IEC 61158-4-4

Edition 4.0 2023-03

INTERNATIONAL STANDARD

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

IEC 61158-4-4:2023-03(en)

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

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

IEC 61158-4-4

Edition 4.0 2023-03

INTERNATIONAL STANDARD

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.20; 35.110

ISBN 978-2-8322-6574-1

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

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

FOREWORD

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

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

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

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

This fourth edition cancels and replaces the third edition published in 2018. This edition constitutes a technical revision.

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

a) Use of extended data size for DLS-user data. This extension is restricted to nodes operating on a P-NET IP network.

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

Draft	Report on voting
65C/1202/FDIS	65C/1243/RVD

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

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

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

INTRODUCTION

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

The data-link protocol provides the data-link service by making use of the services available from the physical layer. The primary aim of this document is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer data-link entities (DLEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

a) as a guide for implementors and designers;
b) for use in the testing and procurement of equipment;
c) as part of an agreement for the admittance of systems into the open systems environment;
d) as a refinement to the understanding of time-critical communications within OSI.

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

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

1 Scope

1.1 General

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

This protocol provides a means of connecting devices through a partial mesh network, such that most failures of an interconnection between two devices can be circumvented. In common practice the devices are interconnected in a non-redundant hierarchical manner reflecting application needs.

1.2 Specifications

This document specifies

a) procedures for the timely transfer of data and control information from one data-link user entity to a peer user entity, and among the data-link entities forming the distributed data-link service provider;

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

1.3 Procedures

The procedures are defined in terms of

a) the interactions between peer DL-entities (DLEs) through the exchange of fieldbus DLPDUs;

b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;

c) the interactions between a DLS-provider and a Ph-service provider in the same system through the exchange of Ph-service primitives.

1.4 Applicability

These procedures are applicable to instances of communication between systems which support time-critical communications services within the data-link layer of the OSI or fieldbus reference models, and which require the ability to interconnect in an open systems interconnection environment.

Profiles provide a simple multi-attribute means of summarizing an implementation's capabilities, and thus its applicability to various time-critical communications needs.

1.5 Conformance

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

2 Normative references

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

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

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

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

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

3 Terms, definitions, symbols and abbreviated terms

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

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

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

3.1 Reference model terms and definitions

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

3.1.1 called-DL-address [7498-3]

3.1.2 calling-DL-address [7498-3]

3.1.3 centralized multi-end-point-connection [7498-1]

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

3.1.5 demultiplexing [7498-1]

3.1.6 DL-address [7498-3]

3.1.7 DL-address-mapping [7498-1]

3.1.8 DL-connection [7498-1]

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

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

3.1.11 DL-connection-mode transmission [7498-1]

3.1.12 DL-connectionless-mode transmission [7498-1]

3.1.13 DL-data-sink [7498-1]

3.1.14 DL-data-source [7498-1]

3.1.15 DL-duplex-transmission [7498-1]

3.1.16 DL-facility [7498-1]

3.1.17 DL-local-view [7498-3]

3.1.18 DL-name [7498-3]

3.1.19 DL-protocol [7498-1]

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

3.1.21 DL-protocol-control-information [7498-1]

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

3.1.23 DL-protocol-version-identifier [7498-1]

3.1.24 DL-relay [7498-1]

3.1.25 DL-service-connection-identifier [7498-1]

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

3.1.27 DL-simplex-transmission [7498-1]

3.1.28 DL-subsystem [7498-1]

3.1.29 DL-user-data [7498-1]

3.1.30 flow control [7498-1]

3.1.31 layer-management [7498-1]

3.1.32 multiplexing [7498-3]

3.1.33 naming-(addressing)-authority [7498-3]

3.1.34 naming-(addressing)-domain [7498-3]

3.1.35 naming-(addressing)-subdomain [7498-3]

3.1.36 (N)-entity [7498-1]
DL-entity
Ph-entity

3.1.37 (N)-interface-data-unit [7498-1]
DL-service-data-unit (N=2)
Ph-interface-data-unit (N=1)

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

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

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

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

3.1.42 peer-entities [7498-1]

3.1.43 Ph-interface-control-information [7498-1]

3.1.44 Ph-interface-data [7498-1]

3.1.45 primitive name [7498-3]

3.1.46 reassembling [7498-1]

3.1.47 recombining [7498-1]

3.1.48 reset [7498-1]

3.1.49 responding-DL-address [7498-3]

3.1.50 routing [7498-1]

3.1.51 segmenting [7498-1]

3.1.52 sequencing [7498-1]

3.1.53 splitting [7498-1]

3.1.54 synonymous name [7498-3]

3.1.55 systems-management [7498-1]

3.2 Service convention terms and definitions

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

3.2.1 acceptor

3.2.2 asymmetrical service

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

3.2.4 deliver (primitive)

3.2.5 DL-confirmed-facility

3.2.6 DL-facility

3.2.7 DL-local-view

3.2.8 DL-mandatory-facility

3.2.9 DL-non-confirmed-facility

3.2.10 DL-provider-initiated-facility

3.2.11 DL-provider-optional-facility

3.2.12 DL-service-primitive;
primitive

3.2.13 DL-service-provider

3.2.14 DL-service-user

3.2.15 DL-user-optional-facility

3.2.16 indication (primitive)
acceptor.deliver (primitive)

3.2.17 multi-peer

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

3.2.19 requestor

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

3.2.21 submit (primitive)

3.2.22 symmetrical service

3.3 Terms and definitions

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

3.3.1
broadcast-Node-address
address used to send broadcasts to all DLEs on a Link

Note 1 to entry: 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 a Broadcast-Node-address is 126.

3.3.2
destination-DL-route
holds a sequence of DL-route-elements, describing the complete route to the destination

Note 1 to entry: This includes both the destination DLSAP and a local component meaningful to the destination DLS-user.

3.3.3
DL-route
combination of a Destination-DL-route and a Source-DL-route

3.3.4
DL-route-element
octet holding a Node-address or an address used by the DLS-user

3.3.5
DLSAP
distinctive point at which DL-services are provided by a single DL-entity to a single higher-layer entity

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

3.3.6
DL(SAP)-address
an individual DLSAP-address, designating a single DLSAP of a single DLS-user

3.3.7
(individual) DLSAP-address
DL-address that designates only one DLSAP within the extended link

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

3.3.8
frame
denigrated synonym for DLPDU

3.3.9
IPNetID
identification of a unique IP network

Note 1 to entry: An IPNetID is translated into an IP-address and a UPD port number.

3.3.10
IPNetTable
definition of the relation between IPNetID, IP address, UPD port number and Router NodeAddress, where IPNetID is used as index in the table

3.3.11
IP Range net
definition of the use of the IP network for local access, where nodes can be accessed directly on the same subnet as the client, or through a local Router where the subnets are configured in the local Router

3.3.12
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

3.3.13
no-Confirm-Node-address
address used to indicate that a request or response is Unconfirmed

Note 1 to entry: The value of a No-Confirm-Node-address is 0.

3.3.14
node
single DL-entity as it appears on one local link

3.3.15
node-address
address which uniquely identifies a DLE on a Link

Note 1 to entry: The value of a Node-address can be in the range of 0 to 127, with the values 0, 126 and 127 reserved for special purposes.

3.3.16
normal class device
device which replies to requests from other normal class devices, and initiates transmissions

Note 1 to entry: Such a device can act as a server (responder) and as a client (requestor) – this is also called a peer.

3.3.17
Type 4-route
a route that holds a sequence of Type 4-route-elements

Note 1 to entry: A Type 4-route is defined as an encoded DL-route, with one of the formats used when transmitting the DLPDU on the Link. The Type 4-route format can be Simple, Extended, Complex, Immediate or IP.

3.3.18
Type 4-route-element
octet, holding a 7-bit DL-route-element or Remaining-route-length, and a 1-bit source/destination designator

3.3.19
receiving DLS-user
DL-service user that acts as a recipient of DL-user-data

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

3.3.20
sending DLS-user
DL-service user that acts as a source of DL-user-data

3.3.21
service-Node-address
address reserved for service purposes only

Note 1 to entry: 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 it is possible that their receipt can be acknowledged or not. 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.

3.3.22
simple class device
device which replies to requests from normal class devices, and can act as a server or responder only

3.3.23
source-DL-route
route that holds a sequence of DL-route-elements, describing the complete route back to the source

3.3.24
UDP port number
port number from where a Server can receive requests

Note 1 to entry: The UDP port number is 34378 for Normal UDP port. The UDP port number is 34379 for Secure UDP port.

Note 2 to entry: These UDP port numbers are registered with the IANA (Internet Assigned Numbers Authority).

Note 3 to entry: There are two different UPD port numbers: Normal UDP port and Secure UDP port.

3.3.25
UDP range net
definition of the use of the IP network for remote access, where a node cannot be accessed directly on the same subnet as the client

Note 1 to entry: The IPNetTable holds a NAT Router IP address and access to the node is obtained through this NAT Router.

Note 2 to entry: The NAT Router shall hold a table that translates the UDP port number to the actual server node IP address and UDP port number.

3.3.26
virtual link-access token
basis for the link-access system

Note 1 to entry: It is called virtual because the token is not explicitly sent from one normal-class DLE to another, but implicitly passed as the link is idle.

3.4 Symbols and abbreviations

3.4.1 Constants, variables, counters and queues

3.4.1.1	**BNA**	broadcast node address
3.4.1.2	**C(LAC)**	link access counter
3.4.1.3	**C(LIC)**	link idle counter
3.4.1.4	**SNA**	service node address
3.4.1.5	**NCNA**	no confirm node address
3.4.1.6	**Q(UR)**	user request queue
3.4.1.7	**V(ACPDU)**	acknowledge confirmed PDU
3.4.1.8	**V(AUPDU)**	acknowledge unconfirmed PDU
3.4.1.9	**V(BR)**	bit rate
3.4.1.10	**V(DC)**	device class (simple or normal)
3.4.1.11	**V(DMRT)**	default max retry time
3.4.1.12	**V(MID)**	max indication delay
3.4.1.13	**V(NA)**	node address
3.4.1.14	**V(NDLE)**	number of DLEs
3.4.1.15	**V(PNR)**	permitted number of retries
3.4.1.16	**IPNetTable**	Table to convert IPNetID to IP-addresses

3.4.2 Miscellaneous

3.4.2.1	**RCL/ACK**	response comes later / acknowledge

4 Data Link Protocol Definition

4.1 Overview of the DL-protocol

4.1.1 General

The DLL provides connectionless data transfer services for limited-size DLSDUs from one DLS-user to one or more (broadcast) DLS-users.

A DLE is implicitly connected to one PhE and to a single DLSAP. This means that when a local DLS-user issues a service primitive at a certain DLSAP, the DLE and hence the Link is implicitly selected.

A DLE always delivers received DLSDUs at the same DLSAP, and hence to the same DLS-user.

This concept is illustrated in Figure 1.

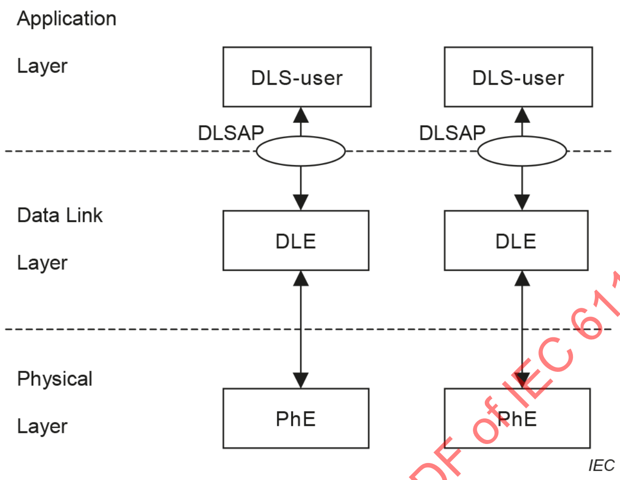

Figure 1 – Relationship of PhE, DLE and DLS-user

Each DLE has a Node-address. Node-addresses uniquely identify DLEs within the same Link.

A DL-route-element is an octet, which can hold a Node-address, or an address used by the DLS-user.

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

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

A DL-route is defined as a Destination-DL-route and a Source-DL-route.

4.1.2 Functional classes

The functional class of a DLE determines its capabilities, and thus the complexity of conforming implementations. Two functional classes are defined:

- Simple class, including only responder functionality (server).
- Normal class, including initiator and responder functionality (client and server, also called peer).

4.1.3 Functions of the DLL

4.1.3.1 General

The functions of the DLL are those necessary to bridge the gap between the services available from the PhL and those offered to DLS-users. The functions are:

As a responder (in Simple class or Normal class DLEs):

a) Receive a DLPDU from a remote DLE, perform frame check, parse the received DLPDU into its DL-protocol information and data components, and generate a DLS-user indication primitive. Possibly wait for a DLS-user request or response primitive, convert it to a DLPDU, and send that DLPDU to the remote DLE.

b) Receive a single PhIDU specifying LINK-IDLE, and use that to time-out when waiting for a DLS-user request primitive.

As an initiator (in Normal class DLEs):

c) Convert a DLS-user request primitive to a DLPDU, queue it, and send it to a remote DLE (or all DLEs at the Link if broadcast) at the first opportunity. Possibly wait for an Acknowledge or Immediate-reply DLPDU from the remote DLE, and (if an Immediate-reply DLPDU is received) generate a DLS-user indication primitive.

d) Receive an SPDU, and use the associated data to check or gain Link-access synchronization.

e) Receive a single PhIDU specifying LINK-IDLE, use that to keep Link-access synchronized, and possibly to initiate sending a DLPDU from the queue if the queue is not empty, or if the queue is empty, to send an SPDU for Link-access synchronization.

These functions are illustrated in Figure 2 to Figure 4.

4.1.3.2 Acknowledged vs. confirmed

The terms acknowledged and unacknowledged are used to describe whether the receiving DLE must acknowledge the receipt of a DLPDU or not. The terms confirmed and unconfirmed are used to describe whether the receiving DLS-user must confirm the receipt of a DLSDU or not.

The variable V(ACPDU) – Acknowledge Confirmed PDU – defines whether the DLE must acknowledge the receipt of Confirmed DLPDUs. The variable V(AUPDU) – Acknowledge Unconfirmed PDU – defines whether the DLE must acknowledge the receipt of Unconfirmed DLPDUs.

A special case is when the first Node-address in a received DLPDU is equal to the Broadcast-Node-address (BNA). In this case, the receiving DLE shall never acknowledge the receipt of the DLPDU.

4.1.3.3 Half-duplex and full duplex

Unless otherwise stated, the PhL is assumed to support half-duplex transfer. However, a PhL supporting full duplex is allowed.

Full duplex systems allow up to 125 DLEs on a Link, all of Normal class. Each DLE is allowed to transmit immediately, that is, there is no Link Access system. DLEs supporting full duplex PhEs have separate state machines for receive and transmit, as illustrated in Figure 5 and Figure 6.

In full duplex systems, Confirmed as well as Unconfirmed DLPDUs are unacknowledged.

PhLs supporting full duplex shall not provide Link-Idle indications.

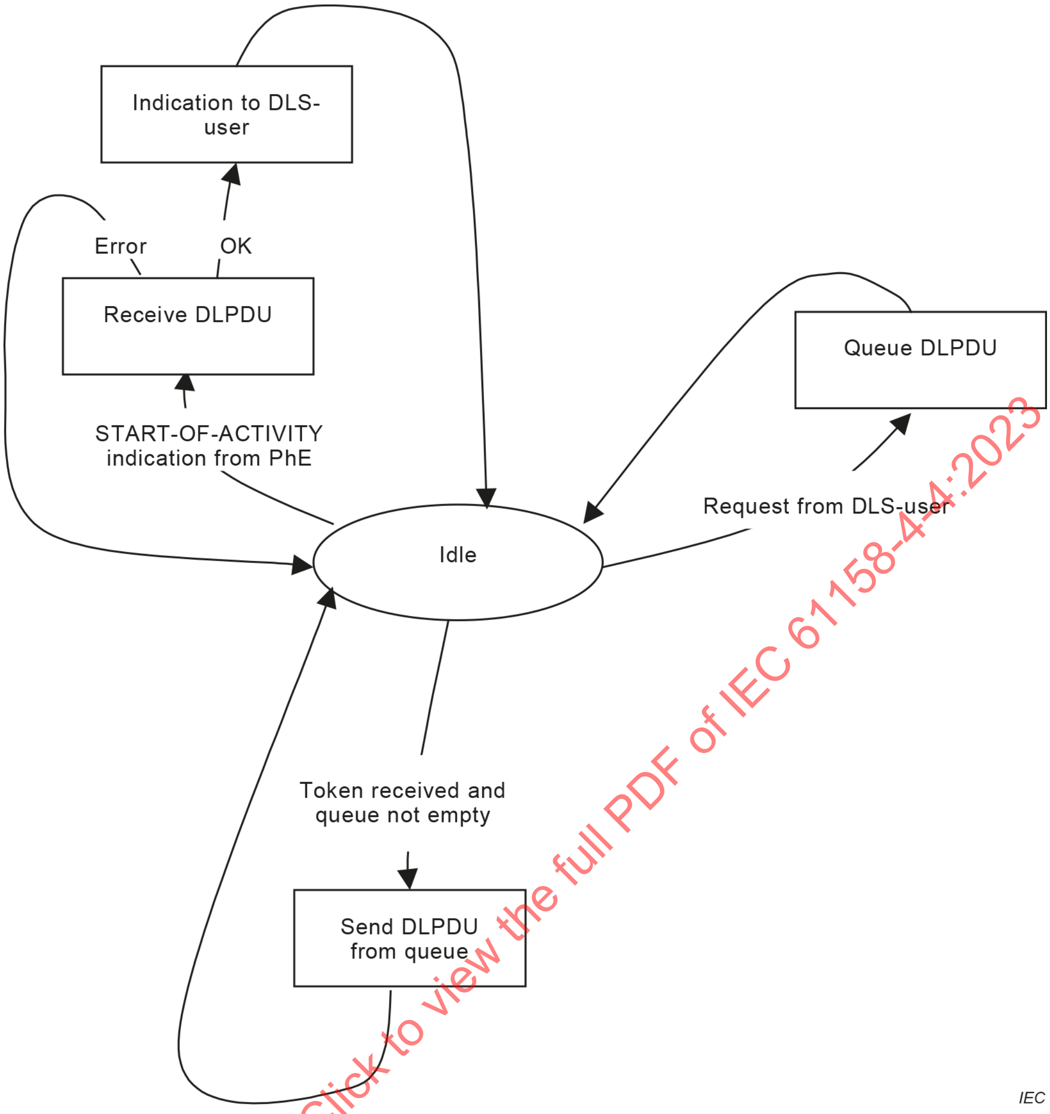

Figure 2 – DLE state diagram for confirmed and unconfirmed, unacknowledged DLPDUs

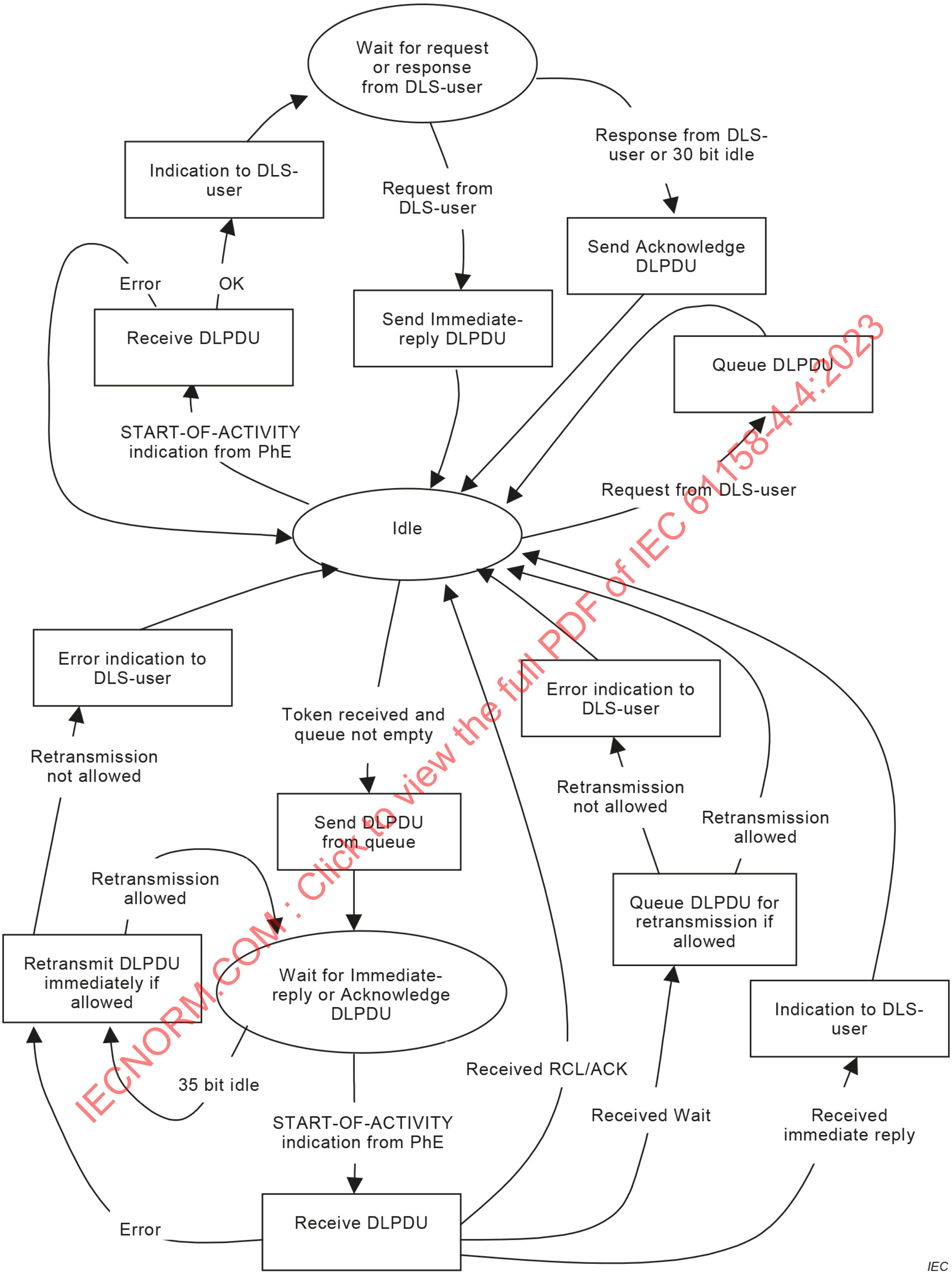

Figure 3 – DLE state diagram for confirmed acknowledged DLPDUs

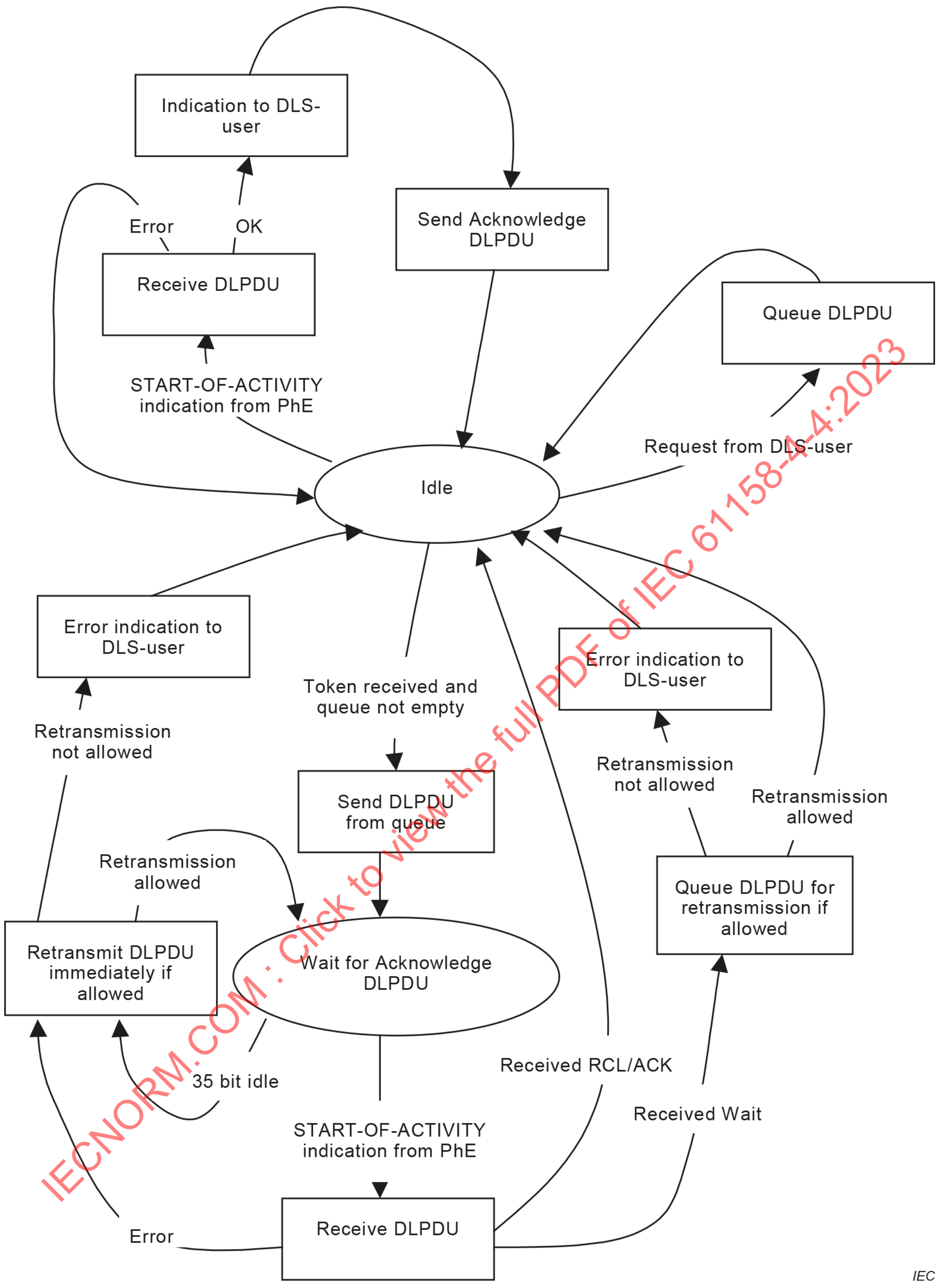

Figure 4 – DLE state diagram for unconfirmed acknowledged DLPDUs

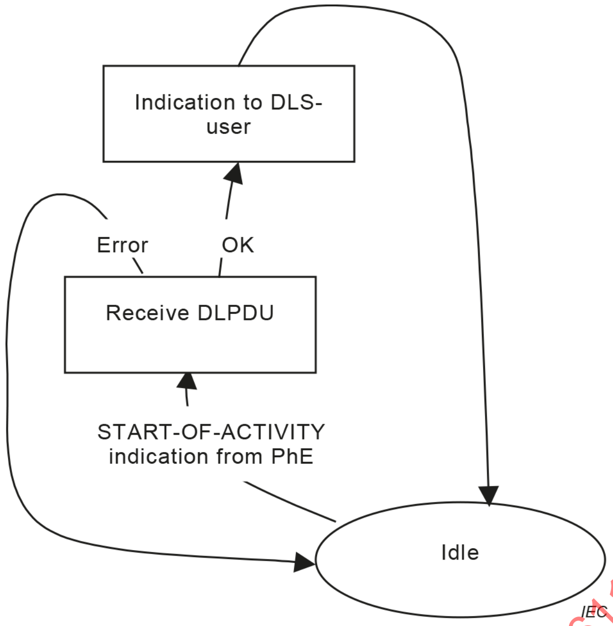

Figure 5 – Full duplex DLE receive state diagram

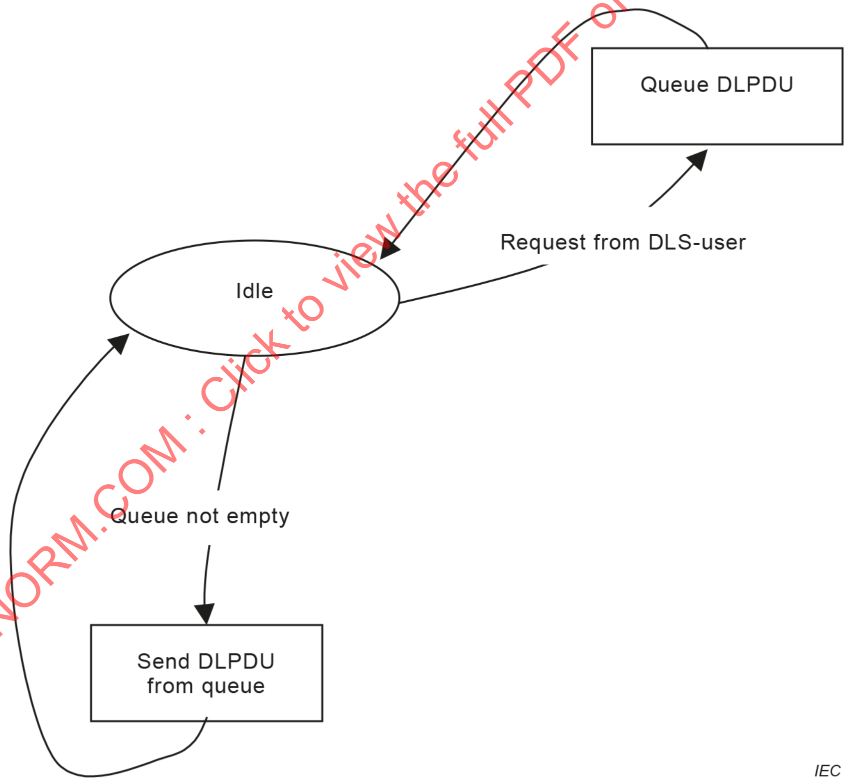

Figure 6 – Full duplex DLE transmit state diagram

4.1.3.4 DLPDU types

Four different types of DLPDUs are defined.

a) Confirmed – used to send confirmed requests between DLS-users.

b) Unconfirmed – used to send responses or unconfirmed requests between DLS-users.

c) Acknowledge – used by DLEs to acknowledge receipt of Confirmed or Unconfirmed DLPDUs. The receipt of Acknowledge DLPDUs shall never be acknowledged.

d) Immediate-reply – used to send responses between DLS-users. The receipt of Immediate-reply DLPDUs shall never be acknowledged.

4.1.3.5 SPDU types

Only one type of SPDU (Support Protocol Data Unit) is defined.

a) Sync – used to send Link access synchronization information between DLEs. An SPDU holds the Node-address of the DLE holding the Virtual Link-access token. An SPDU can be "stand-alone" or part of an Acknowledge or Immediate-reply DLPDU.

4.1.3.6 Responder role, receiving a DLPDU from the PhE

This action includes a sequence of steps, as described in the following.

a) Receive a single PhIDU specifying START-OF-ACTIVITY. This PhIDU holds a Node address. This address is examined to determine whether its value is equal to the Node-address of this DLE, or equal to the Broadcast-Node-address (BNA) or the Service-Node-Address (SNA). If not, ignore this sequence and wait for the next PhIDU specifying START-OF-ACTIVITY.
b) Receive a sequence of PhIDUs from the PhE, specifying DATA, concatenate them to a received DLPDU, compute a frame check sequence over the entire sequence of received data as specified by the value of V(FCM) – FrameCheckMethod, and, if necessary, check for the proper value. If the value is not correct, ignore the DLPDU and wait for the next PhIDU specifying START-OF-ACTIVITY.
c) Convert the received DLPDU into its DL-protocol control information and data components.
d) Generate a DLS-user indication primitive.
e) If the DLPDU received from the remote DLE is of type Confirmed, and the receipt of the DLPDU shall be acknowledged, according to the rules described in 4.1.3.2, wait for a request or response primitive from the local DLS-user.

   If no request or response primitive is issued from the local DLS-user in time (before a PhIDU specifying "LINK-IDLE for 30 bit periods" is received from the PhE), generate and immediately send an Acknowledge DLPDU. This DLPDU shall specify "Wait" if this DLE is of Simple class, and "Response Comes Later / Acknowledge" ("RCL/ACK") if this DLE is of Normal class.

   If a response primitive is issued from the local DLS-user in time, generate and immediately send an Acknowledge DLPDU, specifying "Wait" if this DLE is of Simple class, and "RCL/ACK" if this DLE is of Normal class.

   If a request primitive is issued from the local DLS-user in time, convert it into an Immediate-reply DLPDU and send it immediately. After sending, wait for the next PhIDU specifying START-OF-ACTIVITY.
f) If the DLPDU received from the remote DLE is of the Confirmed type, and the receipt of the DLPDU shall not be acknowledged, wait for the next PhIDU specifying START-OF-ACTIVITY.
g) If the DLPDU received from the remote DLE is of the Unconfirmed type, and the receipt of the DLPDU shall be acknowledged, according to the rules described in 4.1.3.2, generate and immediately send an Acknowledge DLPDU, specifying RCL/ACK. After sending, wait for the next PhIDU specifying START-OF-ACTIVITY.
h) If the DLPDU received from the remote DLE is of the Unconfirmed type, and the receipt of the DLPDU shall not be acknowledged, wait for the next PhIDU specifying START-OF-ACTIVITY.

4.1.3.7 Responder role, receiving a PhIDU specifying LINK-IDLE

As a responder, when waiting for a request or response primitive from the local DLS-user, the receipt of a PhIDU from the PhE specifying "LINK-IDLE for 30 bit periods" is used to timeout waiting for the DLS-user. The possible actions resulting from the timeout are defined in 4.1.3.6.

4.1.3.8 Initiator role, managing request primitives from the local DLS-user

This action includes a sequence of steps, as described in the following:

a) Convert a request primitive from the local DLS-user into a DLPDU, queue it, and send it to a remote DLE (or all DLEs on the Link if broadcast) at the first opportunity.

b) If the DLPDU sent is of type Unconfirmed, and the receiving DLE should acknowledge the receipt, according to the rules defined in 4.1.3.2, wait for an Acknowledge DLPDU from the remote DLE specifying RCL/ACK. If no acknowledge is received in time (before a PhIDU specifying "LINK-IDLE for 35 bit periods" is received from the PhE), immediately re-transmit the DLPDU if the permitted number of transmission retries have not been sent. If the permitted number of transmission retries have failed, do nothing, and this action is completed.

c) If the DLPDU sent is of type Unconfirmed, and the receiving DLE should not acknowledge the receipt, this action is completed.

d) If the DLPDU sent is of type Confirmed, and the receiving DLE should acknowledge the receipt, wait for an Immediate-reply DLPDU holding the response, or an Acknowledge DLPDU, from the remote DLE.

   If an Acknowledge DLPDU is received from the remote DLE in time (before a PhIDU specifying "LINK-IDLE for 35 bit periods" is received from the PhE), and the acknowledge specifies "RCL/ACK", this action is completed. If the acknowledge specifies "Wait", queue the DLPDU for retransmission if the associated retry timer has not expired. If the retry timer has expired, generate a DLS-user indication primitive with the appropriate error information.

   If an Immediate-reply DLPDU holding the response is received in time from the remote DLE, convert the received DLPDU into its DL-protocol control information and data components, and generate a DLS-user indication primitive.

   If neither acknowledge nor response is received from the remote DLE in time, re-transmit the DLPDU immediately (while this DLE still holds the Virtual Link-access token) if the permitted number of transmission retries have not been sent. If the permitted number of transmission retries have failed, generate a DLS-user indication primitive with the appropriate error information.

e) If the DLPDU sent is of type Confirmed, and the receiving DLE should not acknowledge the receipt, this action is completed.

4.1.3.9 Initiator role, link-access

The Link-access system is based on a so-called Virtual Link-access token. Virtual because the token is not explicitly sent from one Normal class DLE to another, but implicitly passed as the Link is idle.

The following DLE variables and counters are used by the Link-access system.

– V(NA) – Node-address. Each DLE on a Link is uniquely identified by its Node-address, the value of which is stored in V(NA). The value of V(NA) shall be different in all DLEs on the Link.

– V(NDLE) – Number of DLEs – holds the maximum number of Normal class DLEs on the Link. The value of V(NA) shall be lower than or equal to the value of V(NDLE). The value of V(NDLE) shall not exceed 32. The value of V(NDLE) shall be the same in all DLEs on the Link.

– C(LAC) – Link Access Counter – holds the Node-address of the DLE holding the Virtual Link-access token. The value of C(LAC) will be the same in all DLEs on the Link.

– C(LIC) – Link Idle Counter – holds information on, for how long the Link has been idle. The value of C(LIC) will be the same in all DLEs on the Link.

Figure 7 illustrates the functionality of the Link-access system. The "Action" line describes the use of the Link. The first action is that the DLE having Node-address 2 sends a Confirmed DLPDU, and receives the corresponding Immediate-reply DLPDU. The second action is that the

DLE having Node-address 3 sends an Unconfirmed DLPDU. Then, after a long idle period, the DLE with Node-address 2 sends a Sync SPDU.

The DLE having Node-address 4 is not present. Had it been present, DLE4 should have sent the Sync SPDU, as the Link had been idle for 360 bit periods when it "received" the Virtual Link-access token. The next DLE holding the token is DLE1, which is present and therefore sends the Sync SPDU.

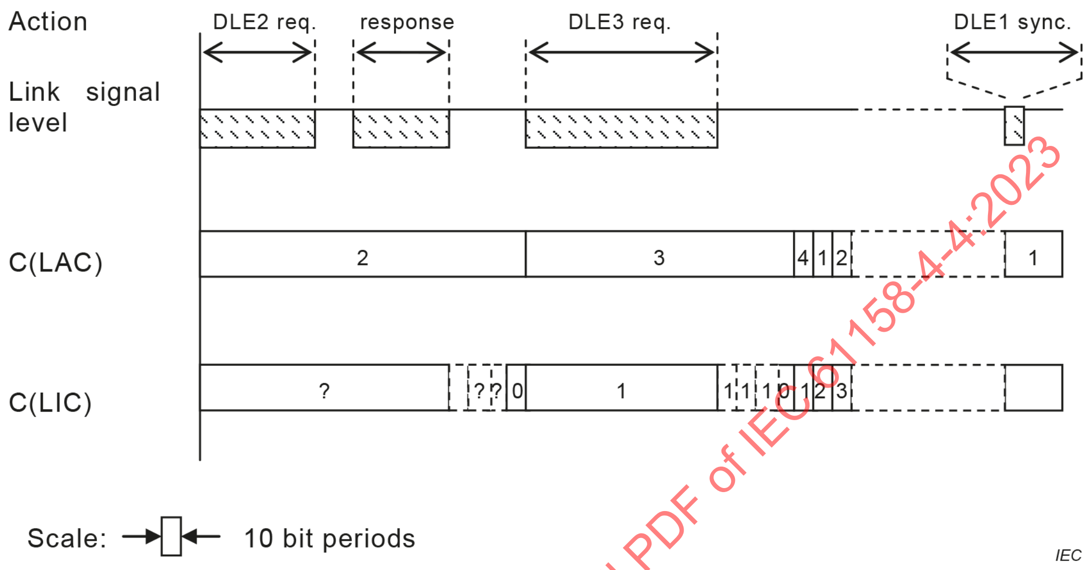

Figure 7 – Link access example

Each single PhIDU specifying LINK-IDLE holds information on, whether the Link has been idle for 30 bit periods, for 35 bit periods, or for 40 or more bit periods in the associated status parameter.

Each time a LINK-IDLE specifying that the Link has been idle for 40 or more bit periods is received, the value of C(LAC) – Link Access Counter – and the value of C(LIC) – Link Idle Counter – is incremented by 1. When the value of C(LAC) becomes higher than the value of V(NDLE), the value of C(LAC) is set to 1.

Each time a LINK-IDLE specifying that the Link has been idle for 30 bit periods is received, the value of C(LIC) is set to 0.

If, immediately after incrementing C(LAC), the value of C(LAC) is equal to the Node-address of this DLE, it means this DLE holds the Virtual token, and therefore is allowed to send (and possibly re-transmit) a DLPDU from the queue. This shall be initiated immediately, by sending a START-OF-ACTIVITY-2 to the PhE. It is a task of the implementation to ensure that the transmission is initiated within 7 bit-periods after receipt of the LINK-IDLE service primitive. If the queue is empty, the DLE shall check the value of C(LIC), to see for how long time the Link has been idle. If the value of C(LIC) is equal to or higher than 33, it means the Link has been idle for 360 bit periods or more. If this applies, the DLE shall send a Sync SPDU for Link-access synchronization. This shall be done immediately, by sending a START-OF-ACTIVITY-2 to the PhE. The associated data field shall specify Source, and hold the Node-address of this DLE. This system is used to keep the idle counters in all PhEs on the Link, and thus the values of C(LAC) and C(LIC) in all DLEs on the Link, synchronized.

Each single PhIDU specifying START-OF-ACTIVITY holds a Node-address and a Source/Destination designator in the associated data field. If the Node-address is a source Node-address, it identifies the DLE holding the Virtual Link-access token at this moment. Such a PhIDU forms a complete Sync SPDU.

When the DLE receives a Sync SPDU, it shall compare the received Node-address with the value of C(LAC). If the 2 values are equal, it means the DLE is synchronized to the other DLEs on the Link. If they are not equal, it means the DLE is out of synchronization. As long as the DLE is out of synchronization, it is only allowed to act as a responder. Subclause 4.6 describes how to gain Link-access synchronization again.

4.1.4 Service assumed from the PhL

4.1.4.1 General

Subclause 4.1.4 defines the assumed Physical Service (PhS) primitives and their constraints on use by the DLE.

4.1.4.2 Assumed primitives of the PhS

The granularity of transmission in the fieldbus protocol is one octet. This is the granularity of PhS-user data exchanged at the PhL – DLL interface.

4.1.4.3 PhS management services

The PhS is assumed to provide the following service primitives to get and set PhE parameter values:

a) Ph-SETVALUE request (parameter name, new value)

b) Ph-SETVALUE confirm (status),

c) Ph-GETVALUE request (parameter name),

d) Ph-GETVALUE confirm (current value).

These services are used by the DLE to

1) set the bit rate, as a result of bit rate changing through DL-management;

2) get the bit rate, as a result of bit rate or Max Indication Delay reading through DL-management. The value of Max Indication Delay is calculated from the current value of bit rate, and shall indicate a value corresponding to 30 bit periods.

4.1.4.4 PhS transmission and reception services

The PhS is assumed to provide the following service primitives for transmission and reception:

a) Ph-DATA request (class, data),

b) Ph-DATA indication (class, data, status),

c) Ph-DATA confirm (status),

where

class — specifies the Ph-interface-control-information (PhICI) component of the Ph-interface-data-unit (PhIDU).

For a Ph-DATA request, its possible values are

START-OF-ACTIVITY-1 – the PhE shall enable its driver, and initiate transmission by transmitting the associated data parameter as an "Address character". The PhE shall do this immediately, though not until the value of the PhE's idle counter has reached 11.

START-OF-ACTIVITY-2 – the PhE shall enable its driver, and initiate transmission by transmitting the associated data parameter as an "Address character". The PhE shall do this immediately, though not until the value of the PhE's idle counter modulus 10 has reached 2.

DATA – the PhE shall transmit the associated data parameter as a “Data character” immediately.

END-OF-ACTIVITY – the PhE shall wait till transmission of all formerly received data from the DLE has finished, and then disable its driver. The associated data parameter shall not be transmitted.

For a Ph-DATA indication, its possible values are

START-OF-ACTIVITY – the PhE has received an “Address character”, the value of which is reported in the associated data parameter. The associated status parameter specifies success or the locally detected reason for failure.

DATA – the PhE has received a “Data character”, the value of which is reported in the associated data parameter. The associated status parameter specifies success or the locally detected reason for failure.

LINK-IDLE – the PhE has detected, that the signal level on the Link has been “Idle” for 30, 35, 40, 50, 60… bit periods. The associated status parameter specifies if the Link has been idle for 30 bit periods, for 35 bit periods, or for 40 or more bit periods.

NOTE The PhE holds an idle counter. This counter is incremented by one each time the signal level on the Link has been idle for one bit period. Each time the signal level is not idle, the idle counter is cleared. When the idle counter reaches 30, the PhE reports this with a Ph-DATA indication of class LINK-IDLE, and associated status indicating 30 bit periods. Five bit periods later, if the Link is still idle, the PhE reports this with another Ph-DATA indication of class LINK-IDLE, and associated status indicating 35 bit periods. Five bit periods later, if the Link is still idle, the PhE reports this with another Ph-DATA indication of class LINK-IDLE, and associated status indicating 40 or more bit periods. This goes on for each 10 bit period with indications specifying 40 or more bit periods, till the signal level on the Link is no longer idle.

data – specifies the Ph-interface-data (PhID) component of the PhIDU. It consists of one octet of Ph-user data to be transmitted (Ph-DATA request), or one octet of Ph-user data which was received (Ph-DATA indication).

status – specifies either success or the locally detected reason for failure, or specifies if the associated LINK-IDLE indication indicates "30", "35" or "40 or more" bit periods of idle after Link activity.

The Ph-DATA confirm primitive provides the feedback necessary to enable the DLE to report failures such as Link short-circuit or noise resulting in framing error to the DLS-user, and provides the critical physical timing necessary to prevent the DLE from starting a second transmission before the first is complete.

4.1.4.5 Transmission of Ph-user data

When a DLE has a DLPDU to transmit, and the Link-access system gives that DLE the right to transmit, then the DLE shall send the DLPDU, including a concatenated FCS. Making a sequence of Ph-DATA requests as follows does this:

a) the first request shall specify START-OF-ACTIVITY-11 if the DLPDU to transmit is an Acknowledge or Immediate-reply DLPDU, or if the transmission is an immediate re-transmission of a Confirmed or Unconfirmed DLPDU. The first request shall specify START-OF-ACTIVITY-2 if transmission of a Confirmed or Unconfirmed DLPDU from the queue is commenced;
b) this first request shall be followed by consecutive requests specifying DATA, and concluded by a single request specifying END-OF-ACTIVITY.

The PhE signals its completion of each Ph-DATA request, and its readiness to accept a new Ph-DATA request, with a Ph-DATA confirm primitive. The status parameter of the Ph-DATA confirm primitive conveys the success or failure of the associated Ph-DATA request.

4.1.4.6 Reception of Ph-user data

The PhE reports a received transmission with Ph-DATA indications, which shall consist of either a single indication specifying START-OF-ACTIVITY, or a single indication specifying START-OF-ACTIVITY followed by consecutive indications specifying data. Each indication has an associated status parameter, specifying successful reception of the associated data, or the locally detected reason for failure.

4.2 General structure and encoding of PhIDUs and DLPDUs, and related elements of procedure

4.2.1 PhIDU structure and encoding

Each PhIDU consists of Ph-interface-control-information and in some cases one octet of Ph-interface-data (see 4.1.4). When the DLE transmits a DLPDU, it computes a frame check sequence for the DLPDU as specified in 4.2.2, concatenates the DLPDU and the frame check sequence, and transmits the concatenated pair as a sequence of PhIDUs as follows.

a) The DLE issues a single Ph-DATA request primitive with PhICI specifying START-OF-ACTIVITY-2 if sending from the queue, and specifying START-OF-ACTIVITY-11 if sending an Acknowledge or Immediate-reply DLPDU, or if re-transmitting because of missing acknowledge. The request primitive is accompanied by one octet holding the first octet from the DLPDU as Ph-interface-data. After that, the DLE awaits the consequent Ph-DATA confirm primitive.
b) The DLE issues a sequence of Ph-DATA request primitives with PhICI specifying DATA, each accompanied by one octet of the DLPDU as Ph-interface-data, from second to last octet of the DLPDU, and after each Ph-DATA request primitive awaits the consequent Ph-DATA confirm primitive.
c) If the value of V(FCM) – FrameCheckMethod – specifies reduced frame check, the DLE issues a single Ph-DATA request primitive with PhICI specifying DATA, accompanied by one octet holding the computed FCS as Ph-interface-data, and after the Ph-DATA request primitive awaits the consequent Ph-DATA confirm primitive. If the value of V(FCM) – FrameCheckMethod – specifies normal frame check, the DLE issues a sequence of Ph-DATA request primitives with PhICI specifying DATA, each accompanied by one octet of the FCS as Ph-interface-data, from first to last octet of the FCS, and after each Ph-DATA request primitive awaits the consequent Ph-DATA confirm primitive. If the value of V(FCM) – FrameCheckMethod – specifies None frame check, the transmission is finished.
d) The DLE issues a single Ph-DATA request primitive with PhICI specifying END-OF-ACTIVITY, and awaits the consequent Ph-DATA confirm primitive.

It is a task of the implementation to ensure that there are no idle periods between the octets of a transmitted DLPDU.

The DLE forms a received DLPDU by concatenating the sequence of octets received as Ph-interface-data of consecutive Ph-DATA indications, computing a frame check sequence for those received octets as specified in 4.2.2, and compares the received FCS value with the computed, as follows.

1) The DLE received a single Ph-DATA indication primitive with PhICI specifying START-OF-ACTIVITY, accompanied by one octet of the received DLPDU as Ph-interface-data, and initializes its computation of an FCS for the received DLPDU.
2) The DLE receives a sequence of Ph-DATA indication primitives with PhICI specifying DATA, each accompanied by one octet of the received DLPDU as Ph-interface-data, incrementally computes an FCS on the received octet, and concatenates all, or all except the last one or two as specified by V(FCM), of those received octets to form the received DLPDU. During reception, the DLE encodes the DLPDU being received to compute the number of octets forming the DLPDU.
3) When the DLE has received the last Ph-DATA indication, it compares the value(s) (if any – depending on frame check method) of the computed FCS to zero:
   a) if the value(s) is (are) zero, then the DLE reports the reconstructed DLPDU as a correctly received DLPDU suitable for further analysis;
   b) if the value(s) is (are) not zero, the DLE ignores the received DLPDU, and performs no further actions related to the received DLPDU.

4.2.2 Frame check sequence

4.2.2.1 General

The value of the DLE local variable V(FCM) determines which frame check method to use.

The following frame check methods are defined: "Normal", "Reduced" and “None”.

4.2.2.2 Normal frame check method

The "Normal" frame check method uses two frame-check codes, FCA and FCB. The method gives a “Hamming Distance” of 4 for a codeword size of 64 bits. This means that up to three (Hamming Distance minus one) randomly-located error bits within a 64-bit wide window will be detected. Any burst of errors up to 15 bits in length and any error with an odd number of bits in error will be detected.

At the transmitting DLE, the following sequence is followed.

a) Before transmitting the first octet of the DLPDU, clear the two variables FCA and FCB.
b) For each octet of the DLPDU to be sent, exclusive OR the value of the octet to be sent to the value of FCA, and store the result in FCA. Exclusive OR the value of the octet to be sent to the value of FCB, rotate the result one bit left, and store the final result in FCB. This is done in the order in which the octets are sent.
c) When the last octet of the DLPDU has been sent, send the value of FCA, exclusive OR the value of FCA to the value of FCB, rotate the result one bit left, and store the final result in FCB.
d) When FCA has been sent, send the value of FCB.

At the receiving DLE, the following sequence is followed:

a) Before receiving the first octet of the DLPDU, clear the two variables FCA and FCB.
b) For each octet of the DLPDU received, exclusive OR the value of the received octet to the value of FCA, and store the result in FCA. Exclusive OR the value of the received octet to the value of FCB, rotate the result one bit left, and store the final result in FCB. This shall be done in the order in which the octets are received.
c) When the first octet of the FCS has been received, and the normal frame check computation performed, check that the value of FCA is equal to zero.
d) When the last octet of the FCS has been received, and the normal frame check computation performed, check that the value of FCB is equal to zero.

4.2.2.3 Reduced frame check method

The "Reduced" frame check method uses one frame-check code, FC. The method gives a “Hamming Distance” of 2 for the whole DLPDU, which means any single error bit will be detected. A single error burst up to 8 bits in length will also be detected.

At the transmitting DLE, the following sequence is followed.

a) Before transmitting the first octet of the DLPDU, clear the variable FC.
b) For each octet of the DLPDU to be sent, add the value of the octet to be sent to the value of FC, without carry, and store the result in FC.
c) When the last octet of the DLPDU has been sent, send the 2's complement of the value of FC.

At the receiving DLE, the following sequence is followed:

a) Before receiving the first octet of the DLPDU, clear the variable FC.

b) For each octet of the DLPDU received, add the value of the received octet to the value of FC without carry, and store the result in FC.

c) When the FCS has been received, and the normal frame check computation performed, check that the value of FC is equal to zero.

4.2.2.4 None frame check method

The "None" frame check method uses no frame check. This method is only used for IP networks. In this case the DLPDU is data within a frame on the IP network and the IP network specifies the frame check.

4.2.3 Common DLPDU structure, encoding and elements of procedure

4.2.3.1 DLPDU fields

Each DLPDU consists of a Type 4-route field, a Control-status field, a Data-field-format field, and for most DLPDUs a Data field. An FCS field (see 4.2.2) which is used to check the integrity of the received DLPDU can be appended before transmission, and removed after reception.

4.2.3.2 Type 4-route field

4.2.3.2.1 General route field format

The first field in each DLPDU is a Type 4-route field. The Type 4-route field holds a Type 4-route and consists of 2-30 octets, called Type 4-route-elements. Each Type 4-route-element is an octet, holding a 7-bit DL-route-element or Remaining-route-length, and a 1-bit Source/Destination designator. Five different Type 4-route field formats are defined: "Simple", "Extended", "Complex", "Immediate" and “IP”. The Type 4-route field format is indicated by the sequence of Source/Destination designators.

The Source/Destination designator is physically located as bit 8 in the octet. A value of "0" designates "Destination", and a value of "1" designates "Source".

4.2.3.2.2 Simple Type 4-route format

Type 4-route fields of Simple format consist of one destination Type 4-route-element followed by one source Type 4-route-element, as illustrated in Figure 8.

0	**Destination address**
1	**Source address**

IEC

Figure 8 – Simple Type 4-route format

The Destination address identifies the DLE to receive the DLPDU. The Source address identifies the transmitting DLE.

Simple routes are used when sending Confirmed or Unconfirmed DLPDUs holding requests to DLEs of simple class. The DLPDU is of type Unconfirmed if the Destination address is equal to the Broadcast-Node-Address (BNA).

4.2.3.2.3 Extended Type 4-route format

Type 4-route fields of Extended format consist of two destination Type 4-route-elements followed by two source Type 4-route-elements, as illustrated in Figure 9.

0	Destination address
0	Destination address
1	Source address
1	Source address

IEC

Figure 9 – Extended Type 4-route format

The first Destination address identifies the DLE to receive the DLPDU. The second Destination address is used by the DLS-user. The first Source address identifies the transmitting DLE. The second Source address is used by the DLS-user.

Extended routes are used when sending Confirmed or Unconfirmed DLPDUs holding requests to DLEs of normal class. The DLPDU is Unconfirmed if the value of the first Destination address equals BNA.

4.2.3.2.4 Complex Type 4-route format

Type 4-route fields of Complex format consist of more than 2 destination Type 4-route-elements followed by 2 or more source Type 4-route-elements, as illustrated in Figure 10.

0	Destination address
0	Destination address
0	Remaining-route-length
1	Source address
1	Source address

IEC

Figure 10 – Complex Type 4-route format

The first Destination address identifies the DLE to receive the DLPDU. The remaining Destination addresses are used by the DLS-user. The third Type 4-route-element holds the number of Type 4-route-elements following the third Type 4-route-element. The first Source address identifies the transmitting DLE. The remaining source addresses (maybe except the last) are used by the DLS-user.

Complex routes are used when sending Confirmed or Unconfirmed DLPDUs holding requests or Unconfirmed DLPDUs holding responses to DLEs of normal class. The DLPDU is Unconfirmed if the value of the last Source address equals 0, or the value of one of the Destination addresses equals BNA.

4.2.3.2.5 Immediate Type 4-route format

Type 4-route fields of Immediate format consist of one source Type 4-route-element followed by one destination Type 4-route-element, as illustrated in Figure 11.

1	Source address
0	Destination address

IEC

Figure 11 – Immediate Type 4-route format

The Source address identifies the DLE to receive the DLPDU. The Destination address identifies the transmitting DLE. The Source address does NOT identify the transmitting DLE, but the DLE that transmitted the request resulting in this DLPDU. As always, the first octet in the Type 4-route identifies the DLE to receive the DLPDU.

Immediate routes are used when sending Acknowledge or Immediate-reply DLPDUs to DLEs of normal class.

4.2.3.2.6 IP Type 4-route format

Type 4-route fields of IP format consist of more than 2 destination Type 4-route-elements followed by 2 or more source Type 4-route-elements, as illustrated in Figure 12.

0	Destination address
0	Destination address
0	Remaining-route-length
1	Source address
1	Source address

IEC

Figure 12 – IP Type 4-route format

The first Destination address is the IPNetID, which identifies the IP net to receive the DLPDU. The value of IPNetID shall be in the range of 0-127. The values 0, 126 and 127 are reserved for special purposes.

The second Destination address identifies the DLE to receive the DLPDU. The remaining Destination addresses are used by the DLS-user. The third Type 4-route-element holds the number of Type 4-route-elements following the third Type 4-route-element. The first Source address identifies the transmitting IP net. The second Source address identifies the transmitting DLE. The remaining source addresses (maybe except the last) are used by the DLS-user.

Complex routes are used when sending Confirmed or Unconfirmed DLPDUs holding requests or Unconfirmed DLPDUs holding responses to DLEs of normal class. The DLPDU is Unconfirmed if the value of the last Source address equals 0, or the value of one of the Destination addresses equals BNA.

4.2.3.3 Control-status field

The second field in each DLPDU is a Control-status field. This field consists of 1 octet, used by the DLE in conjunction with the Type 4-route field format and the Data-field-format field to determine the DLPDU type. If the DLPDU type is Acknowledge, the Control-status field holds status information to the DLE. The coding of the Control-status field is illustrated in Figure 13.

If the format of the Type 4-route-field is Immediate, the DLPDU is of type Immediate-reply or Acknowledge. The value of the Control-status field indicates (in conjunction with the Data-field-format field) whether the DLPDU is of type Immediate-reply or Acknowledge, according to the following:

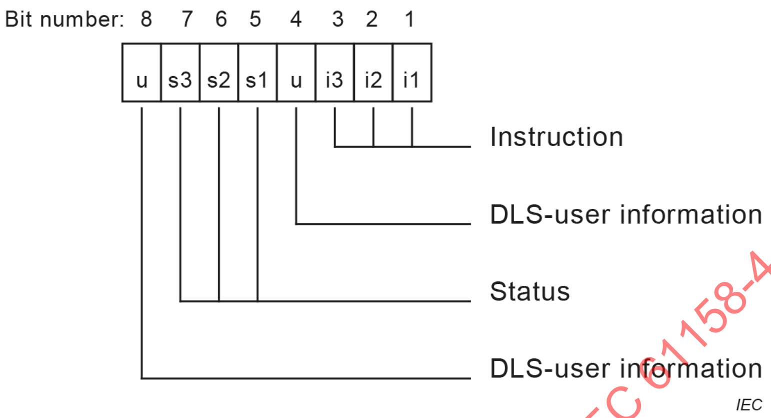

Figure 13 – Control-status format

The range for the Instruction subfield is 0 to 7. The range for the Status subfield is 0 to 7.

The DLPDU is an Acknowledge DLPDU specifying Wait if all of these conditions are fulfilled:

a) the Type 4-route-field format is immediate,
b) the value of the Instruction subfield is greater than 0,
c) the value of the Data-size subfield in the Data-field-format field equals 0,
d) the value of the Status subfield is 4.

The DLPDU is an Acknowledge DLPDU specifying RCL/ACK if all of these conditions are fulfilled:

a) the Type 4-route-field format is immediate,
b) the value of the Instruction subfield is greater than 0,
c) the value of the Data-size subfield in the Data-field-format field equals 0,
d) the value of the Status subfield is 5.

If neither of these two apply, the DLPDU is not an Acknowledge DLPDU. Acknowledge DLPDUs hold information for the DLE, and the receipt of an Acknowledge DLPDU does not result in a DLS-user indication.

4.2.3.4 Data-field-format field

The third field in each DLPDU is a Data-field-format field.

If the Type 4-route-field in the DLPDU defines an IP Type 4-route format and the size of the Data field is more than 63 octets, then this field consists of two octets. In all other cases this field consists of one octet.

If this field consists of one octet, it holds a 6-bit subfield indicating Data-size, and a 2-bit subfield for DLS-user information, as illustrated in Figure 14.

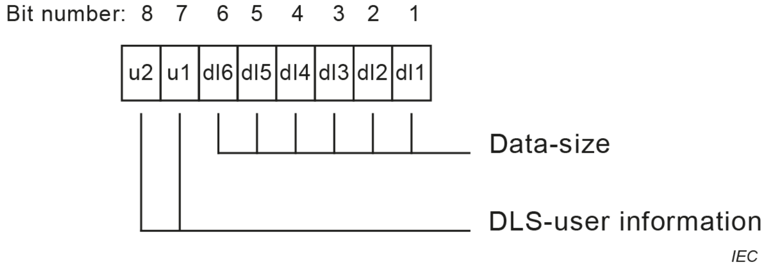

Figure 14 – Data-field-format, one octet

The range for the Data-size subfield is 0 to 63 for a one octet Data field format. The value indicates the number of octets in the Data-field of the DLPDU, and does not include the FCS octet(s).

If this field consists of two octets, it holds a 14-bit subfield indicating Data-size, and a 2-bit subfield for DLS-user information, as illustrated in Figure 15.

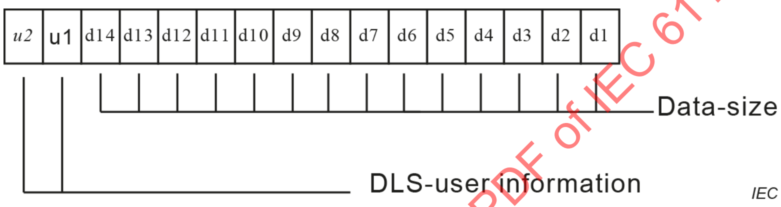

Figure 15 – Data field format, two octets

The range for the Data-size subfield is 63 to 1280 for a two octet Data field format. The value indicates the number of octets in the Data-field of the DLPDU, and does not include the FCS octet(s).

4.2.3.5 Data field

The next field in the DLPDU is the Data field. The size and DLS-user interpretation of this field is indicated in the Data-field-format and Control-status fields.

4.3 DLPDU-specific structure, encoding and elements of procedure

4.3.1 DLPDU types

Table 1 – Summary structure of DLPDUs

DLPDU type	Type 4-route format	Destination Node-addresses	Last Source address	Control-status	Data size	Data
Confirmed	Simple	≠ BNA	≠ 0	Any	> 2	user data
Confirmed	Extended	≠ BNA	≠ 0	Any	> 2	user data
Confirmed	Complex	≠ BNA	≠ 0	Any	> 2	user data
Confirmed	IP	≠ BNA	≠ 0	Any	> 2	user data
Unconfirmed	Simple	= BNA	≠ 0	Any	> 2	user data
Unconfirmed	Extended	= BNA	≠ 0	Any	> 2	user data
Unconfirmed	Complex	= BNA	≠ 0	Any	> 2	user data
Unconfirmed	Complex	≠ BNA	= 0	Any	≥ 0	user data
Unconfirmed	IP	= BNA	≠ 0	Any	> 2	user data
Unconfirmed	IP	≠ BNA	= 0	Any	≥ 0	user data
Immediate-reply	Immediate	Any	Any	Any	≥ 0	user data
Acknowledge	Immediate	Any	Any	= Wait / RCL/ACK	= 0	-

The DLPDU type is indicated by the Type 4-route format, the contents of the Destination Node-addresses in the Type 4-route, the contents of the last Source address in the Type 4-route, the contents of Control-status, and the contents of the data size subfield of Data-field-format, as shown in Table 1.

When the value in the column Destination Node-addresses is "≠ BNA", it means that none of the Node-addresses in the Type 4-route field are = BNA. When the value in the column Destination Node-addresses is "= BNA", it means that at least one of the Node-addresses in the Type 4-route field is = BNA.

4.3.2 Confirmed DLPDU

4.3.2.1 Use of confirmed DLPDUs

A Confirmed DLPDU is used

a) to request the transfer of a limited amount of transparent user data from another DLS-user to the requesting DLS-user;

b) to transfer a limited amount of transparent user data from the requesting DLS-user to another DLS-user;

c) to transfer a limited amount of transparent user data from the requesting DLS-user to another DLS-user, and at the same time request the transfer of the same amount of transparent user data from that other DLS-user to the requesting.

4.3.2.2 Structure of confirmed DLPDUs

The structure of confirmed DLPDUs is shown in Table 2.

Table 2 – Structure of confirmed DLPDUs

Route format	Destination node-addresses	Last source node address	Control-status	Data size	Data
Simple	≠ BNA	≠ 0	DLS-user info	> 2	DLS-user data
Extended	≠ BNA	≠ 0	DLS-user info	> 2	DLS-user data
Complex	≠ BNA	≠ 0	DLS-user info	> 2	DLS-user data
IP	≠ BNA	≠ 0	DLS-user info	> 2	DLS-user data

4.3.2.3 Sending the confirmed DLPDU

A confirmed DLPDU is selected for transmission on the Link when the DLPDU is the first in the queue, and the DLE receives the Virtual Link-access token (except for Full duplex). Once selected, the DLPDU is removed from the queue, and transmission of the DLPDU commences. If the receipt of the DLPDU must be acknowledged, according to the rules described in 4.1.3.2, the DLPDU shall be transmitted until either

a) an Immediate-reply DLPDU is received, or

b) an Acknowledge DLPDU is received, or

c) the original transmission and the permitted maximum number or transmission retries, V(MRC), have all failed to elicit one of the permissible reply DLPDUs.

In addition to the above, the transmitting DLE shall act according to the rules described in 4.1.3.8.

4.3.2.4 Receiving the confirmed DLPDU

A received confirmed DLPDU shall be treated as follows by the receiving DLE.

NOTE The next alternative is used to detect the reception of a duplicated confirmed DLPDU resulting from an immediate retry by the current token-holding DLE, which itself was probably caused by an error detected during receipt of the earlier Acknowledge or Immediate-reply DLPDU.

a) If

   1) the receipt of the DLPDU shall be acknowledged, according to the rules described in 4.1.3.2, and

   2) no PhE LINK-IDLE indication primitive, specifying the Link has been idle for 40 or more bit periods, has been received since the last DLPDU was received, and

   3) the contents of the Type 4-route field in the just received DLPDU is exactly the same as the contents of the Type 4-route field in the last DLPDU received,

   then the receiving DLE shall

      i) retransmit the prior-transmitted acknowledge or immediate-reply DLPDU immediately, and

      ii) discard the received DLPDU and not forward it to the DLS-user.

b) If a) does not apply, the receiving DLE shall act according to the rules described in 4.1.3.6.

4.3.3 Unconfirmed DLPDU

4.3.3.1 Use of Unconfirmed DLPDUs

An Unconfirmed DLPDU is used

a) by a requesting DLS-user to transfer a limited amount of transparent user data from the requesting DLS-user to one or more other DLS-users;

b) by a responding DLS-user to transfer a limited amount of transparent user data from the responding DLS-user to the requesting, as a response to a received confirmed DLPDU;

c) by a responding DLS-user to acknowledge the receipt of a limited amount of transparent user data from the requesting DLS-user, as a response to a received confirmed DLPDU.

4.3.3.2 Structure of unconfirmed DLPDUs

The structure of unconfirmed DLPDUs is shown in Table 3.

Table 3 – Structure of unconfirmed DLPDUs

Route format	Destination node-addresses	Last source node-address	Control-status	Data size	Data
Simple	= BNA	≠ 0	DLS-user info	> 2	DLS-user data
Extended	= BNA	≠ 0	DLS-user info	> 2	DLS-user data
Complex	= BNA	≠ 0	DLS-user info	> 2	DLS-user data
Complex	≠ BNA	= 0	DLS-user info	≥ 0	DLS-user data
IP	= BNA	≠ 0	DLS-user info	> 2	DLS-user data
IP	≠ BNA	= 0	DLS-user info	≥ 0	DLS-user data

4.3.3.3 Sending the unconfirmed DLPDU

An unconfirmed DLPDU is selected for transmission on the Link when the DLPDU is the first in the queue, and the DLE receives the Virtual Link-access token (except for Full duplex). Once selected, the DLPDU is removed from the queue, and transmission of the DLPDU commences. If the receipt of the DLPDU must be acknowledged, according to the rules described in 4.1.3.2, the DLPDU shall be transmitted until either

a) an Acknowledge DLPDU is received, or

b) the original transmission and the permitted maximum number or transmission retries, V(MRC), have all failed to elicit one of the permissible reply DLPDUs.

In addition to the above, the transmitting DLE shall act according to the rules described in 4.1.3.8.

4.3.3.4 Receiving the unconfirmed DLPDU

A received unconfirmed DLPDU shall be treated as follows by the receiving DLE.

NOTE The next alternative is used to detect the reception of a duplicated Unconfirmed DLPDU resulting from an immediate retry by the current token-holding DLE, which itself was probably caused by an error detected during receipt of the earlier Acknowledge or Immediate-reply DLPDU.

a) If

   1) the receipt of the DLPDU shall be acknowledged, according to the rules described in 4.1.3.2, and

   2) no PhE LINK-IDLE indication primitive, specifying the Link has been idle for 40 or more bit periods, has been received since the last DLPDU was received, and

   3) the contents of the Type 4-route field in the just received DLPDU is exactly the same as the contents of the Type 4-route field in the last DLPDU received,

   then the receiving DLE shall

      i) retransmit the prior-transmitted Acknowledge DLPDU immediately, and

      ii) discard the received DLPDU and not forward it to the DLS-user.

b) If a) does not apply, the receiving DLE shall act according to the rules described in 4.1.3.6.

4.3.4 Acknowledge DLPDU

4.3.4.1 Use of acknowledge DLPDUs

An Acknowledge DLPDU is used

a) by a responding DLE to acknowledge the receipt of a Confirmed or Unconfirmed DLPDU;

b) by a responding DLS-user to acknowledge the receipt of a Confirmed DLPDU (the Acknowledge DLPDU is transmitted by the DLE as a result of a request service primitive from the local DLS-user specifying DLSDU type acknowledge).

4.3.4.2 Structure of acknowledge DLPDUs

The structure of acknowledge DLPDUs is shown in Table 4.

Table 4 – Structure of acknowledge DLPDU

Route format	Destination node-addresses	Last source node-address	Control-status	Data size	Data
Immediate	Any	Any	= Wait/RCL/ACK	= 0	-

4.3.4.3 Sending the acknowledge DLPDU

An Acknowledge DLPDU is transmitted as an immediate reply on the Link to acknowledge the receipt of a Confirmed or Unconfirmed DLPDU, according to the rules described in 4.1.3.6. Acknowledge DLPDUs shall never be retransmitted.

4.3.4.4 Receiving the acknowledge DLPDU

A received Acknowledge DLPDU shall be treated according to the rules described in 4.1.3.8. The receipt of Acknowledge DLPDUs shall never be acknowledged.

4.3.5 Immediate-reply DLPDU

4.3.5.1 Use of Immediate-reply DLPDUs

An Immediate-reply DLPDU is used

a) by a responding DLS-user to transfer a limited amount of transparent user data from the responding DLS-user to the requesting, as a response to a received confirmed DLPDU;

b) by a responding DLS-user to confirm the receipt of a limited amount of transparent user data from the requesting DLS-user, as a response to a received confirmed DLPDU.

4.3.5.2 Structure of immediate-reply DLPDU

Table 5 – Structure of immediate-reply DLPDU

Route format	Destination node-addresses	Last source node-address	Control-status	Data size	Data
Immediate	Any	Any	≠ Wait/RCL/ACK	≥ 0	DLS-user-data

4.3.5.3 Sending the immediate-reply DLPDU

An immediate-reply DLPDU is transmitted as an immediate reply on the Link to acknowledge the receipt of a confirmed DLPDU, according to the rules described in 4.1.3.6. Immediate-reply DLPDUs shall never be retransmitted.

4.3.5.4 Receiving the immediate-reply DLPDU

A received immediate-reply DLPDU shall be treated according to the rules described in 4.1.3.8. The receipt of immediate-reply DLPDUs shall never be acknowledged.

4.4 DL-service elements of procedure

4.4.1 Receipt of a DL-UNITDATA request primitive

4.4.1.1 DLE request functions

When the DLE receives a DL-UNITDATA request primitive from the local DLS-user, the DLE shall determine whether the request primitive holds a response for immediate transmission, or a request or response to be queued.

a) If
   1) the DLE is waiting for a request or response primitive from the local DLS-user, according to the rules described in 4.1.3.6, and
   2) the contents of the user-specified Destination DL-route parameter in the request is the same as the contents of the Source DL-route parameter in the indication primitive generated by this DLE after receipt of the Confirmed DLPDU from the remote DLE,

   the request primitive holds a response for immediate transmission. The DLE shall form and immediately transmit an Immediate-reply DLPDU.
b) If a) does not apply, then the request primitive holds a request or a response to be queued. The DLE shall form a Confirmed or Unconfirmed DLPDU.

   If the request is accepted, the DLE shall append the DLPDU to the queue for transmission at the first opportunity. The DLPDU is appended to the queue at a position based on the user-specified priority. The specified value can be any integral number from 0 to 255. The DLPDU is placed in front of all DLPDUs currently in the queue having a lower priority, where 255 indicates the highest priority.

The DLE shall create and start a retry timer with duration based on the user-specified Maximum retry time. If the specified value is other than zero, then the duration of this timer shall be equal to that user-specified Maximum retry time; otherwise, the duration shall be 2 000 ms. DL-management can override this default duration. The retry timer shall be associated with the DLPDU.

The DLE shall create and clear a retry counter. The retry counter shall be associated with the DLPDU.

The DLE shall associate the user-specified parameters DL-route and Local source ID with the DLPDU, for later use in DL-route conversion.

If the request is rejected, and the user-specified DL-route specifies Confirmed, the DLE shall immediately report the reason for failure in a DL-UNITDATA indication primitive.

4.4.1.2 Forming an Immediate-reply DLPDU

The forming of an Immediate-reply DLPDU is described in the following.

The first Node-address in the Destination DL-route is copied to the address subfield of the first Type 4-route-element, and the Source/Destination designator in that element is set to TRUE. The Node-address of the DLE is copied to the address field of the second element, and the Source/Destination designator in that element is set to FALSE.

The user-specified Control-status parameter is stored in the Control-status (C-S) field of the DLPDU.

The user-specified Data-field-format parameter is stored in the Data-field-format (DFF) field of the DLPDU.

The user-specified Data unit (DLSDU) (the size of which is indicated in bit 1-6 of the Data-field-format field) is stored in the Data field of the DLPDU.

4.4.1.3 Forming a Confirmed or Unconfirmed DLPDU

The forming of a Confirmed or Unconfirmed DLPDU is described in the following.

The DLE's Node-address shall be inserted in front of all other addresses in the source DL-route.

Request Type 4-route generation encodes the resulting DL-route into a Type 4-route with Simple, Extended, Complex or IP format, and stores the result in the Type 4-route field of the DLPDU. Request Type 4-route generation is described in 4.5.2.

The user-specified Control-status parameter is stored in the Control-status (C-S) field of the DLPDU.

The user-specified Data-field-format parameter is stored in the Data-field-format (DFF) field of the DLPDU.

The user-specified Data unit (DLSDU) (the size of which is indicated in bit 1-6 of the Data-field-format field) is stored in the Data field of the DLPDU.

4.4.2 Receipt of a DL-UNITDATA response primitive

4.4.2.1 DLE response functions

When the DLE receives a DL-UNITDATA response primitive from the local DLS-user, the DLE shall determine whether the response primitive shall result in the immediate transmission of an Acknowledge DLPDU.

a) If
   1) the DLE is waiting for a request or response primitive from the local DLS-user, according to the rules described in 4.1.3.6, and
   2) the contents of the user-specified Destination DL-route parameter in the response is the same as the contents of the Source DL-route parameter in the indication primitive generated by this DLE after receipt of the Confirmed DLPDU from the remote DLE,

   then the response primitive shall result in the immediate transmission of an Acknowledge DLPDU.
b) If a) does not apply, then the response primitive is ignored by the DLE. In this situation, the DLE has already sent an autonomously generated Acknowledge DLPDU.

If the response primitive holds an acknowledge for immediate transmission:

The DLE shall form and immediately transmit an Acknowledge DLPDU.

4.4.2.2 Forming an Acknowledge DLPDU

The forming of an Acknowledge DLPDU is described in the following.

The parameters used to form an Acknowledge DLPDU are those from the indication causing the request specifying acknowledge.

The first Node-address in the Source DL-route parameter of the indication is copied to the address subfield of the first Type 4-route-element, and the Source/Destination designator in