.

NOTE 3 Use of some of these protocol types is restricted by their copyright holders. In all cases a particular Data Link protocol Type can be used without restriction when coupled with the same Type Physical Layer and Application Layer protocols, or with other combinations as specified in IEC 61784. Use of the various protocol Types in other combinations may require permission of their respective copyright holders.

1.2 Specifications

The principal objective of this part of IEC 61158 is to specify the characteristics of conceptual Data Link 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. It is this latter objective which gives rise to the diversity of services standardized in this part of IEC 61158, and the corresponding protocols standardized in IEC 61158-4.

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, and

b) the correlation of paired request and confirm, or indication and response, primitives.

1.3 Conformance

This part of IEC 61158 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 Service definition standard. Instead, conformance is achieved through implementation of conforming Data Link protocols that fulfill any given Type of Data Link Services as defined in this part of IEC 61158.

1.4 Scope of type-specific clauses and subclauses

The different Types of Data Link services defined by this standard are each presumed self-consistent, but in general are unrelated to the other Types of service.

Where a clause or subclause heading explicitly designated one or more Types, that clause or subclause applies only to that (those) Type(s) and all references within that clause or subclause are with respect to that (those) Type(s).

Where a clause does not explicitly designate specific Types, or neither a subclause nor its containing clause explicitly designate specific Types, then that material is presumed to apply to multiple Types.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 7498, *Information technology – Open Systems Interconnection – Basic Reference Model*

ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model — Basic Reference Model: The Basic Model*

ISO/IEC 7498-3:1997, *Information technology – Open Systems Interconnection – Basic Reference Model — Basic Reference Model: Naming and addressing*

ISO/IEC 8886:1996, *Information technology – Open Systems Interconnection – Data Link Service Definition*

ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

IEC 61158-4:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 4: Data link protocol specification*

IEEE Std 1451.2:1997, *A Smart Transducer Interface for Sensors and Actuators – Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheets (TEDS) Formats*

3 Terms and definitions

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

3.1 Reference model terms and definitions

This part of IEC 61158 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 [7498-1]

3.1.15 flow control [7498-1]

3.1.16 (N)-layer [7498-1]
DL-layer (N=2)
Ph-layer (N=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 [7498-1]
DL-service (N=2)
Ph-service (N=1)

3.1.32 (N)-service-access-point [7498-1]
DL-service-access-point (N=2)
Ph-service-access-point (N=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 DL-user-data [7498-1]

3.2 Service convention terms and definitions

This part of IEC 61158 also makes use of the following terms defined in ISO/IEC 10731 as they apply to the Data Link Layer:

3.2.1 acceptor

3.2.2 asymmetrical service

3.2.3 confirm (primitive);
requestor.deliver (primitive)

3.2.4 deliver (primitive)

3.2.5 DL-confirmed-facility

3.2.6 DL-facility

3.2.7 DL-local-view

3.2.8 DL-mandatory-facility

3.2.9 DL-non-confirmed-facility

3.2.10 DL-provider-initiated-facility

3.2.11 DL-provider-optional-facility

3.2.12 DL-service-primitive; primitive

3.2.13 DL-service-provider

3.2.14 DL-service-user

3.2.15 DL-user-optional-facility

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

3.2.17 multi-peer

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

3.2.19 requestor

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

3.2.21 submit (primitive)

3.2.22 symmetrical service

3.3 Common Data Link Service terms and definitions

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

For the purpose of this part of IEC 61158, the following definitions also apply:

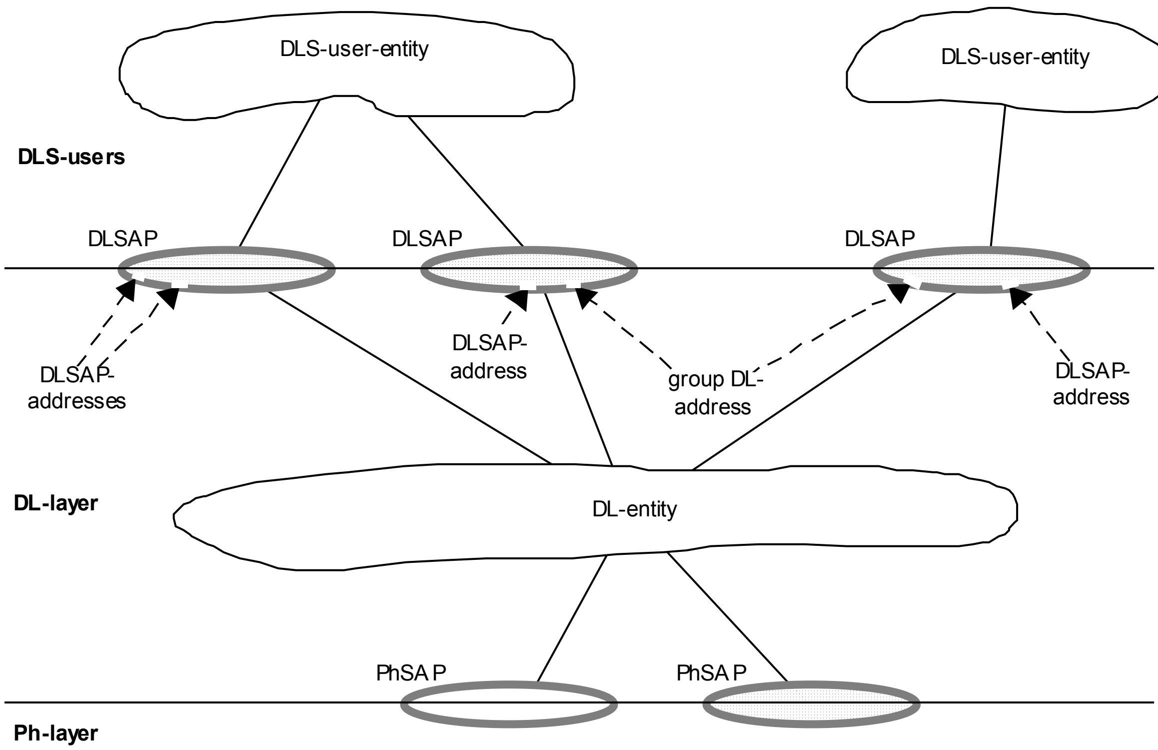
3.3.1 DL-segment, link, local link

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

3.3.2 DLSAP

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

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



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.

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

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

3.3.3 DL(SAP)-address

either an individual DLSAP-address, designating a single DLSAP of a single DLS-user, or a group DL-address potentially designating multiple DLSAPs, each of a single DLS-user

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

3.3.4 (individual) DLSAP-address

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

3.3.5 extended link

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

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

3.3.6 frame

denigrated synonym for DLPDU

3.3.7 group DL-address

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

3.3.8 node

single DL-entity as it appears on one local link

3.3.9 receiving DLS-user

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

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

3.3.10 sending DLS-user

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

3.4 Type 1: Additional Data Link Service terms and definitions

NOTE For historic reasons, some of the following definitions are more extensive than is common to the Definition clauses of other IEC standards.

3.4.1 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

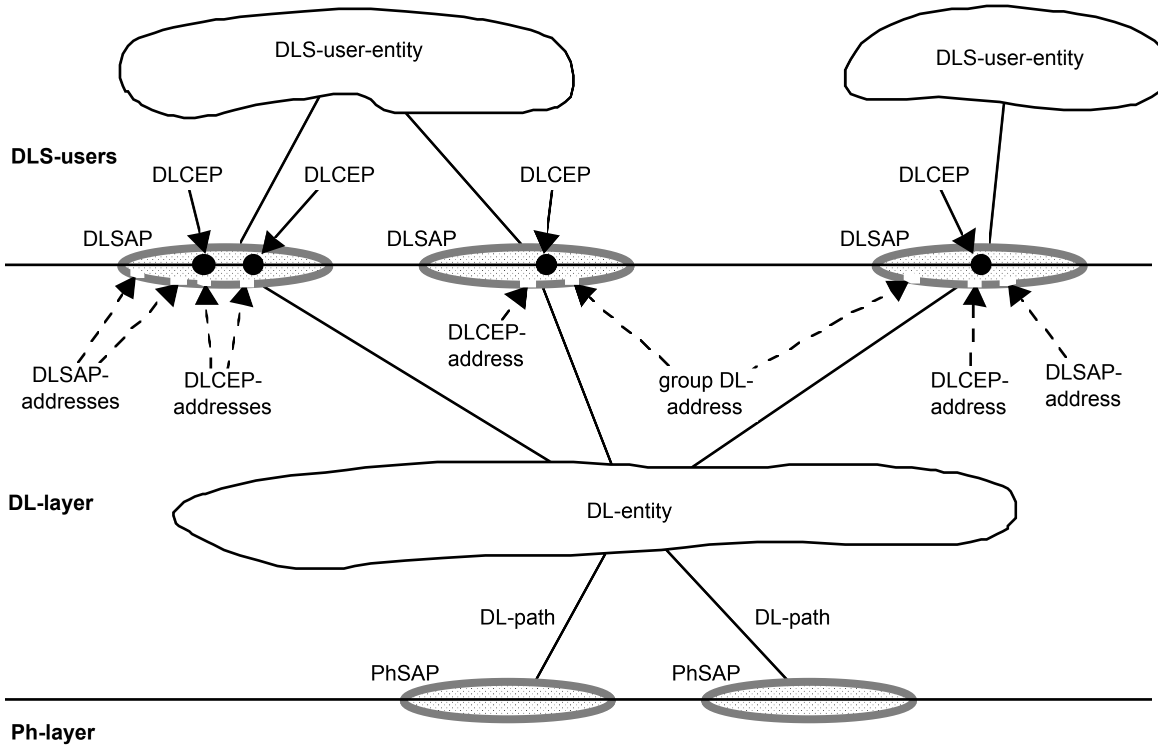
3.4.2 DLCEP-address

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



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

NOTE 4 This figure also shows the relationships of DL-paths and PhSAPs.

Figure 3 – Relationships of DLCEPs and DLCEP-addresses to DLSAPs, DLSAP-addresses and group DL-addresses

3.4.3 DLSEP-address

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

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

3.4.4 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 Some prior national standards have referred to this role as a “master” role

3.4.5 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 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 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

3.4.6 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 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

3.4.7 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 Some prior national standards have referred to this role as a “slave” role

3.4.8 timeliness, DL-timeliness

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

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

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

NOTE These concepts also support migration from previous national standards

3.4.9 transaction

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

NOTE 1 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 A DL-entity can be both an initiator and a responder in the same transaction.

3.5 Type 2: Additional Data Link Service terms and definitions

3.5.1 application

function or data structure for which data is subscribed or published

3.5.2 behavior

indication of how the object responds to particular events. Its description includes the relationship between attribute values and services

3.5.3 bridge, DL-router

DL-relay entity which performs selective store-and-forward and routing functions to connect two or more separate DL-subnetworks (links) to form a unified DL-subnetwork (the extended link)

3.5.4 cyclic

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

3.5.5 device

physical hardware connection to the link

NOTE A device may contain more than one node.

3.5.6 DL-subnetwork

series of nodes connected by PhEs and, where appropriate, DL-routers

3.5.7 DLPDU

Data Link Protocol Data unit

NOTE A DLPDU consists of a source MAC ID, zero or more Lpackets, and an FCS, as transmitted or received by an associated PhE.

3.5.8 error

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

3.5.9 fixed tag

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

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

NOTE Identification of the target node(s) is included in the two octet tag

3.5.10 frame

denigrated synonym for DLPDU

3.5.11 generic tag

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

3.5.12 guardband

time slot allocated for the transmission of the moderator DLPDU

3.5.13 link

collection of nodes with unique MAC IDs. Ph-segments connected by Ph-repeaters make up a link; links connected by DL-routers make up an extended link (sometimes called a local area network)

3.5.14 Lpacket

well-defined sub-portion of a DLPDU containing (among other things)

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

3.5.15 moderator

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

3.5.16 moderator DLPDU

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

3.5.17 multipoint DLC

centralized multi-end-point DL-connection offering DL-simplex-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). A multipoint DLC always provides asymmetrical service.

3.5.18 node

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

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

3.5.19 peer-to-peer DLC

point-to-point DL-connection offering DL-simplex-transmission between a single distinguished sending DLS-user and a single distinguished receiving DLS-user

NOTE A peer-to-peer DLC always provides asymmetrical service.

3.5.20 rogue

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

3.5.21 scheduled

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

3.5.22 tMinus

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

3.5.23 tone

the instant of time which marks the boundary between two NUTs

3.5.24 unscheduled

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

3.6 Type 3: Additional Data Link Service terms and definitions

3.6.1 acknowledgement DLPDU

reply DLPDU that contains no DLSDU

3.6.2 bit time

time to transmit one bit

3.6.3 clock synchronization

represents a sequence of interactions to synchronize the clocks of all time receivers by a time master

3.6.4 controller_type

hardware class of the communications entity

3.6.5 data DLPDU

DLPDU that carries a DLSDU from a local DLS-user to a remote DLS-user

3.6.6 DL_status, DLM_status

status that specifies the result of the execution of the associated request

3.6.7 GAP

range of station (DLE) DL-addresses from this station (TS) to its successor (NS) in the logical token ring, excluding stations above HSA

3.6.8 isochronous mode

constant cycle of messages from a time master for synchronization of other stations

3.6.9 local DLE

DLE in a current master station that initiates the current transaction

3.6.10 local DLS-user

DLS-user that initiates the current service

3.6.11 publisher

transmitter of messages for consumption by subscribers

3.6.12 region/segment address

address extension that identifies a particular fieldbus subnetwork

3.6.13 remote DLE

addressed DLE of a service request (that is, the intended receiving DLE of any resulting request DLPDU)

3.6.14 remote DLS-user

addressed DLS-user of a service request (that is, the intended receiver of any resulting indication primitive)

3.6.15 reply DLPDU

DLPDU transmitted from a remote DLE to the initiating (local) DLE, and possibly other DLEs

NOTE When the remote DLE is a Publisher, the reply DLPDU also can be sent to several remote DLEs.

3.6.16 request DLPDU

DLPDU that carries either a request for data or a DLSDU or both from a local DLS-user to a remote DLS-user

3.6.17 response DLPDU

reply DLPDU that carries a DLSDU from a remote DLS-user to local DLS-user

3.6.18 station

master or slave device containing a DLE

3.6.19 subscriber

receiver of messages produced by a publisher

3.6.20 time event

message that represents a trigger for a moment of time

3.6.21 time master

device which is able to send clock synchronization messages

3.6.22 time receiver

fieldbus device able to be time synchronized by a time master

3.6.23 token passing

medium access method, in which the right to transmit is passed from master station to master station in a logical ring

3.7 Type 4: Additional Data Link Service terms and definitions

3.7.1 broadcast-node-address

used to send broadcasts to all DLEs on a Link

NOTE All DLEs on a Link receive all DLPDUs where the first Node-address is equal to the Broadcast-Node-Address. Such DLPDUs are always unconfirmed, and their receipt is never acknowledged. The value of the Broadcast-Node-Address is 126

3.7.2 destination-DL-route

holds a sequence of DL-route-elements, describing the complete route to the destination

NOTE This includes both the destination DLSAP and a local component meaningful to the destination DLS-user.

3.7.3 DL-route-element

an octet, holding a Node DL-address or an address used by the DLS-user

3.7.4 DLS-user address

uniquely identifies a DLS-user locally

3.7.5 full DL-route

the combination of a destination-DL-route and a source-DL-route

3.7.6 maximum indication delay

indicates to the DLS-user the maximum time interval for the DLS-user to prepare a response after receiving an indication requiring a response

NOTE If the DLS-user is unable to prepare a response within Maximum Indication Delay, the DLS-user is required to issue a DL-UNITDATA request with a DLSDU type indicating ACKNOWLEDGE. As a result the DLE will transmit an acknowledging DLPDU on the Link.

3.7.7 maximum retry time

indicates to the DLE for how long time retransmission of the request may be performed, as a result of Wait acknowledges from the remote DLE or DLS-user

3.7.8 no-confirm-node-address

indicates that a request or response is Unconfirmed

NOTE The value of the No-Confirm-Node-address is 0

3.7.9 node-address

uniquely identifies a DLE on a Link

NOTE The value of a Node-address can be in the range of 0-127. The values 0, 126 and 127 are reserved for special purposes

3.7.10 normal class device

device which replies to requests from other normal class devices, and initiates transmissions

NOTE Such a device can act as a server (responder) and as a client (requestor) – this is also called a peer.

3.7.11 service-node-address

an address reserved for service purposes only

NOTE All DLEs on a Link receive all DLPDUs where the first Node-address is equal to the Service-Node-Address. Such DLPDUs can be Confirmed or Unconfirmed, and their receipt may or may not be acknowledged. The Service-Node-Address can be used on Links with only two DLEs — the requesting Normal class DLE and the responding Simple or Normal class DLE. The value of the Service-Node-Address is 127.

3.7.12 simple class device

device which replies to requests from normal class devices

NOTE Such a device can act as a server or responder only

3.7.13 source-DL-route

holds a sequence of DL-route-elements, describing the complete route back to the source

3.8 Type 6: Additional Data Link Service terms and definitions

3.8.1 Additional definitions related to IEEE Std 1451.2

3.8.1.1 TEDS

Transducer Electronic Data Sheet. Non-volatile memory that stores the configuration of a smart field device

3.8.1.2 transducer

sensor or an actuator

3.8.1.3 trigger

signal from a smart NCAP that can trigger synchronized internal or external actions, on the local or extended link

3.8.2 Variations on the common definitions of 3.1 through 3.4

3.8.2.1 bridge, DL-relay

DL-relay entity which performs synchronization between links (buses) and may perform selective store-and-forward and routing functions to connect two or more separate DL-subnetworks (links) to form a unified DL-subnetwork (the extended link)

3.8.2.2 DLCEP-address

see 3.4.2

3.8.2.3 peer DLC, GP channel

point-to-point DL-connection offering DL-duplex-transmission between two peer DLS-users where each can be a Calling DLS-user and each can be a Called DLS-user

NOTE A peer DLC is configured to provide either Confirmed or Acknowledged service

3.8.2.4 multi-peer DLC

multi-end-point DL-connection offering DL-simplex-transmission between a single distinguished DLS-user, known as the **publisher** or **publishing** DLS-user, and a set of one or more peer distinguished 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)

NOTE A multi-peer DLC always provides DL-simplex service, from the publisher to the subscribers.

3.8.2.5 master DLCEP

peer DLCEP which controls the direction of data transfers on a GPC channel

3.8.2.6 slave DLCEP

peer DLCEP which may only initiate data transfers on a GPC channel when authorized by the Master peer DLCEP

3.8.3 Additional definitions

3.8.3.1 1WAY method:

channel which can only support a one-way flow of data

3.8.3.2 2WAY method

channel which can support a two-way flow of data

3.8.3.3 active nodes and channels

nodes and channels that are able to communicate on the local link

NOTE Active nodes have No-Comm-Node equal 0. Active channels have No-Comm-Node and No-Comm-Channel both equal 0

3.8.3.4 aliasing

any difference between the Time Stamps placed on a Event by two different DLEs

3.8.3.5 assembly (reassembly)

process of combining separate grains in the proper order to recreate a DLSDU as it was presented at the remote sending DLS

3.8.3.6 LBL (Log-Bus-Length)

base-2 logarithm of the number of slots in one Bus-cycle

3.8.3.7 BL ≠ BSS

referenced parameter can take on all values from 0 up to Bus Length, but excluding all Bus-Sync-slot numbers

3.8.3.8 bogus node

DLE found by the Conductor during startup that is not listed on the Node List

3.8.3.9 buffer-queue

memory area used for communicating DLS-user-data between the DLS-user and DLE in a local node for one direction of a channel, which may take the form of either a buffer or a FIFO (first-in first-out) limited-depth queue

3.8.3.10 bus-config

dynamic set of values that determine the timing of the bus in one installation

3.8.3.11 bus-configuration

dynamic set of values of Bus-Config, node, channel, trigger and forwarding-entity parameters that applies to one installation

3.8.3.12 bus-cycle

uniform progression of slots from 0 to the Bus-Cycle-Length

NOTE Each DLE maintains a sense of the current slot (a counter), which it uses to synchronize all Real-time actions.

3.8.3.13 bus-cycle-length

number of slots in a Bus-Cycle

NOTE It is always equal a power of 2 equal to 2^{LBL}. It's value is a configuration issue.

3.8.3.14 bus-end-sync-slot

last or "cycle-closing" slot of a bus cycle

NOTE The Conductor uses it to synchronize the bus cycle counter of all other active DLEs with the Conductor's own us cycle.

3.8.3.15 bus-sync-period

number of slots between bus sync slots

3.8.3.16 bus-sync-slot

variable duration slot used by the Conductor to fine tune synchronization of the slot clocks of all other active DLEs with the Conductor's slot-clock

NOTE The first Bus-Sync-slot is at Bus-Sync-Period slots.

3.8.3.17 bus-time

common sense of communications system time that is shared by all DLEs on the extended link

3.8.3.18 called DLS-user

higher-layer entity that accepts the transfer of DLS-User-Data

3.8.3.19 calling DLS user

higher-layer entity that initiates the transfer of DLS-User-Data

3.8.3.20 CD-semaphore

interlocked signaling mechanism that controls which DLS-user can transmit next on a GPC channel, that is, which specifies the Channel Direction

3.8.3.21 channel (DLC)

set of equally spaced slots per Bus-Cycle that conveys either:

a) measurements or control signals that can be received by one or more DLS-users using the SCAN or EXSCAN channel classes, or
b) interactions between two specific DLS-users using the GPC or GPA channel classes, or
c) messages between DLMS-users using the DLM-connectionless service

3.8.3.22 channel method (QoS building block)

attribute that determines one aspect of the fundamental operation of a channel. It applies to all the slots of one channel on a local or extended link

3.8.3.23 channel direction method

determines the allowed direction(s) of data flow on the channel

3.8.3.24 channel class

collection of channel-methods that together define the total operation and QoS of a channel

3.8.3.25 channel selector

identifies any slot during which data for a particular channel is or should be present on the local link

3.8.3.26 conductor

the DLE that is synchronizing all traffic on a bus, and which may also configure all DLEs and channels on that bus

NOTE The Conductor does not schedule traffic as does a Type 1 Link Active Scheduler (LAS).

3.8.3.27 configuration database

contains the configured values of the DLE parameters for each DLE channel, trigger and half-forwarding-port for one usage (experiment/batch/process) of the bus

3.8.3.28 configured nodes

DLEs that have been successfully configured per the Node List

3.8.3.29 conformance-class

fixed set values of parameters that are defined as an integrated set and which apply to multiple installations

3.8.3.30 data

generic term used to refer to any information carried over a fieldbus

3.8.3.31 data field

atomic DLSDU carried by the UNITARY method

3.8.3.32 data-packaging method

attribute that determines whether DLSDUs sent on the channel will be granulated or not

3.8.3.33 destination DLSAP

DLSAP that accepts the transfer of a message using the DLM-connectionless method

3.8.3.34 DL-buffer identifier

local identifier for the data passed in a buffer by the associated primitive, shared by the DLS provider and DLS-user

3.8.3.35 DL-Event-time

time at which an Event occurred, as calculated by the Conductor

3.8.3.36 DL-queue identifier

local identifier for the data passed in a queue by the associated primitive, shared by the DLS provider and DLS-user, or by the DLMS provider and DLMS-user

3.8.3.37 DLC-identifier, channel-ID

Identifier for one Channel (DLC) from within all Channels on a local link

NOTE The value of the Channel-ID is equal to the relative index of the first slot (after the Bus-cycle-End slot) assigned to this channel.

3.8.3.38 DLM-connectionless service

service used for bus configuration with less integrity than GPC or GPA connection mode services, in which the DLS does not relate any DLM-connectionless DLPDU to any other DLPDU or make any retry attempt on detection of an error

3.8.3.39 DLMS-user

user of the DL-Management service, generally, of the DLM bus configuration service

3.8.3.40 DLMSDU

service data unit sent or received by a DLMS-user, generally, with structure and content specified by this standard

3.8.3.41 DLSDU-length

length in octets of an item of DLS-user-data

3.8.3.42 forwarding-bridge-DLE

DLE that synchronizes the bus cycles and slot clocks for two or more connected local links, and which can be configured to forward DLSDUs among the connected links

3.8.3.43 general purpose acknowledged (GPA) class

Provides reliable communications for higher layer protocols that depend on classical one-way DL-data-delivery.

3.8.3.44 general purpose confirmed (GPC) class

channel which provides reliable communications for higher layer protocols that depend on classical one-way DL-data-delivery

3.8.3.45 grain: (partial DLSDU)

channel which provides reliable communications for higher layer protocols that depend on classical two-way DL-data-delivery

3.8.3.46 GRANULAR channel

channel which can perform Granulation and Assembly.

3.8.3.47 GRANULAR method

channel on which each DLSDU is transferred in one or more Grains.

NOTE When the Calling DLS-user on a GRANULAR channel submits DLSDUs longer than the data “payload” of the DLPDUs now in use on this bus (that is, the DLSDU-length exceeds the configured max-data-length of the bus), the DLE **granulates** (segments) each DLSDU into multiple DLPDUs. Each grain carries a one octet header that allows reconstruction by the called DLE of the original submitted DLSDU. The receiving DLE assembles the grains and validates the DLSDU before delivery to the Called DLS-user..

3.8.3.48 granulation (segmentation)

see 3.8.3.47

3.8.3.49 HLP (higher level protocol)

a DLS-user protocol

3.8.3.50 HLP-PDU

protocol information conveyed by the DLE between two or more communicating DLS-users which are using the same HLP

3.8.3.51 host

DLE that can become a Conductor and synchronize operation of a bus, and which may source and/or sink “data” on a local or extended link

3.8.3.52 I/O channel

hardware interface to a “real world” transducer

NOTE This concept maps directly onto the “data transfer” channels of the protocol.

3.8.3.53 local link

single bus with a single Conductor

3.8.3.54 low order part of a data item

least significant (written as rightmost) octets/bits of the item

3.8.3.55 max-data-length

maximum length of the data field of any DLPDU which may be transmitted on an extended link

NOTE max-data-length is an element of Bus-Config

3.8.3.56 max-DLSDU-buffer-size

maximum buffer size for the conformance-class in use on this Bus

3.8.3.57 max-DLSDU-length

maximum buffer size for the conformance-class in use on a bus

3.8.3.58 max-DLE-group-address

maximum number of Group DLE addresses per local link

3.8.3.59 max-net-level

maximum depth of the nest of real links

3.8.3.60 max-time-stamp

maximum number of time-stamp buffers and messages that can be present in one DLSAP

3.8.3.61 max-trigger

maximum number of triggers that can be present in one DLSAP

3.8.3.62 message

single event or upper level protocol triggered transmission of a DLSDU

3.8.3.63 missing node

node that is listed on the Node-List that can NOT communicate on this local link

3.8.3.64 net-ID, link-address

hierarchical identification of the subnetwork on which a DLE resides in the real network topology, identifying the local link as seen from the TOP level (L0) of the extended link looking DOWN, used by DLM-connectionless services

3.8.3.65 node

generic term for the DLE, sometimes also including its associated DLS- or DLMS-user

3.8.3.66 node-list

list that contains the bus configuration information for each node, channel, half-forwarding-port and trigger which should be active on the local link

3.8.3.67 non-retentive buffer (queue of depth one)

buffer (equivalently, a queue of depth one) for which DLSDU (or DLMSDU) presence is cleared immediately after the contained DLS-user-data has been transmitted by the DLE or read by the DLS-user (or DLMS-user)

3.8.3.68 num-time-stamp

number of time-stamp-buffers present in a DLSAP

3.8.3.69 offset slot, channel-ID

starting slot number for designating a specified channel

3.8.3.70 parameter set by DL-Management

a parameter specified by the Profile currently in use or by the Bus-Configuration process

3.8.3.71 protocol Information field

either the DLPDU field (in Transfer DLPDUs) or Bus-PDU field (in Bus-Sync DLPDUs)

3.8.3.72 publisher DLSAP

DLCEP that attempts to transmit a DLPDU in each slot assigned to a specific channel on a multi-peer DLC

3.8.3.73 QoS-set

selector that identifies, for one DLCEP, a specific set of QoS options selected from those specified for the conformance-class in use, therby implicitly retricting the set of DLCEPs that can support a specific DLS-user in a matching role, that is, those DLCEPs that can participate in a specified DLS-user protocol

3.8.3.74 retentive-buffer

buffer for which transmission or reading of the DLSDU does not cause the DLE to empty the buffer, where the sending DLS-user clears the buffer by writing to it and the receiving DLE overwrites the value in the buffer when it receives a new DLSDU

3.8.3.75 scan-rate

number of slots per second assigned to a channel

3.8.3.76 slot

basic unit of time allocation on a bus

3.8.3.77 source DLSAP

DLSAP that initiates the transfer of a message using the DLM-connectionless method

3.8.3.78 subscriber DLCEP

DLCEP that attempts to receive the DLPDUs present in the slots assigned to a specific channel on a multi-peer DLC

3.8.3.79 system

extended link with all of its attached DLEs

3.8.3.80 system manager

agent which sets up the system configuration in the node list and monitors configuration and operation of the system

3.8.3.81 synchronizer-DLE

synchronizes the Bus-Cycles and Slot-Clocks between 2 or more local links, but does not support data transfers between local links

3.8.3.82 time stamp

attribute of a datum which provides the bus time at which the underlying event was detected by the submitting DLS-user

3.8.3.83 time-stamping of Events

recording a time-stamp concurrent with an external signal and appending the recorded time-stamp to a later transmitted Event message

NOTE The time stamp appended to the message may be approximately concurrent with the time the relevant Event occurred, even if the Event message is transmitted much later.

3.8.3.84 transmission Delay

any unwanted delay in transmitting transducer readings

3.8.3.85 transparent data conveyance

guarantee that for each DLPDU transmitted, the called DLS-user receives either an error indication or exactly the "data" submitted by the calling DLS-user

3.8.3.86 UNITARY Method:

channel on which a complete DLSDU is always transferred in a single Transfer DLPDU.

3.8.3.87 VID (of a DLE)

externally visible identification of a DLE and its contained objects, identical to the DLE's DLEA and used by DLM-connectionless services

3.8.3.88 virtual topology

connection mode topology assigned by the Bus-configuration DLMS-user, which need not map directly to physical topology.

3.9 Type 7: Additional Data Link Service terms and definitions

3.9.1 acknowledgement response DLPDU

information that the recipient of an acknowledged message emits in order to signal either the proper reception of the message or the lack of available resources to store the message, received by the DLE on the local link that emitted the message which requested the acknowledgement

3.9.2 basic cycle

sequence of scanning by the Bus Arbitrator of:

a) a set of DLCEP-identifiers for variables, requests, and cyclical application messages,
b) plus a window provided for aperiodic exchanges,
c) plus a window provided for message services,
d) plus a window provided for synchronization

3.9.3 basic transaction

succession of DLPDUs related to a single DL-service instance

3.9.4 bus arbitrator (BA)

DLE that controls each data producer's right to access the medium

NOTE At any given instant one and only one Bus Arbitrator is active in each DL-segment of a DL-subnetwork.

3.9.5 control field

portion of an emitted or received DLPDU that gives the nature of the data exchanged and the type of exchange

3.9.6 destination address

three octets specifying the DL-segment of the DLE to whom the message is sent, and the destination DLSAP's sub-address within the local link

3.9.7 DL-segment, local link

set of devices that respect the DL-protocol and that are interconnected through a PhL. Only one Bus Arbitrator is active on a single DL-segment

3.9.8 DLCEP-identifier

two octets specifying a link-local DLCEP-identifier associated with a system variable. A DLCEP-identifier uniquely designates a single DL-accessible variable within the local link

3.9.9 DLCEP-identifier DLPDU

information that a Bus Arbitrator emits to allocate the local link to a data publisher for the purpose of exchanging a variable

3.9.10 end of message transaction indication DLPDU

information that the source entity of a message emits in order to return link access control to the Bus Arbitrator at the end of a message transaction

3.9.11 identified variable (or simply "variable")

DLL system variable for which an associated DLCEP-identifier has been defined

3.9.12 invalid DLCEP-identifier

a DLCEP-identifier not recognized locally

3.9.13 macrocycle

set of basic cycles needed for all cyclical DLCEP-identifiers to be scanned

3.9.14 message DLPDU identifier

information that a Bus Arbitrator emits to allocate the medium to a source DLE for a message transfer

3.9.15 message response DLPDU

information that a data publisher emits in response to a message identifier DLPDU. This information is received and retained by the desired destination entity or entities

3.9.16 periodic scanning of variables

action by the Bus Arbitrator that guarantees the cyclical exchange of variables

NOTE This is the basic principle of the Type 7 DL-service and protocol.

3.9.17 published identified variable

variable that corresponds to a DLCEP-identifier for which the DLE emits data

3.9.18 request DLPDU identifier

the information that a Bus Arbitrator emits to allocate the medium to the initiator of an explicit request for a buffer transfer

3.9.19 request response DLPDU

the information that the initiator of an explicit request for a buffer transfer emits in response to a request identifier DLPDU. This information is received by the Bus Arbitrator

3.9.20 source address

three octets specifying the local link-id of the entity sending the message, and the source DLSAP's sub-address within the local link

3.9.21 subscribed identified variable

variable that corresponds to a DLCEP-identifier for which the DLE receives data

3.9.22 triggered message scanning

function of a Bus Arbitrator that makes it possible to transfer messages

3.9.23 triggered periodic scanning of messages

function of a Bus Arbitrator that makes it possible to request triggered message exchanges cyclically

3.9.24 triggered periodic scanning of variables

function of a Bus Arbitrator that makes it possible to request triggered variable transfers cyclically

3.9.25 triggered scanning of variables

function of a Bus Arbitrator that makes possible the non-cyclical exchange of variables

3.9.26 turnaround time

time interval between reception or emission of the last MAC symbol of a DLPDU, signaled by a SILENCE indication from the PhL, and the reception or emission of the first MAC symbol of the subsequent DLPDU, signaled by an ACTIVITY indication from the PhL, both as measured in a given station

3.9.27 variable response DLPDU

information that a data producer emits in response to a DLCEP-identifier DLPDU, which also alerts data consumers to the relevance of the immediately time-proximate DLPDU.

3.10 Type 8: Additional Data Link Service terms and definitions

3.10.1 device

slave or master

3.10.2 device code

two octets which characterize the properties of a slave

3.10.3 DL-segment

group of slaves in consecutive order

3.10.4 DL-segment level

nesting level number of a DL-segment

3.10.5 master

DL-entity controlling the data transfer on the local link and initiating the medium access of the slaves by starting the DLPDU cycle

3.10.6 slave

DL-entity accessing the medium only after being initiated by the preceding slave or master

4 Symbols and abbreviations

4.1 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.

4.1.1 DL- Data Link layer (as a prefix)

4.1.2 DLC DL-connection

4.1.3 DLCEP DL-connection-end-point

4.1.4 DLE DL-entity (the local active instance of the Data Link layer)

4.1.5 DLL DL-layer

4.1.6 DLPCI DL-protocol-control-information

4.1.7 DLPDU DL-protocol-data-unit

4.1.8 DLM DL-management

4.1.9 DLME DL-management Entity (the local active instance of DL-management)

4.1.10 DLMS DL-management Service

4.1.11 DLS DL-service

4.1.12 DLSAP DL-service-access-point

4.1.13 DLSDU DL-service-data-unit

4.1.14 FIFO First-in first-out (queuing method)

4.1.15 OSI Open systems interconnection

4.1.16 Ph- Physical layer (as a prefix)

4.1.17 PhE Ph-entity (the local active instance of the Physical layer)

4.1.18 PhL Ph-layer

4.1.19 QoS Quality of service

4.2 Type 1: Additional symbols and abbreviations

4.2.1 DLSEP DL-schedule-end-point

4.3 Type 2: Additional symbols and abbreviations

4.3.1 MAC ID the DL-address of a node

4.3.2 MDS medium dependent sublayer

4.3.3 NUT network (actually, local link) update time

NOTE The use of the term "network" in the preceding definition is maintained for historic reasons, even though the scope involved is only a portion of a single DL-subnetwork.

4.3.4 SMAX MAC ID of the maximum scheduled node

4.3.5 Tx transmit

4.3.6 TUI table unique identifier

4.3.7 UCMM unconnected message manager

4.3.8 UMAX MAC ID of maximum unscheduled node

4.3.9 USR unscheduled start register

4.4 Type 3: Additional symbols and abbreviations

4.4.1 ACK Acknowledge(ment) DLPDU

4.4.2 cnf confirm primitive

4.4.3 CS DLS: Clock Synchronization

4.4.4 DA Destination address of a DLPDU

4.4.5 DAE Destination address extension(s) of a DLPDU – conveys D_SAP_index or destination Region/Segment address or both

4.4.6 DS DL/DLM_status: Disconnected station – local DL-entity not in logical token ring or disconnected from line

4.4.7 D_SAP Destination-service-access-point – a DLSAP which identifies the remote DLS-user.

4.4.8 D_SAP_index Destination-service-access-point – that component of a DLSAP-address which designates a DLSAP and remote DLS-user within the remote DLE

4.4.9 DXM Data exchange Multicast

4.4.10 EXT Address extension bit of a DLPDU

4.4.11 FC frame control (DLPDU type) field of a DLPDU

4.4.12 G GAP update factor – the number of token cycles between GAP maintenance (update) cycles

4.4.13 HSA Highest station address installed (configured) on this fieldbus

4.4.14 ind indication primitive

4.4.15 IoM Isochronous Mode

4.4.16 LMS List of Master stations

4.4.17 LR DL/DLM_status: Local resource not available or not sufficient

4.4.18 LS DL/DLM_status: Local service not activated at DLSAP or local DLSAP not activated

4.4.19 MSRD DLS: Send and Request Data with Multicast Reply

4.4.20 NA DL/DLM_status: No acknowledgement/response (DL/DLM_status of the service primitive)

4.4.21 NIL locally existing value, but not fixed by IEC 61158-3

4.4.22 NO DL/DLM_status: Not ok

4.4.23 NR DL/DLM_status: No response – DL/DLM-data acknowledgement negative and send data ok

4.4.24 NS Next station, the station to which this Master will pass the token

4.4.25 OK DL/DLM_status: Service finished according to the rules

4.4.26 RDH DL/DLM_status: Response DL-data high and no resource for send data

4.4.27 RDL DL/DLM_status: Response DL/DLM-data low and no resource for send data

4.4.28 req request primitive

4.4.29 RR DL/DLM_status: No resource for send data and no response DL-data available (acknowledgement negative) (DL/DLM_status of the service primitive)

4.4.30 RS DL/DLM_status: No service or no remote address activated at remote-service-access-point (acknowledgement negative) (DL/DLM_status of the service primitive)

4.4.31 SA Source address of a DLPDU

4.4.32 SAE Source address extension(s) of a DLPDU – conveys S_SAP_index or source Region/Segment address or both

4.4.33 SC Single character acknowledge DLPDU

4.4.34 SDA DLS: Send Data with Acknowledge

4.4.35 SDN DLS: Send Data with no Acknowledge

4.4.36 SRD DLS: Send and Request Data with Reply

4.4.37 S_SAP Source-service-access-point – a DLSAP which identifies the local DLS-user which initiates a transaction.

4.4.38 S_SAP_index Source-service-access-point index – a component of a DLSAP-address which designates that DLSAP within the DLE at which the transaction is being initiated

4.4.39 SYN Synchronizing bits of a DLPDU (period of IDLE) – it guarantees the specified DLPDU integrity and facilitates receiver synchronization

4.4.40 SYNCHT Synchronization telegram, indicates the start of a new cycle in IoM

4.4.41 t_{BIT} Bit time, DL-symbol period – the time to transmit one bit on the fieldbus ; 1/(data signaling rate in bit/s)

4.4.42 T_{CT} Cycle Time - the requested duration for one cycle in IoM

4.4.43 T_{QUI} Quiet time, transmitter fall time (line state uncertain time) or repeater switch time or both – the time a transmitting station needs to wait after the end of a DLPDU before enabling its receiver

4.4.44 **T_{RDY}** Ready time – the time after which the transmitting master will expect a reply DLPDU

4.4.45 **T_{RR}** Real rotation time – the time between the last successive receptions of the token by the observing master station

4.4.46 **TS** This station

4.4.47 **T_{SDI}** Station delay of initiator – the time a master station will wait before sending successive DLPDUs

4.4.48 **T_{SDR}** Station delay of responder – the actual time a responder needs to generate a reply DLPDU

4.4.49 **T_{SET}** Setup time – the time between an event (for example interrupt SYN timer expired) and the necessary reaction (for example enabling a receiver)

4.4.50 **T_{SH}** Time shift – the time a real isochronous cycle deviates from the requested duration for one cycle in IoM

4.4.51 **T_{SL}** Slot time – the maximum time a master station waits for a reply DLPDU

4.4.52 **T_{SYN}** Synchronization time – the period of IDLE before the beginning of a DLPDU after which a station enables its receiver; the required minimum inter-DLPDU idle period to guarantee DLPDU integrity and a valid DLPDU

4.4.53 **T_{SYNI}** Synchronization interval time – the maximum time that a receiving station waits for the required inter-DLPDU idle period, of duration T_{SYN}, to occur before it detects a bus fault

4.4.54 **T_{TR}** Target rotation time – the anticipated time for one token cycle, including allowances for high and low priority transactions, errors and GAP maintenance

4.4.55 **UE** DL/DLM_status: negative acknowledgement – remote user interface error

4.5 Type 4: Additional symbols and abbreviations

none

4.6 Type 6: Additional symbols and abbreviations

4.6.1 **64K** 2^{16} = 65 536

4.6.2 **B or Q** Buffer or Queue

4.6.3 **CEP_sel** a DLCEP selector within a DLE

4.6.4 **ConfigDB** Configuration Data base

4.6.5 **DLC** DL-connection, also, a Channel–

4.6.6 **DLEA** The symbol for a DLE-address

4.6.7 **SAP_sel** a DLSAP selector

4.6.8 **TDMA** Time Division Multiple Access

4.7 Type 7: Additional symbols and abbreviations

4.7.1 BA Bus Arbitrator

4.7.2 B_Dat_Cons Buffer which contains the value of the subscribed data

4.7.3 B_Dat_Prod Buffer which contains the value of the published data

4.7.4 B_Req1/2 Buffer containing the list of DL-identifiers that are the object of a specified explicit request for a transfer at the priority 1 (urgent) or 2 (normal)

4.7.5 RQ_Inhibit Indicator used to manage explicit requests for buffer transfers

4.7.6 Q_IDRQ1/2 Queue for the DL-identifiers requested, received by the BA at priority 1 (urgent) or 2 (normal)

4.7.7 Q_Msg_Aper Queue which contains messages to be emitted that are associated with aperiodic exchanges

4.7.8 Q_Msg_Cyc Queue which contains messages to be emitted that are associated with cyclical exchanges

4.7.9 Q_Req1/2 Queue containing the list of DL-identifiers that are the object of a free explicit request for a transfer at the priority 1 (urgent) or 2 (normal)

4.8 Type 8: Additional symbols and abbreviations

none

5 Conventions

5.1 General conventions

This part of IEC 61158 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.

5.1.1 Parameters

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 part of IEC 61158 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 part of IEC 61158. 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.

(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.

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.

5.2 Type 1: Additional conventions

5.2.1 Parameters

The parameter conventions of 5.1.1 are extended as follows:

The code used to specify the type of usage of the parameter on the primitive and parameter direction specified in the column is extended to include:

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.

The parameter-specific constraints are extended to include:

(≤) 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).

5.2.2 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 part of IEC 61158 is a purely local issue. Nevertheless, these identifiers are specified explicitly in these primitives to provide a descriptive means

a) of canceling (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).

5.3 Type 2: Additional conventions

none

5.4 Type 3: Additional conventions

In the diagrams which illustrate the DLS and DLM interfaces, dashed lines indicate cause-and-effect or time-sequence relationships between actions at different stations, while solid lines with arrows indicate cause-and-effect time-sequence relationships which occur within the DLE-provider at a single station.

The following notation, a shortened form of the primitive classes defined in 5.1, is used in the figures.

req request primitive

ind indication primitive

cnf confirm primitive (confirmation)

5.5 Type 4: Additional conventions

none

5.6 Type 6: Additional conventions

5.6.1 Parameters

The parameter conventions of 5.1.1 are extended to include all of those of 5.2.1. Additionally, the code used to specify the type of usage of the parameter on the primitive and parameter direction specified in the column is extended to include:

X — availability of the parameter is determined by DL-configuration conditions imposed by the Bus Configuration DLS-user.

5.6.2 Identifiers

These are identical to the identifiers described in 5.2.2.

5.7 Type 7: Additional conventions

The diagrams used to describe the sequence of primitives are composed of:

a) Vertical lines, representing the User-DLL interface,

b) Lines with arrows representing the time sequence of the primitives at the interface

c) Dotted lines, defining the relationships between the primitives. When no dotted lines exist, the action is local. Two types of dotted lines are used:

   1) Long, crossing the service provider area to reach the remote user, defining a direct action on the remote entity

   2) Short, beginning or ending at the middle of the service provider area. This defines an action to or from the Bus Arbitrator.

5.8 Type 8: Additional conventions

none

6 Type 1: Overview of the Data Link Service

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

d) Quality of Service (QoS) selection. The DLS provides DLS-users with a means to request and to agree upon a quality of service for the transfer of data. QoS is specified by means of QoS parameters representing aspects such as mode of operation, transit delay, accuracy and reliability.

e) Addressing. The DLS allows the DLS-user to identify itself and to specify the DLSAPs between which a DLC is to be established. DL-addresses have only regional significance within a specific DL-segment. Extended DL-addresses have only regional significance within a specific DL-subsystem over a set of bridged DL-segments. Therefore, it is not appropriate to define a global addressing structure.

   NOTE The DLS is required to differentiate between the individual systems that are physically or logically connected to a 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.

6.1.1 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 **path**s.

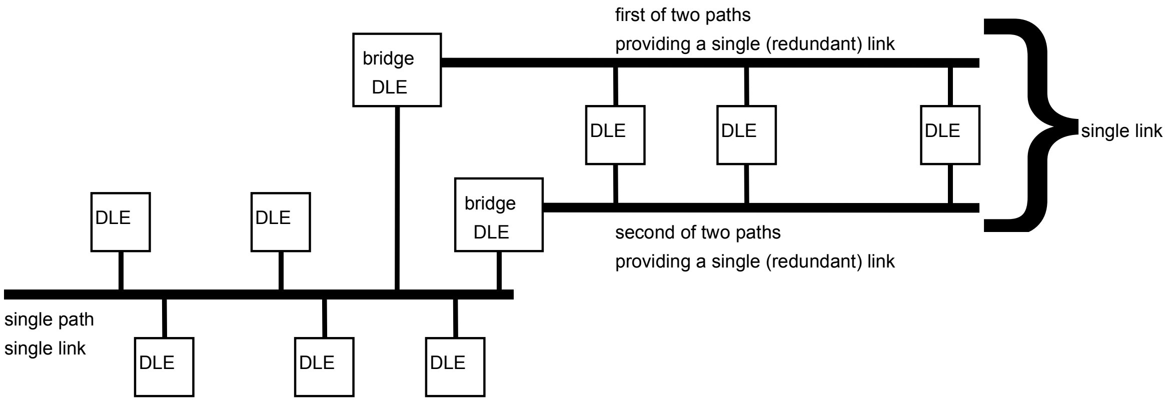


Figure 4 – Example of paths, links, bridges, and the extended link

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 4.

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.

6.1.2 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-service-access-points (DLSAPs), data-link-connection-end-points (DLCEPs), and data-link-entities (DLEs).

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

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

6.1.2.1 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 (see 3.3.6).

d) DLE-specific DLCEP-addresses;

e) other DLCEP-addresses;

f) specific aspects of a specific DLE; or

g) specific aspects of a group of DLEs.

6.1.2.2 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.

6.1.2.3 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.

6.2 Types and classes of Data Link Service

There are four types of DLS:

a) a DL(SAP)-address, queue and buffer management service (defined in Clause 7);
b) a connection-mode data transfer service with four classes of service (defined in Clause 8);
c) a connectionless-mode data transfer service with three classes of service (defined in Clause 9); and
d) a time and transaction scheduling service (defined in Clause 10).

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 11.

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

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

Static QoS attributes are selected once for an entire type of DLS. **Dynamic** QoS attributes are selected independently at each 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 part of IEC 61158.

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.

6.3.1 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.

6.3.2 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.

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

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

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

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

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

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

1) **ORDINARY** — each DLPDU 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.

6.3.4 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 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 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 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.

6.3.5 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.

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 5:

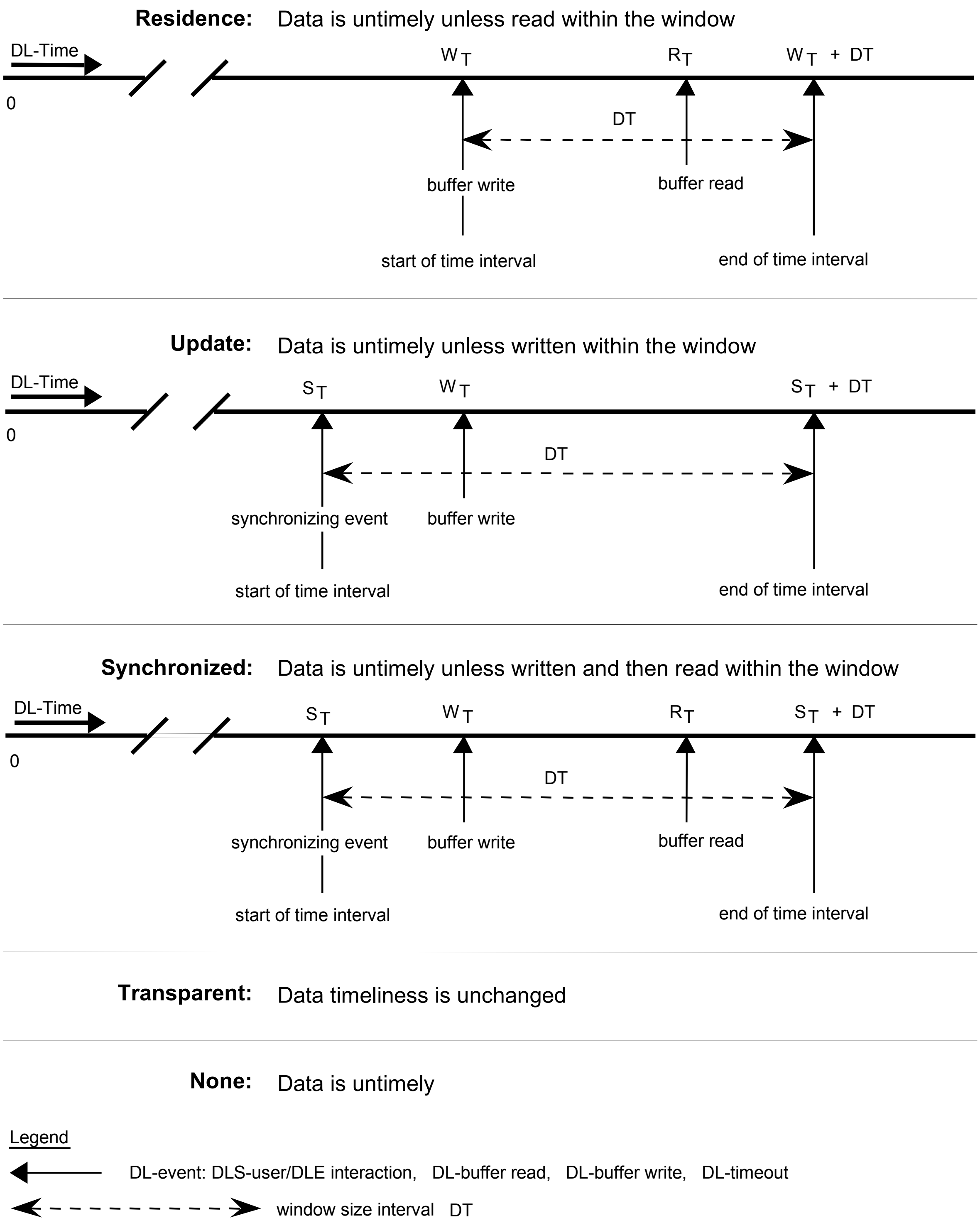


Figure 5 – Types of DL-timeliness
In terms of elapsed DL-time and events at the assessing DLCEP

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);

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

NOTE 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);

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

NOTE 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);

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

NOTE 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 DL-time is described in Clause 10.

7 Type 1: DL(SAP)-address, queue and buffer management Data Link Service

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

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

a) A means for creating and deleting a retentive buffer, 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.

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

This part of IEC 61158 uses the abstract model for a layer service defined in ISO/IEC 10731, Clause 5, 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.

7.3 Sequence of primitives at one DLSAP

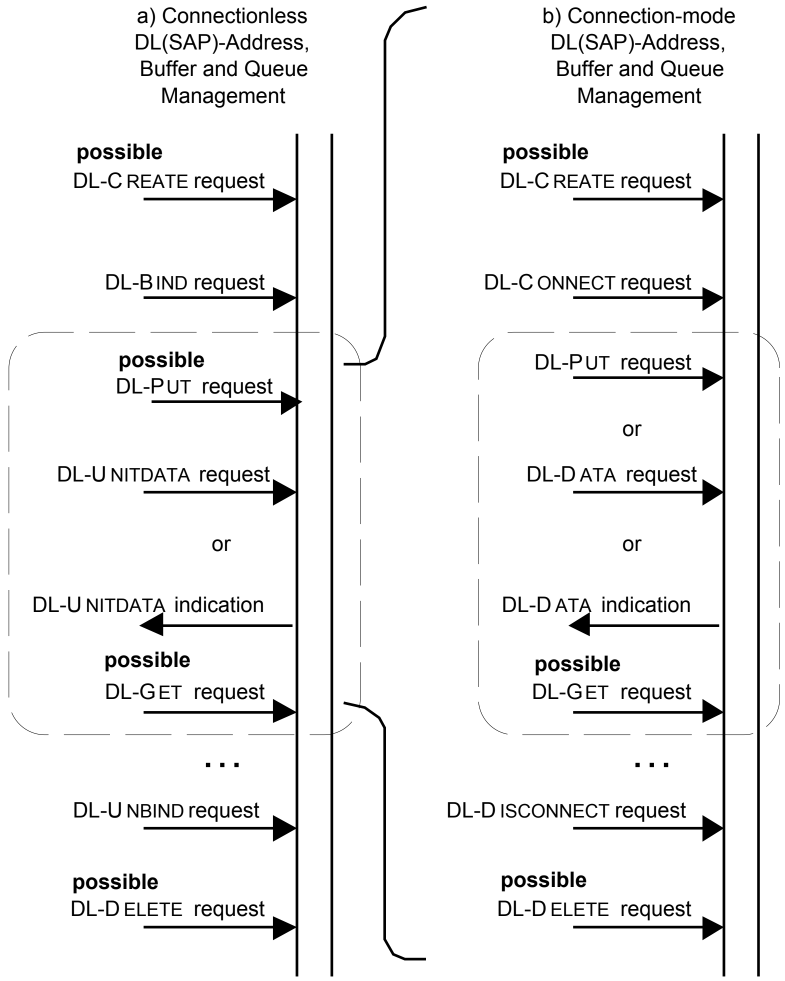
7.3.1 Constraints on sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in 7.4 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 6.

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

Service	Primitive	Parameter
Buffer or queue creation	DL-CREATE request	(*in* Buffer-or-queue DLS-user-identifier, Queuing policy, Maximum DLSDU size, *out* Status, Buffer-or-queue DL-identifier)
Buffer or queue deletion	DL-DELETE request	(*in* Buffer-or-queue DL-identifier, *out* Status)
DL(SAP)-address activation	DL-BIND request	(*in* 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, *out* Status, DL(SAP)-address DL-identifier)
DL(SAP)-address deactivation	DL-UNBIND request	(*in* DL(SAP)-address DL-identifier)
Update buffer	DL-PUT request	(*in* Buffer DL-identifier, DLS-user-data, DLS-user-data-timeliness, *out* Status)
Copy buffer or dequeue	DL-GET request	(*in* Buffer-or-queue DL-identifier, *out* 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 6 – Sequence of primitives for the DL(SAP)-address, queue and buffer management DLS

7.4 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 7.) 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.

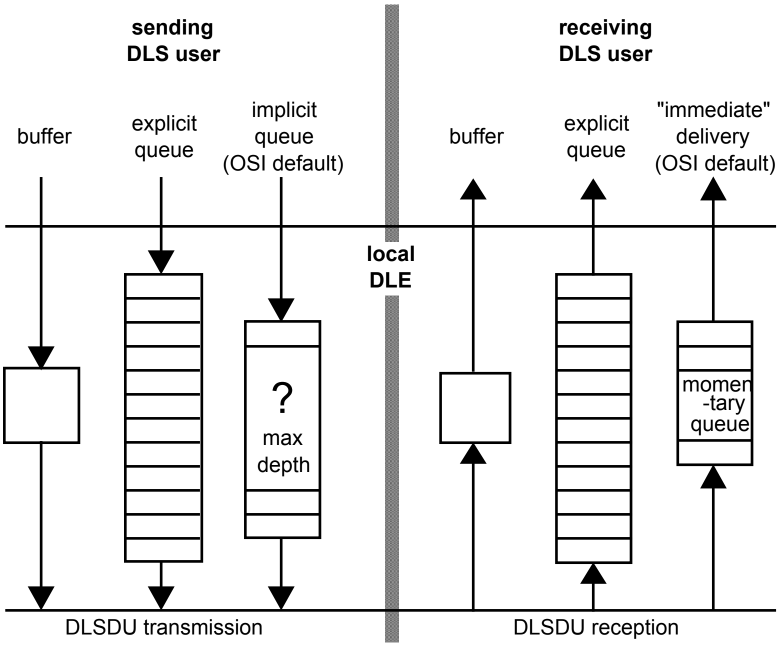


Figure 7 – Supported methods of data management for transmission and delivery

7.4.1 Create

7.4.1.1 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.

7.4.1.2 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

7.4.1.2.1 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.

7.4.1.2.2 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.

7.4.1.2.2.1 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.

7.4.1.2.3 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.

7.4.1.2.4 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 clause of IEC 61158-3.

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.

7.4.1.2.5 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.

7.4.1.3 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 6.

7.4.2 Delete

7.4.2.1 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.

7.4.2.2 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

7.4.2.2.1 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.

7.4.2.2.2 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 clause of IEC 61158-3.

7.4.2.3 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 6.

7.4.3 Bind

7.4.3.1 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.

7.4.3.2 Types of parameters

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

7.4.3.2.1 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.

7.4.3.2.2 DL(SAP)-address

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

7.4.3.2.3 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 4 – DL(SAP)-address-management bind primitive and parameters

DL-BIND Parameter name	Request input	Request 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

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.

7.4.3.2.3.1 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 9.

7.4.3.2.3.2 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.

7.4.3.2.4 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 6.3.1.

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

d) explicit bindings to a buffer are permitted;

e) explicit bindings to a queue are permitted; and

f) 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 DLSDUs, and can choose when to process those DLSDUs.

7.4.3.2.4.1 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.

7.4.3.2.4.2 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.

7.4.3.2.4.3 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.

7.4.3.2.5 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 DLSDUs 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.

7.4.3.2.5.1 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.

7.4.3.2.5.2 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.

7.4.3.2.5.3 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.

7.4.3.2.6 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 6.3. 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.

7.4.3.2.6.1 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.

7.4.3.2.6.2 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.

7.4.3.2.6.3 DLPDU authentication

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

7.4.3.2.6.4 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.

7.4.3.2.7 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 clause of IEC 61158-3.

7.4.3.2.8 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.

7.4.3.3 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 6.

7.4.4 Unbind

7.4.4.1 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.

7.4.4.2 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

7.4.4.2.1 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.

7.4.4.3 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 6.

7.4.5 Put

7.4.5.1 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.

7.4.5.2 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	Request output
Buffer DL-identifier	M	
DLS-user-data	U	
DLS-user-data-timeliness	U	
Status		M

7.4.5.2.1 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).

7.4.5.2.2 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.

7.4.5.2.3 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.

7.4.5.2.4 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 clause of IEC 61158-3.

7.4.5.3 Sequence of primitives

The sequence of primitives in using a buffer is defined in the time sequence diagram of Figure 6.

7.4.6 Get

7.4.6.1 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 7.4.5.2.1, 8.3.2.9.1a), and 9.5.2.2.3) 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.

7.4.6.2 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

DL-GET Parameter name	Request input	Request 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

7.4.6.2.1 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).

7.4.6.2.2 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 clause of IEC 61158-3.

7.4.6.2.3 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.

7.4.6.2.4 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.

7.4.6.2.4.1 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.

7.4.6.2.4.2 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.

7.4.6.2.4.3 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.

7.4.6.2.4.4 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.

7.4.6.2.5 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).

7.4.6.2.6 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.

7.4.6.2.6.1 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.

7.4.6.2.6.2 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.

7.4.6.2.6.3 Time-of-production

This parameter is present only when

a) the sending DLCEP specified TRUE for its time-of-production QoS parameter (see 8.3.2.10.1.4); and

b) the value of the sender-and-remote-DLE-timeliness parameter (see 7.4.6.2.6.2) 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.

7.4.6.3 Sequence of primitives

The sequence of primitives in using a buffer or queue is defined in the time sequence diagram of Figure 6.

8 Type 1: Connection-mode Data Link Service

8.1 Facilities of the connection-mode Data Link Service

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

a) A means to establish (see Figure 8) either

   1) a peer DLC between two DLS-users for exchanging DLSDUs between the two DLS-users, or

   NOTE 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 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.

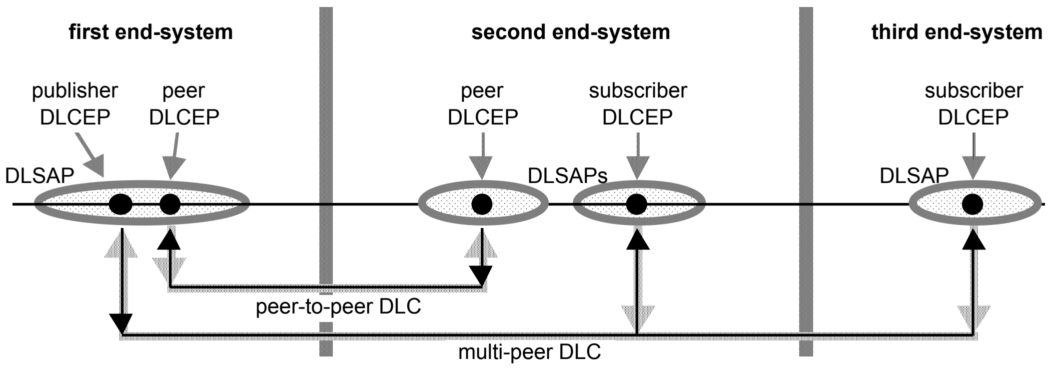


Figure 8 – Peer-to-peer and multi-peer DLCs and their DLCEPs

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 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 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 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 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 8.3.2.2). 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 True connectionless services also are available, independent of these four classes. See Clause 9.

8.2 Model of the connection-mode Data Link Service

This clause of IEC 61158 uses the abstract model for a layer service defined in ISO/IEC 10731, 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.

8.2.1 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 8.3.2.10 and the scheduling primitives of 10.5.2 and, 10.5.3. 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 5.2.2.)

8.2.2 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 7.

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 8.4. 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.

8.2.2.1 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 9).

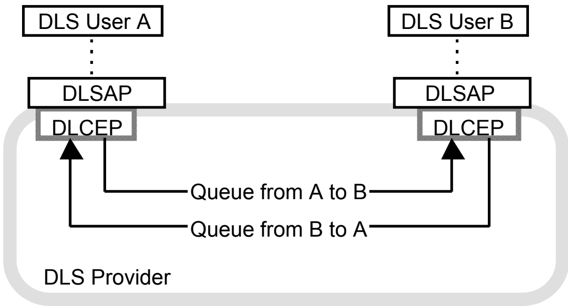


Figure 9 – OSI abstract queue model of a peer DLC between a pair of DLS-users

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 (8.2.2.4); 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 8.2.2.3).
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.

8.2.2.2 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.

8.2.2.3 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

Key

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 shall, after some unspecified time, perform all the permitted deletions.

8.2.2.4 Peer DLC / DLCEP reset

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

a) It cannot be removed from an abstract queue by a DLS-user.

b) An abstract queue appears empty to a DLS-user when a synchronization mark object is the next object in the queue.

c) A synchronization mark can be destroyed by a disconnect object (see Table 9).

d) 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:

1) 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.

2) A reset procedure initiated by the DLS-user is represented by the addition, by the DLS-provider, of objects into both abstract queues:

   i) from the reset requestor to the peer DLS-user of a reset object; and

   ii) 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 8.7.3.3).

8.2.2.5 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.

8.2.3 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 7.

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 8.4. 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.

8.2.3.1 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 10).

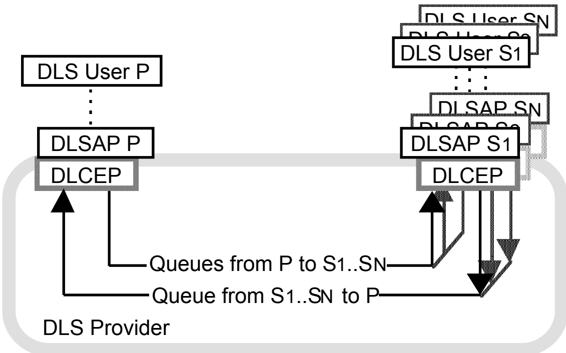


Figure 10 – OSI abstract queue model of a multi-peer DLC between a publishing DLS-user and a set of subscribing DLS-users

Each OSI abstract queue represents one direction of transfer. The ability of a subscribing DLS-user to remove objects from an abstract queue will be determined by the behavior of the publishing DLS-user. The ability of a publishing DLS-user to remove objects from an abstract queue will be determined by the aggregate behavior of the set of the publishing DLS-user and the subscribing DLS-users.

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

a) a connect object, representing a DL-CONNECT primitive and its parameters; and

b) a data object, representing a DL-DATA primitive and its parameters.

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

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 publishing DLS-user may place a connect object in either a single abstract queue or simultaneously in the entire set of publisher-to-subscriber abstract queues. The publisher places all other objects simultaneously in the entire set of publisher-to-subscriber abstract queues. Each subscriber places all objects in the single subscribers-to-publisher abstract queue.

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; and
2) 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 8.2.3.3).
v) An abstract queue has a limited capacity, but this capacity is not necessarily either fixed or determinable. When the sending DLS-user places a new object in an abstract queue that has reached its capacity, the oldest object in the abstract queue is lost.
vi) Depending on the multi-peer DLC's QoS, the abstract queue may occasionally duplicate or lose data objects.

8.2.3.2 Multi-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 multi-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 multi-peer DLC, the abstract queues remain associated with the multi-peer DLC until a disconnect object representing a DL-DISCONNECT primitive is either entered into or removed from the abstract queue.

The publishing DLS-user, which initiates a multi-peer DLC establishment by entering a connect object representing a DL-CONNECT request primitive into each of the separate publisher-to-subscriber abstract queues, is not allowed to enter any other object, other than a disconnect object, into the abstract queues until after the connect object representing the DL-CONNECT confirm primitive has been removed from the merged subscribers-to-publisher abstract queue. A disconnect object can be entered at any time.

A subscribing DLS-user, which initiates a multi-peer DLCEP establishment by entering a connect object representing a DL-CONNECT request primitive into the subscribers-to-publisher abstract queue, 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 its publisher-to-subscriber abstract queue. In the subscriber-specific abstract queue from the publisher to that subscriber, objects other than a disconnect object can be entered only after the publishing DLS-user has entered a connect object representing a DL-CONNECT response primitive. A disconnect object can be entered at any time.

While the multi-peer DLC exists between the publisher and a specific subscriber, the properties exhibited by the two abstract queues between them represent the agreements concerning QoS reached between the publishing DLS-user, that subscribing DLS-user, and the DLS-provider during the DLCEP establishment procedure.

8.2.3.3 Data transfer

Flow control on the multi-peer DLC is represented in this abstract queue model by the management of the abstract queue capacity, allowing objects to be added to the abstract queues. The addition of an object may prevent the addition of a further object, or may cause the loss of the first data object in the abstract queue. The addition of a disconnect object to the abstract queue may cause the loss of all prior objects still in the abstract queue. The ordering of objects in the abstract queue is identical to that implied by Table 9.

Whether the DLS-provider performs actions resulting in deletion or not will depend upon the behavior of the multi-peer DLC's users. With few exceptions, if a subscribing DLS-user does not remove objects from an abstract queue at the same long-term rate as the publishing DLS-user places them in the abstract queue, then some objects will be lost.

8.2.3.4 Multi-peer DLC/DLCEP reset

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

a) Initiation of a reset procedure by the DLS-provider is represented by the introduction of a reset object into one or more abstract queues.
b) A reset procedure initiated by the publishing DLS-user is represented by the addition, by the DLS-provider, of a reset object into each publisher-to-subscriber abstract queue.
c) A reset procedure initiated by a subscribing DLS-user is represented by the addition, by the DLS-provider, of a reset object into the subscribers-to-publisher abstract queue at that subscriber.

NOTE Restrictions on the issuance of certain other types of DL-primitives are associated with the initiation of a reset procedure. These restrictions will result in constraints on the entry of certain object types into the abstract queue until the reset procedure is completed (see 8.7.3.3).

8.2.3.5 Multi-peer DLC subscriber query

At any time during the existence of a multi-peer DLC, its publishing DLS-user can query whether there are any subscribing DLS-users.

NOTE This query does not result in the identification of any of these receiving DLS-users.

This query occurs within the scope of the multi-peer DLC, and thus cannot precede DLC establishment, nor can either the query or its confirmation follow DLC release. In all other respects, this query is asynchronous to the other activities of the multi-peer DLC.

8.2.3.6 Multi-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 DLCEP, and possibly a DLC, release procedure. The release procedure may be destructive with respect to other objects in the queues. The release procedure eventually results in the emptying of the abstract queues and the disassociation of the abstract queues from the multi-peer DLC. For a release initiated by a subscribing DLS-user, only the publisher-to-subscriber abstract queue associated with that subscribing DLS-user is emptied and dissociated from the multi-peer DLC; the abstract queues associated with other DLS-users are unaffected.

The insertion of a disconnect object may also represent the rejection of a multi-peer DLC/DLCEP establishment attempt or the failure to complete multi-peer DLC/DLCEP establishment.

8.3 Quality of connection-mode service

The term "Quality of Service" (QoS) refers to certain characteristics of a DLC as observed between the connection end-points. QoS describes aspects of a DLC which are attributable solely to the DLS-provider. QoS can only be properly determined when DLS-user behavior does not constrain or impede the performance of the DLS.

Once a DLC is established, the DLS-users at the DLCEPs have the same knowledge and understanding of the actual QoS of the DLC.

8.3.1 Determination of QoS for connection-mode service

QoS is described in terms of QoS parameters. These parameters give DLS-users a method of specifying their needs, and give the DLS-provider a basis for protocol selection. In this clause of IEC 61158-3, all QoS parameters are selected on a per-connection basis during the establishment phase of a DLCEP, or of a DLC and DLCEP.

Some of the DLC's QoS parameters are defined in 6.3; others are defined in 8.3.2. The selection procedures for these parameters are described in detail in 8.5.2.3. Once the DLC is established, throughout the lifetime of the DLC, the agreed QoS values are not reselected at any point in time.

8.3.2 Definition of QoS parameters

QoS attributes are as follows:

8.3.2.1 DLCEP class

Each DLC/DLCEP establishment request specifies the class of the DLCEP. The three choices for DLCEP class are:

a) **PEER** — the DLS-user can exchange DLSDUs with one other peer DLS-user;

b) **PUBLISHER** — the DLS-user can send DLSDUs to a set of zero or more associated subscribing DLS-users, and may be able to receive DLSDUs from any of those subscribing DLS-users; or

c) **SUBSCRIBER** — the DLS-user can receive, and request, DLSDUs from the associated publishing DLS-user, and may be able to send DLSDUs to that publishing DLS-user.

The default QoS value for the DLCEP class is PEER.

8.3.2.2 DLCEP data delivery features

Both members of a peer DLC, or the publishing DLS-user of a multi-peer DLC, specify the data delivery features of the DLC's DLCEP(s). The available choices of DLCEP data delivery features depends on the class of DLS provided (see Table 10). The five choices for DLCEP data delivery features, and their effects, are:

a) **CLASSICAL** — the DLS-user can send data which will be delivered without loss, duplication or misordering to the receiving DLS-user(s). All relevant DLS-users will be notified of any loss of synchronization on the DLC.

b) **DISORDERED** — the DLS-user can send data which will be delivered immediately upon receipt to the receiving DLS-user(s), without duplication but potentially in a different order than the order in which it was originally sent. All relevant DLS-users will be notified of any unrecoverable loss of DLS-user-data or loss of synchronization on the DLC.

c) **ORDERED** — the DLS-user can send data which will be delivered immediately upon receipt to the receiving DLS-user(s), without duplication or misordering, but with potential loss of some DLS-user-data. Loss of DLS-user-data will not be reported, and recovery of DLS-user data lost prior to the last-reported DLS-user data will not be attempted.

   NOTE Recovery of lost DLS-user data subsequent to that last reported is permitted but is not required.

d) **UNORDERED** — the DLS-user can send data which will be delivered immediately upon receipt to the receiving DLS-user(s). Loss, duplication and misordering of DLS-user-data will not be detected or reported. No attempt will be made by the DLS-provider to recover from such events.

e) **NONE** — the DLS-user cannot send data in this direction of data transfer.

There are four classes of connection-mode DLS:

A CLASSICAL, ORDERED and UNORDERED peer and multi-peer DLCs are all supported; DISORDERED peer and multi-peer DLCs may be supported;

B only ORDERED and UNORDERED peer and multi-peer DLCs, and CLASSICAL peer DLCs, are supported; CLASSICAL and DISORDERED multi-peer DLCs are not supported; DISORDERED peer DLCs may be supported;

C only ORDERED and UNORDERED peer and multi-peer DLCs are supported; CLASSICAL and DISORDERED peer and multi-peer DLCs are not supported; and

D only UNORDERED peer and multi-peer DLCs are supported; ORDERED, CLASSICAL and DISORDERED peer and multi-peer DLCs are not supported.

Table 10 – Attributes and class requirements of DLCEP data delivery features

DLCEP data delivery features	action on lost DLSDUs	action on duplicate DLSDUs	action on mis-ordered DLSDUs	permitted forms of buffer and queue use	support on a peer-to-peer DLC (note 4)	support on a publisher-to-subscribers DLC (note 4)	support on a subscribers-to-publisher DLC (note 4)
NONE	none	none	none	none	Mandatory in all classes	Mandatory in all classes	Mandatory in all classes
UNORDERED (default)	not detected	not detected	not detected (note 2)	buffer ⇒ buffer buffer ⇒ queue queue ⇒ queue (note 3)	Mandatory in all classes	Mandatory in all classes	Optional in all classes
ORDERED	detected	dis-carded	dis-carded	buffer ⇒ buffer buffer ⇒ queue queue ⇒ queue (note 3)	Mandatory in classes A,B,C	Mandatory in classes A,B,C	not possible
CLASSICAL	recovered (note 1)	dis-carded	recovered: delivered in order	queue ⇒ queue	Mandatory in classes A,B	Mandatory in class A	not possible
DISORDERED	recovered (note 1)	dis-carded	delivered as received	queue ⇒ queue	Optional in classes A,B	Optional in class A	not possible

NOTE 1 An unrecoverable data loss causes the DLE to initiate a reset procedure.

NOTE 2 DL-communication topologies in which misordering is possible cause UNORDERED DLCs to be upgraded to ORDERED during DLC negotiation.

NOTE 3 Queue-to-queue use provides a QoS similar to connectionless service.

NOTE 4 The DLE should support all DLCEP data-delivery features required of its DLS class. (The DLE classes are introduced in 8.1.) If the DLE does not support DISORDERED but does support CLASSICAL, then the DLE upgrades the DLCEP data delivery features from DISORDERED to CLASSICAL. In all other cases, if the DLE does not support a requested DLCEP data delivery feature which is optional for its DLS class, then the DLE rejects the DLC/DLCEP establishment request.

On a peer DLC, the QoS value for the DLCEP data delivery features may be chosen independently for each direction of data transfer. The default QoS value in each direction for the DLCEP data delivery features is UNORDERED.

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

Selection of any of the following QoS attribute values may cause the DLS-provider to upgrade DLCEP data delivery features from UNORDERED to ORDERED:

1) specification of a maximum DLSDU size (see 8.3.2.8) greater than the maximum number of DLS-user octets that can be conveyed by a DLPDU of the specified DLL priority (see 8.3.2.3); or
2) specification of DL-timeliness criteria (see 8.3.2.10) other than none for transmissions from a peer or publisher DLCEP; or
3) selection of a DLCEP data delivery feature other than unordered or none for the other (reverse) direction of a peer DLCEP.

This upgrade may also occur when multiple active paths can exist between the DLC's participating DLS-providers.

The DLS provider may also upgrade from DISORDERED to CLASSICAL during this negotiation process.

No other negotiation of either of the DLCEP's data delivery features is permitted.

Table 10 summarizes these DLCEP data delivery features, their attributes, and the DLS classes in which they are available.

8.3.2.3 DLL priority

This parameter is defined and its default value specified in 6.3.1 and 7.4.3.2.6.1.

NOTE DLC initiation and DLC termination DLPDUs are sent at TIME-AVAILABLE priority on the link, but are restricted to convey no more DLS-user-data than is permitted for NORMAL priority DLPDUs.

8.3.2.4 DLL maximum confirm delay

This parameter is defined and its default values specified in 6.3.2 and 7.4.3.2.6.2. Failure to complete a DL-CONNECT or DL-RESET request within the specified interval results in a DLS-provider initiated release of the DLCEP, and possibly of the DLC.

NOTE For DLEs which do not support a time resolution of 1 ms, the requested time interval may be rounded up to the next-greatest multiple of the resolution which the DLE does support.

This parameter provides user-specifiable bounds on the extent of DLC error recovery effort.

8.3.2.5 DLPDU authentication

This parameter is defined and its default value is specified in 6.3.3 and 7.4.3.2.6.3. The DLS-user may override that default value when establishing a DLCEP.

8.3.2.6 Residual activity

Each DLC/DLCEP establishment request, and each response, specifies 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.

NOTE This background transmission is known as DLC **residual activity**.

The default value for this parameter is FALSE, unless overridden by DL-management or by a DLPDU-authentication attribute of MAXIMAL.

8.3.2.7 DL-scheduling-policy

This parameter is defined and its default value is specified in 6.3.4 and 7.4.3.2.6.4. The DLS-user may override that default value when establishing a DLCEP. When the DLCEP is bound as sender to a buffer, then the DLS-user should ensure that this parameter has the value EXPLICIT.

8.3.2.8 Maximum DLSDU sizes

Each DLC/DLCEP establishment request, and each response, specifies an upper bound on the size (in octets) of DLSDUs which will be offered for transmission, and an upper bound on the size of DLSDUs which are acceptable for reception. For peer DLCs, the maximum DLSDU size permitted will be the smallest of that offered by the sender, that permitted by the receiver, and that permitted by DLL management. For multi-peer DLCs, the maximum DLSDU size permitted will be the smaller of that offered by the publisher and that permitted by the publisher's DLL management. For subscribers of multi-peer DLCs, the DLC will be refused by the DLS-provider if the maximum DLSDU size established by the publisher is larger than

a) that permitted by the subscriber; or

b) that permitted by the subscriber's DLL management.

The default value for both the sender's and receiver's maximum DLSDU size is the maximum number of DLS-user octets which can be carried by a single DLPDU of the specified DLL Priority (see 6.3.1). The DLS-provider should always support that DLSDU size.

NOTE These negotiation rules do not preclude either derivative protocol specifications or real implementations from using a fixed, small set of sizes when allocating buffers or entries in a queue.

8.3.2.9 DLCEP buffer-or-queue bindings

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

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

a) One queue or retentive buffer can be bound to a DLCEP to convey DLSDUs from the DLS-user to the DLS-provider.

b) One queue or retentive buffer can be bound to a DLCEP to convey DLSDUs from the DLS-provider to the DLS-user.

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

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

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

8.3.2.9.1 Binding to a retentive buffer

When a sending retentive buffer is bound to a DLCEP by a DLS-user:

a) A DL-PUT request primitive overwrites the buffer with a DLSDU.

   NOTE After creation, the buffer is empty. After it has once been written, a buffer bound to a DLCEP can never again be empty.

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

c) A DL-COMPEL request primitive causes the transmission, at the first opportunity, of the contents of the buffer at the moment of transmission; the primitive does not itself specify a DLSDU. As a consequence, the only meaningful scheduling policy for buffers is EXPLICIT.

When a receiving retentive buffer is bound to a DLCEP by a DLS-user:

d) A DL-BUFFER-RECEIVED indication primitive notifies the DLS-user of the overwriting of the buffer by a newly-received DLSDU; the primitive does not itself specify a DLSDU.
e) A DL-GET request primitive copies the DLSDU from the buffer.

8.3.2.9.2 Binding to a FIFO queue

When a sending FIFO queue of maximum depth *K* is bound to a DLCEP by a DLS-user:

a) A DL-PUT request primitive is not permitted.
b) A DL-DATA request primitive attempts to append a DLSDU to the queue, but fails if the queue already contains *K* DLSDUs. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

When a receiving FIFO queue of maximum depth *K* is bound to a DLCEP by a DLS-user

c) A DL-GET request primitive attempts to remove a DLSDU from the queue, but fails if the queue is empty.
d) A DL-DATA indication primitive notifies the receiving DLS-user of the result of appending a newly-received DLSDU to the receive queue; the primitive does not itself specify a DLSDU.

8.3.2.9.3 Default bindings

When these binding options are not specified, the conventional implicitly-queued sending and direct receiving interfaces between DLS-user and DLS-provider are employed, and each associated DL-DATA primitive contains a DLSDU:

a) DL-Put and DL-Get request primitives are not permitted.
b) A DL-Data request primitive by the sending DLS-user attempts to append a DLSDU to the implicit queue, but fails if the queue is full. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.
c) A DL-Data indication primitive notifies a receiving DLS-user of a newly-received DLSDU, and conveys that DLSDU to the DLS-user. No apparent queuing is provided within the DLL.

8.3.2.10 DLCEP timeliness

Each DLCEP establishment request, and each response, can specify DL-timeliness QoS criteria which are to apply to information sent or received at that DLCEP. (See 6.3.5.)

8.3.2.10.1 Sender timeliness

This set of sub-parameters applies only when the buffer-and-queue-bindings-as-sender specify a retentive buffer.

8.3.2.10.1.1 DL-timeliness class

The DLS-user should specify one of the four types of DL-timeliness specified in 6.3.5 — RESIDENCE, UPDATE, SYNCHRONIZED, or TRANSPARENT — or should specify NONE, indicating that DL-timeliness is not to be provided.

The default value for this parameter is NONE.

8.3.2.10.1.2 Time window size (ΔT)

When the value for the DL-timeliness-class parameter is RESIDENCE, UPDATE or SYNCHRONIZED, then the DLS-user should specify the applicable time window size (ΔT), with a permitted maximum of 60 s.

8.3.2.10.1.3 Synchronizing DLCEP

When the value for the DL-timeliness-class parameter is UPDATE or SYNCHRONIZED, then the DLS-user should specify the DL-identifier (see 8.5.2.1.2) of the DLCEP whose activity (DL-BUFFER-SENT indication or DL-BUFFER-RECEIVED indication (see 8.7.2) is to be used as the synchronizing activity for the timeliness computation. If the synchronizing DLCEP disconnects before the referencing DLCEP, then after that disconnection the resulting sender timeliness is FALSE.

8.3.2.10.1.4 Time of production

When the value for the DL-timeliness-class parameter is RESIDENCE, UPDATE, SYNCHRONIZED or TRANSPARENT, then the DLS-user may specify that the DL-time of DLS-user-data production is to be distributed to the consuming peer DLS-users to facilitate their assessment of DLS-user-timeliness. Permitted values for this parameter are TRUE and FALSE.

The default value for this parameter is FALSE.

8.3.2.10.1.5 Receiver timeliness

This set of sub-parameters applies only when the buffer-and-queue-bindings-as-receiver specify a retentive buffer.

8.3.2.10.1.6 DL-timeliness class

The DLS-user should specify one of the four types of DL-timeliness specified in 6.3.5 — RESIDENCE, UPDATE, SYNCHRONIZED, or TRANSPARENT — or should specify NONE, indicating that DL-timeliness is not to be provided.

The default value for this parameter is NONE.

8.3.2.10.1.7 Time window size (ΔT)

When the value for the DL-timeliness-class parameter is RESIDENCE, UPDATE or SYNCHRONIZED, then the DLS-user should specify the applicable time window size (ΔT), with a permitted maximum of 60 s.

8.3.2.10.1.8 Synchronizing DLCEP

When the value for the DL-timeliness-class parameter is UPDATE or SYNCHRONIZED, then the DLS-user should specify the DL-identifier (see 8.5.2.1.2) of the DLCEP whose activity (DL-BUFFER-SENT indication or DL-BUFFER-RECEIVED indication (see 8.7.2) is to be used as the synchronizing activity for the timeliness computation. If the synchronizing DLCEP disconnects before the referencing DLCEP, then after that disconnection the resulting receiver timeliness is FALSE.

8.4 Sequence of primitives

8.4.1 Concepts used to define the connection-mode DLS

The service definition uses the following concepts:

a) A DLC can be dynamically established (or terminated) between the DLS-users for the purpose of exchanging data. It is also possible statically to pre-establish an unordered or ordered DLC.
b) Associated with each DLC, certain measures of QoS are agreed between the DLS-provider and the DLS-users when the DLC is established.

c) The DLC allows transmission of DLS-user-data and preserves the data's division into DLSDUs:
   - The transmission of this data may be subject to flow control, depending on the DLCEP class and data delivery features.
   - The transmission and reception of this data may also be subject to timeliness assessments, which are combined with any previously-determined data timeliness to determine the overall timeliness of the DLS-user-data.
d) The DLC can be returned to a defined state, and the activities of either or both of the two DLS-users synchronized, by use of a Reset Service.
e) Failure to provide the requested service may be signaled to the DLS-user. There are three classes of failure:
   1) failures involving termination of the DLC;
   2) failures involving duplication, loss or disordering of user data, as permitted by the DLC's QoS, but without loss of the DLC; and

      NOTE This includes resetting of the DLC.
   3) failures to provide the requested QoS, but without loss of user data or loss of the DLC.

8.4.2 Constraints on sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in 8.5 through 8.7 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur. Other constraints, such as flow control of data, will affect the ability of a DLS-user or a DLS-provider to issue a primitive at any particular time.

The connection-mode primitives and their parameters are summarized in Table 11 and Table 12.

Table 11 – Summary of DL-connection-mode primitives and parameters (portion 1)

Phase	Service	Primitive	Parameter
DLC / DLCEP Establishment	DLC / DLCEP Establishment	DL-CONNECT request	(*in* DLCEP DLS-user-identifier, Called DL(SAP)-address (2) or Called DLCEP-address or UNKNOWN, Calling DLSAP-address DL-identifier or Calling DLCEP DL-identifier, optional Calling DLCEP-address, QoS parameter set, limited DLS-user-data, *out* DLCEP DL-identifier)
		DL-CONNECT indication	(*out* DLCEP DL-identifier, Called DL(SAP)-address DLS-user-identifier, Calling DLSAP-address, (2) QoS parameter set, limited DLS-user-data)
		DL-CONNECT response	(*in* DLCEP DLS-user-identifier, Responding DL(SAP)-address DL-identifier or Responding DLCEP DL-identifier, optional Responding DLCEP-address, (2) QoS parameter set, limited DLS-user-data)
		DL-CONNECT confirm	(*out* Responding DL(SAP)-address (2) or UNKNOWN, QoS parameter set, limited DLS-user-data)
		DL-CONNECTION-ESTABLISHED indication	(*out* DLCEP DLS-user-identifier)
DLC / DLCEP Release	DLC / DLCEP Release	DL-DISCONNECT request	(*in* DLCEP DL-identifier, Reason, limited DLS-user-data)
		DL-DISCONNECT indication	(*out* DLCEP-identifier-type, DLCEP DLS-user-identifier, DLCEP DL-identifier, Originator, Reason, limited DLS-user-data)

Table 12 – Summary of DL-connection-mode primitives and parameters (portion 2)

Phase	Service	Primitive	Parameter
Data Transfer	Normal Data Transfer	DL-DATA request	(*in* DLCEP DL-identifier, DLS-user-data)
		DL-DATA indication	(*out* DLCEP DLS-user-identifier, Queue DLS-user-identifier, DLS-user-data)
		DL-DATA confirm	(*out* Status)
		DL-BUFFER-SENT indication	(*out* DLCEP DLS-user-identifier, Buffer DLS-user-identifier)
		DL-BUFFER-RECEIVED indication	(*out* DLCEP DLS-user-identifier, Buffer DLS-user-identifier, Duplicated DLSDU)
		DL-GET-NEXT-SEQUENCE-NUMBER request	(*in* DLCEP DL-identifier, *out* Status, Sequence-number)
	DLC / DLCEP Reset / Resynchronize	DL-RESET request	(*in* DLCEP DL-identifier, Reason, limited DLS-user-data)
		DL-RESET indication	(*out* DLCEP DLS-user-identifier, Originator, Reason, limited DLS-user-data)
		DL-RESET response	(*in* DLCEP DL-identifier, limited DLS-user-data)
		DL-RESET confirm	(*out* limited DLS-user-data, Status)
		DL-RESET-COMPLETED indication	(*out* DLCEP DLS-user-identifier)
	Subscriber Query	DL-SUBSCRIBER-QUERY request	(*in* DLCEP DL-identifier)
		DL-SUBSCRIBER-QUERY confirm	(*out* Status)

NOTE 1 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, DL-buffer or DL-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, DL-buffer or DL-queue to the DLS-user at the DLS interface.

NOTE 2 If the DLS-user has issued a DL-Bind request for a DL(SAP)-address, then a parameter referencing that DL(SAP)-address can also take the form of a DL(SAP)-address DL-identifier.

NOTE 3 The method by which a confirm primitive is correlated with its corresponding preceding request primitive, or a response primitive is correlated with its corresponding preceding indication primitive, is a local matter.

8.4.2.1 Relation of primitives at the two DLC end-points

With few exceptions, a primitive issued at one DLCEP will have consequences at another DLCEP. The relations of primitives at one DLCEP to primitives at another DLCEP of the same DLC are defined in the appropriate subclause in 8.5 through 8.7, and are summarized in the diagrams of Figure 11 through Figure 16.

However, a DL-DISCONNECT request or indication primitive may terminate any of the other sequences before completion. A DL-RESET request or indication primitive may terminate a data transfer sequence before completion.

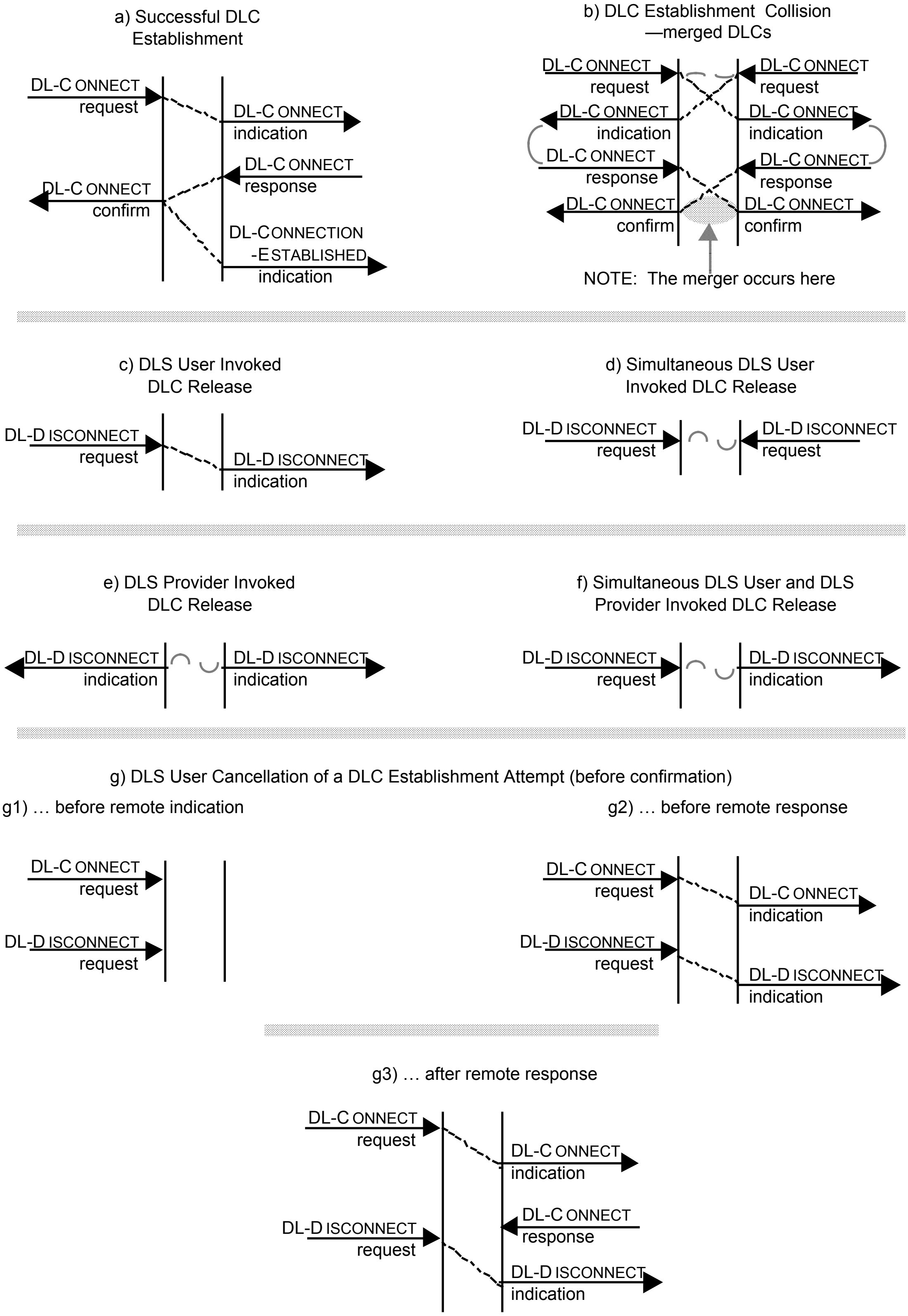


Figure 11 – Summary of DL-connection-mode service primitive time-sequence diagrams for peer DLCs (portion 1)

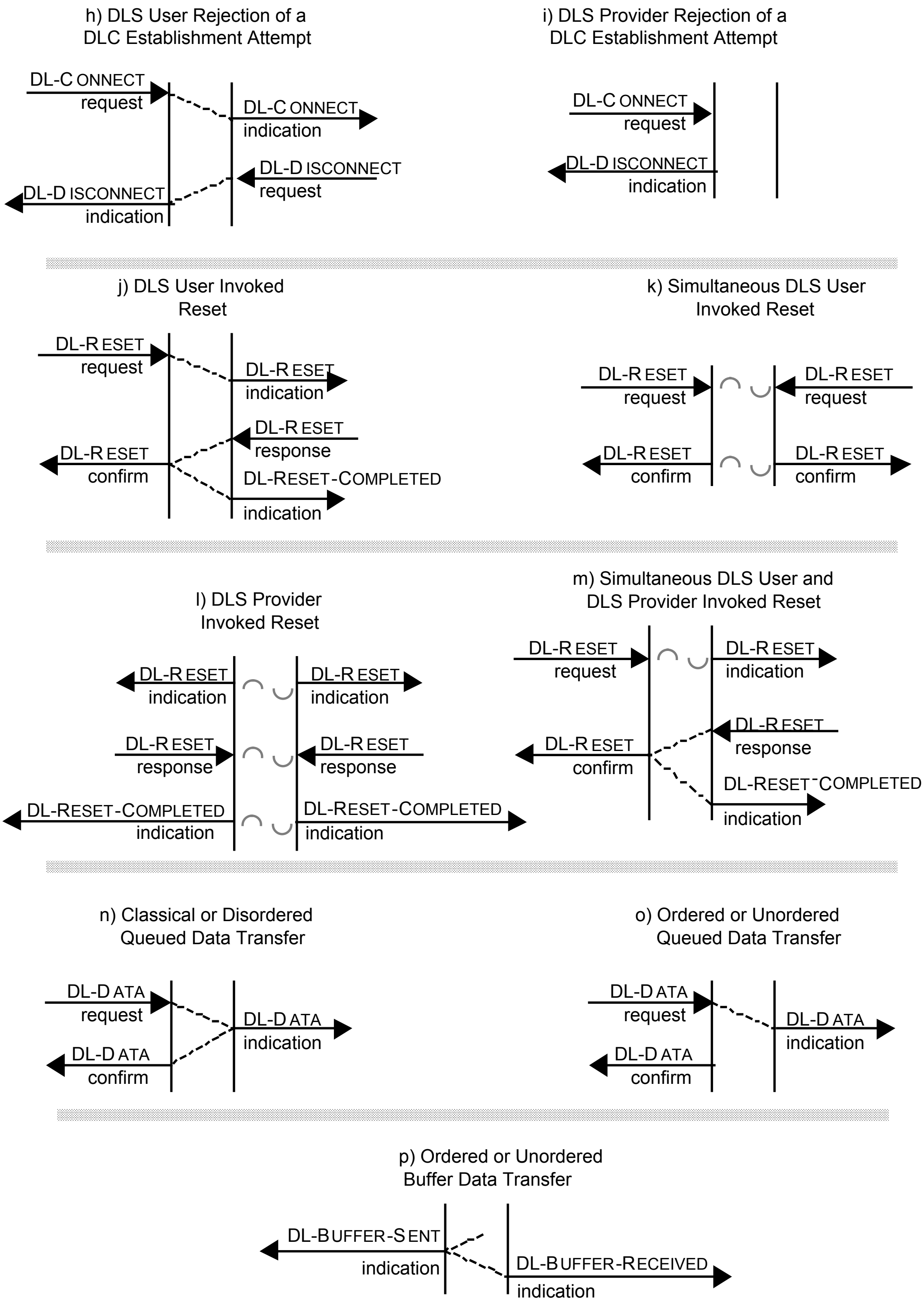


Figure 12 – Summary of DL-connection-mode service primitive time-sequence diagrams for peer DLCs (portion 2)

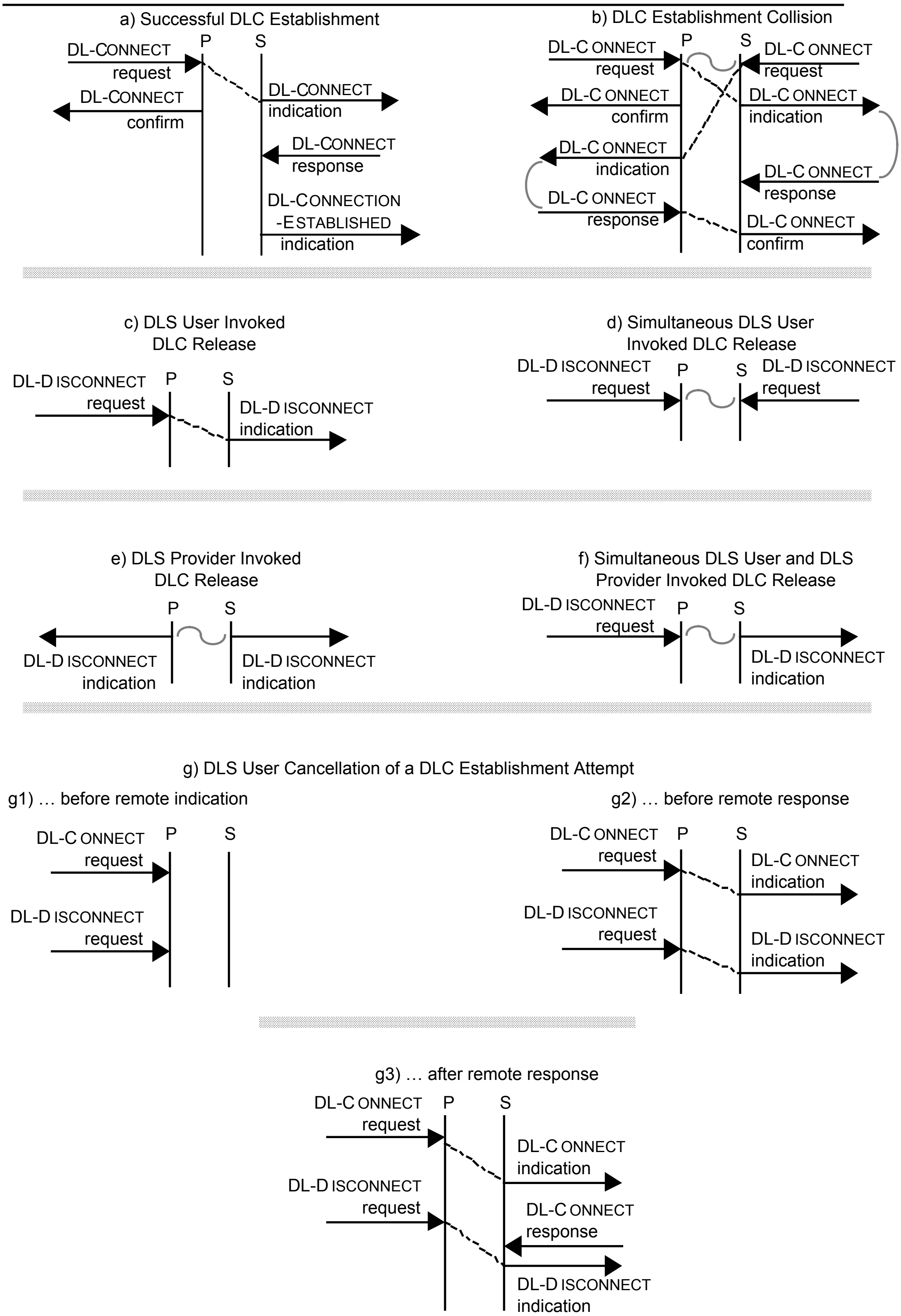


Figure 13 – Summary of DL-connection-mode service primitive time-sequence diagrams for publishers of a multi-peer DLC (portion 1)

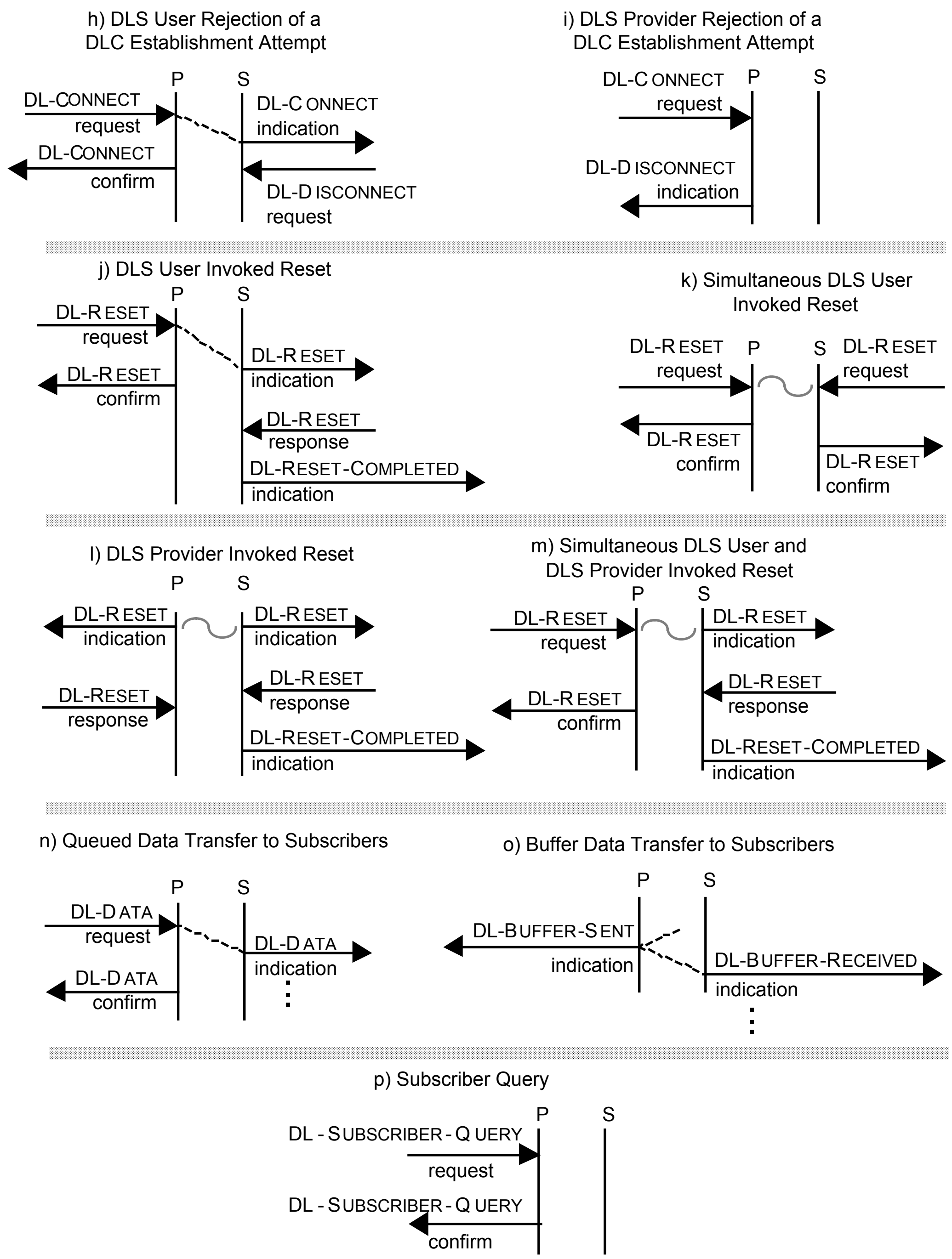


Figure 14 – Summary of DL-connection-mode service primitive time-sequence diagrams for publishers of a multi-peer DLC (portion 2)

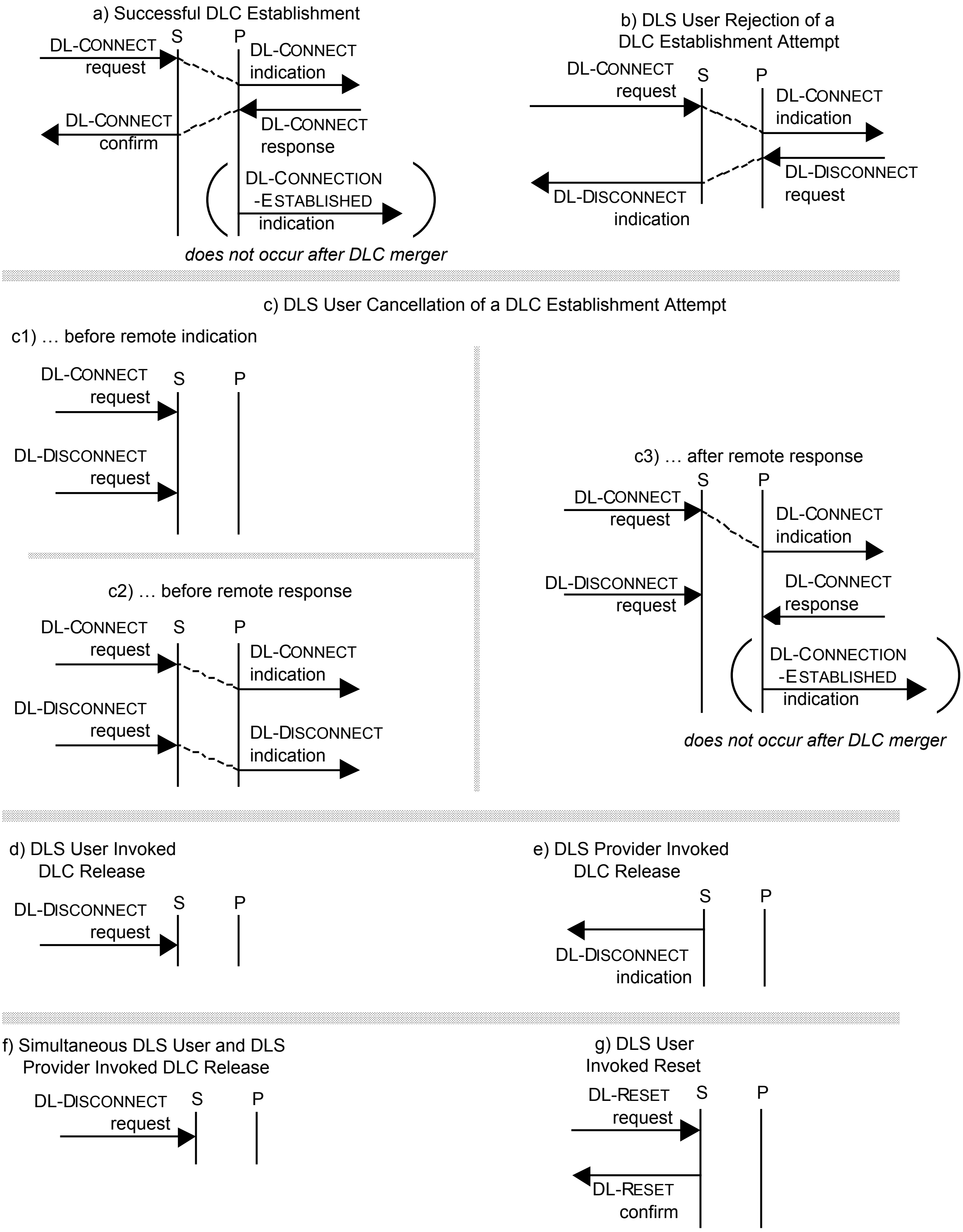


Figure 15 – Summary of additional DL-connection-mode service primitive time-sequence diagrams for a multi-peer DLC subscriber where the diagrams differ from the corresponding ones for a publisher (portion 1)

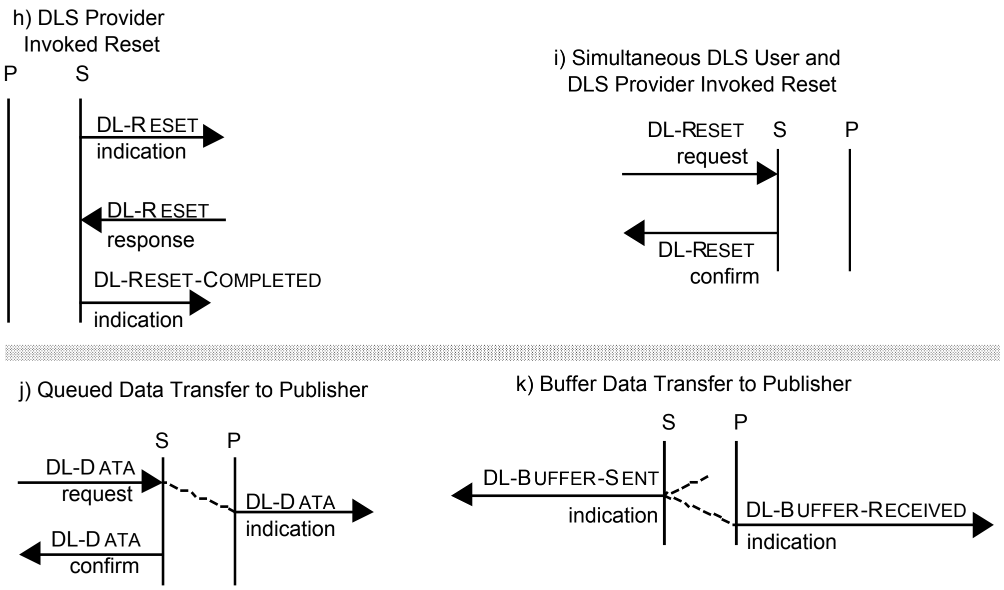


Figure 16 – Summary of additional DL-connection-mode service primitive time-sequence diagrams for a multi-peer DLC subscriber where the diagrams differ from the corresponding ones for a publisher (portion 2)

8.4.2.2 Sequence of primitives at one DLC end-point

The possible overall sequences of primitives at a DLCEP are defined in the state transition diagram, Figure 17. In the diagram:

a) DL-Disconnect stands for either the request or the indication form of the primitive in all cases.

b) The labeling of the states “local DLS-user initiated reset pending” (6) and “other reset pending” (7) indicate the party that started the local interaction, and does not necessarily reflect the value of the originator parameter.

c) The “idle state” (1) reflects the absence of a DLCEP. It is the initial and final state of any sequence, and once it has been re-entered the DLCEP is released.

d) The use of a state transition diagram to describe the allowable sequences of service primitives does not impose any requirements or constraints on the internal organization of any implementation of the service.

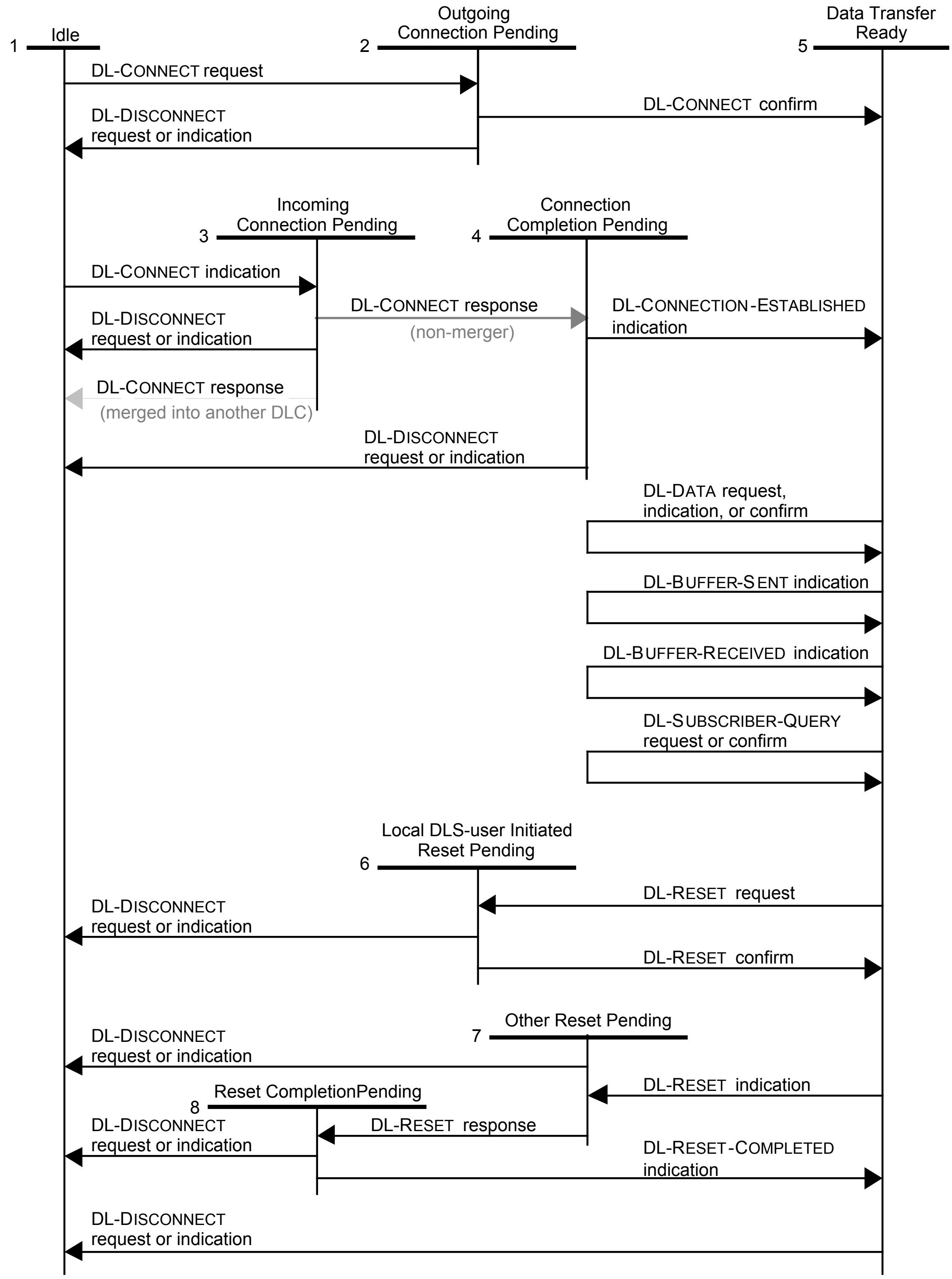


Figure 17 – State transition diagram for sequences of DL-connection-mode service primitives at a DLCEP

8.5 Connection establishment phase

8.5.1 Function

The DLC / DLCEP establishment service primitives can be used to establish a DLCEP, and possibly a DLC.

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

Simultaneous DL-CONNECT request primitives at the two DLSAPs may be merged into one DLC by the concurrently requesting-and-responding DLS-users as indicated in Figure 23 and Figure 24.

8.5.2 Types of primitives and parameters

Table 13 and Table 14 indicate the types of primitives and the parameters needed for DLC/DLCEP establishment.

Table 13 – DLC / DLCEP establishment primitives and parameters (portion 1)

DL-CONNECT Parameter name	Request input	Request output	Indication output	Response input	Confirm output
DLCEP DLS-user-identifier	M			M	
DLCEP DL-identifier		M	M		
Called address	M		M (=)		
Calling address	M		M (=)		
Responding address				M	M (=)
Calling DLCEP-address	CU			CU	
QoS parameter set					
DLCEP class	U		M (=)	U (1)	M (=)
DLCEP data delivery features					
from requestor to responder(s)	U		M (=, 2)	U (=, 2)	M (=)
from responder(s) to requestor	U		M (=, 2)	U (=, 2)	M (=)
DLL priority	U		M (=)	U (≤)	M (=)
Maximum confirm delay					
on DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY	U		M (=)	U	M (=)
on DL-DATA	U		M (=)	U	M (=)
DLPDU-authentication	U		M (≥, 3)	U	M (≥, 3)
Residual activity					
as sender	U		M (=)	U (≤, 4)	M (=)
as receiver	U		M (=)	U (≤, 4)	M (=)
DL-scheduling-policy	U			U	
Maximum DLSDU sizes (5)					
from requestor	CU		M (≤)	CU (≤)	M (=)
from responder	CU		M (≤)	CU (≤)	M (=)
Buffer-and-queue bindings (5)					
as sender	CU			CU	
as receiver	CU			CU	
Sender timeliness (5)					
DL-timeliness-class	CU		M (=)	CU	M (=)
Time window size (ΔT)	CU			CU	
Synchronizing DLCEP	CU			CU	
Time-of-production	CU		C(=)	CU	C (=)
Receiver timeliness (5)					
DL-timeliness-class	CU		M (=)	CU	M (=)
Time window size (ΔT)	CU			CU	
Synchronizing DLCEP	CU			CU	
DLS-user-data	U		M (=)	U	M (=)

NOTE 1 The DLCEP classes should match, Peer with Peer, and Publisher with Subscriber.

NOTE 2 The DLCEP data delivery feature UNORDERED may be upgraded to ORDERED, and DISORDERED may be upgraded to CLASSICAL.

NOTE 3 8.3.2.5 specifies that DLPDU-authentication may be upgraded from ORDINARY to SOURCE to MAXIMAL.

NOTE 4 8.3.2.6 specifies that a residual activity value of FALSE may be upgraded to TRUE.

NOTE 5 These parameters are conditional on the negotiated directions of data delivery, and in the case of timeliness, on the specified timeliness class.

NOTE 6 The method by which a confirm primitive is correlated with its corresponding preceding request primitive, or a response primitive is correlated with its corresponding preceding indication primitive, is a local matter.

Table 14 – DLC / DLCEP establishment primitives and parameters (portion 2)

DL-CONNECTION-ESTABLISHED Parameter name	Indication output
DLCEP DLS-user-identifier	M (1)

NOTE 1 The DLCEP DLS-user-identifier equals the DLS-user-identifier specified in the corresponding DL-CONNECT response primitive.

8.5.2.1 Local-view identifiers

8.5.2.1.1 DLCEP DLS-user-identifier

This parameter specifies a means of referring to the DLCEP in output parameters of other local DLS primitives which convey the name of the DLCEP from the local DLE to the local DLS-user. It is specified by the DLS-user on DL-CONNECT request and response primitives, and is used by the DLE to refer to the DLC-end-point in DLS output parameters of other DLC primitives.

The naming-domain of the DLCEP DLS-user-identifier is the DLS-user's local-view.

8.5.2.1.2 DLCEP DL-identifier

This parameter, which is returned by the DLS-provider on DL-CONNECT request and indication primitives, provides a local means of identifying a specific DLC-end-point in input parameters of other local DLS primitives which convey the name of the DLC-end-point from the local DLS-user to the local DLE.

The naming-domain of the DLCEP DL-identifier is the DL-local-view.

8.5.2.2 Addresses

The parameters that take addresses as values refer to either DL(SAP)-addresses or DLCEP-addresses. A DLCEP-address, which is used within a DL-protocol to identify a DLCEP, is structurally similar to a DLSAP-address, and is drawn from the same overall address space.

8.5.2.2.1 Called address

This parameter conveys an address identifying the remote DLSAP(s) at which the DLC is to be established. It should be either

a) a DLSAP-address;

b) a group DL-address;

NOTE If the called address for a peer or subscriber DLC is a group DL-address, then the DLC will be established with the DLSAP whose response is first received, and the others will be disconnected when and if they respond.

c) a DLCEP-address; or

NOTE Direct knowledge of a DLCEP-address can be obtained only through extra-protocol means. The ability to specify such an address supports migration from previous national standards and facilitates the use of simple pre-configured devices within open systems.

d) the value UNKNOWN.

For a) through c), where the called address is known, it takes the form of

1) a DL-address in the request primitive; and

NOTE 1 If the Called Address is a DL(SAP)-address, and the requesting DLS-user has issued a DL-BIND request for the Called Address, then this parameter also can take the form of a DL-identifier in the request primitive.

NOTE 2 In some cases such a DL-CONNECT request may not result in corresponding DL-CONNECT indication(s) and response(s).

2) a local DLS-user-identifier for a DL(SAP)-address in the indication primitive.

For d), when the requesting DLS-user is attempting to establish

i) a peer DLCEP, and the DLS-user has learned the calling-DLCEP-address assigned to the requested DLCEP by extra-protocol means, and the remote peer DLS-user is specifying a DLCEP-address as the called address in its attempts to establish a DLC; or

ii) a publishing DLCEP, and both the publishing and subscribing DLS-users have learned the DLCEP-address assigned to the DLC by extra-protocol means then the requesting DLS-user may specify in this parameter the value UNKNOWN, rather than a DL(SAP)-address. In such a case, when the calling address is not itself a DLCEP DL-identifier, then it should also specify the calling DLCEP-address as described in 8.5.2.2.4.

NOTE A DL-address which does not designate an active DL-address of a DLE is not recognized by that DLE. Therefore it is not possible for a DLE to initiate a DLC erroneously by being addressing at an “inactive DLSAP-address or inactive DLCEP-address of the DLE”; such addresses do not exist.

8.5.2.2.2 Calling address

a) This parameter conveys an address of the local DLSAP from which the DLC has been requested. This parameter is a DLSAP-address in the indication primitive; except as in b), it takes the form of a local DLSAP-address DL-identifier in the request primitive.

   NOTE If the receiving DLS-user has issued a DL-BIND request for the Calling Address, then this parameter also can take the form of a DLSAP-address DLS-user-identifier in the indication primitive.

b) When the requesting DLS-user is the publisher of an existing multi-peer DLC, and wishes to use the request to solicit or add new subscribers to that DLC, then the requesting DLS-user should specify in this parameter the DLCEP DL-identifier returned by the earlier DL-CONNECT request or indication primitive which was used to establish that multi-peer DLC, rather than a DLSAP-address DL-identifier.

   NOTE The QoS parameters of the new DL-CONNECT request primitive should be compatible with those of the already-existing multi-peer DLC.

8.5.2.2.3 Responding address

This parameter normally conveys a DLSAP-address of the DLSAP with which the DLC has been established. In the response primitive this parameter takes the form of a local DLSAP-address DL-identifier; in the confirm primitive this parameter is either a DLSAP-address or the value **UNKNOWN**.

NOTE 1 The value UNKNOWN occurs when establishing a multi-peer PUBLISHER DLC.

NOTE 2 If the receiving DLS-user has issued a DL-BIND request for the Responding Address, then this parameter also can take the form of a DLSAP-address DLS-user-identifier in the confirm primitive.

When

a) the responding DLS-user is the publisher of an existing multi-peer DLC, and wishes to join the requesting DLS-user to that DLC as a new subscriber; or

b) the responding DLS-user detects a DL-Connect request collision and wishes to merge requested and indicated DLCs,

then the responding DLS-user should specify in this parameter the DLCEP DL-identifier returned by the earlier DL-CONNECT request or indication primitive used to establish that DLC, rather than a DLSAP-address DL-identifier. The DLS-provider interprets this response as an attempt to merge the newly-indicated DLC into the existing DLC, after which the newly-indicated DLCEP no longer exists.

NOTE 3 The QoS parameters of the new DL-CONNECT response primitive should be compatible with those of the already-existing DLC.

NOTE 4 The DLCEP-identifier for the newly-indicated DLCEP is no longer valid.

NOTE 5 No other primitives can or will be issued at that DLCEP.

8.5.2.2.4 Calling DLCEP-address

This optional parameter conveys a specific DLCEP-address (structurally similar to a DLSAP-address, and drawn from the same overall address space), for use as the local DLCEP-address for the DLC. This parameter is absent when the specified calling address in the request primitive, or the responding address in the response primitive, is a DLCEP DL-identifier.

When this parameter is both permitted and absent, and when the DLCEP requires a local DLCEP-address, then the DLE chooses a DLCEP-address from those available to it, and assigns the DLCEP-address to the DLC.

8.5.2.3 Quality of Service parameter set

This parameter consists of a list of sub-parameters. For all parameters common to request and indication primitives, or to response and confirm primitives, the values on the primitives are related so that

a) on the request primitive, any defined value is allowed, provided that the specified value is consistent with the values of the other parameters of the same invocation of the request primitive;

b) on the response primitive, any defined value is allowed, subject to the constraints specified in the following subclauses; and

c) on the indication (or confirm) primitive, the quality of service indicated may differ from the value specified for the corresponding request (or response) primitive as indicated in Table 13.

In general the negotiation rules result in choosing the lesser of two offered levels of functionality. DLS-users which find the resulting QoS to be unacceptable should respond by issuing a DL-DISCONNECT request, specifying as a reason the unacceptability of the negotiated QoS. See 8.3.2 for the definitions of these QoS parameters.

8.5.2.3.1 DLCEP class

If the value specified in the DL-CONNECT indication is PEER, the response should be PEER. If the value specified in the DL-CONNECT indication is PUBLISHER, then the response should be SUBSCRIBER. If the value specified in the DL-CONNECT indication is SUBSCRIBER, then the response should be PUBLISHER.

8.5.2.3.2 DLCEP data delivery features

The specified values (from-requestor-to-responder(s), and from-responder(s)-to-requestor) should be returned in the DL-CONNECT response, except that the value UNORDERED may be upgraded to ORDERED, and the value DISORDERED may be upgraded to CLASSICAL.

8.5.2.3.3 DLL priority

Any defined priority lower than or equal to that specified in the corresponding indication is allowed for the response. Thus the value URGENT may be downgraded to NORMAL or TIME-AVAILABLE, and NORMAL may be downgraded to TIME-AVAILABLE.

8.5.2.3.4 DLL maximum confirm delay

This parameter is not negotiated by the DLE.

8.5.2.3.5 DLPDU-authentication

This parameter is negotiated by the DLE as specified in 6.3.3.

8.5.2.3.6 Residual activity

This parameter is negotiated by the DLE as specified in 6.3.3 and 8.3.2.6.

8.5.2.3.7 DL-scheduling-policy

This parameter is not negotiated by the DLE.

8.5.2.3.8 Maximum DLSDU sizes

Any defined value less than or equal to that specified in the corresponding indication is allowed for the response. The DLS-provider may reduce the sizes specified by the requesting DLS-user to meet DLSDU size limits set by local DLS management.

8.5.2.3.9 DLCEP buffer and queue bindings

The DLS-provider rejects any attempt to establish a DLC between a sending queue and a receiving buffer. All other possible combinations are permitted.

8.5.2.3.10 Timeliness

This parameter is not negotiated by the DLE.

8.5.2.4 DLS-user-data

This parameter allows the transmission of DLS-user-data between DLS-users, without modification by the DLS-provider. The DLS-user may transmit any integral number of octets up to the limit for DLPDUs of NORMAL priority.

NOTE The number of octets of DLS-user-data permitted is limited to that available for DLPDUs of NORMAL priority, even though DLC initiation DLPDUs are conveyed at TIME-AVAILABLE priority, to ensure that the maximum size of such DLPDUs, including all DLC parameters needed for negotiation, is not larger than the largest DLPDU used in the data transfer service.

8.5.3 Sequence of primitives

The sequence of primitives in a successful DLC/DLCEP establishment is defined by the time-sequence diagram in Figure 18 through Figure 24. These DLC/DLCEP establishment procedures may fail either

a) due to the inability of the DLS-provider to establish a DLCEP, and possibly a DLC; or

b) due to the unwillingness of the called DLS-user to accept a DL-Connect indication primitive.

NOTE For these cases, see the DLC/DLCEP release service (see 8.6.4).

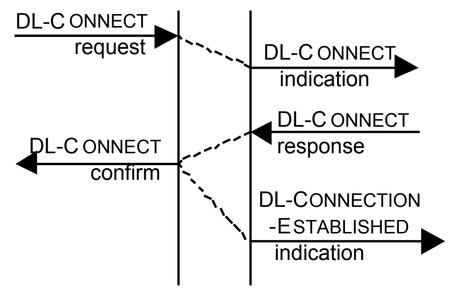


Figure 18 – Peer DLC/DLCEP establishment initiated by a single DLS-user

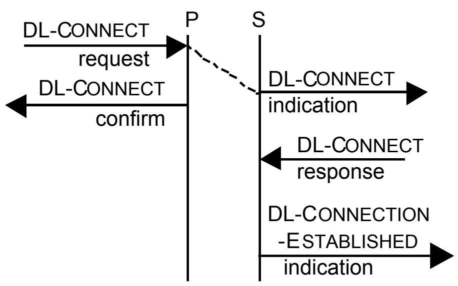


Figure 19 – Multi-peer DLC/DLCEP establishment initiated by the publishing DLS-user

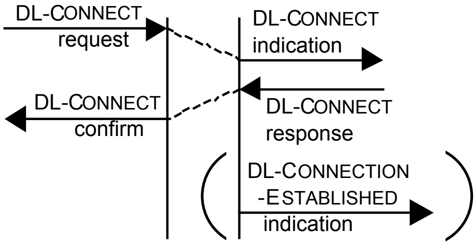


Figure 20 – Multi-peer DLC/DLCEP establishment initiated by a subscribing DLS-user

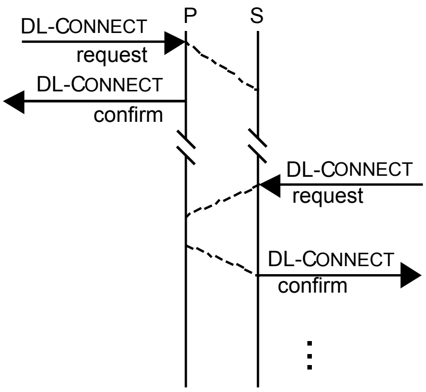


Figure 21 – Multi-peer DLC/DLCEP establishment using known DLCEP addresses initiated first by the publishing DLS-user

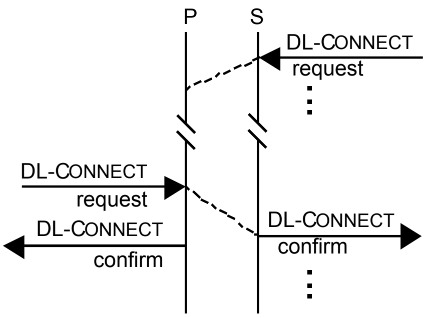


Figure 22 – Multi-peer DLC/DLCEP establishment using known DLCEP addresses initiated first by one or more subscribing DLS-users

8.5.3.1 DLC establishment collision

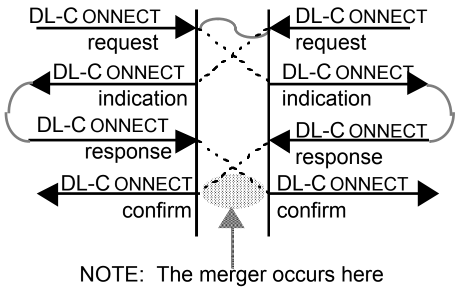


Figure 23 – Peer DLC/DLCEP establishment initiated simultaneously by both peer DLS-users, resulting in a merged DLC

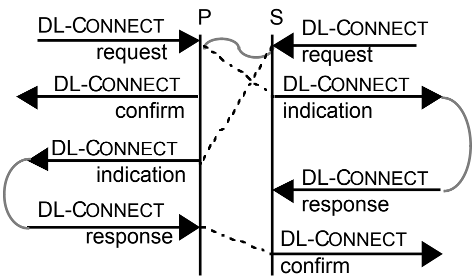


Figure 24 – Multi-peer DLC/DLCEP establishment initiated simultaneously by both publishing and subscribing DLS-users, resulting in a merged DLC

8.6 Connection release phase

8.6.1 Function

The DLC / DLCEP release service primitives are used to release a DLCEP, and possibly a DLC. The release may be initiated by any of the following:

a) either or both of the DLS-users, to release an established DLCEP;

b) the DLS-provider to release an established DLCEP (all failures to maintain a DLC at a DLCEP are indicated in this way);

c) the DLS-user, to reject a DL-Connect indication;

d) the DLS-provider, to indicate its inability to establish a requested DLC / DLCEP; or

e) the DLS-user that issued the DL-Connect request, to abandon the DLC / DLCEP establishment attempt before the DLCEP has been made available for use by receipt of a DL-CONNECT confirm primitive.

Initiation of the release service element is permitted at any time regardless of the current phase of the DLCEP. Once a release service has been initiated, the DLCEP will be disconnected and any associated buffers or queues will be unbound from the DLCEP.

A DL-DISCONNECT request cannot be rejected. The DLS does not guarantee delivery of any DLSDU associated with the DLC once the release phase is entered. Nevertheless, the release service may convey a limited amount of user data from a releasing peer (or publisher) to a peer (or subscriber), provided that neither that peer (or subscriber) nor the DLS-provider has concurrently invoked the release service.

NOTE 1 These primitives can only be used to release a DLCEP that was established by a prior DL-CONNECT primitive; they cannot be used to release a DLCEP that was established by prior local DL-management actions.

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

8.6.2 Types of primitives and parameters

Table 15 indicates the types of primitives and the parameters needed for DLC/DLCEP release.

8.6.2.1 DLCEP-identifier-type

This parameter indicates whether the associated DLCEP identifier is

a) a DLS-user-identifier which was provided by the DLS-user as a parameter of the associated prior DL-Connect request or response primitive; or

b) a DL-identifier which was provided to the DLS-user as a parameter of the immediately-prior DL-Connect indication primitive.

The latter case should occur only when the DLS-provider disconnects the DLCEP after issuing a DL-CONNECT indication primitive and before receiving a corresponding DL-CONNECT response or DL-DISCONNECT request primitive from the local DLS-user.

Table 15 – DLC / DLCEP release primitives and parameters

DL-DISCONNECT Parameter name	Request input	Indication output
DLCEP-identifier-type		M
DLCEP DLS-user-identifier		C
DLCEP DL-identifier	M	C
Originator		M
Reason	U	M (=)
DLS-user-data	U	M (=)

8.6.2.2 DLCEP DLS-user-identifier

This parameter has the same value as the DLCEP DLS-user-identifier parameter provided with the DL-CONNECT request or response primitive that occurred during DLCEP initiation. It is present on the DL-DISCONNECT indication primitive except when disconnection occurs after issuing a DL-CONNECT indication primitive and before receiving a corresponding DL-CONNECT response or DL-DISCONNECT request primitive.

8.6.2.3 DLCEP DL-identifier

The DLCEP DL-identifier parameter has the same value as the DLCEP DL-identifier parameter returned by the DL-CONNECT request or indication primitive that initiated the DLCEP. It is always present on the DL-DISCONNECT request primitive, and is present on the DL-DISCONNECT indication primitive only when disconnection occurs after issuing a DL-CONNECT indication primitive and before receiving a corresponding DL-CONNECT response or DL-DISCONNECT request primitive.

8.6.2.4 Originator

This parameter identifies the source of the release. Its value indicates either the remote DLS-user, the remote DLS-provider, the local DLS-provider, or that the originator is unknown.

8.6.2.5 Reason

This parameter gives information about the cause of the release. The value conveyed in this parameter is as follows:

a) When the originator parameter indicates a DLS-provider-generated release, the value is one of

   1) “disconnection — permanent condition”;
   2) “disconnection — transient condition”;
   3) “disconnection after timeout”;
   4) “connection rejection — calling DLSAP DL-identifier is invalid”;
   5) “connection rejection — DL(SAP)-address unknown”;
   6) “connection rejection — DLSAP unreachable, permanent condition”;
   7) “connection rejection — DLSAP unreachable, transient condition”;
   8) “connection rejection — QoS not available, permanent condition”;
   9) “connection rejection — QoS not available, transient condition”;
   10) “connection rejection after timeout”; or
   11) “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 clause of IEC 61158-3.

b) When the originator parameter indicates a DLS-user-initiated release, the value is one of
    1) “disconnection — normal condition”;
    2) “disconnection — abnormal condition”;
    3) “disconnection — terminated by unbind of source DLSAP-address”;
    4) “connection rejection — unacceptable QoS, permanent condition”;
    5) “connection rejection — non-QoS reason, permanent condition”;
    6) “connection rejection — transient condition”; or
    7) “reason unspecified”.
c) When the originator parameter indicates an unknown originator, the value of the Reason parameter is “reason unspecified”. This allows parameters to be implied when they cannot be explicitly conveyed in a DL-protocol.

8.6.2.6 DLS-user-data

This parameter allows the transmission of DLS-user-data between DLS-users, without modification by the DLS-provider. The DLS-user may transmit any integral number of octets up to the limit for DLPDUs of NORMAL priority. Delivery of this data to the remote DLS-user(s) is not assured.

NOTE 1 The number of octets of DLS-user-data permitted is limited to that available for DLPDUs of NORMAL priority, even though DLC termination DLPDUs are conveyed at TIME-AVAILABLE priority, to provide uniformity with the DLS-user-data permitted in the DL-CONNECT service.

NOTE 2 The delivery of this data is assured only for the case specified in 8.6.3a).

8.6.3 Sequence of primitives when releasing an established DLC/DLCEP

The sequence of primitives depends on the origins of the release action. The sequence may be

a) initiated by one DLS-user, with a request from that DLS-user leading to an indication to the other;
b) initiated by both DLS-users, with a request from each of the DLS-users;
c) initiated by the DLS-provider, with an indication to each of the DLS-users; or
d) initiated independently by one DLS-user and the DLS-provider, with a request from the originating DLS-user and an indication to the other DLS-user.

In cases b) and d), any DLS-user data associated with a DLS-user-initiated request may not be delivered to the remote DLS-user(s).

The sequences of primitives in these four cases are defined by the time-sequence diagrams in Figure 25 through Figure 34.

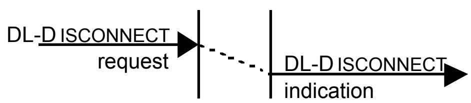


Figure 25 – Peer DLS-user invocation

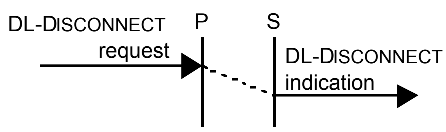


Figure 26 – Publishing DLS-user invocation

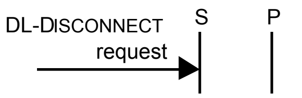


Figure 27 – Subscribing DLS-user invocation

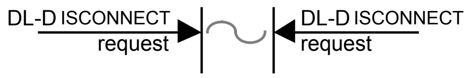


Figure 28 – Simultaneous invocation by both DLS-users

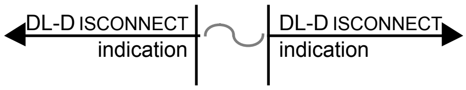


Figure 29 – Peer DLS-provider invocation

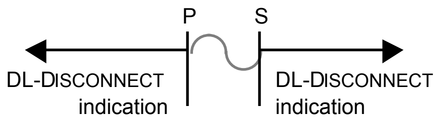


Figure 30 – Publishing DLS-provider invocation

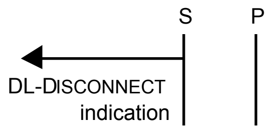


Figure 31 – Subscribing DLS-provider invocation

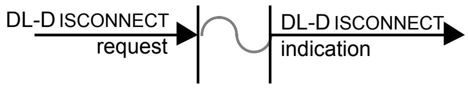


Figure 32 – Simultaneous peer DLS-user and DLS-provider invocations

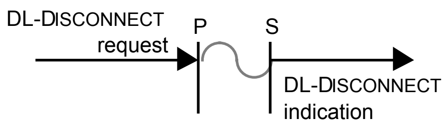


Figure 33 – Simultaneous publishing DLS-user and DLS-provider invocations

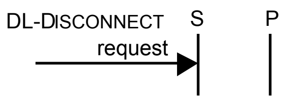


Figure 34 – Simultaneous subscribing DLS-user and DLS-provider invocations

8.6.4 Sequence of primitives in a DLS-user rejection of a DLC / DLCEP establishment attempt

A DLS-user may reject a DLC/DLCEP establishment attempt by using a DL-DISCONNECT request. The originator parameter in the DL-DISCONNECT primitives will indicate DLS-user-initiated release. The sequence of events is defined in the time-sequence diagram in Figure 35 through Figure 37.

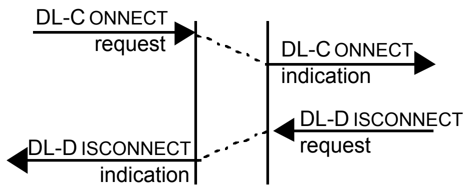


Figure 35 – Sequence of primitives in a peer DLS-user rejection of a DLC/DLCEP establishment attempt

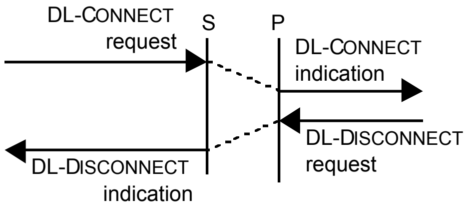


Figure 36 – Sequence of primitives in a publishing DLS-user rejection of a DLC/DLCEP establishment attempt

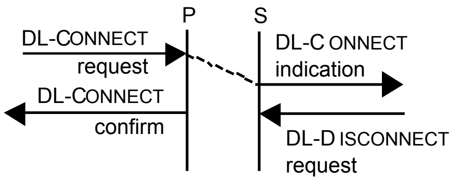


Figure 37 – Sequence of primitives in a subscribing DLS-user rejection of a DLC/DLCEP establishment attempt

If the DLS-provider is unable to establish a DLC/DLCEP, it indicates this to the requestor by a DL-DISCONNECT indication. The originator parameter in this primitive indicates a DLS-provider-originated release. The sequence of events is defined in the time-sequence diagram in Figure 38.

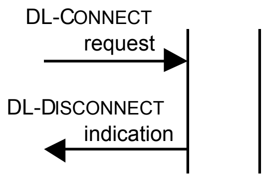


Figure 38 – Sequence of primitives in a DLS-provider rejection of a DLC/DLCEP establishment attempt

If the DLS-user, having previously sent a DL-CONNECT request and not having received a DL-CONNECT confirm or DL-DISCONNECT indication, wishes to abort the DLC/DLCEP establishment attempt, the DLS-user should issue a DL-DISCONNECT request. The resulting sequence of primitives is dependent upon the relative timing of the primitives involved and the transit delay of the DLS-provider as defined by the time-sequence diagrams in Figure 39 through Figure 43. No information can be implied by detecting which of these alternatives occur.

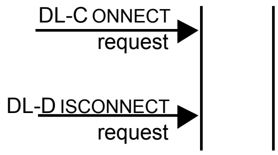


Figure 39 – Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: both primitives are destroyed in the queue

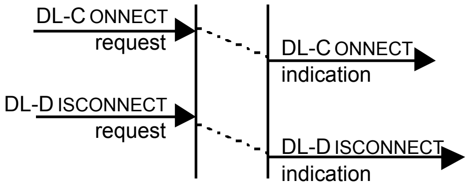


Figure 40 – Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: DL-DISCONNECT indication arrives before DL-CONNECT response is sent

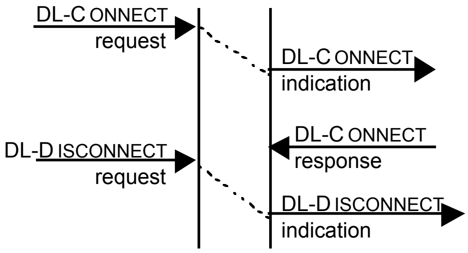


Figure 41 – Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: peer DL-DISCONNECT indication arrives after DL-CONNECT response is sent

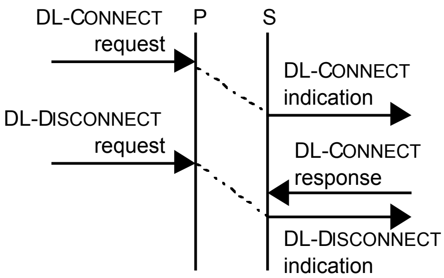


Figure 42 – Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: publisher's DL-DISCONNECT indication arrives after DL-CONNECT response is sent

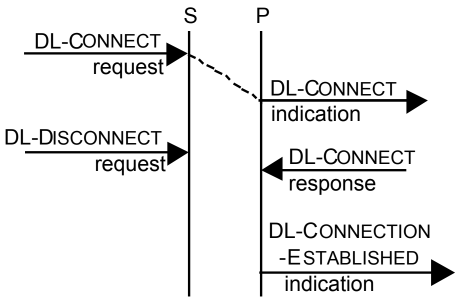


Figure 43 – Sequence of primitives in a DLS-user cancellation of a DLC/DLCEP establishment attempt: subscriber's DL-DISCONNECT request arrives after DL-CONNECT request has been communicated to the publisher

8.7 Data transfer phase

8.7.1 Queue data transfer

8.7.1.1 Function

The queue data transfer service primitives provide for conveyance of user data (DLSDUs) in either direction, or in both directions simultaneously, on a DLC. The DLS preserves the boundaries of the DLSDUs.

8.7.1.2 Types of primitives and parameters

Table 16 indicates the types of primitives and the parameters needed for queue data transfer.

Table 16 – Queue data transfer primitive and parameters

DL-DATA	Request	Indication	Confirm
Parameter name	input	output	output
Request DLS-user-identifier	M		
DLCEP DL-identifier	M		
DLCEP DLS-user-identifier		M	
Queue DLS-user-identifier		C	
DLS-user-data	M	C (=)	
Status			M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.

8.7.1.2.1 DLCEP DL-identifier

This parameter, specified in the DL-DATA request primitive, has the same value as the DLCEP DL-identifier parameter returned by the DL-CONNECT request or indication primitive that initiated the DLCEP.

8.7.1.2.2 DLCEP DLS-user-identifier

This parameter has the same value as the DLCEP DLS-user-identifier parameter returned by the DL-CONNECT confirm or DL-CONNECTION-ESTABLISHED indication primitive that occurred during initiation of the DLCEP at which the DLS-user-data was received.

8.7.1.2.3 Queue DLS-user-identifier

This parameter has the same value as a queue DLS-user-identifier parameter from a prior DL-CREATE primitive (or as created by DL-management), and indicating the same queue as the one specified in the appropriate-direction buffer-or-queue-binding parameter (see 8.5.2.3.9) of the DL-CONNECT request or response primitive that initiated the DLCEP.

8.7.1.2.4 DLS-user-data

This parameter allows the transmission of data between DLS-users without alteration by the DLS-provider. The initiating DLS-user may transmit any integral number of octets greater than zero, up to the limit negotiated for the implied direction of data transfer.

8.7.1.2.4.1 Request primitive

If the initiating DLS-user has bound a buffer to the DLCEP as a source, then this primitive is invalid and the DLC is terminated by the DLS-provider, with a DL-DISCONNECT indication issued to the appropriate DLS-user(s).

If the initiating DLS-user has bound a FIFO queue of maximum depth *K* to the DLCEP as a source, then a DL-DATA request primitive attempts to append a DLSDU to the queue, but fails if the queue already contains *K* DLSDUs. If the append operation is successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue. A DL-PUT request primitive is not permitted. Instead, the queue serves to limit the number of DLS-user requests which can be outstanding (that is, not yet confirmed to the DLS-user).

If the initiating DLS-user has not bound a buffer or FIFO queue to the DLCEP as a source, then a DL-DATA request primitive attempts to append a DLSDU to an implicit queue of indeterminate capacity, but fails if the queue is full. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

At the moment of the request, if the explicit or implicit FIFO queue is full, then the request will be rejected with an appropriate failure status on the DL-DATA request primitive.

8.7.1.2.4.2 Indication primitive

If the receiving DLS-user has bound a FIFO queue to the DLCEP as a sink, and that queue is not full, then

a) the newly-received DLSDU is appended to that queue, and the DLS-user-data parameter is omitted from the associated DL-Data indication primitive. The DLSDU can be read using a DL-Get request primitive.

If the receiving DLS-user has no binding to the DLCEP as a sink, then

b) an implicit queue of indeterminate capacity is used as a temporary receive queue, after which the DLSDU is delivered as the DLS-user-data parameter of the associated DL-DATA indication primitive.

NOTE When a buffer is bound to the DLCEP as a sink, a DL-DATA-Indication primitive cannot occur.

If it is not possible to append the received DLSDU to the implicit or explicit receive queue, then

c) if the DLCEP's data delivery features are unordered or ordered, then the DLSDU is discarded and a DL-Data indication primitive is not issued to the DLS-user; or

d) if the DLCEP's data delivery features are classical or disordered, then the DLSDU is retained by either the sending or receiving DLE (or both) until such delivery is possible, or until the DLCEP is reset or disconnected, or until the DLSDU is no longer available at the sending DLCEP (which will in turn cause a reset at the receiving DLCEP).

A DL-DATA-Indication primitive reporting the DLSDU's receipt occurs at the receiving DLS-user's DLSAP.

8.7.1.2.5 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) "temporary failure — queue full";

c) "failure — incorrect amount of requesting DLS-user data specified";

d) "failure — incompatible DLC state";

e) "failure — reset or disconnection";

f) "failure — timeout"; or

g) "failure — reason unspecified".

NOTE 1 The only compatible DLC state is Data Transfer Ready (see Figure 17).

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 clause of IEC 61158-3.

8.7.1.3 Sequence of primitives

The operation of the DLS in transferring DLSDUs can be modeled as a queue of unknown size within the DLS-provider. The ability of a DLS-user to issue a DL-DATA request or of a DLS-provider to issue a DL-DATA indication depends on the behavior of the receiving DLS-user and the resulting state of the queue.

The sequence of primitives in a successful queue to queue data transfer is defined in the time-sequence diagrams in Figure 44 through Figure 46. The sequence of primitives in a failed queue to queue data transfer is defined in the time-sequence diagram in Figure 47.

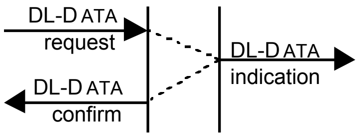


Figure 44 – Sequence of primitives for a CLASSICAL or DISORDERED peer-to-peer queue-to-queue data transfer

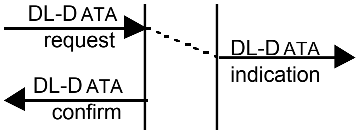


Figure 45 – Sequence of primitives for an ORDERED or UNORDERED peer-to-peer, or an UNORDERED subscriber-to-publisher queue-to-queue data transfer

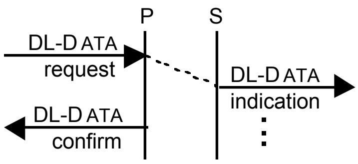


Figure 46 – Sequence of primitives for a publisher-to-subscribers queue-to-queue data transfer

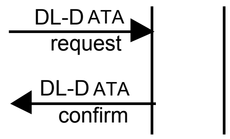


Figure 47 – Sequence of primitives for a failed queue-to-queue data transfer

If a DL-RESET or a DL-DISCONNECT primitive occurs, then the above sequences of causality may be overridden. In such a case, if a DL-DATA request primitive is outstanding, then a corresponding DL-DATA confirm primitive is issued before the reset or disconnection is signaled to the DLS-user.

8.7.2 Buffer data transfer

8.7.2.1 Function

The buffer data transfer primitives provide a means of notifying the DLS-user responsible for a buffer that the buffer was just transmitted, or that a DLSDU was just received into the buffer.

NOTE A DL-BUFFER-SENT indication will never be coincident with a DL-DATA confirm, and may or may not be related to a DL-DATA indication at one or more remote DLS-users. A DL-BUFFER-RECEIVED indication will never be coincident with a DL-DATA indication, and cannot be related to a DL-DATA request at a remote DLS-user.

8.7.2.2 Types of primitives and parameters

Table 17 and Table 18 indicate the types of primitives and the parameters needed for buffer data transfer.

Table 17 – Buffer sent primitive and parameter

DL-BUFFER-SENT **Parameter name**	**Indication** output
DLCEP DLS-user-identifier	M
Buffer DLS-user-identifier	M

Table 18 – Buffer received primitive and parameter

DL-BUFFER-RECEIVED **Parameter name**	**Indication** output
DLCEP DLS-user-identifier	M
Buffer DLS-user-identifier	M
DLSDU sequencing inference	M

8.7.2.2.1 DLCEP DLS-user-identifier

This parameter has the same value as the DLCEP DLS-user-identifier parameter returned by the DL-CONNECT confirm or DL-CONNECTION-ESTABLISHED indication primitive that occurred during initiation of the DLCEP at which the DLS-user-data was received.

8.7.2.2.2 Buffer DLS-user-identifier

This parameter has the same value as a buffer DLS-user-identifier parameter from a prior DL-CREATE primitive (or as created by DL-management), designating the same retentive buffer as the one specified in the appropriate-direction buffer-or-queue-binding parameter (see 8.5.2.3.9) of the DL-CONNECT request or response primitive that initiated the DLCEP.

8.7.2.2.3 DLSDU sequencing inference

This parameter allows the DLS-user to determine whether the DLSDU was inferred by the DLL

a) to be a duplicate of a previously-received DLSDU, or

b) to imply the loss of one or more previously-transmitted but unreceived DLSDUs, or

c) neither of the above.

When the indication is of the re-receipt of a known duplicated DLSDU, this parameter has the value "DUPLICATE"; when the indication is of the loss of one or more DLSDUs, this parameter has the value "LOSS"; in all other cases it has the value "NONE". Since UNORDERED DLCs do not provide a basis for inferring sequencing, this parameter always has the value NONE when the DLSDU was received at a DLCEP whose data delivery features are UNORDERED.

NOTE The DLS-user may be able to use this information to distinguish between problems within the remote peer DLS-user-entity and problems within the distributed DLS-provider.

8.7.2.3 Sequence of primitives

The sequence of primitives in a successful buffer to buffer, or buffer to queue, data transfer is defined in the time-sequence diagrams in Figure 48 through Figure 51.

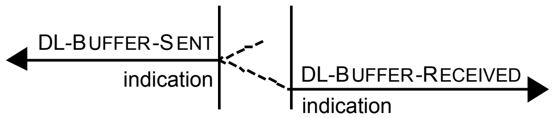


Figure 48 – Sequence of primitives for an ORDERED or UNORDERED peer to peer, or an UNORDERED subscriber to publisher, buffer to buffer data transfer

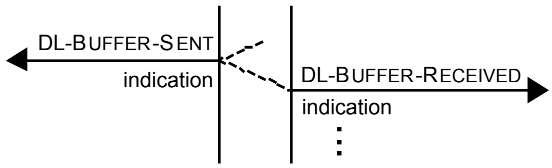


Figure 49 – Sequence of primitives for a publisher to subscribers buffer to buffer data transfer

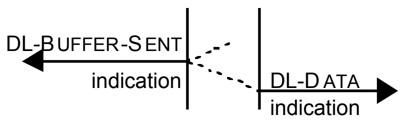


Figure 50 – Sequence of primitives for an ORDERED or UNORDERED peer to peer, or an UNORDERED subscriber to publisher, buffer to queue data transfer

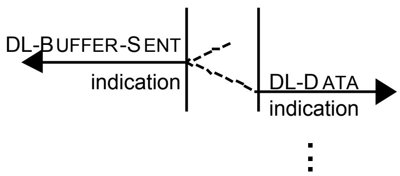


Figure 51 – Sequence of primitives for a publisher to subscribers buffer to queue data transfer

The above sequences of primitives may remain incomplete if a DL-RESET or a DL-DISCONNECT primitive occurs.

8.7.3 Reset

8.7.3.1 Function

The Reset service may be used

a) by the DLS-user of a peer or publisher DLCEP, to resynchronize the use of the DLC, and optionally to exchange a limited amount of user data with the DLC's other DLS-user(s); or

b) by the DLS-provider, to report detected loss of data unrecoverable within the DLS. All loss of data that does not involve loss of the DLC is reported in this way.

If the DLC is congested, invocation of the Reset service will unblock the flow of DLSDUs by causing the DLS-provider

1) to discard DLSDUs; and

2) to notify the appropriate DLS-user(s) that did not invoke Reset that a Reset has occurred.

The service will be completed in a finite time, whether previously-queued DLSDUs are accepted by the DLS-user or not. Any DLSDUs not delivered to the DLS-users before completion of the service will be discarded by the DLS-provider.

NOTE 1 A Reset may require a recovery procedure to be performed by the DLS-users.

NOTE 2 These primitives cannot result in the disconnection of DLCEPs that were established by prior local DL-management actions.

8.7.3.2 Types of primitives and parameters

Table 19 and Table 20 indicate the types of primitives and the parameters needed for the reset service.

Table 19 – DLC/DLCEP reset primitives and parameters (portion 1)

DL-RESET Parameter name	Request input	Indication output	Response input	Confirm output
DLCEP DL-identifier	M			
DLCEP DLS-user-identifier		M		
Originator		M		
Reason	U	M (=)		
DLS-user-data	U	M (=)	U	M (=)
Status				M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive, and by which a response primitive is correlated with its corresponding preceding indication primitive, is a local matter.				

Table 20 – DLC/DLCEP reset primitives and parameters (portion 2)

DL-RESET-COMPLETED Parameter name	Indication output
DLCEP DLS-user-identifier	M

8.7.3.2.1 DLCEP DL-identifier

The DLCEP DL-identifier parameter has the same value as the DLCEP DL-identifier parameter returned by the DL-CONNECT request or indication primitive that initiated the DLCEP.

8.7.3.2.2 DLCEP DLS-user-identifier

This parameter has the same value as the DLCEP DLS-user-identifier parameter returned by the DL-CONNECT confirm or DL-CONNECTION-ESTABLISHED indication primitive that occurred during initiation of the DLCEP at which the DL-RESET or DL-RESET-COMPLETED indication primitive was received.

8.7.3.2.3 Originator

This parameter indicates the source of the Reset; the set of values is identical to that specified in 8.6.2.2. The parameter's value indicates either the remote DLS-user, the remote DLS-provider, the local DLS-provider, or that the originator is unknown.

8.7.3.2.4 Reason

This parameter gives information about the cause of the reset. The value conveyed in this parameter is as follows:

a) When the originator parameter indicates a DLS-provider-generated reset, the value is one of
   1) "resynchronization after activation of a DL-management-established DLCEP";
   2) "resynchronization after timeout";
   3) "resynchronization after maximum number of retransmission requests or attempts";
   4) "resynchronization after detected sequence number error";
   5) "resynchronization after other detected DLCEP state inconsistencies";
   6) "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 clause of IEC 61158-3.

b) When the originator parameter indicates a DLS-user-initiated reset, the value is one of
   1) "resynchronization after DLS-user timeout";
   2) "resynchronization after DLS-user-detected DLS-user-state inconsistencies";
   3) "reason unspecified"; or

   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 clause of IEC 61158-3.

c) When the originator parameter indicates an unknown originator, the value of the Reason parameter is "reason unspecified". This allows parameters to be implied when they cannot be explicitly conveyed in the DL-protocol.

8.7.3.2.5 DLS-user-data

This parameter allows the transmission of DLS-user-data between DLS-users, without modification by the DLS-provider. The DLS-user may transmit any integral number of octets up to the limit for DLPDUs of the DLC's priority.

NOTE The delivery of this data is assured only for the case where the reset service is invoked by only one peer or publishing DLS-user, and not simultaneously by another DLS-user or the DLS-provider, and where the reset service completes before the initiation of a subsequent reset or disconnect.

8.7.3.2.6 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 — incompatible DLC state";
c) "failure — disconnection"; or

d) “failure — reason unspecified”.

NOTE 1 The only compatible DLC state is Data Transfer Ready (see Figure 17).

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 clause of IEC 61158-3.

8.7.3.3 Sequence of primitives

The interaction between each DLS-user and the DLS-provider should be one of the following exchanges of primitives:

a) a DL-RESET request from the DLS-user, followed by a DL-Reset confirm from the DLS-provider; or

b) a DL-RESET indication from the DLS-provider, followed by a DL-Reset response from the DLS-user, followed by a DL-RESET-COMPLETED indication from the DLS-provider.

The DL-RESET request acts as a synchronization mark in the stream of DLSDUs that are sent by the issuing DLS-user. The DL-RESET indication acts as a synchronization mark in the stream of DLSDUs that are received by the peer DLS-user. The DL-RESET response acts as a synchronization mark in the stream of DLSDUs that are sent by the responding DLS-user. The DL-RESET confirm acts as a synchronization mark in the stream of DLSDUs that are received by the DLS-user that originally issued the reset.

When the requesting DLS-user is the publisher of a multi-peer DLC, then this behavior is modified to account for the fact that synchronization marks corresponding to DL-RESET response primitives are not actually sent to the publishing DLCEP, but rather a DL-RESET confirm primitive is locally generated at the publishing DLCEP after sending the DL-RESET request synchronization mark to the subscribing DLCEPs.

In general, the resynchronization properties of the Reset service are that

1) No DLSDU sent by the DLS-user before the DL-RESET request or response in that sent stream is delivered to the peer DLS-user after the corresponding DL-Reset indication or confirm in that received stream.

The DLS-provider discards all DLSDUs submitted before the issuing of the DL-Reset request that have not been delivered to the peer DLS-user when the DLS-provider issues the DL-Reset indication.

Also, the DLS-provider discards all DLSDUs submitted before the issuing of the DL-Reset response that have not been delivered to the requestor of the DL-Reset when the DLS-provider issues the DL-Reset confirm.

2) No DLSDU sent by the DLS-user after the synchronization mark in that sent stream is delivered to another DLS-user **before** the synchronization mark in that received stream.

However, this partitioning of DLSDUs into pre-reset and post-reset epochs is only approximate when the DLCEP data delivery features are UNORDERED, because the lack of complete ordering includes a lack of complete ordering of the entries in the abstract queues.

The complete sequence of primitives depends upon the origin of the Reset action and the occurrence or otherwise of conflicting origins. Thus the Reset service may be

i) invoked by one DLS-user of a peer DLC, and possibly simultaneously by the DLS-provider, leading to interaction a) with that DLS-user and interaction b) with the peer DLS-user;

ii) invoked by both DLS-users of a peer DLC, leading to interaction a) with both DLS-users;

iii) invoked by the DLS-provider of a peer DLC, leading to interaction b) with both DLS-users;

iv) invoked by the subscribing user of a multi-peer DLC, or by that part of the DLS-provider associated with the subscribing DLS-user, leading to interaction a) with that DLS-user, and no interaction with the DLC’s other DLS-user(s);

v) invoked by that part of the DLS-provider associated with the publishing DLS-user, leading to interaction b) with the DLC’s DLS-users; or

vi) invoked by the publishing user of a multi-peer DLC, and possibly simultaneously by one or more subscribing users, leading to interaction a) with the invoking DLS-user(s) and interaction b) with all the DLC's other DLS-users.

The sequences of primitives for these six alternatives are defined in the time-sequence diagrams in Figure 52 through Figure 62.

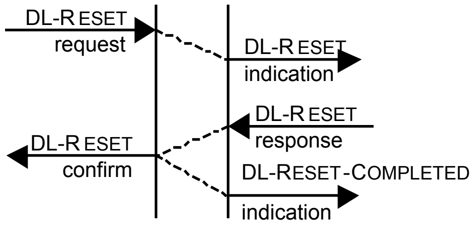


Figure 52 – Sequence of primitives in a peer DLS-user initiated Reset

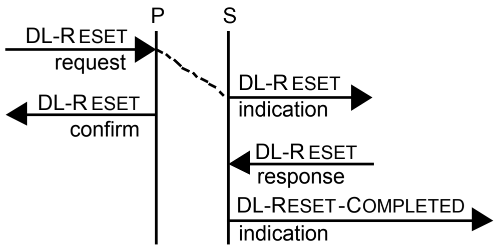


Figure 53 – Sequence of primitives in a publishing DLS-user initiated Reset

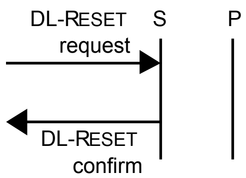


Figure 54 – Sequence of primitives in a subscribing DLS-user initiated Reset

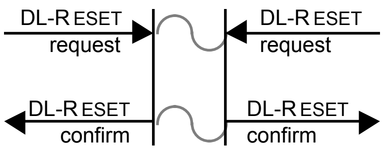


Figure 55 – Sequence of primitives in a simultaneous peer DLS-users initiated Reset

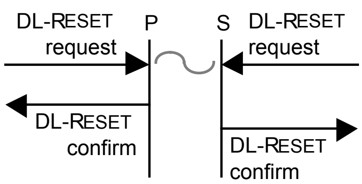


Figure 56 – Sequence of primitives in a simultaneous multi-peer DLS-users initiated Reset

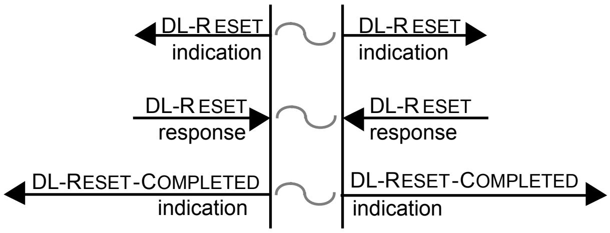


Figure 57 – Sequence of primitives in a peer DLS-provider initiated Reset

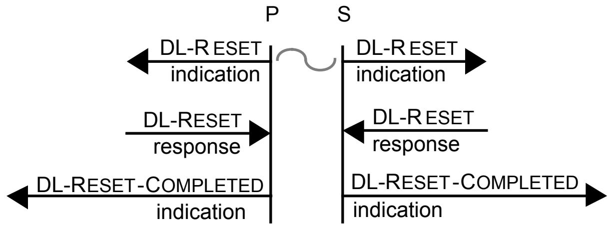


Figure 58 – Sequence of primitives in a publishing DLS-provider initiated Reset

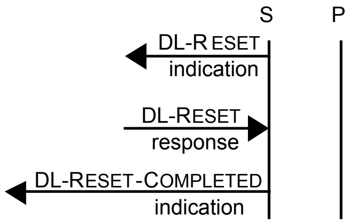


Figure 59 – Sequence of primitives in a subscribing DLS-provider initiated Reset

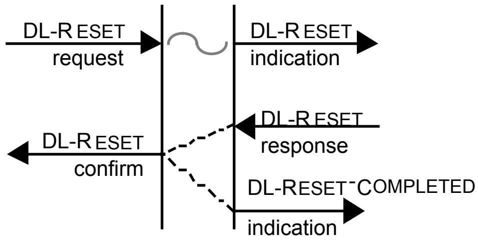


Figure 60 – Sequence of primitives in a simultaneous peer DLS-user and DLS-provider initiated Reset

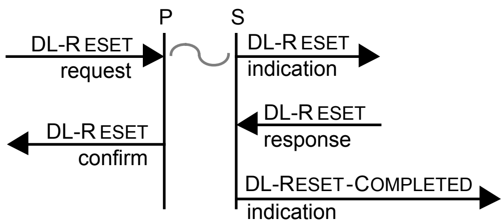


Figure 61 – Sequence of primitives in a simultaneous publishing DLS-user and DLS-provider initiated Reset

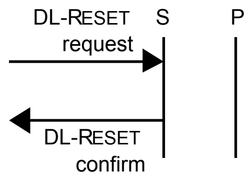


Figure 62 – Sequence of primitives in a simultaneous subscribing DLS-user and DLS-provider initiated Reset

The above sequences of primitives may remain incomplete if a DL-DISCONNECT primitive occurs. In such a case, if a DL-RESET request primitive is outstanding, then a corresponding DL-RESET confirm primitive should be issued before the disconnection is signaled to the DLS-user.

8.7.4 Subscriber query

8.7.4.1 Function

The subscriber query service primitives provide for the publishing DLS-user to query the existence of any subscribers on a multi-peer DLC. The information returned specifies whether the set of subscribing DLS-users of the multi-peer DLC appears to be empty.

8.7.4.2 Types of primitives and parameters

Table 21 indicates the types of primitives and the parameters needed for subscriber query.

Table 21 – Subscriber query primitives and parameters

DL-SUBSCRIBER-QUERY Parameter name	Request input	Confirm output
DLCEP DL-identifier	M	
DLCEP DLS-user-identifier		M
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

8.7.4.2.1 DLCEP DL-identifier

The DLCEP DL-identifier parameter has the same value as the DLCEP DL-identifier parameter returned by the DL-CONNECT request or indication primitive that initiated the DLCEP.

8.7.4.2.2 DLCEP DLS-user-identifier

This parameter has the same value as the DLCEP DLS-user-identifier parameter returned by the DL-CONNECT confirm or DL-CONNECTION-ESTABLISHED indication primitive that occurred during initiation of the DLCEP at which the DL-SUBSCRIBER-QUERY request was received.

8.7.4.2.3 Status

This parameter allows the DLS-user to determine whether the requested DLS was initiated successfully, or failed for the reason specified, and whether any receiving DLS-users appear to exist or not. The value conveyed in this parameter is as follows:

a) “success” — a subscriber exists;

b) “indeterminate — timeout”;

c) “failure — incompatible DLC state”;

d) “failure — disconnection”; or

e) “failure — reason unspecified”.

NOTE 1 The status “failure — timeout” may be interpreted with an unknown degree of confidence as “success — no subscribers”

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 which provides services as specified in this clause of IEC 61158-3.

8.7.4.3 Sequence of primitives

The sequence of primitives in a successful subscriber is defined in the time-sequence diagram in Figure 63. The scheduling of DL-SUBSCRIBER-QUERY is always IMPLICIT.

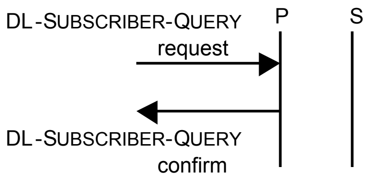


Figure 63 – Sequence of primitives for Subscriber Query

The above sequence of primitives may remain incomplete if a DL-DISCONNECT primitive occurs.

9 Type 1: Connectionless-mode Data Link Service

9.1 Facilities of the connectionless-mode Data Link Service

NOTE These facilities may not be adequate of themselves to provide the ordered-delivery enhancement to the basic connectionless OSI data link services of ISO/IEC 8886 which ISO/IEC 15802-1 promises to users of local area network medium access control protocols. In such a case it may be necessary to use an ORDERED DLC and a convergence sub-protocol to emulate the enhanced connectionless services of ISO/IEC 15802-1, with a PEER DLC used to support point-to-point (unicast) services, and a MULTI-PEER DLC used to support multicast services.

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

a) A means of transferring DLSDUs of limited length from one source DLSAP to a destination DLSAP or group of DLSAPs, without establishing or later releasing 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. QoS for this transmission can be selected by the sending DLS-user.

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

b) A means by which the status of delivery to that destination DLSAP can be returned to the source DLSAP.

c) A means by which previously submitted DLSDUs of limited length can be exchanged between two DLSAPs, and status about the exchange can be provided to the DLS-users, without establishing or later releasing a DLC. The transfer of the DLSDUs is transparent, in that the boundaries of the DLSDUs and the contents of the 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 1 The length of a DLSDU is limited because of internal mechanisms employed by the DL-protocol (see ISO/IEC 7498-1).

   NOTE 2 Because this is a unitdata service it is inherently asymmetric, the status available to the two DLS-users reflects that asymmetry. Each DLS-user receives status that the exchange occurred, but only the one with the DL(SAP)-address role of INITIATOR can receive status that its DLSDU was successfully delivered to the other DLS-user.

   NOTE 3 The ability to constrain a DLSAP-address bound in a RESPONDER role to unitdata-exchange only with a specified DLSAP-address bound in an INITIATOR role can be viewed as a form of asymmetric light-weight peer-to-peer unordered DLC.

d) A means by which a DLS-user can be notified that such an exchange was attempted, but that no DLSDUs were available for the exchange.

e) A means by which a DLS-user can query if there are any DLS-users that can receive DLSDUs sent to a specified (usually group) DL(SAP)-address.

9.2 Model of the connectionless-mode Data Link Service

This clause of IEC 61158 uses the abstract model for a layer service defined in ISO/IEC 10731, 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.

9.2.1 Model of DL-connectionless-mode unitdata transmission

A defining characteristic of data-link connectionless-mode unitdata transmission is the independent nature of each invocation of the DLS.

As a descriptive aid, the data-link connectionless-mode unitdata transmission service, as provided between any two DLSAPs, can be modeled in the abstract as an association between the two DLSAPs. This association is permanent, but its activation requires autonomous actions, in the form of appropriate DL-BIND requests, at the two DLSAPs.

Only one type of object, the unitdata object, can be handed over to the DLS-provider, via a DLSAP, for transmission. In Figure 64, DLS-user A represents the DLS-user that passes objects to the DLS-provider. DLS-user B represents the DLS-user that accepts objects from the DLS-provider.

The operations that are performed by the DLS-provider for a particular DLL association depend on the QoS specified by the requesting DLS-user. Awareness of the characteristics of the DLS provider is part of the DLS-user's à priori knowledge of the OSI environment.

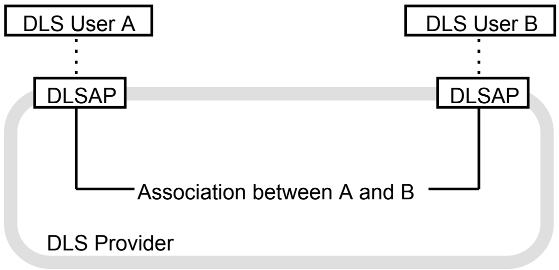


Figure 64 – Model for a data-link connectionless-mode unitdata transmission or unitdata exchange

9.2.2 Model of DL-connectionless-mode unitdata exchange

A defining characteristic of data-link connectionless-mode unitdata exchange is the independent nature of each invocation of the DLS.

As a descriptive aid, the data-link connectionless-mode unitdata exchange service, as provided between any two DLSAPs, can be modeled in the abstract as an association between the two DLSAPs. This association is permanent, but its activation requires autonomous actions, in the form of appropriate DL-BIND requests, at the two DLSAPs.

Only one type of object, the unitdata object, can be handed over to the DLS-provider, via a DLSAP, for transmission. In Figure 64, DLS-user A represents the DLS-user which has configured its relevant DLSAP-address in the role of INITIATOR. DLS-user B represents the DLS-user which has configured its relevant DLSAP-address in the role of CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER.

For each DLSAP-address, the configuring DL-BIND service specifies up to three buffers which can provide DLSDUs for transmission, and up to three buffers or queues which can hold received DLSDUs.

Each invocation of the unitdata-exchange DLS specifies a service priority, and causes each DLE to select, from those sending buffers which are bound at the specified or higher priority, the highest-priority DLSDU which is available for transmission.

The unitdata-exchange DLS consists of two phases. During the first phase, DLS-user A's DLE selects the DLSDU for transmission, if any, and sends it to DLS-user B's DLE, where the DLSDU is received and placed in the receive buffer or queue, if any, associated with the DLSDU's priority.

During the second phase, if a DLSDU was sent but was unable to be delivered in the first phase, then an error report is returned to DLS-user A's DLE. Otherwise, if either no DLSDU was sent, or the sent DLSDU was successfully placed in the appropriate receive buffer or queue, then DLS-user B's DLE selects the DLSDU for transmission, if any, and sends it to DLS-user A's DLE, where the DLSDU is received and placed in the receive buffer or queue, if any, associated with the DLSDU's priority.

For each of the two DLSAPs involved in the transaction, if a DLSDU was either sent or received at the DLSAP, or if the DLSAP-address binding specified that unitdata-exchange indications are always required, then the DLE issues a DL-UNITDATA-EXCHANGE indication primitive at that DLSAP.

9.3 Quality of connectionless-mode service

The term “Quality of Service” (QoS) refers to certain characteristics of a connectionless-mode data transmission as observed between the DLSAPs. QoS describes aspects of a connectionless-mode data transmission that are attributable solely to the DLS-provider. QoS can only be properly determined when DLS-user behavior does not constrain or impede the performance of the DLS.

Whether the QoS during each instance of connectionless-mode data transmission appears the same to each DLS-user associated with the service depends

a) on the nature of their association; and

b) on the type of information, concerning the nature of the service, made available to the DLS-user(s) by the DLS-provider prior to the invocation of the service.

9.3.1 Determination of QoS for connectionless-mode service

A basic characteristic of a connectionless-mode service is that no long-term dynamic association is set up between the parties involved. Thus, associated with each DL-connectionless-mode data transmission, certain measures of QoS are requested by the sending or initiating DLS-user when the primitive action is initiated.

NOTE The ability to constrain a DLSAP-address bound in a CONSTRAINED-RESPONDER role to unitdata-exchange only with a specified DLSAP-address bound in an INITIATOR role can be viewed as a form of asymmetric association by configuration, but without any DL-related communication between the correspondent DLS-users.

9.3.2 Definition of QoS parameters

The QoS attributes for connectionless service are:

9.3.2.1 DLL priority

Connectionless data transfer and data exchange primitives specify the priority of the transferred DLSDU(s). This parameter is defined and its default value specified in 6.3.1 and 7.4.3.2.6.1.

9.3.2.2 DLL maximum confirm delay

This parameter is defined and its default value specified in 6.3.2 and 7.4.3.2.6.2.

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 the resolution that the DLE does support.

9.3.2.3 Remote-DLE-confirmed

This Boolean parameter specifies whether the DLS-user requests confirmation of receipt of the associated DLSDU by the (implicitly identified) remote DLE. Its permissible values are **TRUE** and **FALSE**, and its default value is FALSE.

NOTE When providing a unitdata service, selection of the value TRUE inevitably uses more link capacity than does selection of the value FALSE.

9.4 Sequence of primitives

9.4.1 Constraints on sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in 9.5 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur. Other constraints, such as flow control of data, will affect the ability of a DLS-user or a DLS-provider to issue a primitive at any particular time.

The connectionless-mode primitives and their parameters are summarized in Table 22.

Table 22 – Summary of DL-connectionless-mode primitives and parameters

Service	Service subtype	Primitive	Parameter
Data Transfer	Unitdata	DL-UNITDATA request	(*in* Called DL(SAP)-address, Calling DLSAP-address DL-identifier, QoS parameter set, limited DLS-user-data)
		DL-UNITDATA indication	(*out* Called DL(SAP)-address DLS-user-identifier, Calling DLSAP-address, QoS parameter set, Queue DLS-user-identifier, limited DLS-user-data)
		DL-UNITDATA confirm	(*out* Status)
	Unitdata Exchange	DL-UNITDATA-EXCHANGE indication (3)	(*out* Local DLSAP-address DLS-user-identifier, Remote DLSAP-address, QoS parameter set, Buffer-or-queue DLS-user-identifier, Status)
Listener Query	Listener Query	DL-LISTENER-QUERY request	(*in* DL(SAP)-address, QoS parameter)
		DL-LISTENER-QUERY confirm	(*out* Status)

NOTE 1 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, 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, schedule, buffer-or-queue to the DLS-user at the DLS interface.

NOTE 2 The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.

NOTE 3 DL-UNITDATA-EXCHANGE indication occurs in isolation, without either request or confirm primitives, at both DLSAPs involved in an instance of the explicitly scheduled UNITDATA-EXCHANGE service. The DL-COMPEL-SERVICE and DL-SCHEDULE-SEQUENCE services (see 10.5.2, 10.5.3) provide the means for the explicit scheduling of the UNITDATA-EXCHANGE service.

9.4.2 Relation of primitives at the end-points of connectionless service

With few exceptions, a primitive issued at one DLSAP will have consequences at one or more other DLSAPs. The relations of primitives of each type at one DLSAP to primitives at the other DLSAPs are defined in the appropriate subclause in 9.5; all these relations are summarized in the diagrams of Figure 65.

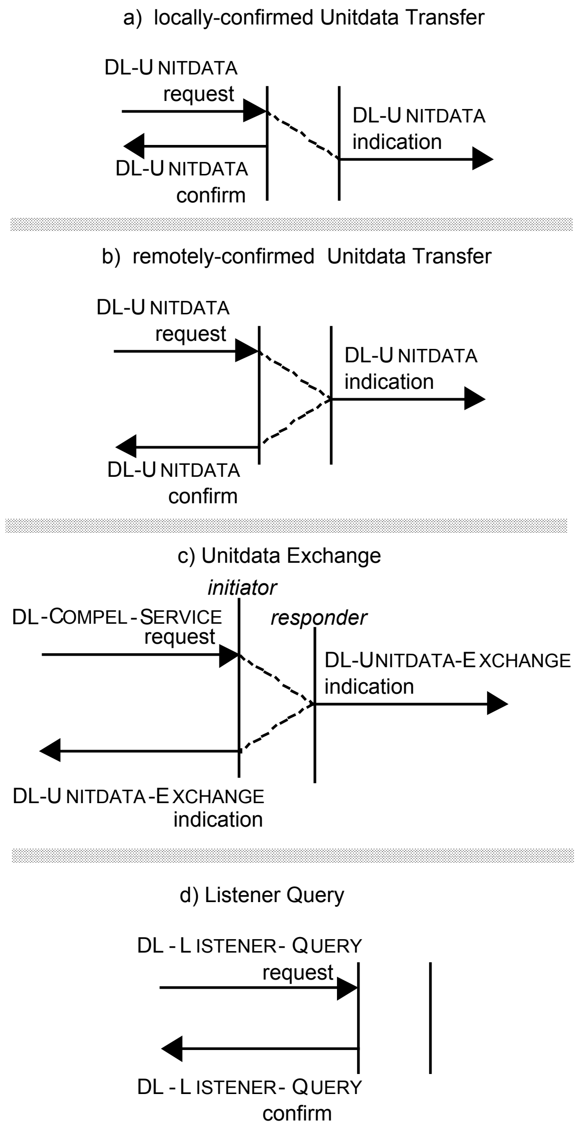


Figure 65 – Summary of DL-connectionless-mode service primitive time-sequence diagrams

9.4.3 Sequence of primitives at one DLSAP

The possible overall sequences of primitives at a DLSAP are defined in the state transition diagram, Figure 66. In the diagram, the use of a state transition diagram to describe the allowable sequences of service primitives does not impose any requirements or constraints on the internal organization of any implementation of the service.

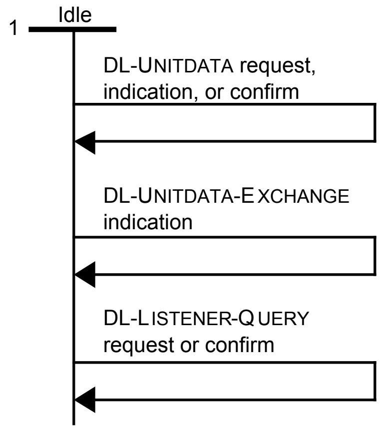


Figure 66 – State transition diagram for sequences of connectionless-mode primitives at one DLSAP

9.5 Connectionless-mode functions

DL-connectionless-mode unitdata transmission and unitdata exchange service primitives can be used to transmit independent DLSDUs from one DLSAP to another DLSAP. Each DLSDU is transmitted in a single DLPDU. The DLSDU is independent in the sense that it bears no relationship to any other DLSDU transmitted through an invocation of the DLS. The DLSDU is self-contained in that all the information required to deliver the DLSDU is presented to the DLS-provider, together with the user data to be transmitted, in a single service access.

A DLSDU transmitted using DL-connectionless-mode unitdata transmission or unitdata exchange services is not considered by the DLS-provider to be related in any way to any other DLSDU. Although the DLS maintains the integrity of individual DLSDUs, it does not necessarily deliver them to the receiving DLS-user in the order in which they are presented by the sending DLS-user.

NOTE Although the DL-UNITDATA-EXCHANGE service provides status about the delivery or transmission of one DLSDU when it reports the reception of a second DLSDU, there is no constraining relationship between the DLSDUs themselves.

No means are provided by which the receiving DLS-user may control the rate at which the sending DLS-user may send DLSDUs (peer-to-peer flow control). The DLS-provider will not maintain any state information about the flow of information between DLSAPs. Flow control exerted by the DLS-provider upon a sending DLS-user can only be described for a specific interface.

9.5.1 Data transfer

9.5.1.1 Function

This service provides the facilities of 20a) and 20b). It can be used to transmit an independent, self-contained DLSDU from one DLSAP to another DLSAP in a single service access, and to return the status of that delivery to the originating DLSAP.

This service can also be used to transmit an independent, self-contained DLSDU from one DLSAP to a group of DLSAPs, all in a single service access. In this case delivery status is not available to the originating DLSAP.

A DLSDU transmitted using DL-connectionless-mode data transfer is not considered by the DLS-provider to be related in any way to any other DLSDU. Although the DLS maintains the integrity of individual DLSDUs, it does not necessarily deliver them to the receiving DLS-user in the order in which they are presented by the sending DLS-user.

NOTE The possibility, probability and reasons for such misordering are protocol-specific.

9.5.1.2 Types of primitives and parameters

Table 23 indicates the types of primitives and the parameters needed for the DL-connectionless-mode unitdata transmission service. This service may be initiated from any DLSAP-address whose binding DL(SAP)-role is BASIC. This service may be addressed to any DL(SAP)-address whose binding DL(SAP)-role is BASIC or GROUP.

Table 23 – DL-connectionless-mode unitdata transfer primitives and parameters

DL-UNITDATA Parameter name	Request input	Indication output	Confirm output
Called address	M	M (=)	
Calling address	M	M (=)	
QoS parameter set			
DLL priority	U	M (=)	
DLL maximum confirm delay	U		
Remote-DLE-confirmed	U		
Queue DLS-user-identifier		C	
DLS-user-data	M	C (=)	
Status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

9.5.1.2.1 Addresses

The parameters that take addresses as values (see 9.5.1.2.1.1 and 9.5.1.2.1.2) both refer to DL(SAP)-addresses.

9.5.1.2.1.1 Called address

This parameter conveys an address identifying the remote DLSAP(s) with which the DLS is to be provided. It is a DL(SAP)-address in the request primitive, but takes the form of a local DL(SAP)-address DLS-user-identifier in the indication primitive(s). It may be a DLSAP-address or a group DL-address.

NOTE If the requesting DLS-user has issued a DL-BIND request for the Called Address, then this parameter also can take the form of a DL(SAP)-address DL-identifier in the request primitive.

9.5.1.2.1.2 Calling address

This parameter conveys an address of the local DLSAP from which the DLS has been requested. It is a DLSAP-address in the indication primitive, but takes the form of a local DLSAP-address DL-identifier in the request primitive.

NOTE If the receiving DLS-user has issued a DL-BIND request for the Calling Address, then this parameter also can take the form of a DLSAP-address DLS-user-identifier in the indication primitive.

9.5.1.2.2 Quality of Service

This parameter consists of a list of sub-parameters. For each parameter, the values on the primitives are related so that

a) on the request primitive, any defined value is allowed, consistent with the other parameters; and
b) on the indication primitive, the quality of service indicated is equal to the value specified for the corresponding request primitive.

9.5.1.2.2.1 DLL priority

This is the priority of the actual DLSDU which is sent to the remote peer DLS-user.

9.5.1.2.2.2 DLL maximum confirm delay

This QoS attribute is not reported on the indication primitive.

9.5.1.2.2.3 Remote-DLE-confirmed

When the called address is a group DL-address the specified value for this QoS attribute should be FALSE. This QoS attribute is not reported on the indication primitive.

9.5.1.2.3 Queue DLS-user-identifier

This parameter has the same value as a Queue DLS-user-identifier parameter from a prior DL-CREATE primitive (or as created by DL-management). It is present when an explicit queue was specified for reception at the indicated priority by the DL-BIND request primitive which established the DL(SAP)-address indicated in the same primitive.

9.5.1.2.4 DLS-user data

This parameter allows the transmission of data between DLS-users without alteration by the DLS-provider. The initiating DLS-user may transmit any integral number of octets greater than zero, up to the limit determined by the DLL priority QoS parameter specified in the service request.

9.5.1.2.4.1 Request primitive

If the initiating DLS-user has bound a FIFO queue of maximum depth *K* to the DLSAP-address at the specified priority as a source, then a DL-UNITDATA request primitive attempts to append a DLSDU to the queue, but fails if the queue already contains *K* DLSDUs. If the append operation is successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue. A DL-PUT request primitive is not permitted. Instead, the queue serves to limit the number of DLS-user requests which can be outstanding (that is, not yet confirmed to the DLS-user).

If the initiating DLS-user has not bound a FIFO queue to the DLSAP-address at the specified priority as a source, then a DL-UNITDATA request primitive attempts to append a DLSDU to an implicit queue (for this DLSAP-address and priority) of indeterminate capacity, but fails if the queue is full. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

At the moment of the request, if the explicit or implicit FIFO queue is full, then the request is terminated and a DL-UNITDATA confirm primitive is issued immediately with an appropriate negative status.

9.5.1.2.4.2 Indication primitive

If the receiving DLS-user has bound a FIFO queue to the DL(SAP)-address at the received DLSDU's priority as a sink, and that queue is not full, then

a) the newly-received DLSDU is appended to that queue, and the DLS-user-data parameter is omitted from the associated DL-UNITDATA indication primitive. The DLSDU can be read using a DL-GET request primitive.

If no such binding exists, then

b) an implicit queue of indeterminate capacity is used as the receive queue, and the DLSDU is delivered as the DLS-user-data parameter of the associated DL-UNITDATA indication primitive.

If it is not possible to append the received DLSDU to the implicit or explicit receive queue, then

c) the DLSDU is discarded and a DL-UNITDATA indication primitive is not issued to the DLS-user.

A DL-UNITDATA indication primitive reporting the DLSDU's receipt occurs at the receiving DLS-user's DLSAP.

9.5.1.2.5 Status

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

a) "success";
b) "failure — sending queue full";
c) "failure — no requesting DLS-user data specified";
d) "failure — requested QoS unavailable";
e) "failure — calling DLSAP DL-identifier is invalid";
f) "failure — incompatible DL(SAP)-role for calling address";
g) "failure — incompatible DL(SAP)-role for called address";
h) "failure — terminated by unbind of source DLSAP-address";
j) "failure — resource limitation in responder";
k) "failure — fault in responder";
m) "failure — timeout before transmission";
n) "failure — timeout after transmission, before acknowledgment"; or
p) "failure — reason unspecified".

NOTE 1 Failure g) can result from specifying a group DL-address as the called address, and requiring remote DLE confirmation.

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 clause of IEC 61158-3.

9.5.1.3 Sequence of primitives

The sequence of primitives in a successful unitdata transfer is defined in the time-sequence diagrams in Figure 67 through Figure 69.

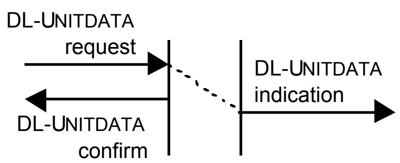


Figure 67 – Sequence of primitives for a successful locally-acknowledged connectionless-mode unitdata transfer

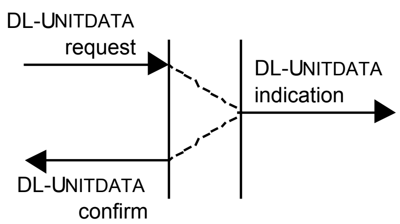


Figure 68 – Sequence of primitives for a successful remotely-acknowledged connectionless-mode unitdata transfer

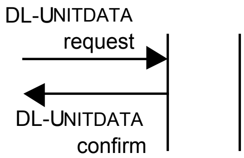


Figure 69 – Sequence of primitives for an unsuccessful connectionless-mode unitdata transfer

9.5.2 Data exchange

NOTE DL-connectionless-mode data exchange services are an extension of the Unitdata services specified in ISO/IEC 7498-1.

9.5.2.1 Function

The DL-connectionless-mode unitdata exchange service is invoked through use of the DL-COMPEL-SERVICE service (see 10.5.2). The DL-connectionless-mode unitdata exchange service can be used

a) to send a self-contained DLSDU from one local DLSAP, the initiator, to another DLSAP, the responder;

b) to cause a second independent self-contained DLSDU, which is ready for transmission at that responder DLSAP, to be returned to the initiator DLSAP; and

c) when neither DLSAP has a DLSDU ready to transmit, to notify the associated DLS-users that such an exchange was attempted.

The DLSDUs are independent in the sense that they bear no relationship to any other DLSDUs transmitted through other invocations of the DLS. The DLSDUs are self-contained in that all the information required to deliver each DLSDU is presented to the DLS-provider, together with the user data to be transmitted, in a single DL-PUT (see 7.4.5) service access. Thus no initial establishment or subsequent release of a DLC is required.

A DLSDU transmitted using DL-connectionless-mode data exchange is not considered by the DLS-provider to be related in any way to any other DLSDU. Although the DLS maintains the integrity of individual DLSDUs, it does not necessarily deliver them to the receiving (responding or initiating) DLS-user in the order in which they are presented by the sending (initiating or responding, respectively) DLS-user.

NOTE The possibility, probability and reasons for such misordering are protocol-specific.

Limited means are provided by which the receiving DLS-user may control the rate at which the sending DLS-user may send DLSDUs (peer-to-peer flow control).

1) The initiating DLS-user limits the rate at which both it and the responding DLS-user can send DLSDUs by controlling the frequency of the DL-Unitdata-Exchange service.

2) The responding DLS-user can limit the rate at which it can receive DLSDUs of a specified priority by explicitly binding a queue at that priority as a sink, and keeping the queue full. The initiating DLS-user will be informed that the responding DLE discarded the received DLSDU (due to resource limitations), and may use this information as a form of back-pressure flow control.

The DLS-provider will not maintain any state information about the flow of information between DLSAPs. Flow control exerted by the DLS-provider upon a sending DLS-user can only be described for a specific interface.

9.5.2.2 Types of primitives and parameters

Table 24 indicates the primitive and the parameters used by the DL-connectionless-mode unitdata exchange service. This service may occur between any DLSAP-address (the initiator) whose binding DL(SAP)-role is INITIATOR, and any DLSAP-address (the responder)

a) whose binding DL(SAP)-role is UNCONSTRAINED RESPONDER; or
b) whose binding DL(SAP)-role is CONSTRAINED RESPONDER and whose associated remote-DL(SAP)-address as specified in the relevant DL-BIND request primitive (see 7.4.3.2.3.2) is equal to the initiator-DLSAP-address.

Table 24 – DL-connectionless-mode unitdata exchange primitive and parameters

DL-UNITDATA-EXCHANGE Parameter name	Indication at INITIATOR output	Indication at RESPONDER output
Local address	M (1)	M (2)
Remote address	M (2)	M (1)
QoS parameter set		
DLL priority of sent DLS-user data	C (3)	C (4)
DLL priority of received DLS-user data	C (4)	C (3)
Buffer-or-queue DLS-user-identifier	C	C
Status	M	M

NOTE 1 These two parameters designate the same DLSAP-address.
NOTE 2 These two parameters designate the same DLSAP-address.
NOTE 3 These two DLL priorities are equal.
NOTE 4 These two DLL priorities are equal.

9.5.2.2.1 Addresses

The parameters that take addresses as values (9.5.2.2.1.1, 9.5.2.2.1.2) both refer to DLSAP-addresses — one which has a DL(SAP)-role of INITIATOR, and one which has a DL(SAP)-role of CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER.

9.5.2.2.1.1 Local address

This parameter conveys an address identifying the local DLSAP at which the DLS occurred. It takes the form of a DLSAP-address DLS-user-identifier in the indication primitive.

9.5.2.2.1.2 Remote address

This parameter conveys an address identifying the peer DLSAP at which the DLS occurred. It takes the form of a DLSAP-address in the indication primitive.

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

9.5.2.2.2 Quality of Service

This parameter consists of a list of sub-parameters. The first two sub-parameters are determined during the execution of this instance of the unitdata-exchange service, and are reported as part of the service indication primitives.

Two other sub-parameters affect the execution of the service instance, but are not explicitly reported. They are derived from the QoS parameters of the relevant DL-BIND request and DL-COMPEL-SERVICE request primitives, and are summarized here.

9.5.2.2.2.1 DLL priority of sent DLS-user data

This is the priority of the actual DLSDU, if any, which is sent to the remote peer DLS-user, and is greater than or equal to the transaction priority specified in the DL-COMPEL-SERVICE request primitive which initiates the DL-UNITDATA-EXCHANGE service.

9.5.2.2.2.2 DLL priority of received DLS-user data

This is the priority of the actual DLSDU, if any, which is received from the remote peer DLS-user, and is greater than or equal to the transaction priority specified in the DL-COMPEL-SERVICE request primitive which initiates the DL-UNITDATA-EXCHANGE service.

9.5.2.2.2.3 DLL priority of transaction

This QoS attribute is implicit, and is the priority specified in the DL-COMPEL-SERVICE request primitive (see 10.5.2) which initiated this instance of the DL-UNITDATA-EXCHANGE service.

9.5.2.2.2.4 DLL maximum confirm delay

This QoS attribute is implicit, and is derived from the corresponding attribute of the DL-BIND request primitive (see 7.4.3) for the initiating DLSAP-address.

9.5.2.2.2.5 Indicate-null-UNITDATA-EXCHANGE-transactions

This QoS attribute is implicit, and is derived from the corresponding attribute of the DL-BIND request primitive (see 7.4.3) for the local DLSAP-address.

9.5.2.2.3 Buffer-or-queue DLS-user-identifier

This parameter has the same value as a buffer-or-queue DLS-user-identifier parameter from a prior DL-CREATE primitive (or as created by DL-management). It is always present, because an explicit buffer or queue necessarily was specified for reception at the indicated priority by the DL-BIND request primitive which established the DL(SAP)-address indicated in the same primitive.

9.5.2.2.4 DLS-user data

NOTE The DLS-user data conveyed by this primitive is identified indirectly by the buffer-or-queue DLS-user-identifiers specified in the primitives; the primitives do not themselves directly specify DLS-user data.

The DL-connectionless-mode unitdata exchange service allows the exchange of data between DLS-users without alteration by the DLS-provider. The initiating and replying DLS-users may each transmit any integral number of octets greater than zero, up to the limits determined by the transaction priority specified in the DL-COMPEL-SERVICE request primitive (see 10.5.2) which initiates the DL-UNITDATA-EXCHANGE service, and further constrained by the actual DLL priority of the selected DLSDU.

The service name DL-UNITDATA-EXCHANGE reflects the potentiality for two-way exchange of unitdata, but the actual service also supports unidirectional data transport in either direction. The service also succeeds when neither party has a DLSDU to exchange.

9.5.2.2.4.1 Indication primitive at the initiator

a) If the initiating DLS-user
   1) has bound a buffer to the initiating DLSAP-address,
      i) as a sending buffer; and
      ii) at the transaction priority specified in the DL-COMPEL-SERVICE request primitive (see 10.5.2) which initiates the DL-UNITDATA-EXCHANGE service, or at a higher priority, and
   2) if that buffer is not empty,

then

— the DLSDU from the highest-priority non-empty buffer is sent to the responding DLS-user of the DL-UNITDATA-EXCHANGE service, together with a request for the highest priority available DLSDU whose priority is at least the transaction's priority; and

— if the selected buffer is non-retentive, then the buffer is emptied upon successful completion of the DL-UNITDATA-EXCHANGE service.

b) Otherwise, if a) does not apply, then a null DLSDU is sent to the responding DLS-user of the DL-UNITDATA-EXCHANGE service, together with a request for the highest priority available DLSDU whose priority is at least the transaction's priority.

NOTE 1 The initiator and responder addresses and QoS are provided as part of the UNITDATA-EXCHANGE invocation, and not from information associated with the buffered DLSDU.

NOTE 2 DLSDUs can be put in a sending buffer either

— by use of the DL-PUT request primitive (see 7.4.5); or

— as a result of a DL-BUFFER-RECEIVED or DL-UNITDATA-EXCHANGE indication primitive which uses the same buffer for DLSDU receipt.

c) If a reply DLPDU is received, and the reply DLPDU conveys a DLSDU, then
   1) if the initiating DLS-user bound a buffer or FIFO queue of maximum depth *K* to the initiator DLSAP-address, as a receiving buffer or queue, at the priority of the response DLSDU, then
      i) the newly-received DLSDU is put in the receive buffer, or an attempt is made to append the DLSDU to the FIFO receive queue; or
      ii) a DL-UNITDATA-EXCHANGE indication primitive notifies the initiating DLS-user of the result of putting the newly-received DLSDU in the receive buffer, or of attempting to append it to the FIFO receive queue. When not reporting an error, the DL-UNITDATA-EXCHANGE indication primitive notifies the initiating DLS-user of the priority of the received reply DLSDU.

      NOTE The DL-UNITDATA-EXCHANGE indication primitive does not convey the received DLSDU; the DLSDU can be read using a DL-GET request primitive.
   2) otherwise, when 1) does not apply, then the received DLSDU is discarded and a DL-UNITDATA-EXCHANGE indication primitive notifies the initiating DLS-user of the data loss.

d) If a reply DLPDU is received, but the reply DLPDU does not convey a DLSDU, then a DL-UNITDATA-EXCHANGE indication primitive notifies the initiating DLS-user of the completion of the unitdata exchange service, unless
   1) no DLSDU has been transferred in either direction; and
   2) the Indicate-null-UNITDATA-EXCHANGE-transactions (see 7.4.3.2.3.1) parameter which was specified as part of the DL-BIND request associated with the initiating DLSAP-address specified that such indications were not to occur.

e) If no reply DLPDU is received, and a timeout occurs, then a DL-UNITDATA-EXCHANGE indication primitive notifies the initiating DLS-user of the error.

9.5.2.2.4.2 Indication primitive at the responder

If a DLPDU is received as part of the DL-UNITDATA-EXCHANGE service, then:

a) If the DL(SAP)-role of the DLS-user at the responding DLSAP-address
   1) is BASIC, INITIATOR or GROUP; or
   2) is CONSTRAINED RESPONDER, and the DLSAP-address of the initiator is not equal to the Remote-DLSAP-address which necessarily was specified as part of the DL-Bind request associated with the responding DLSAP-address,

then a reply DLPDU is sent to inform the initiator of the error, and no DL-UNITDATA-EXCHANGE indication primitive occurs at the responder.

b) Otherwise, when a) does not apply, then if a DLSDU was sent by the initiator as part of the DL-UNITDATA-EXCHANGE service, then
   1) if the receiving DLS-user has bound a buffer or FIFO queue of maximum depth *K* to the called DLSAP-address, as a receiving buffer or queue, at the priority of the response DLSDU, then the newly-received DLSDU is copied into the receive buffer, or is appended to the FIFO receive queue if possible.

NOTE The DL-UNITDATA-EXCHANGE indication primitive does not itself specify the received DLSDU; the DLSDU can be read using a DL-GET request primitive.

2) otherwise, when 1) does not apply, the newly-received DLSDU is discarded.

c) Otherwise, when neither a) nor b) applies, then if no DLSDU was sent as part of the DL-UNITDATA-EXCHANGE service, then a consequent local DL-UNITDATA-EXCHANGE indication primitive notifies the responding DLS-user of the lack of a received DLSDU.

d) When b) or c) applies, then

1) if the responding DLS-user has bound a buffer to the responding DLSAP-address as a sending buffer, at the received transaction's priority or at a higher priority, and if that buffer is not empty, then

i) the DLSDU from the highest-priority non-empty buffer is sent as a reply to the initiating DLS-user of the DL-UNITDATA-EXCHANGE service; and

ii) if the selected buffer is non-retentive, then the buffer is set empty after completion of this instance of the DL-UNITDATA-EXCHANGE service.

NOTE DLSDUs can be put in a sending buffer either

— by use of the DL-PUT request primitive (see 7.4.5); or

— as a result of a DL-BUFFER-RECEIVED indication or DL-UNITDATA-EXCHANGE indication primitive which uses the same buffer for DLSDU receipt.

2) Otherwise, when 1) does not apply, then the absence of such data is reported to the initiating DLS-user of the DL-UNITDATA-EXCHANGE service.

When no error has occurred, and no DLSDU has been transferred in either direction, then the indicate-null-UNITDATA-EXCHANGE-transactions (see 7.4.3.2.3.1) parameter which necessarily was specified as part of the DL-BIND request associated with the responding DLSAP-address determines whether the DL-UNITDATA-EXCHANGE indication occurs or not. In all other cases the primitive is issued to the responding DLS-user.

If a DL-UNITDATA-EXCHANGE indication primitive is issued to the responding DLS-user, then the primitive notifies that DLS-user of

— the applicable error condition, if any; or

— the receipt and priority, or lack of receipt, of a DLSDU from the requesting DLS-user; and the priority of the selected reply DLSDU, if any.

9.5.2.2.5 Status

This parameter allows the DLS-user to determine whether an instance of the DL-UNITDATA-EXCHANGE service was provided successfully, or failed for the reason specified. The value conveyed in this parameter to the initiator are as follows:

a) "success";

b) "failure — resource limitation in initiator";

c) "failure — resource limitation in responder";

d) "failure — resource limitation in DLS-provider";

e) "failure — fault in responder";

f) "failure — fault in DLS-provider";

g) "failure — responder address paired with different initiator address";

h) "failure — incompatible DL(SAP)-role for responder address";

j) "failure — incompatible DL(SAP)-role for initiator address";

k) "failure — terminated by unbind of source DLSAP-address";

m) "failure — no responding DLS-user data specified";

n) "failure — timeout before transmission";

p) "failure — timeout after transmission, before reply"; or

q) "failure — reason unspecified".

The value conveyed in this parameter to the responder are as follows:

1) "success";
2) "failure — resource limitation in responder";
3) "failure — fault in responder";
4) "failure — no responding DLS-user data specified"; or
5) "failure — reason unspecified".

NOTE Addition to, or refinement of, these lists of values to convey more specific diagnostic and management information is permitted in a DL-protocol standard which provides services as specified in this clause of IEC 61158-3.

9.5.2.3 Sequence of primitives

The sequence of primitives in a successful data exchange is defined in the time-sequence diagram in Figure 70.

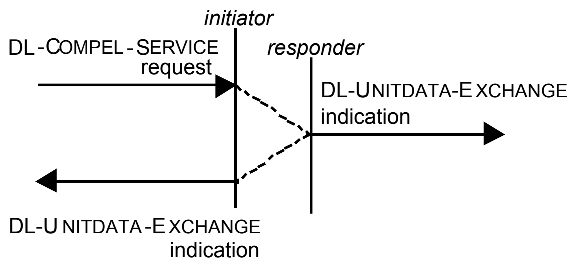


Figure 70 – Sequence of primitives for connectionless-mode unitdata exchange

9.5.3 Listener query

9.5.3.1 Function

The listener query service primitives provide for a DLS-user to query the existence of any listeners for a DLSAP-address or a group DL-address. The only information returned is whether the set of listeners appears to be empty.

9.5.3.2 Types of primitives and parameters

Table 25 indicates the types of primitives and the parameters needed for listener query.

Table 25 – Listener query primitives and parameters

DL-LISTENER-QUERY Parameter name	Request input	Confirm output
Called address	M	
QoS Parameters		
Maximum confirm delay	U	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

9.5.3.2.1 Called address

This parameter identifies the DLSAP-address or group DL-address about which the DLS-user is querying.

9.5.3.2.2 QoS parameters

9.5.3.2.2.1 Maximum confirm delay

This parameter allows the DLS-user to specify the maximum time duration permitted for the completion of the primitive. This parameter is defined and its default value is specified in 6.3.2 and 7.4.3.2.6.2.

The interval specified is the maximum permissible delay between the issuing of the DL-LISTENER-QUERY request primitive and the issuing of the corresponding DL-LISTENER-QUERY confirm primitive.

9.5.3.2.3 Status

This parameter allows the DLS-user to determine whether the requested DLS was initiated successfully, or failed for the reason specified, and whether any listening DLS-users appear to exist or not. The value conveyed in this parameter is as follows:

a) "success — a listener exists";

b) "failure — timeout"; or

c) "failure — reason unspecified".

NOTE 1 The status "failure — timeout" may be interpreted with an unknown degree of confidence as "success — no listeners"

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 which provides services as specified in this clause of IEC 61158-3.

9.5.3.3 Sequence of primitives

The sequence of primitives in a successful listener query is defined in the time-sequence diagram in Figure 71. The scheduling of DL-LISTENER-QUERY is always IMPLICIT.

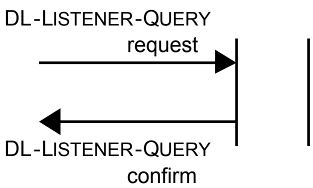


Figure 71 – Sequence of primitives for connectionless-mode listener query

10 Type 1: Time and scheduling guidance Data Link Service

10.1 Facilities and classes of the time and scheduling guidance Data Link Service

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

a) A means by which a DLS-user can request the current value of DL-time from the DLS-provider. This internal DL-time can have a resolution of a fraction of a microsecond and has a period of over 50 years. DLEs on the extended link which support a common sense of DL-time can usually synchronize that common time sense to within 1 ms, and in some cases to within 10 μs. Thus this DLS provides a means for DLS-users to indirectly synchronize and schedule their activities via this shared sense of DL-time.

b) A means by which a DLS-user can compel the DLS-provider to complete one already-issued DLS-request primitive whose execution has been deferred at the issuing DLS-user's request, and which was issued by the DLS-user itself with a local DLSAP-address or from a local DLCEP.

c) A means by which a local DLS-user can compel the DLS-provider to complete one already-issued DLS-request primitive whose execution has been deferred at the remote DLS-user's request, and which was issued by that remote DLS-user that is the peer or publisher connected to the local DLS-user's peer or subscriber DLCEP.

d) A means by which a publishing or subscribing DLS-user to an UNORDERED or ORDERED multi-peer DLC can compel the DLS-provider to distribute the current value of the buffer associated with the publishing DLCEP to the subscribing DLS-users at their subscribing DLCEPs.

e) A means by which a DLS-user can compel the DLS-provider to initiate the DL-UNITDATA-EXCHANGE service at a specified priority between a specified local DLSAP-address whose DL(SAP)-role binding is INITIATOR and another specified DLSAP-address whose DL(SAP)-role binding is UNCONSTRAINED RESPONDER or CONSTRAINED RESPONDER.

f) A means by which a DLS-user can group one or more requests of the types enumerated in b), c), d) or e) into a sequence, and schedule that sequence for either one-time or periodic or repetitive DLS.

g) A means by which a DLS-user can cancel a scheduled sequence of the type specified in f).

h) A means by which a DLS-user can dynamically select a subset of a previously-scheduled sequence of the type specified in f).

There are eight defined classes of time-related DLS:

1) **1 μs**, which offers a time-sense granularity of 1 μs or less, and a maximal time differential from the DL-subnetwork's time sense of 1 μs or less;

2) **10 μs**, which offers a time-sense granularity of 10 μs or less, and a maximal time differential from the DL-subnetwork's time sense of 10 μs or less;

3) **100 μs**, which offers a time-sense granularity of 100 μs or less, and a maximal time differential from the DL-subnetwork's time sense of 100 μs or less;

4) **1 ms**, which offers a time-sense granularity of 1 ms or less, and a maximal time differential from the DL-subnetwork's time sense of 1 ms or less;

5) **10 ms**, which offers a time-sense granularity of 10 ms or less, and a maximal time differential from the DL-subnetwork's time sense of 10 ms or less;

6) **100 ms**, which offers a time-sense granularity of 100 ms or less, and a maximal time differential from the DL-subnetwork's time sense of 100 ms or less;

7) **1 s**, which offers a time-sense granularity of 1 s or less, and a maximal time differential from the DL-subnetwork's time sense of 1 s or less; and

8) **unknown**, which offers a time-sense based on observing the most recent time-synchronization broadcast on the DL-subnetwork, with a granularity and maximal time differential from the DL-subnetwork's time sense based on that system-dependent frequency of broadcast.

NOTE These classes provide an aggregate measure of local and reference DLE clock resolution and drift rate, and of the frequency of time distribution on the local link.

10.2 Model of the time and scheduling guidance Data Link Service

This clause of IEC 61158 uses the abstract model for a layer service defined in ISO/IEC 10731, Clause 5. The model defines interactions between the DLS-user and the DLS-provider which take place at a DLSAP. Information is passed between the DLS-user and the DLS-provider by DLS-primitives which convey parameters.

10.3 Quality of scheduling guidance service

The term “Quality of Service” (QoS) refers to certain characteristics of scheduling guidance for data transmission as observed between the DLSAPs and DLCEPs. QoS describes aspects of scheduling guidance which are attributable solely to the DLS-provider. QoS can only be properly determined when DLS-user behavior does not constrain or impede the performance of the DLS.

10.4 Sequence of primitives at one DLE

10.4.1 Constraints on sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in 10.5 may occur. The constraints determine the order in which primitives occur, but may not fully specify when they may occur. Other constraints may affect the ability of a DLS-user or a DLS-provider to issue a primitive at any particular time.

The scheduling guidance primitives and their parameters are summarized in Table 26.

Table 26 – Summary of DL-scheduling-guidance primitives and parameters

Service	Primitive	Parameter
Time Query	DL-TIME request	(*out* DL-time-quality, DL-time)
Compel Service	DL-COMPEL-SERVICE request	(*in* Action class, optional Schedule DL-identifier, *out* Status)
Schedule Sequence	DL-SCHEDULE-SEQUENCE request	(*in* Schedule DLS-user-identifier. Sequence definition, optional Sequence priority, optional Schedule type, Starting conditions, Cycle specifications, optional DLSEP-address, *out* Schedule DL-identifier)
	DL-SCHEDULE-SEQUENCE confirm	(*out* Status, Scheduled starting time)
Cancel Schedule	DL-CANCEL-SCHEDULE request	(*in* Schedule DL-identifier)
	DL-CANCEL-SCHEDULE confirm	<none>
	DL-CANCEL-SCHEDULE indication	(*out* Schedule DLS-user-identifier, Reason)
Subset Schedule	DL-SUBSET-SEQUENCE request	(*in* Schedule DL-identifier, Subset mask)
	DL-SUBSET-SEQUENCE confirm	(*out* Status)

NOTE 1 DL-identifiers in parameters are local and assigned by the DLS-provider. DLS-user-identifiers in parameters are local and assigned by the DLS-user. Both types of identifiers are used by both the DLS-user and DLS-provider, as appropriate, to designate a specific DL(SAP)-address, DLCEP, schedule, buffer-or-queue at the DLS interface.

NOTE 2 The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.

The relationships among the scheduling guidance primitives at a single DLE are shown in Figure 72.

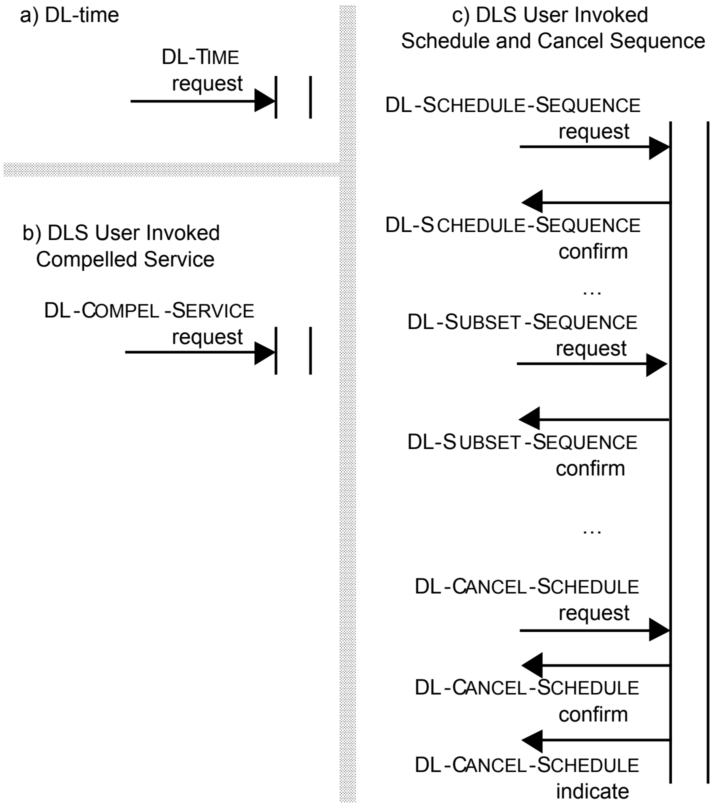


Figure 72 – Summary of time and scheduling-guidance service primitive time sequence diagrams

10.5 Scheduling guidance functions

Scheduling guidance functions permit a DLS-user to influence the timing or manner in which the DLS is provided. The use of many of these functions may occur more in a DL-management context than in an operational context. However, the functions are defined in this clause of IEC 61158-3 so that they may be used in an operational context.

NOTE It is anticipated that the operational use of these functions will prove increasingly important as experience is gained in the standardized use of time-critical communications.

10.5.1 DL-time

10.5.1.1 Function

The DL-scheduling-guidance DL-Time service primitive provides for a DLS-user to request the current sense of DL-time from the DLS-provider.

The base unit of this DL-time is 1 ms, with a potential resolution finer than 1 μs. The representation of DL-time should be unique within a period of 50 years.

This time sense should be synchronizable among DLEs on the extended link to within 1 μs, 10 μs, 100 μs, 1 ms, 10 ms, 100 ms, or 1 s, as determined by the time-synchronism classes of the devices in the synchronization path and any limitations imposed by the data rates of their interconnecting Physical Layer subsystems.

NOTE 1 Synchronization to within 10 μs may not be possible in systems where one of the PhEs has a signaling rate of less than 1 Mbit/s, or where the asymmetry in PhL propagation delays between transmit (A → B) and receive (B → A) on any one local link exceeds 2 µs. Synchronization to within 1 ms may not be possible in systems where one of the PhEs has a signaling rate of less than 4 kbit/s, or where the asymmetry in PhL propagation delays between transmit (A → B) and receive (B → A) on any one local link exceeds 250 µs. Similar considerations apply for the other time-synchronism classes.

NOTE 2 Higher layer entities may correlate the DL-time with external time such as UTC or local human time, or may cause the DL-time to be based on such external time.

NOTE 3 Implementations are permitted to maintain an arbitrary granularity consistent with their time-synchronism class, provided that they do so in a manner which is interoperable at both the DLS-interface and within the DL-protocol.

10.5.1.2 Types of primitives and parameters

Table 27 indicates the primitive and parameters of DL-time.

Table 27 – DL-time primitive and parameters

DL-TIME **Parameter name**	**Request** output
DL-time-quality	M
DL-time	M

10.5.1.2.1 DL-time-quality

This parameter conveys both

NOTE A formal DL-programming-interface specification would include the detailed encoding of this multi-component parameter.

a) the reference source, if any, of DL-time:
   1) universal coordinated time (UTC) electronically synchronized with a reference source;

      NOTE This may be obtained from an appropriate source such as a specialized radio receiver or a local atomic clock.

   2) universal coordinated time (UTC), whether obtained from a human or by electronic means, which can drift relative to the reference source;
   3) human-entered social time (that is, local date and time in some time zone); or
   4) DLS-provider-originated time (that is, not synchronized to any external time sense);

      NOTE This time sense will be measured relative to the activation of the DLS-provider and will not be correlated with any sense of external time.

b) the time-synchronism maintained by the DLS-provider when propagating that time from the DL-time source to the local DLE, and within all of the DLEs involved in that time propagation path. That time-synchronism includes
   1) the granularity of the synchronization between the DL-time source and the local DLE; and
   2) the number of intervening links on the DL-time propagation path from the DL-time source DLE to the local DLE, where a value of zero indicates that the DL-time originated within the local DLE itself.

The time-synchronism may also convey additional DL-protocol-specific information.

10.5.1.2.2 DL-time

This parameter conveys the current DLL synchronized sense of time to the DLS-user. DLS-users can use this DL-time to achieve close time synchronization throughout a single extended link. To facilitate synchronization with other DLS-users on the local link, without any interference due to synchronization over the extended link, the DL-time may be represented as a sum of more than one component.

10.5.1.3 Sequence of primitives

The sequence of primitives for obtaining the DL-time is defined in the time-sequence diagram in Figure 73.

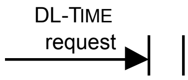


Figure 73 – Sequence of primitives for DL-time

10.5.2 Compel service

10.5.2.1 Function

The DL-scheduling-guidance Compel-Service service-primitive provides a means by which

a) a DLS-user can compel the DLL to complete one already-issued DLS-request primitive whose execution has been deferred (awaiting explicit scheduling) at the issuing DLS-user’s request, and which was issued by the DLS-user itself with a local DLSAP-address or from a local DLCEP;

b) a local DLS-user can compel the DLL to complete one already-issued DLS-request primitive whose execution has been deferred (awaiting explicit scheduling) at the remote DLS-user’s request, and which was issued by that remote DLS-user that is the peer or publisher connected to the local DLS-user’s peer or subscriber DLCEP;

c) a DLS-user to an UNORDERED or ORDERED DLC can compel the DLL to distribute the current value of a buffer associated with the local DLCEP to the remote DLS-user(s) of that DLC;

d) a DLS-user of an UNORDERED or ORDERED peer DLC can compel the DLL to distribute the current value of a buffer associated with the remote peer DLCEP to the requesting DLS-user;

e) a publishing or subscribing DLS-user to an UNORDERED or ORDERED multi-peer DLC can compel the DLL to distribute the current value of a buffer associated with the publishing DLCEP to the subscribing DLS-users at their subscribing DLCEPs; and

f) a DLS-user can compel the DLL to initiate the DL-UNITDATA-EXCHANGE service (see 9.5.2) at a specified priority between a specified local (that is, bound to the DLS-user’s DLSAP) DLSAP-address whose DL(SAP)-role binding is INITIATOR and another specified DLSAP-address whose DL(SAP)-role binding is CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER.

10.5.2.2 Types of primitives and parameters

Table 28 indicates the primitive and parameters of the Compel Service service.

Table 28 – DL-scheduling-guidance Compel Service primitive and parameters

DL-COMPEL-SERVICE	**Request**	
Parameter name	input	output
Action class	M	
DLCEP DL-identifier	C	
Local-DLSAP-address DL-identifier	C	
Remote-DLSAP-address	C	
DLL-priority	C	
Schedule DL-identifier	U	
Status		M

10.5.2.2.1 Action class

This parameter specifies whether the scheduling action is either

a) to compel execution of a deferred DLS-primitive, or to compel publishing, at a specified local DLCEP;
b) to compel execution of a deferred DLS-primitive, or to compel publishing, at the remote peer or publishing DLCEP associated with a specified local DLCEP,
c) to compel execution of a deferred DLS-primitive from a specified local DLSAP-address, or
d) to compel exchanging unitdata between specified local initiator and remote responder DLSAP-addresses at a specified or higher priority.

The values of this parameter are **LOCAL-DLCEP**, **REMOTE-DLCEP**, **LOCAL-DLSAP-ADDRESS**, and **UNITDATA-EXCHANGE**.

10.5.2.2.1.1 DLCEP DL-identifier

The DLCEP DL-identifier parameter is present when the action class parameter has the value LOCAL-DLCEP or REMOTE-DLCEP. The DLCEP DL-identifier parameter identifies the specific DLC at the local DLS interface on which a deferred DLS-primitive is being compelled to execute, or on which a publisher is being compelled to publish.

If the action class parameter has the value LOCAL-DLCEP, then the compelled execution of the deferred DLS-primitive, or the compelled publishing, occurs at that local DLCEP. If the action class parameter has the value REMOTE-DLCEP, then the compelled execution of the deferred DLS-primitive, or the compelled publishing, occurs at the remote DLCEP which is the peer or publisher of the local DLCEP's DLC.

10.5.2.2.1.2 Local-DLSAP-address DL-identifier

The local-DLSAP-address DL-identifier parameter is present when the action class parameter has the value LOCAL-DLSAP-ADDRESS or UNITDATA-EXCHANGE.

If the action class parameter has the value LOCAL-DLSAP-ADDRESS, then this parameter specifies the DLSAP-address at the local DLS interface at which a deferred DL-UNITDATA request primitive is being compelled to execute.

If the action class parameter has the value UNITDATA-EXCHANGE, then this parameter specifies the DLSAP-address at the local DLS interface at which a compelled implicit DL-UNITDATA-EXCHANGE service (see 9.5.2) is initiated. The DLSAP shall be bound to this DLSAP-address with a DL(SAP)-role of INITIATOR.

10.5.2.2.1.3 Remote-DLSAP-address

This parameter is present when the action class parameter has the value UNITDATA-EXCHANGE. This parameter specifies the responder DLSAP-address for the compelled implicit DL-UNITDATA-EXCHANGE service. The peer DLS-user, through its DLSAP, shall be bound to this DLSAP-address with a DL(SAP)-role of UNCONSTRAINED RESPONDER or CONSTRAINED RESPONDER.

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

10.5.2.2.1.4 DLL-priority

This parameter is present when the action class parameter has the value LOCAL-DLSAP-ADDRESS or UNITDATA-EXCHANGE. This parameter specifies the minimum priority permitted for any DLSDU conveyed during the compelled DL-UNITDATA service or the compelled implicit DL-UNITDATA-EXCHANGE service.

10.5.2.2.2 Schedule DL-identifier

This optional parameter specifies the schedule DL-identifier returned by a successful prior DL-SCHEDULE-SEQUENCE confirm primitive whose schedule has not yet been cancelled, and whose associated sequence definition permits dynamic sequence construction.

When this parameter is absent, the DLE schedules the compelled DLS-user request for service on the same IMPLICIT basis as implicitly-scheduled requests (see 6.3.4). When this parameter is present, the DLE dynamically appends the compelled request to the specified sequence for one-time execution when that sequence is next executed by the DLE.

10.5.2.2.3 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 — inappropriate request”; or

c) “failure — reason unspecified”.

NOTE 1 Successful completion of a compel-service request does not imply successful completion of the compelled DLS-primitive or compelled publishing or unitdata-exchange, but only that the DLE has been able to commit to the requested activity.

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 which provides services as specified in this clause of IEC 61158-3.

10.5.2.3 Sequence of primitives

The sequence of primitives in the Compel Service service is defined in the time-sequence diagram of Figure 74.

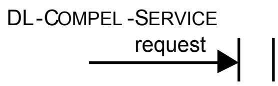


Figure 74 – Sequence of primitives for the Compel Service service

10.5.3 Schedule sequence

10.5.3.1 Function

The DL-scheduling-guidance schedule sequence service primitives provide for a DLS-user to schedule a predefined or dynamically-defined sequence of DL-COMPEL-SERVICE requests for subsequent execution, either

a) one time only, or periodically (that is, cyclically with a specified period) until cancelled, starting at a specified time or within a specified time interval; or

b) repetitively but not intentionally periodically, as frequently as possible, but no more than once per major DLL scheduling cycle of the local link.

NOTE Periodic repetition supports a paradigm commonly used in sampled-data control systems which use mathematical algorithms based on the arithmetic of finite differences. Aperiodic repetition supports a paradigm commonly used by programmable logic controllers, where the basic cycle of the (distributed) machine is repeated as frequently as possible, without concern for a fixed repetition period. These scheduling primitives permit the two to coexist on a single local link.

10.5.3.2 Types of primitives and parameters

Table 29 indicates the types of primitives and the parameters needed for the Schedule Sequence service.

10.5.3.2.1 Local-view DL-identifiers

10.5.3.2.1.1 Schedule DLS-user-identifier

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

The naming-domain of the schedule DLS-user-identifier is the DLS-user's local-view.

10.5.3.2.1.2 Schedule DL-identifier

This parameter, which is returned by the DLS-provider on successful DL-SCHEDULE-SEQUENCE request primitives, provides a local means of identifying a specific scheduling request in input parameters of other local DLS primitives which convey the name of the schedule from the local DLS-user to the local DLE.

The naming-domain of the schedule DL-identifier is the DL-local-view.

Table 29 – DL-scheduling-guidance Schedule Sequence primitives and parameters

DL-SCHEDULE-SEQUENCE	Request		Confirm
Parameter name	input	output	output
Schedule DLS-user-identifier	M		
Schedule DL-identifier		M	
Sequence definition	M		
Sequence priority	U		
Schedule type	U		
Starting conditions			
Desired starting time	CU		
Earliest starting time	CU		
Latest ending time	CU		
Cycle specifications			
Cycle period	C		
Maximum permissible jitter	CU		
DLSEP-address	U		
Status			M
Scheduled starting time			C

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.

10.5.3.2.2 Sequence definition

This parameter specifies a sequence of encoded (see note 2) DL-COMPEL-SERVICE requests, including constraints among the sequence elements on the execution order and contiguous execution requirements of the sequence, and on whether the Subset Sequence service (see 10.5.5) can be used to subset the defined sequence dynamically.

The sequence definition may also specify that additional DL-COMPEL-SERVICE requests may be dynamically appended to the schedule for one-time execution, up to some DLS-user-specified limit on required additional link-capacity, through use of the optional schedule DL-identifier parameter of the DL-COMPEL-SERVICE request primitive. Subsetting does not apply to these dynamically-appended requests.

The DLS-provider will attempt to schedule the sequence according to its internal constraints and the other parameters in the current request.

NOTE 1 Scheduling request which do not require contiguous execution are more readily satisfied than ones which do, since the former permit the newly-requested transactions to be interleaved with currently-scheduled transactions while meeting the other scheduling constraints.

NOTE 2 A formal DL-programming-interface specification would include the details of the sequence definition's encoding.

10.5.3.2.3 Sequence priority

This parameter specifies an optional DLS-provider priority (see 6.3.1) for scheduling of the sequence relative to other such sequences. Its default value is TIME-AVAILABLE.

10.5.3.2.4 Schedule type

This parameter specifies whether the sequence is to be scheduled for one-time or periodic execution, or repetitive but not intentionally periodic execution. Its values are **ONE-TIME**, **PERIODIC** and **REPETITIVE**. Its default value is ONE-TIME.

NOTE REPETITIVE scheduling is a mode of scheduling in which execution of the specified sequence occurs as frequently as possible, but in an equitable fashion that gives access opportunities to all DLEs on the local link. It is provided to support migration of prior national standards.

10.5.3.2.5 Starting conditions

This compound parameter specifies the conditions which determine when the scheduled sequence will be executed. These parameters are present when the schedule type parameter has the value PERIODIC or ONE-TIME.

10.5.3.2.5.1 Desired starting time

This parameter specifies the internal DL-time (see 10.5.1) at which execution of the specified sequence of DLS requests should start. The value of this parameter should be either **IMMEDIATE** (meaning as-soon-as-possible) or a future DL-time within the interval from the current moment to the current moment plus 26 h. The default value for this parameter is IMMEDIATE. The granularity of any specified DL-time should be as described in 10.5.1.1.

NOTE The interval of 26 h provides the ability to schedule events one full day in advance, even when that day has 25 hours to account for changes in external social (human) time.

10.5.3.2.5.2 Earliest starting time

This parameter specifies the earliest internal DL-time at which execution of the specified sequence of DLS requests may start. The value of this parameter should be either IMMEDIATE or a future DL-time within the interval from the current moment to the moment specified by the Desired Starting Time parameter. The default value for this parameter is IMMEDIATE. The granularity of any specified DL-time should be as described in 10.5.1.1.

10.5.3.2.5.3 Latest ending time

This parameter specifies the latest internal DL-time by which execution of the first instance of the specified sequence of DLS requests needs to complete. The value of this parameter should be a future DL-time within the interval from the moment specified by the Desired Starting Time parameter to the current moment plus 26 h. The default value for this parameter is the current moment plus 26 h — the maximum specifiable. The granularity of this parameter should be as described in 10.5.1.1.

10.5.3.2.6 Cycle specifications

This compound parameter specifies the conditions which determine the repetition interval when the scheduled sequence is to be executed repeatedly with a fixed periodicity. These parameters are present when the schedule type parameter has the value PERIODIC.

NOTE No such conditions are needed when the sequence is to be executed repetitively.

10.5.3.2.6.1 Cycle period

This parameter specifies the increment in DL-time between successive starting times for the scheduled sequence. Its value should be less than 26 h. The granularity of this parameter should be as described in 10.5.1.1.

NOTE The DLS-provider will impose a lower limit on this parameter, based upon link characteristics and configuration, and possibly on existing schedule commitments,

10.5.3.2.6.2 Maximum permissible jitter

This parameter specifies a single upper bound on the deviation in the starting time of any single period from the nominal starting time for that period, where that nominal starting time is the Scheduled Starting Time returned by the DL-SCHEDULE-SEQUENCE confirm primitive plus a multiple of the specified Cycle Period parameter. The value of this parameter should be less than the value of the Cycle Period parameter. The default value for this parameter is zero. The granularity of this parameter should be as described in 9.5.1.1.

NOTE Specifying a non-zero value for this parameter permits improved link utilization, and may permit requests to be added to an existing schedule which would otherwise have to be rejected due to conflicting timing requirements.

10.5.3.2.7 DLSEP-address

This optional parameter conveys a specific DLSEP-address (structurally similar to a DLCEP-address, and drawn from the same address space), for use as the local DLSEP-address for scheduled sequence execution. When this parameter is absent, the DLE chooses a DLSEP-address from those available to it, and associates the DLSEP-address with the scheduled sequence.

10.5.3.2.8 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 sequence definition”;
c) “failure — insufficient resources”;
d) “failure — cannot commit to requested start window”;
e) “failure — cannot commit to requested periodicity”;
f) “failure — parameter violates management constraint”;
g) “failure — number of requests violates management constraint”; or
h) “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 which provides services as specified in this clause of IEC 61158-3.

10.5.3.2.9 Scheduled starting time

This parameter is present when the Schedule Type parameter specifies ONE-TIME or PERIODIC and the Status parameter indicates that the sequence was scheduled successfully. This parameter specifies the internal DL-time at which execution of the specified sequence of DLS requests started or is scheduled to start. The value of this parameter is a DL-time within the interval from the moment of the corresponding request to that moment plus 26 h. The granularity of this parameter should be as described in 10.5.1.1.

10.5.3.3 Sequence of primitives

The sequence of primitives in successfully scheduling a defined sequence, and later possibly canceling the scheduled sequence, is defined in the time-sequence diagram of Figure 75.

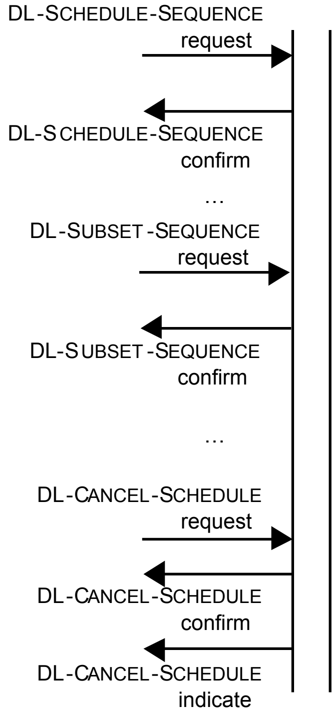


Figure 75 – Sequence of primitives for the sequence scheduling services

10.5.4 Cancel schedule

10.5.4.1 Function

The DL-scheduling-guidance cancel schedule service primitives provide for a DLS-user to cancel a previously scheduled sequence, and release the schedule DL-identifier and DLS-provider resources associated with that schedule.

NOTE This cancellation and release may not be instantaneous.

In some cases the DLS provider may find it necessary to cancel a previously-committed schedule. Such an event is indicated to the scheduling DLS-user by the DL-SCHEDULE-SEQUENCE indication primitive.

10.5.4.2 Types of primitives and parameters

Table 30 indicates the types of primitives and the parameters needed for the cancel schedule service.

Table 30 – DL-scheduling-guidance Cancel Schedule primitives and parameters

DL-CANCEL-SCHEDULE Parameter name	Request input	Confirm output	Indication output
Schedule DL-identifier	M		
Schedule DLS-user-identifier			M
Reason			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

10.5.4.2.1 Schedule DL-identifier

The Schedule DL-identifier parameter specifies the schedule DL-identifier returned by a successful prior DL-SCHEDULE-SEQUENCE confirm primitive whose schedule has not yet been cancelled. The DLS provider will cancel the schedule associated with the DL-identifier and release the schedule DL-identifier and associated DLS provider resources.

10.5.4.2.2 Schedule DLS-user-identifier

This parameter has the same value as the schedule DLS-user-identifier parameter from the corresponding earlier DL-SCHEDULE-SEQUENCE request primitive.

10.5.4.2.3 Reason

This parameter gives information about the cause of the sequence cancellation. The value conveyed in this parameter is as follows:

a) “schedule preemption”;

b) “loss-of-schedule — rescheduling required”; or

c) “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 which provides services as specified in this clause of IEC 61158-3.

10.5.4.3 Sequence of primitives

The sequence of primitives in canceling a scheduled sequence is defined in the time-sequence diagram in Figure 75. Unrequested cancellation by the DLS provider is also shown.

10.5.5 Subset sequence

10.5.5.1 Function

The DL-scheduling-guidance subset sequence service primitives provide for a DLS-user to select dynamically a subset of a previously-scheduled subsettable sequence.

NOTE This subsetting action may not be instantaneous.

10.5.5.2 Types of primitives and parameters

Table 31 indicates the types of primitives and the parameters needed for the subset sequence service.

Table 31 – DL-scheduling-guidance Subset Sequence primitives and parameters

DL-SUBSET-SEQUENCE Parameter name	Request input	Confirm output
Schedule DL-identifier	M	
Subset mask	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

10.5.5.2.1 Schedule DL-identifier

The Schedule DL-identifier parameter specifies the schedule DL-identifier returned by a successful prior DL-SCHEDULE-SEQUENCE confirm primitive whose schedule has not yet been cancelled. The DLS provider will subset the schedule associated with the identifier.

10.5.5.2.2 Subset mask

The subset mask specifies the subset of the defined sequence which is to be excluded from the operational schedule.

NOTE 1 A formal DL-programming-interface specification would include the details of the subset mask's encoding.

NOTE 2 The name and definition of this parameter are not intended to constrain the form of a conforming implementation. In particular, the subset mask need not be represented as a Boolean vector.

10.5.5.2.3 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 — sequence is not defined";

c) "failure — sequence is not subsettable"; 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 which provides services as specified in this clause of IEC 61158-3.

10.5.5.3 Sequence of primitives

The sequence of primitives in subsetting a scheduled sequence is defined in the time-sequence diagram in Figure 75.

11 Types 1 and 4: DL-management Service

11.1 Scope and inheritance

This clause defines the form of DL-management services for protocols which implement the DLS specified in clauses 6 through 10 and 14. Only the form is specified, as the specifics of permitted parameters are dependent on the protocol which implements these services.

This noteworthy difference of this clause from the prior five clauses is the intended class of users; this clause is intended for use by a management client, while the prior five clauses provide services to any client.

11.2 Facilities of the DL-management service

DL-management facilities provide a means for

a) writing DLE configuration parameters;

b) reading DLE configuration parameters, operational parameters and statistics;

c) commanding major DLE actions; and

d) receiving notification of significant DLE events.

Together these facilities constitute the DL-management-service (DLMS).

11.3 Model of the DL-management service

This clause of IEC 61158 uses the abstract model for a layer service defined in ISO/IEC 10731, Clause 5. The model defines local interactions between the DLMS-user and the DLMS-provider. Information is passed between the DLMS-user and the DLMS-provider by DLMS primitives that convey parameters.

11.4 Constraints on sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in 11.5 through 11.8 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur.

The DL-management primitives and their parameters are summarized in Table 32. The only primitives with a time-sequence relationship are shown in Figure 76.

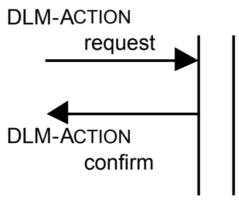


Figure 76 – Sequence of primitives for the DLM action service

Table 32 – Summary of DL-management primitives and parameters

Service	Primitive	Parameter
writing managed objects	DLM-SET request	(*in* DLM-object-identifier, Desired-value, *out* Status)
reading managed objects	DLM-GET request	(*in* DLM-object-identifier, *out* Status, Current-value)
commanding actions	DLM-ACTION request	(*in* Desired-action, Action-qualifiers)
	DLM-ACTION confirm	(*out* Status, Additional-information)
notification of events	DLM-EVENT indication	(*out* DLM-event-identifier, Additional-information)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

11.5 Set

11.5.1 Function

This primitive can be used to set (write) the value of a DLE configuration parameter.

11.5.2 Types of parameters

Table 33 indicates the primitive and parameters of the set DLMS.

Table 33 – DLM-Set primitive and parameters

DLM-SET Parameter name	Request	
	input	output
DLM-object-identifier	M	
Desired-value	M	
Status		M

11.5.2.1 DLM-object-identifier

This parameter specifies the primitive or composite object within the DLE whose value is to be altered. The naming-domain of the DLM-object-identifier is the DLM-local-view.

11.5.2.2 Desired-value

This parameter specifies the desired value for the DLM-object specified by the associated DLM-object-identifier. Its type is identical to that of the specified DLM-object.

11.5.2.3 Status

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

a) “success”;
b) “failure — DLM-object-identifier is unknown”;
c) “failure — desired-value is not permitted”; 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 clause of IEC 61158-3.

11.6 Get

11.6.1 Function

This primitive can be used to get (read) the value of a DLE configuration parameter, operational parameter or statistic.

11.6.2 Types of parameters

Table 34 indicates the primitive and parameters of the get DLMS.

Table 34 – DLM-Get primitive and parameters

DLM-GET Parameter name	Request	
	input	output
DLM-object-identifier	M	
Status		M
Current-value		C

11.6.2.1 DLM-object-identifier

This parameter specifies the primitive or composite object within the DLE whose value is being requested. The naming-domain of the DLM-object-identifier is the DLM-local-view.

11.6.2.2 Status

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

a) “success”;

b) “failure — DLM-object-identifier is unknown”; 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 clause of IEC 61158-3.

11.6.2.3 Current-value

This parameter is present when the status parameter indicates that the requested service was performed successfully. This parameter specifies the current value for the DLM-object specified by the associated DLM-object-identifier. Its type is identical to that of the specified DLM-object.

11.7 Action

11.7.1 Function

This primitive can be used to command a specified action of the DLE.

11.7.2 Types of parameters

Table 35 indicates the primitives and parameters of the action DLMS.

Table 35 – DLM-Action primitive and parameters

DLM-ACTION Parameter name	Request input	Confirm output
Desired-action	M	
Action-qualifiers	C	
Status		M
Additional-information		C
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

11.7.2.1 Desired-action

This parameter specifies the desired action which the DLE is to take.

11.7.2.1.1 Action-qualifiers

This optional parameter specifies additional action-specific parameters which serve to qualify the commanded action.

11.7.2.2 Status

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

a) “success”;
b) “failure — action is unknown”;
c) “failure — action is not permitted in current DLE state”;
d) “failure — action did not complete”; 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 clause of IEC 61158-3.

11.7.2.2.1 Additional-information

This optional parameter provides action-specific additional information which augments the returned status.

11.7.3 Sequence of primitives

The sequence of primitives in a successful DLM-commanded action is defined in the time-sequence diagram in Figure 76.

11.8 Event

11.8.1 Function

This primitive is used to report the occurrence of a DL-event which could be of significance to DL-management.

11.8.2 Types of parameters

Table 36 indicates the primitive and parameters of the event DLMS.

Table 36 – DLM-Event primitive and parameters

DLM-EVENT Parameter name	Indication output
DLM-event-identifier	M
Additional-information	C

11.8.2.1 DLM-event-identifier

This parameter specifies the primitive or composite event within the DLE whose occurrence is being announced. The naming-domain of the DLM-event-identifier is the DLM-local-view.

11.8.2.1.1 Additional-information

This optional parameter provides event-specific additional information.

12 Type 2: Connection-mode and connectionless-mode Data Link Service

12.1 Overview

12.1.1 Data transfer services

The primary task of a DLE is to determine, in co-operation with other DLEs on the same local link, the granting of permission to transmit on the medium. At its upper interface, the DLL provides services to receive and deliver service data units (DLSDUs) for higher level entities.

NOTE 1 The following access mechanisms are not visible to the higher level entities. They are described here as an aid to understanding the purpose and use of DLS parameters and services that are visible to higher layer entities.

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

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

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

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

a) the current node to hear the end of the transmission from the previous node, and
b) the current node to begin transmitting, and
c) the next node to hear the beginning of the transmission from the current node.

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

NOTE 2 The calculation of slot time is the responsibility of System Management.

Each NUT is divided into three major parts: scheduled, unscheduled, and guardband as shown in Figure 77. This sequence is repeated in every NUT. The implicit token passing mechanism is used to grant access to the medium during both the unscheduled and scheduled intervals.

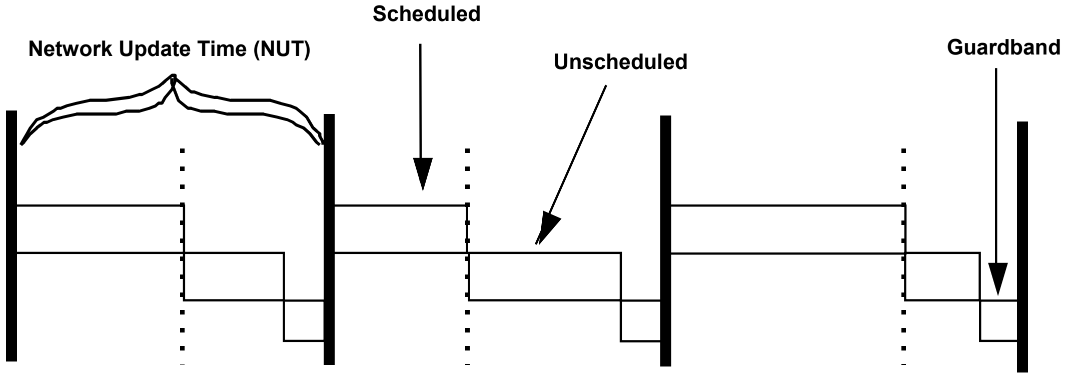


Figure 77 – NUT structure

During the scheduled part of the NUT, each node, starting with node 0 and ending with node SMAX, gets a chance to transmit time-critical (scheduled) data. SMAX is the MAC ID of the highest numbered node that has access to the medium during the scheduled part of the NUT. Every node between 0 and SMAX has only one opportunity to send one DLPDU of scheduled data in each NUT. The opportunity to access the medium during the scheduled time is the same for each node in every NUT. This allows data that is transmitted during the scheduled portion of the NUT to be sent in a predictable and deterministic manner.

Figure 78 shows how the permission to transmit is granted during the scheduled time. The DLS-user regulates the amount of data that each node may transmit during this scheduled token pass.

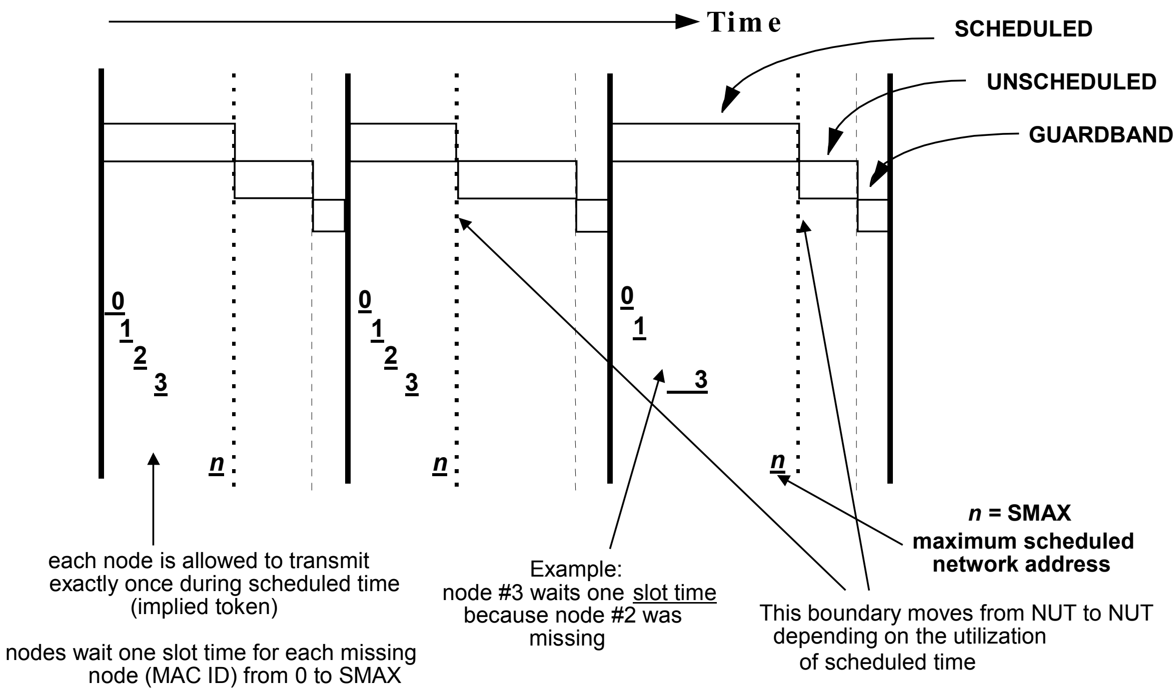


Figure 78 – Medium access during scheduled time

During the unscheduled part of the NUT, each node from 0 to UMAX shares the opportunity to transmit one DLPDU of non-time-critical data in a round robin fashion, until the allocated NUT duration is exhausted. UMAX is the MAC ID of the highest numbered node that has access to the medium during the unscheduled part of the NUT. The round robin method of access opportunity enables every node between 0 and UMAX to have zero, one or many opportunities to send unscheduled data depending on how much of the NUT remains after the completion of the scheduled time. Variations in scheduled traffic means the opportunity to access the medium during the unscheduled time may be different for each node in every NUT.

Figure 79 shows how the permission to transmit is granted during the unscheduled time. The MAC ID of the node that goes first in the unscheduled part of the NUT is incremented by 1 for each NUT. The unscheduled token begins at the MAC ID specified in the unscheduled start register (USR) of the previous moderator DLPDU. The USR increments by one modulo (UMAX+1) each NUT. If the USR reaches UMAX before the guardband, it returns to zero and the token pass continues.

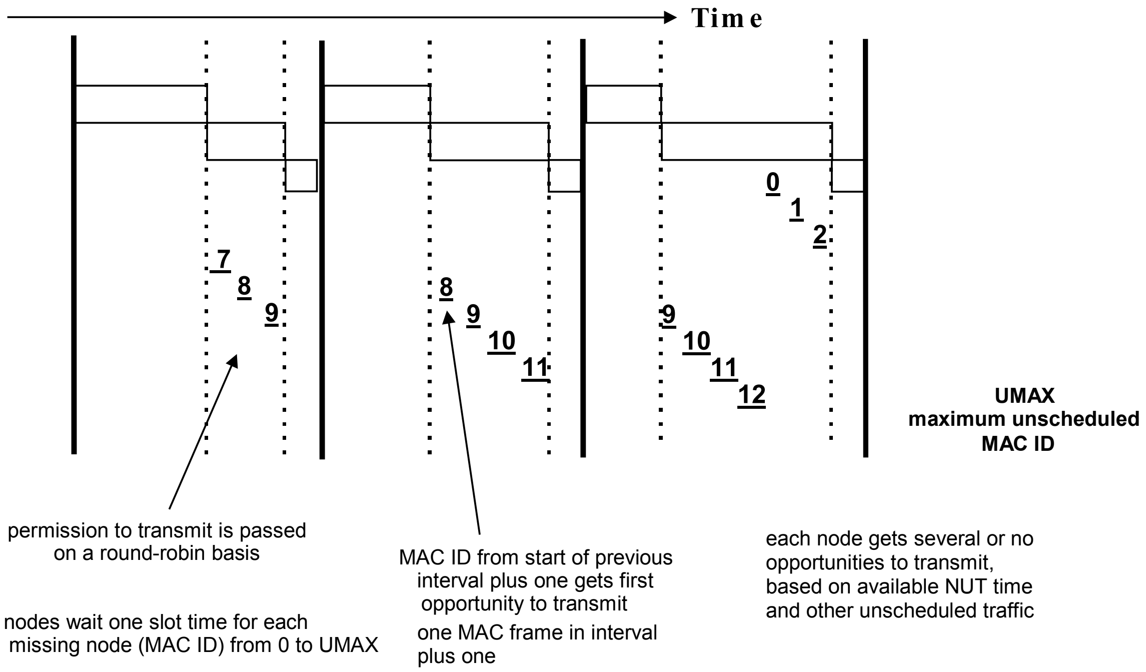


Figure 79 – Medium access during unscheduled time

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

1) it keeps the NUT timers of all nodes synchronized;
2) it publishes critical link parameters enabling all DLEs on the local link to share a common version of important local link values such as NUT, slot time, SMAX, UMAX, USR, etc.

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

Situations that may cause disruption of the DL-protocol arise due to problems in the underlying PhL service. Some examples of the types of PhL problems which can disrupt the DL-protocol are:

— induced noise within the distributed PhE;

— poor quality PhE components or installation practices;

— physically connecting two Ph-segments together while the link is operating.

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

12.1.2 DL-management services

DL-management services support:

a) setting of address filters by receiving DLS users;
b) queue maintenance support for sending DLS users;
c) local link synchronization and online change of local link parameters;
d) event reporting of important variables and events within the layer;
e) non-disruptive addition of nodes to the link;
f) tuning of link parameters;
g) time distribution and clock synchronization between nodes.

12.1.3 Timing services

This DLL is quite flexible. It can provide deterministic and synchronized I/O transfer at cyclic intervals up to 1 ms and node separations up to 25 km. This performance is adjustable online by configuring the link parameters of the local link. These parameters, which govern the access to the link, can be tuned as required to match different applications. DL-management allows these parameters to be changed online, while the local link is operating, it also allows the local link to continue functioning while connections to new nodes are added and removed.

DLEs can maintain clock synchronization across the extended link with a precision better than 10 µs.

12.2 Facilities of the Data Link Service

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

a) A means of transferring DLSDUs of limited length between two or more DLS-users who have negotiated peer or multipoint connection-mode services, see Figure 80.

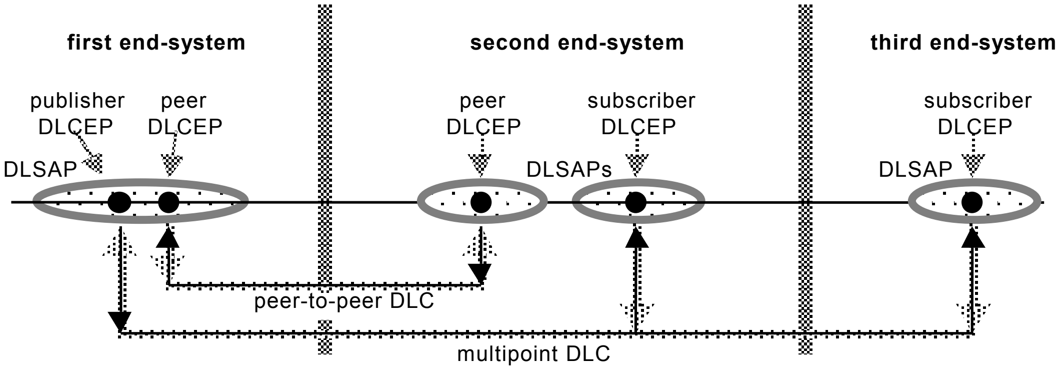


Figure 80 – Queue model for the peer and multipoint DLS, DLSAPs and their DLCEPs

b) A means of maintaining time synchronization for service execution and cyclic transfer of DLSDUs based on selected QoS parameters.

c) A means of transferring DLSDUs of limited length from one source DLSAP to a destination DLSAP or group of DLSAPs, without establishing or later releasing 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. QoS for this transmission can be selected by the sending DLS-user.

NOTE The length of a DLSDU is limited because of internal mechanisms employed by the DL-protocol

d) A means by which the status of dispatch to the destination DLSAP or group of DLSAPs can be returned to the source DLSAP.

e) A means of canceling either a specific outstanding DLSDU transfer service request, or all outstanding DLSDU transfer service requests of a specified QoS.

12.3 Model of the Data Link Service

12.3.1 General

This clause of IEC 61158 uses the abstract model for a layer service defined in ISO/IEC 10731, 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.

12.3.2 DLS-instance identification

A DLS-user is able to distinguish among several DLCEPs at the same DLSAP. This is done by an address structure named generic-tag and supported by address filtering services available to each receiving DLS-user.

For connectionless service, a DLS-user is able to distinguish among several DLSAPs using an address structure named fixed-tag. Address filtering services are available for each receiving DLS-user.

A local identification mechanism is provided for each use of the DLS which needs to correlate a confirmation or subsequent cancellation request with its associated request.

12.3.3 Model of abstract queue concepts

12.3.3.1 General

After establishment of the DLC using a generic-tag address, there exists a relationship between the publishing DLS-user and the subscribing DLS-user(s).

DL services using a fixed-tag address do not need establishment as they use pre-defined fixed relationships between permanent DLSAPs associated with each DLS-user.

As a means of specifying these relationships, an abstract queue model of a multipoint DLC, which is described in the following subclauses, is used.

NOTE 1 Establishment and management of a DLC and its identifying generic-tag is provided by higher layer entities above the DLS-interface.

NOTE 2 The internal mechanisms that support the operation of the DLS are not visible to the DLS-user.

12.3.3.2 Queue model concepts

The queue model represents the operation of a multipoint DLS in the abstract by a set of abstract queues linking the sending DLSAP-user with the receiving DLSAP-user(s) — one queue per receiving DLSAP (see Figure 81).

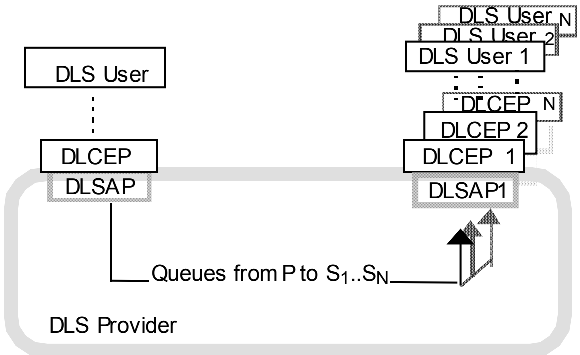


Figure 81 – Queue model of a multipoint DLS between a sending DLS-user and one or more receiving DLS-users

Each queue represents one direction of transfer. The ability of a sending or receiving DLS-user to remove objects from a queue is determined by the behavior of the DLS provider.

DLSDU objects identified by DL-generic-tag primitives or DL-fixed-tag primitives and their parameters may be placed in the abstract queue by the sending DLS-user and will be delivered to receiving DLS-users as determined by the DLSDU object's associated address and QoS parameters.

Queue management services are available to the sending DLS-user for flushing unsent objects from a transmit queue. These may be either identified individual objects or all objects loaded at a specific QoS.

12.3.4 QoS features

12.3.4.1 Sending priority and timing

The available QoS options for the connection-mode and connectionless-mode services are sending priority and timing.

The choice of sending priority implicitly selects the timing characteristics of the DLS supplier execution of the transmission. Three alternative priorities are available: scheduled, high and low.

NOTE 1 To ensure guaranteed access, the active master keeper uses scheduled priority for regular publication of a TUI fixed tag message containing the current Table Unique Identifier (TUI). The TUI is a unique reference to the current link and node configuration parameters. All participating DLEs receive the TUI and use it to ensure their link details are current.

NOTE 2 High and low priorities are recommended for all connectionless-mode services except those involved with TUI messages.

NOTE 3 High and Low priority are used only in a local sense to set the order of servicing locally submitted DLS-user-data; they do not have link-wide connotations.

12.3.4.2 Scheduled priority

This QoS provides accurate time-based cyclic and acyclic sending of DLSDUs. The execution timing for this scheduled service can be accurate and repeatable to better than 1 ms.

12.3.4.3 High priority

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

12.3.4.4 Low priority

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

12.3.5 DLS-TxStatus

This parameter allows a sending DLS-user to determine the status of a corresponding requested transmission. The value conveyed in this parameter is as follows:

a) "OK" — success — message successfully sent;

b) "TXABORT" — failure — sending process failed;

c) "FLUSHED" — failure — message has been removed from the pending queue before being sent.

NOTE 1 The FLUSHED status is only used in response to the Queue maintenance service of subclause 12.7.

NOTE 2 The parameter value OK is not an indication that the message has been received

12.3.6 Receive queues

The receiving DLS-user has an implicit queue of indeterminate capacity which is used as the receive queue, and the DLSDU is delivered as the DLS-user-data parameter of the associated indication primitive.

If it is not possible to append the received DLSDU to the receive queue, then the DLSDU is discarded and an indication primitive is not issued to the DLS-user.

12.4 Sequence of primitives

12.4.1 Constraints on sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in 12.5 and 12.6 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur. Other aspects of actual system operation, such as PhL problems affecting messages in transit, will affect the ability of a DLS-user or a DLS provider to issue a primitive at any particular time.

The connection-mode and connectionless-mode primitives and their parameters are summarized in Table 37.

Table 37 – Summary of connection-mode and connectionless-mode primitives and parameters

Service	Service subtype	Primitive	Parameter
Data Transfer	**Connection-mode**	DL-GENERIC-TAG request	(*in* request DLS-user-identifier, DLS-user-data, DLS-QoS, DLS-generic-tag)
		DL-GENERIC-TAG indication	(*out* DLS-user-data, DLS-generic-tag)
		DL-GENERIC-TAG confirm	(*out* DLS-TxStatus)
	Connectionless-mode	DL-FIXED-TAG request	(*in* request DLS-user-identifier, DLS-user-data, DLS-QoS, DLS-fixed-tag, DLS-destination-DLE-ID)
		DL-FIXED-TAG indication	(*out* DLS-user-data, DLS-fixed-tag, DLS-source-DLE-ID)
		DL-FIXED-TAG confirm	(*out* DLS-TxStatus)
Super-vision	**Queue maintenance**	DL-FLUSH-SINGLE-REQUEST request	(in request DLS-user-identifier)
		DL-FLUSH-SINGLE-REQUEST confirm	(*out* DLS-TxStatus)
		DL-FLUSH-REQUESTS-BY-QOS request	(*in* DLS-QoS)
		DL-FLUSH-REQUESTS-BY-QOS confirm	<none>
	Tag filter	DL-ENABLE-TAG request	(*in* DLS-tag)
		DL-ENABLE-TAG confirm	(*out* DLS-result)
		DL-DISABLE-TAG request	(*in* DLS-tag)
		DL-DISABLE-TAG confirm	(*out* DLS-result)

NOTE 1 Request DLS-user-identifiers are locally assigned by the DLS-user and used to flush a specific request from the DLS-provider's queues.

NOTE 2 The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.

12.4.2 Relation of primitives at DLSAPs

With few exceptions, a primitive issued at one DLSAP will have consequences at one or more other DLSAPs. The relations of primitives of each type at one DLSAP to primitives at the other DLSAPs are defined in 12.5 and 12.6, and summarized in Figure 82.

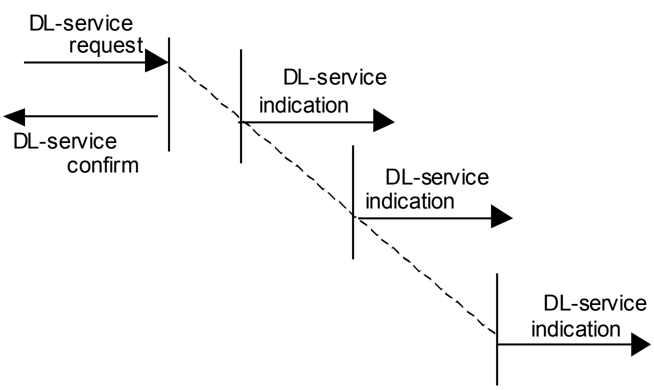


Figure 82 – DLS primitive time-sequence diagram

12.4.3 Sequence of primitives at one DLSAP

The possible overall sequences of primitives at a DLSAP are defined in the state transition diagram shown in Figure 83. In the diagram, the use of a state transition diagram to describe the allowable sequences of service primitives does not impose any requirements or constraints on the internal organization of any implementation of the service.

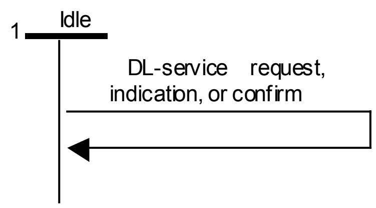


Figure 83 – State transition diagram for sequences of DLS primitives at one DLSAP

12.5 Connection-mode data transfer

12.5.1 General

DL-connection-mode service primitives can be used to transmit DLSDUs from one DLSAP to one or more peer DLSAPs using a generic-tag address to identify a connection between DLS-users . Each DLSDU is transmitted in a single DLPDU. All the information required to deliver the DLSDU is presented to the DLS provider, together with the user data to be transmitted, in a single service access.

DLS-users which are higher layer protocol entities can provide negotiation and management of connections above the DLL through additional interpretation of the DLS-generic-tag.

No means are provided by which the receiving DLS-user may control the rate at which the sending DLS-user may send DLSDUs. This is managed externally by appropriate scheduling tools which match the capability of sending and receiving DLS users and the configured service schedule of the DLS provider.

12.5.2 Function

This service provides the facilities of 12.2 a), b), c), d) and e). It can be used to transmit a DL-connection-mode DLSDU from one DLSAP to another or to a group of DLSAPs, in a single service access.

NOTE Delivery status if required is provided by higher-layer services provided by the DLS-user, it is not returned as part of the local DLS invocation.

In the absence of errors, the DLS provider maintains the integrity of individual DLSDUs, and delivers them to the receiving DLS-users in the order in which they are presented by the sending DLS-user.

12.5.3 Types of primitives and parameters

12.5.3.1 Primitive specifications

Table 38 indicates the types of primitives and the parameters needed for the DL-connection-mode transmission service.

Table 38 – DL-connection-mode transfer primitives and parameters

DL-GENERIC-TAG Parameter name	Request input	Indication output	Confirm output
Request DLS-user-identifier (handle)	M		
DLS-user-data (packet)	M	M(=)	
DLS-QoS (priority)	M		
DLS-generic-tag	M	M(=)	
DLS-TxStatus			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

12.5.3.2 Request DLS-user-identifier

This parameter, which is specified by the DLS-user on cancelable DL-request primitives, provides a local means by which the DLS-user can subsequently attempt to cancel that request through a DL-FLUSH-SINGLE queue maintenance request. The naming-domain of this identifier is the DLS-user-local-view.

12.5.3.3 DLS-user data

This parameter provides the data to be transmitted between DLS-users without alteration by the DLS provider. The initiating DLS-user may transmit any integral number of octets greater than zero, up to the limit determined by the service type parameter specified in the service request.

12.5.3.4 DLS-QoS

This parameter is specified in 12.3.4.

12.5.3.5 DLS-generic-tag

This parameter conveys a connection identification or DLSAP-address identifying the remote DLSAP(s) to which the DLS is to be provided. It is a DL(SAP)-address in the request primitive, but takes the form of a local DL(SAP)-address DLS-user-identifier in the indication primitive(s). It may be a DLSAP-address or a multi-cast DL-address.

12.5.3.6 Request primitive

If the initiating DLS-user has implemented a FIFO queue of maximum depth K as a source queue for the DLSAP-address at the specified QoS priority, then a DL-GENERIC-REQUEST primitive attempts to append a DLSDU to the queue, but fails if the queue already contains K DLSDUs. If the append operation is successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

NOTE 1 The queue provides a means of managing multiple DLS-user requests for the efficiency advantage of combining them in a single transmission opportunity.

NOTE 2 The queue depth K is implementation specific.

12.5.3.7 Indication primitive for DLSDUs associated with generic tags

The receiving DLS-user is able to identify Generic Tag values of interest to it and pass them to the local DLS provider using the DLS-tag-filter management services. The set of local tag values are used to filter arriving associated DLSDUs. For DLSDUs with associated Generic tags that are acceptable to the filter, the following indication parameters are delivered to the local DLS-user.

— DLS-user-data;

— DLS-generic-tag, the value of the generic tag associated with the DLSDU.

12.5.3.8 DLS-TxStatus

This parameter is specified in 12.3.5.

12.5.4 Sequence of primitives

The sequence of primitives in a successful or unsuccessful generic-tag transfer is defined in the time-sequence diagrams in Figure 84 and Figure 85.

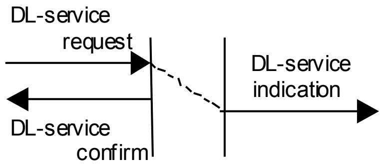


Figure 84 – Sequence of primitives for a successful connection-mode transfer

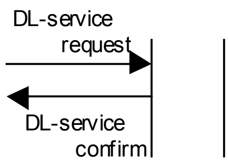


Figure 85 – Sequence of primitives for an unsuccessful connection-mode transfer

12.6 Connectionless-mode data transfer

12.6.1 General

DL-connectionless-mode service primitives can be used to transmit independent DLSDUs from one DLSAP to another DLSAP using a fixed-tag address to identify the destination DLSAP. Each DLSDU is transmitted in a single DLPDU. The DLSDU is independent in the sense that it bears no relationship to any other DLSDU transmitted through an invocation of the DLS. The DLSDU is self-contained in that all the information required to deliver the DLSDU is presented to the DLS provider, together with the user data to be transmitted, in a single service access.

No means are provided by which the receiving DLS-user may control the rate at which the sending DLS-user may send DLSDUs. This is managed externally by appropriate scheduling tools which match the capabilities of sending and receiving DLS-users with the configured service schedule of the DLS provider.

12.6.2 Function

This service provides the facilities of 12.2 b), c), d) and e). It can be used to transmit an independent, self-contained DLSDU from one DLSAP to a group of DLSAPs, all in a single service access. Delivery status is not returned as part of the local DLS invocation.

A DLSDU transmitted using DL-connectionless-mode data transfer is not considered by the DLS provider to be related in any way to any other DLSDU. In the absence of errors, it maintains the integrity of individual DLSDUs, and delivers them to the receiving DLS-users in the order in which they are presented by the sending DLS-user.

12.6.3 Types of primitives and parameters

12.6.3.1 Primitive specifications

Table 39 indicates the types of primitives and the parameters needed for the DL-connectionless-mode transmission service.

Table 39 – DL-connectionless-mode transfer primitives and parameters

DL-FIXED-TAG	REQUEST	INDICATION	CONFIRM
Parameter name	input	output	output
Request DLS-user-identifier (handle)	M		
DLS-user-data (packet)	M	M(=)	
DLS-QoS (priority)	M		
DLS-fixed-tag	M	M(=)	
DLS destination-DLE-ID (station MAC ID)	M		
DLS-source-DLE-ID (station MAC ID)		M	
DLS-TxStatus			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

12.6.3.2 Request DLS-user-identifier

This parameter is specified in 12.5.3.2.

12.6.3.3 DLS-user data

This parameter provides the data to be transmitted between DLS-users without alteration by the DLS provider. The initiating DLS-user may transmit any integral number of octets greater than zero, up to the limit inherent for the specified service.

12.6.3.4 DLS-QoS

This parameter is specified in 12.3.4.

NOTE DLS-scheduled-priority is generally reserved for generic-tag connection-mode services. The only normal exception is for periodic TUI fixed tag messages published by the master keeper to ensure that all DLS providers share a common sense of link parameters.

12.6.3.5 DLS-fixed-tag

This parameter specifies the destination DLSAP in the DLE identified by the DLS-destination-DLE-ID address. The DLSAP to be used is selected from the set of Fixed Tag service types available in the destination DLE.

The set of Fixed-tag services available to the DLS-user are listed in Table 40.

Table 40 – Fixed tag services available to the DLS-user

Fixed tag service code (hexadecimal)	**Meaning of service**
01 — 08	vendor specific
09	ping request
0A – 14	vendor specific
15	tMinus
16 — 28	vendor specific
29	ping reply
2A — 3F	vendor specific
70 – 7F	vendor-specific
83	UCMM
88	Keeper UCMM
8C	Time distribution
F0 — FF	vendor specific

Fixed-tag service codes in the vendor-specific range may be assigned by the DLS-user.

The UCMM fixed tag is reserved for DLS-users wishing to send messages via the Unconnected Message Manager object in the destination DLE.

The Keeper UCMM fixed tag is reserved for DLS-users wishing to send messages via the Keeper Unconnected Message Manager object in the destination DLE.

Specific uses for other fixed tags in the table are presented in the remainder of this clause.

NOTE All other fixed tags are reserved or used internally by the DLS provider.

12.6.3.6 DLS-destination-DLE-ID

This parameter conveys the node DL-address of the destination node; it is a MAC ID address.

12.6.3.7 Request primitive

If the initiating DLS-user has implemented a FIFO queue of maximum depth *K* to the DLSAP-address at the specified priority as a source, then a DL-request primitive attempts to append a DLSDU to the queue, but fails if the queue already contains *K* DLSDUs. If the append operation is successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue. The queue serves to assemble multiple DLS-user requests for the efficiency advantage of combining them in a single transmission opportunity for the specified QoS or better.

NOTE The queue depth *K* is implementation specific.

12.6.3.8 Indication primitives

12.6.3.8.1 General

The receiving DLS-user has an implicit queue of indeterminate capacity which is used as the receive queue, and the DLSDU is delivered as the DLS-user-data parameter of the associated indication primitive.

If it is not possible to append the received DLSDU to the receive queue, then the DLSDU is discarded and an indication primitive is not issued to the DLS-user.

12.6.3.8.2 Indication for fixed tag DLSDUs

The receiving DLS-user is able to identify a number of Fixed Tag values of interest to it and pass them to the local DLS provider using the DLS-tag-filter management services. The set of local tag values are used to filter associated arriving DLSDUs. For DLSDUs with associated Fixed tags that are acceptable to the filter, the following indication parameters are delivered to the local DLS-user.

— DLS-user-data;

— DLS-fixed-tag, the value of the fixed tag service code associated with the DLSDU;

— DLS-source-DLE-ID, the source DLE MAC ID.

12.6.3.8.3 DLS-source-DLE-ID

This parameter conveys an address identifying the local DLE from which the fixed tag DLSDU has been sent. It is a DLE MAC ID address on the local link.

12.6.3.9 DLS-TxStatus

This parameter is specified in 12.3.5.

12.6.4 Sequence of primitives

The sequence of primitives in a successful or unsuccessful fixed-tag transfer is defined in the time-sequence diagrams in Figure 86 and Figure 87.

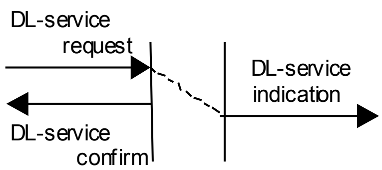


Figure 86 – Sequence of primitives for a successful connectionless-mode transfer

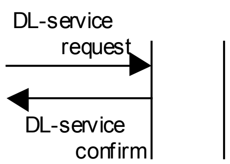


Figure 87 – Sequence of primitives for an unsuccessful connectionless-mode transfer

12.7 Queue maintenance

12.7.1 Function

DLS-send requests are held in a pending queue by the DLS provider until the requested sending opportunity is available. This queue is not visible to the DLS-user. To support efficient operation, the queue maintenance service is provided to de-queue pending requests that have not been sent.

12.7.2 Types of primitives and parameters

12.7.2.1 Primitive specifications

Table 41 indicates the primitives and parameters of the DL-queue maintenance service. This is a local service at each DLSAP.

Table 41 – DL-queue maintenance primitives and parameters

DL-FLUSH SINGLE-REQUEST **Parameter name**	**Request** input	**Confirm** output
request DLS-user-identifier (handle)	M	
DLS-TxStatus		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

DL-FLUSH REQUESTS-BY-QOS **Parameter name**	**Request** input	**Confirm** output
DLS-QoS (priority)	M	
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

12.7.2.2 request DLS-user-identifier and DLS-QoS

The request DLS-user-identifier and DLS-QoS parameters have the same meanings as specified in 12.5. Their purpose in these primitives is to identify the set of requests, or the single request, which is to be flushed from the request queue if they have not yet been irrevocably committed for transmission.

12.7.2.3 DLS-TxStatus

The DLS-TxStatus parameter has the same meaning and purpose as specified in 12.5.

12.7.3 Request primitive

When used with a DL-FLUSH REQUESTS-BY-QOS request, all untransmitted transfers at that QoS priority are cancelled.

When used with a DL-FLUSH SINGLE-REQUEST request, only the specified individual transfer is cancelled.

12.7.4 Confirmation primitive

12.7.4.1 DL-Flush-single-request

When the single pending transfer identified by request DLS-user-id has been cancelled, the confirmation for the original transfer request (DL-GENERIC-TAG or DL-FIXED-TAG) is returned with the DLS-TxStatus specifying the value FLUSHED.

12.7.4.2 DL-Flush-requests-by-QoS

When all pending transfers of the specified QoS have been cancelled, a confirmation is returned.

12.7.5 Sequence of primitives

The sequence of primitives for a queue maintenance request are defined in the time sequence diagrams of Figure 88

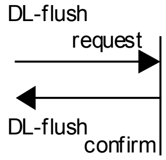


Figure 88 – Sequence of primitives for a queue maintenance request

12.8 Tag filter

12.8.1 Function

By default, the receiving DLS provider accepts and processes only the DLS-fixed-tag messages which have the fixed-tag value of 00 (moderator tag) and all other messages are discarded.

The tag filter service allows the DLS user to enable or disable reception of other messages based on the contents of their DLS parameter tag.

The DLS provider will deliver incoming messages to the DLS-user only for DLS-tags that have been enabled.

12.8.2 Types of primitives and parameters

12.8.2.1 Primitive specifications

Table 42 indicates the primitives and parameters of the DL-connectionless-mode queue maintenance service. This is a local service at each DLSAP.

Table 42 – DL-connectionless-mode tag filter primitives and parameters

DL-ENABLE-TAG DL-DISABLE-TAG	**Request**	**Confirm**
Parameter name	input	output
DLS-tag	M	
DLS-result		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

12.8.2.2 request DLS-user-identifier and DLS-tag

These parameters have the same meanings and purpose as specified in 12.5. The DLS-tag can be either a DLS-generic-tag of a DLS-fixed-tag.

12.8.2.3 DLS-result

This parameter conveys the status of the corresponding request:

a) TRUE — the service request completed successfully;

b) FALSE — the service request failed to complete successfully.

NOTE If the DLS provider is unable to accept filtering requests for additional generic tags, the status returned will be FALSE.

12.8.3 Sequence of primitives

The sequence of primitives for a tag filter request are defined in the time sequence diagrams of Figure 89

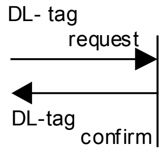


Figure 89 – Sequence of primitives for a tag filter request

13 Type 2: DL-management Services

13.1 Sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in this clause may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur. Other aspects of actual system operation, such as PhL problems affecting messages in transit, will affect the ability of a DLS-user or a DLS provider to issue a primitive at any particular time.

The DL-management primitives and their parameters are summarized in Table 43.

Table 43 – Summary of DL-management primitives and parameters

Service	Service subtype	Primitive	Parameter
Management	**Local link synchronization**	DLM-TONE INDICATION	(*out* DLMS-cycle)
	Synchronized parameter change	DLM-SET-PENDING request	(*in* DLMS-configuration-data)
		DLM-SET-PENDING confirm	(*out* DLMS-result)
		DLM-GET-PENDING request	<none>
		DLM-GET-PENDING confirm	(*out* DLMS-configuration-data)
		DLM-SET-CURRENT request	(*in* DLMS-configuration-data)
		DLM-SET-CURRENT confirm	(*out* DLMS-result)
		DLM-GET-CURRENT request	<none>
		DLM-GET-CURRENT confirm	(*out* DLMS-configuration-data)
		DLM-TMINUS-START-COUNTDOWN request	(*in* DLMS-start-count)
		DLM-TMINUS-START-COUNTDOWN confirm	(*out* DLMS-result)
		DLM-TMINUS-ZERO indication	<none>
	Event reports	DLM-EVENT indication	(*out* DLMS-event, DLMS-source-DLE-ID)
	Bad FCS	DLM-BAD-FCS indication	(*out* DLMS-channel)
	Current moderator	DLM-CURRENT-MODERATOR indication	(*out* DLMS-source-DLE-ID)
	Enable moderator	DLM-ENABLE-MODERATOR request	(*in* DLMS-enable-moderator)
		DLM-ENABLE-MODERATOR confirm	(*out* DLMS-enable-moderator)
	Power-up and Online	DLM-POWER-UP indication	<none>
		DLM-ONLINE request	(*in* DLMS-online)
		DLM-ONLINE confirm	(*out* DLMS-online)
	Listen only	DLM-LISTEN-ONLY request	(*in* DLMS-listen only)
		DLM-LISTEN-ONLY confirm	(*out* DLMS-listen only)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

13.2 Link synchronization

13.2.1 Function

The scheduled QoS is based on a repeating cycle of DLS transmission opportunities which are time locked to better than 1 ms. The basic time interval is the NUT or Network Update Time and an incrementing count is maintained for each NUT within the repeating cycle. This service indicates to the DLMS-user the current NUT count within the cycle.

13.2.2 Types of primitives and parameters

13.2.2.1 Primitive specifications

Table 44 indicates the primitives and parameters of the Link synchronization service. This is a local service.

Table 44 – Link synchronization primitives and parameters

DLM-TONE	Indication
Parameter name	output
DLMS-cycle	M

13.2.2.2 DLMS-cycle

This parameter indicates the interval count for the NUT which has just been received within the overall cycle of scheduled access intervals. The DLS provider uses internal timing facilities to simulate this indication if expected moderator DLPDUs are not available.

13.2.3 Sequence of primitives

The sequence of primitives for a link synchronization are defined in the time sequence diagrams of Figure 90

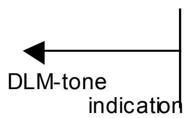


Figure 90 – Sequence of primitives for a local link synchronization

13.3 Synchronized parameter change

13.3.1 Function

All DLEs maintain two local copies of DLMS-configuration-data parameters: current and pending. The current copy is used for the ongoing operation of the DLS. The pending copy is maintained to allow a synchronized change of DLS configuration parameters. This service manages these DLMS-configuration-data parameters and their changeover.

At the system management level, a required set of DLMS-configuration-data parameters and the count down trigger for a change-over are distributed to all DLMS-users using data transmit services and fixed tags (link parameters tag and tMinus tag).

The synchronized parameter change service enables each DLMS-user to transfer required configuration-data values to the local DLS provider.

The moderator fixed tag DLPDU contains a parameter, called tMinus, that counts down to zero as a trigger to synchronize the change-over from current to pending sets of the DLS configuration parameters. The DLM-TMINUS-START-COUNTDOWN request from a DLMS-user causes its local DLS provider to participate in a tMinus countdown, and, if the node is the moderator, it initializes the tMinus parameter of the moderator. The moderator decrements this parameter count before transmitting each moderator DLPDU until the parameter equals zero. When tMinus transitions from 1 to 0, each local DLS provider participating in the countdown locally generates a DLM-TMINUS-ZERO indication and copies its pending DLMS-configuration-data parameters into its current copy. If the tMinus field transitions to 0 from any value except 1, the countdown is aborted and no DLM-TMINUS-ZERO indication is generated.

13.3.2 Types of primitives and parameters

13.3.2.1 Primitive specifications

Table 45 indicates the primitives and parameters of the DLM synchronized parameter change service. This is a local service.

Table 45 – Synchronized parameter change primitives and parameters

DLM-SET-PENDING DLM-SET-CURRENT	Request	Confirm
Parameter name	input	output
DLMS-configuration-data	M	
DLMS-result		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

DLM-GET-PENDING DLM-GET-CURRENT	Request	Confirm
Parameter name	input	output
DLMS-configuration-data		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

DLM-tMINUS-START-COUNTDOWN	Request	Confirm
Parameter name	input	output
DLMS-start-count	M	
DLMS-result		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

DLM-tMINUS-ZERO	Indication
Parameter name	output
<none>	

13.3.2.2 DLMS-result

This parameter has the same meaning and purpose as specified in 12.8 for DLS-result.

13.3.2.3 DLMS-configuration-data

This parameter conveys the set of configuration data values specified in Table 46

Table 46 – DLMS-configuration-data

Subparameter	Meaning
my_addr	the MAC ID of this DLE
NUT_length	the length of the NUT in 10 μs increments
SMAX	highest MAC ID allowed to transmit scheduled
UMAX	highest MAC ID allowed to transmit unscheduled
slotTime	time allowed for Ph layer line turnaround in 1 μs increments
blanking	time to disable RX after DLPDU in 1600 ns increments
gb_start	10 μs intervals from start of guardband to tone
gb_center	10 μs intervals from start of moderator to tone
modulus	modulus of the interval counter for intervals in a cycle of NUTs
gb_prestart	transmit cut-off, 10 μs intervals before tone, may not transmit past this limit

13.3.2.4 DLMS-start-count

In all DLEs but the moderator, the presence of this parameter enables the local DLS provider to track the tMinus countdown contained in successive moderator messages. and when the count changes from 1 to 0, to change to the pending set of DLS configuration parameters previously requested by the local DLMS user. If the final tMinus transition to 0 is from any value other than 1, the change of configuration data parameters is aborted.

If the local DLE is the moderator, this parameter initializes the tMinus parameter in the moderator messages and initiates its decrementing by 1 for each successive moderator message until it reaches 0.

If the final tMinus transition is from 1 to 0, this indication is locally generated by each participating DLS provider and passed to local DL-management, which then transforms any pending link DLS configuration parameters into current parameters.

13.3.3 Sequence of primitives

The sequence of primitives for synchronized parameter change are defined in the time sequence diagrams of Figure 91 and Figure 92.

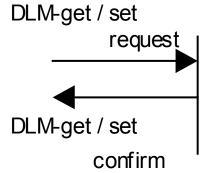


Figure 91 – Sequence of primitives for a DLM-get/set parameters request

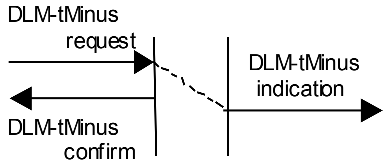


Figure 92 – Sequence of primitives for a DLM-tMinus change request

13.4 Event reports

13.4.1 Function

The event report service provides indications to DL-management about events internal to the DLS provider.

13.4.2 Types of primitives and parameters

13.4.2.1 Primitive specifications

Table 47 indicates the primitives and parameters of the event report service. This is a local service.

Table 47 – Event report primitives and parameters

DLM-EVENT Parameter name	Indication output
DLMS-event	M
DLMS-source-DLE-ID	C

13.4.2.2 DLMS-event

This parameter takes one of the values in Table 48

Table 48 – DLMS events being reported

DLMS event	Description
DLMS_EV_rxGoodFrame	A good DLPDU was received. This includes DLPDUs that contain no data (null DLPDUs), but excludes moderator DLPDUs.
DLMS_EV_txGoodFrame	A good DLPDU was transmitted. This includes DLPDUs that contain no data (null DLPDUs), but excludes moderator DLPDUs.
DLMS_EV_badFrame	A damaged DLPDU was received. The apparent source MAC ID of the transmitting DLE is reported via the optional parameter.
DLMS_EV_errA	A bad DLPDU was received on channel A of the physical medium, or a good DLPDU was received on channel B and PH-FRAME indication from channel A stayed FALSE.
DLMS_EV_errB	A bad DLPDU was received on channel B of the physical medium, or a good DLPDU was received on channel A and PH-FRAME indication from channel B stayed FALSE.
DLMS_EV_txAbort	A transmit DLPDU was terminated with an abort sequence.
DLMS_EV_NUT_overrun	NUT is not large enough to accommodate all the scheduled traffic.
DLMS_EV_dribble	Scheduled Lpackets could not be sent during scheduled time.
DLMS_EV_nonconcurrence	An event was detected that indicates that this node is out of step with the access control protocol.
DLMS_EV_rxAbort	A DLPDU was received that was terminated with an abort sequence.
DLMS_EV_lonely	Have not heard a DLPDU from another node on the link for 8 NUTs
DLMS_EV_dupNode	Another node on the link is using this node's MAC ID.
DLMS_EV_noisePulse	Ph-LOCK indication went TRUE then FALSE before PH-FRAME indication went TRUE, but Ph-LOCK indication was not TRUE long enough to indicate a possibly damaged DLPDU.
DLMS_EV_collision	PH-FRAME indication was TRUE when this node was about to transmit.
DLMS_EV_invalidModAddress	A moderator was received from a node that does not have the lowest MAC ID on the link.
DLMS_EV_rogue	A moderator DLPDU was received that does not match the link configuration information at this node.
DLMS_EV_deafness	Cannot hear the moderator DLPDU even though other link traffic is present.
DLMS_EV_supernode	A moderator was received from MAC ID 0.

13.4.2.3 DLMS-source-DLE-ID

This parameter is used in conjunction with the DLMS_EV_badFrame event to indicate the probable transmitting DLE.

NOTE As the DLPDU was damaged, the indicated DLMS-source-DLE-ID may not be correct.

13.4.3 Sequence of primitives

The sequence of primitives for an event indication are defined in the time sequence diagrams of Figure 93

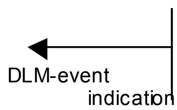


Figure 93 – Sequence of primitives for a DLM-event indication

13.5 Bad FCS

13.5.1 Function

The BAD-FCS indication service alerts the DLMS-user that a received DLPDU had an invalid frame check sequence.

13.5.2 Types of primitives and parameters

13.5.2.1 Primitive specifications

Table 49 indicates the primitives and parameters of the bad-FCS service. This is a local service.

Table **49 – Bad FCS primitives and parameters**

DLM-BAD-FCS **Parameter name**	**Indication** output
DLMS-channel	M

13.5.2.2 DLMS-channel

This parameter indicates which PhE provided the DLPDU that failed the FCS check. The parameter value is either CHANNEL-A or CHANNEL-B indicating the PhL channel on which the erroneous DLPDU was received. This indication is provided not more than once per erroneous DLPDU per channel.

13.5.3 Sequence of primitives

The sequence of primitives for a bad-FCS are defined in the time sequence diagram of Figure 94.

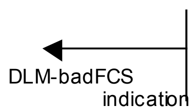


Figure 94 – Sequence of primitives for a DLM-bad-FCS indication

13.6 Current moderator

13.6.1 Function

This service informs the DLMS-user which DLE is the current moderator.

13.6.2 Types of primitives and parameters

13.6.2.1 Primitive specifications

Table 50 indicates the primitives and parameters of the current moderator indication. This is a local service.

Table 50 – Current moderator primitives and parameters

DLM-CURRENT-MODERATOR **Parameter name**	**Indication** output
DLMS-source-DLE-ID	M

13.6.2.2 DLMS-source-DLE-ID

This parameter indicates an address identifying the source DLE MAC ID for the most recently received moderator DLPDU on the local link.

13.6.3 Sequence of primitives

The sequence of primitives for a current moderator indication are defined in the time sequence diagrams of Figure 95

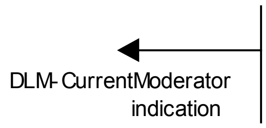


Figure 95 – Sequence of primitives for a DLM-current-moderator indication

13.7 Enable moderator

13.7.1 Function

This service enables the DLMS user to enable and disable the ability of its local DLS provider to join the group of DLEs that co-operate in assigning one of its members to the role of current moderator

13.7.2 Types of primitives and parameters

13.7.2.1 Primitive specifications

Table 51 indicates the primitives and parameters of the enable moderator service. This is a local service.

Table 51 – Enable moderator primitives and parameters

DLM-ENABLE-MODERATOR Parameter name	Request input	Confirm output
DLMS-enable-moderator	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

13.7.2.2 DLMS-enable-moderator

This parameter takes values TRUE and FALSE which respectively enable or disable the moderator capability in the local DLS provider.

13.7.3 Sequence of primitives

The sequence of primitives for enable moderator are defined in the time sequence diagrams of Figure 96

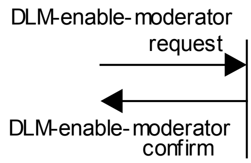


Figure 96 – Sequence of primitives for a DLM-enable-moderator request

13.8 Power-up and online

13.8.1 Function

This service enables the DLMS-user to request its local DLS provider to enter an online state or an offline state.

13.8.2 Types of primitives and parameters

13.8.2.1 Primitive specifications

Table 52 indicates the primitives and parameters of the power-up and online services. This is a local service.

Table 52 – Power-up and online primitives and parameters

DLM-POWER-UP Parameter name	Indication output
<none>	

DLM-ONLINE Parameter name	Request input	Confirm output
DLMS-online	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

13.8.2.2 DLM-power-up

The DLM-power-up indication notifies that the Data Link Layer has completed its initialization.

NOTE Following this initialization indication the DL user can continue the process of going on line by sending I'm alive fixed tag messages to inform all other DL users.

13.8.2.3 DLM-online

At power-up, the local DLS provider waits until the request DLMS-online parameter equals TRUE. The DLS provider then begins the process of going online and reports TRUE or FALSE, representing success or failure transition to online, respectively, via the confirmation parameter.

When the request parameter DLMS-online is FALSE, the local DLS provider goes offline at the end of the current NUT, and reports back the new status via the confirmation parameter. When offline, the local DLS provider does not transmit, and ignores any link activity.

13.8.3 Sequence of primitives

The sequence of primitives for power-up and online are defined in the time sequence diagrams of Figure 97 and Figure 98.

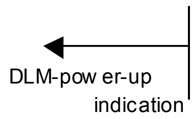


Figure 97 – Sequence of primitives for a DLM-power-up indication

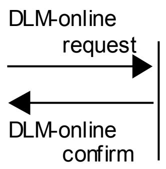


Figure 98 – Sequence of primitives for a DLM-online request

13.9 Listen only

13.9.1 Function

This service enables the DLMS user to enable and disable the ability of its local DLS provider to transmit.

13.9.2 Types of primitives and parameters

13.9.2.1 Primitive specifications

Table 53 indicates the primitives and parameters of the listen only service. This is a local service.

Table 53 – Listen-only primitives and parameters

DLM-LISTEN-ONLY Parameter name	Request input	Confirm output
DLMS-listen-only	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

13.9.2.2 DLMS-listen-only

This parameter takes values TRUE and FALSE which respectively enable or disable the DLPDU transmission capability in the local DLS provider. When the enable parameter is FALSE, transmission of DLPDUs is disabled, however the ability of the DLE to receive DLPDUs is not affected.

13.9.3 Sequence of primitives

The sequence of primitives for listen only are defined in the time sequence diagrams of Figure 99

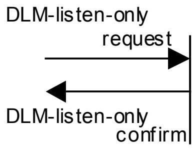


Figure 99 – Sequence of primitives for a DLM-listen-only request

13.10 Time distribution

13.10.1 Function

The moderator DLPDU provides a common reference marker that is synchronized between all nodes on the local link. By distributing and processing time stamps relative to the reference instead of to the time distribution message, implementations are simplified whilst accuracy is improved by several orders of magnitude. Phase and frequency synchronization is inherent in this DL-protocol to a very high level of accuracy. The accuracy of time synchronization using the time distribution format defined in this clause is implementation dependent, however it can be better than 10 μs.

13.10.2 Types of primitives and parameters

13.10.2.1 General

Table 54 is a summary of the DLMS time and time quality parameters sent as DLMS-user data by the time distribution service.

Table 54 – DLMS time and time quality parameters

Subparameter	Meaning
revision	revision of time distribution format
leap	leap second offset
goodness	time relay control field
gse	global squared error relative to ultimate master
dctz	distribution channel time zero
ts_ref	time stamp of previous reference pulse
ts_tx	time stamp of this message's transmission

13.10.2.2 revision

This parameter shall have the value zero. This parameter represents the revision of the time distribution format.

13.10.2.3 leap

This parameter specifies the Universal Coordinated Time (UTC) leap seconds. This number, when added to the system time, shall give actual UTC time. This number takes unpredictable jumps as dictated by the US Naval Observatory. If zero, then the number of leap seconds is unknown.

NOTE This parameter should not be used in any control situations, but may be needed in some time relays to distribution channels that are based on UTC rather than Global Positioning Satellite System (GPSS) time.

13.10.2.4 goodness

13.10.2.4.1 Definition

This compound parameter specifies the source quality of the distributed time and the number of hops in the time distribution path.

13.10.2.4.2 source quality

This subparameter indicates the quality of the source of the distributed time as shown in Table 55.

Table 55 – Time distribution source quality

Value	Meaning
7	Absolute system time — acting as master
6	Absolute system time — acting as dependent
5	Human set system time — acting as master
4	Human set system time — acting as dependent
3	Lock established to node on distribution channel other than this one
2	Lock established to node on this channel; system time unknown
1	(invalid)
0	Not synchronized with any other node

13.10.2.4.3 stratum

This parameter specifies the number of time relays between this message and a source of absolute time. A value of 0 signifies an exact reference, and the value is incremented for every intermediate time relay. If the priority field is set to zero (lock not achieved), or the number of intermediate time relays exceeds 15, the ctrl parameter is set to 15. Bits 3 through 11 are reserved and shall have the value zero.

NOTE A time relay is a DL-router which distributes time synchronization messages on its connected links based on the time synchronization messages received on its other links.

13.10.2.5 gse

This parameter indicates the cumulated rms stability squared. This parameter should approximate the node's worst-case stability relative to the rest of the system. The units of this parameter are $(100\ ns)^2$. When the rms stability is unknown or not yet determined, the value for this parameter is $FFFFFFFF_{16}$.

13.10.2.6 dctz

This parameter indicates the system time offset from the distribution channel's arbitrary time zero, established when the distribution channel and local link are synchronized. The unit of measurement is 100 ns.

13.10.2.7 ts_ref

This parameter indicates the time stamp of the last tone following a moderator DLPDU which had its interval_count equal to zero. The value of zero indicates that this value is not known. System time zero is defined as that originally used for the Global Positioning Satellite System: 12:00 midnight, Jan 6, 1980 GMT. The unit of measurement is 100 ns.

NOTE This DLL does not use GPS time, it uses only the original time zero point for GPS. So the 1999 roll-over of GPS time has no relevance to system time.

13.10.2.8 ts_tx

This parameter indicates the time stamp at the transmission of this message. The value of zero indicates that this value is not known. System time zero is defined in 13.10.2.7.

14 Type 3: Connectionless-mode Data Link Service

14.1 General

This clause describes the interface between a DLE and a Data Link service user (DLS-user). The services of this interface are typical of those needed in application fields such as process control, factory automation, power distribution, building automation and other primary process industries:

– General purpose data transfer service

– Time transfer service

14.2 Model of the connectionless-mode Data Link Service

14.2.1 Overview

This subclause describes the abstract model for data and time transfer services. The model defines interactions between the DLS-user and the DLL that take place at the DLSAPs. Information is passed between the DLS-user and the local DLE by DLS primitives and their associated parameters.

The DLS-user is provided with the following data and time transfer services:

– Acknowledged connectionless data transfer:
  Send Data with Acknowledge (SDA)

– Unacknowledged connectionless data transfer:
  Send Data with No Acknowledge (SDN)

– Two-way connectionless data exchange:
  Send and Request Data with Reply (SRD)

– M-way connectionless data exchange:
  Send and Request Data with Multicast Reply (MSRD)

– Acknowledged connectionless time transfer:
  Clock Synchronization (CS)

These services permit a DLS-user in a master station, called the local DLS-user, to send DLS-user data or time information (a DLSDU) to a DLS-user, called the remote DLS-user, at either a single remote station (SDN, SDA, SRD, MSRD) or at all remote stations (SDN, CS).

Two of these services (SRD and MSRD) permit a DLSDU to be returned by that single remote station (in an immediate reply) as part of a single transaction. These same two services can be used to retrieve a DLSDU from that remote station without first sending a DLSDU. Additionally, the MSRD service permits a DLSDU to be returned by the remote station as a multicast message.

NOTE All of these services are considered optional.

14.2.2 Acknowledged connectionless data transfer: Send Data with Acknowledge (SDA)

This service permits the local DLS-user to send a DLSDU to a single remote station. At the remote station the DLSDU, if the respective DLPDU is transferred error-free, is delivered by the remote DLE to its local DLS-user. The originating local DLS-user receives a confirmation concerning the receipt or non-receipt of the DLSDU by the remote DLS-user. If an error occurred during the transfer, the originating DLE repeats the data transfer up to a configured maximum number of times.

14.2.3 Unacknowledged connectionless data transfer: Send Data with No Acknowledge (SDN)

This service permits a local DLS-user to transfer a DLSDU to a single remote station (Unicast), or to all other remote stations (Broadcast) at the same time. The local DLS-user receives a confirmation acknowledging the completion of the transfer, but not whether the DLPDU was duly received. At each addressed remote station this DLSDU, if the respective DLPDU is received error-free, is delivered to a single local DLS-user (Unicast), to the appropriate set of local DLS-users (Multicast), or to all local DLS-users (Broadcast) . There is no confirmation to the sending DLS-user that such an intended delivery has taken place.

14.2.4 Two-way connectionless data exchange: Send and Request Data with Reply (SRD)

This service variant permits a local DLS-user to transfer a DLSDU to a DLS-user at a single remote station and as part of the same transaction, to transfer to the requesting DLS-user either a DLSDU that was previously made available by the remote DLS-user, or a status that a DLSDU is not available or that an error has been detected. At the remote station the received DLSDU, if the respective DLPDU is error-free, is delivered to the remote DLS-user. The service permits the local DLS-user to specify a null DLSDU, thereby requesting a DLSDU from the remote DLS-user without concurrently transferring a DLSDU to the remote DLS-user.

The local DLS-user receives either the requested DLSDU, or a notification that no DLSDU was available, or a notification of the type of error that was detected. The first two alternatives also confirm the receipt by the remote DLS-user of the DLSDU sent by the initiating local DLS-user.

If an error occurs during the transmission, the local DLE repeats (as part of the same transaction) the transmission of the initiator’s DLSDU, if any, including the request for a returned DLSDU, up to a configured maximum number of times.

14.2.5 M-way connectionless data exchange: Send and Request Data with Multicast-Reply (MSRD)

This service permits a local DLS-user to transfer a DLSDU to a DLS-user at a single remote station and as part of the same transaction, to transfer to the requesting DLS-user and to the appropriate set of remote DLS-users (Multicast-Reply) a DLSDU that was previously made available by the remote DLS-user. If a DLPDU is not available by the remote DLS-user, or an error has been detected the requesting DLS-user receives a status. At the addressed remote station the received DLSDU, if the respective DLPDU is error-free, is delivered to the remote DLS-user. The service permits the local DLS-user to specify a null DLSDU, thereby requesting a DLSDU from the remote DLS-user without concurrently transferring a DLSDU to the remote DLS-user.

The local DLS-user and the appropriate set of remote DLS-users receive the data requested by the local DLS-user, or the local DLS-user only receives either a notification that no data was available or a notification of the type of error that was detected. The first two alternatives also confirm the receipt by the remote DLS-user of the DLSDU sent by the initiating local DLS-user. There is no guarantee of correct receipt of the requested DLPDU (Multicast-Reply) at all other remote DLS-users; acknowledgement does not occur.

If an error occurs during the transmission, the local DLE repeats (as part of the same transaction) both the transmission of the initiator’s DLSDU, if any, and the request for a returned DLSDU, up to a configured maximum number of times.

14.2.6 Clock Synchronization (CS)

This service sequence permits the local DLS-user of the time master to distribute a DLSDU to all remote time receivers.

As part of the service sequence the time master transmit a time event message at first. Upon reception of a CS time event request the local DLE of the time master measures the send delay time between reception of the request and transmitting of the appropriate DLPDU while the remote DLEs start the measurement of the receiving delay after reception of this DLPDU.

Upon reception of a positive CS time event confirmation together with the send delay time the DLS-user passes a CS clock value request to the local DLE as second part of the service sequence to distribute the DLSDU to all remote time receivers. If the respective DLPDU is transferred error-free the remote time receivers stop the receive delay measurement and deliver the DLSDU together with the receive delay time to their local DLS-user.

14.3 Sequence of primitives

14.3.1 Constraints on services and primitives

These fieldbus services are realised through a number of DLS primitives. A request primitive is used by a DLS-user to request a service. A confirm primitive is returned to the DLS-user upon completion of the service.

An indication primitive is used to report a non-requested event to an appropriate DLS-user. Non-requested events include reception of DLS-user data from a local DLS-user addressed to the remote DLS-user.

The DLS and their primitives are summarised in Table 56.

Table 56 – Summary of DL services and primitives

Service	Primitive	Possible for the following stations
Acknowledged connectionless data transfer: Send Data with Acknowledge (SDA)	DL-DATA-ACK request DL-DATA-ACK confirm	master
	DL-DATA-ACK indication	master and slave
Unacknowledged connectionless data transfer: Send Data with No Acknowledge (SDN)	DL-DATA request DL-DATA confirm	master
	DL-DATA indication	master and slave
Two-way connectionless data exchange: Send and Request Data with Reply (SRD)	DL-DATA-REPLY request DL-DATA-REPLY confirm	master
	DL-DATA-REPLY indication	master and slave
	DL-REPLY-UPDATE request DL-REPLY-UPDATE confirm	master and slave
M-way connectionless data exchange: Send and Request Data with Multicast Reply (MSRD)	DL-MCT-DATA-REPLY request DL-MCT-DATA-REPLY confirm	master
	DL-MCT-DATA-REPLY indication	master and slave
	DL-DXM-DATA-REPLY indication	master and slave
	DL-REPLY-UPDATE request DL-REPLY-UPDATE confirm	master and slave
Acknowledged connectionless time event and clock transfer: Clock Synchronization (CS)	DL-CS-TIME-EVENT request DL-CS-TIME-EVENT confirm	master
	DL-CS-CLOCK-VALUE request DL-CS-CLOCK-VALUE confirm	master
	DL-CS-CLOCK-VALUE indication	master and slave

14.3.2 Relation of primitives at the end-points of connectionless services

The major temporal relationships of service primitives are shown in Figure 100 to Figure 104.

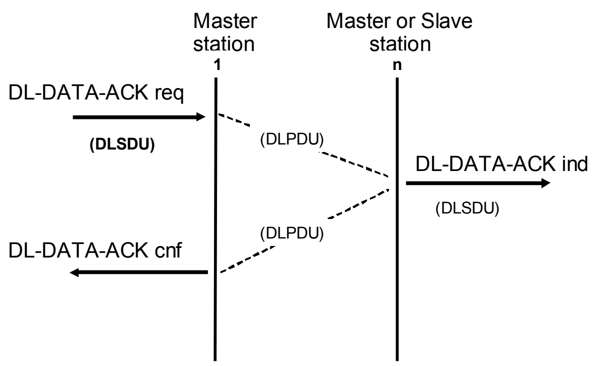


Figure 100 – SDA service

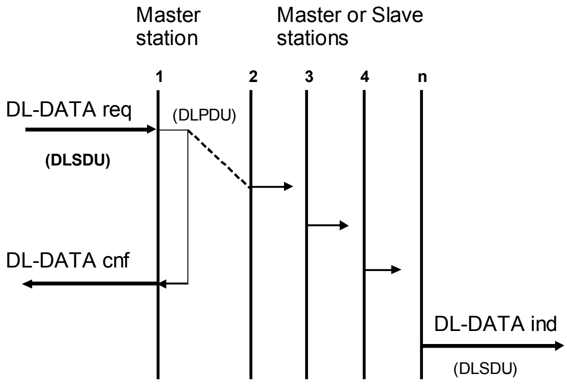


Figure 101 – SDN service

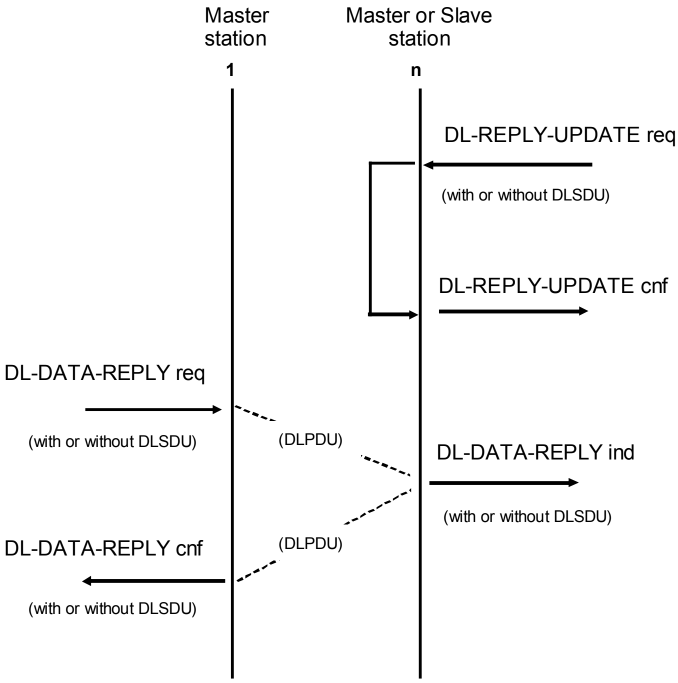


Figure 102 – SRD service

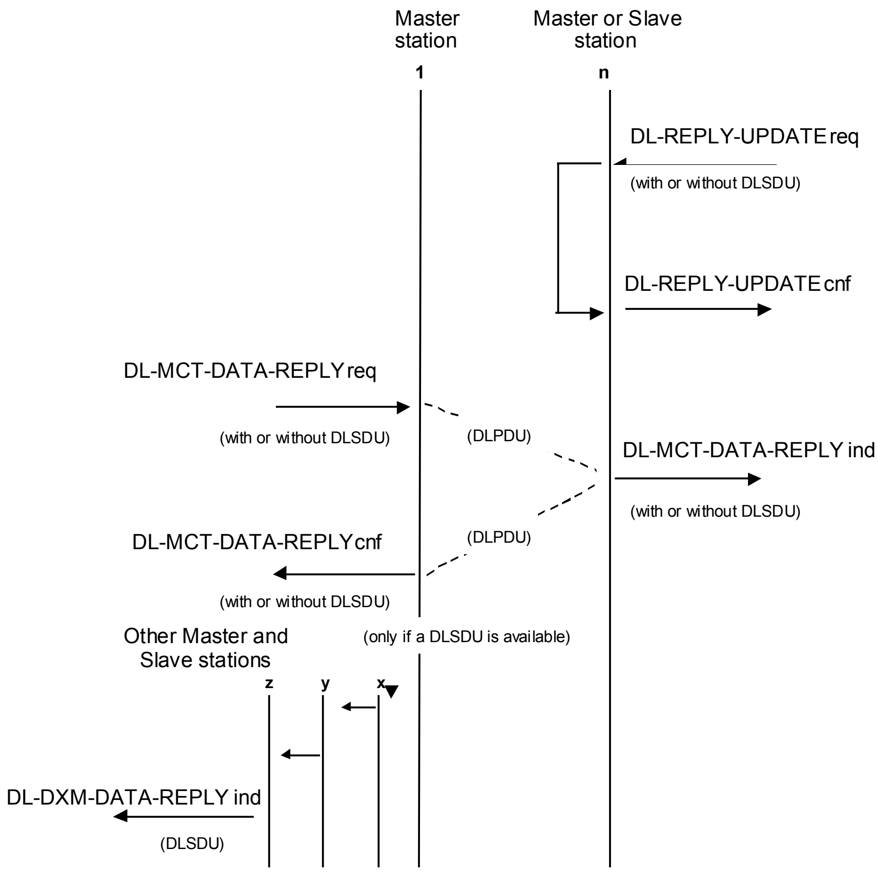


Figure 103 – MSRD service

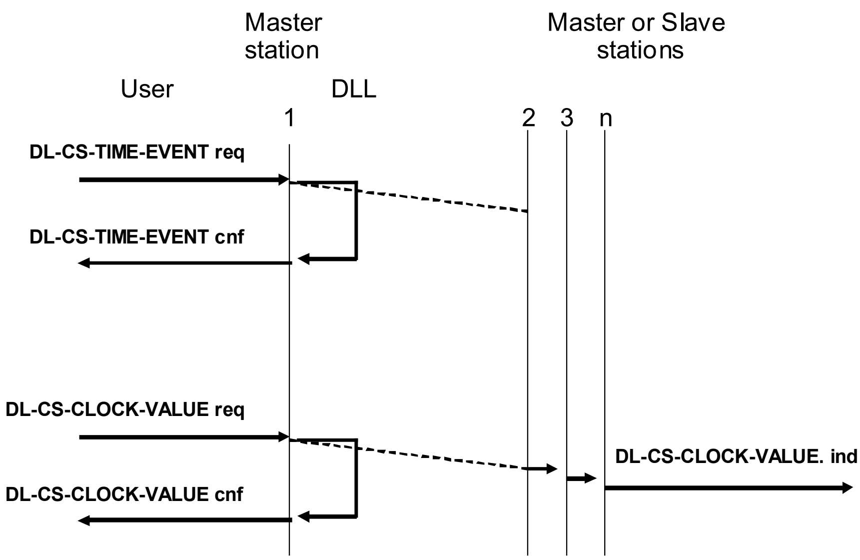


Figure 104 – CS service

14.3.3 Addressing

14.3.3.1 Address (individual)

Each DL-entity on the link is designated by a DL-address. The range of individual DL-addresses is limited, from 0 to a maximum of 126. An extended link is designated by an address extension (a region/segment address). The DL-address 127 is used for broadcast and multicast messages.

14.3.3.2 DLSAP-index

The DLSAP-index designates the DLSAP, the point of communication with the DLS-user. The range of usable DLSAP-indexes is limited, from 0 to 63, CS and NIL. The DLSAP-index 63 is used for broadcast messages. The DLSAP-index NIL means that the default DLSAP is addressed. The DLSAP-index CS is reserved for clock synchronization only. If the DLSAP-indexes CS or NIL are used in a DL service request, then the corresponding DLPDU contains no DLSAP-index (DAE or SAE) for efficiency reasons.

The DLSAP-index serves both as

a) address of a DLSAP within the DL-entity, and

b) the DLSAP-identifier for the DLS-user.

14.3.3.3 Global address

The global address is used to designate more than one DLS-user. A group of DLS users is addressed by the global address (127) in conjunction with a DLSAP-index with a value different from 63 whose interpretation throughout the link is that of a multicast address. All DLS users are addressed by the global address (127) in conjunction with the DLSAP-index 63 (see 14.4.2.3.2.3).

14.4 Detailed description of DL services

14.4.1 Send Data with Acknowledge (SDA)

14.4.1.1 Function

The local DLS-user prepares a DLSDU for the remote DLS-user and passes it to the local DLE (DL-entity) as the DLSDU parameter of a DL-DATA-ACK request primitive. The local DLE accepts the service request, forms an appropriate DLPDU containing the DLSDU, and tries to send the DLPDU to the remote DLE.

Upon receiving the data DLPDU error-free, the remote DLE immediately starts transmitting the requested acknowledgement DLPDU to the initiating DLE.

The local DLE waits for an acknowledgement DLPDU from the remote DLE. If this acknowledgement DLPDU is not received within the slot time T_{SL} or a erroneous DLPDU is received, the local DLE again transmits the data DLPDU to the remote DLE. If no error free acknowledgement DLPDU is received after a number of retransmissions equal to max_retry_limit, the local DLE reports the negative status in a confirm primitive which it issues to the local DLS-user.

When an error free acknowledgement DLPDU is received, the local DLE passes a completion status to the local DLS-user by means of a DL-DATA-ACK confirm primitive, conveying either successful completion of the requested service or the type of error detected.

During the transfer of the data and the receipt of the associated acknowledgement, no other traffic takes place on the fieldbus. If the data DLPDU was received error-free, the remote DLE passes the DLSDU and address information conveyed by the DLPDU to the remote DLS-user by means of a DL-DATA-ACK indication primitive. Retransmission does not result in duplicate DL-DATA-ACK indication primitives.

14.4.1.2 Types of primitives and parameters

Table 57 indicates the primitives and parameters of the SDA service.

Table 57 – SDA data ack primitives and parameters

DL-DATA-ACK Parameter name	Request input	Indication output	Confirm output
Service_class	M	M (=)	(Note)
D_addr	M	M (=)	(Note)
D_SAP_index	M	M (=)	(Note)
S_addr		M	
S_SAP_index	M	M (=)	(Note)
DLSDU	M	M (=)	
DL_status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.			

14.4.1.3 SDA request primitive

14.4.1.3.1 Use of the primitive

This primitive is passed from the local DLS-user to the local DLE to send DLS-user data to a remote DLS-user using the SDA service. Receipt of the primitive results in the transmittal of the DLSDU by the local DLE employing the procedure appropriate for the SDA service. While processing a SDA request (that is, while waiting for the acknowledgement) the DLE does not attempt to transmit any unrelated DLPDUs.

14.4.1.3.2 Parameters of the primitive

14.4.1.3.2.1 Service_class

This parameter specifies the priority for the data transfer. There are two priorities:

High Priority (**high**): Time-critical messages, such as alarms, synchronization and coordination data.

Low Priority (**low**): Less urgent messages, such as process, diagnostic and program data.

14.4.1.3.2.2 D_addr

The D_addr (destination-address) parameter specifies the DL-address of the remote DLE. The value 127, designating the global address used for broadcast or multicast messages, is not permitted.

NOTE The Type 3 protocol defined in IEC 61158-4 further describes and restricts DL-addresses.

14.4.1.3.2.3 D_SAP_index

The D_SAP_index (destination-service-access-point index) parameter specifies the destination service-access-point of the remote DLS-user within the remote DLE designated by the D_addr parameter. The D_SAP_index values 63, which specifies BROADCAST, and CS are not permitted.

NOTE It is possible for efficiency reasons to omit DLSAP indexes from DLPDUs. In that case the D_SAP_index parameter is set to NIL, which means that the default DLSAP in the receiving DLE is addressed.

14.4.1.3.2.4 S_SAP_index

The S_SAP_index (source-service-access-point index) parameter specifies a destination service-access-point of the local DLS-user. The S_SAP_index values 63, which specifies BROADCAST, and CS are not permitted.

NOTE It is possible for efficiency reasons to omit DLSAP indexes from DLPDUs. In that case the S_SAP_index parameter is set to NIL, which means that on reception the DLSDU is inferred to have been sent from the default DLSAP of the sending DLE.

14.4.1.3.2.5 DLSDU

This parameter specifies the DLS-user data that is to be transferred by the DLE. The minimum size of the DLSDU is one octet. The maximum size is between 242 and 246 octets, depending on whether region/segment addresses and an explicit D_SAP_index and S_SAP_index are also provided.

14.4.1.4 SDA indication primitive

14.4.1.4.1 Use of the primitive

This primitive is passed from the addressed remote DLE to the addressed remote DLS-user upon successful receipt of a SDA data DLPDU and transmission of an acknowledgement DLPDU. Receipt of a duplicate SDA data DLPDU (with no other intervening DLPDUs) does not cause the indication primitive to be repeated.

14.4.1.4.2 Parameters of the primitive

14.4.1.4.2.1 Service_class

This parameter specifies the priority of the received SDA request DLPDU.

14.4.1.4.2.2 D_addr

This parameter specifies the destination DL-address of the received SDA data DLPDU. The global address (127) for broadcast or multicast messages is not permitted.

NOTE The Type 3 protocol defined in IEC 61158-4 further describes destination DL-addresses.

14.4.1.4.2.3 S_addr

This parameter specifies the DL-address of the initiating DLE. S_addr specifies the source DL-address of the received SDA request DLPDU. S_addr shall be an individual address; the global address (127) for broadcast or multicast messages is not permitted.

NOTE The Type 3 protocol defined in IEC 61158-4 further describes source DL-addresses.

14.4.1.4.2.4 D_SAP_index, S_SAP_index,

These parameters specify the source and destination service-access-points of the received SDA data DLPDU within their respective DLEs.

14.4.1.4.2.5 DLSDU

This parameter specifies the DLS-user data sent by the remote DLS-user which initiated the service.

14.4.1.5 SDA confirm primitive

14.4.1.5.1 Use of the primitive

This primitive is passed from the local DLE to the local DLS-user upon completion of the corresponding service request. When DL_status indicates a temporary error, the local DLS-user may assume that a subsequent repetition may be successful. When DL_status indicates a permanent error, the local DLS-user may assume that a subsequent repetition may not be successful. Other method should be used to deal this type of error.

NOTE For the local errors LS, LR, DS and IV no attempt has been made to transmit the DLS-user data.

14.4.1.5.2 Parameters of the primitive

14.4.1.5.2.1 DL_status

This parameter indicates the success or failure of the corresponding SDA request and whether a temporary or a permanent error exists. Permitted values for this parameter are specified in Table 58.

Table 58 – Values of DL_status for the SDA data ack service

Short name	Status	Definition	temporary or permanent
OK	success	Service completed without error	–
RR	failure	Resources of the remote DLE not available or not sufficient	t
UE	failure	Remote DLS interface error	p
RS	failure	Service at remote DLSAP not activated, or D_addr not contained in the access parameter at the remote DLSAP	p
LS	failure	Service at local DLSAP not activated	p
LR	failure	Resources of the local DLE not available or not sufficient	t
NA	failure	No reaction, or no plausible reaction (ACK or RES), from remote DLE	t
DS	failure	Local DL-entity not in logical token ring or disconnected from line	p
IV	failure	Invalid parameters in request	–

14.4.2 Send Data with No Acknowledge (SDN)

14.4.2.1 Function

The local DLS-user prepares a DLSDU for a single remote DLS-user, for a group of remote DLS-users, or for all remote DLS-users. The DLSDU is passed to the local DLE via the DLS interface by means of a DL-DATA request primitive. The DLE accepts the service request and tries to send the data to the remote DLE or to all remote DLEs.

The sending DLE returns a local confirmation of transmittal to the local DLS-user by means of a DL-DATA confirm primitive. The receiving DLE(s) attempt to deliver the received DLSDU to the specified DLS-user(s).

There is no confirmation of correct receipt at the remote DLEs or of delivery to the intended DLS-user(s); acknowledgements do not occur. When the DLSDU is transmitted, it reaches all remote DLEs approximately concurrently (ignoring signal propagation delays). Each addressed remote DLE that has received the data DLPDU error-free passes the DLSDU and associated addressing information to the local DLS-user by means of a DL-DATA indication primitive.

14.4.2.2 Types of primitives and parameters

Table 59 indicates the primitives and parameters of the SDN service.

Table 59 – SDN data primitives and parameters

DL-DATA Parameter name	Request input	Indication output	Confirm output
Service_class	M	M (=)	(Note)
D_addr	M	M (=)	(Note)
D_SAP_index	M	M (=)	(Note)
S_addr		M	
S_SAP_index	M	M (=)	(Note)
DLSDU	M	M (=)	
DL_status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.			

14.4.2.3 SDN request primitive

14.4.2.3.1 Use of the primitive

This primitive is passed from the local DLS-user to the local DLE to send DLS-user data to a single, a group of, or all remote DLS-users using the SDN service. Receipt of the primitive results in the transmittal of the DLSDU by the local DLE employing the procedure appropriate for the SDN service.

14.4.2.3.2 Parameters of the primitive

14.4.2.3.2.1 Service_class, S_SAP_index, DLSDU

These parameters have the same meaning as described in 14.4.1.3.2.

14.4.2.3.2.2 D_addr

This parameter specifies the destination DL-address of the SDN data DLPDU. The global address (127) for broadcast or multicast messages is permitted; it designates the set of all receiving DLEs.

NOTE See Note of 14.4.1.4.2.2.

14.4.2.3.2.3 D_SAP_index

The parameter has a meaning similar to that described in 14.4.1.3.2.3. A value of 63 specifies BROADCAST; each receiving DLE delivers the received DLSDU to all local DLS-users if the BROADCAST DLSAP has been activated. The D_SAP_index value CS is not permitted.

A distinct dedicated D_SAP_index is required for each multicast group; each receiving DLE delivers the received DLSDU to the appropriate local DLS-user if that dedicated DLSAP has been activated.

14.4.2.4 SDN indication primitive

14.4.2.4.1 Use of the primitive

This primitive is passed from the remote DLE to the remote DLS-user upon receipt of a SDN data DLPDU.

14.4.2.4.2 Parameters of the primitive

14.4.2.4.2.1 Service_class, S_addr, S_SAP_index, DLSDU

These parameters have the same meaning as described in 14.4.1.4.2.

14.4.2.4.2.2 D_addr

This parameter specifies the destination DL-address of the received SDN data DLPDU. The global address (127) for broadcast or multicast messages is permitted.

NOTE See Note of 14.4.1.4.2.2.

14.4.2.4.2.3 D_SAP_index

This parameter specifies the destination service-access-point of the received SDN data DLPDU. A value of 63 specifies BROADCAST; each receiving DLE delivers the received DLSDU to all local DLS-users if the BROADCAST DLSAP has been activated. The D_SAP_index value CS is not permitted.

A distinct dedicated D_SAP_index is required for each multicast group; each receiving DLE delivers the received DLSDU to the appropriate local DLS-user if that dedicated DLSAP has been activated.

14.4.2.5 SDN confirm primitive

14.4.2.5.1 Use of the primitive

This primitive is passed from the local DLE to the local DLS-user upon completion of the corresponding service request. When DL_status indicates a temporary error, the local DLS-user may assume that a subsequent repetition may be successful. When DL_status indicates a permanent error, the local DLS-user may assume that a subsequent repetition may not be successful. Other method should be used to deal this type of error.

NOTE For the local errors LS, LR, DS and IV no attempt has been made to transmit the DLS-user data.

14.4.2.5.2 Parameters of the primitive

14.4.2.5.2.1 DL_status

This parameter indicates the local success or failure of the associated SDN request. The possible values of this parameter are specified in Table 60.

Table 60 – Values of DL_status for the SDN data service

Short name	Status	Definition	**temporary** or **permanent**
OK	success	Transmission of data completed by local DL-entity	–
LS	failure	Service at local DLSAP or local DLSAP not activated	p
LR	failure	Resources of the local DLE not available or not sufficient	t
DS	failure	Local DL-entity not in logical token ring or disconnected from line	p
IV	failure	Invalid parameters in request	–

14.4.3 Send and Request Data with Reply (SRD)

14.4.3.1 Function

The local DLS-user prepares a DLSDU for the remote DLS-user and passes it to the local DLE as the DLSDU parameter of a DL-DATA-REPLY request primitive, requesting data from the remote DLS-user at the same time. The local DLE accepts the service request, forms an appropriate DLPDU containing the DLSDU, and tries to send the DLPDU to the remote DLE, requesting that a DLSDU previously prepared by the remote DLS-user be sent in reply.

Alternatively, if the local DLS-user has no DLSDU to send, it passes the DL-DATA-REPLY request primitive to the DLE without a DLSDU parameter. In this case the local DLE accepts the service request, forms an appropriate DLPDU not containing a DLSDU, and tries to send the DLPDU to the remote DLE, requesting that a DLSDU previously prepared by the remote DLS-user be sent in reply.

Upon receiving the request DLPDU, the remote DLE immediately starts transmitting a reply DLPDU to the initiating DLE, if the remote DLS-user had previously prepared a DLSDU for this reply (by means of a DL-REPLY-UPDATE request primitive). If no DLSDU is available for transmission, or if an error occurred, then an acknowledgement DLPDU with appropriate status information is returned instead addressed to the initiating DLE only.

The receiving DLE passes the DLSDU, if any, received from the initiating DLE, together with status concerning the transmitted reply, to its local DLS-user with a DL-DATA-REPLY indication primitive.

The local DLE waits for a reply DLPDU from the remote DLE. If this reply DLPDU is not received error-free within the slot time T_{SL}, the local DLE again transmits the request DLPDU to the remote DLE. If no reply DLPDU is received error-free after a number of retransmissions equal to max_retry_limit, the local DLE reports the negative status in a confirm primitive which it issues to the local DLS-user.

When a reply DLPDU is received, the local DLE passes the conveyed DLSDU, if any, together with a completion status to the local DLS-user by means of a DL-DATA-REPLY confirm primitive; this status conveys either successful completion of the requested service or the type of error detected.

The remote DLS-user is responsible for having prepared a valid DLSDU, ready for transmission by the remote DLE. The remote DLS-user passes a DL-REPLY-UPDATE request primitive to the remote DLE to convey the DLSDU to that DLE, where it awaits a remotely-initiated SRD transfer request. The DLE informs the DLS-user of the completion of this request by means of a DL-REPLY-UPDATE confirm primitive.

14.4.3.2 Types of primitives and parameters of SRD data-reply

Table 61 indicates the primitives and parameters of the SRD data reply service.

Table 61 – SRD data reply primitives and parameters

DL-DATA-REPLY Parameter name	Request input	Indication output	Confirm output
Service_class	M	M	(Note)
D_addr	M	M (=)	(Note)
D_SAP_index	M	M (=)	(Note)
S_addr		M	
S_SAP_index	M	M (=)	(Note)
DLSDU	U	U (=)	U
Reference		U	
Update_status		M	
DL_status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.			

14.4.3.3 SRD data-reply request primitive

14.4.3.3.1 Use of the primitive

This primitive is passed from the local DLS-user to the local DLE

a) optionally, to send DLS-user data to a remote DLS-user, and
b) simultaneously to request previously-prepared DLS-user data from that DLS-user

both through use of the SRD service.

Receipt of this primitive results in the transmittal of the DLSDU by the local DLE employing the procedure appropriate for the SRD service. While processing a SRD request (that is, while waiting for the reply and during any retry attempts) the DLE does not attempt to transmit any unrelated DLPDUs.

14.4.3.3.2 Parameters of the primitive

14.4.3.3.2.1 Service_class

This parameter has the same meaning as described in 14.4.1.3.2.1.

14.4.3.3.2.2 D_addr, S_SAP_index, DLSDU

The D_addr, S_SAP_index and DLSDU parameters have the same meaning as described in 14.4.1.3.2.

14.4.3.3.2.3 D_SAP_index

This parameter specifies the destination service-access-point of the remote DLS-user within the remote DLE designated by the D_addr parameter. The specified remote DLSAP may also have an associated DLSDU which has been prepared by that DLSAPs DLS-user. The D_SAP_index values 63, which specifies BROADCAST, and CS are not permitted.

NOTE It is possible for efficiency reasons to omit DLSAP indexes from DLPDUs. In that case the D_SAP_index parameter is set to NIL, which means that the default DLSAP in the receiving DLE is addressed.

14.4.3.4 SRD data-reply indication primitive

14.4.3.4.1 Use of the primitive

This primitive is passed from the addressed remote DLE to the remote DLS-user upon receipt of a SRD request DLPDU and transmission of a reply DLPDU. Receipt of a duplicate SRD request DLPDU (with no other intervening DLPDUs) does not cause the indication primitive to be repeated.

However, no indication primitive occurs when

a) both the received SRD request DLPDU and the reply DLPDU contain null (zero length) DLSDUs, and

b) the addressed remote DLS-user has configured the D_SAP to not signal such events.

NOTE 1 This behaviour is configured through the Indication_mode parameter of the DL-management DLSAP Activate Responder primitive (see 15.5.8).

NOTE 2 This non-reporting does not affect the storage resources of the responding DLE.

14.4.3.4.2 Parameters of the primitive

14.4.3.4.2.1 Service_class, D_addr, D_SAP_index, S_addr, S_SAP_index, DLSDU

These parameters have the same meaning as described in 14.4.1.4.2.

14.4.3.4.2.2 Reference

This optional parameter is used to identify the DLSDU that was passed upon receipt of a SRD request DLPDU.

14.4.3.4.2.3 Update_status

This parameter indicates whether or not the response data (DLSDU) has been passed to the initiating local DLE. Permitted values for this parameter are specified in Table 62.

Table 62 – Values of Update_status for the SRD data reply service

Short name	Status	Definition	temporary or permanent
NO	failure	No reply data (DLSDU) transmitted	t
LO	success	Low priority reply data transmitted	–
HI	success	High priority reply data transmitted	–

14.4.3.5 SRD data-reply confirm primitive

14.4.3.5.1 Use of the primitive

This primitive is passed from the local DLE to the local DLS-user upon completion of the corresponding service request. DL_status indicates the completion status of the request and, when successful, the presence or absence of a returned DLSDU. When DL_status indicates a temporary error, the local DLS-user may assume that a subsequent repetition may be successful. When DL_status indicates a permanent error, the local DLS-user may assume that a subsequent repetition may not be successful. Other method should be used to deal this type of error.

14.4.3.5.2 Parameters of the primitive

14.4.3.5.2.1 DLSDU

This optional parameter returns the DLS-user data sent by the remote DLE, if any. This parameter will not appear, if the DL_status is different from DL, DH, RDL and RDH.

14.4.3.5.2.2 DL_status

This parameter indicates the success or failure of the corresponding SRD request. The values UE, RS, LS, LR, NA, DS and IV as specified for SDA (see Table 58) are possible. Additional possible values are specified in Table 63.

Table 63 – Additional values of DL_status for the SRD data reply service

Short name	Status	Definition	temporary or permanent
DL	success	Positive acknowledgement for sent data, reply data (DLSDU) with low priority available	–
DH	success	Positive acknowledgement for sent data, reply data with high priority available	–
NR	success	Positive acknowledgement for sent data, negative acknowledgement for reply data, as not available in the remote DLE	t
RDL	failure	Negative acknowledgement for sent data, resources of the remote DLE not available or not sufficient, reply data with low priority available	t
RDH	failure	Negative acknowledgement for sent data, resources of the remote DLE not available or not sufficient, reply data with high priority available	t
RR	failure	Negative acknowledgement for sent data, resources of the remote DLE not available or not sufficient, reply data not available	t

14.4.3.6 Types of primitives and parameters of SRD reply-update

Table 64 indicates the primitives and parameters of the SRD reply-update service.

Table 64 – SRD reply-update primitives and parameters

DL-REPLY-UPDATE	Request	Confirm
Parameter name	input	output
Service_class	M	(Note)
S_SAP_index	M	(Note)
DLSDU	U	
Transmit_strategy	M	
Reference	U	
DL_status		M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.

14.4.3.7 SRD reply-update request primitive

14.4.3.7.1 Use of the primitive

This primitive is passed from the local DLS-user to the local DLE to convey a DLSDU which can be retrieved by a remotely-initiated invocation of the SRD service. The local DLE associates the DLSDU with the specified S_SAP_index in a way which avoids update concurrent with any ongoing SRD transaction at that S_SAP_index. This primitive is only useful in conjunction with remote invocation of the SRD service; of itself it does not cause any transmission of the conveyed DLSDU.

14.4.3.7.2 Parameters of the primitive

14.4.3.7.2.1 Service_class

This parameter has the same meaning as described in 14.4.3.3.2.1.

14.4.3.7.2.2 S_SAP_index

This parameter specifies the service access point of the local DLS-user which makes the request. The S_SAP_index values 63, which specifies BROADCAST, and CS are not permitted.

14.4.3.7.2.3 DLSDU

This optional parameter specifies the DLS-user data which is to be used to update the data associated with the specified S_SAP_index.

14.4.3.7.2.4 Transmit_strategy

This parameter specifies whether the update is transmitted only once (SINGLE) or many times (MULTIPLE). In the case of "MULTIPLE" any DLSDU associated with the S_SAP_index is transferred with each subsequent SRD.

In the case of "SINGLE" the association of the DLSDU with the S_SAP_index is terminated after the first apparently-successful SRD exchange (and any immediately-following retries). This causes subsequent SRD exchanges do not return a DLSDU until the DLS-user associates a new DLSDU with the S_SAP_index.

14.4.3.7.2.5 Reference

This optional parameter is used to identify the DLSDU that was passed by a DL-REPLY-UPDATE request primitive.

14.4.3.8 SRD reply-update confirm primitive

14.4.3.8.1 Use of the primitive

This primitive is passed from the local DLE to the local DLS-user upon completion of the corresponding service request. When DL_status indicates a temporary error, the local DLS-user may assume that a subsequent repetition may be successful. When DL_status indicates a permanent error, the local DLS-user may assume that a subsequent repetition may not be successful. Other method should be used to deal this type of error.

14.4.3.8.2 Parameters of the primitive

14.4.3.8.2.1 DL_status

This parameter indicates the result of the corresponding DL-REPLY-UPDATE REQuest primitive. The possible values are specified in Table 65.

Table 65 – Values of DL_status for the SRD reply-update service

Short name	Status	Definition	temporary or permanent
OK	success	Update data (DLSDU) loaded	–
LS	failure	Service at local DLSAP or local DLSAP not activated	p
LR	failure	Resources of the local DLE not available or not sufficient	t
IV	failure	Invalid parameters in request	–

14.4.4 Send and Request Data with Multicast reply (MSRD)

14.4.4.1 Function

The local DLS-user prepares a DLSDU for the remote DLS-user and passes it to the local DLE as the DLSDU parameter of a DL-MCT-DATA-REPLY request primitive, requesting data from the remote DLS-user (Publisher) at the same time. The local DLE accepts the service request, forms an appropriate DLPDU containing the DLSDU, and tries to send the DLPDU to the remote DLE (Publisher), requesting that a DLSDU previously prepared by the remote DLS-user be multicast in reply to the appropriate set of DLEs (Subscribers) which have configured their corresponding DLSAP in the intention to subscribe to this particular publisher (DLSAP activate subscriber service).

Alternatively, if the local DLS-user has no DLSDU to send, it passes the DL-MCT-DATA-REPLY request primitive to the DLE without a DLSDU parameter. In this case the local DLE accepts the service request, forms an appropriate DLPDU not containing a DLSDU, and tries to send the DLPDU to the remote DLE (Publisher), requesting that a DLSDU previously prepared by the remote DLS-user be multicast in reply.

Upon receiving the request DLPDU error-free, the remote DLE (Publisher) immediately starts transmitting a reply DLPDU to the initiating DLE and the appropriate set of remote DLEs (Subscribers) by sending the response using the destination address DA = 127 (Broadcast) and a specified D_SAP_index if the remote DLS-user had previously prepared a DLSDU for this reply (by means of a DL-REPLY-UPDATE request primitive). If no DLSDU is available for transmission, or if an error occurred, then an acknowledgement DLPDU with appropriate status information is returned instead addressed to the initiating DLE only.

The receiving DLE (Publisher) passes the DLSDU, if any, received from the initiating DLE, together with status concerning the transmitted reply, to its local DLS-user with a DL-DATA-REPLY indication primitive.

The local DLE waits for a reply DLPDU from the remote DLE (Publisher). If this reply DLPDU is not received error-free within the slot time T_{SL}, the local DLE again transmits the request DLPDU to the remote DLE (Publisher). If no reply DLPDU is received error-free after a number of retransmissions equal to max_retry_limit, the local DLE reports the negative status in a confirm primitive which it issues to the local DLS-user.

When a reply DLPDU is received, the local DLE passes the conveyed DLSDU, if any, together with a completion status to the local DLS-user by means of a DL-MCT-DATA-REPLY confirm primitive; this status conveys either successful completion of the requested service or the type of error detected.

The DLEs (Subscribers) which receive the reply DLPDU addressed to the destination address DA = 127 (Broadcast) and the specified D_SAP_index indicate this to their DLS-user with a DL-DXM-REPLY indication primitive. The completion status of a DL-DXM-REPLY indication is always set to successful completion.

The remote DLS-user is responsible for having prepared a valid DLSDU, ready for transmission by the remote DLE (Publisher). The remote DLS-user passes a DL-REPLY-UPDATE request primitive to its local DLE to convey the DLSDU to that DLE, where it awaits a remotely-initiated MSRD transfer request. The DLE informs the DLS-user of the completion of this request by means of a DL-REPLY-UPDATE confirm primitive.

14.4.4.2 Types of primitives and parameters of MSRD MCT-data-reply

Table 66 indicates the primitives and parameters of the MSRD MCT data reply service.

Table 66 – MSRD MCT data reply primitives and parameters

DL-MCT-DATA-REPLY Parameter name	Request input	Indication output	Confirm output
Service_class	M	M	(Note)
D_addr	M	M (=)	(Note)
D_SAP_index	M	M (=)	(Note)
S_addr		M	
S_SAP_index	M	M (=)	(Note)
DLSDU	U	U (=)	U
Update_status		M	
Reference		U	
DL_status			M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.

14.4.4.3 MSRD MCT-DATA-REPLY request primitive

14.4.4.3.1 Use of the primitive

This primitive is passed from the local DLS-user to the local DLE

a) optionally, to send DLS-user data to a remote DLS-user, and
b) simultaneously to request previously-prepared DLS-user data to be published from that DLS-user

both through use of the MSRD service.

Receipt of this primitive results in the transmittal of the DLSDU by the local DLE employing the procedure appropriate for the MSRD service. While processing a MSRD request (that is, while waiting for the reply and during any retry attempts) the DLE does not attempt to transmit any unrelated DLPDUs.

14.4.4.3.2 Parameters of the primitive

14.4.4.3.2.1 Service_class

This parameter has the same meaning as described in 14.4.1.3.2.1.

14.4.4.3.2.2 D_addr, S_SAP_index, DLSDU

The D_addr, S_SAP_index and DLSDU parameters have the same meaning as described in 14.4.1.3.2.

14.4.4.3.2.3 D_SAP_index

This parameter specifies the DLSAP of the remote DLS-user within the remote DLE (Publisher) designated by the D_addr parameter. The specified remote DLSAP may also have an associated DLSDU which has been prepared by that DLSAPs DLS-user. The D_SAP_index values 63, which specifies Broadcast, and CS are not permitted.

14.4.4.4 MSRD MCT-data-reply indication primitive

14.4.4.4.1 Use of the primitive

This primitive is passed from the addressed remote DLE (Publisher) to the remote DLS-user upon receipt of a MSRD request DLPDU and transmission of a reply DLPDU. Receipt of a duplicate MSRD request DLPDU (with no other intervening DLPDUs) does not cause the indication primitive to be repeated.

However, no indication primitive occurs when

a) both the received MSRD request DLPDU and the reply DLPDU contain null (zero length) DLSDUs, and

b) the addressed remote DLS-user has configured the D_SAP to not signal such events.

NOTE 1 This behaviour is configured through the Indication_mode parameter of the DL-management DLSAP Activate Responder primitive (see 15.5.8).

NOTE 2 This non-reporting does not affect the storage resources of the responding DLE.

14.4.4.4.2 Parameters of the primitive

14.4.4.4.2.1 Service_class, D_addr, D_SAP_index, S_addr, S_SAP_index, DLSDU

These parameters have the same meaning as described in 14.4.3.4.2.

14.4.4.4.2.2 Reference

This parameter has the same meaning as described in 14.4.3.4.2.2.

14.4.4.4.2.3 Update_status

This parameter indicates whether or not the response data (DLSDU) has been passed to the initiating local DLE and to all other remote DLEs (Subscribers). Permitted values for this parameter are specified in Table 62 (see 14.4.3.4.2.3).

14.4.4.5 MSRD MCT-data-reply confirm primitive

14.4.4.5.1 Use of the primitive

This primitive is passed from the local DLE to the local DLS-user upon completion of the corresponding service request. DL_status indicates the completion status of the request and, when successful, the presence or absence of a returned DLSDU. When DL_status indicates a temporary error, the local DLS-user may assume that a subsequent repetition may be successful. When DL_status indicates a permanent error, the local DLS-user may assume that a subsequent repetition may not be successful. Other method should be used to deal this type of error.

14.4.4.5.2 Parameters of the primitive

14.4.4.5.2.1 DLSDU

This optional parameter returns the DLS-user data sent by the remote DLE, if any. This parameter will not appear, if the DL_status is different from DL, DH, RDL and RDH.

14.4.4.5.2.2 DL_status

This parameter has the same meaning as described in 14.4.3.5.2.2.

14.4.4.6 Type of primitive and parameters of MSRD DXM data reply

Table 67 indicates the primitives and parameters of the MSRD DXM data reply service.

Table 67 – MSRD DXM data reply primitive and parameters

DL-DXM-DATA-REPLY Parameter name	Indication output
Service_class	M
D_addr	M
D_SAP_index	M
S_addr	M
S_SAP_index	M
DLSDU	M

14.4.4.7 MSRD DXM data reply indication primitive

14.4.4.7.1 Use of the primitive

This primitive is passed from the remote DLE (Subscriber) to the remote DLS-user upon receipt of a reply DLPDU addressed to the DLE with the destination address DA = 127 (Broadcast) and the specified D_SAP_index to convey a DLSDU which has been retrieved by a remotely-initiated invocation of the MSRD service. This primitive is only possible in conjunction with remote invocation of the MSRD service.

14.4.4.7.2 Parameters of the primitive

14.4.4.7.2.1 Service_class

This parameter has the same meaning as described in 14.4.1.3.2.1.

14.4.4.7.2.2 S_addr, S_SAP_index

These parameters have the same meaning as described in 14.4.3.4.2.

14.4.4.7.2.3 DLSDU

This parameter returns the DLS-user data sent by the remote DLE (Publisher).

14.4.4.7.2.4 D_addr

This parameter specifies the destination DL-address of the received reply DLPDU. The global address (127) is the only allowed value.

NOTE See Note of 14.4.1.4.2.2.

14.4.4.7.2.5 D_SAP_index

This parameter specifies the destination service-access-point of the received reply DLPDU. The D_SAP_index value 63, which specifies Broadcast, and CS are not permitted. Each receiving remote DLE (Subscriber) delivers the received DLSDU to its DLS-user.

NOTE See Note of 14.4.1.3.2.3

14.4.4.8 SRD reply-update request primitive

For a description of this primitive and its parameters, see 14.4.3.7.

14.4.4.9 SRD reply-update confirm primitive

For a description of this primitive and its parameters, see 14.4.3.8.

14.4.5 Clock Synchronization (CS)

14.4.5.1 Function

The local DLS-user passes a DL-CS-TIME-EVENT request primitive to the local DLE to start the clock synchronization sequence. The local DLE accepts the service request, forms an appropriate DLPDU to transmit a Time Event indicating the start of clock synchronization to all remote DLEs (time receivers) which support clock synchronzation.

Upon reception of a DL-CS-TIME-EVENT request primitive the local DLE starts a send-delay-timer to measure the delay between the reception of the primitive and sending of the appropriate DLPDU. The remote DLEs (time receivers) start the receive-delay-timer after reception of such a DLPDU.

The local DLE passes a DL-CS-TIME-EVENT confirm primitive together with the send delay time and the status of transmission to the local DLS-user.

Upon a positive confirmation the local DLS-user passes a DL-CS-CLOCK-VALUE request primitive with a DLSDU which contains clock informations to the local DLE. The local DLE accepts the service request, forms an appropriate DLPDU containing the DLSDU, and tries to send the DLPDU to all remote DLEs (time receivers) which support clock synchronization. When the DLPDU is sent, the local DLE passes a DL-CS-CLOCK-VALUE confirm primitive together with a completion status to the local DLS-user. The remote DLEs (time receivers) that receive the DLPDU error-free stop their receive-delay-timer. The receiving DLEs (time receivers) pass the DLSDU received from the initiating DLE (time master) together with the receive delay time and the status concerning the transmission to its local DLS-user by means of a DL-CS-CLOCK-VALUE indication primitive.

14.4.5.2 Types of primitives and parameters of the CS time event

Table 68 indicates the primitives and parameters of the CS time event service.

Table 68 – CS time event primitives and parameters

DL-CS-TIME-EVENT Parameter name	Request input	Confirm output
D_addr	M	(Note)
D_SAP_index	M	(Note)
S_SAP_index	M	(Note)
Send_delay_time		C
DL_status		M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.

14.4.5.3 CS time event request primitive

14.4.5.3.1 Use of the primitive

This primitive is passed to the local DLE (time master) to send an appropriate DLPDU to a group of remote DLEs (time receivers) which have activated the CS DLSAP. Upon reception of this primitive the local DLE (time master) starts its send-delay-timer to measure the internal delay between receiving the primitive and transmitting the appropriate DLPDU.

14.4.5.3.2 Parameters of the primitive

14.4.5.3.2.1 D_addr

The D_addr (destination-address) parameter specifies the DL-address of the remote DLE. The global address (127) is permitted; it designates the set of all receiving DLEs.

NOTE See Note of 14.4.1.4.2.2.

14.4.5.3.2.2 D_SAP_index

The D_SAP_index parameter specifies the DLSAP of the remote DLS-user within the remote DLE designated by the D_addr parameter. The D_SAP_index value CS is the only allowed one.

14.4.5.3.2.3 S_SAP_index

The S_SAP_index parameter specifies a DLSAP of the local DLS-user. The S_SAP_index value CS is the only allowed value.

14.4.5.4 CS time event confirm primitive

14.4.5.4.1 Use of the primitive

After successful transmission of the DLPDU initiated by the appropriate request primitive the local DLE stops its send-delay-timer and returns a local confirmation of transmittal together with the send delay time to the local DL-user by means of a DL-CS-TIME-EVENT confirm primitive.

14.4.5.4.2 Parameters of the primitive

14.4.5.4.2.1 Send_delay_time

This conditional parameter contains the delay time which elaps between the CS time event request and the transmitting of the appropriate DLPDU. This parameter is not present when the resulting DL_status is different from OK and SV.

14.4.5.4.2.2 DL_status

This parameter indicates the success or failure of the associated request and whether a temporary or a permanent error exists. Table 69 specifies the permitted values.

Table 69 – Values of DL_status for the CS time event service

Short name	Status	Definition	temporary or permanent
OK	success	The parameter Send_delay_time is available.	–
LS	failure	Service at local DLSAP or local DLSAP not activated (send-delay-timer has not been started).	p
LR	failure	Resources local not available or not sufficient (send-delay-timer has not been started).	p
DS	failure	Local DL/Ph Entity not in logical token ring or disconnected from line (send-delay-timer has not been started).	t
SV	failure	Sequence violation (subsequent CS time event services without CS clock value service in between)	p
IV	failure	Invalid parameters in request	–

14.4.5.5 Types of primitives and parameters of the CS clock value

Table 70 indicates the primitives and parameters of the CS clock value service.

Table 70 – CS clock value primitives and parameters

DL-CS-CLOCK-VALUE Parameter name	Request input	Indication output	Confirm output
D_addr	M	M(=)	(Note)
D_SAP_index	M	M(=)	(Note)
S_addr		M	
S_SAP_index	M	M(=)	(Note)
DLSDU	M	M(=)	
Receive_delay_time		M	
CS_status		M	
DL_status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.			

14.4.5.6 CS clock value request primitive

14.4.5.6.1 Use of the primitive

The DL-user passes a DLSDU by meaning of a CS clock value request primitive. The local DLE prepares an appropriate DLPDU and tries to transmit this DLPDU.

14.4.5.6.2 Parameters of the primitive

14.4.5.6.2.1 D_addr, D_SAP_index, S_SAP_index

These parameters have the same meaning as described in 14.4.5.3.2.

14.4.5.6.2.2 DLSDU

This parameter specifies the DLS-user data that is to be transferred by the local DLE (time master). The size of this DLSDU is fixed to 18 octets.

14.4.5.7 CS clock value indication primitive

14.4.5.7.1 Use of the primitive

This primitive is passed from the remote DLE (time receiver) to the adressed DLS-user upon reception of a Clock Value DLPDU. If the Clock Value DLPDU was received error-free the local DLE stops their receive-delay-timer and calculate the receive delay time.

14.4.5.7.2 Parameters of the primitive

14.4.5.7.2.1 D_addr

This parameter specifies the destination DL-address of the received Clock Value DLPDU. The global address (127) is permitted; it designates the set of all receiving DLEs.

NOTE See Note of 14.4.1.4.2.2.

14.4.5.7.2.2 S_addr

This parameter specifies the DL-address of the initiating DLE. The S_addr shall be an individual address; the global address (127) for broadcast or multicast messages is not permitted.

NOTE See Note of 14.4.1.4.2.3.

14.4.5.7.2.3 D_SAP_index, S_SAP_index

These parameters specify the source and destination DLSAPs of the received Clock Value DLPDU within their respective DLEs.

14.4.5.7.2.4 DLSDU

This parameter has the same meaning as described in 14.4.5.6.2.2.

14.4.5.7.2.5 Receive_delay_time

This parameter contains the receive delay time between the reception of a Time Event DLPDU and the end of the Clock Value DLPDU reception. In case of a sequence violation the value of this parameter is zero.

14.4.5.7.2.6 CS_status

This parameter indicates the success or failure of the Clock Synchronization sequence. Table 71 specifies the parameter values.

Table 71 – Values of CS_status for the CS clock value service

Short name	Status	Definition	**temporary** or **permanent**
OK	success	clock synchronization sequence orderly executed	–
SV	failure	clock synchronization sequence not orderly executed	t/p

14.4.5.8 CS clock value confirm primitive

14.4.5.8.1 Use of the primitive

The local DLE (time master) returns a local confirmation of transmittal to the local DL-user by means of this primitive including a status information about the success or failure of the transmittal and the validity of the clock synchronization sequence.

14.4.5.8.2 Parameters of the primitive

14.4.5.8.2.1 DL_status

This parameter indicates success or failure of the associated request and whether a temporary or a permanent error exists. Table 72 specifies the parameter values.

Table 72 – Values of DL_status for the CS clock value service

Short name	Status	Definition	**temporary** or **permanent**
OK	success	Transmission of CS clock value completed by local DL-entity	–
LS	failure	Service at local DLSAP or local DLSAP not activated	p
LR	failure	Resources local not available or not sufficient	p
DS	failure	Local FDL/Ph Entity not in logical token ring or disconnected from line	t
SV	failure	Sequence violation	p
IV	failure	Invalid parameters in request	–

15 Type 3: DL-management Service

15.1 General

This clause describes the interface between a DLE and a DL-management user (DLMS-user). The services of this interface are needed for the protocol which implements the DLS specified in Clause 14.

15.2 Facilities of the DLMS

DL-management organises the initialisation, configuration, event and error handling between the DLMS-user and the logical functions in the DLE. The following functions are provided to the DLMS-user.

a) Reset of the local DLE
b) Request for and modification of the actual operating parameters and of the counters of the local DLE
c) Notification of unexpected events, errors, and status changes, both local and remote
d) Request for identification and for the DLSAP configuration of the local DLE
e) Activation and deactivation of local DLSAPs

15.3 Services of the DL-management

15.3.1 Overview of services

DL-management provides the following services to the DLMS-user:

a) Reset
b) Set Value
c) Get Value
d) Event
e) Ident
f) DLSAP Status
g) DLSAP Activate
h) DLSAP Activate Responder
i) DLSAP Activate Subscriber
j) DLSAP Deactivate

The services Reset, Set Value, Event and DLSAP Activate are considered mandatory. The services Get Value, Ident, DLSAP Status, DLSAP Activate Subscriber and DLSAP Deactivate are considered optional. The service DLSAP Activate Responder is considered mandatory for slaves and optional for masters.

15.3.2 Reset

The DLMS-user employs this service to cause DL-management to reset the DLE. A reset is equivalent to power on. The DLMS-user receives a confirmation thereof.

15.3.3 Set value

The DLMS-user employs this service to assign new values to the variables of the DLE. The DLMS-user receives a confirmation whether the specified variables have been set to the new values.

15.3.4 Get value

This service enables DL-management to read variables of the DLE. The response of the DL-management returns the actual value of the specified variables.

15.3.5 Event

DL-management employs this service to inform the DLMS-user about certain events or errors in the DLL.

15.3.6 Ident

When requesting the identification service, a distinction is made between master and slave stations. By employing this service the DLMS-user of a slave station determines the version data of the local DLE's hardware and software. When employing this service in the case of a master station, the DLMS-user may additionally request the same type of information from a remote station.

15.3.7 DLSAP status

The DLMS-user employs this service to inform itself about the configuration of DLSAPs of the local DLE.

15.3.8 DLSAP activate

This service enables the DLMS-user to activate and to configure a local DLSAP for the reply services (SRD and MSRD). Excluded from this is the responder function for the reply services. The DLMS-user receives a confirmation on the execution of the service from DL-management.

15.3.9 DLSAP activate responder

The DLMS-user employs this service to activate a local DLSAP for the responder function for the reply services (SRD and MSRD). The DLMS-user receives a confirmation of execution of the service from DL-management.

15.3.10 DLSAP activate subscriber

The DLMS-user employs this service to activate a local DLSAP for the subscriber function of the MSRD service. The DLMS-user receives a confirmation of execution of the service from DL-management.

15.3.11 DLSAP deactivate

The DLMS-user employs this service to cause DL-management to deactivate a local DLSAP. DL-management returns a confirmation thereof.

15.4 Overview of interactions

DL-management services and their primitives are summarised in Table 73.

Table 73 – Summary of DL-management services and primitives

Service	Primitive	Possible for the following stations
Reset	DLM-RESET request DLM-RESET confirm	local: master and slave
Set Value	DLM-SET-VALUE request DLM-SET-VALUE confirm	local: master and slave
Get Value	DLM-GET-VALUE request DLM-GET-VALUE confirm	local: master and slave
Event	DLM-EVENT indication	local: master and slave
Ident	DLM-IDENT request DLM-IDENT confirm	local: master and slave remote: master
DLSAP Status	DLM-DLSAP-STATUS request DLM-DLSAP-STATUS confirm	local: master and slave
DLSAP Activate	DLM-DLSAP-ACTIVATE request DLM-DLSAP-ACTIVATE confirm	local: master and slave
DLSAP Activate Responder	DLM-DLSAP-ACTIVATE-RESPONDER request DLM-DLSAP-ACTIVATE-RESPONDER confirm	local: master and slave
DLSAP Activate Subscriber	DLM-DLSAP-ACTIVATE-SUBSCRIBER request DLM-DLSAP-ACTIVATE-SUBSCRIBER confirm	local: master and slave
DLSAP Deactivate	DLM-DLSAP-DEACTIVATE request DLM-DLSAP-DEACTIVATE confirm	local: master and slave

The temporal relationships of the DL-management primitives are shown in Figure 105 through Figure 107.

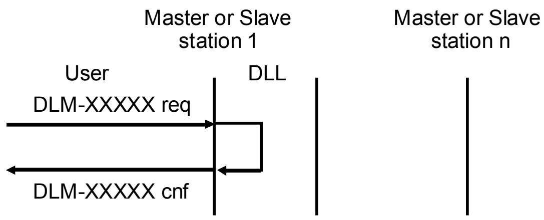


Figure 105 – Reset, Set value, Get value, Ident (local), DLSAP status, DLSAP activate, DLSAP activate responder, DLSAP activate subscriber and DLSAP deactivate services

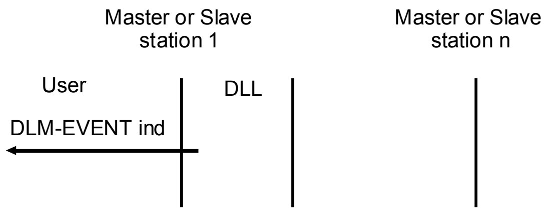


Figure 106 – Event service

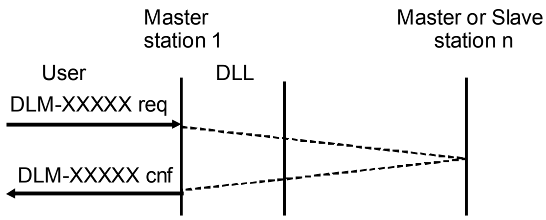


Figure 107 – Ident (remote) service

15.5 Detailed specification of services and interactions

15.5.1 Reset

15.5.1.1 Function

The DLMS-user passes a DLM-RESET request primitive to DL-management causing it to reset the DLE. This is carried out in the same manner as at a Power On. The DLE assumes the "Offline"-status and all DLE variables (operational parameters/counters) are cleared. As a result DL-management passes a DLM-RESET confirm primitive to the DLMS-user to indicate the success or failure of the corresponding service request.

15.5.1.2 Types of primitives and parameters

Table 74 indicates the primitives and parameters of the Reset service.

Table 74 – Reset primitives and parameters

DLM-RESET	Request	Confirm
Parameter name	input	output
DLM_status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

15.5.1.3 Parameters of the primitive

15.5.1.3.1 DLM_status

This parameter indicates the success or failure of the associated reset service request. Permitted values for this parameter are specified in Table 75.

Table 75 – Values of DLM_status for the reset service

Short name	**Status**	**Definition**	**temporary** or **permanent**
OK	success	The Reset function was carried out successfully	–
NO	failure	The Reset function was not carried out success-fully	t/p
IV	failure	Invalid parameters in request	–

15.5.2 Set Value

15.5.2.1 Function

The DLMS-user passes a DLM-SET-VALUE request primitive to DL-management to assign a desired value to one or more specified variables of the DLE. After receiving this primitive DL-management tries to select these variables and to set the new values. If the requested service was executed DL-management passes a DLM-SET-VALUE confirm primitive to the DLMS-user to indicate the success or failure of the corresponding service request.

15.5.2.2 Types of primitives and parameters

Table 76 indicates the primitives and parameters of the Set Value service.

Table 76 – Set value primitives and parameters

DLM-SET-VALUE	**Request**	**Confirm**
Parameter name	input	output
Variable_name (1 to n)	M	
Index (1 to k)	C	
Desired_value (1 to n)	M	
DLM_status (1 to n)		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

15.5.2.3 Parameters of the primitives

15.5.2.3.1 Variable_name

This array parameter specifies one or more variables (1 to n) which are to be assigned values from the corresponding elements of the Desired_value parameter. The selectable variables are operating parameters and counters; they are specified in Table 77 and Table 78.

Table 77 – Mandatory DLE-variables

Operating parameters	
Name	**Definition**
TS	DL-address of this station
Data_rate	Data signalling rate of this fieldbus
Medium_redundancy	Availability of redundant media
HW-Release	Hardware release number
SW-Release	Software release number
T_{SL}	Slot time
min T_{SDR}	Smallest station delay time
max T_{SDR} (Note 1)	Largest station delay time
T_{QUI} (Note 1)	Transmitter fall time (line state uncertain time) or repeater switch time
T_{SET} (Note 1)	Setup time
T_{TR} (Note 1)	Target rotation time
G (Note 1)	GAP update factor
in_ring_desired (Note 1)	Request entry into or exit out of the logical token ring
HSA (Note 1)	Highest station address of a master station on this fieldbus
max_retry_limit (Note 1)	Maximum number of retries
T_{CSI} (Note 2)	Clock synchronization interval
Isochronous_mode	Selects the operation of the isochronous mode
SYNCHT (Note 3)	Contents of the SYNCH DLPDU
T_{CT} (Note 3)	Isochronous cycle time
maxT_{SH} (Note 3)	Allowed maximal time shift

NOTE 1 This applies only to master stations

NOTE 2 This applies only to stations able to support clock synchronization

NOTE 3 This applies only to master stations able to work in isochronous mode

Table 78 – Optional DLE-variables

Counters	
Name	**Definition**
DLPDU_sent_count (Notes 1, 2)	Number of DLPDUs sent
Retry_count (Notes 1, 2)	Number of DLPDU repeats
DLPDU_sent_count_sr (Notes 1, 2)	List of numbers of DLPDUs sent per station
Error_count (Notes 1, 2)	List of numbers of no or erroneous responses per station
SD_count (Note 2)	Number of correct start delimiters (from PhE)
SD_error_count (Note 2)	Number of defective start delimiters (from PhE)

NOTE 1 This applies only to master stations.

NOTE 2 If a counter reaches its maximum value, then both this counter and its comparison counter are stopped. If a counter is reset to zero the related co-operative counter is also reset to zero, then these counters are free to count again.

15.5.2.3.2 Index

This conditional parameter is a selector for one or more entries (1 to k), used when a variable contains an array or list of values. Possible values for each entry of the list are 0 to 126.

NOTE The parameter is only used for the counters "DLPDU_sent_count_sr" and "Error_count".

15.5.2.3.3 Desired_value

This array parameter specifies the actual value to be written to the variables (1 to n) that are specified by the Variable_name parameter. This parameter specifies a list of one or more (1 to n) new values for the specified DLE-variables. The permissible value or range of values for each of these variables is specified in Table 79 and Table 80.

Table 79 – Permissible values of mandatory DLE-variables

Operating parameters	
Variable	**Range of values**
TS	one octet address field, DL-address value 0 to 126
Data_rate	9,6; 19,2; 31,25; 45,45; 93,75; 187,5; 500; 1500, 3000; 6000; 12000 kbit/s and others
Medium_redundancy	single; redundant
HW-Release	LE_HR; Visible String [length 0 to 32]
SW-Release	LE_SR; Visible String [length 0 to 32]
T_{SL}	52 to 2^{16}-1 (bit times)
min T_{SDR}	2^0 to 2^{16}-1 (bit times)
max T_{SDR}	2^0 to 2^{16}-1 (bit times)
T_{QUI}	0 to 2^8-1 (bit times)
T_{SET}	2^0 to 2^8-1 (bit times)
T_{TR}	2^0 to 2^{24}-1 (bit times)
G	1 to 100
in_ring_desired	TRUE; FALSE
HSA (Note 1)	2 to 126
max_retry_limit	0 to 8 (preferably 1)
T_{CSI}	2^0 to 2^{32}-1 (bit times)
Isochronous_mode	0 to 3 (see Table 81)
SYNCHT (Note 2)	2 (octets)
T_{CT}	2^0 to 2^{24}-1 (bit times), not exceeding 32 ms
maxT_{SH}	1 to 256 (bit times)
NOTE 1 Additionally the value 0 is possible for isochronous mode operation. NOTE 2 For further explanation, refer to IEC 61158-5.	

Table 80 – Permissible values of optional DLE-variables

Counters	
Variable	**Value**
DLPDU_sent_count	0
Retry_count	0
DLPDU_sent_count_sr[Index]	0
Error_count[Index]	0
SD_count	0
SD_error_count	0

Table 81 – Meaning of the values for the parameter isochronous_mode

Value	Meaning
0	No isochronous operation
1	Station shall work as isochronous Master
2	Station shall work as isochronous Master; delayed isochronous cycles are skipped
3	Station shall work as additional Master in a Fieldbus system working in isochronous mode

Table 82 and Table 83 give an overview of the most important operating parameters and their default values in the fieldbus system.

Table 82 – Default reaction times and operating parameters for a master station for asynchronous transmission

Operating parameters	**Data rate** (kbit/s)					
	≤ 187,5	**500**	**1500**	**3000**	**6000**	**12000**
T_{RDY} (t_{BIT})	< 11	< 11	< 11	< 11	< 11	< 11
T_{SDI} (t_{BIT})	≤ 70	≤ 150	≤ 200	≤ 250	≤ 450	≤ 800
default values						
T_{SL} (t_{BIT})	100	200	300	400	600	1000
min T_{SDR} (t_{BIT})	11	11	11	11	11	11
max T_{SDR} (t_{BIT})	60	100	150	250	450	800
T_{SET} (t_{BIT})	1	1	1	4	8	16
T_{QUI} (t_{BIT})	0	0	0	3	6	9
G	10	10	10	10	10	10
HSA	126	126	126	126	126	126
max_retry_limit	1	1	1	2	3	4

NOTE In a Multi-master environment some operating parameters may have to be set to higher values. Especially the T_{SL} may be extended because of token passing. Each master should have the possibility to receive the token and if necessary react (send a request DLPDU or token) properly.

Table 83 – Default reaction times and operating parameters for a slave station with asynchronous transmission

Operating parameters	Data rate (kbit/s)					
	≤ 187,5	500	1500	3000	6000	12000
max T_{SDR} (t_{Bit})	≤ 60	≤ 100	≤ 150	≤ 250	≤ 450	≤ 800
default values						
min T_{SDR} (t_{Bit})	11	11	11	11	11	11

In process application areas, coupling between the synchronous transmission and the asynchronous transmission (according to IEC 61158-2, Type 3) is done by Ph-couplers. In the case of the synchronous data rate 31,25 kbit/s, the correlated data rate on the asynchronous side should be 45,45 kbit/s or 93,75 kbit/s. Table 84 and Table 85 indicate the required range of parameters.

Table 84 – Default reaction times and operating parameters for master stations for coupling of synchronous and asynchronous transmission segments

Operating parameters	Data rate (kbit/s)		
	Synchronous	Asynchronous	
	31,25	45,45	93,75
T_{RDY} (t_{BIT})	< 11	< 11	< 11
T_{SDI} (t_{BIT})	≤ 70	≤ 70	≤ 70
default values			
Preamble_Extension (bit)	1		
T_{PTG} (t_{BIT})	0		
T_{SL} (t_{BIT})	150	640	2500 (7200) (Note)
min T_{SDR} (t_{BIT})	11	11	11
max T_{SDR} (t_{BIT})	100	400	1000 (3800) (Note)
T_{SET} (t_{BIT})	30	95	95
T_{QUI} (t_{BIT})	0	0	0
G	10	10	10
HSA	126	126	126
max_retry_limit	1	1	1
NOTE The value within parentheses is for a maximum DLSDU length between 65 and 244 octets in request and acknowledge/response DLPDUs. The value before the parentheses is for a maximum DLSDU length of 64 octets or less in request and acknowledge/response DLPDUs.			

Table 85 – Default reaction times and operating parameter for slave stations for coupling of synchronous and asynchronous transmission segments

Operating parameters	Data rate (kbit/s)		
	Synchronous	Asynchronous	
	31,25	45,45	93,75
max T_{SDR} (t_{BIT})	≤ 100	≤ 250	≤ 250
	default values		
Preamble_Extension (bit)	1		
T_{PTG} (t_{BIT})	0		
min T_{SDR} (t_{BIT})	11	11	11

The values for asynchronous transmission (45,45 and 93,75 kbit/s) are valid only for slaves operating directly on an asynchronous transmission segment.

15.5.2.3.4 DLM_status

This array parameter indicates for each variable in the corresponding request, the success or failure of that component of the associated Set Value service request. Permitted values for the individual components of this array parameter are specified in Table 86.

Table 86 – Values of DLM_status for the set value service

Short name	Status	Definition	temporary or permanent
OK	success	The variable has been set to the new value	–
NO	failure	The variable does not exist or could not be set to the new value	t/p
IV	failure	Invalid parameters in request	–

15.5.3 Get Value

15.5.3.1 Function

The DLMS-user passes a DLM-GET-VALUE request primitive to DL-management to read the current value(s) of one or more variables of the DLE. After receipt of this primitive DL-management tries to select the specified variables and to deliver their current values and passes a DLM-GET-VALUE confirm primitive to the DLMS-user to indicate the success or failure of the corresponding service request. This primitive returns as a parameter one or more of the requested variable values.

15.5.3.2 Types of primitives and parameters

Table 87 indicates the primitives and parameters of the Get Value service.

Table 87 – Get value primitives and parameters

DLM-GET-VALUE Parameter name	Request input	Confirm output
Variable_name (1 to n)	M	
Index (1 to k)	C	
Current_value (1 to n)		M
DLM_status (1 to n)		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

15.5.3.3 Parameters of the primitives

15.5.3.3.1 Variable_name

This array parameter specifies one or more variables (1 to n) whose values are to be read. The variables that may be selected are specified in Table 77 and Table 78. In master stations the additional variables specified in Table 88 also may be selected.

Table 88 – Additional mandatory DLE-variables in master stations

Operating parameters (mandatory)	
Name	**Definition**
T_{RR}	Real rotation time
LMS	List of master stations in the logical ring
GAPL	List all of stations in the own GAP

15.5.3.3.2 Index

This conditional parameter is a selector for one or more entries (1 to k), used when a variable contains an array or list of values. Possible values for each entry of the list are 0 to 126.

NOTE The parameter is only used in case of the variables "DLPDU_sent_count_sr" and "Error_count".

15.5.3.3.3 Current_value

This array parameter specifies the actual value of the (1 to n) variables that were specified by the Variable_name parameter of the corresponding request. The permissible value, or range of values, for each of these variables is specified in Table 79, Table 80 and Table 89.

Table 89 – Permissible values of the additional DLE-variables in master stations

Operating parameters	
Variable	**Range of values**
T_{RR}	2^0 to 2^{24}-1 (bit times)
LMS	preferably maximum 32 DL-addresses (0 to 126), optionally up to 127 DL-addresses
GAPL	max. 126 DL-addresses (0 to 126) inclusive DLE status
Counters	
Variable	**Range of values**
DLPDU_sent_count	0 to 2^{32}-1
Retry_count	0 to 2^{16}-1
SD_count	0 to 2^{32}-1
SD_error_count	0 to 2^{16}-1
DLPDU_sent_count_sr	max. 126 entries of 0 to 2^{32}-1
Error_count	max. 126 entries of 0 to 2^{16}-1

The value of the parameter Current_value is not defined, if the value of the parameter DLM_status is different from OK.

15.5.3.3.4 DLM_status

This array parameter indicates for each variable in the associated Get Value service request the success or failure of the execution of the service. Permitted values for this parameter are specified in Table 90.

Table 90 – Values of DLM_status for the get value service

Short name	Status	Definition	**temporary** or **permanent**
OK	success	The variable could be read	–
NO	failure	The variable does not exist or could not be read	t/p
IV	failure	Invalid parameters in request	–

15.5.4 Event

15.5.4.1 Function

The DLE informs DL-management that it has detected a faulty condition or an event. After that DL-management passes a DLM-EVENT indication primitive to the DLMS-user to inform it about important error conditions or events in the DLL.

15.5.4.2 Types of primitives and parameters

Table 91 indicates the primitive and parameters of the Event service.

Table 91 – Event primitive and parameters

DLM-EVENT Parameter name	Indication output
Event/Fault	M
T_{SH}	C

15.5.4.3 Parameters of the primitives

15.5.4.3.1 Event/Fault

This parameter specifies the event that took place or the fault type. The various events and faults are shown in Table 92.

Table 92 – Mandatory DLL events and fault types

Name		Definition
Time_out		No bus activity
Not_synchronized		Synchronization not detected within interval T_{SYNI}
In_ring	(Note 1)	This master stations has been taken into the logical token ring
Out_of_ring	(Note 1)	This master station has been taken out of the logical token ring not on its own initiative
GAP_event	(Note 1)	A change in the GAPL has occurred
Duplicate_address	(Note 1)	A duplication of this station's DL-address exists in the logical token ring
Faulty_transceiver	(Note 1)	The transmitter or receiver of this station is malfunctioning
Double_token	(Note 1)	While waiting for a response the master station receives a request DLPDU or token DLPDU
HSA_error	(Note 1)	master station receives a token DLPDU with DA or SA higher then local HSA
State_conflict	(Note 1)	MAC of master station has detected an internal state conflict
Synch	(Note 2)	Marks the beginning of an isochronous cycle
Synch_Delay	(Note 2)	Synch delay has occurred
NOTE 1 This event applies only to master stations		
NOTE 2 This event applies only to master stations in isochronous mode		

15.5.4.3.2 T_{SH}

This conditional parameter is only present, if the parameter Event/Fault has the value “Synch_Delay”. It contains the time shift which marks the time difference between the end of an isochronous cycle and sending of a synch message.The permissible values are shown in Table 93.

Table 93 – Permissible values of T_{SH}

Operating parameters	
Variable	**Range of values**
T_{SH}	0 to 2^{16}-1

15.5.5 Ident

15.5.5.1 Function

By means of a DLM-IDENT request primitive the DLMS-user requests DL-management to carry out a station identification.

If the user requests the identification of a remote station, the DLE issues a corresponding request DLPDU to this station by means of a request Ident with reply. The remote DLE immediately replies with a DLPDU containing the Ident data of the remote DLE. If the identification refers to the local DLE, the DLE immediately replies with the Ident data.

The DL-management returns the requested data to the DLMS-user by means of a DLM-IDENT confirm primitive to indicate the success or failure of the corresponding service request.

15.5.5.2 Types of primitives and parameters

Table 94 indicates the primitives and parameters of the Ident service.

Table 94 – Ident primitives and parameters

DLM-IDENT Parameter name	Request input	Confirm output
DL-addr	M	(Note)
Ident_list		M
DLM_status		M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.

15.5.5.3 Parameters of the primitives

15.5.5.3.1 DL-addr

This parameter specifies in the case of a remote request the DL-address of the remote station. The global address is not permitted. In the case of local requests the parameter specifies the local DLE's own DL-address (TS).

15.5.5.3.2 Ident_list

The value of the parameter Ident_list is not defined, if the value of the parameter DLM_status is different from OK. In all other cases the parameter specifies a list of values for the identification of a station as shown in Table 95:

Table 95 – Ident_list for the ident service

Item number	Type	Meaning	Definition
1	Le_vn	Length of Vendor_name in octets	0 to 196
2	Le_ct	Length of Controller_type in octets	0 to 196
3	Le_hr	Length of HW_release in octets	0 to 196
4	Le_sr	Length of SW_release in octets	0 to 196
5	Vendor_name	Name of the vendor	Visible String [length 0 to 196] (ISO 7 bit code, b8=0)
6	Controller_type	Sort of the controller	Visible String [length 0 to 196] (ISO 7 bit code, b8=0)
7	HW_release	Version number of the hardware	Visible String [length 0 to 196] (ISO 7 bit code, b8=0)
8	SW_release	Version number of the software	Visible String [length 0 to 196] (ISO 7 bit code, b8=0)

NOTE The overall length of the parameter Ident_list shall not exceed 200 octets.

15.5.5.3.3 DLM_status

This parameter indicates the success or failure of the associated Ident request service. Permitted values for this parameter are specified in Table 96 and Table 97.

Table 96 – Values of DLM_status for the ident service (local)

Short name	Status	Definition	temporary or permanent
OK	success	The Identification has been be carried out	–
LR	failure	Ident data not available at the local DLE	t
IV	failure	Invalid parameters in request	–

Table 97 – Values of DLM_status for the ident service (remote)

Short name	Status	Definition	temporary or permanent
OK	success	The Identification has been be carried out	–
NA	failure	No or no plausible reaction (ACK or RES) from remote station	t
DS	failure	Local DL-entity not in logical token ring or disconnected from line	p
LR	failure	Resources of the local DLE not available or not sufficient	t
NR	failure	Negative acknowledgement for Ident data, as not available in the remote DLE	p
IV	failure	Invalid parameters in request	–

15.5.6 DLSAP Status

15.5.6.1 Function

The DLMS-user passes a DLM-DLSAP-STATUS request primitive to DL-management to request the configuration of a DLSAP_index with respect to the DL-services. The DLE immediately responds by means of DLSAP status data of the addressed DLSAP_index.

DL-management passes the DLSAP configuration data to the DLMS-user by means of a DLM-DLSAP-STATUS confirm primitive to indicate the success or failure of the corresponding service request.

15.5.6.2 Types of primitives and parameters

Table 98 indicates the primitives and parameters of the DLSAP Status service.

Table 98 – DLSAP status primitives and parameters

DLM-DLSAP-STATUS Parameter name	Request input	Confirm output
DLSAP_index	M	(Note)
Access		M
Service_type (1 to n)		M
Role_in_service_list (1 to n)		M
DLM_status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.		

15.5.6.3 Parameters of the primitives

15.5.6.3.1 DLSAP_index

This parameter specifies the local DLSAP_index whose configuration is requested. All the DLSAP_index values 0 to 63, CS and NIL are permitted. If the configuration of the default DLSAP_index is to be requested, the parameter DLSAP_index shall have the value NIL.

15.5.6.3.2 Access

This parameter with the values ALL or 0 to 126 specifies the access protection. "ALL" means that all remote stations have access to this DLSAP_index. A single remote station (value 0..126 and, if applicable, region/segment address) means that only the specified remote station has access.

15.5.6.3.3 Service_type

This array parameter specifies the DL-services (1 to n) that are activated at the remote or local DLSAP_index. The following values are permissible:

SDA, SDN, SRD, MSRD and CS

15.5.6.3.4 Role_in_service_list

This array parameter specifies the configuration for the activated DL-services (1 to n). The following values are permissible:

- Initiator The station initiates the respective service, but does not respond to it.
- Responder The station responds to the respective service, but does not initiate it.
- Both The station both initiates and responds to the respective service.

15.5.6.3.5 DLM_status

This parameter indicates the success or failure of the associated DLM_status service request. Permitted values for this parameter are specified in Table 99.

Table 99 – Values of DLM_status for the DLSAP status service

Short name	Status	Definition	temporary or permanent
OK	success	The DLSAP Status could be read.	–
LR	failure	Ident data not available at the local DLE	t
LS	failure	The local DLSAP is not activated.	d
IV	failure	Invalid parameters in request.	–

15.5.7 DLSAP Activate

15.5.7.1 Function

This service provides the DLMS-user with the ability to activate and to configure a local DLSAP for individual DL-services other than the responder functions for the reply services (SRD and MSRD). The responder function for the SRD reply service is activated by means of the DLSAP Activate Responder service, while the responder function for the MSRD reply service is activated by means of the DLSAP Activate Subscriber service.

After receipt of the DLM-DLSAP-ACTIVATE request primitive from the DLMS-user, DL-management activates and configures the corresponding local DLSAP. Then DL-management passes a DLM-DLSAP-ACTIVATE confirm primitive to the DLMS-user to indicate the success or failure of the corresponding service request.

15.5.7.2 Types of primitives and parameters

Table 100 indicates the primitives and parameters of the DLSAP Activate service.

Table 100 – DLSAP activate primitives and parameters

DLM-DLSAP-ACTIVATE	Request	Confirm
Parameter name	input	output
S_SAP_index	M	(Note)
Access	M	(Note)
Service_list	M	
DLM_status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.		

15.5.7.3 Parameters of the primitives

15.5.7.3.1 S_SAP_index

This parameter specifies the local DLSAP that is to be activated and configured. The S_SAP_index values 0 to 63, CS and NIL are permissible.

15.5.7.3.2 Access

This parameter with the values ALL or 0 to 126 is used for access protection and specifies whether all remote stations (ALL) or only an individual remote station (0 to 126 and, if applicable, Region or Segment address) may have access to the DLSAP to send or request data. The parameter is only valid for the responder function(s), that is, when a component of a Role_in_service_list has a value of RESPONDER or BOTH.

15.5.7.3.3 Service_list

This compound parameter specifies a list of subparameters, see Table 101.

Table 101 – DLSAP activate service_list

Item number	Name
1	Service_list_length (4 to 3n+1)
2	First service_activate
3	First role_in_service
4	First DLSDU_length_list
5	Second service_activate
6	Second role_in_service
7	Second DLSDU_length_list
.	…
n	n-th service_activate
n+1	n-th role_in_service
n+2	n-th DLSDU_length_list
NOTE 1 ≤ n ≤ 4	

15.5.7.3.4 Service_activate

This subparameter specifies the DL-service that is to be activated for this DLSAP. The following values can be specified:

SDA, SDN, SRD, MSRD and CS.

NOTE The values SRD, MSRD and CS for the parameter Service_activate is allowed for master stations only.

15.5.7.3.5 Role_in_service

This subparameter specifies the configuration for the service to be activated. The following values can be specified:

- Initiator — The station initiates the respective service, but does not respond to it.
- Responder — The station responds to the respective service, but does not initiate it. Not permitted for SRD and MSRD.
- Both — The station both initiates and responds to the respective service. Not permitted for SRD and MSRD.

15.5.7.3.6 DLSDU_length_list

This compound subparameter specifies a list of maximum DLSDU lengths. Its structure is dependent on the DL-service activated as specified in 15.5.7.3.4.

For the SDA, SDN, SRD, MSRD and CS services the list has a fixed structure as shown in Table 102.

Table 102 – DLSAP activate DLSDU_length_list (SDA, SDN, SRD, MSRD and CS)

Item number	Name
1	Max_DLSDU_length_req_low
2	Max_DLSDU_length_req_high
3	Max_DLSDU_length_ind/cnf_low
4	Max_DLSDU_length_ind/cnf_high

Max_DLSDU_length_req_low and Max_DLSDU_length_req_high specify the maximum length of the low or high priority DLSDU, respectively, that can be passed to the initiator by means of the request primitive for the SDA, SDN, SRD, MSRD and CS services.

Max_DLSDU_length_ind/cnf_low and Max_DLSDU_length_ind/cnf_high specify the maximum length of the DLSDU to be received at an indication of the SDA, SDN and CS services at the responder, or at a confirmation of the SRD and MSRD service at the initiator.

The length of the DLSDU may be 0 to 246 octets. When using S_SAP_index, D_SAP_index and region/segment addresses a maximum of 242 octets is permissible.

Depending on the Service_activate and Role_in_service the combinations of DLSDU lengths, shown as columns, in Table 103 through Table 105, are permissible:

Table 103 – DLSDU lengths of SDA and SDN as used in the DLSAP activate service

Service: SDA and SDN															
Length	**Initiator**			**Responder**			**Both**								
1	x	–	x	–	–	–	x	–	x	–	x	–	x	x	x
2	–	x	x	–	–	–	–	x	–	x	–	x	x	x	x
3	–	–	–	x	–	x	x	–	–	x	x	x	x	–	x
4	–	–	–	–	x	x	–	x	x	–	x	x	–	x	x

NOTE 1 1 to 4 denote the item numbers of lengths as in Table 102

NOTE 2 x means length > 0; – means length = 0

Table 104 – DLSDU lengths of SRD and MSRD as used in the (master station) DLSAP activate service

Service: SRD and MSRD												
Length	**Initiator** (Note 1)											
1	–	–	–	x	–	–	x	x	x	x	–	x
2	–	–	–	–	x	x	–	x	x	–	x	x
3	x	–	x	x	–	x	–	x	–	x	x	x
4	–	x	x	–	x	–	x	–	x	x	x	x

NOTE 1 Responder only with DLSAP Activate Responder. Both not allowed

NOTE 2 1 to 4 denote the item numbers of lengths as in Table 102

NOTE 3 x means length > 0; – means length = 0

Table 105 – DLSDU lengths of CS as used in the DLSAP activate service

Service: CS			
Length	**Initiator**	**Responder**	**Both**
1	x	–	x
2	–	–	–
3	–	x	x
4	–	–	–
NOTE 1 1 to 4 deNote the item numbers of lengths as in Table 102			
NOTE 2 x means length > 0; – means length = 0			

15.5.7.3.7 DLM_status

This parameter indicates the success or failure of the associated DLSAP Activate service request. Permitted values for this parameter are specified in Table 106.

Table 106 – Values of DLM_status for the DLSAP activate service

Short name	**Status**	**Definition**	**temporary** or **permanent**
OK	success	The S_SAP is activated as requested	–
NO	failure	The S_SAP is not activated (already activated or resources not available or not sufficient)	t/p
IV	failure	Invalid parameters in request	–

15.5.8 DLSAP Activate Responder

15.5.8.1 Function

The DLMS-user passes a DLM-DLSAP-ACTIVATE-RESPONDER request primitive to DL-management to activate and to configure a local DLSAP for the responder function of the reply services (SRD and MSRD). DL-management activates and configures the corresponding local DLSAP as "Responder" and passes a DLM-DLSAP-ACTIVATE-RESPONDER confirm primitive to the DLMS-user to indicate the success or failure of the corresponding service request.

15.5.8.2 Types of primitives and parameters

Table 107 indicates the primitives and parameters of the DLSAP Activate Responder service.

Table 107 – DLSAP activate responder primitives and parameters

DLM-DLSAP-ACTIVATE-RESPONDER Parameter name	Request input	Confirm output
S_SAP_index	M	(Note)
Access	M	(Note)
DLSDU_length_list	M	
Indication_mode	M	
Publisher_enabled	M	
DLM_status		M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.

15.5.8.3 Parameters of the primitives

15.5.8.3.1 S_SAP_index

This parameter specifies the local DLSAP for which the Responder functions are to be activated. Any SRD or MSRD service instance which designates this local DLSAP will cause a corresponding SRD or MSRD service indication to be passed to the associated DLS-user. The S_SAP_index values 0 to 62 and NIL are permitted.

15.5.8.3.2 Access

This parameter has the same meaning as described in 15.5.7.3.2.

15.5.8.3.3 DLSDU_length_list

This compound parameter specifies a list of maximum DLSDU lengths. The structure of this list is shown in Table 108; it is identical to that shown in Table 102 but the semantics of the list components are slightly different.

Table 108 – DLSDU_length_list for the DLSAP activate responder service

Item number	Name
1	Max_DLSDU_length_req_low
2	Max_DLSDU_length_req_high
3	Max_DLSDU_length_ind_low
4	Max_DLSDU_length_ind_high

Max_DLSDU_length_req_low and Max_DLSDU_length_req_high specify the maximum length of the low or high priority DLSDU, respectively, that can be associated with the specified local DLSAP by a Reply-update request primitive.

Max_DLSDU_length_ind_low and Max_DLSDU_length_ind_high specify the maximum length of the DLSDU that can be received at the specified local DLSAP during an instance of the SRD and MSRD service respectively.

Each of these maximum lengths may be specified as 0 to 246 octets. When using S_SAP_index, D_SAP_index, and region/segment addresses a maximum of 242 octets is permissible.

The permissible combinations of DLSDU lengths, shown as columns, as a responder are specified in Table 109:

Table 109 – DLSDU length of SRD and MSRD as used in the DLSAP activate responder service

Service: SRD or MSRD												
Length	**Responder**											
1	x	–	x	x	–	–	x	x	x	x	–	x
2	–	x	x	–	x	x	–	x	x	–	x	x
3	–	–	–	x	–	x	–	x	–	x	x	x
4	–	–	–	–	x	–	x	–	x	x	x	x

NOTE 1 1 to 4 denote the item numbers of lengths as in Table 108

NOTE 2 x means length > 0; – means length = 0

15.5.8.3.4 Indication_mode

The parameter Indication_mode with the values ALL/DATA/UNCHANGED specifies whether, in the case of the SRD or MSRD service, the DL-DATA-REPLY indication primitive always shall be generated (ALL), or whether it shall be omitted (DATA) when both the received DLPDU (request) and the corresponding reply DLPDU contain null (zero length) DLSDUs.

The update of the access configuration of a local DLSAP is performed by setting this parameter to the value "UNCHANGED". In this case only the parameter "Access" is overwritten and all other parameters are unchanged.

15.5.8.3.5 Publisher_enabled

The parameter Publisher_enabled with the value TRUE specifies that in the case of the MSRD DLPDU the response DLPDU shall be sent as multicast. In case of the parameter Publisher_enabled has the value FALSE the MSRD DLPDU shall be ignored.

15.5.8.3.6 DLM_status

This parameter indicates the success or failure of the associated DLSAP activate responder service request. Permitted values for this parameter are shown in Table 110.

Table 110 – Values of DLM_status for the DLSAP activate responder service

Short name	Status	Definition	temporary or permanent
OK	success	The local DLSAP is activated as requested	–
NO	failure	Indication_mode "ALL/DATA": The local DLSAP is not activated successfully (already activated or resources not available or not sufficient) Indication_mode "UNCHANGED": The Access parameter of the local DLSAP is not overwritten, because the DLSAP was not activated before	t/p t/p
IV	failure	Invalid parameters in request	–

15.5.9 DLSAP Activate Subscriber

15.5.9.1 Function

The DLMS-user passes a DLM-DLSAP-ACTIVATE-SUBSCRIBER request primitive to DL-management to activate and to configure a local DLSAP for the subscriber function of the MSRD service. DL-management activates and configures the corresponding local DLSAP as Subscriber and passes a DLM-DLSAP-ACTIVATE-SUBSCRIBER confirm primitive to the DLMS-user to indicate the success or failure of the corresponding service request.

15.5.9.2 Types of primitives and parameters

Table 111 indicates the primitives and parameters of the DLSAP Activate Subscriber service.

Table 111 – DLSAP activate subscriber primitives and parameters

DLM-DLSAP-ACTIVATE-SUBSCRIBER Parameter name	Request input	Confirm output
S_SAP_index	M	(Note)
DLSDU_length_list	M	
DLM_status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.		

15.5.9.3 Parameters of the primitives

15.5.9.3.1 S_SAP_index

This parameter specifies the local DLSAP for which the Subscriber functions are to be activated. Any MSRD service instance which designates this local DLSAP will cause a corresponding MSRD service primitive DL-DXM-DATA-REPLY indication to be passed to the associated DLS-user. The S_SAP_index values 0 to 62 and NIL are permitted.

15.5.9.3.2 DLSDU_length_list

This compound parameter specifies a list of maximum DLSDU lengths. The structure of this list is shown in Table 112.

Table 112 – DLSDU_length_list for the DLSAP activate subscriber service

Item number	Name
1	Max_DLSDU_DXM_length_ind_low
2	Max_DLSDU_DXM_length_ind_high

Max_DLSDU_DXM_length_ind_low and Max_DLSDU_DXM_length_ind_high specify the maximum length of the DLSDU that can be received at the specified local DLSAP during an instance of the MSRD service.

Each of these maximum lengths may be specified as 0 to 246 octets. When using S_SAP_index, D_SAP_index, and region/segment addresses a maximum of 242 octets is permissible.

The permissible combination of DLSDU lengths, shown as columns, as a subscriber are specified in Table 113:

Table 113 – DLSDU lengths of MSRD as used in the DLSAP activate subscriber service (master and slave stations)

Service: MSRD			
Length	**Subscriber**		
1	X	–	X
2	–	X	X
NOTE 1 1 to 4 denote the item numbers of lengths as in Table 112 NOTE 2 x means length > 0; – means length = 0			

15.5.9.3.3 DLM_status

This parameter indicates the success or failure of the associated DLSAP activate subscriber service request. Permitted values for this parameter are specified in Table 114.

Table 114 – Values of DLM_status for the DLSAP activate subscriber service

Short name	Status	Definition	**temporary** or **permanent**
OK	success	The local DLSAP is activated as requested	–
NO	failure	The local DLSAP is not activated (already activated or resources not available or not sufficient)	t/p
IV	failure	Invalid parameters in request	–

15.5.10 DLSAP Deactivate

15.5.10.1 Function

The DLMS-user employs this service to deactivate all DL-services for a local DLSAP. After receipt of a DLM-DLSAP-DEACTIVATE request primitive from the DLMS-user, DL-management tests whether a reply DLPDU is still pending and deactivates the specified DLSAP for all services either directly (if no reply is pending) or after receipt of the pending reply. Immediately after this DL-management passes a DLM-DLSAP-DEACTIVATE confirm primitive to the DLMS-user to indicate the success or failure of the corresponding service request.

15.5.10.2 Types of primitives and parameters

Table 115 indicates the primitives and parameters of the DLSAP Deactivate service.

Table 115 – DLSAP deactivate primitives and parameters

DLM-DLSAP-DEACTIVATE Parameter name	Request input	Confirm output
S_SAP_index	M	(Note)
DLM_status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The descriptions in IEC 61158-4 and IEC 61158-5 assume that the indicated input parameter values of the request primitive are returned as output parameter values of the corresponding confirm primitive.		

15.5.10.3 Parameters of the primitives

15.5.10.3.1 S_SAP_index

This parameter specifies the local DLSAP that is to be deactivated for all DL-services. The S_SAP_index values 0 to 63, CS and NIL are permitted.

15.5.10.3.2 DLM_status

This parameter indicates the success or failure of the associated DLSAP-deactivate service request. Permitted values for this parameter are specified in Table 116.

Table 116 – Values of DLM_status for the DLSAP-deactivate service

Short name	**Status**	**Definition**	**temporary** or **permanent**
OK	success	The local DLSAP is deactivated	–
NO	failure	The local DLSAP has not been activated	P
IV	failure	Invalid parameters in request	–

16 Type 4: Data Link Service and concepts

16.1 Overview

The DLS provides for the transparent 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) 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 DLSDUs, nor does it interpret the structure or meaning of that information. It may, however, restrict the amount of information that can be transferred as an indivisible unit.

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

b) Reliable transfer of data. The DLS relieves the DLS-user from concerns regarding insertion, corruption, loss or duplication of data.

c) Prioritized data transfer. The DLS provides DLS-users with a means to prioritize requests.

d) Queue. The DLS provides the requesting DLS-user with a prioritized FIFO queue, where each queue item can hold a single DLSDU.

16.1.1 Overview of DL-naming (addressing)

A DLE is implicitly connected to a single PhE, and (separately) to a single DLSAP and associated DLS-user. A DLE always delivers received DLSDUs at the same DLSAP, and hence to the same DLS-user. This concept is illustrated in Figure 108.

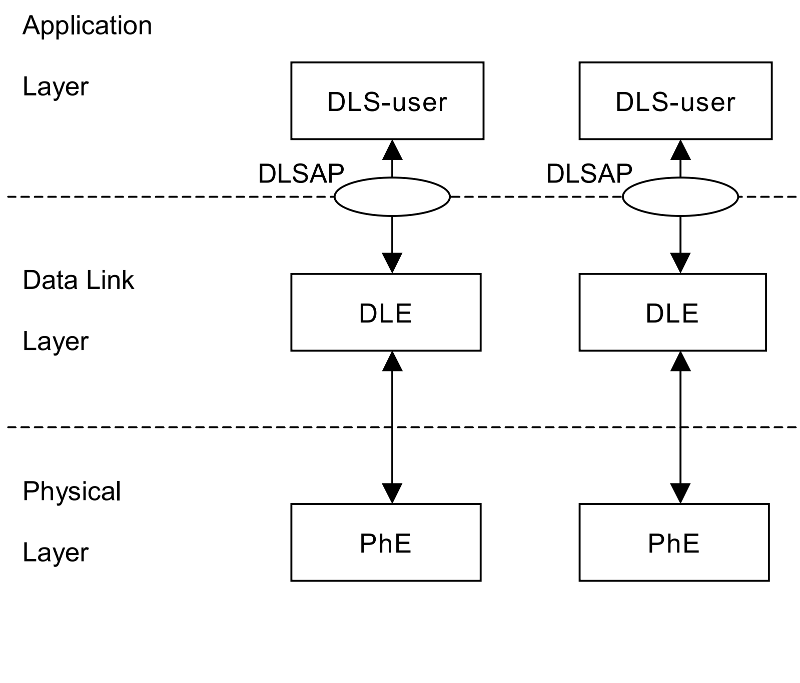


Figure 108– Relationship of PhE, DLE and DLS-users

Each DLE has a Node DL-address. Node DL-addresses uniquely identify DLEs within the local Link.

A DL-route-element is an octet, which can hold either a Node DL-address or a higher-layer address used by the DLS-user.

A destination-DL-route holds a sequence of DL-route-elements, describing the complete route to the destination DLSAP plus a local component meaningful to the destination DLS-user.

A source-DL-route holds a sequence of DL-route-elements, describing the complete route back to the source DLSAP plus a local component meaningful to the source DLS-user.

A full DL-route is defined as a destination-DL-route and a source-DL-route.

16.2 Types and classes of Data Link Service

There are two types of DLS:

a) a connectionless-mode data transfer service, providing confirmed and unconfirmed data transfer (defined in 16.5.2 and 16.5.3);

b) a management service. The type 4 management service provides services for reading and writing managed objects (DLM-SET and DLM-GET requests), as defined in Clause 11.

16.3 Functional classes

The functional class of a DLE determines its capabilities, and thus the complexity of conforming implementations. Two functional classes are defined:

a) Simple class, including only responder functionality (server).

b) Normal class, including initiator and responder functionality (client and server, also called peer).

16.4 Facilities of the connectionless-mode Data Link Service

The DLS provides a means of transferring DLSDUs of limited length from one source DLS-user to one or more destination DLS-users. 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 (other than limited length) imposed by the DLS.

16.5 Model of the connectionless-mode Data Link Service

16.5.1 General

A defining characteristic of data-link connectionless-mode unitdata transmission is the independent nature of each invocation of the DLS.

Only one type of object, the unitdata object, can be submitted to the DLS-provider for transmission.

The DLS-user issuing a request primitive specifies whether the request is to be confirmed by the remote DLS-user, or not. This is specified in the Destination-DL-route and Source-DL-route parameters of the DL-UNITDATA request primitive. If the remote DLS-user confirms a request, it does this by issuing a new, independent DL-UNITDATA request primitive.

16.5.2 Unconfirmed request

The DLE of the requesting DLS-user forms a DLPDU which includes the submitted DLSDU and sends the DLPDU to the receiving DLE. The receiving DLE delivers the received DLSDU to the DLS-user by a DL-UNITDATA indication primitive. The value of the Confirmation expected parameter of this indication is FALSE.

16.5.3 Confirmed request

The DLE of the requesting DLS-user forms a DLPDU which includes the submitted DLSDU and sends the DLPDU to the receiving DLE. The receiving DLE delivers the received DLSDU to the DLS-user by a DL-UNITDATA indication primitive. The value of the Confirmation expected parameter of this indication is TRUE.

If the receiving DLS-user is unable to handle the indication immediately, the receiving DLS-user should issue a DL-UNITDATA response primitive within the time specified by Maximum Indication Delay.

If the receiving DLS-user either

a) does not reply with a DL-UNITDATA response primitive or a DL-UNITDATA request primitive within the interval Maximum Indication Delay from receipt of the triggering DL-UNITDATA indication primitive, or

b) does reply with a DL-UNITDATA response primitive within the interval Maximum Indication Delay from receipt of the triggering DL-UNITDATA indication primitive

then the receiving DLE transmits an acknowledging DLPDU to the original requesting DLE. The following actions depend on whether the replying DLE is of Simple or Normal class.

1) If the replying DLE is of Simple class, the acknowledge DLPDU from the replying DLE specifies "WAIT". In this case, the original requesting DLE requeues the original request DLPDU at the lowest possible priority for retransmission at the next opportunity. When the replying DLS-user has prepared the response, it should await the repeated request from the original requesting DLE, and this time reply by issuing a DL-UNITDATA request primitive within the time interval Maximum Indication Delay.

The action in the original requesting DLE of requeuing the original request for retransmission is repeated as long as the replying DLE keeps responding with "WAIT" acknowledges, or until retransmission has been attempted for the time interval specified in the Maximum Retry Time configuration parameter.

2) If the replying DLE is of Normal class, the acknowledge DLPDU from the replying DLE specifies "RESPONSE COMES LATER / ACKNOWLEDGE". In this case, the original requesting DLE does nothing further. When the DLS-user at the replying DLE has prepared the response, it should reply by issuing a DL-UNITDATA request primitive. The replying DLE forms an appropriate DLPDU and queues it for transmission at the first opportunity.

16.6 Sequence of primitives

16.6.1 Constraints on sequence of primitives

This subclause defines the constraints on the sequence in which the primitives defined in 16.6.2 and Table 117 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur.

Table 117 – Summary of DL-connectionless-mode primitives and parameters

Service	Service subtype	Primitive	Parameter
Data Transfer	Unitdata	DL-UNITDATA request	(*in* Destination-DL-route, Source-DL-route, Priority, Maximum retry time, Control status, Data field format, DLSDU)
		DL-UNITDATA indication	(*out* Destination-DL-route, Source-DL-route, Confirmation expected, Control status, Data field format, DLSDU)
		DL-UNITDATA response	(*in* Destination-DL-route, Source-DL-route)

16.6.2 Relation of primitives at the end-points of connectionless service

16.6.2.1 General

A request primitive issued at one DLSAP will have consequences at one or more other DLSAPs. These relations are summarized in Figure 109 and Figure 110.

16.6.2.2 Confirmed and unconfirmed UNITDATA request

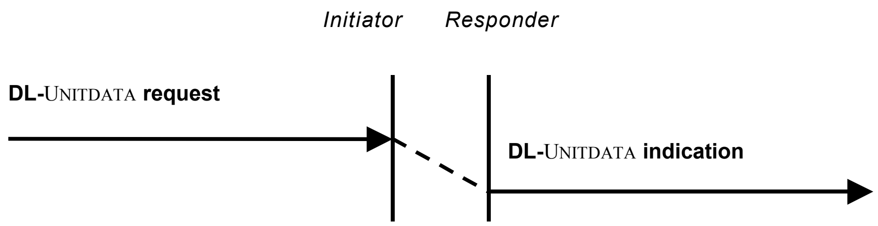


Figure 109 – Confirmed and unconfirmed UNITDATA request time-sequence diagram

16.6.2.3 Repeated Confirmed UNITDATA request

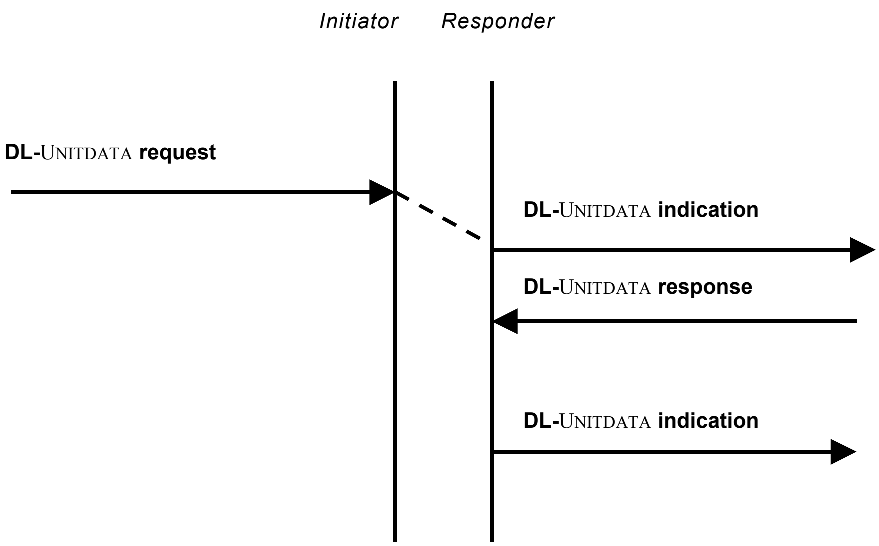


Figure 110– Repeated confirmed request time-sequence diagram

16.6.3 Sequence of primitives at one DLSAP

The possible overall sequences of primitives at one DLSAP are defined in the state transition diagram of Figure 111.

NOTE Since there is no conformance to IEC 61158-3 (see 1.3), the use of a state transition diagram to describe the allowable sequences of service primitives does not impose any requirements or constraints on the internal organization of any implementation of the service.

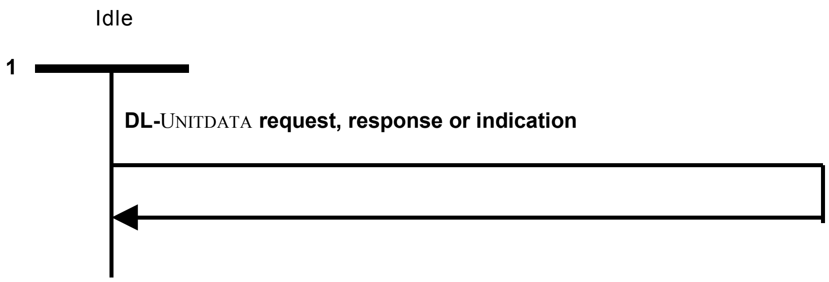


Figure 111 – State transition diagram for sequences of primitives at one DLSAP

16.7 Connectionless-mode data transfer functions

16.7.1 General

DL-connectionless-mode unitdata service primitives are used to transmit independent DLSDUs from one DLS-user to one or more other DLS-users. Each DLSDU is transmitted in a single DLPDU. The DLSDU is independent in the sense that it bears no relationship to any other DLSDU transmitted through another invocation of the DL-service by the same DLS-user. The DLSDU is self-contained in that all the information required to deliver the DLSDU is presented to the DL-provider, together with the user data to be transmitted, in a single service access.

16.7.2 Types of primitives and parameters

16.7.2.1 General

Table 118 indicates the types of primitives and the parameters needed for the DL-connectionless-mode unitdata service.

Table 118 – Unitdata transfer primitives and parameters

DL-UNITDATA Parameter name	Request input	Indication output	Response input
Destination-DL-route	M	M	M
Source-DL-route	M	M	M
Priority	U		
Maximum retry time	U		
Confirmation expected		M	
Control-status	M	M(=)	
Data-field-format	M	M(=)	
Data unit (DLSDU)	M	M(=)	

16.7.2.2 Request primitive

This primitive causes the DLE to create a DLPDU and append it to the transmit queue for transmission at the first opportunity, after all preceding higher-priority DLPDUs in the queue.

If the transmission fails, the DLE delivers error information to the requesting DLS-user by a DL-UNITDATA indication primitive, provided that the requesting DLS-user expects a confirmation. The Control-status parameter of this indication specifies the reason for failure. The DLSDU parameter of this indication is null.

16.7.2.3 Indication primitive

This primitive is used by a receiving DLE to deliver a received DLSDU to the addressed DLS-user.

16.7.2.4 Response primitive

This primitive is use by a receiving DLS-user which

a) is not able to generate an expected confirmation within an appropriate time interval, and

b) wishes to indicate that it has received the requesting DLSDU and is preparing a response.

16.7.2.5 Destination-DL-route

This parameter is a sequence of DL-route-elements defining the route to the responder (request) or to the requestor (response). (see 3.7.2)

This parameter of a request can also indicate that the requesting DLS-user does not expect a confirmation from the receiving DLS-user. If the value of one or more Node DL-addresses in the Destination-DL-route is equal to the Broadcast-Node DL-address, the requesting DLS-user does not expect a confirmation.

NOTE DL-route elements holding Node DL-addresses can hold the value of the Broadcast-Node DL-address. This means that a broadcast DLPDU can be transmitted to all DLEs on a local link.

16.7.2.6 Source-DL-route

This parameter is a sequence of DL-route-elements, defining the reverse route to the requestor (request) or responder (response). (see 3.7.13)

This parameter can also indicate that the requesting DLS-user does not expect a confirmation from the receiving DLS-user. If the value of the last element of the Source-DL-route is equal to the No-Confirm-Node DL-address, the service is unconfirmed.

16.7.2.7 Priority

This user-optional parameter specifies the initial priority of the request. The DLPDU resulting from the request is appended to the queue in the DLE at a position based on the value of this parameter. This value can be any integral number between 0 and 255. The DLPDU is placed in front of all DLPDUs already in the queue having a lower priority, where 255 indicates the highest possible priority.

16.7.2.8 Maximum retry time

This user-optional parameter specifies how long the local DLE should retry the transmission of the request as a result of WAIT acknowledge DLPDUs received from the remote DLE. Wait acknowledge DLPDUs are a result of the DL-UNITDATA response primitive described in 16.7.2.4. A DLE retries a transmission by re-appending the DLPDU to the transmit queue, but with a priority of 0 (the lowest possible).

16.7.2.9 Confirmation expected

This parameter indicates to the receiving DLS-user whether the requesting DLS-user expects a confirmation or not. If the requesting DLS-user expects a confirmation, the receiving DLS-user should issue a new, independent DL-UNITDATA request primitive.

Confirmation expected can hold the following values:

TRUE, indicating the requesting DLS-user expects a confirmation.

FALSE, indicating the requesting DLS-user does not expect a confirmation.

16.7.2.10 Control-status

This parameter is one octet. The requesting DLS-user should specify a value where at least one of the low-order three bits is non-zero. If the accompanying DLSDU is conveyed successfully to the addressed DLS-user, then this parameter will be delivered unchanged in the corresponding parameter of the indication to the receiving DLS-user.

If the transmission of a request fails and the requesting DLS-user expects a reply DLSDU, the DLE delivers error information to the requesting DLS-user by a DL-UNITDATA indication primitive. The value conveyed in this corresponding parameter of an indication is specified in Table 119:

Table 119 – Control-status error codes

Value (hexadecimal)	Meaning
00	failure — no response
18	failure — wait too long
38	failure — route error
80	failure — frame check error
88	failure — overrun/framing error
90	failure — link short circuit
98	failure — DLE is simple class
A0	failure — out of synchronization
*X*1 – *X*7	success (where *X* = any digit value)

16.7.2.11 Data-field-format

This parameter holds one octet of information for the DLS-user on the interpretation of the DLSDU contents. The parameter of a request will be delivered unchanged in the corresponding parameter of the indication to the receiving DLS-user.

16.7.2.12 DLSDU

This parameter conveys DLS-user data; its size may be any integral number of octets between 0 and 63.

17 Type 6: Data Link Service and concepts

17.1 Fundamental concepts

17.1.1 Underlying model of services

17.1.1.1 General

This clause describes the real-time aspects of the connection-mode Data Transfer service, for which design and performance are optimized. Many of these concepts also apply to the DLM-connectionless-mode service.

This DLL is a slotted Time Division Multiple Access (TDMA) system. Time on the bus is divided into equal length intervals called **slots**. Each DLE listens to traffic, and may send traffic, only during a slot assigned to it. A local link has 2^{LBL} (typically, LBL ≤ 16) slots during one bus cycle. Figure 112 illustrates an example system.

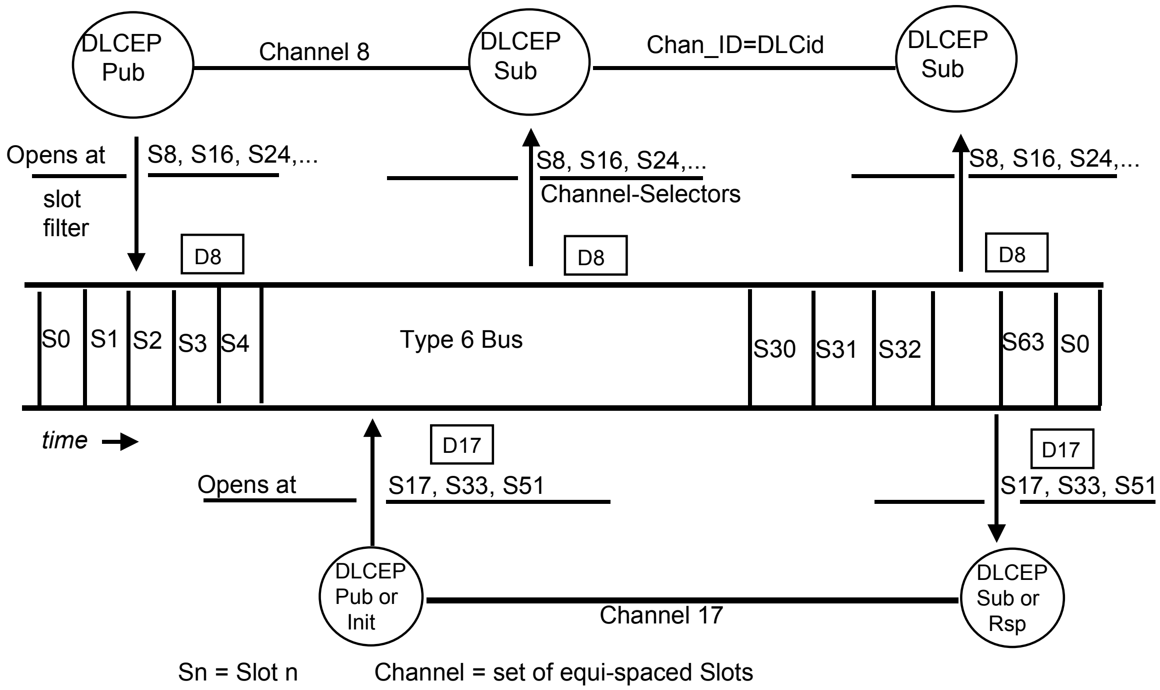


Figure 112 – EXAMPLE: TDMA bus operation using slots and channels.

17.1.1.2 Bus access and data transmission

Bus access, data transmission and data reception is coordinated by each DLE based on a shared perception of which Channel and Slot now occupies the bus.

17.1.1.3 Channels

Each variable measured by a sensor, or stimulus sent to an actuator, uses a dedicated channel, which is configured for, and occupies, one or more slots per Bus-Cycle.

When one of the slots which is assigned to one of the channels in a node is noted by that node DLE's Bus-Cycle clock, it either transmits or receives (as assigned) the data which is present on the bus during that slot.

NOTE 1 The slots on a local bus can be distributed by configuration among typically 10s or 100s or 1000s (up to Bus-Cycle-Len-1) of channels to achieve the scan rates and variable coverage needed for the transducers and higher layer protocols used in the specific real world system.

NOTE 2 Each transducer (device) can have multiple measurement and command “virtual signals” that represent different functions or different sensor locations.

When multiple nodes (or DLEs) listen on one channel, this channel is known as a Multicast Channel.

All slots on the bus are the same length for one bus configuration and mode of operation. Each slot may be assigned to only one channel. Each Channel may be scanned one to many times during each Bus-Cycle.

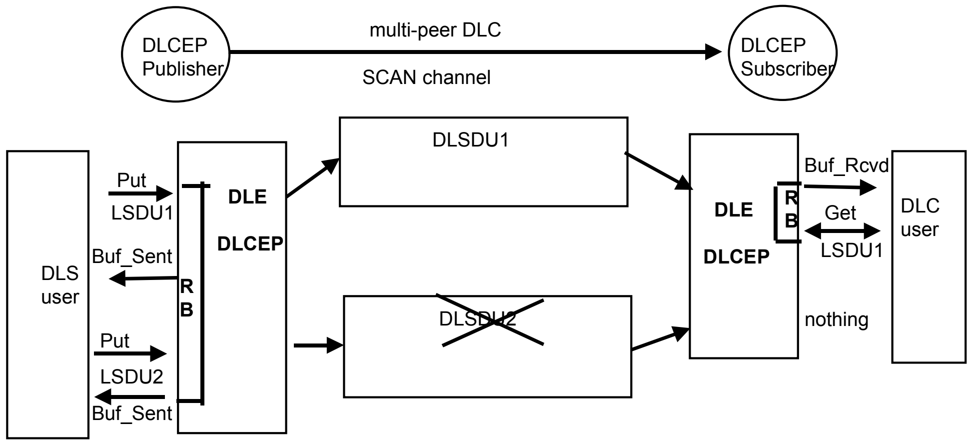
Figure 113 Illustrates slot and channel assignments and operation of a bus with a highly abbreviated bus cycle.

Start **abbreviated bus cycle 1** **end** shown for an artificial Bus-Cycle with 16 Slots and 3 Scan Classes the typical Bus-Cycle has 64K (65536) Slots and 15 Scan Classes																**Start of bus cycle 2**	
Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	Slot 9	Slot 10	Slot 11	Slot 12	Slot 13	Slot 14	Slot 15	Slot 0	**End-user selected scan rates**
Ch A		Ch A		Ch A		Ch A		Ch A		Ch A		Ch A		Ch A	*Bus Sync period*	Ch A	ScanClass 1, Channel A @ max. Scan Rate
	Ch B				Ch C				Ch B				Ch C		*Bus Sync period*		Scan-Class 2, Channels B, C @ 25% of max. Scan Rate
			Ch D				*Ch E Syn Ch*				Ch F				*Bus Sync period*		*Scan-Class 3, Channels D ,E, F* *@ 12,5% of max Scan Rate*

Figure 113 – Fundamental concepts: slots, channels, scan classes, bus cycles and bus synchronization

17.1.1.3.1 SCAN channel-class

SCAN class channels transfer short, scanned, state data, for example, Measurements and Commands with minimal DL-overhead using the 1WAY and UNITARY methods. Its Data Delivery is UNORDERED as defined in 8.2.2.2. The operation of the SCAN class is depicted in Figure 114.

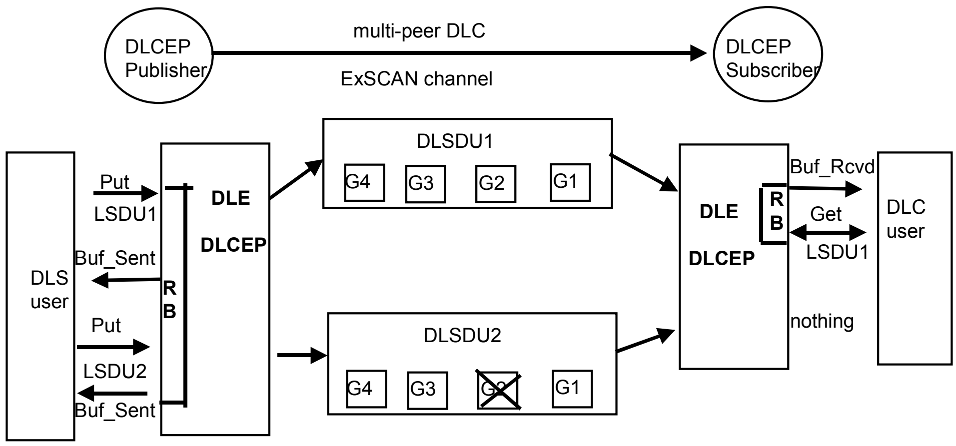


Legend
RB:Retentive Buffer

Figure 114 – The operation of the SCAN channel-class and its DLS-user interactions

17.1.1.3.2 ExSCAN channel-class

ExSCAN channel-class transfers long, scanned, state data, for example, Measurements and Commands, whose length exceeds Max-Data-Length, with minimal DL-overhead using the 1WAY and GRANULAR methods. Its Data Delivery is UNORDERED as defined in 8.2.2.2. Figure 115 depicts the operation of the ExSCAN channel-class.



Legend
RB:Retentive Buffer

Figure 115 – The operation of the ExSCAN channel-class and its DLS-user interaction

17.1.1.3.3 General Purpose Acknowledged (GPA) channel-class

The General Purpose Acknowledged (GPA) channel-class provides reliable communications for higher layer protocols that depend on classical one-way DL-data-delivery. It uses the 2WAY and GRANULAR methods. Its Data Delivery QoS is CLASSICAL as defined in 8.2.2.2. Operation of the GPA channel-class is depicted in Figure 116.

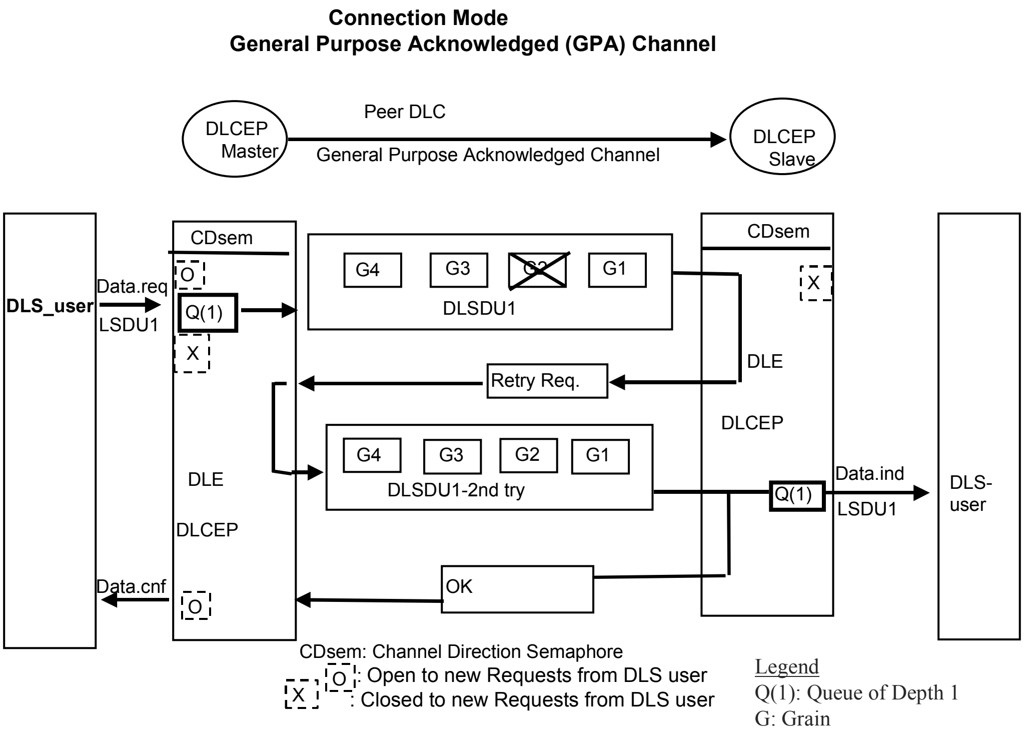


Figure 116 – The operation of the GPA channel-class and its DLS-user interactions

17.1.1.3.4 General Purpose Confirmed (GPC) channel-class

The General Purpose Confirmed (GPC) channel-class provides reliable communications for higher layer protocols that depend on classical two-way DL-data-delivery. It uses the 2WAY and GRANULAR methods. Its Data Delivery QoS is CLASSICAL as defined in 8.2.2.2. Operation of the GPC channel-class is depicted in Figure 117.

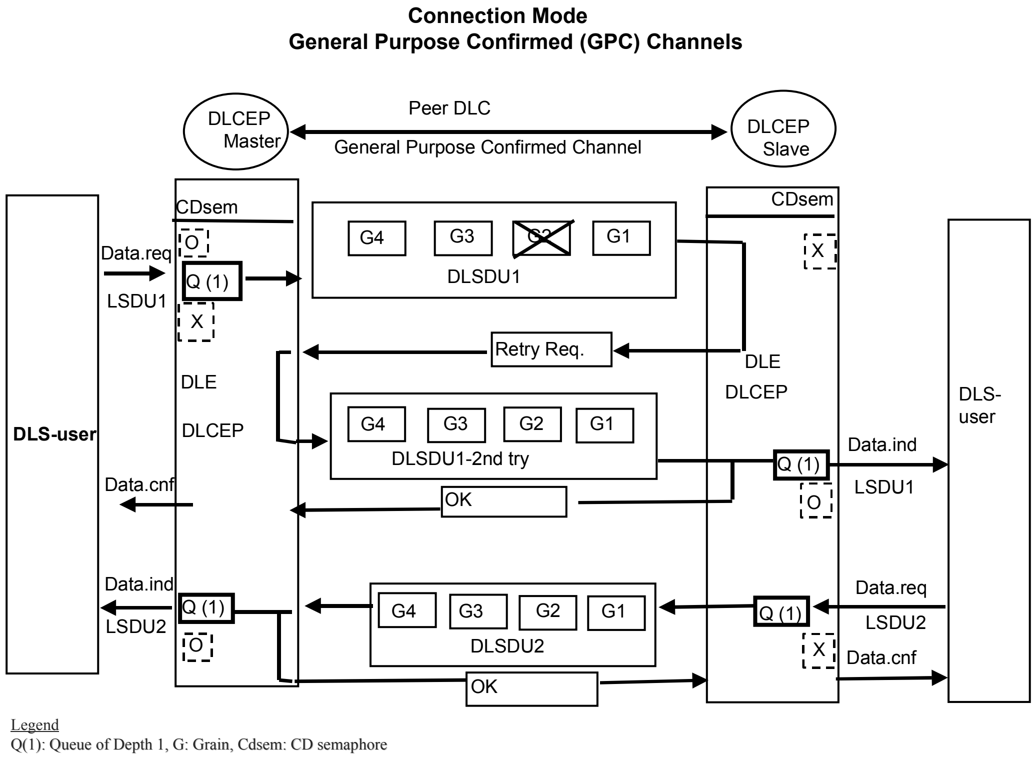


Figure 117 – The operation of the GPC channel-class and its DLS-user interactions

17.1.1.4 Publishers and subscribers

Only one DLE (the Publisher) may transmit DLPDUs in any given slot on SCAN and ExSCAN channels but any number of DLEs (the Subscriber(s)) may listen to the DLPDUs. These concepts are illustrated in Figure 112. This is equivalent to stating that each SCAN and ExSCAN Channel may have one Publisher and one or more Subscribers.

17.1.2 Additional attributes of the real-time connection-mode DLS

a) The DLS assumes that the same variable is always transmitted on a SCAN or EXSCAN channel by the same DLS-user (except as in c) below).

b) The DLS assumes that the data conveyed across a SCAN or EXSCAN channel is of a state nature and is repeated sufficiently often that retry procedures are not justified.

NOTE Data can be transferred much faster and with much lower system bus overhead on SCAN or EXSCAN channels.

c) If multiple DLS-users share the role of Publisher on a SCAN or EXSCAN channel, these DLS-users use a separate path to allocate use of the channel between them.

d) The DLS-user in the calling node is responsible for priority on GPA or GPC channels, submitting messaging transfers in the desired order.

e) The DLS effectively guarantees correct completion of transfers on any GPC or GPA channel. In the event that errors occur and retries are required, the Called and Calling DLEs will “lock up” the affected channel until either a correct transfer or a power loss or a management action occurs.

17.1.3 Other features

17.1.3.1 Bus synchronization and the conductor

The Conductor's DLE and each other node's DLE cooperate to Synchronize all nodes on a local link or extended link.

The Conductor node starts-up and synchronizes bus operation. The Conductor is the one Host configured or chosen by the contention protocol as the (active) Conductor.

The Conductor does not actively schedule traffic on the link, as is done by a Type 1 LAS.

17.1.3.2 Synchronous bus operation with minimal jitter

17.1.3.2.1 General

features that support this QoS include:

17.1.3.2.2 Common slot length

One common Slot Length is used for a given bus configuration and mode of operation. This Slot length is dynamically configurable in units of 1 Bit-Time.

17.1.3.2.3 Compressed bus overhead

All bus overhead is compressed into a single very short DLPDU, plus a moderate frequency channel. This allows the bus overhead to “fit neatly” into the scan sequence, as shown in Figure 113.

17.1.3.2.4 Scan flexibility

Scan rates are binary sub-multiples of the Bus-Cycle

This provides great flexibility in assigning Channel Scan Rates and reduces DLE complexity.

17.1.3.2.5 Very high occupancy synchronous scanning

One Channel can use up to 50% of the total bus Slots. This supports very high speed, truly synchronous scanning of I/Os.

17.1.3.2.6 Tunable scan rate

Slot-Length can be increased above its minimum value for this DLSDU-Length and topology to “fine tune” the bus Scan-Rate.

17.1.4 Model of real-time connection-mode services

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

In particular, the DLS provides for the following:

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

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

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

c) Reliable transfer of data. The DLS relieves the DLS-user from concerns regarding insertion or corruption of data and loss, duplication or misordering of data. In some cases of very rare unrecovered errors in the Data Link Layer, duplication or loss of DLSDUs may occur.

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

d) Quality of Service (QoS) selection. The DLS provides the Bus-configuration DLS-user a method of selecting a quality of service for the transfer of data. QoS is specified by the configuration of DLCEPs. QoS-set provides a means of assuring that DLCEPs used on the same connection are properly matched.

e) DL-Addressing and Channel-identifiers. The DLS allows the Bus Configuration DLS-user to specify the DLCEPs between which a DLC is to be established. Local DL-addresses have only regional significance within a specific DL-segment. Extended DL-addresses have global significance within a specific DL-system over a set of bridged DL-segments. Therefore, it is appropriate to define a global addressing structure that is used for all DLM-connectionless (Configuration) traffic.

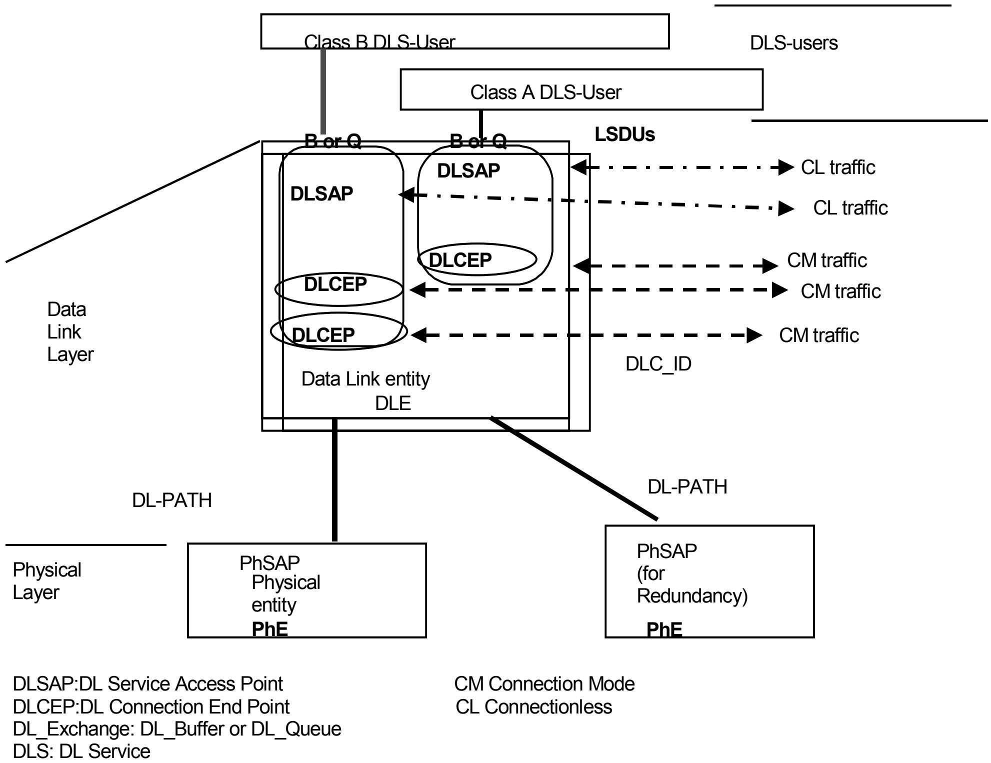
f) Common time sense. The DLS can provide the DLS-user with a sense of time that is common to all DLS-users on the extended link.

g) Queues and buffers. The DLS can provide the sending or receiving DLS-user with either a FIFO queue of depth one, e.g. a non-retentive buffer, or a retentive buffer, where each queue item or buffer can hold a single DLSDU.

17.1.5 Overview of DL-naming and addressing

17.1.5.1 General

DL-names, known conventionally as DL-addresses, are identifiers from a defined 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-service-access-points (DLSAPs), Individual data-link-connection-end-points (DLCEPs), and data-link-entities (DLEs). The relationships of these objects and the types of service they can support are shown in Figure 118.



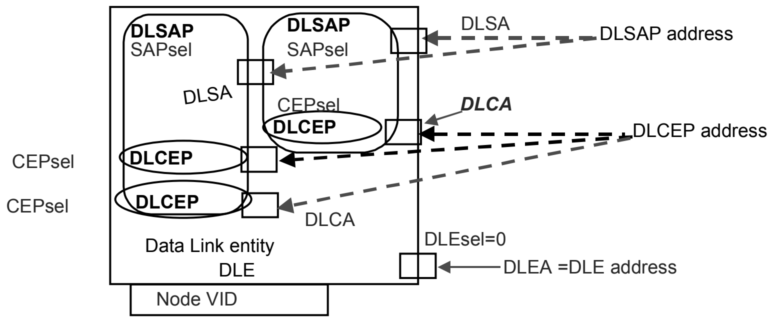
NOTE 1 DLSAPs and PhSAPs are depicted as adjacent to the boundary between two adjacent layers, with physical DLSDU interchange memory denoted as Buffer-or-Queue.

NOTE 2 This figure also shows the relationships of DL-paths and PhSAPs.

Figure 118 – Relationships of DLSAPs, DLCEPs, DLEs and DLS-users, and allowed classes of traffic from DLSAPs and DLCEPs

17.1.5.2 DLM-connectionless DL-addresses

The components of a DLM-connectionless DL-address, and the Node Visible Identification are shown in Figure 119.



DL address relationships

DLSAP_address(DLSA)=DLEA+SAPsel
DLCEP_address(DLCA)=DLEA+CEPsel
DLEA=DLSAP(0)_address[DLEsel=0]
node local_ID=DLEBA=VID= High order 8 bits of DLEA
node global_ID=Net_ID.VID, Net_ID={X}

DL address Symbols

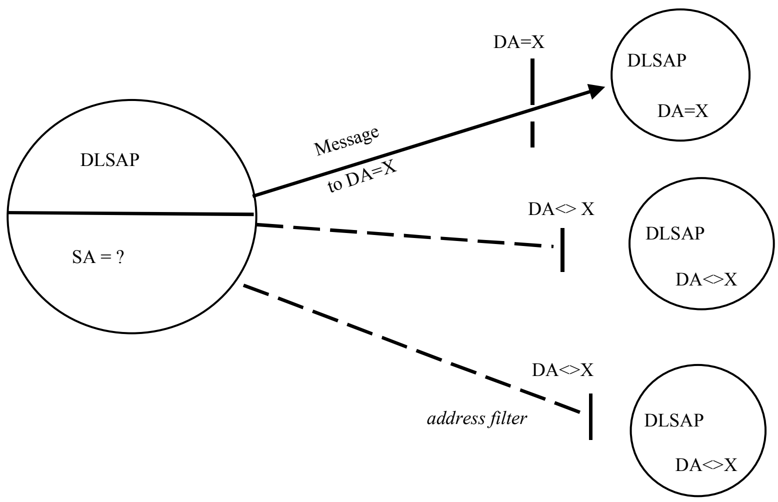
DLSAP: DLSAP_address
DLCA: DLCEP_address
DLEA: DLE_address
SAPsel: DLSAP selector within DLE
CEPsel: DLCE selector within DLE
DLEsel: DLE selector (=0)
Node VID: externally visible identification of DLE & its contained objects
DLEBA: DLE Base Address

Figure 119 – DLM-connectionless DL-addresses and node visible identification

NOTE 1 DLM-connectionless DL-addresses are used for the DLM-connectionless service, including configuration.

NOTE 2 A single DL-entity may have multiple DLSAP-addresses associated with a single DLSAP.

The operation of DLM-connectionless addresses is shown in Figure 120.



Legend
DA: Destination DLSAP Address
SA: Source DLSAP Address

Figure 120 – DLM-connectionless DL-addressing operation

17.1.5.3 Predefined DLM-connectionless addresses

17.1.5.3.1 General

These addresses are strictly hierarchical for ease of recognition.

Each field [shown in brackets] has the most significant selector value written as its leftmost component.

Predefined DLM-connectionless addresses, as depicted in Figure 119 are:

17.1.5.3.2 local-DLEA-address

[Net-ID={0}] [DLEA with SAP-sel = 0]

The numerical value of local DLEA-address = DLSAP(0)-address [SAP-sel=0] of the referenced Node.

17.1.5.3.3 local-DLSAP-address

[Net-ID={0}] [DLEA] [SAP-sel]

The numerical value of local-DLSAP-address = [Net-ID={0}] [DLEA of the referenced Node] [SAP-sel of the referenced SAP within that Node]

17.1.5.3.4 local-DLCEP-address

[Net-ID={0}] [DLEA] [SAP-sel] [CEP-sel]

The numerical value of local-DLCEP-address = [Net-ID={0}] [DLEA of the referenced Node] [SAP-sel of the referenced SAP within that Node] [CEP-sel of the referenced CEP within that Node]

17.1.5.3.5 global DLEA-address

[Net-ID={X}] [DLEA with SAP-sel = 0]

17.1.5.3.6 global DL-address

[Net-ID={X}] [DLEA] [SAP-sel]

17.1.5.3.7 global DLCEP-address

[Net-ID={X}] [DLEA] [SAP-sel] [CEP-sel]

17.1.5.4 DL-channel-identifiers

17.1.5.4.1 General

A DL-channel on any link has a single identifier, which represents the image of that channel on the referenced link, as depicted in Figure 121.

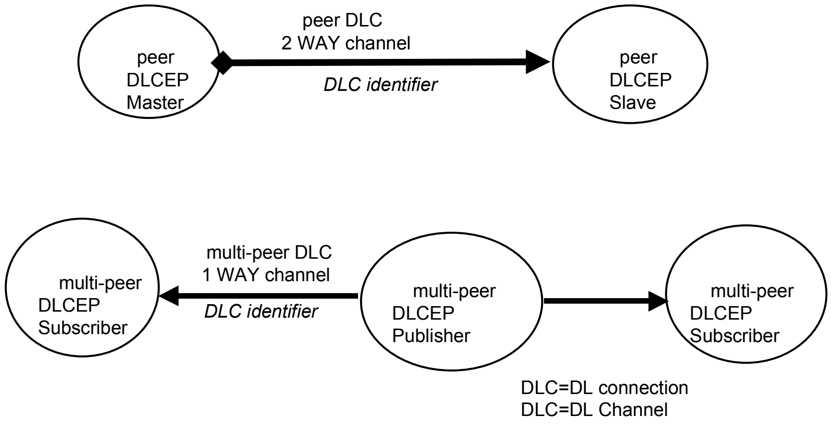


Figure 121 – Peer and multipoint DLCs, their DLC-identifiers and related DLCEP types and roles

NOTE 1 A DLC is identified by a DLC-ID, which is unique on any link. DLC-IDs are used for real-time connection mode communications.

NOTE 2 This figure also shows the types of DLCs and the types and roles of their participating DLCEPs.

17.1.5.4.2 DLC-id on the local Link

DL-identifier = DLC-ID on local link

the local DLC-id for a channel is the slot number of the first slot assigned to this channel on the local link.

The relationship between each DLC-id and the DLCEP-addresses used for configuration of this channel is known to, and managed by, the Bus Configurator.

17.1.5.4.3 DLC-id on a link within the extended link

The DLC-id for a channel on any link is the slot number of the first slot assigned to this channel on this link.

The relationship between each DLC-id and the DLCEP-addresses used for configuration of this channel is known to, and managed by, the Bus Configurator.

NOTE A DLC and its DLC-ID may appear on one or more lower links of the extended link without appearing on the upper links, for example, in Figure 122, N2 or N3 may be invisible from H1 but visible from N6.

17.1.6 Overview of buffers and queues

17.1.6.1 General

A dual-access interlocked memory block which is formally part of a DLSAP and provides the physical interface between the DLS-user (or DLMS-user) and the DLE in a local node for one direction of a channel. It is accessed by its assigned DLS-user (or DLMS-user) as specified in 17.3 and 17.4. For DLSDUs, this memory may be either a buffer or a queue. For DLMSDUs this memory is a queue of depth 1.

17.1.6.2 Non-retentive-buffer, or queue of depth 1

The DLSDU (or DLMSDU) presence is cleared by the DLE immediately upon transmission or upon extraction by the DLS-user (or DLMS-user)

17.1.6.3 Retentive-buffer

The DLSDU presence is not cleared by the DLE. This is a responsibility of the DLS-user

17.1.7 Overview of DL-subnetwork structuring

This is identical to the DL-subnetwork structuring described in 6.1.1.

Figure 122 shows an example extended link, including both real and virtual network topologies.

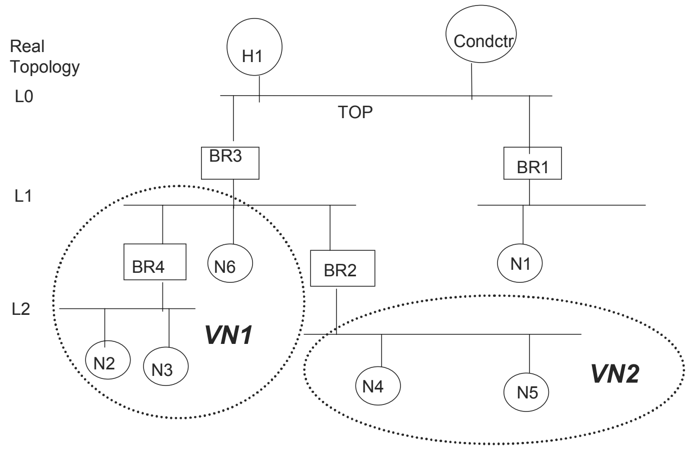


Figure 122 –Real and virtual topologies of an extended link

NOTE In a virtual topology, a DLC and its DLC-ID may appear on one or more lower links of the extended link without appearing on the upper links. For example, in Figure 122, N2 or N3 may be invisible from H1 but visible from N6. Virtual topologies are useable only for Connection mode traffic.

Net-ID is the hierarchical identification of the subnetwork on which a given node resides in the real network topology, see Figure 122. It identifies the local link in a Bridged DL-subnetwork. It identifies the local link as seen from the TOP level (L0) of the extended link looking DOWN, and is of the form:

$\{X\} \Rightarrow \{[a].[b].[c]\}$

when the Max-Level =3, as shown in Figure 122. Allowed values of Net_ID are specified in 17.4.4.4.5.

NOTE 1 max-Level limits the depth of the maximum nest of links in the real topology .

NOTE 2 max-Level does not limit the depth of the maximum nest of virtual topology channels.

17.1.8 Types and classes of Data Link Service

There are three types of DLS:

a) a synchronized configured connection-mode data transfer service with four classes of service, defined in 17.3;

b) a DLM-connectionless-mode data transfer service with one class of service, defined in 17.4;

c) real-time coordination services, defined in 17.5, including::

   1) a high accuracy common sense of time for all nodes

   2) an autonomous high accuracy time stamping service for alarms and events;

   3) one or more high accuracy trigger signals, for internal or external use, which are synchronized with all other nodes on the extended link.

17.2 Quality of service (QoS)

17.2.1 General

The term "Quality of Service" (QoS) refers to those aspects that are under the direct control of the DLS-provider.

17.2.2 QoS attribute categories

The categories of QoS attributes supported are:

Static and Inherent QoS attributes are inherently satisfied by the operation of the protocol.

Quasi-static QoS attributes and are selected during Bus-configuration on a per DLCEP basis.

Dynamic QoS attributes may be specified by the DLS-user independently at each DLS invocation.

17.2.3 Connection-mode QoS

17.2.3.1 DLSDU-length (dynamic QoS attribute)

All data transfer requests specify a dynamic DLSDU-length parameter which is used to determine whether the DLS-user-data can be conveyed on the associated channel.

For SCAN channels this maximum is max-data-length.

For GPC, GPA and ExSCAN channels this maximum is determined by max-DLSDU-length.

NOTE The following equation applies to all GRANULAR transfers. The notation used is from Excel5.

max-DLSDU-length = MIN (max-DLSDU-size from Table 120, max-DLSDU-buffer-size) (3)

Max-DLSDU-buffer-size is set by the conformance-class in use. Its range is 64 to 1792 octets in the same increments as shown for max-DLSDU-size in Table 120.

Table 120 – Correspondence of max-DLSDU-size and max-data-length for GPC, GPA and ExSCAN channels

Max-data-length (octets)	Max-DLSDU-size-(octets)
2	64
3	128
4 or 5	256
6 to 9	512
10 to 17	1024
18 to 33	1792

When max-DLSDU-length is such that the requested number of requested DLS-user-octets can not be conveyed, a confirmation with status of "data too long" is made available to the calling DLS-user.

17.2.3.2 QoS-set (dynamic QoS attribute)

The dynamic QoS-set parameter allows

a) matching of the configured usage of the two peer DLCEPs associated with a GPA or GPC channel and

b) matching of the calling DLCEP associated with a GPA or GPC channel with its associated DLS-user.

c) matching by the called DLS-user with its associated DLCEP on a GPA or GPC channel.

Its value may be assigned as 0 to 15 inclusive.

The semantics associated with values of QoS-set are determined by the conformance-class in use rather than by this standard.

NOTE 1 The called DLS-user may check that the value the QoS-set parameter of the DL-Data-indication meets its requirements. A status of "timeout" may indicate a failure to match the QoS-set of called DLS-user and DLCEP.

NOTE 2 All negotiations of DLS QoS-set are conducted by matching values of QoS-set. There is no concept of one QoS-set value being “included” within another QoS-set value, or of further examination of QoS-set values.

17.2.3.3 Response-request (dynamic QoS attribute)

This dynamic attribute on a GPC channel is used by the DLS-user associated with the Master DLCEP to request and allow the DLS-user associated with the Slave DLCEP to send a reply DLSDU.

Its value may be

TRUE — a reply DLSDU is requested or

FALSE — no reply DLSDU is requested or allowed

17.2.3.4 max-confirm-delay (quasi-static QoS attribute)

Max-confirm-delay is a quasi-static parameter which specifies an upper bounds on the maximum time duration permitted for the completion of a sequence of connection-oriented DLS primitives on a GPC channel. :

It may be configured to the values 10 to 31 max-transfer-periods, inclusive.

NOTE The minimum value allows for passage through a 4 level Bridged network and the round down of max-transfer-period.

Max-transfer-period is the period required to transfer one maximum length DLSDU on a GPC channel on the local link, including the time for the called DLS-user on a GPC channel to form its reply DLSDU. Its value is rounded down to 64 Scan-periods.

17.2.4 Inherent QoS of real-time and connection-mode DLS

17.2.4.1 General

This DL-protocol has additional inherent QoS attributes for real time and connection-mode services. The current Bus-configuration primarily determines the values achieved for these attributes. Thus the QoS of these services is INHERENT rather than STATIC, DYNAMIC OR SEMI-STATIC. None of the inherent QoS parameters are visible at the DLS-user to AE interface. These QoS parameters are:

17.2.4.2 Scan rate

For any given data rate, the maximum aggregate scan rate of any bus is effectively fixed.

NOTE This aggregate scan rate can be divided in many ways among a diverse population of Channels to meet user needs.

17.2.4.3 Jitter

This is any deviation from a regular synchronous pattern in either a) sampling a transducer b) transmission of data values for the SCAN & ExSCAN classes c) imposition of a trigger signal. Jitter should be measured both from Time to Time within one Node on the length and across multi-nodes on length.

With DLE clocks accurate to 100 parts per million, and with one Sync-slot per 128 slots, the maximum Jitter on a local bus, absent a catastrophic failure, will be one Bit-time.

Jitter on extended links will be increased by one or two Bit-times per Bridge traversed.

17.2.4.4 Node clock and time stamp aliasing

Time Stamp Aliasing is any difference between the Time Stamps placed on the same event by two different DLEs.

With DLE clocks accurate to 100 parts per million, and with one Sync-slot per 128 slots, the maximum aliasing between Node clocks, and thus Time Stamps, on the local link, absent a catastrophic failure will be two Bit-times.

Aliasing on extended links will be increased by one or two Bit-times per Bridge traversed.

17.2.4.5 Transmission Delay

This is any unwanted delay in transmitting transducer readings on SCAN and ExSCAN channels .

Transmission delay on the local link can be less than or equal to two Bit-times, the resolution at which a Trigger can be synchronized to the Scan Period of a channel.

Transmission delay on an extended link depends on the loading of the DL-subnetwork and the rationing of data rates from the TOP to the BOTTOM levels and the Bus-configuration process. The minimum delay is one Slot-Time per bridge traversed.

17.3 Connection mode services

17.3.1 General

This section provides a conceptual definition of the services provided by the DLS-provider to the DLS-user(s). Nothing in this section shall be construed to constrain the actual implementations of the interactions at the DLS-provider to DLS-user interface.

17.3.2 Primitives and parameters used on SCAN and ExSCAN channels

The Primitives and Parameters used on SCAN and ExSCAN channels are shown in Table 121, Figure 114 and Figure 115:

Table 121 – Primitives and parameters used on SCAN and ExSCAN channels

Function	Location	Primitive	Direction	Parameters
submit DLSDU	Publisher	DL-PUT	to DLS	DLC-identifier DLSDU-buffer-identifier DLSDU-length
confirm to publishing DLS-user	Publisher	DL-BUFFER-SENT	from DLS	DLC-identifier DLSDU-buffer-identifier Status
notify subscribing DLS-user	Subscriber(s)	DL-BUFFER-RECEIVED	from DLS	DLC-identifier
retrieve DLSDU	Subscriber(s)	DL-GET	to/from DLS	DLC-identifier DLSDU-buffer-identifier DLSDU-length
NOTE In this table, time increases from top to bottom.				

17.3.3 Specification of primitives and parameters used on SCAN and ExSCAN channels

17.3.3.1 General

The data transfer primitives used on SCAN and ExSCAN channels are DL-PUT, DL-BUFFER-SENT, DL-BUFFER-RECEIVED and DL-GET.

17.3.3.2 DL-Put

DL-Put conveys the DLSDU to be transmitted from the local DLS-user to the Publisher DLE. Its parameters are specified in Table 122.

Table 122 – DL-Put primitive and parameters

DL-PUT Parameter name	Request input
DLC-identifier	M
DLSDU-buffer-identifier	M
DLSDU-length	M

The Publisher DLE transmits DLPDUs via the channel identified by DLC-identifier of the DL-PUT primitive.

The Publisher DLE retrieves DLSDUs submitted by the local DLS-user from the retentive buffer identified by DLSDU-buffer-identifier of the DL-PUT primitive.

17.3.3.3 DL-Buffer-sent

DL-Buffer-sent conveys the local status of the transmission from the Publisher DLE to its local DLS-user. Its parameters are specified in Table 123.

Table 123 – DL-BUFFER-SENT primitive and parameters

DL-BUFFER-SENT Parameter name	Confirm output
DLC-identifier	M
DLSDU-buffer-identifier	M
Status	M

Publisher DLEs notify their local DLS-user of which channel has transmitted a DLPDU by the DLC-identifier of the DL-BUFFER-SENT primitive.

Publisher DLEs notify their local DLS-user of which DLSDU has been transmitted by the DLSDU-buffer-identifier of the DL-BUFFER-SENT primitive.

Status may take on the values

"Data-sent"

"Data-too-long"

17.3.3.4 DL-Buffer-received

DL-BUFFER-RECEIVED notifies the local DLS-user that the Subscriber DLE has just received a DLSDU. Its parameters are specified in Table 124.

Table 124 – DL-BUFFER-RECEIVED primitive and parameters

DL-BUFFER-RECEIVED Parameter name	Indication output
DLC-identifier	M

Subscriber DLEs notify their local DLS-user of which channel has received a DLPDU via the DLC-identifier of the DL-BUFFER-RECEIVED primitive.

17.3.3.5 DL Get

DL-GET conveys the just received DLSDU from the Subscriber DLE to its local DLS-user. Its parameters are specified in Table 125.

Table 125 – DL-GET primitive and parameters

DL-GET	Request	
Parameter name	input	output
DLC-identifier	M	
DLSDU-buffer-identifier		M
DLSDU-length		M

The local DLS-user notifies the Subscriber DLE as to which channel's data it wishes to retrieve by the DLC-identifier of the DL-GET primitive.

Subscriber DLEs notify their local DLS-user as to which DLSDU has been received on DLC-identifier via the DLSDU-buffer-identifier of the DL-GET primitive.

17.3.4 Primitives and Parameters used on GPA channels

The Primitives and Parameters used on GPA channels are shown in Table 126 and Figure 116:

Table 126 – Primitives and parameters used on GPA channels

Function	Location	Primitive	Direction	Parameters
submit DLSDU	Master	DL-DATA-request	to DLE	DLC-identifier DLSDU queue identifier DLSDU-length QoS-set
notify calling peer DLS-user	Master	possible DL-DATA-confirm	from DLE	DLC-identifier DLSDU queue identifier Status
notify called peer and provide DLSDU	Slave	DL-DATA-indication	from DLE	DLC-identifier DLSDU queue identifier DLSDU-length QoS-set
confirm to calling peer DLS-user	Master	DL-DATA-confirm	from DLE	DLC-identifier DLSDU queue identifier Status
NOTE In this table, time increases from top to bottom.				

17.3.5 Primitives and Parameters used on GPC channels

The Primitives and Parameters used on GPC channels are shown in Table 127, Table 128 and Figure 117:

Table 127 – Primitives and parameters used on GPC channels

Function	Location	Primitive	Direction	Parameters
submit DLSDU and request reply	Master	DL-DATA-REQUEST	to DLE	DLC-identifier DLSDU queue identifier DLSDU-length QoS-set response-request
notify calling peer DLS-user	Master	possible DL-DATA-CONFIRM	from DLE	DLC-identifier DLSDU queue identifier Status
notify called peer DLS-user, provide DLSDU and request response	Slave	DL-DATA-INDICATION	from DLE	DLC-identifier DLSDU queue identifier DLSDU-length QoS-set response-request
confirm to calling peer DLS-user	Master	DL-DATA-CONFIRM	from DLE	DLC-identifier DLSDU queue identifier Status
submit DLSDU	Slave	DL-DATA-REQUEST	to DLE	DLC-identifier DLSDU queue identifier DLSDU-length QoS-set
notify calling peer DLS-user	Slave	possible DL-DATA-CONFIRM	from DLE	DLC-identifier DLSDU queue identifier Status
notify called peer DLS-user and provide DLSDU	Master	DL-DATA-INDICATION	from DLE	DLC-identifier DLSDU queue identifier DLSDU-length QoS-set
confirm to calling peer DLS-user	Slave	DL-DATA-CONFIRM	from DLE	DLC-identifier DLSDU queue identifier Status

NOTE In this table, time increases from top to bottom.

Table 128 – Primitives and parameters used to disconnect GPC channels

Function	Location	Primitive	Direction	Parameters
submit Disconnect request	Master	DL-DISCONNECT-REQUEST	to DLE	DLC-identifier
notify peer DLS-user	Slave	DL-DISCONNECT-INDICATION	from DLE	DLC-identifier
submit Disconnect request	Slave	DL-DISCONNECT-REQUEST	to DLE	DLC-identifier
notify peer DLS-user	Master	DL-DISCONNECT-INDICATION	from DLE	DLC-identifier
submit Disconnect request	Slave	DL-DISCONNECT-REQUEST	to DLE	DLC-identifier
notify peer DLS-user	Master	DL-DISCONNECT-INDICATION	from DLE	DLC-identifier
submit Disconnect request	Master	DL-DISCONNECT-REQUEST	to DLE	DLC-identifier
notify peer DLS-user	Slave	DL-DISCONNECT-INDICATION	from DLE	DLC-identifier
NOTE 1 In this table, time increases from top to bottom. NOTE 2 DL-Disconnect primitives take precedence over DL-Data primitives.				

17.3.6 Specification of DL primitives and parameters used on GPC and GPA channels

17.3.6.1 General

The primitives used on GPA and GPC channels are DL-DATA-request, DL-DATA-indication and DL-DATA-confirm.

17.3.6.2 DL-Data

17.3.6.2.1 General

DL-DATA-request conveys the DLSDU to be transmitted from the local DLS-user to the calling DLE.

DL-DATA-indication conveys the just received DLSDU from the called DLE to its local DLS-user.

DL-DATA-confirm conveys the status of the immediately previous transmission from the calling DLE to its local DLS-user. The parameters of the DL-DATA-primitives are specified in Table 129.

Table 129 – DL-DATA primitives and parameters

DL-DATA	Request	Indication	Confirm
Parameter name	input	output	output
DLC-identifier	M	M	M
DLSDU queue identifier	M	M	M
DLSDU-length	M	M (=)	
QoS-set	M	M (=)	
response-request	C	C (=)	
Status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

17.3.6.2.2 DLC-identifier

The Calling DLE transmits DLPDUs via the channel identified by DLC-identifier of the DL-DATA-request primitive. Calling DLEs notify their local DLS-user of the status of the transmission by the DLC-identifier of the DL-DATA-confirm primitive.

Called DLEs notify their local DLS-user of which channel has received a DLPDU via the DLC-identifier of the DL-DATA-indication primitive.

17.3.6.2.3 DLSDU-queue-identifier

The Calling DLE retrieves DLSDUs submitted by the local DLS-user from the queue of depth 1 identified by DLSDU-queue-identifier of the DL-DATA-request primitive. Calling DLEs notify their local DLS-user of which DLSDU has been transmitted by the DLSDU-queue-identifier of the DL-DATA-confirm primitive.

Called DLEs notify their local DLS-user as to which DLSDU has been received on DLC-identifier via the DLSDU-queue-identifier of the DL-DATA-indication primitive.

17.3.6.2.4 Status

Status may take on the values:

"Data-transferred-correctly"

"Data-too-long"

"QoS-set mismatch"

"Timeout"

17.3.6.3 DL-Disconnect

17.3.6.3.1 General

DL-Disconnect-request conveys a disconnect request from the local DLS-user to the peer DLS-user on a GPC channel

DL-Disconnect-indication conveys a disconnect request from the peer DLS-user to the local DLS-user on a GPC channel. The parameters of the DL-DISCONNECT primitives are specified in Table 130.

Table 130 – DL-DISCONNECT primitives and parameters

DL-DISCONNECT Parameter name	Request input	Indication output
DLC-identifier	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

17.3.6.3.2 DLC-identifier

The Calling DLE transmits DL-DISCONNECT-requests via the channel identified by DLC-identifier of the DL-DISCONNECT-request primitive.

Called DLEs notify their local DLS-user of which channel has received a DL-DISCONNECT-request via the DLC-identifier of the DL-DISCONNECT-indication primitive.

17.4 Connectionless management service

17.4.1 General

This DLM-service is an adaptation of the Connectionless service defined ISO/IEC 8886 and in Clause 9 of this document. It is shown in Figure 123. It is restricted to DL-management and is used solely during the START and CHECK states, where a standardized set of bus timing parameters (Bus-Config) that support DLM-connectionless service exist. It is used for configuration of all nodes, channels, forwarding entities and triggers.

17.4 provides a conceptual definition of the services provided by the DLS-provider to the DLMS-user(s). Nothing in this section shall be construed to constrain the actual implementations of the interactions at the DLS-provider to DLMS-user interface.

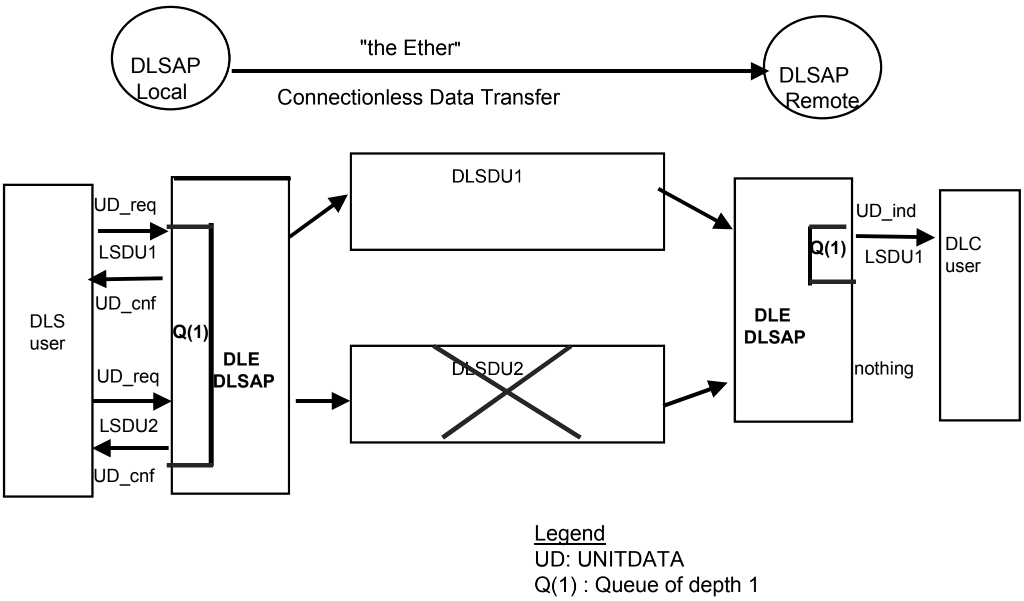


Figure 123 – Operation of DLM-connectionless service and its user interactions

Each DLSAP recognizes one or more DLSAP-addresses which the DLE may associate with any contained DLCEP addresses, as shown in Figure 119.

17.4.2 DLM-Connectionless QoS

17.4.2.1 DLSDU-length, (dynamic QoS attribute)

All DLM-connectionless data transfer requests specify an associated dynamic, DLMSDU-length parameter which is used to determine whether the DLMSDU can be conveyed on the associated channel.

The max-DLMSDU-length is 21 octets

17.4.3 Primitives and parameters used for DLM-connectionless service

17.4.3.1 General

The Primitives and Parameters used for DLM-Connectionless service are shown in Table 131.

Table 131 – Primitives and parameters of the DLM-connectionless service

Function	Location	Primitive	Direction	Parameters
submit DLMSDU	Local DLSAP	DLM-UNITDATA-request	to DLE	DLC-identifier DLMSDU queue identifier DLMSDU-length Calling Address Called Address
notify calling peer DLMS-user	Local DLSAP	DLM-UNITDATA-confirm	from DLE	DLC-identifier DLMSDU queue identifier Status
notify called peer and provide DLMSDU	Remote DLSAP	DLM-UNITDATA-indication	from DLE	DLC-identifier DLMSDU queue identifier DLMSDU-length Calling Address Called Address
NOTE In this table, time increases from top to bottom.				

17.4.3.2 Unitdata

17.4.3.2.1 General

The primitives used for DLM-connectionless service are DLM-UNITDATA-request, DLM-UNITDATA-indication and DLM-UNITDATA-confirm.

DLM-UNITDATA-request conveys the DLMSDU to be transmitted from the local DLMS-user to the calling DLE.

DLM-UNITDATA-indication conveys the just received DLMSDU from the called DLE to its local DLMS-user.

DLM-UNITDATA-confirm conveys the local status of the immediately prior transmission from the calling DLE to the local DLMS-user.

The parameters of the DLM-UNITDATA primitives are shown in Table 132.

Table 132 – Primitives and parameters of the DLM-UNITDATA service

DLM-UNITDATA	Request	Indication	Confirm
Parameter name	input	output	output
DLC-identifier	M	M	M
DLMSDU queue identifier	M	M	M
DLMSDU-length	M	M (=)	
Calling Address	M	M (=)	
Called Address	M	M (=)	
Status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

17.4.3.2.2 DLC-identifier

The Calling DLE transmits DLPDUs via the channel identified by DLC-identifier of the DLM-UNITDATA-request primitive. Calling DLEs notify their local DLMS-user of the local status of the transmission by the DLC-identifier of the DLM-UNITDATA-confirm primitive.

Called DLEs notify their local DLMS-user of which channel has received a DLPDU via the DLC-identifier of the DLM-Unitdata-indication primitive.

17.4.3.2.3 DLMSDU-queue-identifier

The Calling DLE retrieves DLMSDUs submitted by the local DLMS-user from the queue of depth 1 identified by DLMSDU-queue-identifier of the DLM-UNITDATA-request primitive. Calling DLEs notify their local DLMS-user of which DLMSDU has been transmitted by the DLMSDU-queue-identifier of the DLM-UNITDATA-confirm primitive.

Called DLEs notify their local DLMS-user as to which DLMSDU has been received on DLC-identifier via the DLMSDU-queue-identifier of the DLM-UNITDATA-indication primitive.

17.4.3.2.4 Calling address

Calling Address is the address of the Local DLSAP. It is formatted according to 17.1.5.2 and 17.1.5.4.

17.4.3.2.5 Called address

Called Address is the address of the Remote DLSAP. It is formatted according to 17.1.5.2 and 17.1.5.4.

17.4.3.2.6 Status

Status may take on the values:

"Data-transmitted"

"Data-too-long"

17.4.4 DLM-connectionless DL-addressing

17.4.4.1 General

Addressing in DLM-connectionless mode is shown in Figure 119, which shows DLM-connectionless DL-addresses, symbols and their relationships, and Node-Visible-Identification.

17.4.4.2 Hierarchical address regimes

The four main DLM-connectionless address regimes are:

a) DLE-addresses – (Node-VIDs)
b) DLSAP-selector
c) DLCEP-selector
d) Link-addresses – (Link topology identifiers, Net-ID)

17.4.4.3 General form of DLM-connectionless DL-addresses

DLM-connectionless DL-addresses are of the form and encoding shown in Figure 124.

The General form of DLM-connectionless DL-addresses is:

Link-address . DLE-address . DLSAP-selector . DLCEP-selector (10)

! first bit transmitted			last bit ! transmitted
Link-address = Net-ID	**DLE address = Node-VID**	**DLSAP-selector**	**DLCEP-selector**
1 to 3 octets	1 octet	1 octet	1 octet

Figure 124 – General form and encoding of DLM-connectionless DL-addresses

17.4.4.4 DL-addresses - individual, predefined and group addresses

17.4.4.4.1 General

The individual, group and predefined DL-addresses supported are:

a) the selector of all objects in the referenced address regime

b) the selector(s) of a user defined group or functional group of objects in the referenced address regime

c) the selector of one (individual) object in the referenced address regime

d) the selector for all links in the extended link

NOTE The groupings supported are those appropriate to the Configuration process.

17.4.4.4.2 DLE-addresses and node VIDs

17.4.4.4.2.1 General

The format of all DLE-addresses is USINT8

17.4.4.4.2.2 Individual object selectors

The range of Individual DLE-addresses per link is 2 to 255.

The value of an individual DLE's Node-VID is the value of its DLE-address, as shown in Figure 119.

17.4.4.4.2.3 All object selector

The all object selector for all DLEs on the local link is 0.

17.4.4.4.2.4 User defined group selectors

The User defined group object selectors for groups of DLEs on the local link are 8 to max-group-DLE, inclusive.

The value of max-group-DLE is a conformance-class issue. Its range is 8 to 15.

17.4.4.4.3 DLSAP-selectors

17.4.4.4.3.1 General

The format of all DLSAP-selectors is USINT8

17.4.4.4.3.2 Individual object selectors

The range of Individual DLE DLSAP-selectors per DLE is 2 to 255, inclusive. The maximum number and semantics of these individual addresses is a conformance-class issue.

17.4.4.4.3.3 All object selector

The all object selector for all DLSAPs in the (local) DLE is 0.

17.4.4.4.4 DLCEP-selectors

17.4.4.4.4.1 General

The format of all DLCEP-selectors is USINT8

Predefined ranges of DLCEP-selectors apply to channels and triggers.

17.4.4.4.4.2 Individual channel selectors

The range of Individual DLCEP-selectors per DLSAP that is assigned to channels in a DLE is 8 to 255. The maximum number and semantics of these individual selectors is a conformance-class issue.

17.4.4.4.4.3 Individual trigger selectors

The range of Individual DLCEP-selectors per DLSAP that is assigned to triggers in a DLSAP is 2 to 7, inclusive. The maximum number and semantics of these trigger selectors is a conformance-class issue. Table 133 shows Predefined Trigger assignments.

Table 133 – Predefined trigger assignments

Value	Trigger Assignments
2	the Global I/O Trigger
3 to 7	Individual I/O Triggers

17.4.4.4.4.4 All object selector

The all object selector for all DLCEPs in one (local) DLSAP is 0.

17.4.4.4.5 Link address

17.4.4.4.5.1 General

The length of the Link-address depends on the depth of the topology of links in the extended link. Table 134 shows:

a) the length and specific form of Link-address vs. depth of the real Nest of links in the extended -link.
b) the range of allowed values of Net-ID, the component of Link-address that identifies a specific, individual link within the extended -link
c) the range of allowed values of the selector for individual links at this level, and
d) the selector for all links on the extended link.

Table 134 – Format of link-addresses

Depth of real nest of links	Length of link-address OCTET STRING	Specific form of concatenated Link-address	Real topology of extended-link	Range of allowed values of Net-ID	Selector for all links on the extended link	Selector for individual links at this level
0	1	[z = 0]	local link only	0, 1	0	1
1	1	[z =2 to 255]		2 to 255	0	2 to 255
2	2	[y =2 to 255] [z =2 to 255]		2 to 255	0	2 to 255
3	3	[x =2 to 255] [y =2 to 255] [z =2 to 255]	maximum depth of nest of real links	2 to 255	0	2 to 255

The actual depth of the real nest of links on the extended link is 0 to max-Net-level.

The value of max-Net-level is a conformance-class issue. Its range is 0 to 3.

17.4.4.4.5.2 Link-address selectors

The format of all Link-address-selectors is an OCTETSTRING of the length shown in Table 134, column 2

17.4.4.4.5.3 Individual link selectors

The range of Individual DLE Link-address-selectors per level of the extended-link is 2 to 255, except when referencing the local-link. The maximum number and semantics of these individual selectors is a conformance-class issue.

The selector for the local link is 1

17.4.4.4.5.4 All object selector

The all object selector for all links on the extended-link is all 0s.

17.5 Real-time services

17.5.1 General

These services are Real-time Scheduling, Time stamping of Events, Triggers and Common Time. The DLMS provides the system manager with a means to select and configure the internal scheduling of the distributed DLS-provider. This configuration controls when each DLE will recognize its opportunities for communication and triggering activities.

17.5.2 Common sense of bus time

All nodes on a local link or an extended link maintain a common sense of Bus-time (local DL-Time-offset), in uS, which can be related to System-time by the Conductor. The accuracy of the common time sense across all nodes is a Bus configuration issue.

NOTE The source and external synchronization of System-time is a user choice. It might be UTC or the current time in the local time zone.

The primitives and parameters to support the DL-Time-offset function are as shown in Table 135.

Table 135 – DL-Time-offset primitive and parameters

DL-TIME-OFFSET Parameter name	Request output
local-offset	M

17.5.2.1 DL Time-class

The values of the DL-Time-Class attribute relate primarily to Bit_Rate and the number of Bridges traversed in an extended link, as shown in Table 136.

Table 136 – DL-Time-Classes

Data-rate	Number of bridges traversed	DL-Time-Class
≥ 1,25 Mbit/s	0	1 us
≥ 5 Mbit/s	3	1 us
≥ 0,625 Mbit/s	3	10 us
≥ 0,0785 Mbit/s	3	100 us

17.5.3 Time-stamping of events.

A node may support time-stamping of Events. Its DLE inserts its local DL-time-offset, when notified by the DLS-user, into a time-stamp-buffer selected by time-stamp-ID,. This buffer may be appended to the associated Event message by the DLS-user before that message is submitted to the DLE for transmission. Thus the time stamp appended to the message may be approximately concurrent with the time the relevant Event occurred, even if the Event message is sent much later.

NOTE The DLS-user may notify the DLE to record DL-time and NcycleLO via a zero latency mechanism.

Each DLSAP may have 0 to max-time-stamp non-retentive time-stamp-buffers. Max-time-stamp is a conformance-class issue and may range from 0 to 1 inclusive.

Time-stamping eliminates the extra latency that DLS-user activity would introduce in recording time-stamps.

Inter-DLE errors in the time-stamp are a function of inter-DLE errors in Bus-Time.

The primitives and parameters to support the DL-Time-stamp retrieval function are shown in Table 137.

Table 137 – DL-Time-Stamp retrieval primitives and parameters

DL-TIME-STAMP Parameter name	Request input	Confirm output
time-stamp-ID	C	
local-DL-Time-offset		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

The local-DL-Time-offset, e.g. the Time-stamp, of an Event can be related to System-time by the Conductor. The primitives and parameters to support the DL-Event-time function are shown in Table 138.

Table 138– DL-Event-Time primitive and parameters

DL-EVENT-Time Parameter name	Request output
Conductor's-System-time	M

17.5.4 Scheduling

The system manager establishes the schedule for all real-time SCAN or ExSCAN class data transfers and for all triggers in the Node-List.

This list is downloaded to each DLE during the Bus-configuration phase of Bus Startup. After the Conductor starts the bus in the OPERATIONAL mode, these scheduled activities are enforced by each DLE.

This inherent QoS is set by the Bus-configuration process to EXPLICIT. There are no related primitive exchanges between the DLS-provider and DLS-user.

17.5.5 Triggers

17.5.5.1 External trigger service

DLEs derive trigger signals, which may trigger a variety of internal or external actions, from their own Bus-Cycle counter and Slot-clock.

Since triggers do not consume bus transmission capacity, any number of actions may be triggered simultaneously without affecting concurrent data transfers.

Trigger signals may be issued a configured number of Slot-Times ahead of the slot that scans an I/O channel. This allows for latency in the external circuitry or apparatus.

This inherent QoS is set by the Bus-configuration process. There are no related primitive exchanges between the DLS-provider and DLS-user.

17.5.5.2 I/O trigger service

Each device DLE may generate one or more IoTrigger signals. The IoTrigger definition complies with the IEEE 1451.2 definition.

As specified in IEEE 1451.2, an IoTrigger signal can be applied directly to a device port or by the NCAP to either:

— all the I/O channels in the device (TEDS value is0); or

— a specific I/O channel defined by the device TEDS.

NOTE Routing of trigger signals is not a DLE responsibility.

The number of IoTrigger signals per device is not specified.

This inherent QoS is set by the Bus-configuration process. There are no related primitive exchanges between the DLS-provider and DLS-user, although there may be hardware interactions between these entities.

18 Type 7: Data Link services and concepts

18.1 Field of application, object

18.1.1 Field of application

This standard applies to a Data Link Layer appropriate for the exchange of data between transmitters, actuators, and programmable controllers within a manufacturing process.

18.1.2 Object

This standard specifies the DLL services. The object is to define:

a) the services provided at the conceptual interface between the Data Link and the DLS-users, and

b) the role of the Bus Arbitrator.

The standard is based on services provided by the Physical layer (IEC 61158-2) to the conceptual interface between the Physical and Data Link layers. (See Figure 1)

18.2 General description of services

18.2.1 General

Two types of data transmission services are provided:

a) the first handles connection-oriented buffer transfers between pre-established point-to-multipoint DLCs on the same local link;

b) the second handles acknowledged or unacknowledged connectionless message transfers between single DLSAPs, or unacknowledged message transfers from a single DLSAP to a group of DLSAPs on the extended link.

NOTE The standard term for data exchanged between DLS-users is DLS-user-data, or DLSDU [ISO/IEC 7498-1]. For purposes of clarity, the expressions "buffer transfer" and "message transfer" are used to distinguish between the two types of communications services, connection-oriented and connectionless, respectively, that are offered by this DLS,

There are also two types of buffer transfer services:

1)-**Cyclical buffer transfer**. Variable names and periods are defined when the system is configured, and are based on application needs. Cyclical exchanges are automatically triggered by the communications system without the user requesting them,

2) **Explicit request for buffer transfer**. Upon user request the value(s) of one or more variables are circulated.

The message transfer service also has two forms:

3) **Cyclical messages transfer**. Resources and periods are defined when the system is configured and are based on application needs. Cyclical transfers are automatically triggered by the communications system without the user requesting them,

4) **Aperiodic message transfer**. Upon user request one or more messages are circulated.

18.2.2 Addressing

The DL-addressing model for a system includes two different types of addressing: one for buffer transfer services and the other for message transfer services.

<u>For buffer transfers:</u> each variable in the system is associated with a DLCEP-identifier that characterizes it within the system in a unique manner.

Entities participating in a buffer transfer are not identified explicitly. Rather, they are identified indirectly as subscriber(s) or publisher of the identified variable.

Each variable has only one publisher.

For message transfer: one or more DLSAP-addresses are defined within each DLE. These DLSAP-addresses give access to a message transfer service.

Each DLSAP-address identifies an access point to a message service linked to a DLS-user entity.

Variable addressing is restricted to the local link. The addressing mechanism makes it possible to identify variables and exchanges independent of the producing and consuming DLEs. For buffer transfers all relationships between the various DLS-users are known and defined when the system is configured. Each DLCEP-identifier characterizes a single system variable and thus establishes a relationship between the unique publisher of the variable and the subscriber(s) of the variable.

Buffer transfers use the local broadcast medium and are restricted to the local link: the DLCEP-identifier and the value of a variable are made available to all DLEs on the local link. The DLCEP-identifier associated with the variable allows each DLE to recognize whether or not it is the publisher or a subscriber of the value associated with the identified variable.

Message transfers use the local broadcast medium, and bridges to traverse the extended link. During the message transaction two DLSAP-addresses are indicated in order to establish contact between the communicating entities:

a) A 24-bit destination DL(SAP)-address that encodes the link-id of the destination local link and the sub-address of the destination DLSAP or group of DLSAPs within that local link.
b) A 24-bit source DLSAP-address that encodes the link-id of the source DLE’s local link and the sub-address of the source DLSAP within that local link.

Each DLSAP-address specifies a DLS-user of the message service (for both emission and reception). This DL-address is unique within the extended link.

18.2.3 Flow control

Dynamic flow control for the exchange of variables is unnecessary. The volume of data exchanged as a result of cyclical traffic is constant, and is defined upon configuration of the system in a manner compatible with local link capacity.

Subscribers store only the last value received; a new exchange overrides the previous value.

An acknowledgement mechanism makes it possible to control message transfer flows. In addition, sequence numbering of messages avoids message duplication. A subscriber accepts a message only if that subscriber can store the message. In no case can a message overwrite a previously received message.

18.2.4 Detection of DLPDU duplication/loss

Detection mechanisms apply to errors resulting from communications problems or out-of-service DLEs.

DLPDU loss is accounted for in the finite state machines that describe the DL-protocol.

Duplication of a DLPDU can only occur with message transfers. The sequence numbering mechanism makes it possible to detect message duplication and avoid delivery of duplicate messages.

18.2.5 Overall description of medium allocation

18.2.5.1 General

An element known as the **Bus Arbitrator (BA)** controls the right of each data publisher to access the medium. It does this by emitting a DLPDU containing a link-local DL-identifier — either a DLSAP-address or a DLCEP-address. At any given instant there should be only one active Bus Arbitrator on each local link.

Each transaction belongs to one of the three medium allocation classes defined below:

a) Cyclical buffer transfers, message transfers or service request polling,

b) Explicit request for buffer transfer,

c) Explicit request for message transfer.

18.2.5.2 Cyclical buffer transfers, message transfers or service request polling

18.2.5.2.1 General

The Bus Arbitrator initiates transactions in a configured order. When one transaction has been completed the Bus Arbitrator begins the following transaction according to guidelines defined when the system is configured.

The procedure for each type of transaction is as follows:

18.2.5.2.2 Buffer transfer

For a buffer transfer, a basic transaction consists of the following phases:

a) The Bus Arbitrator broadcasts a variable DLCEP-identifier DLPDU.

b) The sole publisher of the information required then broadcasts a variable response DLPDU. During this phase subscribers take the information from the local link. Figure 125 shows the various phases of a buffer transfer transaction.

NOTE The term "publisher" designates the sole DLE connected to the local link that is configured as having the responsibility of emitting the variable associated with the bus-arbitrator-emitted DLCEP-identifier DLPDU immediately preceding on the local link. The term “subscriber” refers to any DLE which is configured to receive copies of a published variable and make those copies available to an associated DLS-user.

During a buffer transfer the publisher can, using specific features of the response DLPDU, transmit to the BA an explicit request for additional buffer transfers or message transfers.

18.2.5.2.3 Message transfer

For a message transfer, a basic transaction consists of the following phases:

a) The Bus Arbitrator broadcasts a message DL-identifier DLPDU.

b) The addressed DLE sends a message DLPDU

c) If the message DLPDU is addressed to a single DLSAP and requests an acknowledgement, the DLE associated with that DLSAP-address sends an acknowledgement DLPDU.

Steps b) and c) may be repeated a limited number of times if an expected acknowledgment DLPDU is not received error-free.

d) The originally-addressed DLE concludes the message exchange sequence by transmitting an end-of-transaction DLPDU to the Bus Arbitrator.

18.2.5.2.4 Service request polling

For a service request poll, a basic transaction consists of the following phases:

a) The Bus Arbitrator broadcasts a request DL-identifier DLPDU.
b) The initiator of the request replies with a request response DLPDU.
c) At a subsequent time of the Bus Arbitrator's choosing, one or more requested transactions identical in form to the cyclical buffer transfer transaction follow.

18.2.5.3 Explicit request for buffer transfer

The Bus Arbitrator services an explicit request for buffer transfer according to guidelines defined when the system is configured. The procedure employed is that of 18.2.5.2.2.

18.2.5.4 Explicit request for message transfer

The Bus Arbitrator services an explicit request for message transfer according to guidelines defined when the system is configured. The procedure employed is that of 18.2.5.2.3.

18.2.5.5 Bus arbitrator basic principles

The time interval separating the reception or emission of the end of one DLPDU and the emission or reception of the following DLPDU is known as the station's turnaround time, whether the station's function be that of a publisher or subscriber, message originator or receiver/acknowledger, or Bus Arbitrator.

A more detailed definition of turnaround time, and the impact of turnaround time on DL-timers, is given in the corresponding portion of IEC 61158-4.

The role of the Bus Arbitrator is to "give the floor" to each variable publisher or message originator, taking into account the services required according to the three medium allocation classes just defined.

The Bus Arbitrator thus has three types of functions:

a) Periodic triggering of buffer transfers, message transfers and request polling;
b) Triggered scanning of buffer transfers;
c) Triggered initiation of message transfers.

In addition, the Bus Arbitrator can provide

d) a synchronization function in order to guarantee the constant length of scanning cycles.

Each of these four functions is provided in a specific window: a periodic window, an aperiodic variable window, an aperiodic message window, and a synchronization window, respectively. These four windows constitute a basic scanning cycle.

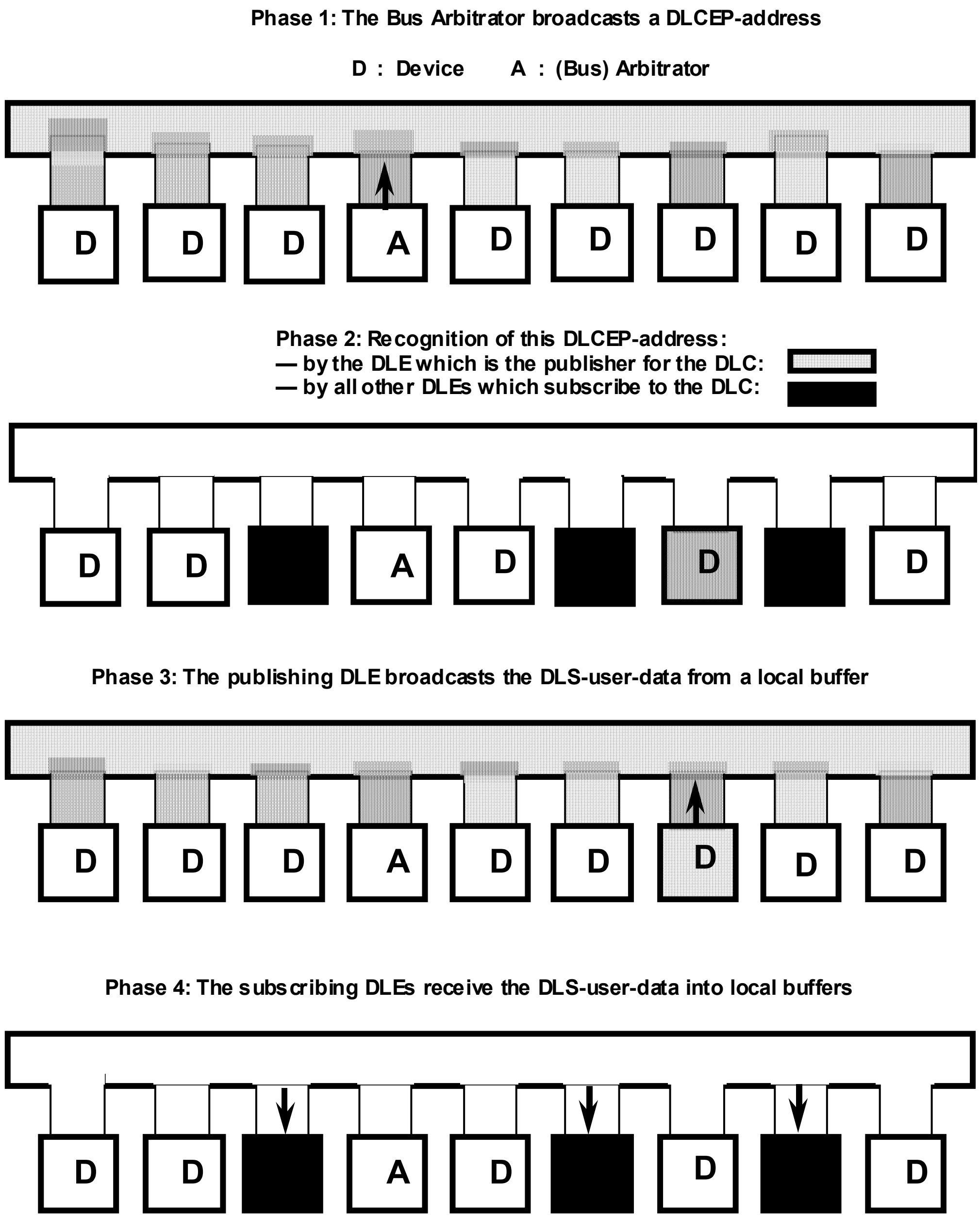


Figure 125 – General description of medium allocation

The medium access technique, shown in Figure 125, has the following characteristics:

1) Broadcasting of identified variables;

2)-Maximal efficiency in cyclical buffer transfers;

3 System Management can set parameters for medium sharing when the system is configured;

4) Guaranteed access time for cyclical buffer transfers, under all circumstances and regardless of the number of requests for triggered buffer transfers and message transfers;

5) Possibility of triggering a transaction in accordance with a global clock, that is, a clock that indicates the same time for all stations.

In addition, the medium access technique makes it possible to:

6) Give cyclical exchanges highest priority;

7) Respect the scanning period associated with each variable;

8) Give different priorities to triggered messages transfers and buffer transfers. These transactions are triggered in adjustable windows: the lengths of the "aperiodic variable" and "aperiodic message" windows are defined in terms of maximum limits set when the system is configured;
9) Change the effective priority of aperiodic transactions by inserting them in the periodic window.

18.2.6 Use of DL-identifiers

A DLSAP-address-identifier is used by the DLS-user and the DLS-provider to communicate a DLSAP-address. This information can take the form of a DLSAP-address whose naming domain is the extended link, or, when the DLSAP is local to the DLE, an identifier of local scope which identifies that DLSAP-address to both the DLE and the DLS-user.

A DL(SAP)-address-identifier is used by the DLS-user and the DLS-provider to communicate a DL(SAP)-address. This information can take the form of a DLSAP-address or group DL-address whose naming domain is the extended link, or, when the DLSAP is local to the DLE, an identifier of local scope which identifies that DLSAP-address to both the DLE and the DLS-user.

A DLCEP-identifier is used by the DLS-user and the DLS-provider to communicate the identity of a DLCEP. This information can take the form of a DLCEP-address whose naming domain is the local link, or, when the DLCEP is local to the DLE, an identifier of local scope which identifies that DLCEP to both the DLE and the DLS-user.

A DL-identifier is used by the DLS-user and the DLS-provider to communicate the identity of a DL-request for service. This information can take the form of a DL-address whose naming domain is the local link, or, when local to the DLE, an identifier of local scope which identifies that DL-request to both the DLE and the DLS-user.

NOTE Such DL-identifiers are used primarily in support of cyclical polling by the Bus Arbitrator in support of the DL-SPEC-UPDATE explicit request for buffer transfer service (18.7). (See also 18.8 and 18.9.)

18.3 Sequences of primitives

18.3.1 Constraints on services and primitives

There is no specific order in the execution of the different services.

A request primitive is used by the DLS-user to request a service. A confirmation primitive is returned to the DLS-user at the completion of the service. An indication primitive is used to report to the DLS-user the receipt of new DLS-User Data or the receipt of a new message.

18.3.2 Primitives on buffer transfers

The DL-services and their parameters are summarized in Table 139

Table 139 – Summary of DL-services and primitives for buffer transfers

Service	Primitive	Parameter
Update Buffer	DL-PUT request	(in DLCEP-identifier, DLS-user-data)
	DL-PUT confirm	(*out* Status)
Copy Buffer	DL-GET request	(*in* DLCEP-identifier)
	DL-GET confirm	(*out* DLS-user data, Status)
Buffer transfer	DL-BUFFER-SENT indication	(*out* DLCEP-identifier)
	DL-BUFFER-RECEIVED indication	(*out* DLCEP-identifier)
Explicit request for buffer transfer	DL-SPEC-UPDATE request	(*in* Specified DL-identifier, List of DL-identifiers requested*)*
	DL-SPEC-UPDATE confirm	(*out* Status)
	DL-FREE-UPDATE request	(*in* List of DL-identifiers requested, Priority)
	DL-FREE-UPDATE confirm	(*out* Status)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

18.3.3 Primitives on message exchanges

The DL-services and their parameters are summarized in Table 140

Table 140 – Summary of DL-services and primitives for message exchanges

Service	Primitive	Parameter
Unacknowledged message transfer	DL-MESSAGE request	(*in* Specified DL-identifier, Destination DL(SAP)-Address, Source DLSAP-Address, DLS-user-data)
	DL-MESSAGE indication	(*out* Destination DL(SAP)-Address, Source DLSAP-Address DLS-user-data)
	DL-MESSAGE confirm	(*out* Status)
Acknowledged message transfer	DL-MESSAGE-ACK request	(*in* Specified DL-identifier, Destination DLSAP-Address, Source DLSAP-Address, DLS-user-data)
	DL-MESSAGE-ACK indication	(*out* Destination DLSAP-Address, Source DLSAP-Address, DLS-user-data)
	DL-MESSAGE-ACK confirm	(*out* Status)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

18.4 Buffer writing

18.4.1 Function

This service allows the DLS-user to transfer data to the local DLE for later use in buffer transfers where the DLE is the publisher. Associated primitives are DL-PUT request and DL-PUT confirm.

18.4.2 Sequence of primitives

The sequence of primitives in a successful buffer writing is shown in Figure 126:

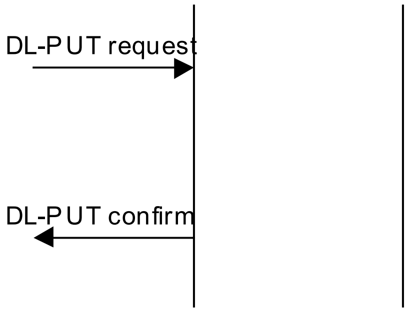


Figure 126 – Primitives associated with the buffer writing service

18.4.3 Types of primitives and parameters

18.4.3.1 General

Table 141 indicates the types of primitives and parameters of writing a buffer.

Table 141 – DL-Put primitives and parameters

DL-PUT Parameter name	Request input	Confirm output
DLCEP-identifier	M	
DLS-user-data	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

18.4.3.2 Request primitive

DL-PUT request allows the DLS-user to transfer the value of a variable (DLS-user-data) to the DLE for the DLE's use in subsequent buffer transfers at the specified DLCEP.

18.4.3.3 DLCEP-identifier

This parameter unambiguously designates the variable within the local link. This identifier corresponds to a variable published by the DLE. It can take the form of a local identifier or of a link-local DLCEP-address.

18.4.3.4 DLS-user-data

This parameter replaces the value previously stored in the buffer associated with the corresponding DLCEP-identifier. The maximum amount of data which can be stored in a buffer is 128 octets.

NOTE One expected Application Service Entity, MPS (IEC 61158-5 Type 7), uses DLS-user-data that is always two octets or more in length.

18.4.3.5 Confirm primitive

A DL-PUT confirm primitive follows a DL-PUT request primitive and provides an account on the progress of the action requested.

18.4.3.6 Status

This parameter reports on the writing operation. Possible values of this parameter are:

a) Success — the writing operation has been accomplished properly
b) Failure — semantic error in the request (unknown DLCEP-identifier, amount of DLS-user-data exceeds the supported buffer size of 128 octets),
c) Failure — invalid DLCEP-identifier (a DLCEP-identifier can in fact be invalidated by System Management),
d) Failure — problem with access to buffer associated with the variable (buffer availability)

This last status value indicates that the DLE is concurrently accessing the buffer and that the implementation does not support that concurrency.

18.5 Buffer reading

18.5.1 Function

This service allows DLS-user data to be transferred from the DLE to the DLS-user. Associated primitives are DL-GET request and DL-GET confirm.

18.5.2 Sequence of primitives

The sequence of primitives in a successful buffer reading is shown in Figure 127:

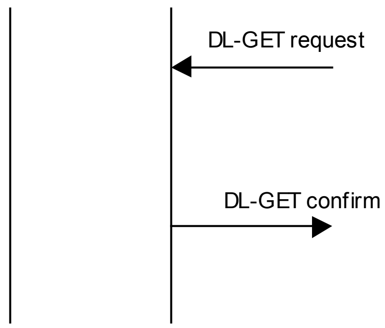


Figure 127 – Primitives associated with the buffer reading service

18.5.3 Types of primitives and parameters

18.5.3.1 General

Table 142 indicates the type of primitives and parameters of reading a buffer:

Table 142 – DL-Get primitives and parameters

DL-GET Parameter name	Request input	Confirm output
DLCEP-identifier	M	
Status		M
DLS-user-data		C
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

18.5.3.2 Request primitive

DL-GET request allows the DLS-user to read the value of a variable received through the DLL.

Use of the primitive does not erase the stored value, which can be reread by another similar DL-GET request primitive.

18.5.3.3 DLCEP-identifier

This parameter unambiguously designates the variable within the local link. This identifier corresponds to a variable subscribed by the DLE. It can take the form of a local identifier or of a link-local DLCEP-address.

18.5.3.4 Confirm primitive

A DL-GET confirm primitive follows a DL-GET request primitive and provides an account on the progress of the requested action.

18.5.3.5 Status

This parameter reports on the reading of the variable's value. Possible values of this parameter are:

a) Success — the reading operation has been accomplished properly
b) Failure — unknown DLCEP-identifier
c) Failure — invalid DLCEP-identifier (a DLCEP-identifier can in fact be invalidated by System Management),
d) Failure — problem with access to buffer associated with the variable (buffer availability)

This last status value indicates that the DLE is concurrently accessing the buffer and that the implementation does not support that concurrency.

18.5.3.6 DLS-user-data

This parameter, which is meaningful when the Status parameter returns a value of Success, provides the last value previously stored in the buffer associated with the corresponding DLCEP-identifier. The maximum amount of data which can be stored in a buffer is 128 octets.

18.6 Buffer transfer

18.6.1 Function

This service notifies the DLS-user of each time that a published variable is sent or received. Associated primitives are DL-BUFFER-SENT indication and DL-BUFFER-RECEIVED indication.

18.6.2 Sequence of primitives

The sequence of primitives in a successful buffer transfer is shown in Figure 128:

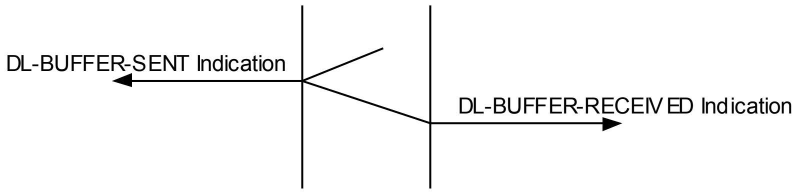


Figure 128 – Primitives associated with the buffer transfer service

18.6.3 Types of primitives and parameters

18.6.3.1 General

Table 143 indicates the type of primitives and parameters for a buffer sent indication

Table 143 – DL-Buffer-Sent primitive and parameter

DL-BUFFER-SENT Parameter name	Indication input
DLCEP-identifier	M

18.6.3.2 Buffer sent indication

A DL-BUFFER-SENT indication informs the DLS-user that the published variable associated with the specified DLCEP-identifier has just been emitted on the bus.

NOTE The buffer associated with that DLCEP-identifier may have been written previously by the DLS-user using the DL-PUT request primitive.

18.6.3.3 DLCEP-identifier

This parameter unambiguously designates the variable that has just been emitted on the bus. It can take the form of a local identifier or of a link-local DLCEP-address.

Table 144 indicates the type of primitives and parameters for a buffer received indication.

Table 144 – DL-Buffer-Received primitive and parameter

DL-BUFFER-RECEIVED Parameter name	Indication input
DLCEP-identifier	M

18.6.3.4 Buffer received indication

A DL-BUFFER-RECEIVED indication informs the DLS-user that a subscribed identified variable has just been correctly received. The value of the variable is thus available in the B_Dat_Cons buffer associated with the variable and may be read using the DL-GET request primitive.

18.6.3.5 DLCEP-identifier

This parameter unambiguously designates the variable that has just been received by the subscribing DLE. It can take the form of a local identifier or of a link-local DLCEP-address.

The DLS-user can access the current value of the variable by reading the buffer associated with that DLCEP-identifier (DL-GET).

18.7 Explicit request for buffer transfer

18.7.1 Function

This service makes it possible for an entity to explicitly request the broadcasting of one or more link-local DLCEP-addresses. Since each link-local DLCEP-address is associated with a variable, the service triggers the exchange of these variables.

Two types of service are offered:

a) The explicit request for a buffer transfer is linked to a DLCEP-identifier specified when the service is requested. This service is known as an **specified explicit request**.

The initiator of this request is a DLS-user that may or may not be a publisher or subscriber of the variable(s) requested. The request is fulfilled during the Bus Arbitrator's periodic or aperiodic scanning cycle, according to configuration.

Associated primitives are DL-SPEC-UPDATE request and DL-SPEC-UPDATE confirm.

b) The explicit request for a buffer transfer is not linked to a DL-identifier when the service is requested. This service is known as a **free explicit request**.

The initiator of this request is a DLS-user that may or may not be a publisher or subscriber of the variable(s) requested. The request is fulfilled during the Bus Arbitrator's aperiodic scanning cycle.

Associated primitives are DL-FREE-UPDATE request and DL-FREE-UPDATE confirm.

NOTE A single DLCEP-identifier cannot be configured concurrently for both a specified explicit request and a free explicit request.

With the free explicit request service, two levels of priority (urgent and normal) are possible.

With the specified explicit request service only urgent priority is offered.

When the Bus Arbitrator receives an (specified or free) explicit request for a buffer transfer it initiates a transaction in conformance with the buffer transfer service described in 18.2.5.2.2.

NOTE Two types of service are provided for explicit requests for buffer transfer because of the possible uses of this service.

The specified explicit request service incorporates a mechanism for overriding previous requests. A buffer for potential requests is attached to each DLCEP-identifier reserved for this service upon configuration. These buffers contain the lists of DL-identifiers whose broadcasting has been explicitly requested.

A request associated with a given specified DL-identifier thus overrides any previous request using that DL-identifier.

A specified explicit request associated with a given DLCEP-identifier can be filled during the Bus Arbitrator's periodic or aperiodic scanning cycle. A means exists for choosing between the two types of scanning cycle, by indicating whether or not the request needs to pass through the Bus Arbitrator.

Thus the specified explicit request service makes it possible to fill an explicit request for a buffer transfer during the Bus Arbitrator's periodic scanning cycle. In this case the recovery service provided by the DLS-user can be used. This service is defined in the document.

The free explicit request service has a mechanism for placing requests in a queue. Two queues for requests are provided: one for urgent requests, and one for other requests.

A DLCEP-identifier's attachment to one of these latter queues is dynamic.

The DLCEP-identifier that carries the request is chosen by the DLE.

NOTE The DLCEP-identifier chosen is the DLCEP-identifier of the first DLCEP-identifier DLPDU emitted on the medium after the service has been requested which was not otherwise configured in support of the specified explicit request service.

The free explicit request service makes it possible to rapidly include a buffer transfer request in the Bus Arbitrator's periodic scanning, since the request can be carried by the first response given by the requesting DLE for a published DLCEP-identifier.

In the protocol a single request response DLPDU is emitted even if several requests are stored in the two queues, but there are as many confirmation primitives as there are requests. Additional restrictions on the implementation of this service are that

1) a request response DLPDU is limited to a maximum of 64 DL-identifiers, and
2) all of the DL-identifiers specified in a single free explicit request shall be conveyed to the Bus Arbitrator in the same request response DLPDU; the list cannot be split across two such DLPDUs.

18.7.2 Specified explicit request sequence of primitives

The sequence of primitives in a successful specified explicit update is shown in Figure 129:

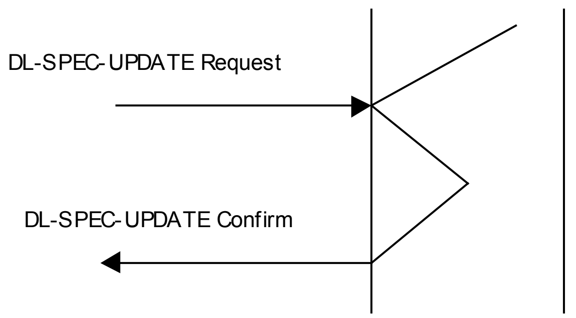


Figure 129 – Primitives associated with the specified explicit request for a buffer transfer

18.7.3 Specified explicit request primitives and parameters

18.7.3.1 General

Table 145 indicates the types of primitives or parameters needed for a specified explicit update

Table 145 – DL-Spec-Update primitives and parameters

DL-SPEC-UPDATE Parameter name	Request input	Confirm output
Specified DLCEP-identifier	M	
List Of Requested DL-identifiers	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

18.7.3.2 Request primitive

A DL-SPEC-UPDATE request allows the DLS-user to request the circulation of one or more identified variables, while specifying the DLCEP that will be used to transmit the request to the Bus Arbitrator. All previous requests using this same DLCEP are thus overridden.

18.7.3.3 Specified DLCEP-identifier

This parameter states which DLCEP-identifier will carry the buffer transfer request. A request from any DLS-user is initiated upon emission of the identified variable response corresponding to the selected DLCEP-identifier. This DLCEP-identifier should belong to the periodic portion of the Bus Arbitrator's scanning table and should be configured for the DL-SPEC-UPDATE service and not the DL-FREE-UPDATE service.

If the DLCEP-identifier is properly configured, the request associated with the specified DLCEP-identifier is fulfilled as part of the Bus Arbitrator's periodic scanning.

18.7.3.4 List of requested DL-identifiers

This parameter contains the list of DL-identifiers associated with the variables that the requester wants broadcast. This list is destined for the Bus Arbitrator; it contains a maximum of 64 DL-identifiers.

18.7.3.5 Confirm primitive

A DL-SPEC-UPDATE confirm follows a DL-SPEC-UPDATE request and provides the DLS-user with the status of the requested exchange.

18.7.3.6 Status

This parameter indicates that the request to broadcast one or more DL-identifiers has been taken into account by the Bus Arbitrator, and that the list of DL-identifiers has been transmitted to the Bus Arbitrator. This primitive does not guarantee that the Bus Arbitrator has indeed received the list, nor does it indicate that the associated variables requested have actually been exchanged.

This parameter has the following values:

a) Success — the request was taken into account by the Bus Arbitrator
b) Failure — the specified DLCEP-identifier is unknown or not configured for this service

   NOTE The confirmation is immediate in this case.

c) Failure — the request was overridden by a new request which has been made on the same specified DLCEP-identifier
d) Failure — problem with access to buffer associated with the variable (buffer availability)

This last status value indicates that the DLE is concurrently accessing the buffer and that the implementation does not support that concurrency.

18.7.4 Free explicit update sequence of primitives

The sequence of primitives in a successful free explicit update is shown in Figure 130:

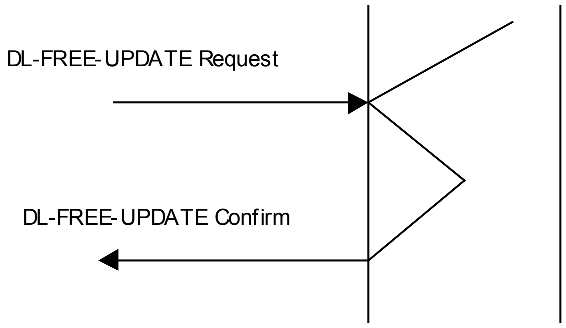


Figure 130 – Primitives associated with the free explicit request for a buffer transfer

18.7.5 Free explicit update primitives and parameters

18.7.5.1 General

Table 146 indicates the types of primitives or parameters needed for an free explicit update.

Table 146 – DL-Free-Update primitives and parameters

DL-FREE-UPDATE **Parameter name**	**Request** input	**Confirm** output
List Of Requested DL-identifiers	M	
Priority	M	
Status		M

18.7.5.2 Request primitive

A DL-FREE-UPDATE request allows the DLS-user to request the circulation of one or more identified variables. The request is carried to the Bus Arbitrator by the first buffer transfer response DLPDU emitted by the initiating entity that fills the following conditions:

— the source DLCEP-address associated with that response DLPDU is not configured for the DL-SPEC-UPDATE service, and

— that DLCEP-address is part of the periodic portion of the Bus Arbitrator's scanning table.

18.7.5.3 List of requested DL-identifiers

This parameter contains the list of DL-identifiers associated with the variables that the requester wants broadcast. This list is destined for the Bus Arbitrator; it contains a maximum of 64 DL-identifiers.

18.7.5.4 Priority

This parameter indicates to the Bus Arbitrator whether the request is to be processed in urgent or normal mode.

NOTE To avoid differentiating the two priorities, the remainder of this document will indicate priority using "i", that is , i=1 for urgent or high priority, or i=2 for normal priority.

18.7.5.5 Confirm primitive

A DL-FREE-UPDATE confirm follows a DL-FREE-UPDATE request and provides the DLS-user with the status of the requested exchange.

18.7.5.6 Status

This parameter indicates that the request to broadcast one or more DL-identifiers has been taken into account by the Bus Arbitrator, and that the list of DL-identifiers has been transmitted to the Bus Arbitrator. This primitive does not guarantee that the Bus Arbitrator has indeed received the list, nor does it indicate that the associated variables requested have actually been exchanged.

This parameter has the following values:

a) Success — the request was taken into account by the Bus Arbitrator

b) Failure — the request queue was full.

NOTE The confirmation is immediate in this case.

18.8 Unacknowledged message transfer

18.8.1 Function

The DLL provides the DLS-user with a connectionless unacknowledged message transfer service. For aperiodic message transfers, this service uses an instance of the buffer transfer service from the source DLE to request a service opportunity from the Bus Arbitrator. For cyclical message transfers the Bus Arbitrator circulates message DL-identifier DLPDUs in the periodic window.

Unacknowledged message services are either point-to-point or multipoint within the extended link. When services are multipoint, the destination address is a group DL-address recognized by zero or more DLEs. If the unacknowledged message service transaction involves more than one local link, the message needs to pass sequentially across multiple links between the originating and intended destination DLEs. The resulting message propagation path forms an acyclic subgraph of the extended link. Bridge (DL-relay) DLEs connected between successive links of this subgraph transfer the message from the link on which it is received to the one or more links on which it needs to be sent to reach the intended destination DLEs.

For the unacknowledged message transfer service, the initiator of the request is the source of the message. For cyclical message transfer, the message DL-identifier is periodically circulated by the Bus Arbitrator independent of any requests made or not made by the source DLE.

18.8.2 Sequence of primitives

The sequence of primitives in a successful unacknowledged message transfer is shown in Figure 131:

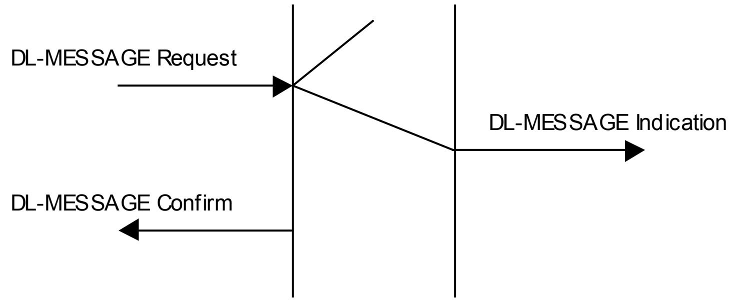


Figure 131 – Primitives associated with the unacknowledged message transfer request service

18.8.3 Types of primitives and parameters

18.8.3.1 General

Table 147 indicates the types of primitives or parameters for an unacknowledged message transfer:

Table 147 – DL-Message primitives and parameters

DL-MESSAGE	Request	Indication	Confirm
Parameter name	input	input	output
Specified DL-identifier	M		
Destination DL(SAP)-address	M	M (=)	
Source DLSAP-address	M	M (=)	
DLS-user-data	M	M (=)	
Status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

18.8.3.2 Request primitive

A DL-MESSAGE request allows a DLS-user to request the transmission of an unacknowledged message to the specified DL(SAP)-address.

18.8.3.3 Specified DL-identifier

If the message transfer is cyclical, this parameter is a DL-identifier configured for cyclical message service. This DL-identifier is linked to a queue for messages to be emitted.

If the message transfer is aperiodic, this parameter takes on the value NIL. When the service is requested, this parameter value makes it possible to refer the request to the queue for aperiodic message transfers.

18.8.3.4 Destination DL(SAP)-address

This parameter specifies a DLSAP-address or group DL-address, identifying the DLSAP or group of DLSAPs which is the destination of the message.

18.8.3.5 Source DLSAP-address

This parameter specifies the local DLSAP, associated with the DLS-user, which is to be the attributed source of the message.

18.8.3.6 DLS-user-data

This parameter specifies the information which is being conveyed between the corresponding DLS-users.

18.8.3.7 Indication primitive

A DL-MESSAGE indication signals the arrival of an unacknowledged message to a DLS-user associated with the destination DL(SAP)-address.

NOTE This parameter does not indicate whether the sender’s mode of transfer was aperiodic or cyclical.

18.8.3.8 Confirm primitive

A DL-MESSAGE confirm provides the initiating DLS-user with a report on the transmission of an unacknowledged message.

18.8.3.9 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 — the message was sent,
b) Failure — the queue for messages to be emitted is filled (confirmation is immediate in this case),
c) Failure — the specified DL-identifier is unknown or not configured for this service (confirmation is immediate in this case).

18.9 Acknowledged message transfer

18.9.1 Function

The DLL provides the DLS-user with a connectionless acknowledged message transfer service. For aperiodic message transfers, this service uses an instance of the buffer transfer service from the source DLE to request a service opportunity from the Bus Arbitrator. For cyclical message transfers, the Bus Arbitrator circulates message DL-identifier DLPDUs in the periodic window.

Acknowledged message transfers are point-to-point within the extended link. If the acknowledged message service transaction involves more than one local link, the message needs to pass sequentially across multiple links between the originating and intended destination DLEs. On each such link, the acknowledgement is furnished by the receiving DLE on that link, which is either a forwarding bridge (DL-relay) DLE or the intended destination DLE, to the DLE which transmitted the message on that link, which is either the original sending DLE or a forwarding bridge DLE. This acknowledgement only acknowledges the proper transmission and reception on that local link.

NOTE IEC 61158-3 does not further specify the role of the bridge DLE. Requirements for bridges are specified in IEC 61158-4.

For the acknowledged message transfer service, the initiator of the request is the source of the message. For the cyclical message transfer service, the message DL-identifier is periodically circulated by the Bus Arbitrator independent of any requests made or not made by the source DLE.

18.9.2 Sequence of primitives

The sequence of primitives in a successful acknowledged message transfer is shown in Figure 132:

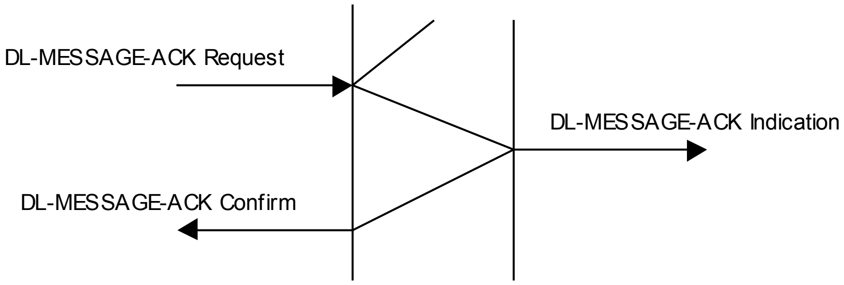


Figure 132 – Primitives associated with the acknowledged message transfer request service

18.9.3 Types of primitives and parameters

18.9.3.1 General

Table 148 indicates the types of primitives or parameters needed for an acknowledged message transfer.

Table 148 – DL-Message-Ack primitives and parameters

DL-MESSAGE-ACK Parameter name	Request input	Indication input	Confirm output
Specified DL-identifier	M		
Destination DLSAP-address	M	M (=)	
Source DLSAP-address	M	M (=)	
DLS-user-data	M	M (=)	
Status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

18.9.3.2 Request primitive

A DL-MESSAGE-ACK request allows a DLS-user to request the transmission of an acknowledged message to the specified DLSAP-address.

18.9.3.3 Specified DL-identifier

If the message transfer is cyclical, this parameter is a DL-identifier configured for cyclical message service. This DL-identifier is linked to a queue for messages to be emitted.

If the message transfer is aperiodic, this parameter takes on the value NIL. When the service is requested, this parameter value makes it possible to refer the request to the queue for aperiodic message transfers.

18.9.3.4 Destination DLSAP-address

This parameter specifies the DLSAP which is the destination of the message.

18.9.3.5 Source DLSAP-address

This parameter specifies the local DLSAP, associated with the DLS-user, which is to be the attributed source of the message.

18.9.3.6 DLS-user-data

This parameter specifies the information which is being conveyed between the corresponding DLS-users.

18.9.3.7 Indication primitive

A DL-MESSAGE-ACK indication signals the arrival of an acknowledged message to the DLS-user associated with the destination DLSAP-address.

NOTE This parameter does not indicate whether the sender's mode of transfer was aperiodic or cyclical.

18.9.3.8 Confirm primitive

A DL-MESSAGE-ACK confirm provides the initiating DLS-user with a report on the transmission and initial reception of an acknowledged message.

18.9.3.9 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 — message positively acknowledged, either by the addressed DLE or by an intermediate bridge which will forward the message.

b) Failure — message negatively acknowledged, when the destination DLE's queue for received messages is filled,

c) Failure — the queue for messages to be emitted is filled.

NOTE The confirmation is immediate in this case.

d) Failure — the specified DL-identifier is unknown or not configured for this service

NOTE The confirmation is immediate in this case.

e) Failure — expiration of the acknowledgement DLPDU reception timer or faulty transmission.

19 Type 8: Data Link Service and concepts

19.1 Overview

This type provides a connection-oriented subset of services, specified in ISO/IEC 8886, on pre-established DLCs. It also uses a subset of the models for buffers, queues, DLCs, DL(SAP)s and DLCEPs described in Clause 6 through Clause 9.

The DLS provides the sending or receiving DLS-user with either a FIFO queue or a retentive buffer, where each queue item or buffer can hold a single DLSDU.

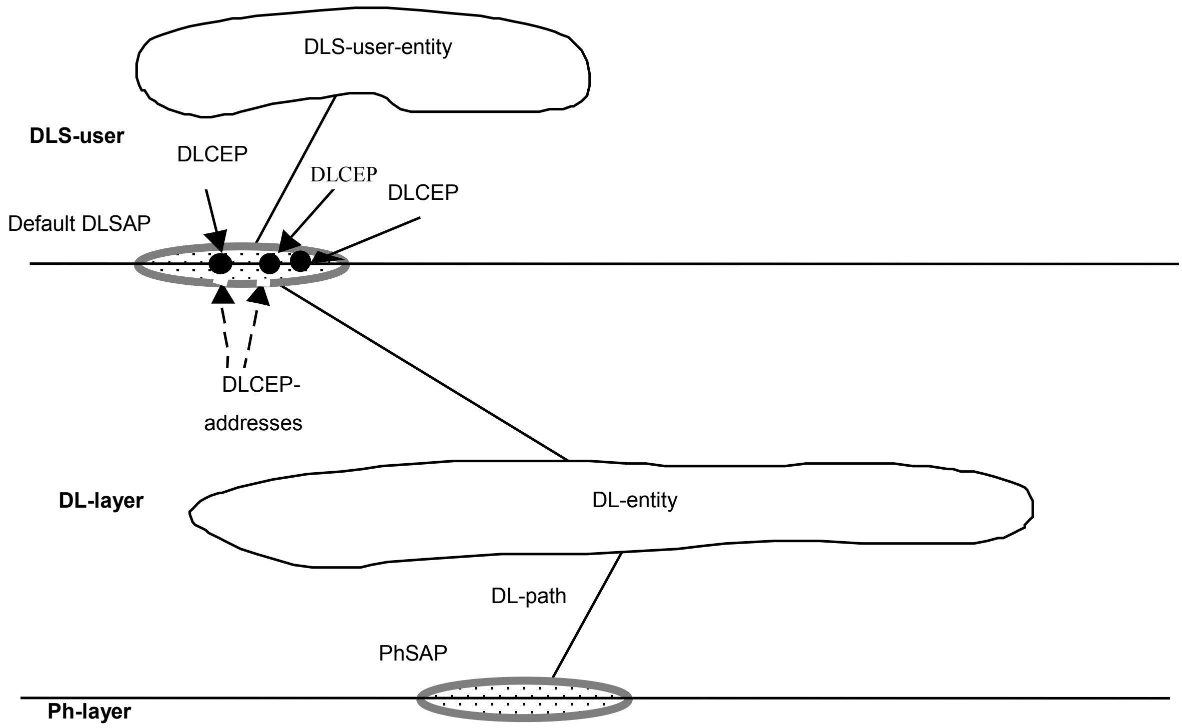
DL-names, known conventionally as DL-addresses, are identifiers from a defined identifier space — the DL-address-space — which serve to name objects within the scope of the data link layer. The objects that need to be named within the DLL are data-link-connection-end-points (DLCEPs).

The DL-address-space from which DL-addresses are drawn may be partitioned into sub-spaces of DL-addresses due to the class of the device in which the DLS-entity resides, and the class of the addressed DLCEP.

Only two DLSAPs are supported by a DLE: a single default DLSAP for sending and receiving data, and a single default management DLSAP to invoke local DL-management services. As these DLSAPs are accessed locally, they have no DLSAP DL-address assigned to them. The DLSAP used is determined implicitly by the type of service primitive selected.

A DLS-user may need to distinguish among several DLCEPs at the same DLSAP for sending and receiving data; thus a local DLCEP-identification mechanism is also provided. All primitives issued at such a DLSAP within the context of a DLC use this mechanism to identify the local DLCEP. The naming-domain of this DLCEP-identification is the DL-local-view.

The relationship between DLSAPs, DLCEPs and DLCEP DL-addresses used for data transfer services is shown in Figure 133.



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 Only one DLS-user-entity can be associated with any given DL-entity.

NOTE 4 Only one default DLSAP is supported.

NOTE 5 Only connection oriented DL-services are supported. All DLCs are pre-configured and pre-established. No DLSAP addresses are assigned.

Figure 133 – Relationships of DLCEPs and DLCEP-addresses to default DLSAP

The DLS provides three classes of DLCEPs:

a) **PEER** — the DLS-user can exchange DLSDUs with one other peer DLS-user;

b) **PUBLISHER** — the DLS-user can send DLSDUs to a set of zero or more associated subscribing DLS-users;

c) **SUBSCRIBER** — the DLS-user can receive DLSDUs from the associated publishing DLS-user.

NOTE The DLCEP Classes PUBLISHER and SUBSCRIBER only support one conveyance path from the publisher DLCEP to each subscriber DLCEP. No conveyance path from a subscriber DLCEP to the publisher DLCEP exists.

All buffers and queues are pre-created and bound to DLCEPs. The DLS-user cannot directly create, delete, bind or unbind buffers or queues.

DLCEPs of class PEER always use queues; DLCEPs of classes PUBLISHER and SUBSCRIBER always use buffers which are bound to them.

DLCEPs of class PEER are used only for confirmed data transfer; DLCEPs of classes PUBLISHER and SUBSCRIBER are used only for unconfirmed data transfers.

All DLCs are pre-defined and pre-established by local DL-management before any DLS-user is granted access to the DLS.

All information used during creation of buffers and queues and establishing of DLCs is stored by local DL-management. The means by which a DLS-user can obtain this information from local DL-management is a local issue, beyond the scope of this standard.

A buffer is referenced by a Buffer DL-identifier assigned by local DL-management during creation. As each buffer or queue is associated with (bound to) a single DLCEP, a DLCEP DL-identifier or DLCEP DL-address (if assigned to the DLCEP) can also be used to reference the buffer or queue bound to this DLCEP. Local DL-management can provide the DLS-user with the facility to inter-convert the reference types.

19.2 Sequence of primitives

19.2.1 Constraints on sequence of primitives

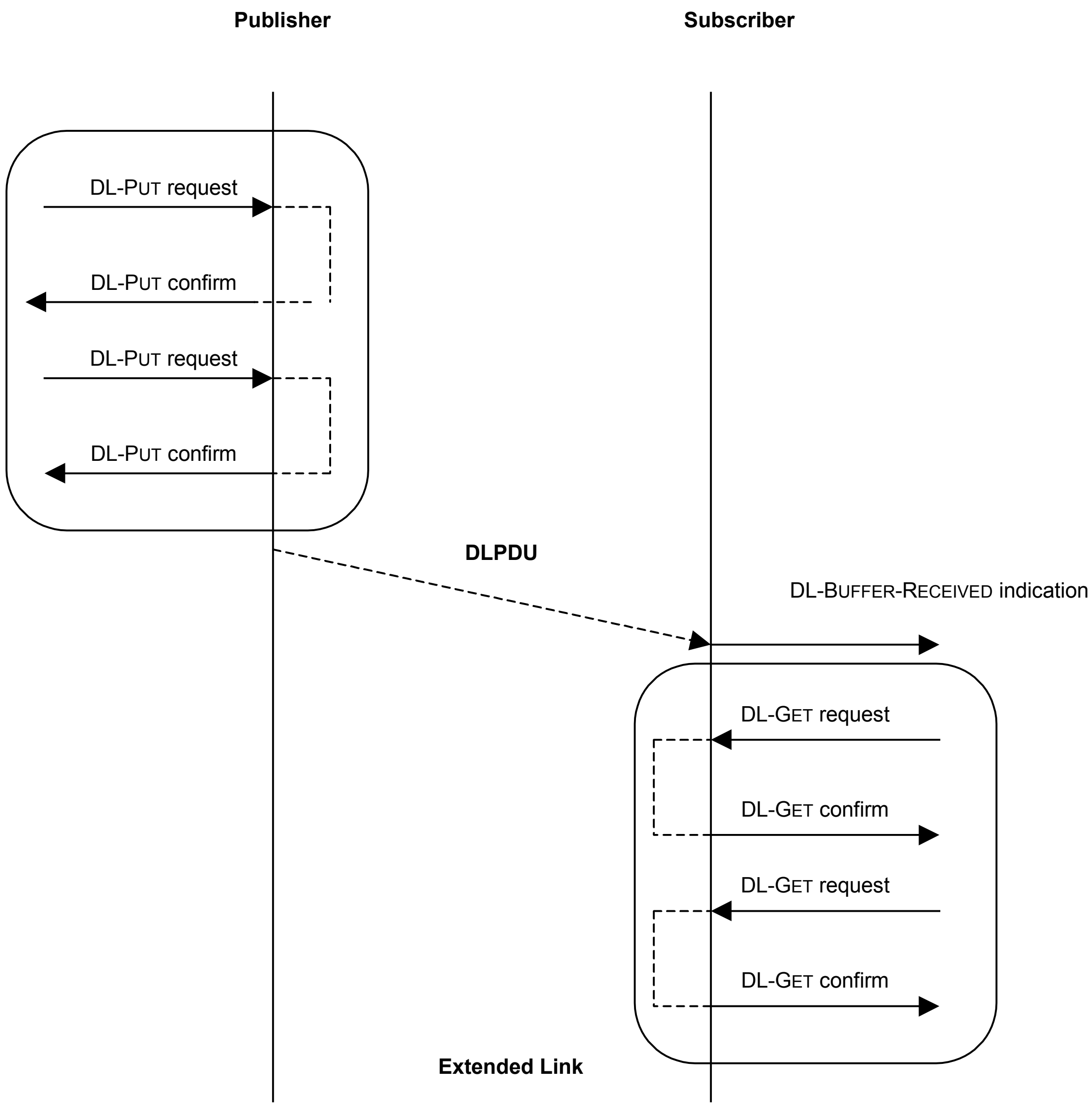
This subclause defines the constraints on the sequence in which the primitives defined in 19.3 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur.

In order to request a service, the DLS-user uses a request primitive. A confirmation primitive is returned to the DLS-user after the service has been completed. The arrival of a service request is indicated to the remote DLS-user by means of an indication primitive. The connection-mode primitives and their parameters are summarized in Table 149. The major relationships among the primitives at two DLC end-points are shown in Figure 134 through Figure 136.

Table 149 – Summary of DL-connection-mode primitives and parameters

Service	Primitive	Parameter
Put buffer	DL-PUT request	(*in* Buffer DL-identifier DLS-user-data)
	DL-PUT confirm	(*out* Status)
Get buffer	DL-GET request	(*in* Buffer DL-identifier)
	DL-GET confirm	(*out* Status, DLS-user-data)
Buffer received	DL-BUFFER-RECEIVED indication	(*out* Status)
Normal data transfer	DL-DATA request	(*in* DLCEP DL-identifier, DLS-user-data)
	DL-DATA indication	(*out* DLCEP DL—identifier, DLS-user-data)
	DL-DATA confirm	(*out* Status)
NOTE The method by which a DL-DATA confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

The sequence of primitives of a successful normal data transfer is defined in the time-sequence diagrams in Figure 135. The sequence of primitives in a failed normal data transfer is defined in the time-sequence diagram in Figure 136.



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

NOTE 2 The request primitives within the outlined areas are locally confirmed.

Figure 134 – Sequence of primitives for the buffer data transfer

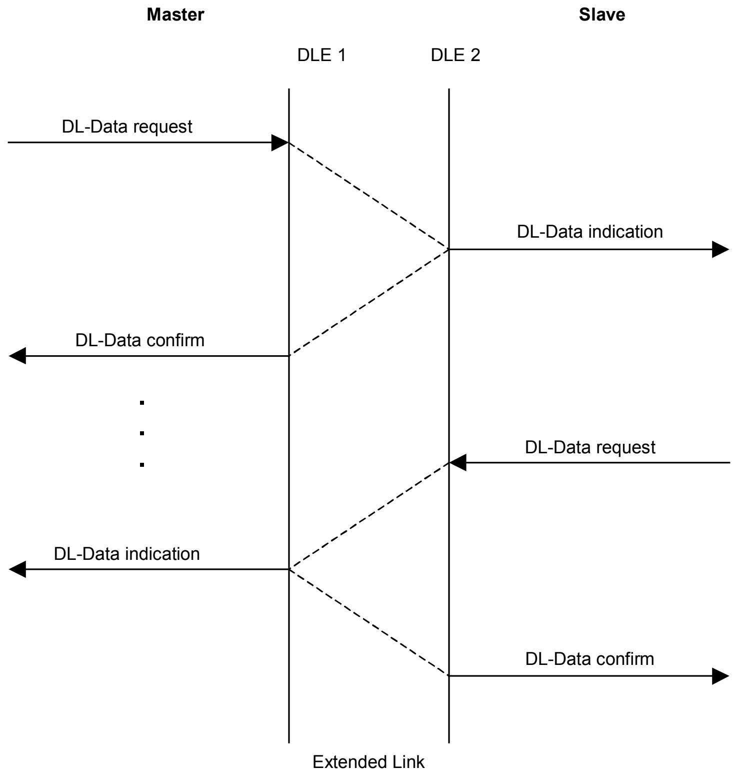


Figure 135 – Normal data transfer service between a master and a slave

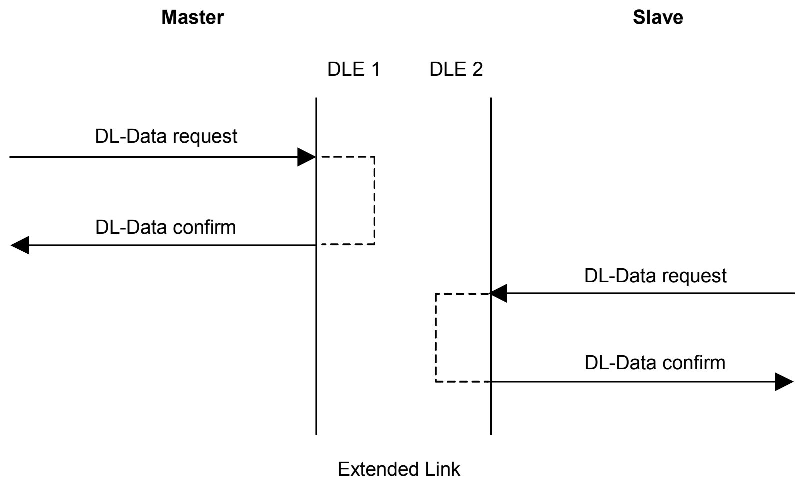


Figure 136 – Sequence of primitives for a failed normal data transfer

19.3 Connection-mode Data Link services

19.3.1 Put buffer

19.3.1.1 Function

The DLS-user uses this service to write directly to the specified buffer. The service is locally processed after the DL-PUT request primitive has arrived. The DLE communicates the successful processing of the service to the DLS-user by means of a DL-PUT confirmation primitive (immediate confirmation).

19.3.1.2 Types of parameters

Table 150 indicates the primitive and parameters of the Put Buffer DLS.

Table 150 – Put buffer primitive and parameters

DL-PUT Parameter name	Request input	Confirm output
Buffer DL-identifier	M	
DLS-user-data	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

19.3.1.2.1 Buffer DL-identifier

This parameter specifies the local DL-identifier of the buffer as assigned by local DL-management.

19.3.1.2.2 DLS-user-data

This parameter contains the user-data which is to be written to the buffer.

19.3.1.2.3 Status

This parameter indicates the success or failure of the preceding request. The value conveyed in this parameter is as follows:

a) “OK — success — service completed”

b) “IV — failure — invalid parameters in the request”

19.3.2 Get buffer

19.3.2.1 Function

The DLS-user uses this service to directly read the specified buffer. The service is locally processed after the DL-GET request primitive has arrived. The DLE communicates the successful processing of the service to the DL-user by means of a DL-GET confirmation primitive (immediate confirmation).

19.3.2.2 Types of primitives and parameters

19.3.2.2.1 General

Table 151 indicates the primitives and parameters of the Get Buffer DLS.

Table 151 – Get buffer primitive and parameters

DL-GET Parameter name	Request input	Confirm output
Buffer DL-identifier	M	
DLS-user-data		C
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

19.3.2.2.2 Buffer DL-identifier

This parameter specifies the local DL-identifier of the buffer as assigned by local DL-management.

19.3.2.2.3 DLS-user-data

This parameter is present when the preceding request primitive was successfully executed. This parameter contains the data which was read from the buffer.

19.3.2.2.4 Status

This parameter indicates the success or failure of the preceding request. The value conveyed in this parameter is as specified in 19.3.1.2.3.

19.3.3 Buffer received

19.3.3.1 Function

The DLS-provider uses this service to inform the DLS-user about the successful update of all buffers.

NOTE If the DLS-provider detects an error, no DL-BUFFER-RECEIVED indication is generated.

19.3.3.2 Types of primitives and parameters

19.3.3.2.1 General

Table 152 indicates the primitive and parameters of the Buffer Received DLS.

Table 152 – Buffer received primitive and parameters

DL-BUFFER-RECEIVED Parameter name	Indication output
Status	M

19.3.3.2.2 Status

This parameter indicates the successful update of all buffers. The value conveyed in this parameter is as follows:

NOTE 1 In this protocol all buffers on all DLCs are updated at the same time.

a) “OK — success — buffer received without errors”

NOTE 2 If the DLE detects an error no DL-BUFFER-RECEIVED indication is generated.

19.3.4 Normal data transfer

19.3.4.1 Function

This service allows a DL-user (called local user) to send DLS-user-data to a single remote DLE. If the remote DLE receives the data correctly, the data is reported to its DLS-user. The requesting DLS-user receives a confirmation indicating the receipt or non-receipt of the user data at the remote DLE.

19.3.4.2 Types of primitives and parameters

19.3.4.2.1 General

Table 153 indicates the types of primitives and the parameters needed for normal data transfer.

Table 153 – Normal data transfer primitive and parameters

DL-DATA Parameter name	Request input	Indication output	Confirm output
DLCEP DL-identifier	M	M	
DLS-user-data	M	M(=)	
Status			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.			

19.3.4.2.2 DLCEP DL-identifier

This parameter specifies the DLCEP for which the data transfer service is to occur. The identifier was assigned by local DL-management when the DLC was pre-established.

19.3.4.2.3 DLS-user-data

This parameter contains the user data to be transmitted (request) or which was received (indication).

19.3.4.2.4 Status

The link status parameter indicates the success or failure of the preceding request. The value conveyed in this parameter is as follows:

a) "OK — success — service completed"

b) "RR — failure — resources of the remote DLCEP not available or insufficient"

c) "LR — failure — resources of the local DLCEP not available or insufficient"

d) "NA — failure — no response, or not a plausible response (acknowledge response), of the remote device"

e) "DS — failure — DLE not synchronized at the moment"

f) "IV — failure — invalid parameter in the request"

20 Type 8: DL-management Service

20.1 Scope

This clause defines the administrative DL-management services (DLMS) which are available to the DLMS user, together with their service primitives and the associated parameters. (see Figure 1) All DL-management services use the default management DLSAP.

20.2 Facilities of the DL-management service

The service interface between the DLMS-user and the DLE makes the following functions available:

a) reset of the local DLE;

b) request and change the current operating parameters of the local DLE;

c) indication of unexpected events, errors, and status changes, either local or remote;

d) read-out of the active DL-subnetwork configuration;

e) read-out of the current DL-subnetwork configuration;

f) setting of a DL-subnetwork configuration.

Together these facilities constitute the DLMS.

20.3 Overview of services

20.3.1 General

DL-management provides the following services to all DLMS-users:

a) reset

b) event

DL-management may provide the following services to DLMS-users:

c) set value

d) get value

DL-management provides the following additional services to DLMS-users of DLEs that are functioning as a DLL master:

e) get current configuration

f) get active configuration

g) set active configuration

20.3.2 Reset

The DLMS-user uses this required service to cause DL-management to reset the DLE. The DLMS-user receives a confirmation for this service.

20.3.3 Event

DL-management uses this required service to inform the DLMS-user of certain events or detected errors in the DLE.

20.3.4 Set value

The DLMS-user uses this service to set a new value to the variables of the DLE. It receives a confirmation on whether the specified variable(s) assumed the corresponding specified value(s).

20.3.5 Get value

The DLMS-user uses this service to read variables of the DLE.

20.3.6 Get current configuration (master only)

The DLMS-user of the master DLE uses this required service to read the current DL-subnetwork configuration.

20.3.7 Get active configuration (master only)

The DLMS-user of the master DLE uses this required service to read the active DL-subnetwork configuration.

20.3.8 Set active configuration (master only)

The DLMS-user of the master DLE uses this required service to set a DL-subnetwork configuration.

20.4 Overview of interactions

The DL-management primitives and their parameters are summarized in Table 154. The sequences of DL-management primitives are defined by the time-sequence diagrams in Figure 137 through Figure 143.

NOTE DL-management services which provide an immediate response use only a request primitive with output parameters; this provides correlation of the response with the request. Those services where the response can be much delayed and where the response information is independent of the number of, or time sequence of, the corresponding requests, use separate request and indication primitives.

Table 154– Summary of DL-management primitives and parameters

Service	Primitive	Parameter
Reset	DLM-RESET request	(<none>)
	DLM-RESET confirm	(out Status)
Event	DLM-EVENT indication	(out Event-identifier, Additional-information)
Set value	DLM-SET-VALUE request	(in Variable-name, Desired-value)
	DLM-SET-VALUE confirm	(out Status)
Get value	DLM-GET-VALUE request	(in Variable-name)
	DLM-GET-VALUE confirm	(out Status, Current-value)
Get current configuration	DLM-GET-CURRENT-CONFIGURATION request	(in Desired Configuration)
	DLM-GET-CURRENT-CONFIGURATION confirm	(out Status, Current Configuration)
Get active configuration	DLM-GET-ACTIVE-CONFIGURATION request	<none>
	DLM-GET-ACTIVE-CONFIGURATION confirm	(out Status, Active Configuration)
Set active configuration	DLM-SET-ACTIVE-CONFIGURATION request	(in Active Configuration)
	DLM-SET-ACTIVE-CONFIGURATION confirm	(out Status, Additional-information)

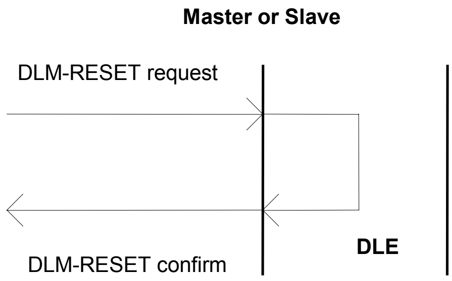


Figure 137 – Sequence of primitives for the reset service

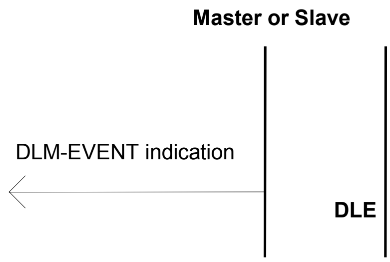


Figure 138 – Sequence of primitives for the event service

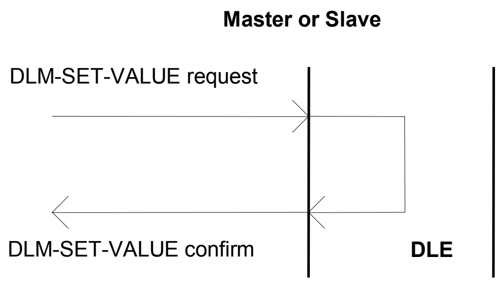


Figure 139 – Sequence of primitives for the set value service

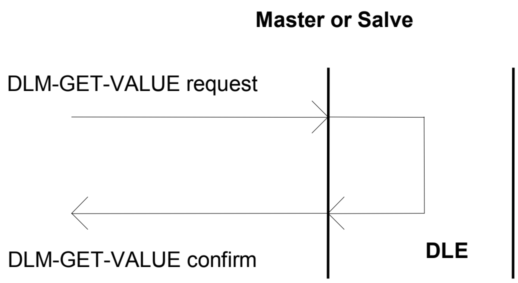


Figure 140 – Sequence of primitives for the get value service

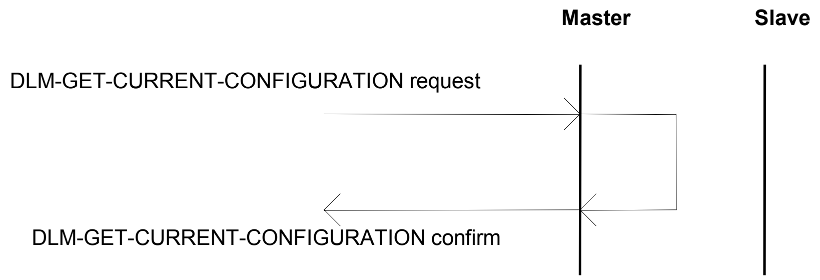


Figure 141 – Sequence of primitives for the get current configuration service

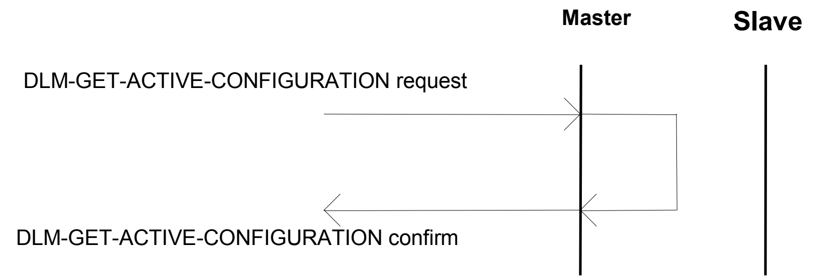


Figure 142 – Sequence of primitives for the get active configuration service

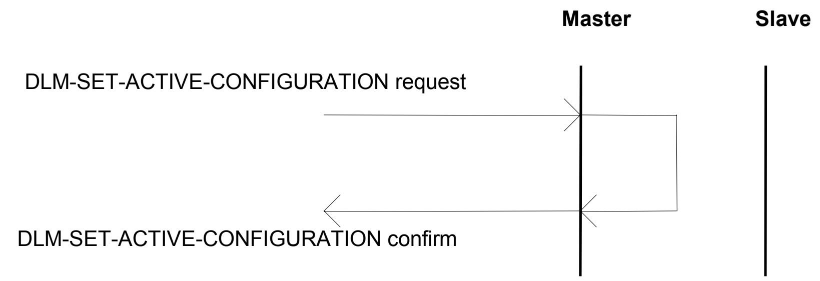


Figure 143 – Sequence of primitives for the set active configuration service

20.5 Detailed specification of services and interactions

20.5.1 Reset

20.5.1.1 Function

This required service is used to reset the DLE. The confirm primitive is issued immediately prior to the DLE reset action.

20.5.1.2 Types of parameters

20.5.1.2.1 General

Table 155 indicates the primitives and parameters of the DLM-RESET service.

Table 155 –Reset service primitives and parameters

DLM-RESET Parameter name	Request input	Confirm Output
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

20.5.1.2.2 Status

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

a) "OK — success; reset completed"; or

b) "NOK — failure".

20.5.2 Event

20.5.2.1 Function

This required service is used to report the occurrence of a DL-event which could be of significance to DL-management. After link-related errors have been reported, the DLMS-provider carries out a configuration check (determination of the current configuration by connection all outgoing interfaces). If the configuration differs from the configuration prior to the detection of the link-related error, the DLMS-provider automatically generates an event with information on the configuration change.

20.5.2.2 Types and parameters

20.5.2.2.1 General

Table 156 indicates the primitives and parameters of the DLM-EVENT service.

Table 156 – Event service primitive and parameters

DLM-Event Parameter name	Indication output
Event-identifier	M
Additional-information	C

20.5.2.2.2 Event identifier

This parameter specifies the event within the DLE whose occurrence is being announced. The possible values are defined in the corresponding part of IEC 61158-4.

20.5.2.2.3 Additional-information

This optional parameter provides event-specific additional information.

20.5.3 Set value

20.5.3.1 Function

This optional service can be used to set (write) the value of a DLE configuration parameter.

20.5.3.2 Types of parameters

20.5.3.2.1 General

Table 157 indicates the primitives and parameters of the DLM-SET-VALUE service

Table 157 – Set value service primitives and parameters

DLM-SET-VALUE Parameter name	Request input	Confirm Output
Variable-name	M	
Desired-value	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

20.5.3.2.2 Variable-name

This parameter specifies the variable within the DLE whose value is to be altered. The selectable variables are defined in the corresponding part of IEC 61158-4.

20.5.3.2.3 Desired-value

This parameter specifies the desired value for the selected variable. The permitted values or value ranges are defined in IEC 61158-4, Type 8.

20.5.3.2.4 Status

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

a) “OK — success — variable value updated”;
b) “NOK — failure — variable does not exist or could not assume the new value”; or
c) “IV — failure — invalid parameters in the request”.

20.5.4 Get Value

20.5.4.1 Function

This optional service can be used to get (read) the value of a DLE variable.

20.5.4.2 Types of parameters

20.5.4.2.1 General

Table 158 indicates the primitives and parameters of the DLM-GET-VALUE service.

Table 158 – Get value service primitives and parameters

DLM-GET-VALUE Parameter name	Request Input	Confirm Output
Variable-name	M	
Status		M
Current-value		C
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

20.5.4.2.2 Variable-name

This parameter specifies the variable within the DLE whose value is being requested. The selectable variables are defined in the corresponding part of IEC 61158-4.

This parameter receives the current value for the selected variable.

20.5.4.2.3 Status

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

a) “OK — success — the variable could be read”;

b) “NOK — failure — the variable does not exits or could not be read”; or

c) “IV — failure — invalid parameters in the request”.

20.5.4.2.4 Current-value

This parameter is present when the status parameter indicates that the requested service was performed successfully. This parameter specifies the current value of the selected variable.

20.5.5 Get current configuration

20.5.5.1 Function

This service is required for a master DLE; it is not available for a slave. The DLMS-user of the master DLE can use this service to get (read) the current DL-configuration of the extended link.

NOTE 1 The DLMS-provider is expected to use ID cycles to detect the currently connected slaves and transfer the detected configuration to the DLMS-user in the current configuration parameter.

NOTE 2 The DL-subnetwork configuration parameter specifies the configuration of the DL-subnetwork which is implied by successful completion of the service.

NOTE 3 The DLMS-provider employs a similar function at DLL start-up to open all outgoing interfaces of the slaves.

20.5.5.2 Types and parameters

20.5.5.2.1 General

Table 159 indicates the primitives and parameters of the Get Current Configuration DLMS.

Table 159 – Get current configuration service primitives and parameters

DLM-GET-CURRENT-CONFIGURATION Parameter name	Request Input	Confirm Output
Desired Configuration	M	
Status		M
Current configuration		C
Additional-information		C
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

20.5.5.2.2 Desired configuration

This parameter specifies the desired configuration of the extended link after the service has completed.

a) "CLOSED — the outgoing interfaces of all slaves are closed";

b) "OPEN — the outgoing interfaces of all slaves are open".

20.5.5.2.3 Status

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

a) "OK — success";

b) "NOK — failure — an error was detected when a DL-segment was connected"; or

c) "IV — failure — no ID cycles could be run (DL-subnetwork error)".

20.5.5.2.4 Current configuration

This compound parameter specifies the current configuration of the extended link. The parameter has the structure specified in Table 161.

20.5.5.2.5 Additional-information

This optional parameter provides status-specific additional information.

20.5.6 Get active configuration

20.5.6.1 Function

This service is required for a master DLE; it is not available for a slave. The DLMS-user of the master DLE can use this service to get (read) the active DL-configuration of the extended link.

NOTE This is a local service without implied activity on the local link.

20.5.6.2 Types and parameters

20.5.6.2.1 General

Table 160 indicates the primitives and parameters of the get active configuration service.

Table 160 –Get active configuration service primitives and parameters

DLM-GET-ACTIVE-CONFIGURATION Parameter name	Request input	Confirm Output
Status		M
Active configuration		C
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

20.5.6.2.2 Active configuration

This compound parameter specifies the active configuration of the DL-subnetwork. It takes the form of a conceptual list whose entries are ordered according to the physical order of the slaves in the ring. The parameter has the structure specified in Table 161.

Table 161 – The active configuration parameter

Item Number	Type	Description
1	device code	first device code of slave
2	DL-segment level	first DL-segment level of slave
3	device code	second device code of slave
4	DL-segment level	second DL-segment level of slave
...	...	...
2N-1	device code	Nth device code of slave
2N	DL-segment level	Nth DL-segment level of slave

20.5.6.2.3 Status

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

a) “OK — success”; or

b) “NOK — failure”.

20.5.7 Set active configuration

20.5.7.1 Function

This service is required for a master DLE; it is not available for a slave. The DLMS-user of the master can use this service to configure the extended link. If the new configuration cannot be accepted, the exact error cause is communicated to the DLMS-user and the old configuration is retained.

20.5.7.2 Types and parameters

20.5.7.2.1 General

Table 162 indicates the primitives and parameters of the set active configuration service.

Table 162 – Set active configuration service primitives and parameters

DLM-SET-ACTIVE-CONFIGURATION Parameter name	Request input	Confirm Output
Active configuration list	M	
Status		M
Additional-information		C
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

20.5.7.2.2 Active configuration list

This compound parameter specifies the new active configuration of the DL-subnetwork to be generated. The parameter has the structure specified in Table 161.

20.5.7.2.3 Status

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

a) “OK — success”;

b) “NOK — failure — an error was detected when a DL-segment was connected to the ring. The new configuration could not be generated”; or

c) “IV — failure — no ID cycles could be run; a fatal bus error”.

20.5.7.2.4 Additional-information

This optional parameter provides status-specific additional information.

BIBLIOGRAPHY

IEC 60847:1988, *Characteristics of local area networks (LAN)*

IEC 60955:1989, *Process data highway, type C (PROWAY C), for distributed process control systems*

IEC 61158, *Digital data communications for measurement and control — Fieldbus for use in industrial control systems*

IEC 61158-2, *Physical layer specification and service definition*

IEC 61158-5, *Application layer service definition*

IEC 61158-6, *Application layer protocol specification*

IEC 61784, *Digital data communications for measurement and control — Fieldbus for use in industrial control systems — Profile sets for continuous and discrete manufacturing*

ISO/IEC 8802-2:1998, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical link control*

ISO/IEC 8802-4:1990, *Information processing systems– Local area networks – Part 4: Token-passing bus access method and physical layer specifications*

ISO/IEC 15802-1:1995, *Information technology – Telecommunication and information exchange between systems – Local and metropolitan area networks – Common specifications – Part 1: Medium Access Control (MAC) service definition*

IEEE Std 1451.1:1999, *A Smart Transducer Interface for Sensors and Actuators – Network Capable Application Processor (NCAP) Information Model*

ITU-T V.41:1998, *Code-independent error-control system*

INDEX

IEC

Standards Survey

The IEC would like to offer you the best quality standards possible. To make sure that we continue to meet your needs, your feedback is essential. Would you please take a minute to answer the questions overleaf and fax them to us at +41 22 919 03 00 or mail them to the address below. Thank you!

Customer Service Centre (CSC)

International Electrotechnical Commission
3, rue de Varembé
1211 Genève 20
Switzerland

or

Fax to: **IEC**/CSC at +41 22 919 03 00

Thank you for your contribution to the standards-making process.

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

A Prioritaire

RÉPONSE PAYÉE

SUISSE

Customer Service Centre (CSC)
International Electrotechnical Commission
3, rue de Varembé
1211 GENEVA 20
Switzerland

Q1 Please report on **ONE STANDARD** and **ONE STANDARD ONLY**. Enter the exact number of the standard: *(e.g. 60601-1-1)*

....

Q2 Please tell us in what capacity(ies) you bought the standard *(tick all that apply).* I am the/a:

- purchasing agent ❑
- librarian ❑
- researcher ❑
- design engineer ❑
- safety engineer ❑
- testing engineer ❑
- marketing specialist ❑
- other...

Q3 I work for/in/as a: *(tick all that apply)*

- manufacturing ❑
- consultant ❑
- government ❑
- test/certification facility ❑
- public utility ❑
- education ❑
- military ❑
- other...

Q4 This standard will be used for: *(tick all that apply)*

- general reference ❑
- product research ❑
- product design/development ❑
- specifications ❑
- tenders ❑
- quality assessment ❑
- certification ❑
- technical documentation ❑
- thesis ❑
- manufacturing ❑
- other...

Q5 This standard meets my needs: *(tick one)*

- not at all ❑
- nearly ❑
- fairly well ❑
- exactly ❑

Q6 If you ticked NOT AT ALL in Question 5 the reason is: *(tick all that apply)*

- standard is out of date ❑
- standard is incomplete ❑
- standard is too academic ❑
- standard is too superficial ❑
- title is misleading ❑
- I made the wrong choice ❑
- other ....

Q7 Please assess the standard in the following categories, using the numbers:
(1) unacceptable,
(2) below average,
(3) average,
(4) above average,
(5) exceptional,
(6) not applicable

- timeliness ....
- quality of writing....................................
- technical contents...................................
- logic of arrangement of contents ....
- tables, charts, graphs, figures...............
- other ....

Q8 I read/use the: *(tick one)*

- French text only ❑
- English text only ❑
- both English and French texts ❑

Q9 Please share any comment on any aspect of the IEC that you would like us to know:

....
....
....
....
....
....
....
....
....
....
....
....

ISBN 2-8318-6971-4

9 782831 869711

ICS 25.040; 35.100; 35.240.50

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND