IEC 61158-3-7

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 3-7: Data-link layer service definition – Type 7 elements

IEC 61158-3-7:2007(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

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

About IEC publications

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

- Catalogue of IEC publications: www.iec.ch/searchpub

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

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

IEC 61158-3-7

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 3-7: Data-link layer service definition – Type 7 elements

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE **V**

ICS 35.100.20; 25.040.40

ISBN 2-8318-9413-1

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 3-7: Data-link layer service definition – Type 7 elements

FOREWORD

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

NOTE Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission of their respective intellectual-property-right holders.

International Standard IEC 61158-3-7 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This first edition and its companion parts of the IEC 61158-3 subseries cancel and replace IEC 61158-3:2003. This edition of this part constitutes an editorial revision.

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

a) deletion of the former Type 6 fieldbus, and the placeholder for a Type 5 fieldbus data-link layer, for lack of market relevance;
b) addition of new types of fieldbuses;
c) division of this part into multiple parts numbered 3-1, 3-2, …, 3-19.

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

FDIS	Report on voting
65C/473/FDIS	65C/484/RVD

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

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

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

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

NOTE The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

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

INTRODUCTION

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 3-7: Data-link layer service definition – Type 7 elements

1 Scope

1.1 Overview

This part of IEC 61158 provides common elements for basic time-critical messaging communications between devices in an automation environment. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the Type 7 fieldbus data-link layer in terms of

a) the primitive actions and events of the service;

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

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

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

- the Type 7 fieldbus application layer at the boundary between the application and data-link layers of the fieldbus reference model, and
- systems management at the boundary between the data-link layer and systems management of the fieldbus reference model.

1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual data-link layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of data-link protocols for time-critical communications. A secondary objective is to provide migration paths from previously-existing industrial communications protocols.

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

a) the sizes and octet ordering of various multi-octet service parameters, and

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

1.3 Conformance

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

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

2 Normative references

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

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

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

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 Reference model terms and definitions

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

3.1.1 DL-address [7498-3]

3.1.2 DL-connection [7498-1]

3.1.3 DL-connection-end-point [7498-1]

3.1.4 DL-connection-end-point-identifier [7498-1]

3.1.5 correspondent (N)-entities [7498-1]
correspondent DL-entities (N=2)
correspondent Ph-entities (N=1)

3.1.6 (N)-entity [7498-1]
DL-entity (N=2)
Ph-entity (N=1)

3.1.7 flow control [7498-1]

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

3.1.9 DL-name [7498-3]

3.1.10 peer-entities [7498-1]

3.1.11 primitive name [7498-3]

3.1.12 DL-protocol [7498-1]

3.1.13 DL-protocol-connection-identifier [7498-1]

3.1.14 DL-protocol-data-unit [7498-1]

3.1.15 DL-relay [7498-1]

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

3.1.17 (N)-service-access-point [7498-1]
DL-service-access-point (N=2)
Ph-service-access-point (N=1)

3.1.18 DL-service-data-unit [7498-1]

3.1.19 DLS-user-data [7498-1]

3.2 Service convention terms and definitions

This standard also makes use of the following terms defined in ISO/IEC 10731 as they apply to the data-link layer:

3.2.1 acceptor

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

3.2.3 deliver (primitive)

3.2.4 DL-service-primitive;
primitive

3.2.5 DL-service-provider

3.2.6 DL-service-user

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

3.2.8 multi-peer

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

3.2.10 requestor

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

3.2.12 submit (primitive)

3.3 Data-link service terms and definitions

3.3.1
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

3.3.2
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

3.3.3
basic transaction
succession of DLPDUs related to a single DL-service instance

3.3.4
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.

3.3.5
control field
portion of an emitted or received DLPDU that gives the nature of the data exchanged and the type of exchange

3.3.6
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

3.3.7
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

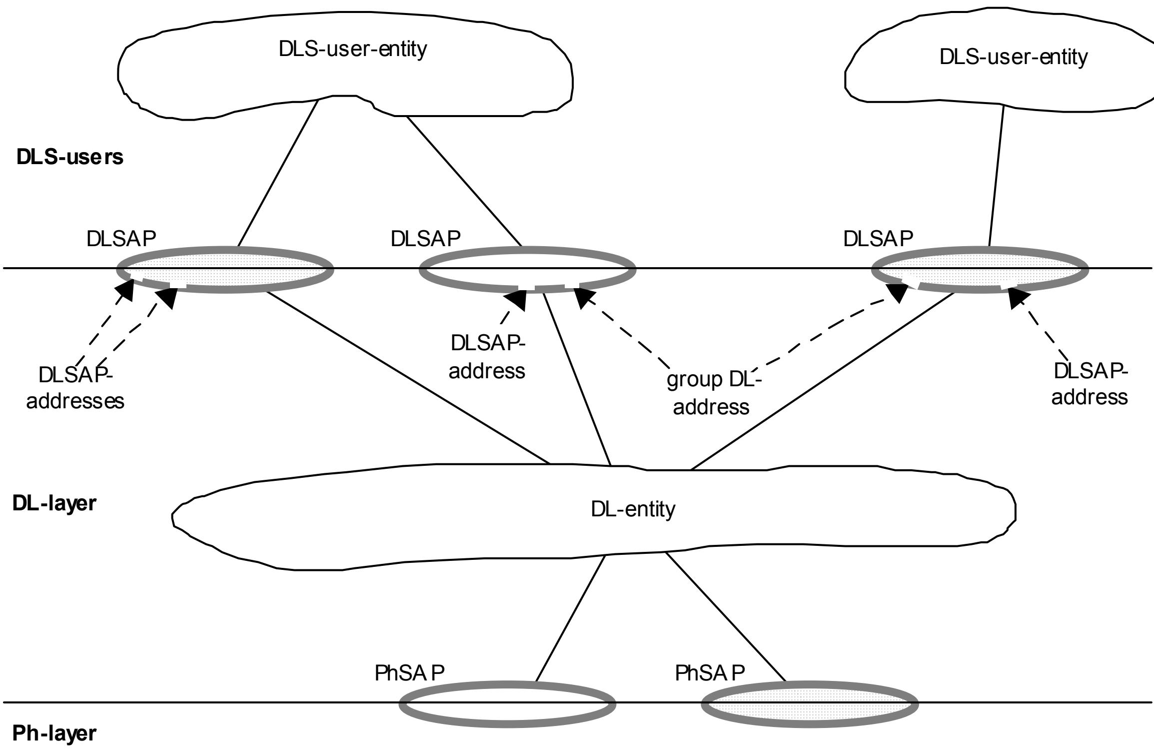
3.3.8
DL-segment, local link
set of devices that respect the DL-protocol and that are interconnected through a PhL. Only one bus-arbitrator is active on a single DL-segment

3.3.9
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

3.3.10
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

3.3.11
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.



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 1 – Relationships of DLSAPs, DLSAP-addresses and group DL-addresses

3.3.12
DL(SAP)-address
either an individual DLSAP-address, designating a single DLSAP of a single DLS-user, or a group DL-address potentially designating multiple DLSAPs, each of a single DLS-user

NOTE This terminology is chosen because ISO/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.

3.3.13
(individual) DLSAP-address
DL-address that designates only one DLSAP within the extended link

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

3.3.14
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

3.3.15
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.

3.3.16
frame
denigrated synonym for DLPDU

3.3.17
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

3.3.18
identified variable (or simply "variable")
DLL system variable for which an associated DLCEP-identifier has been defined

3.3.19
invalid DLCEP-identifier
DLCEP-identifier not recognized locally

3.3.20
macrocycle
set of basic cycles needed for all cyclical DLCEP-identifiers to be scanned

3.3.21
message DLPDU identifier
information that a bus-arbitrator emits to allocate the medium to a source DLE for a message transfer

3.3.22
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

3.3.23
node
single DL-entity as it appears on one local link

3.3.24
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.

3.3.25
published identified variable
variable that corresponds to a DLCEP-identifier for which the DLE emits data

3.3.26
receiving DLS-user
DL-service user that acts as a recipient of DLS-user-data

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

3.3.27
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

3.3.28
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

3.3.29
sending DLS-user
DL-service user that acts as a source of DLS-user-data

3.3.30
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

3.3.31
subscribed identified variable
variable that corresponds to a DLCEP-identifier for which the DLE receives data

3.3.32
triggered message scanning
function of a bus-arbitrator that makes it possible to transfer messages

3.3.33
triggered periodic scanning of messages
function of a bus-arbitrator that makes it possible to request triggered message exchanges cyclically

3.3.34
triggered periodic scanning of variables
function of a bus-arbitrator that makes it possible to request triggered variable transfers cyclically

3.3.35
triggered scanning of variables
function of a bus-arbitrator that makes possible the non-cyclical exchange of variables

3.3.36
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

3.3.37
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.

3.4 Symbols and abbreviations

3.4.1 BA Bus-arbitrator

3.4.2 B_Dat_Cons Buffer which contains the value of the subscribed data

3.4.3 B_Dat_Prod Buffer which contains the value of the published data

3.4.4 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)
3.4.5 DL-	Data-link layer (as a prefix)
3.4.6 DLC	DL-connection
3.4.7 DLCEP	DL-connection-end-point
3.4.8 DLE	DL-entity (the local active instance of the data-link layer)
3.4.9 DLL	DL-layer
3.4.10 DLPCI	DL-protocol-control-information
3.4.11 DLPDU	DL-protocol-data-unit
3.4.12 DLM	DL-management
3.4.13 DLME	DL-management Entity (the local active instance of DL-management)
3.4.14 DLMS	DL-management Service
3.4.15 DLS	DL-service
3.4.16 DLSAP	DL-service-access-point
3.4.17 DLSDU	DL-service-data-unit
3.4.18 FIFO	First-in first-out (queuing method)
3.4.19 OSI	Open systems interconnection
3.4.20 Ph-	Physical layer (as a prefix)
3.4.21 PhE	Ph-entity (the local active instance of the physical layer)
3.4.22 PhL	Ph-layer
3.4.23 Q_IDRQ1/2	Queue for the DL-identifiers requested, received by the BA at priority 1 (urgent) or 2 (normal)
3.4.24 Q_Msg_Aper	Queue which contains messages to be emitted that are associated with aperiodic exchanges
3.4.25 Q_Msg_Cyc	Queue which contains messages to be emitted that are associated with cyclical exchanges
3.4.26 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)
3.4.27 QoS	Quality of service
3.4.28 RQ_Inhibit	Indicator used to manage explicit requests for buffer transfers

3.5 Common conventions

This standard uses the descriptive conventions given in ISO/IEC 10731.

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

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

This standard uses a tabular format to describe the component parameters of the DLS primitives. The parameters that apply to each group of DLS primitives are set out in tables throughout the remainder of this standard. Each table consists of up to six columns, containing the name of the service parameter, and a column each for those primitives and parameter-transfer directions used by the DLS:

— the request primitive's input parameters;

— the request primitive's output parameters;

— the indication primitive's output parameters;

— the response primitive's input parameters; and

— the confirm primitive's output parameters.

NOTE The request, indication, response and confirm primitives are also known as requestor.submit, acceptor.deliver, acceptor.submit, and requestor.deliver primitives, respectively (see ISO/IEC 10731).

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

M — parameter is mandatory for the primitive;

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

C — parameter is conditional upon other parameters or upon the environment of the DLS-user;

(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.

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.

4 Data-link layer services and concepts

4.1 Field of application, object

4.1.1 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.

4.1.2 Object

This standard specifies the DLL services. The object is to define:

a) the services provided at the conceptual interface between the DLE 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.

4.2 General description of services

4.2.1 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.

4.2.2 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.

For buffer transfers: 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.

4.2.3 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.

4.2.4 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.

4.2.5 Overall description of medium allocation

4.2.5.1 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.

4.2.5.2 Cyclical buffer transfers, message transfers or service request polling

4.2.5.2.1 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:

4.2.5.2.2 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 2 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.

4.2.5.2.3 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.

4.2.5.2.4 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.

4.2.5.3 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 4.2.5.2.2.

4.2.5.4 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 4.2.5.2.3.

4.2.5.5 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 turn-around 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 turn-around time, and the impact of turn-around time on DL-timers, is given in the corresponding portion of IEC 61158-4-7.

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.

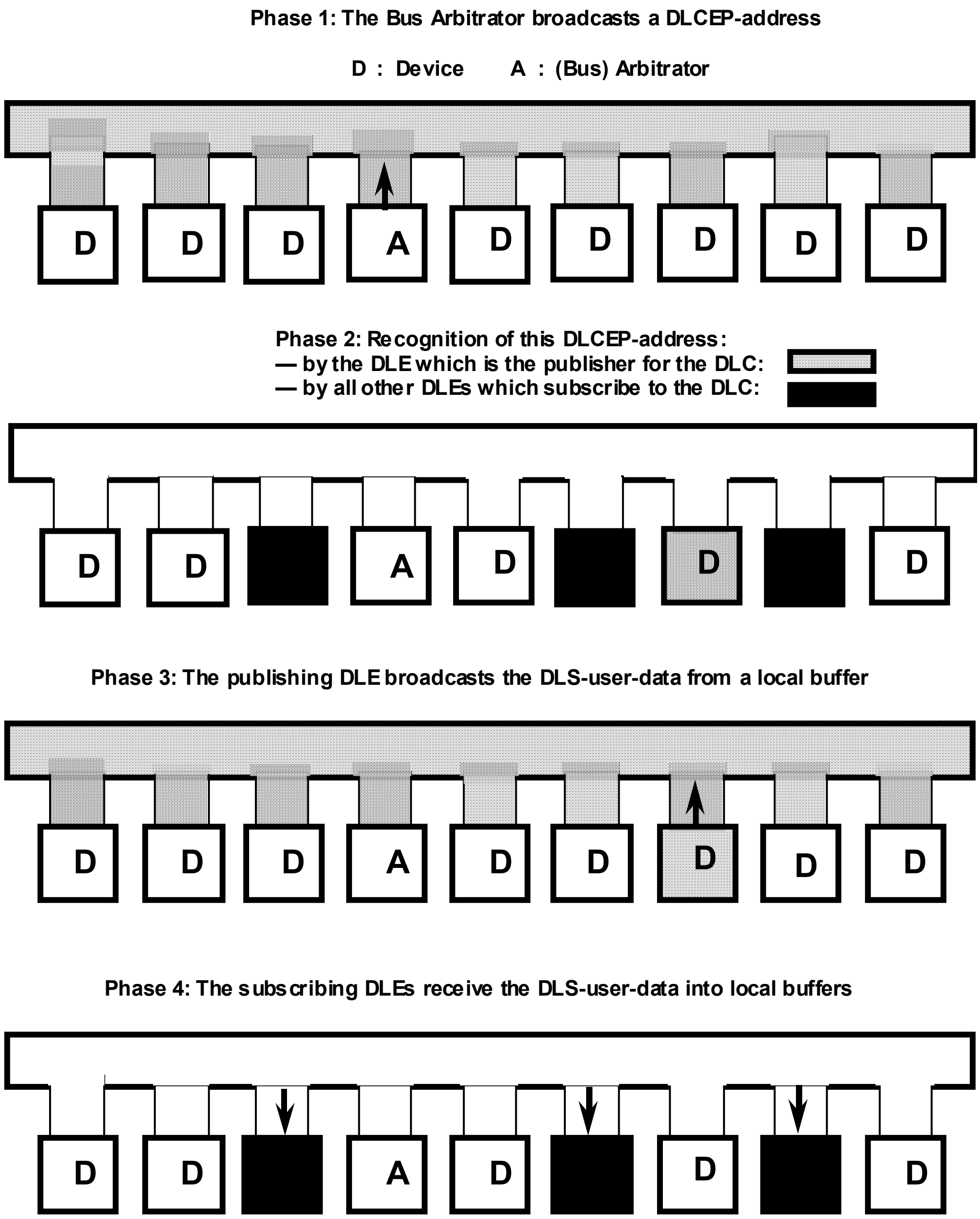


Figure 2 – General description of medium allocation

The medium access technique, shown in Figure 2, 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.

4.2.6 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 (4.7). (See also 4.8 and 4.9.)

4.3 Sequences of primitives

4.3.1 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.

4.3.2 Primitives on buffer transfers

The DL-services and their parameters are summarized in Table 1

Table 1 – 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.		

4.3.3 Primitives on message exchanges

The DL-services and their parameters are summarized in Table 2

Table 2 – 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.		

4.4 Buffer writing

4.4.1 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.

4.4.2 Sequence of primitives

The sequence of primitives in a successful buffer writing is shown in Figure 3:

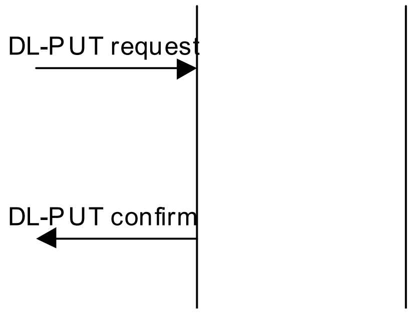


Figure 3 – Primitives associated with the buffer writing service

4.4.3 Types of primitives and parameters

4.4.3.1 General

Table 3 indicates the types of primitives and parameters of writing a buffer.

Table 3 – DL-Put primitives and parameters

DL-PUT	Request	Confirm
Parameter name	input	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.		

4.4.3.2 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.

4.4.3.3 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.

4.4.3.4 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 61158-5-7), uses DLS-user-data that is always two octets or more in length.

4.4.3.5 Confirm primitive

A DL-PUT confirm primitive follows a DL-PUT request primitive and provides an account on the progress of the action requested.

4.4.3.6 Status

This parameter reports on the writing operation. Possible values of this parameter are the following:

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.

4.5 Buffer reading

4.5.1 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.

4.5.2 Sequence of primitives

The sequence of primitives in a successful buffer reading is shown in Figure 4:

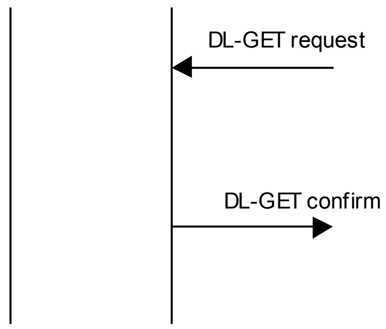


Figure 4 – Primitives associated with the buffer reading service

4.5.3 Types of primitives and parameters

4.5.3.1 General

Table 4 indicates the type of primitives and parameters of reading a buffer:

Table 4 – 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.		

4.5.3.2 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.

4.5.3.3 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.

4.5.3.4 Confirm primitive

A DL-GET confirm primitive follows a DL-GET request primitive and provides an account on the progress of the requested action.

4.5.3.5 Status

This parameter reports on the reading of the variable's value. Possible values of this parameter are the following:

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.

4.5.3.6 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.

4.6 Buffer transfer

4.6.1 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.

4.6.2 Sequence of primitives

The sequence of primitives in a successful buffer transfer is shown in Figure 5:

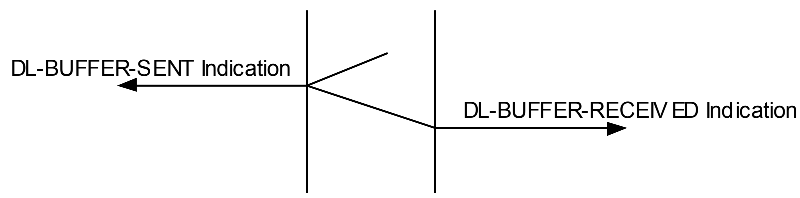


Figure 5 – Primitives associated with the buffer transfer service

4.6.3 Types of primitives and parameters

4.6.3.1 General

Table 5 indicates the type of primitives and parameters for a buffer sent indication

Table 5 – DL-Buffer-Sent primitive and parameter

DL-BUFFER-SENT Parameter name	Indication input
DLCEP-identifier	M

4.6.3.2 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.

4.6.3.3 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 6 indicates the type of primitives and parameters for a buffer received indication.

Table 6 – DL-Buffer-Received primitive and parameter

DL-BUFFER-RECEIVED Parameter name	Indication input
DLCEP-identifier	M

4.6.3.4 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.

4.6.3.5 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).

4.7 Explicit request for buffer transfer

4.7.1 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 following 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 4.2.5.2.2.

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 are conveyed to the bus-arbitrator in the same request response DLPDU; the list cannot be split across two such DLPDUs.

4.7.2 Specified explicit request sequence of primitives

The sequence of primitives in a successful specified explicit update is shown in Figure 6:

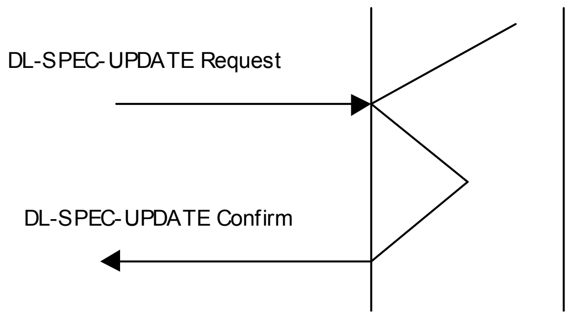


Figure 6 – Primitives associated with the specified explicit request for a buffer transfer

4.7.3 Specified explicit request primitives and parameters

4.7.3.1 General

Table 7 indicates the types of primitives or parameters needed for a specified explicit update

Table 7 – 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.		

4.7.3.2 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.

4.7.3.3 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.

4.7.3.4 List of requested DL-identifiers

This parameter specifies 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.

4.7.3.5 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.

4.7.3.6 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.

4.7.4 Free explicit update sequence of primitives

The sequence of primitives in a successful free explicit update is shown in Figure 7:

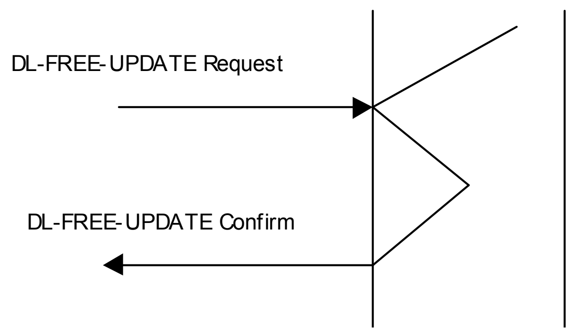


Figure 7 – Primitives associated with the free explicit request for a buffer transfer

4.7.5 Free explicit update primitives and parameters

4.7.5.1 General

Table 8 indicates the types of primitives or parameters needed for an free explicit update.

Table 8 – 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

4.7.5.2 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.

4.7.5.3 List of requested DL-identifiers

This parameter specifies 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.

4.7.5.4 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 standard will indicate priority using "i", that is , i=1 for urgent or high priority, or i=2 for normal priority.

4.7.5.5 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.

4.7.5.6 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.

4.8 Unacknowledged message transfer

4.8.1 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.

4.8.2 Sequence of primitives

The sequence of primitives in a successful unacknowledged message transfer is shown in Figure 8.

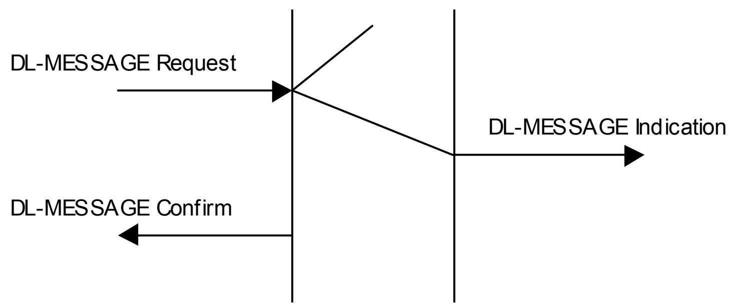


Figure 8 – Primitives associated with the unacknowledged message transfer request service

4.8.3 Types of primitives and parameters

4.8.3.1 General

Table 9 indicates the types of primitives or parameters for an unacknowledged message transfer:

Table 9 – 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.

4.8.3.2 Request primitive

A DL-MESSAGE request allows a DLS-user to request the transmission of an unacknowledged message to the specified DL(SAP)-address.

4.8.3.3 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.

4.8.3.4 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.

4.8.3.5 Source DLSAP-address

This parameter specifies the local DLSAP, associated with the DLS-user, which is to be the attributed source of the message.

4.8.3.6 DLS-user-data

This parameter specifies the information which is being conveyed between the corresponding DLS-users.

4.8.3.7 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.

4.8.3.8 Confirm primitive

A DL-MESSAGE confirm provides the initiating DLS-user with a report on the transmission of an unacknowledged message.

4.8.3.9 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).

4.9 Acknowledged message transfer

4.9.1 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 This standard does not further specify the role of the bridge DLE. Requirements for bridges are specified in IEC 61158-4-7.

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.

4.9.2 Sequence of primitives

The sequence of primitives in a successful acknowledged message transfer is shown in Figure 9:

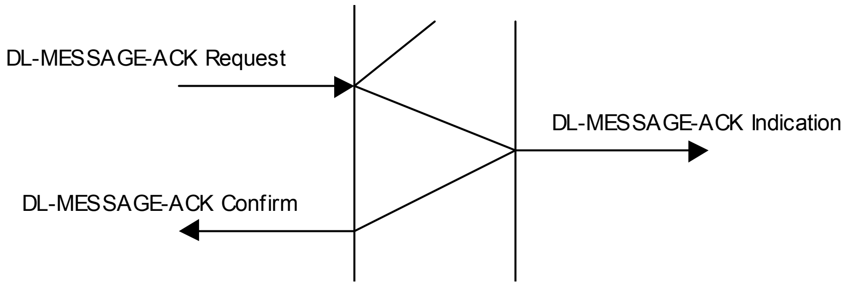


Figure 9 – Primitives associated with the acknowledged message transfer request service

4.9.3 Types of primitives and parameters

4.9.3.1 General

Table 10 indicates the types of primitives or parameters needed for an acknowledged message transfer.

Table 10 – 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. See 1.2.			

4.9.3.2 Request primitive

A DL-MESSAGE-ACK request allows a DLS-user to request the transmission of an acknowledged message to the specified DLSAP-address.

4.9.3.3 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.

4.9.3.4 Destination DLSAP-address

This parameter specifies the DLSAP which is the destination of the message.

4.9.3.5 Source DLSAP-address

This parameter specifies the local DLSAP, associated with the DLS-user, which is to be the attributed source of the message.

4.9.3.6 DLS-user-data

This parameter specifies the information which is being conveyed between the corresponding DLS-users.

4.9.3.7 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.

4.9.3.8 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.

4.9.3.9 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.

Bibliography

IEC/TR 61158-1 (Ed.2.0), *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-2 (Ed.4.0), *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-4-7, *Industrial communication networks – Fieldbus specifications – Part 4-7: Data-link layer protocol specification – Type 7 elements*

IEC 61158-5-7, *Industrial communication networks – Fieldbus specifications – Part 5-7: Application layer service definition – Type 7 elements*

IEC 61158-6-7, *Industrial communication networks – Fieldbus specifications – Part 6-7: Application layer protocol specification – Type 7 elements*

IEC 61784-1 (Ed.2.0), *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
P.O. Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch